

# **MS Access 2003**

## **základní kurz**

### **CZ - EN**

Veškerá práva vyhrazena.

Tento materiál je součástí duševního vlastnictví společnosti GOPAS a.s.  
a jeho šíření podléhá autorskému zákonu. Nesmí být reprodukován či jinak využíván  
v jakékoliv formě ani jako celek, ani žádná jeho část bez písemného souhlasu  
společnosti GOPAS a.s.



# Zaměření kurzu

## MS Access 2003 - základní kurz

Kurz je určen pro uživatele, kteří si potřebují osvojit základy práce s databázovým programem MS Access.

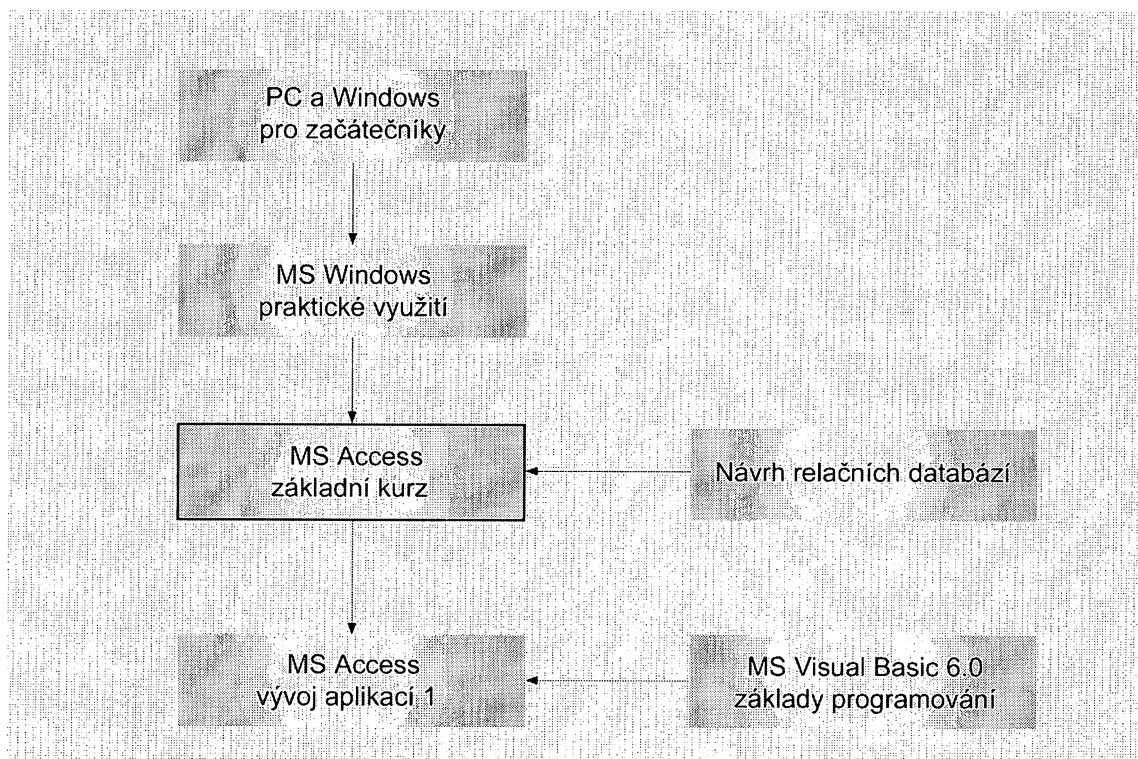
Po absolvování kurzu byste měli umět vytvářet relační databáze, plnit je daty, provádět jejich vyhodnocení a zpracování a vytvářet jednoduché aplikace.

U účastníků tohoto kurzu se předpokládají vstupní znalosti alespoň v rozsahu kurzu **PC a Windows - pro začátečníky a MS Windows - praktické využití**.

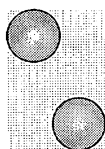
Doporučujeme absolvování kurzu **Návrh relačních databází**, i když tento kurz můžete absolvovat i později.



# Doporučené návaznosti



Doporučený návazný kurz MS Access -vývoj aplikací je určen pro vývojáře, který chce využít všechny programátorské prostředky MS Access včetně MS Visual Basic a moderního přístupu k datům pomocí ActiveX Data Objects. Pro tento kurz je třeba mít základní znalosti programování ve Visual Basicu.



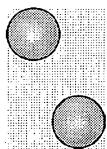
# Obsah

|                                                                                       |    |
|---------------------------------------------------------------------------------------|----|
| Úvod do databází.....                                                                 | 9  |
| Úvod.....                                                                             | 10 |
| Vymezení pojmu databáze.....                                                          | 10 |
| Rozlišujeme pojmy .....                                                               | 10 |
| Tabulky.....                                                                          | 12 |
| Dotazy (Queries).....                                                                 | 12 |
| Formuláře (Forms).....                                                                | 12 |
| Sestavy (Reports).....                                                                | 13 |
| Datové stránky (Pages).....                                                           | 13 |
| Makra (Macros).....                                                                   | 13 |
| Moduly (Modules).....                                                                 | 13 |
| Seznamujeme se s výukovou databází Northwind.....                                     | 14 |
| Navrhování tabulek.....                                                               | 17 |
| Základní režimy zobrazování tabulky.....                                              | 18 |
| Tvorba tabulky.....                                                                   | 18 |
| Návrhové zobrazení tabulky (Design View).....                                         | 19 |
| Datový typ (Data Type).....                                                           | 20 |
| Další důležité vlastnosti polí.....                                                   | 21 |
| Indexy v tabulce, primární klíč tabulky.....                                          | 24 |
| Ověřovací pravidla (Validation Rules).....                                            | 26 |
| Průvodce vyhledáváním (Lookup Wizard).....                                            | 27 |
| Průvodce vyhledáváním pro zadávání hodnot do pole ze seznamu zadaného uživatelem..... | 27 |
| Průvodce vyhledáváním pro zadávání hodnot z pole jiné tabulky.....                    | 28 |
| Zadávání hodnot do polí typů OLE Object.....                                          | 30 |
| Zadávání hodnot do polí typů Hyperlink.....                                           | 31 |
| Kopírování tabulek v databázi.....                                                    | 32 |
| Datový list.....                                                                      | 35 |
| Práce se záznamy v režimu datového listu tabulky Accessu.....                         | 36 |
| Editace záznamu.....                                                                  | 36 |
| Přidání nového záznamu.....                                                           | 36 |
| Odstranění záznamu.....                                                               | 36 |
| Řazení záznamů podle jednoho pole.....                                                | 36 |
| Skrývání polí.....                                                                    | 37 |
| Ukotvování polí.....                                                                  | 37 |
| Práce s vnořeným datovým listem (Subdatasheet).....                                   | 38 |
| Filtrování záznamů v tabulce.....                                                     | 39 |
| Filtrování podle výběru (resp. mimo výběr).....                                       | 39 |
| Filtrování podle formuláře.....                                                       | 39 |
| Rozšířený filtr či řazení.....                                                        | 40 |
| Zápis kritérií.....                                                                   | 40 |
| Řazení vyfiltrovaných dat.....                                                        | 40 |
| Praktická zásada pro filtrování tabulek.....                                          | 40 |
| Tabulky – shrnutí a zobecnění základních pojmů.....                                   | 42 |
| Relace.....                                                                           | 47 |
| Relace a jejich typy.....                                                             | 48 |
| Tvorba relací mezi tabulkami v Accessu.....                                           | 50 |
| Referenční integrita (Referential Integrity) pro relaci 1:N.....                      | 52 |
| Důsledky referenční integrity.....                                                    | 53 |
| Další vlastnosti relace s referenční integritou.....                                  | 53 |
| Normalizace.....                                                                      | 54 |
| První normální forma.....                                                             | 54 |
| Druhá normální forma.....                                                             | 55 |
| Třetí normální forma.....                                                             | 56 |
| Další normální formy.....                                                             | 56 |

# Obsah



|                                                                      |            |
|----------------------------------------------------------------------|------------|
| Čtvrtá normální forma.....                                           | 57         |
| Pátá normální forma.....                                             | 57         |
| <b>Export a import.....</b>                                          | <b>59</b>  |
| Datové formáty pro export, import a propojení .....                  | 60         |
| Export dat .....                                                     | 60         |
| Import dat, propojení tabulek .....                                  | 60         |
| Export do jiné databáze Accessu .....                                | 61         |
| Import z jiné databáze Accessu .....                                 | 61         |
| Import z tabulky Excelu .....                                        | 62         |
| Proč a kdy převádět data z Excelu do Accessu .....                   | 63         |
| Jazyk XML .....                                                      | 64         |
| Export do formátu XML .....                                          | 64         |
| Import dat ve formátu XML .....                                      | 65         |
| Rozdělení databáze.....                                              | 66         |
| Postup rozdělení .....                                               | 66         |
| <b>Úvod do tvorby výběrových dotazů .....</b>                        | <b>69</b>  |
| Dotazy (Queries) .....                                               | 70         |
| Co je to dotaz .....                                                 | 70         |
| Základní druhy dotazů .....                                          | 70         |
| Režimy zobrazení dotazu .....                                        | 70         |
| Práce s dotazem v návrhovém zobrazení .....                          | 71         |
| Zobrazování polí ve výsledku dotazu .....                            | 72         |
| Řazení .....                                                         | 72         |
| Zadávání kritérií dotazu .....                                       | 72         |
| Výrazy Accessu používané v dotazu .....                              | 73         |
| Často používané operátory .....                                      | 73         |
| Operátor Like v kritériální mřížce dotazu .....                      | 74         |
| Často používané logické operátory .....                              | 74         |
| Operátor pro spojování řetězců .....                                 | 74         |
| Precedence vyhodnocování určuje pořadí vyhodnocování operátorů ..... | 75         |
| Časté operandy výrazů používané v dotazech: .....                    | 76         |
| Funkce .....                                                         | 76         |
| Tvorba nových polí v dotazech .....                                  | 79         |
| <b>Základní typy výběrových dotazů .....</b>                         | <b>83</b>  |
| Parametrické dotazy .....                                            | 84         |
| Datové typy parametrů .....                                          | 85         |
| Reakce Accessu na nedodržení typu parametru při jeho zadávání .....  | 86         |
| Dotaz vracející několik největších či nejmenších hodnot .....        | 88         |
| Vyloučení duplicit z výsledku dotazu .....                           | 89         |
| Souhrnné dotazy .....                                                | 90         |
| Definice kritérií pro souhrnné dotazy .....                          | 91         |
| Všeobecné kritérium (Where) .....                                    | 91         |
| Kritérium pro skupiny .....                                          | 92         |
| Volba Expression v řádku Total souhrnného dotazu .....               | 92         |
| Vlastnosti spojení tabulek .....                                     | 94         |
| Základní typy spojení tabulek v Accessu .....                        | 94         |
| Spojení tabulky do sebe sama .....                                   | 96         |
| <b>Akční dotazy .....</b>                                            | <b>101</b> |
| Akční dotazy (Action Queries) .....                                  | 102        |
| Vytvářecí dotaz (Make-table Query) .....                             | 103        |
| Aktualizační dotazy (Update Queries) .....                           | 105        |
| Přídávací dotazy (Append Queries) .....                              | 107        |
| Odstraňovací dotazy (Delete Queries) .....                           | 108        |



# Obsah

|                                                                                   |     |
|-----------------------------------------------------------------------------------|-----|
| Formuláře .....                                                                   | 113 |
| Použití formulářů .....                                                           | 114 |
| Tvorba formulářů .....                                                            | 114 |
| Rozdělení formulářů podle toho, zda jsou či nejsou napojeny na datový zdroj ..... | 115 |
| Způsoby zobrazení formuláře .....                                                 | 116 |
| Základní součásti návrhového zobrazení formuláře .....                            | 117 |
| Příklad .....                                                                     | 118 |
| Základní součásti formuláře ve formulářovém zobrazení .....                       | 120 |
| Základní techniky práce v návrhovém zobrazení formuláře .....                     | 120 |
| Základní vlastnosti formuláře .....                                               | 122 |
| Karta Format .....                                                                | 122 |
| Karta Data .....                                                                  | 123 |
| Karta Event .....                                                                 | 123 |
| Karta Other .....                                                                 | 123 |
| Ovládací prvky formuláře .....                                                    | 127 |
| Vlastnosti ovládacích prvků .....                                                 | 128 |
| Některé důležité ovládací prvky formuláře .....                                   | 130 |
| Seznamy a pole se seznamem (List Box, Combo Box) .....                            | 131 |
| Jak se liší seznam (List Box) od pole se seznamem (Combo Box) .....               | 131 |
| Tlačítko (Command Button) .....                                                   | 132 |
| Kalendář (Calendar) .....                                                         | 133 |
| Podformuláře .....                                                                | 143 |
| Podformuláře (Subforms) .....                                                     | 144 |
| Závislosti objektů .....                                                          | 148 |
| Objekty, pro které nelze zobrazit závislosti .....                                | 148 |
| Zobrazení závislostí .....                                                        | 148 |
| Dědičnost vlastností .....                                                        | 149 |
| Kontingenční tabulky .....                                                        | 151 |
| Kontingenční tabulky (Pivot Tables) Access 2003 .....                             | 152 |
| Postup tvorby .....                                                               | 153 |
| Změna struktury kontingenční tabulky .....                                        | 154 |
| Získání souhrnů v kontingenční tabulce .....                                      | 154 |
| Filtrování dat v kontingenční tabulce .....                                       | 157 |
| Počítání nových sloupců v kontingenční tabulce .....                              | 158 |
| Zobrazení několika prvních či posledních položek .....                            | 158 |
| Kontingenční graf .....                                                           | 159 |
| Datové stránky .....                                                              | 165 |
| Datové stránky (Pages) .....                                                      | 166 |
| Tvorba datové stránky .....                                                       | 166 |
| Způsoby zobrazení datové stránky .....                                            | 166 |
| Návrhové zobrazení datové stránky .....                                           | 167 |
| Tvorba datové stránky v návrhovém zobrazení .....                                 | 169 |
| Vytvoření nadřazeného oddílu .....                                                | 169 |
| Tvorba polí obsahujících souhrnné údaje .....                                     | 170 |
| Vlastnosti seskupování .....                                                      | 171 |
| Důležité součásti panelu Navigation .....                                         | 172 |
| Tabulka sady Office (Office Spreadsheet) .....                                    | 172 |
| Kontingenční tabulky (Pivot Tables) v datových stránkách .....                    | 173 |
| Sestavy .....                                                                     | 177 |
| Sestavy (Reports) .....                                                           | 178 |
| Charakteristika .....                                                             | 179 |
| Tvorba sestavy .....                                                              | 179 |
| Průvodce sestavou (Report Wizard) .....                                           | 180 |
| Základní zobrazení sestavy .....                                                  | 182 |

# Obsah



|                                                |     |
|------------------------------------------------|-----|
| Návrhové zobrazení – časté operace .....       | 182 |
| Řazení a seskupování v sestavě.....            | 184 |
| Vlastnost Group On .....                       | 184 |
| Vlastnost Keep Together .....                  | 185 |
| Tipy a triky pro práci se sestavami.....       | 185 |
| Snímky sestavy .....                           | 187 |
| Výhody používání snímků .....                  | 187 |
| Zhotovení snímku .....                         | 187 |
| Elementy správy databáze .....                 | 193 |
| Otevření databáze.....                         | 194 |
| Vstupní heslo.....                             | 195 |
| Převod databáze do různých verzí Accessu ..... | 196 |
| Komprimace a oprava databáze .....             | 196 |





# Úvod do databází

## *Přehled témat*

- vymezení pojmu databáze
- charakteristika základních objektů Databáze MS Access
- seznámení s výukovou databází Northwind



## Úvod

**Databáze** je souhrn dat uzpůsobený k rychlému vyhledávání informací v něm a také k úpravě těchto dat. Program MS Access je databázový program určený k vytváření malých databází a k práci s nimi.

Termín „malá databáze“ může být pro začátečníka zavádějící. V nápovědě MS Accessu si lze přečíst, že velikost databáze Accessu může být až 2 gigabyte, což je obrovské číslo. V databázovém světě jsou však trochu jiná měřítka (vždyť množství informací, které je dnes nutné uchovávat, je svými nároky na paměťovou kapacitou mnohonásobně větší) a v nich databáze Accessu opravdu patří k malým. Access je však svými schopnostmi dostačující k tvorbě databází určených pro potřeby jednotlivců i malých pracovních skupin. Pokud se požadavky na kapacitu databáze u těchto subjektů zvětší, je možno začít ukládat data v databázích MS SQL serveru, které jsou již většího rozsahu, a prostřednictvím Accessu (ano, opět přijde ke slovu Access) s těmito databázemi spolupracovat.

Program MS Access je součástí MS OFFICE. Proto není udivující, že dovede s ostatními programy tohoto programového balíku spolupracovat. Tato spolupráce se především týká přenosu objektů metodikou OLE mezi aplikacemi.

Databáze Accessu je automaticky uzpůsobená ke sdílení, takže s ní může současně pracovat více uživatelů.

Pomocí verzí Accessu 2000, 2002 (XP) a 2003 je již možné vytvářet pro data umístěná v databázích těchto aplikací WWW stránky. Data zobrazovaná těmito stránkami lze potom Internet Explorerem prohlížet i upravovat.

Tato příručka popisuje základní rysy Accessu a jeho databází. Je orientována na verzi Access 2003, ale její obsah lze vynecháním novinek této verze použít i pro výuku Accessu 2000 a 2002.

## Vymezení pojmu databáze

Databáze obecně - je to souhrn efektivně uložených dat, s nimiž lze snadno pracovat (získávání potřebných informací, úpravy dat apod.).

Databáze Excelu - každá tabulka s názvy sloupců.

Databáze Accessu (je ve svém konečném projevu relační databází). Tvoří ji tabulky, mezi nimiž mohou existovat vzájemné vztahy (relace), a souhrn dalších objektů určených pro práci s daty. Těmito objekty jsou dotazy, formuláře, sestavy, datové stránky, makra a moduly.

## Rozlišujeme pojmy

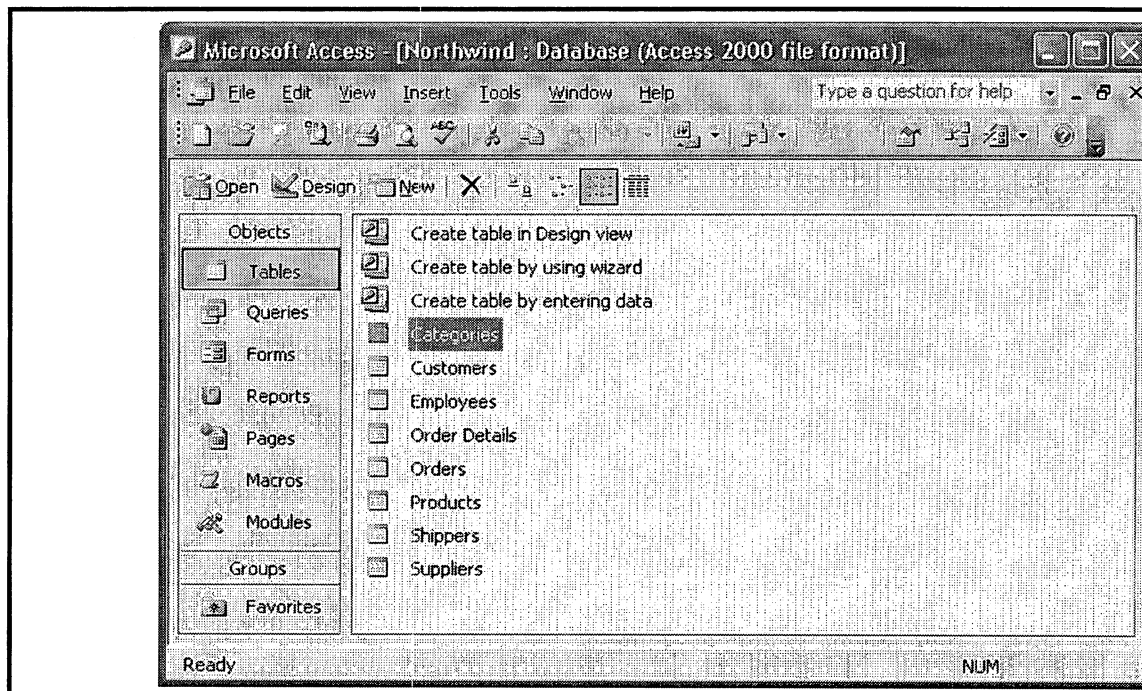
**Databáze**, což je v MS Accessu soubor nejčastěji s příponou mdb, obsahující data a všechny další objekty určené pro práci s těmito daty.

**Databázová aplikace**, což je samotný program MS Access, kterým vytváříme a modifikujeme databáze.

Databáze spolu s příslušnou aplikací tvoří dohromady **databázový systém**.



# Charakteristika základních objektů databáze MS Access



U databázové aplikace můžeme rozlišit část back-end a front-end.

**Část back-end** slouží pro fyzickou práci s daty obvykle umístěnými na pevném disku. V Accessu ji tvoří databázový stroj Microsoft Jet.

**Část front-end** vytváří uživatelské rozhraní pro práci s daty (formuláře a sestavy)

Někdy pod pojmem **back-end** databáze Accessu rozumíme tabulky a pod pojmem front-end to ostatní. Databázi Accessu lze totiž rozdělit na tyto dvě části (příkaz **Tools/Database Utilities/Database Splitter**). Část back-end umístíme na server a je pro všechny uživatele místní sítě společná. Každý jednotlivý uživatel si však dělá svoji část front-end, svoje uživatelské rozhraní pro práci s daty, na vlastním počítači.

Otevře-li uživatel v programu MS Access databázi, objeví se okno databáze. V jedné instanci Accessu může uživatel otevřít pouze jednu databázi (jeden soubor \*.mdb).

V okně databáze lze volit **základní objekty**:

## Objekt

Tabulky (Tables)  
Dotazy (Queries)  
Formuláře (Forms)  
Sestavy (Reports)  
Stránky (Pages)  
Makra (Macros)  
Moduly (Modules)

## Charakteristika

Datová základna databáze  
Pro získávání dat z tabulek a modifikaci dat  
Slouží pro uživatelskou práci s daty  
Tisk dat  
Práce s daty na Internetu  
Automatizace práce s daty  
Programy, které provádějí náročné operace s daty

# Tabulky

|   | Customer ID | Company Name                       | Contact Name       |   |
|---|-------------|------------------------------------|--------------------|---|
| ▶ | ALFKI       | Alfreds Futterkiste                | Maria Anders       | S |
| + | ANATR       | Ana Trujillo Emparedados y helados | Ana Trujillo       | O |
| + | ANTON       | Antonio Moreno Taquería            | Antonio Moreno     | O |
| + | AROUT       | Around the Horn                    | Thomas Hardy       | S |
| + | BERGS       | Berglunds snabbköp                 | Christina Berglund | O |
| + | BLAUS       | Blauer See Delikatessen            | Hanna Moos         | S |
| + | BLONP       | Blondel père et fils               | Frédérique Citeaux | M |
| + | BOLID       | Bólido Comidas preparadas          | Martín Sommer      | O |
| + | BONAP       | Bon app'                           | Laurence Lebihan   | O |
| + | BOTTM       | Bottom-Dollar Markets              | Elizabeth Lincoln  | A |

Record: 1 of 91

Pořadové číslo  
aktivního záznamu

Počet záznamů



## Tabulky

Tvoří datovou základnu databáze. Všechny další objekty databáze čerpají, ať přímo, či nepřímo, data z tabulek.

### Základní pojmy:

**Pole (Field)** – sloupec (Column) tabulky.

**Záznam (Record)** – řádek (Row) tabulky.

**Primární klíč (Primary Key)** – primární klíč je tvořen jedním či více poli tabulky. Jeho hodnota jednoznačně určuje každý záznam tabulky.

**Aktivní záznam** – je v tabulce aktivován. Jeho číslo je uvedeno v navigačním ovládacím prvku tabulky. Tento záznam lze editovat, popřípadě smazat.

## Dotazy (Queries)

Jsou to příkazy pro získávání dat požadovaných vlastností z databáze (výběrové dotazy), či příkazy pro změnu uložených dat (akční dotazy).

## Formuláře (Forms)

Zobrazují data (získaná z tabulek či výběrových dotazů) požadovaným, pro praktické účely výhodným způsobem. V tomto zobrazení lze též data modifikovat i přidávat. Ve formulářích lze dělat i dodatečné výpočty s daty. Slouží tedy pro uživatelskou práci s daty.

Obsah formuláře si může tvůrce sám nadefinovat. U ovládacích prvků (tlačítek, seznamů), umístěných na formulář, mohou nastat různé události (klepnutí na tlačítko, aktivace položky v seznamu, přechod na nový záznam apod.). Jako reakce na tuto událost (obsluha události) může začít nějaká činnost, kterou autor formuláře předepíše (spuštění makra či programu Visual Basicu for Applications).



### **Sestavy (Reports)**

Slouží k vytisknutí dat z tabulek, či dat poskytnutých dotazy. V sestavách lze k tisku seskupovat záznamy podle zadaných kritérií (datum, stejných několik prvních písmen názvu, apod.).

### **Datové stránky (Pages)**

Umožňují zobrazovat i modifikovat data pomocí Internetových prohlížečů, čímž zpřístupní zvolená data síti WWW.

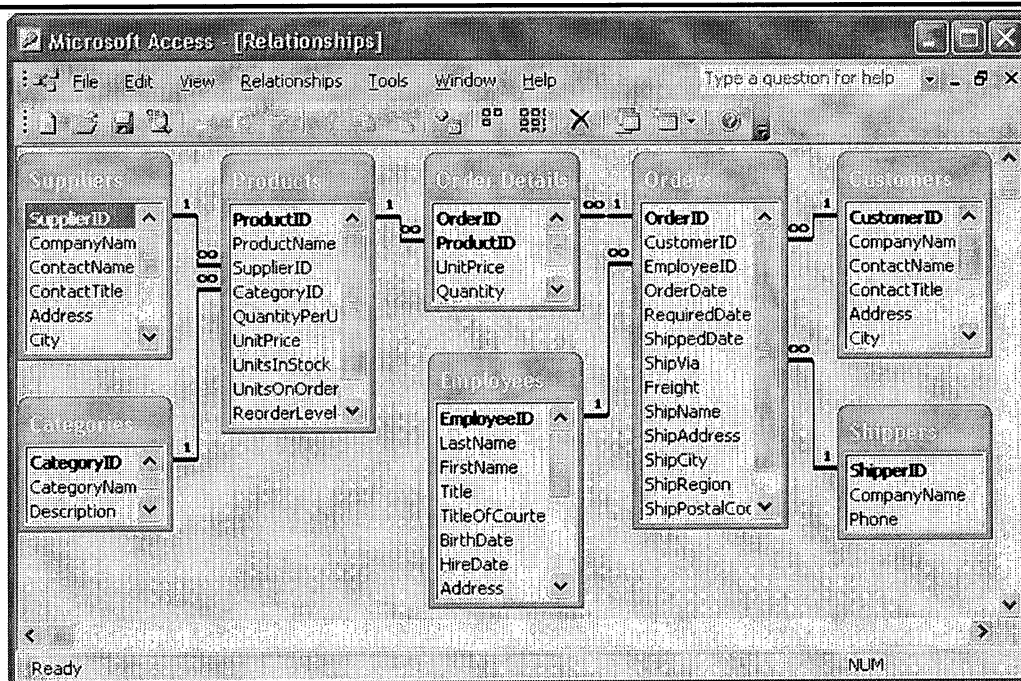
### **Makra (Macros)**

Sada předdefinovaných akcí pomocí nichž lze automatizovat práci s daty.

### **Moduly (Modules)**

Umožňují dělat programy pro práci s daty (v jazyce VBA), a vytvářet tak v rámci Accessu samostatné aplikace. Programy a makra jsou často spouštěny jako reakce na určitou událost (klepnutí na tlačítko, změna aktivního záznamu apod.).

# Seznamujeme se s výukovou databází Northwind



## Seznamujeme se s výukovou databází Northwind

Microsoft dodává s každou instalací Accessu výukovou databázi Northwind. Pomocí této databáze se lze Access snadno naučit.

**Každá databáze obvykle řeší jeden ucelený databázový problém.** V případě Northwindu je to vzájemný vztah mezi zákazníky a dodavateli. Zákazníci píší objednávky na výrobky, které jim dodavatelé dodávají. Objednávky vyřizují zaměstnanci zprostředkovatelské firmy a realizují přepravci.

K pochopení logiky databáze je nejlepší prostudovat **diagram relací**. Získáme jej klepnutím na tlačítko **Relationships** panelu **Database**, který je automaticky přítomen na obrazovce, je-li aktivní okno databáze.

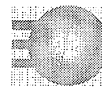
Obdélníky v diagramu znázorňují jednotlivé tabulky databáze. Řádky v nich jsou pole v tabulce. Pole tvořící primární klíč jsou formátovaná tučně.

Podívejme se nejdříve na tabulku **Customers**. **Primárním klíčem** je zde **CustomerID**. Každý zákazník může napsat více objednávek. V tabulce objednávek je proto u každé objednávky též kód zákazníka, který objednávku vyhotovil, uveden. V tomto poli však již mohou být duplicity. Jejich počet pro určitého zákazníka udává, kolik objednávek napsal. Pole **CustomerID** nazýváme **cizím (nevlastním) klíčem (Foreign Key)** v tabulce **Orders**.

**Cizí (nevlastní) klíč** tabulky obsahuje hodnoty primárního klíče té tabulky, s níž je původní tabulka v relaci.

Vztah mezi primárním klíčem tabulky **Customers** a cizím klíčem tabulky **Orders** vytváří mezi těmito tabulkami relaci. Je to relace 1:N, protože jeden zákazník může napsat více objednávek. Někdy tomuto typu relace říkáme jedna ku více.

V diagramu jsou relace znázorněny spojnicemi mezi primárním a cizím klíčem.



Na každé objednávce může být uvedeno více výrobků. V tabulce **Order Details** je každá objednávka pro každý výrobek na ní uvedený umístěna na samostatném řádku - záznamu. Proto jedné objednávce z tabulky **Orders** může odpovídat více řádků v tabulce **Order Details**. Mezi těmito tabulkami je opět relace 1:N.

V tabulce **Order Details** je jako primární klíč zvolena kombinace čísla objednávky a čísla výrobku. Každé z těchto polí samostatně však tvoří cizí klíč v tabulce.

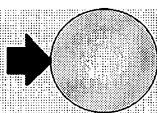
U každého dodávaného výrobku je uvedeno číslo dodavatele, které je cizím klíčem vzhledem k tabulce **Suppliers**. Mezi tabulkami **Suppliers** a **Products** je proto opět vztah 1:N. Stejný vztah najdeme i u tabulek **Products** a **Order Details**.

Výrobky jsou členěny do kategorií (nápoje, cukrovinky apod.). Do každé kategorie může patřit více výrobků, proto je mezi tabulkami **Categories** a **Products** opět relace 1:N.

Každý zaměstnanec zprostředkující firmy může vyřizovat více objednávek. Proto je mezi tabulkami **Employees** a **Orders** opět relace 1:N.

Podobně každý přepravce může realizovat přepravu zboží na více objednávkách, a proto je mezi tabulkami **Shippers** a **Orders** relace 1:N.

## Shrnutí



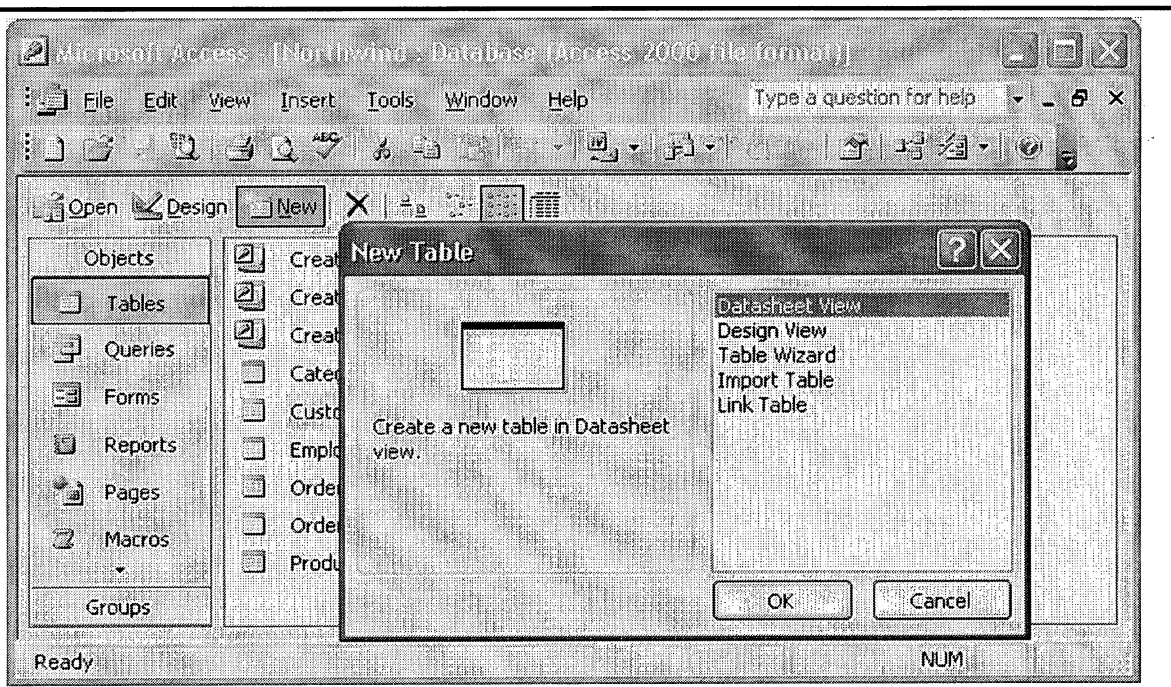
- Databáze je souhrn efektivně uložených dat, s nimiž lze snadno pracovat.
- Databáze Accessu je tvořena propojenými tabulkami a dalšími objekty (dotazy, formuláře, sestavy, datové stránky, makra a moduly).
- Při seznamování s vytvořenou databází si prohlédneme diagram relací (tlačítko **Relationship**).

# Navrhování tabulek

## *Přehled témat*

- tvorba tabulky
- pole tabulky a jejich atributy
- kopírování tabulek

# Tabulky – datová základna databáze

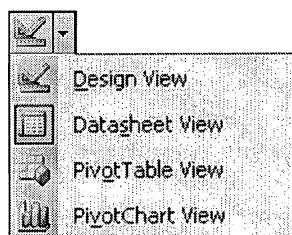


## Základní režimy zobrazování tabulky

V **návrhovém zobrazení (Design View)** vytváříme a modifikujeme pole tabulky, tj. definujeme strukturu tabulky.

V **zobrazení datového listu (Datasheet View)** pracujeme s daty (záznamy) uspořádanými v tabulce podobně jako třeba v programu MS Excel.

Po otevření tabulky přepínáme mezi oběma režimy pomocí příkazů v nabídce **View**, nebo tlačítkem **View** na panelu s nástroji.



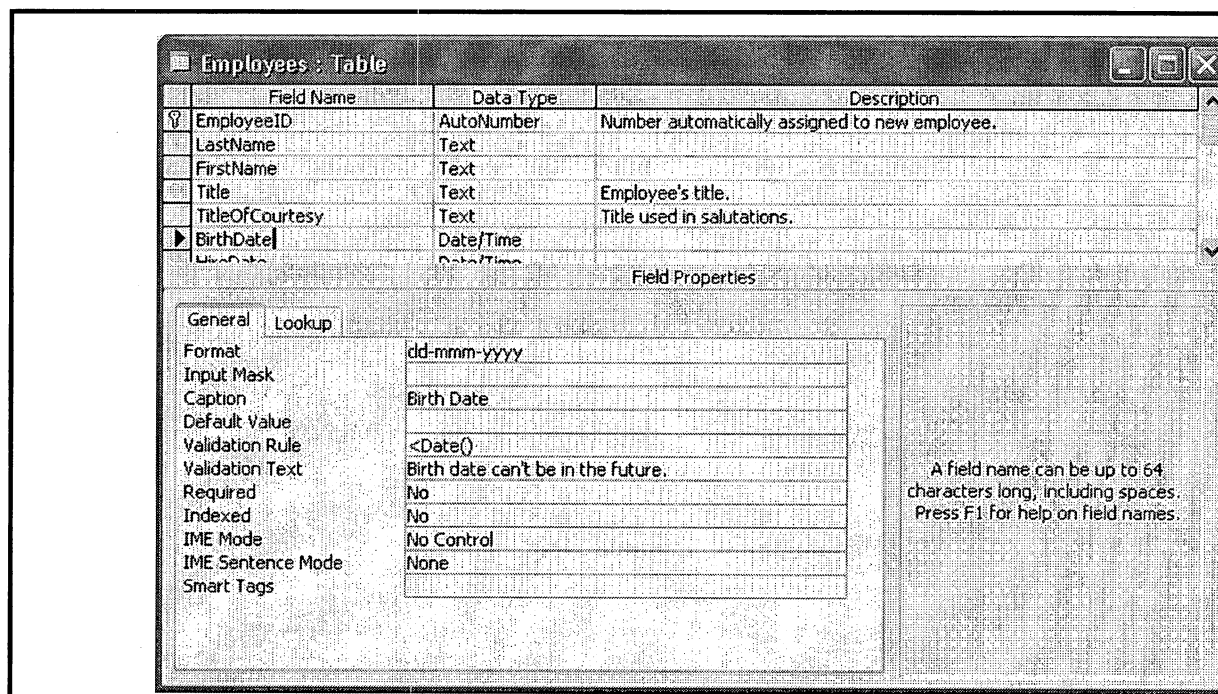
## Tvorba tabulky

V **okně databáze aktivujeme objekt Tables** a klepneme na tlačítko **New**. V okně **New Table** se objeví řada variant. Tvorba tabulky v návrhovém zobrazení skýtá nejvíce možností udělat tabulku podle vlastních představ a též proniknout do podstaty vzniku tabulky.

Tabulky lze též importovat z jiných datových zdrojů (např. sešit Excelu). V databázi lze též ukládat propojení na cizí tabulky (z jiné databáze, sešitu Excelu apod.).



## Návrhové zobrazení tabulky



| Field Name      | Data Type  | Description                                    |
|-----------------|------------|------------------------------------------------|
| EmployeeID      | AutoNumber | Number automatically assigned to new employee. |
| LastName        | Text       |                                                |
| FirstName       | Text       |                                                |
| Title           | Text       | Employee's title.                              |
| TitleOfCourtesy | Text       | Title used in salutations.                     |
| BirthDate       | Date/Time  |                                                |

| Field Properties  |                                    |
|-------------------|------------------------------------|
| General           | Lookup                             |
| Format            | dd-mmm-yyyy                        |
| Input Mask        |                                    |
| Caption           | Birth Date                         |
| Default Value     |                                    |
| Validation Rule   | <Date()                            |
| Validation Text   | Birth date can't be in the future. |
| Required          | No                                 |
| Indexed           | No                                 |
| IME Mode          | No Control                         |
| IME Sentence Mode | None                               |
| Smart Tags        |                                    |

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

### Návrhové zobrazení tabulky (Design View)

V horní části dialogu (mřížce) se pro každé pole povinně uvádí název pole (**Field Name**) a datový typ (**Data Type**). Popis (Description) je nepovinný, usnadní však ostatním uživatelům pochopení významu jednotlivých polí. Je-li tabulka v zobrazení datového listu a klepneme-li do pole, zobrazí se jeho popis ve stavovém řádku Accessu.

Na kartách v dolní části dialogu jsou potom další doplňující informace pro pole, které je právě v mřížce aktivní.

V programu MS Access mohou být názvy dlouhé až 64 znaky a mohou obsahovat libovolnou kombinaci písmen, číslic, mezer a zvláštních znaků s výjimkou tečky, vykřičníku, jednoduchých uvozovek, hranatých závorek a řídicích znaků.





## Datový typ (Data Type)

Je to nejdůležitější vlastnost pole tabulky. Je ho nutno specifikovat pro efektivní ukládání dat v tabulce a pro rychlou práci Accessu s daty.

Datové typy Accessu shrnuje následující tabulka. Více informací o nich lze získat z nápovědy Accessu.

|                 |                                                                       |                                                                                                                                                                |
|-----------------|-----------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Text            | Max. 255 znaků                                                        | Běžné textové informace, jako je příjmení nebo čísla, která se nepoužívají při výpočtech (telefonní číslo, rodné číslo, inventární číslo, popisné číslo).      |
| Memo (poznámka) | Do 65 535 znaků                                                       | Delší texty - poznámky a popisy.                                                                                                                               |
| Number          | 1, 2, 4 nebo 8 bytů. 16 bytů pro replikaci automatického čísla (GUID) | Číselná data, která mají být použita pro výpočty, s výjimkou peněžních výpočtů (použijte datový typ Měna).                                                     |
| Date/Time       | 8 bytů                                                                | Datum a čas.                                                                                                                                                   |
| Currency        | 8 bytů                                                                | Hodnoty měny. Zamezí vzniku zaokrouhlovacích chyb během výpočtů.                                                                                               |
| AutoNumber      | 4 byty                                                                | Jedinečná sekvenční (vzestupná po 1) či náhodná čísla. Vkládají se automaticky s přidáním záznamu.                                                             |
| Yes/No          | 1 bit                                                                 | Může obsahovat pouze jednu ze dvou hodnot - např. Ano/Ne, Pravda/Nepravda, Zapnuto/Vypnuto.                                                                    |
| OLE Object      | Do 1 gigabytu (omezeno prostorem na disku)                            | Objekty (dokumenty Wordu, tabulky Excelu, obrázky, zvuky nebo jiná binární data).                                                                              |
| Hyperlink       | Do 2 048 znaků. Blíže viz nápověda Accessu.                           | Klasické hypertextové odkazy, známé např. z práce na Internetu.                                                                                                |
| Lookup Wizard   | obvykle 4 byty                                                        | Vytváří pole, které umožní vybrat hodnotu z jiné tabulky nebo ze seznamu hodnot pomocí ovládacího prvku rozbalovací seznam. Hodnoty umožní definovat průvodce. |

## Další důležité vlastnosti polí

**Field Size:** U datového typu **Text** určuje maximální počet znaků v poli. U datového typu **Number** určuje velikost a typ čísel v poli (celé číslo, dlouhé celé číslo, desetinné číslo, dvojitá přesnost apod.). Velikost záznamu je **omezena** na 2000 znaků, kromě polí Memo a OLE Object. **Proto se vyplatí stanovit velikost každého pole na míru našim potřebám,** a ne více.

**Format** určuje způsob zobrazení dat v poli. Lze vybírat z předdefinovaných formátů, nebo dělat formáty vlastní.

**Decimal Places:** při nastavení na **Auto** určuje počet desetinných míst formát.

**Input Mask** je vzorovým tvarem (šablonou) pro zadávaná data. Při její tvorbě lze využít průvodce vstupní maskou. Definujeme-li pro určité pole vstupní masku a zároveň nastavíme vlastnost **Format**, má při zobrazení dat přednost hodnota vlastnosti **Format**.

**Caption** je uváděn v záhlaví polí místo názvů těchto polí. Při tvorbě formulářů a sestav je tento titulek automaticky použit v popisku ovládacího prvku (většinou textového pole).

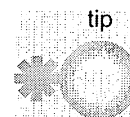
**Default Value** je automaticky nastavena v poli nového záznamu, uživatel ji nemusí zadávat.

**Validation Rule** je podmínka, kterou musí hodnoty pole při jejich zadávání splnit. Není-li tomu tak, nelze hodnotu do pole zadat a Access zobrazí zprávu s **Validation Text**.

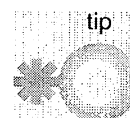
**Required:** možné hodnoty **yes/no**. Je-li nastaveno na **no**, možno nechat pole prázdné. V tom případě obsahuje hodnotu **Null**. V opačném případě (hodnota **yes**) nutno do pole hodnotu zadat.

**Allow Zero Length:** možné hodnoty **yes/no**. Nastavíme-li na **yes**, možno do pole zadat prázdný řetězec (realizujeme zadáním dvou uvozovek bezprostředně za sebou).

*Zadáme-li jako vstupní masku slovo **Password** bez zadání formátu dat, budou se údaje v poli zobrazovat jako hvězdička.*

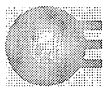


*Nevíme-li, jak hodnotu **Required** nastavit, doporučujeme nastavení **no**. Neznáme-li při nastavení **yes** hodnotu pole, nelze jej nechat prázdné a Access **odmítne vytvářený záznam uložit**.*



*Máme např. textové pole pro zadání telefonního čísla. Nastavíme **Allow Zero Length** na **yes**, **Required** na **no**. Necháme-li toto pole prázdné (Null), pak osoba má telefon, ale my jej neznáme. Zadáme-li do pole prázdný řetězec, pak osoba telefon nemá.*





**Unicode Compression:** Počínaje verzí 2000 kóduje Access znaky pomocí dvoubytového kódu Unicode, který je schopen zakódovat jakoukoli světovou abecedu i nejrůznější matematické symboly. Pro znaky nám běžných abeced (čeština, angličtina apod.) stačí však ke kódování jeden byte, proto nastavujeme hodnotu této vlastnosti zpravidla na **yes** a ke kompresi dochází.

**IME Mode:** je to režim textového editoru sloužícího při vkládání dat. Převádí kódy stisknutých kláves na asijské znaky.

**IME Sentence Mode:** tento režim funguje jako režim IME, ale pouze pro japonštinu.

**Smart Tags:** pomocí průvodce si můžete vybrat, jaké inteligentní značky se mají u pole v tabulce vyskytovat. Inteligentní značky ze zobrazeného dialogového okna se budou chovat tak, jak jste zvyklí z Outlooku.



**Smart Tags** [X]

Available Smart Tags

- ☒ Date
- ☐ Financial Symbol
- ☐ Person Name

Smart Tag Details

Name: Date

Actions: Schedule a Meeting  
Show my Calendar

More Smart Tags OK Cancel

**Smart Tags** [X]

Available Smart Tags

- ☐ Date
- ☐ Financial Symbol
- ☒ Person Name

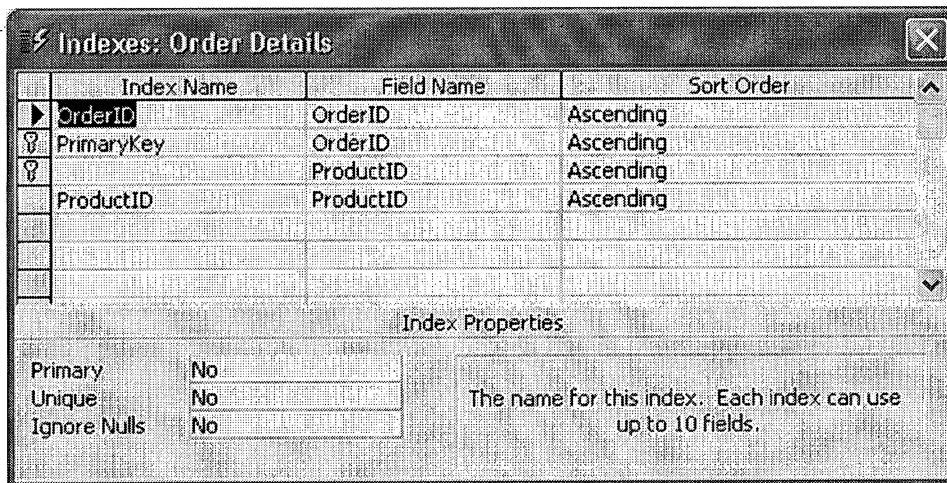
Smart Tag Details

Name: Person Name

Actions: Send Mail  
Schedule a Meeting  
Open Contact  
Add to Contacts

More Smart Tags OK Cancel

# Indexy v tabulce, primární klíč tabulky



## Indexy v tabulce, primární klíč tabulky

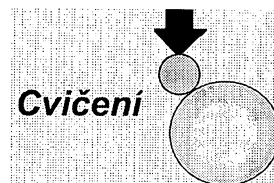
**Vlastnost Indexed** má hodnoty *No*, *Yes (Duplicates OK)*, *Yes (No Duplicates)*. Budeme-li podle tohoto pole často řadit a prohledávat, nastavíme vlastnost na *Yes*. Nastavením *Yes (No Duplicates)* zajistíme, že Access **nepovolí** v tomto poli duplicitní hodnoty, čehož můžeme s výhodou využít. Primární klíč musí mít vždy jedinečný index - *Yes (No Duplicates)*. Seznam indexů v tabulce získáme příkazem **View/Indexes**, či tlačítkem **Indexes** na panelu nástrojů. V dialogovém okně tabulky indexů možno i indexy tvořit. Toto je pro uživatele jediná cesta, jak vytvořit vícenásobný index (podle více polí).

Tvorbu indexu v tabulce lze přirovnat ke tvorbě rejstříku v knize. Chceme-li najít všechny výskyty nějakého hesla v knize bez rejstříku, nutno ji celou přečíst. Hledáme-li záznamy podle určitého pole v tabulce bez indexu, nutno ji celou prohledat. Chceme-li najít výskyt nějakého hesla v knize s rejstříkem, zjistíme z něj číslo stránky a heslo rychle najdeme. Hledáme-li záznam podle určitého pole v tabulce s indexem pro toto pole nadefinovaným, najde Access efektivně v odpovídající indexové tabulce pořadová čísla záznamů obsahujících toto heslo. Zmíněné indexové tabulky však musí Access tvořit a udržovat, což stojí určitou režii. Proto nezavádíme indexy zbytečně.

**Primary Key** jednoznačně identifikuje každý záznam tabulky. Je tvořen jedním nebo více poli (sloupci) tabulky. Nedovoluje hodnoty Null a musí mít vždy jedinečný index (bez duplicit). Ke vzniku tohoto indexu dojde automaticky, tj. samotnou definicí primárního klíče. **Jak vytvoříme primární klíč:** Vyznačíme v návrhovém zobrazení tabulky pole, která jej tvoří a klepneme na tlačítko **Primary Key** ovládacího panelu (též příkaz **Edit/Primary Key**). Chceme-li, aby nám Access generoval pole primárního klíče **automaticky**, nastavíme datový typ tohoto pole na **AutoNumber**.

Zadáváme-li primární klíč sami, nedovolí Access zadat do tabulky duplicitní hodnotu (nedovolí tento záznam uložit - tento fakt můžete zjistit až po zadání všech polí záznamu).

Není třeba, aby každá tabulka měla primární klíč. Je-li však třeba z tabulky vytvořit relaci 1:N na cizí klíč jiné tabulky, je jeho přítomnost nutná.



## Úkoly k procvičování:

- 1 Vytvořme novou databázi. Nazveme ji **ZamDěti**. V databázi navrhne tabulku **Lidi**, do které budeme ukládat záznamy o zaměstnancích naší firmy. Tabulka má pole pro ukládání těchto atributů zaměstnanců: primární klíč, jméno, příjmení, datum narození, plat a dárce krve. Pro každé pole nadefinujeme vhodné vlastnosti. Do tabulky potom zadáme několik záznamů.

Postup:

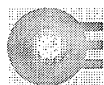
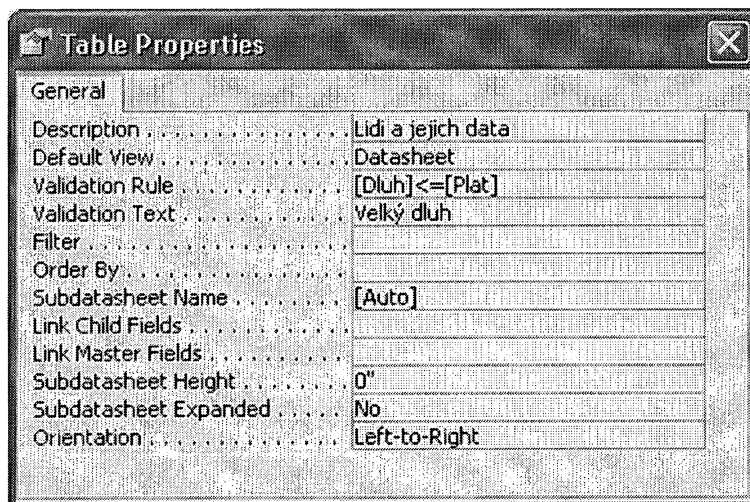
1. Po vydání příkazu **File/New** se objeví podokno úloh, v němž klepneme na položku **Blank database**.
2. Objeví se dialogové okno pro uložení našeho databázového souboru, v němž zadáme jeho název **ZamDěti** a klepneme na tlačítko **Create**.
3. Máme-li předem otevřenou nějakou databázi, je uzavřena, otevírá se naše nová databáze a objeví se její okno.
4. V okně databáze **ZamDěti** aktivujeme objekt **Tables**, klepneme na tlačítko **New** a specifikujeme tvorbu tabulky v návrhovém zobrazení.

V tabulce definujeme pole podle následujícího obrázku

|   | Field Name | Data Type  |
|---|------------|------------|
| 🔑 | PorČíslo   | AutoNumber |
|   | Příjmení   | Text       |
|   | Jméno      | Text       |
|   | DatNar     | Date/Time  |
|   | Plat       | Currency   |
|   | DárceKrv   | Yes/No     |

5. U pole **DatNar** pro datum narození specifikujeme na kartě **General** vstupní masku. Na konci řádku **Input Mask** klepneme na tlačítko s třemi tečkami (výpustek), uložíme tabulku a dále se řídíme pokyny průvodce vstupní maskou. Název formátu masky zadáme **Short Date**.
6. U polí **DatNar** a **DárceKrv** vyplníme titulek **Datum narození** a **Dárce krve**.
7. Klepneme na tlačítko **View** panelu nástrojů pro přechod do **Datasheet View** a na dotaz k uložení tabulky odpovíme kladně.
8. Do tabulky zadáme několik záznamů.

# Ověřovací pravidla



## Ověřovací pravidla (Validation Rules)

### V Accessu existuje

- ověřovací pravidlo pro pole
- ověřovací pravidlo pro záznam.

Jak jsme již uvedli, **ověřovací pravidlo pro pole** je podmínka, kterou musí hodnoty pole při zadávání splnit. Není-li tomu tak, nelze hodnotu do pole zadat a Access zobrazí zprávu s **Validation Text**. Toto pravidlo je kontrolováno při pokusu o ukončení editace pole.

### Příklady ověřovacího pravidla pro pole:

**>=0** : Hodnota pole nesmí být záporná.

**between 10000 and 50000** : Hodnota pole musí ležet v intervalu 10000 až 50000 včetně mezí.

**Like "A\*"** : Hodnota pole musí začínat na písmeno A či a.

**Ověřovací pravidlo pro záznam** je podmínka umožňující testovat navzájem hodnoty více polí v záznamu. Kontroluje se až při pokusu o uložení záznamu. Není-li splněna, nelze záznam uložit a zobrazí se **Validation Text**. Ověřovací pravidlo pro záznam i ověřovací text zadáváme do dialogového okna vlastností tabulky. Získáme jej příkazem **View/Properties** v návrhovém režimu tabulky. Příkaz je obsažen i v místní nabídce tohoto režimu.

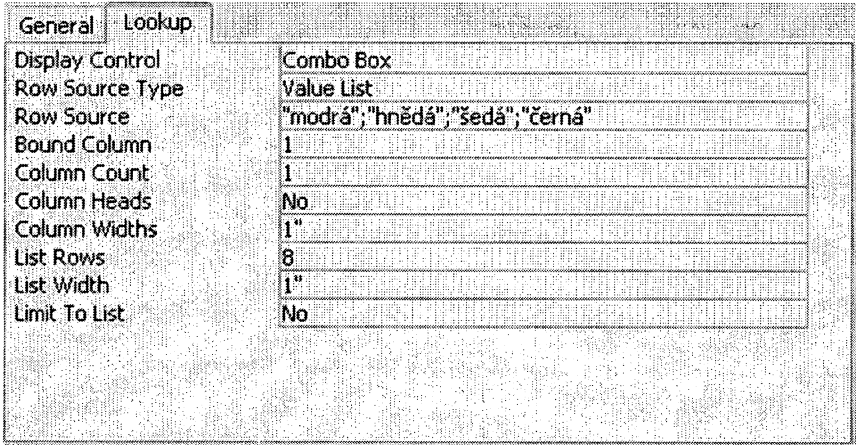
tip



*Možné hodnoty zadávané do pole primárně omezíme volbou vhodného datového typu pole. Definice vstupní masky pro pole je dalším omezovacím nástrojem, který možno použít. K těmto nástrojům dále patří ověřovací pravidla pro pole a pro záznam.*



# Průvodce vyhledáváním



| Property        | Value                             |
|-----------------|-----------------------------------|
| Display Control | Combo Box                         |
| Row Source Type | Value List                        |
| Row Source      | "modrá"; "hnědá"; "šedá"; "černá" |
| Bound Column    | 1                                 |
| Column Count    | 1                                 |
| Column Heads    | No                                |
| Column Widths   | 1"                                |
| List Rows       | 8                                 |
| List Width      | 1"                                |
| Limit To List   | No                                |

## Průvodce vyhledáváním (Lookup Wizard)



V seznamu datových typů Accessu je uveden **Lookup Wizard**. Tuto položku volíme

- chceme-li zadávat hodnoty do pole z určitého seznamu hodnot (např. barvy očí apod.)
- chceme-li zadávat hodnoty do pole z pole jiné tabulky.

## Průvodce vyhledáváním pro zadávání hodnot do pole ze seznamu zadaného uživatelem

Na obrázku je karta **Lookup** návrhového režimu tabulky, která byla vyplněna průvodcem vyhledávání pro pole BarvaOčí. Do políčka **Row Source** lze dodatečně další položky dopisovat. Nastavení vlastnosti **Limit To List** na hodnotu **No** znamená, že do pole BarvaOčí lze zadat i barvy, které náš seznam neobsahuje.

# Průvodce vyhledáváním pro zadávání hodnot z pole jiné tabulky

| Property        | Value                                          |
|-----------------|------------------------------------------------|
| Display Control | Combo Box                                      |
| Row Source Type | Table/Query                                    |
| Row Source      | SELECT Lidi.PořČíslo, Lidi.Příjmení FROM Lidi; |
| Bound Column    | 1                                              |
| Column Count    | 2                                              |
| Column Heads    | No                                             |
| Column Widths   | 0";1"                                          |
| List Rows       | 8                                              |
| List Width      | 1"                                             |
| Limit To List   | Yes                                            |



## Průvodce vyhledáváním pro zadávání hodnot z pole jiné tabulky

Na obrázku je karta **Lookup** návrhového režimu tabulky, která byla vyplněna průvodcem vyhledávání pro pole **Rodič** tabulky **Děti**. Do tohoto pole se ukládají hodnoty primárního klíče z tabulky **Lidi** (u dítěte je to primární klíč jeho rodiče). V poli se však zobrazují příjmení rodičů, a ne jejich primární klíče.

Průvodce vyhledáváním vytvořil hodnotu vlastnosti **Zdroj řádků** jako SQL příkaz `SELECT [Lidi].[PořČíslo], [Lidi].[Příjmení] FROM [Lidi];`. To znamená, že z tabulky **Lidi** se vybírají pole **PořČíslo** a **Příjmení**. Vázaný sloupec je jedna, tj. do pole **Rodič** se ukládá **PořČíslo** (což je primární klíč) z tabulky **Lidi**. Šířka prvního sloupce je však nula, tedy v poli **Rodič** se zobrazuje příjmení, a ne primární klíč rodiče.

Editací příkazu `Select` na

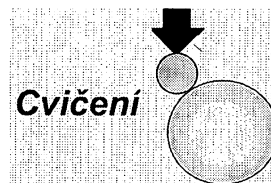
**SELECT [Lidi].[PořČíslo], [Lidi].[Příjmení] & " (" & [Lidi].[PořČíslo] & ")"**  
**FROM [Lidi];** způsobíme, že se v poli **Rodič** tabulky **Děti** bude kromě příjmení rodiče ještě v závorce zobrazovat jeho primární klíč.

| Rodič        |
|--------------|
| Kousal (1)   |
| Bouchal (3)  |
| Bouchal (3)  |
| Kousal (1)   |
| Fousková (4) |
| 0            |

tip



Editaci lze dělat v dialogovém okně **Zoom**, které s výhodou zobrazíme klávesovou zkratkou **SHIFT+F2**.



## Úkoly k procvičování:

- 1 V tabulce **Lidi** doplníme do pole **Plat** ověřovací pravidlo pro pole ve tvaru  $\geq 10000$ . Potom vytvoříme nové pole **Dluh** (typu měna) a ověřovací pravidlo pro záznam  $\text{Dluh} \leq \text{Plat}$ . Pozorujeme chování Accessu po dodání těchto pravidel do tabulky, která již obsahuje zadané záznamy. Postup:

1. Přejdeme do režimu návrhu tabulky **Lidi** a aktivujeme pole **Plat**
2. Doplníme pro něj ověřovací pravidlo a ověřovací text podle následujícího obrázku

|                 |                          |
|-----------------|--------------------------|
| Validation Rule | $\geq 10000$             |
| Validation Text | Nedosažena hodnota 10000 |

3. Po vytvoření pole **Dluh** zobrazíme dialog vlastností tabulky (**View/Properties**) a doplníme ověřovací pravidlo pro záznam a ověřovací text podle následujícího obrázku.

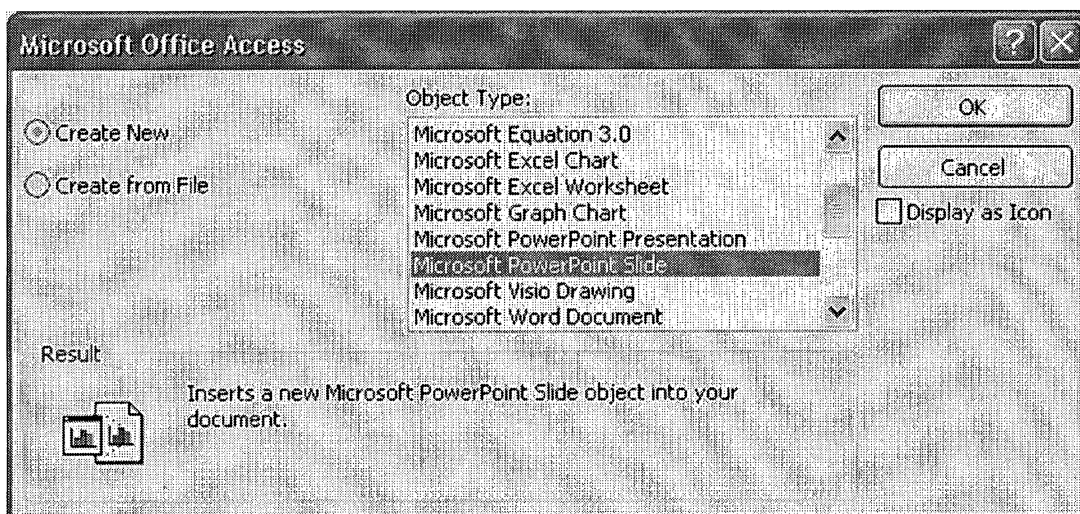
|                 |                                    |
|-----------------|------------------------------------|
| Validation Rule | $[\text{Dluh}] \leq [\text{Plat}]$ |
| Validation Text | Moc velký dluh                     |

- 2 V tabulce **Lidi** vytvoříme pole **BarvaOčí** a nadefinujeme pro něj pomocí průvodce vyhledáváním zadávání barev očí z předem určeného seznamu (viz obrázek k tématu **Průvodce vyhledáváním pro zadávání hodnot do pole ze seznamu zadaného uživatelem** a výklad).
- 3 V databázi **ZamDěti** zhotovíme novou tabulku **Děti** s poli pro tato data: primární klíč, jméno, příjmení, rodič. Pole **Rodič** vytvoříme podle právě probraných skutečností. Access sám udělá mezi tabulkami **Lidi** a **Děti** relaci. Prověřme to.  
Postup:  
Návrhový režim tabulky **Děti** bude vypadat takto:

|  | Field Name | Data Type  |
|--|------------|------------|
|  | PrimKlic   | AutoNumber |
|  | Příjmení   | Text       |
|  | Jméno      | Text       |
|  | Rodič      | Number     |

U pole **Rodič** jsme však nejprve volili datový typ **Lookup Wizard**. Po běhu průvodce (viz předchozí výklad), je nastaven datový typ **Number**.

## Zadávání hodnot do polí typů OLE Object



### Zadávání hodnot do polí typů OLE Object

Do pole datového typu **OLE Object** vkládáme data v režimu datového listu tabulky takto: z kontextové nabídky tohoto pole v určitém záznamu vykonáme příkaz **Insert/Object** a vyplníme dialogové okno.

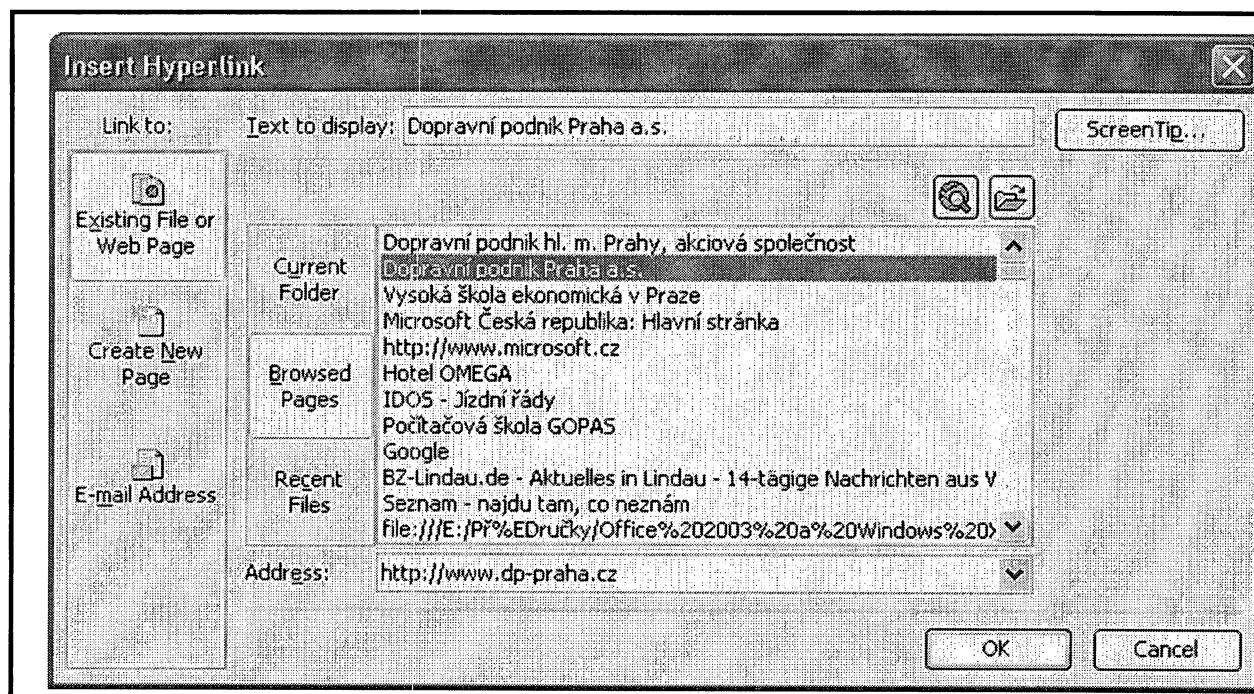
Chceme-li vytvořit nový objekt, vybereme ze seznamu **Object Type** program, kterým objekt zhotovíme.

Máme-li již objekt připraven v souboru, volíme možnost **Create from File**, klepneme na tlačítko **Browse** a specifikujeme tento soubor cestou k němu.

Do pole v tabulce můžete také vložit data, například obrázek, ze schránky Windows.

Vložený objekt se nezobrazuje v polích tabulky, ale lze jej zobrazit poklepáním na pole. Objekty se budou zobrazovat ve vázaných formulářích, vyrobených nad takovou tabulkou.

## Zadávání hodnot do polí typů Hyperlink



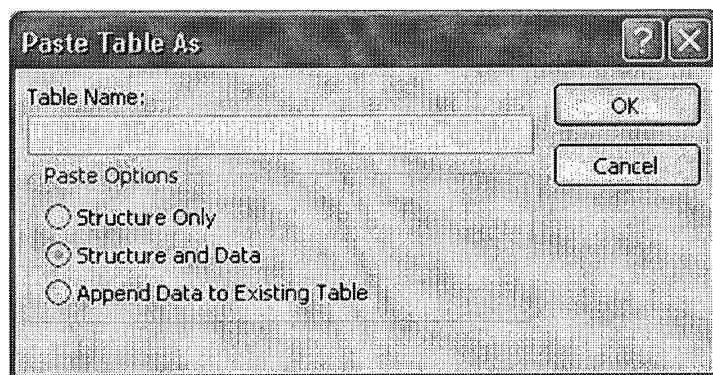
## Zadávání hodnot do polí typů Hyperlink

Do pole datového typu Hyperlink vkládáme data v režimu Datasheet View tabulky takto: z kontextové nabídky tohoto pole v určitém záznamu vykonáme příkaz **Hyperlink/Edit Hyperlink** a vyplníme dialogové okno.

Jako hypertextový odkaz může sloužit adresa WWW stránky, cesta k souboru či specifikace objektu naší databáze nebo adresa elektronické pošty. Text, který vepíšeme do políčka **Text to display**, bude zobrazován v poli editovaného záznamu.



# Kopírování tabulek v databázi



## Kopírování tabulek v databázi

V okně databáze aktivujeme objekt **Tables**.

Vyznačíme konkrétní tabulku a zkopírujeme ji do schránky (Ctrl+C).

Tabulku vložíme ze schránky zpět do okna objektu **Tables** (CTRL+V).

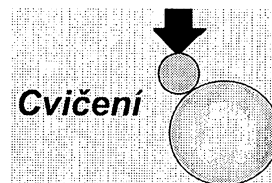
Objeví se dialogové okno **Paste Table As**, v němž zadáme název nové tabulky a vyznačíme jednu z možností vložení.

- **Structure Only:** cílová tabulka nebude obsahovat data zdrojové tabulky, ale bude mít její pole.
- **Structure and Data:** cílová tabulka bude obsahovat i data zdrojové tabulky.
- **Append Data to Existing Table:** data ze zdrojové tabulky připojíme k datům tabulky již existující. Cílová tabulka však musí obsahovat stejná pole jako tabulka zdrojová. Dojde-li při pokusu o připojení k narušení integrity dat, operace neproběhne.

poznámka



*Kopie tabulek děláme zejména z důvodu zálohování dat při testování akčních dotazů v databázi (akční dotazy mění data v tabulkách a po jejich běhu nelze provést vrácení zpět).*

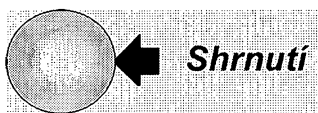


## Úkoly k procvičování:

- 1 Do tabulky **Děti** databáze **ZamDěti** dejme pole typu **Hyperlink** a vyplňme v něm data. Prozkoušejme přitom různé typy odkazů. Postup:
  1. Přejdeme do **Design View** tabulky **Děti**.
  2. Do tabulky přidáme nové pole nazvané **Odkaz** typu **Hyperlink**.
  3. Uložíme strukturu tabulky **Děti** a přejdeme do režimu **Datasheet View**.
  4. Z místní nabídky pole **Odkaz** zvoleného záznamu vykonáme příkaz **Hyperlink/Edit Hyperlink**.
  5. Zadáme určitý typ hypertextového odkazu.

V databázi **ZamDěti** vytvoříme záložní tabulky pro obě tabulky (**Lidi** a **Děti**)

- 2
  1. Aktivujeme okno databáze, objekt **Tables**.
  2. Zkopírujeme do schránky tabulku **Lidi** (CTRL+C).
  3. Obsah schránky vložíme zpět (CTRL+V).
  4. V dialogovém okně, které se objeví, vyplníme název tabulky **LidiZáloha** a aktivujeme možnost vložení **Structure and Data**.
  5. V případě tabulky **Děti** postupujeme obdobně.



- V návrhovém zobrazení (Design View) vytváříme a modifikujeme pole tabulky. V zobrazení datového listu (Datasheet View) pracujeme s daty v tabulce.
- U každého pole tabulky musíme zadat jeho datový typ (Data Type).
- Indexy v tabulce urychlují prohledávání.
- Primární klíč (Primary Key) jednoznačně identifikuje každý záznam tabulky.
- Ověřovací pravidla (Validation Rules) testují vstupní hodnoty polí.
- Chceme-li vybírat hodnoty do pole z určitého seznamu nebo z pole jiné tabulky, zvolme v datovém typu pole Lookup Wizard.

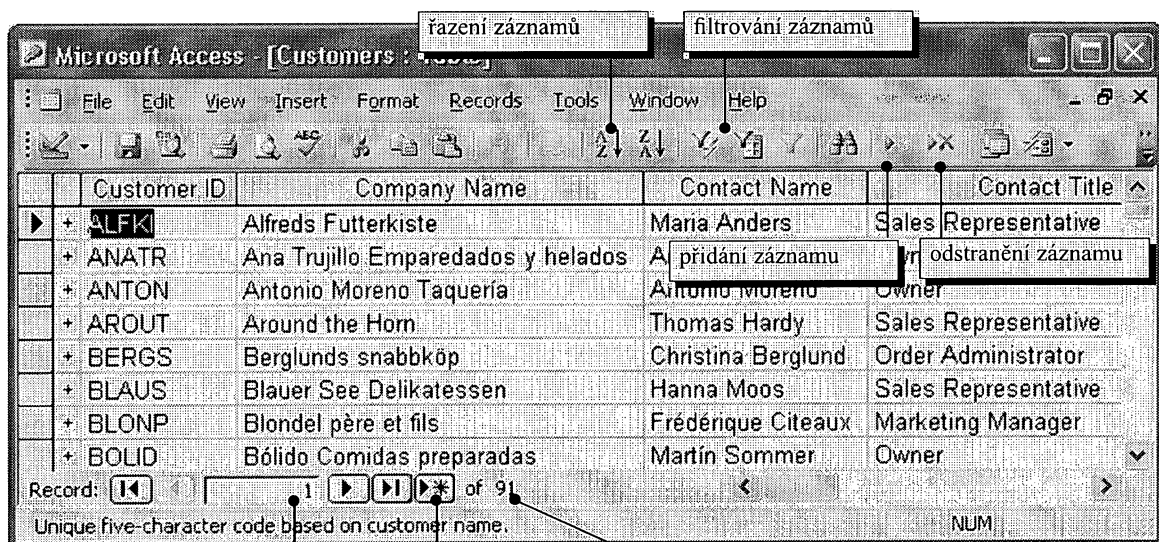


# Datový list

## *Přehled témat*

- práce se záznamy
- skrývání a ukotvování polí
- práce s vnořeným datovým listem
- filtrování záznamů v tabulce
- tabulky - shrnutí a zobecnění základních pojmů

# Práce se záznamy



## Práce se záznamy v režimu datového listu tabulky Accessu

Práce v režimu datového listu se v mnohém se podobá práci na excelovském listu. Zkušený uživatel počítače se naučí mnoho základních obrátů sám. Proto se zmíníme jen o důležitých a složitějších rysech této činnosti.

### Editace záznamu

Záznam aktivujeme klepnutím do pole, jehož obsah chceme změnit, a editaci provedeme. Je-li editované pole aktivováno, lze provedené změny zrušit klávesou **Esc**. Chceme-li aktivovat jiný záznam, uloží se nejprve automaticky obsah editovaného záznamu. Ještě po aktivaci jiného záznamu je možno příkazem **Undo Saved Record** (tlačítkem **Undo**) změny provedené v původním záznamu zrušit.

### Přidání nového záznamu



Klepeme na tlačítko **New Record** na panelu s nástroji či na navigačním ovládacím prvku pod tabulkou a do právě vzniklého nového záznamu zadáme data. Záznam lze zrušit podobně jak bylo popsáno v předchozím odstavci

### Odstranění záznamu



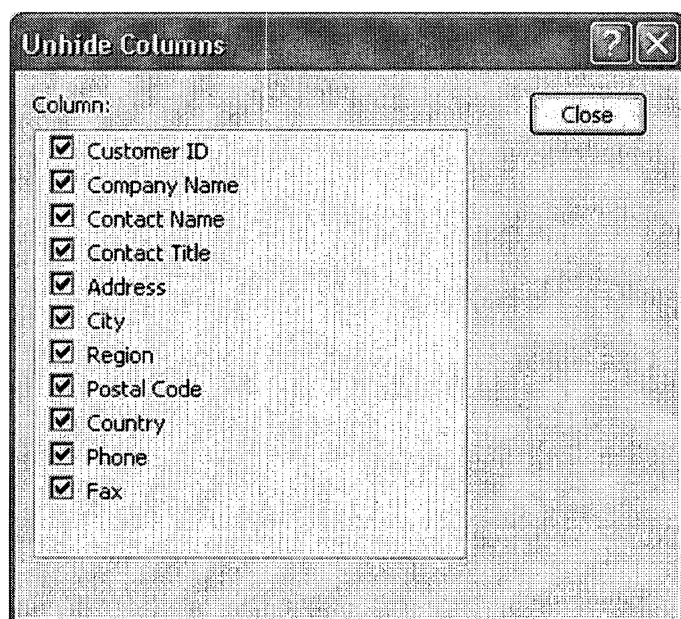
Záznam aktivujeme a klepeme na tlačítko **Delete Record**. **Odstraněný záznam nelze vrátit zpět.** Je-li tabulka v relaci 1:N s jinou, je možné, že z ní nelze za určitých podmínek (Referenční integrita – viz dále) odstraňovat záznamy.

### Řazení záznamů podle jednoho pole



Umístíme kurzor do tohoto pole a klepeme na některé z tlačítek pro řazení.

## Skrývání a ukotvování polí



### Skrývání polí

Pole, která chceme skrýt, vyznačíme klepnutím myši (pro jedno pole) či tahem myši (pro více polí) v záhlaví tabulky. Potom vykonáme příkaz **Format/Hide Columns**.

Zobrazovat či skrývat sloupce lze velice pohodlně v dialogovém okně příkazu **Format/Unhide Columns**.

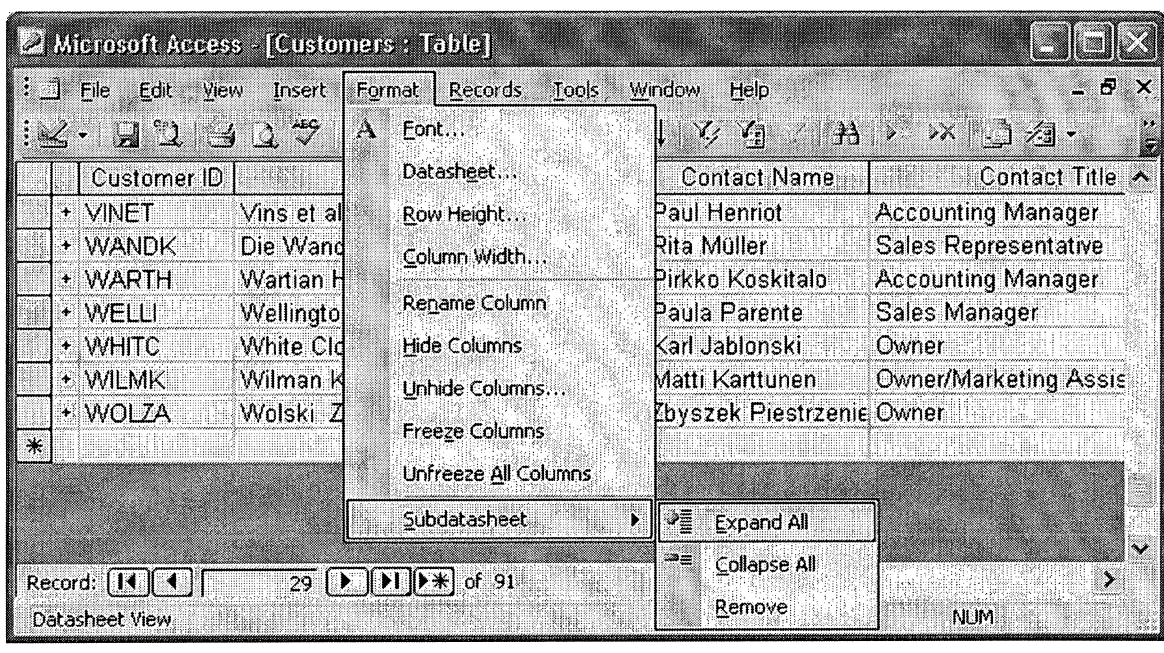
### Ukotvování polí

Sloupce, které se při používání vodorovného posuvníku nemají pohybovat, označíme tahem myši přes záhlaví sloupců a ukotvíme příkazem **Format/Freeze Columns**. Opět uvolníme příkazem **Format/Unfreeze All Columns**.

Ukotvují se pouze sloupce umístěné v tabulce na levém kraji. Označíte-li pro potřeby ukotvení sloupce, které nejsou umístěny vlevo, po ukotvení budou k levému kraji tabulky přesunuty automaticky.



## Práce s vnořeným datovým listem



### Práce s vnořeným datovým listem (Subdatasheet)

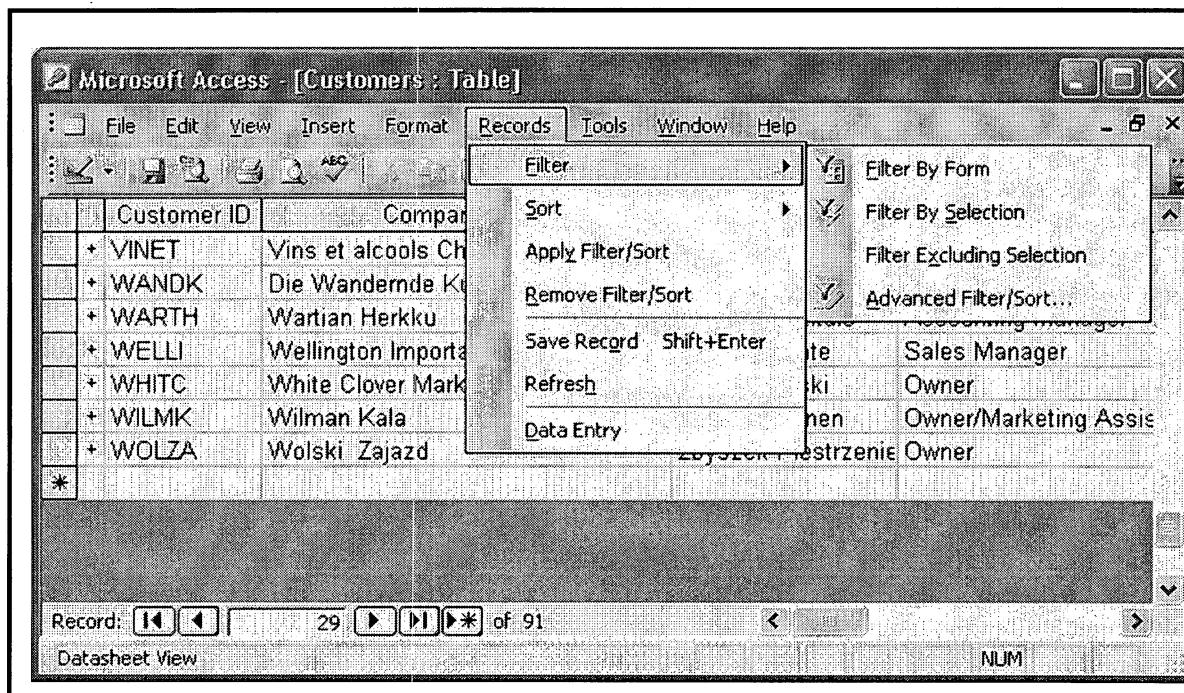
Je-li mezi dvěma tabulkami vytvořena relace 1:N, objeví se na začátku každého záznamu datového listu tabulky na straně jedna znaménko plus. Klepnutím na něj, zobrazíme (rozbalíme) záznamy z tabulky na straně N relace příslušející tomuto záznamu.

Všechny záznamy z tabulky na straně N lze rozbalit/sbalit příkazy nabídky **Format/Subdatasheet**. V nabídce je též příkaz k odebrání vnořeného datového listu.

Vnořený datový list lze přidat do tabulky příkazem **Insert/Subdatasheet**. Jeho dialogové okno je na následujícím obrázku.



## Filtrování záznamů v tabulce



### Filtrování záznamů v tabulce



Filtrováním záznamů tabulky zobrazujeme pouze ty záznamy, které vyhovují určité stanovené podmínce – filtru (projdou tímto filtrem). V nabídce příkazu **Records/Filter** vidíme čtyři typy filtrů. Tlačítka **Filter By Selection**, **Filter By Form**, **Apply/Remove Filter** jsou i na příslušném panelu.

### Filtrování podle výběru (resp. mimo výběr)

Klepeme do hodnoty pole, která se rovná podmínce filtru (existuje-li taková). Této hodnotě se opravdu budou (resp. nebudou) rovnat hodnoty příslušného pole záznamů prošlých filtrem. Potom aplikujeme příkaz **Filter By Selection** (resp. **Filter Excluding Selection**).



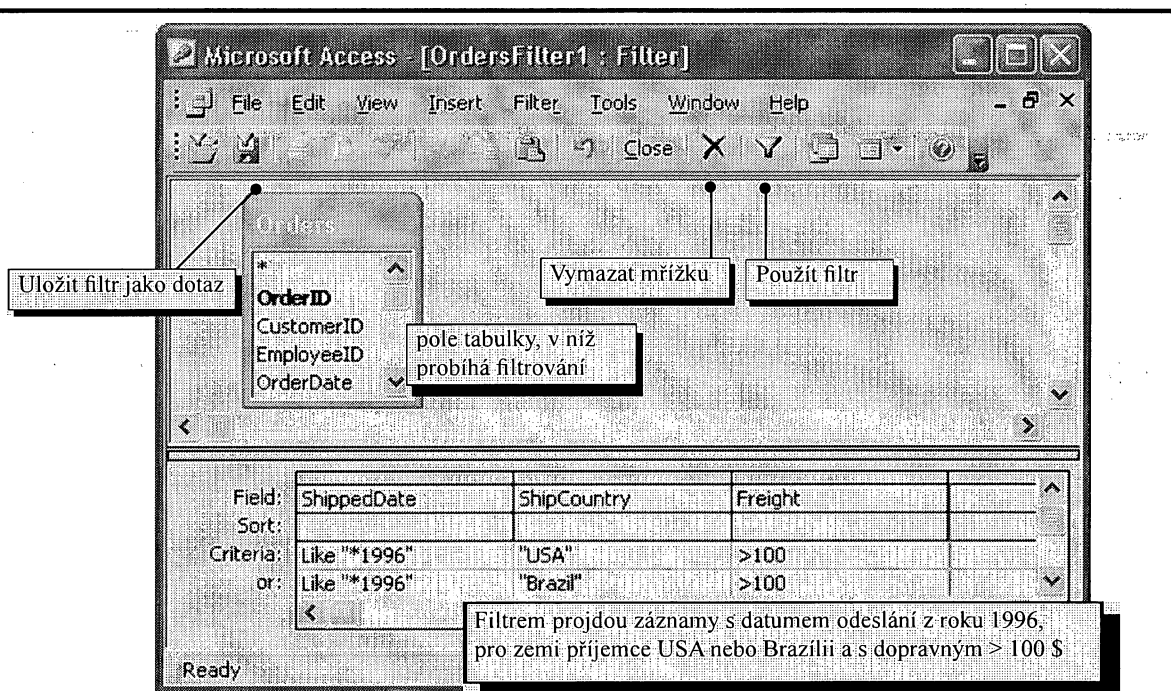
Vyznačíme-li pouze část hodnoty pole, projdou (resp. neprojdou) filtrem podle výběru (resp. filtrem mimo výběr) záznamy obsahující (resp. neobsahující) v tomto poli vyznačenou část. Aktivní filtr lze tlačítkem **Apply/Remove Filter** odstranit. Posledně použitý filtr se v tabulce ukládá a lze jej opět aplikovat tlačítkem **Apply/Remove Filter**.

Na vyfiltrované záznamy lze opět aplikovat filtr podle výběru. Podmínky takto po sobě použitých filtrů jsou spojeny operátorem and, tj. takto vzniklým filtrem projdou pouze záznamy splňující všechny nastavené podmínky zároveň.

### Filtrování podle formuláře

Podmínky zadáváme na zvláštních listech. Podmínky na jednotlivých listech jsou spojovány operátorem OR (NEBO), v rámci listu jsou podmínky na hodnoty polí spojovány operátorem AND (A). Aby záznam prošel filtrem, stačí, když splní všechny podmínky kteréhokoliv listu.

## Rozšířený filtr či řazení



### Rozšířený filtr či řazení

Příkazem **Records/Filtre/Advanced Filter/Sort** zobrazíme okno tohoto filtru. V horní části okna je tabulková část dotazu. V ní jsou znázorněny tabulky s názvy polí. V dolní části okna je kritériální část (mřížka) určená pro tvorbu kritérií filtru.

Do kritériální mřížky vkládáme pole, na která klademe podmínky. Vlastní vkládání pole děláme buď výběrem názvu pole z hlavičky sloupce kritériální mřížky (nazvané Field), nebo tahem názvu pole myši z tabulkové části dotazu do pole kritériální mřížky.

### Zápis kritérií

- Kritériální podmínky umístěné na stejném řádku mřížky jsou spojeny operátorem **and** (u vyfiltrovaného záznamu musí být splněny všechny tyto podmínky).
- Kritériální podmínky umístěné na různých řádcích mřížky jsou spojeny operátorem **or** (stačí, aby byla u záznamu, který prošel filtrem, splněna jedna z těchto podmínek).

### Řazení vyfiltrovaných dat

- V řádku **Sort** volíme **Ascending** či **Descending**. Více polí bude řazeno zleva doprava.

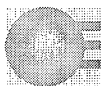
tip

### Praktická zásada pro filtrování tabulek

- Kritéria zadáváme nejprve pomocí filtru podle výběru.
- Teprve v případě, že nějakou podmínku nelze tímto filtrem realizovat, přistoupíme k filtrování podle formuláře, či k rozšířenému filtru.

- 1 Vyzkoušejme, zda lze odstraňovat záznamy z tabulky **Zákazníci** databáze **Northwind**.
- 2 Pracujeme s databází **Northwind**.
  1. V tabulce **Orders** filtrujeme objednávky z roku 1997 z Argentiny nebo Belgie:
    - a) Otevřeme tabulku v Datasheet View. V poli **OrderDate** vyznačíme rok 1997 (u nějakého záznamu z tohoto roku) a klepneme na tlačítko **Filtrovat podle výběru**.
    - b) Vykonáme příkaz **Records/Filter/Advanced Filter/Sort** a do kritériální mřížky umístíme navíc nové pole **ShipCountry**. Nastavíme pro něj kritérium "**Argentina**" or "**Belgium**". Uvozovky si doplní Access sám.
    - c) Klepneme na tlačítko **Apply Filter**.
  2. V tabulce **Orders** filtrujeme objednávky z měsíců červenec nebo srpen.
    - a) Otevřeme tabulku v Datasheet View. V poli **OrderDate** vyznačíme položku **.07.** a klepneme na tlačítko **Filter By Selection**.
    - b) Vykonáme příkaz **Records/Filter/Advanced Filter/Sort** a podmínku kritériální mřížky pro pole **OrderDate** upravíme na  
**Like "\*.07.\*" Or Like "\*.08.\*"**
    - c) Klepneme na tlačítko **Apply Filter**.
  3. V tabulce **Products** vyfiltrujeme výrobky ve sklenicích:  
Otevřeme tabulku v zobrazení datového listu. V poli **QuantityPerUnit** najdeme slovo **bottles** a vyznačíme jeho část **bottle**. Klepneme na tlačítko **Filter By Selection**.
  4. V tabulce **Products** vyfiltrujeme výrobky pro které je minimální úroveň větší, než jednotky na skladě:
    - a) Otevřeme tabulku v Datasheet View.
    - b) Vykonáme příkaz **Records/Filter/Advanced Filter/Sort**.
    - c) Kritériální mřížku vyplníme podle obrázku a klepneme na tlačítko **Apply Filter**.

|           |                 |
|-----------|-----------------|
| Field:    | ReorderLevel    |
| Sort:     |                 |
| Criteria: | >[UnitsInStock] |
| or:       |                 |



## Tabulky – shrnutí a zobecnění základních pojmů

### Jména v Accessu

V programu Microsoft Access mohou být názvy dlouhé až 64 znaků a mohou obsahovat libovolnou kombinaci písmen, číslic, mezer a zvláštních znaků s výjimkou tečky (.), vykřičníku (!), obráceného apostrofu (') a hranatých závorek ([ ]).

### Databáze

je souhrn perzistentních dat, mezi kterými mohou existovat vzájemné vztahy. Zdrojem dat v databázi Access jsou tabulky. V každé uchováváme informaci o jednom typu objektů - entit.

### Třída entit

Okruh všech možných entit stejného typu (knihy, výrobky, zaměstnanci apod.), pro které zakládáme novou tabulku databáze

### Množina entit

Konkrétní prvky ze třídy entit, o nichž jsme si uložili informace v tabulce databáze.

### Návrh tabulky

by měl být proveden tak, aby se do ní daly uložit údaje o kterémkoli prvku ze třídy entit, pro kterou je určena.

### Atributy třídy entit

- Jsou to vlastnosti, kterými popisujeme prvky třídy entit.
- V tabulkách je umísťujeme do sloupců.

### Slouží k

- uložení informace o entitě
- jednoznačné identifikaci entit
- popisu vztahů mezi různými třídami entit.



## Tabulka



Obsahuje data týkající se jedné třídy entit.

Tvoří ji minimálně jeden sloupec-pole a žádný (triviální případ) či více (častější případ) řádků.

Data v i-tém sloupci pocházejí z množiny dat téhož typu nazvané doména i-tého sloupce (domény různých sloupců nemusí být různé).

## Sloupec - Pole

Musí mít jméno, které je uvnitř tabulky jedinečné. Identifikace sloupce:

NázevTabulky.NázevSloupce

[Název Tabulky].[Název Sloupce] – Jsou-li ve jménech povolené nealfanumerické znaky, např. mezery, nutno dát jména do hranatých závorek.

Každý sloupec může obsahovat data pouze jediného předem specifikovaného datového typu.

## Nadklíče

Nadklíč třídy entit je skupina atributů, které jednoznačně každou entitu třídy (řádek tabulky) charakterizují.

Klíč je minimální nadklíč. Odstraníme-li nějaký atribut z klíče, nedostaneme nadklíč. Klíčů může být v tabulce víc.

## Primární klíč

Primární klíč je konkrétní zvolený klíč.

Primární klíč může zadat uživatel, nebo jej vytvoří automaticky Access. V každém případě však musí být jednoznačně řečeno, že jde o primární klíč.

V tabulkách Accessu se povinně nevyžaduje, je však potřebný při tvorbě relací mezi tabulkami.

## Primární klíč vs. pořadové číslo záznamu

Hodnota primárního klíče je neměnná charakteristika záznamu. Přiděluje-li primární klíč Access, pak odstraníme-li záznam z databáze, nepřidělí „takto uvolněnou hodnotu“ novému záznamu.

Pořadové číslo záznamu se naopak může měnit (řazení, odstraňování záznamů).

## Cizí klíč tabulky - nevlastní klíč

Obsahuje hodnoty primárního klíče té tabulky, s níž je původní tabulka v relaci. Nepožaduje se, aby byl klíčem v tabulce. Tedy v jeho hodnotách mohou být duplicity.



## Normalizace tabulek

je testovací proces tabulek, který zajistí, že databázová aplikace bude s databází pracovat co možná nejúčinněji.

Zabrání nadbytečnosti v datech tabulek.

### Vybraná pravidla normalizace

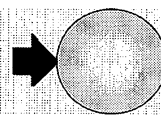
Pole tabulek by měla být atomická (s nedělitelnými údaji).

Tabulka by měla obsahovat primární klíč.

Každé pole tabulky by mělo poskytovat dodatečnou informaci o objektu, který záznam popisuje a pouze o tomto objektu (Boyce-Coddova podmínka).

Pole v tabulce by na sobě neměla být závislá (výjimka: primární klíč - ostatní pole).

## Shrnutí



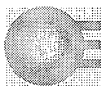
- V Datasheet View tabulky můžeme záznamy editovat, přidávat, odstraňovat i řadit.
- Pro zpřehlednění práce s dlouhými záznamy můžeme pole tabulky skrývat nebo ukotvovat.
- Zobrazíme-li datový list tabulky, která je spojená s jinou tabulkou relací 1:N, objeví se na začátku každého záznamu znaménko plus, pomocí něhož zobrazíme související záznamy z propojené tabulky.
- V Datasheet View tabulky lze použít filtr, který zobrazí pouze záznamy splňující stanovená kritéria.



# Relace

## *Přehled témat*

- typy relací
- tvorba relací
- referenční integrita
- normalizace a normální formy



## Relace a jejich typy

### Relace mezi dvěma tabulkami

popisuje vztah mezi tabulkami. Často se vyjadřuje jako vztah mezi primárním klíčem první tabulky a cizím klíčem druhé.

Rozlišujeme relace 1:1, 1:N, M:N.

### Relace 1:1

Každému záznamu z první tabulky může odpovídat maximálně jeden záznam z tabulky druhé a naopak.

Hodnota primárního klíče každého záznamu v první tabulce může souhlasit s nejvýše jednou hodnotou cizího klíče druhé tabulky a naopak.

Příklad: člověk - platný občanský průkaz

Odpovídá-li každému záznamu z první tabulky přesně jeden záznam z tabulky druhé, budeme říkat, že jde o relaci přesně 1:1.

Tabulka Accessu může obsahovat maximálně 255 polí. Je-li nutné charakterizovat nějakou třídu entit více poli, nutno pro ni vytvořit dvě tabulky. Záznamy v těchto tabulkách jsou v relaci přesně 1:1.

### Relace 1:N, neboli 1:více

Každému záznamu z tabulky na straně jedna může odpovídat více záznamů z tabulky na straně N. Každému záznamu z tabulky na straně N odpovídá maximálně jeden záznam z tabulky na straně jedna.

Jde o relaci mezi primárním klíčem z tabulky na straně jedna a cizím klíčem z tabulky na straně N.

Hodnota primárního klíče každého záznamu v tabulce na straně jedna může souhlasit s hodnotou cizího klíče několika záznamů v tabulce na straně N. Hodnota cizího klíče každého záznamu v tabulce na straně N souhlasí s hodnotou primárního klíče nejvýše jednoho záznamu v tabulce na straně jedna.

Příklad relace: Knihy – nakladatelství. Jedno nakladatelství může vydat více knih, každá kniha byla vydána jedním nakladatelstvím.

### Relace M:N

Každému záznamu jedné tabulky může odpovídat více záznamů druhé tabulky a naopak.

Hodnota primárního klíče každého záznamu v první tabulce může souhlasit s hodnotou odpovídajícího cizího klíče několika záznamů v druhé tabulce a naopak.

Příklad: Autoři - Knihy: Jeden autor může napsat více knih. Jedna kniha může být napsána více autory (spoluautorství).

## Osiřelé záznamy



Jsou záznamy tabulky, kterým nepřísluší odpovídající záznamy v druhé tabulce, s níž je tato v relaci. Všechny tři typy relací připouštějí v obou tabulkách osiřelé záznamy.

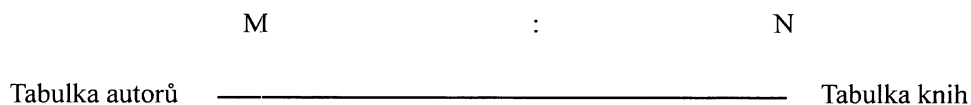
## Převod relace M:N na dvě relace 1:N

Relaci M:N musíme realizovat jako dvě relace 1:N.

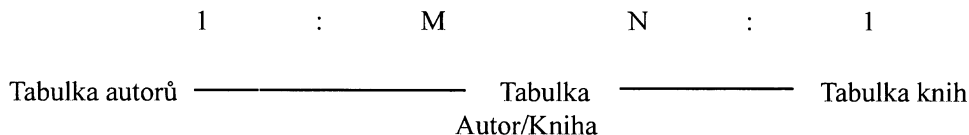
Vytvoříme třetí (tzv. spojovací) tabulku. Do této tabulky přidáme pole cizího klíče pro každou z těchto tabulek. Každá z původních tabulek je s touto spojovací tabulkou v relaci jedna ku více. V případě potřeby můžeme do spojovací tabulky přidat i další pole, stejně jako u jakékoli jiné tabulky.

## Převod relace Autoři-Knihy

Původní relace



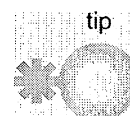
Po převodu na dvě relace jedna ku více



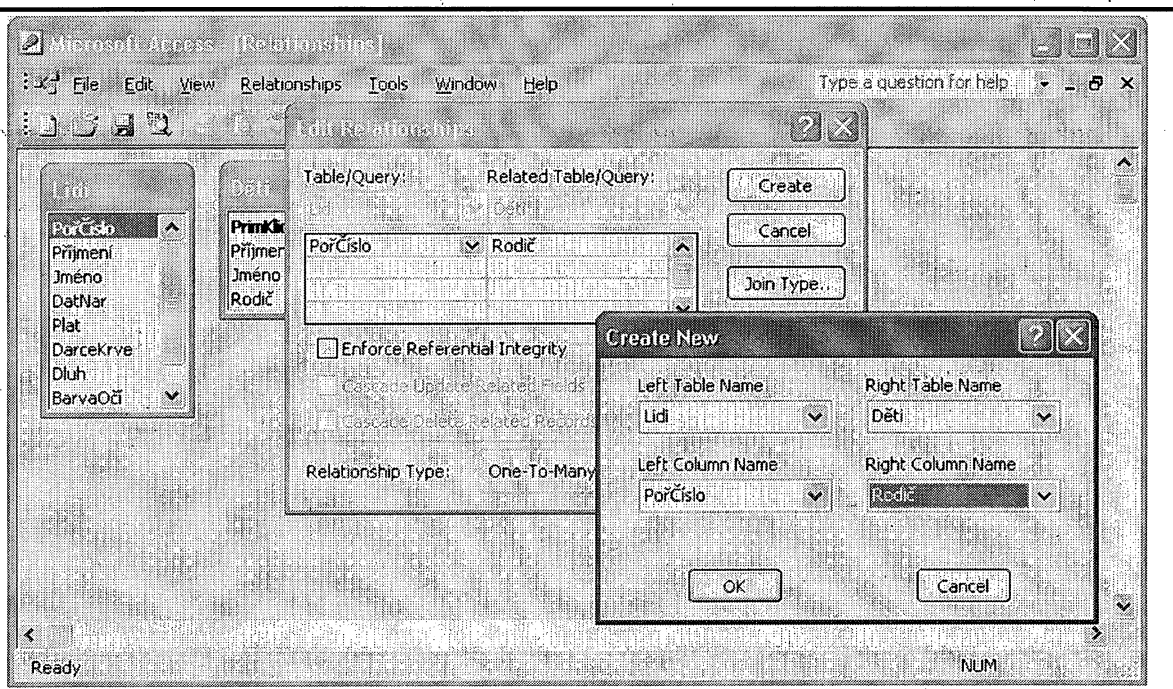
*V převodní tabulce lze s výhodou používat Lookup Wizard. Namísto hodnot primárního klíče autora lze v poli cizí klíč zobrazovat např. příjmení autora apod. (viz pole Rodič v tabulce Děti).*



*V převodní tabulce lze jako primární klíč použít obě dvě pole cizích klíčů.*



# Tvorba relací mezi tabulkami v Accessu



## Tvorba relací mezi tabulkami v Accessu



Je-li aktivní okno databáze, zobrazíme tlačítkem **Relationships** (*Tools/Relationships*) okno **Relationships**.



Tlačítkem **Show Table** (*Relationships/Show Table*) přidáme do okna **Relationships** příslušné tabulky.

Vykonáme příkaz *Relationships/Edit Relationship*.

V jeho dialogovém okně klepneme na tlačítko **Create New**.

V dialogovém okně **Create New** určíme názvy tabulek a názvy polí primárního a cizího klíče.

Typ relace si určuje Access sám.

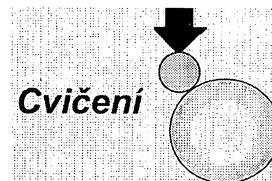
Děláme-li relaci mezi primárními klíči, bude to relace 1:1.

Děláme-li relaci mezi primárním a cizím klíčem, bude to relace 1:N

tip

Relaci mezi dvěma tabulkami lze v okně *Relace* vytvořit přetáhnutím sloupce primárního klíče jedné tabulky na sloupec cizího klíče druhé (či naopak).





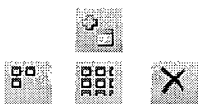
## Úkoly k procvičování:

- 1 V databázi **ZamDěti** zkontrolujeme, zda existuje mezi tabulkami **Lidi** a **Děti** relace. Pokud ano, tak ji odstraníme a vytvoříme znovu:
  - a) Otevřeme databázi **ZamDěti** (*File/Open/ZamDěti.mdb*)
  - b) Klepneme na tlačítko **Relationships** panelu nástrojů **Database**.
  - c) Je-li v okně **Relationships** mezi tabulkami **Lidi** a **Děti** spojnice, klepneme na ni a klávesou **Delete** relaci odstraníme.
  - d) Poté „přeneseme” tahem myši **primární klíč tabulky Lidi** na pole **Rodič** tabulky **Děti**.
  - d) Objeví se okno **Edit Relationships**, klepneme na tlačítko **Create**.

- 2 Prohlédneme si v databázi **Northwind** okno relací. Klepněme do tabulky **Orders** a klávesou **Delete** ji odstraníme z okna relací. Byly též odstraněny relace této tabulky s ostatními?

Při otevřeném okně **Relationships** klepneme na tlačítko **Show Table** a z jeho dialogového okna vložíme tabulku **Orders** zpět do okna **Relationships**. Relace s ostatními tabulkami zůstaly zachovány.

- 3 Práce s oknem **Relationships** v databázi **Northwind**:



Potřebná tlačítka panelu nástrojů **Relationship**, zleva doprava : **Show Table**, **Show Direct Relationships**, **Show All Relationships**, **Clear Layout**.

- a) Otevřeme okno **Relationships** a klepneme na tlačítko **Clear Layout**. Okno **Relationships** bude prázdné.
- b) Klepneme na tlačítko **Show Table** a z jeho dialogového okna vložíme tabulku **Orders** zpět do okna **Relationships**.
- c) Klepneme na tlačítko **Show Direct Relationships**. Zobrazí se pouze relace tabulky **Orders** s jinými tabulkami.
- d) Klepneme na tlačítko **Show All Relationships**. Zobrazí se všechny relace databáze.
- d) Zavřeme okno **Relationships**. Na dotaz k uložení změn odpovíme záporně.
- e) Otevřeme okno **Relationships**. Zjistíme, že jeho rozvržení zůstalo v takové formě, jako před započítím této úlohy.

# Referenční integrita (RI) pro relaci 1:N

**Edit Relationships**

| Table/Query: | Related Table/Query: |
|--------------|----------------------|
| Lidi         | Děti                 |
| PorČíslo     | Rodič                |

☒ Enforce Referential Integrity  
☐ Cascade Update Related Fields  
☐ Cascade Delete Related Records

Relationship Type: One-To-Many

Buttons: OK, Cancel, Join Type..., Create New...



## Referenční integrita (Referential Integrity) pro relaci 1:N

Referenční integrita relaci 1:N zpřísňuje.

Každému záznamu v tabulce na straně N s hodnotou cizího klíče  $\diamond$  NULL musí odpovídat právě jeden záznam v tabulce na straně jedna. To znamená, že v tabulce na straně N nemohou existovat osiřelé záznamy (takové, které nemají odpovídající záznam v tabulce na straně jedna).

Existují-li v tabulce na straně N osiřelé záznamy, nelze referenční integritu mezi tabulkami nadefinovat. Je nutno tyto osiřelé záznamy odstranit nebo doplnit potřebné záznamy v tabulce na straně jedna.

Nemůže-li Access z výše uvedených důvodů vytvořit RI, vypíše následující hlášení

**Microsoft Office Access**

**Microsoft Office Access can't create this relationship and enforce referential integrity.**

Data in the table 'Děti' violates referential integrity rules.  
For example, there may be records relating to an employee in the related table, but no record for the employee in the primary table.

Edit the data so that records in the primary table exist for all related records.  
If you want to create the relationship without following the rules of referential integrity, clear the Enforce Referential Integrity check box.

OK

## Důsledky referenční integrity

Jestliže nastavíme referenční integritu, program Microsoft Access zabrání

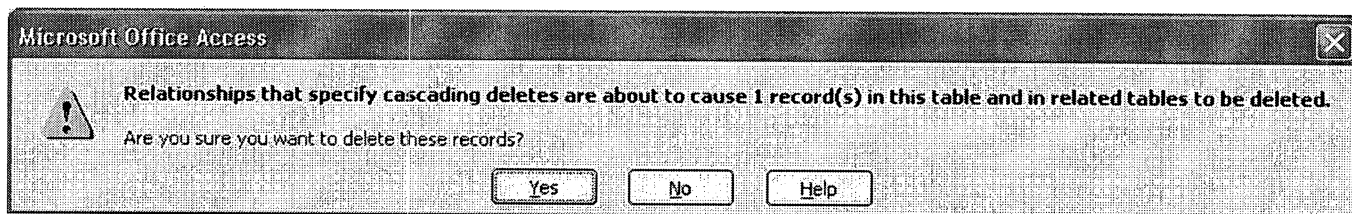
- přidat záznamy do tabulky na straně N, jestliže v tabulce na straně jedna neexistuje odpovídající záznam
- změně hodnoty v tabulce na straně jedna, která by mohla mít za následek vznik osiřelých záznamů v tabulce na straně N
- odstranění záznamů z tabulky na straně jedna, jestliže tabulka na straně N obsahuje příslušné související záznamy (tyto by osiřely)

## Další vlastnosti relace s referenční integritou

Výše uvedené důsledky referenční integrity lze modifikovat následovně.

Po nastavení referenční integrity lze aktivovat možnost **Cascade Updates Related Fields**. V tom případě se změna primárního klíče v tabulce na straně jedna provede i v hodnotách cizích klíčů odpovídajících záznamů.

Po nastavení referenční integrity lze aktivovat možnost **Cascade Delete Related Records**. Odstraníme-li potom záznam z tabulky na straně jedna, dojde k odstranění všech souvisejících záznamů z tabulky na straně N relace. Tuto akci nelze vrátit zpět.



### Úprava relace:

Na spojnici mezi tabulkami poklepeme. Objeví se dialogové okno **Edit Relationships**. Okno lze zobrazit i příkazem **Relationships/Edit Relationship**.

poznámka



# První normální forma

| Číslo zaměstnance | Jméno           | Kód oddělení | Název oddělení       | Číslo projektu 1 | Plnění 1 | Číslo projektu 2 | Plnění 2 |
|-------------------|-----------------|--------------|----------------------|------------------|----------|------------------|----------|
| EN1-36            | Jitka Krákorová | FU           | Finance a účetnictví | 35-152-TC        | 90,00%   |                  |          |
| EN1-38            | David Slánský   | IS           | Informační systémy   | 36-272-TC        | 75,00%   |                  |          |
| EN1-26            | Jan Bejda       | TP           | Technická podpora    | 30-452-T3        | 25,00%   | 30-457-T3        | 40,00%   |
| EN1-33            | Jiří Svoboda    | TP           | Technická podpora    | 30-452-T3        | 50,00%   | 30-482-TC        | 35,00%   |
| EN1-35            | Andrej Meravý   | TP           | Technická podpora    | 30-452-T3        | 25,00%   | 31-238-TC        | 80,00%   |
| EN1-40            | Ondřej Jung     | TP           | Technická podpora    | 31-238-TC        | 20,00%   | 31-241-TC        | 70,00%   |
| *                 |                 |              |                      |                  |          |                  |          |

| Číslo zaměstnance | Jméno  | Příjmení  | Kód oddělení | Název oddělení       | Číslo projektu | Plnění |
|-------------------|--------|-----------|--------------|----------------------|----------------|--------|
| EN1-26            | Jan    | Bejda     | TP           | Technická podpora    | 30-452-T3      | 25,00% |
| EN1-26            | Jan    | Bejda     | TP           | Technická podpora    | 30-457-T3      | 40,00% |
| EN1-33            | Jiří   | Svoboda   | TP           | Technická podpora    | 30-452-T3      | 50,00% |
| EN1-33            | Jiří   | Svoboda   | TP           | Technická podpora    | 30-482-TC      | 35,00% |
| EN1-35            | Andrej | Meravý    | TP           | Technická podpora    | 30-452-T3      | 25,00% |
| EN1-35            | Andrej | Meravý    | TP           | Technická podpora    | 31-238-TC      | 80,00% |
| EN1-36            | Jitka  | Krákorová | FU           | Finance a účetnictví | 35-152-TC      | 90,00% |
| EN1-38            | David  | Slánský   | IS           | Informační systémy   | 36-272-TC      | 75,00% |
| EN1-40            | Ondřej | Jung      | TP           | Technická podpora    | 31-238-TC      | 20,00% |
| EN1-40            | Ondřej | Jung      | TP           | Technická podpora    | 31-241-TC      | 70,00% |
| *                 |        |           |              |                      |                |        |



## Normalizace

Při návrhu databází bychom měli něco vědět o standardech, kterým se říká **normální formy**. Úpravě struktury vaší databáze tak, aby tabulky splňovaly tyto standardy, se říká **normalizace dat**.

Existuje pět normálních forem a každá se vztahuje ke specifickému problému organizace dat. Normální formy se musejí aplikovat postupně, od první normální formy postupně k vyšším.

Normalizace není nezbytně nutná, ale se znalostí normálních forem budete navrhovat svoje tabulky mnohem efektivněji. Většinou se používají první tři normální formy.

## První normální forma

Tabulka je v první normální formě, pokud splňuje dvě podmínky:

- obsahuje pouze jednoduché nedělitelné údaje
- neopakují se v ní stejné skupiny polí.

První tabulka na obrázku tyto podmínky nesplňuje (první zásadu nesplňuje pole **Jméno**, druhou nesplňují opakující se pole **Č. projektu** a **Plnění**)

Druhá tabulka již představuje evidenci stejných údajů v první normální formě. S novým projektem přibudou nové záznamy, ale struktura polí tabulky zůstane stejná.

## Druhá normální forma

| Číslo zaměstnance | Číslo projektu | Plnění |
|-------------------|----------------|--------|
| EN1-26            | 30-452-T3      | 25,00% |
| EN1-33            | 30-452-T3      | 50,00% |
| EN1-35            | 30-452-T3      | 25,00% |
| EN1-36            | 35-152-TC      | 90,00% |
| EN1-38            | 36-272-TC      | 75,00% |
| EN1-40            | 31-238-TC      | 20,00% |
| EN1-26            | 30-457-T3      | 40,00% |
| EN1-40            | 31-241-TC      | 70,00% |
| EN1-35            | 31-238-TC      | 80,00% |
| EN1-33            | 30-482-TC      | 35,00% |
| *                 |                |        |

| Číslo zaměstnance | Jméno  | Příjmení  | Kód oddělení | Název oddělení       |
|-------------------|--------|-----------|--------------|----------------------|
| EN1-26            | Jan    | Bejda     | TP           | Technická podpora    |
| EN1-33            | Jiří   | Svoboda   | TP           | Technická podpora    |
| EN1-35            | Andrej | Meravý    | TP           | Technická podpora    |
| EN1-36            | Jitka  | Krákorová | FU           | Finance a účetnictví |
| EN1-38            | David  | Slánský   | IS           | Informační systémy   |
| EN1-40            | Ondřej | Jung      | TP           | Technická podpora    |
| *                 |        |           |              |                      |

## Druhá normální forma

V tabulce s daty v druhé normální formě musejí být všechny **neklíčové sloupce závislé na úplném primárním klíči**, nejen na jeho části.

V tabulkách se obvykle druhé normální formy dosahuje rozdělením na dvě i více tabulek menších.

Tabulka v první normální formě z příkladu na předchozí straně má složený primární klíč, který tvoří pole **Číslo zaměstnance** a **Číslo projektu**. Každé z těchto polí obsahuje duplicitní údaje. Avšak dvojice údajů v těchto polích jsou jedinečné.

Pole **Plnění** se vztahuje k oběma polím složeného primárního klíče. Údaj říká, kolik odpracoval určitý zaměstnanec na určitém projektu. Jedna tabulka bude muset tedy obsahovat pole **Plnění**, **Číslo zaměstnance** a **Číslo projektu**.

Ostatní pole se vztahují k poli **Číslo zaměstnance**.

V dolní části obrázku vidíte původní tabulku rozdělenou na dvě související, které již jsou ve druhé normální formě.



## Třetí normální forma

|   | Číslo zaměstnance | Číslo projektu | Plnění |
|---|-------------------|----------------|--------|
| ▶ | EN1-26            | 30-452-T3      | 25,00% |
|   | EN1-33            | 30-452-T3      | 50,00% |
|   | EN1-35            | 30-452-T3      | 25,00% |
|   | EN1-36            | 35-152-TC      | 90,00% |
|   | EN1-38            | 36-272-TC      | 75,00% |
|   | EN1-40            | 31-238-TC      | 20,00% |
|   | EN1-26            | 30-457-T3      | 40,00% |
|   | EN1-40            | 31-241-TC      | 70,00% |
|   | EN1-35            | 31-238-TC      | 80,00% |
|   | EN1-33            | 30-482-TC      | 35,00% |

|   | Kód oddělení | Název oddělení       |
|---|--------------|----------------------|
| ▶ | TP           | Technická podpora    |
|   | FU           | Finance a účetnictví |
|   | IS           | Informační systémy   |
|   | TP           | Technická podpora    |

| * | Číslo zaměstnance | Jméno  | Příjmení  | Kód oddělení |
|---|-------------------|--------|-----------|--------------|
| ▶ | EN1-26            | Jan    | Bejda     | TP           |
|   | EN1-33            | Jiří   | Svoboda   | TP           |
|   | EN1-35            | Andrej | Meravý    | TP           |
|   | EN1-36            | Jitka  | Krákorová | FU           |
|   | EN1-38            | David  | Slánský   | IS           |
|   | EN1-40            | Ondřej | Jung      | TP           |
| * |                   |        |           |              |



### Třetí normální forma

Tabulka ve třetí normální formě nesmí obsahovat neklíčová pole, která se přímo nevztahují k primárnímu klíči.

V našem případě je na **Čísle zaměstnance** přímo závislé jen jedno z polí **Kód oddělení** a **Název oddělení**. To druhé je potom závislé nepřímo (tranzitivně).

Zobrazené tabulky jsou již ve třetí normální formě.

### Další normální formy

Vhodnost čtvrté a páté normální formy bývá často předmětem akademických diskusí. Uveďme si jejich charakteristiky a ukázky.

## Čtvrtá normální forma

Tabulka je ve čtvrté normální formě, jestliže neobsahuje dva (nebo více) na sobě nezávislých vícehodnotových sloupců.

Související tabulky 2 a 3 na obrázku představují převod tabulky 1 do čtvrté normální formy.

| Zaměstnanec | Dovednost         | Jazyk    |
|-------------|-------------------|----------|
| Karel       | psaní na stroji   |          |
| Karel       | práce s počítačem |          |
| Karel       |                   | česky    |
| Karel       |                   | anglicky |
| Karel       |                   | německy  |

tabulka 1

| Zaměstnanec | Dovednost         |
|-------------|-------------------|
| Karel       | psaní na stroji   |
| Karel       | práce s počítačem |

tabulka 2

| Zaměstnanec | Jazyk    |
|-------------|----------|
| Karel       | česky    |
| Karel       | anglicky |
| Karel       | německy  |

tabulka 3

## Pátá normální forma

Tabulka je v páté normální formě, jestliže je ve čtvrté normální formě a dále neobsahuje dva (nebo více) vícehodnotových sloupců. V podstatě jde o podobný případ, jako u čtvrté normální formy, ale sloupce tentokrát nemusejí být nezávislé.

Související tabulky 5,6 a 7 na obrázku představují převod tabulky 4 do páté normální formy.

| Prodejce | Společnost | Předmět prodeje     |
|----------|------------|---------------------|
| Novák    | Ford       | osobní automobily   |
| Pokorný  | GM         | osobní automobily   |
| Pokorný  | GM         | nákladní automobily |
| Pokorný  | Ford       | osobní automobily   |

tabulka 4

| Prodejce | Společnost |
|----------|------------|
| Novák    | Ford       |
| Pokorný  | GM         |
| Pokorný  | Ford       |

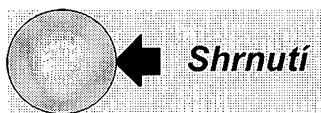
tabulka 5

| Společnost | Předmět prodeje     |
|------------|---------------------|
| Ford       | osobní automobily   |
| GM         | osobní automobily   |
| GM         | nákladní automobily |

tabulka 6

| Prodejce | Předmět prodeje     |
|----------|---------------------|
| Novák    | osobní automobily   |
| Pokorný  | nákladní automobily |
| Pokorný  | osobní automobily   |

tabulka 7



- Access podporuje relace typu 1:1 a 1:N. Pokud jsou dvě tabulky ve vztahu M:N, je třeba vyrobit třetí spojovací tabulku.
- Relace vytváříme v okně relací (tlačítko **Relationships** nebo *Tools/Relationships*).
- Referenční integrita mezi tabulkami spojenými relací 1:N vhodně hlídá přidávání, odstraňování a aktualizaci záznamů.
- Využívejme možnosti nastavení kaskádní aktualizace a kaskádního odstraňování.

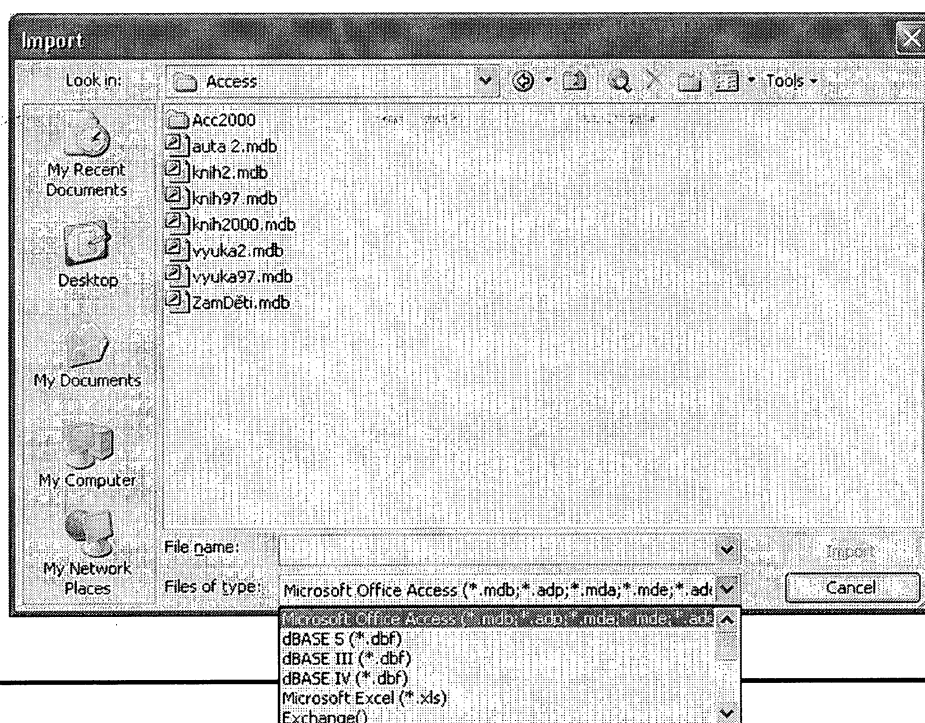


# Export a import

## *Přehled témat*

- export dat
- import dat a propojení tabulek
- jazyk XML
- rozdělení databáze

# Export, import a propojení dat



## Datové formáty pro export, import a propojení

MS Access spolupracuje s běžnými formáty dat. Z nejčastěji používaných uvádíme: jiná databáze Access či projekt Accessu, Dbase, Paradox, Excel, Lotus, textové soubory, soubory HTML, soubory XML, rozhraní ODBC. Bližší údaje najdeme v nápovědě.

## Export dat

V okně databáze vyznačíme objekt a volíme příkaz **File/Export**, či příkaz **Export** z místní nabídky objektu. V seznamu **Save as type** dialogového okna, v němž určíme cílový soubor pro export, vyznačíme datový formát, do kterého exportujeme, a zadáme název souboru, do kterého export proběhne.

## Import dat, propojení tabulek

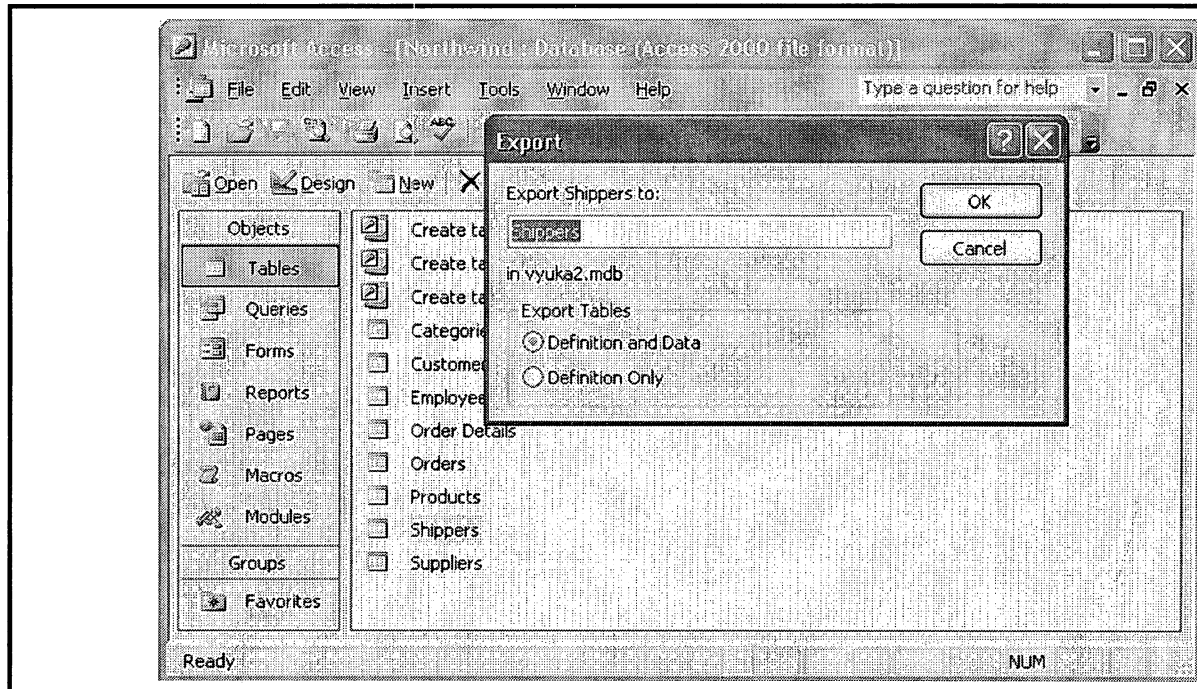
Provedeme příkaz **File/Get External data/Import** resp. **Link Tables** (lze též použít příkazu **Import** či **Link Tables** z místní nabídky okna databáze).

**Při importu** dat vzniká v naší databázi nová cílová tabulka obsahující data z použitého zdroje. Mezi zdrojovými a cílovými daty již není žádný vztah.

**Při propojení** tabulek vzniká v naší databázi propojení na tabulku zvoleného datového zdroje. To znamená, že změny dělané v původní tabulce se propojením přenášejí do tabulky v naší databázi, ale i naopak!

V dalším se budeme podrobněji věnovat exportu a importu z jiné databáze Accessu, importu ze sešitu Excelu a exportu i importu dat ve formátu XML.

## Export do jiné databáze Accessu

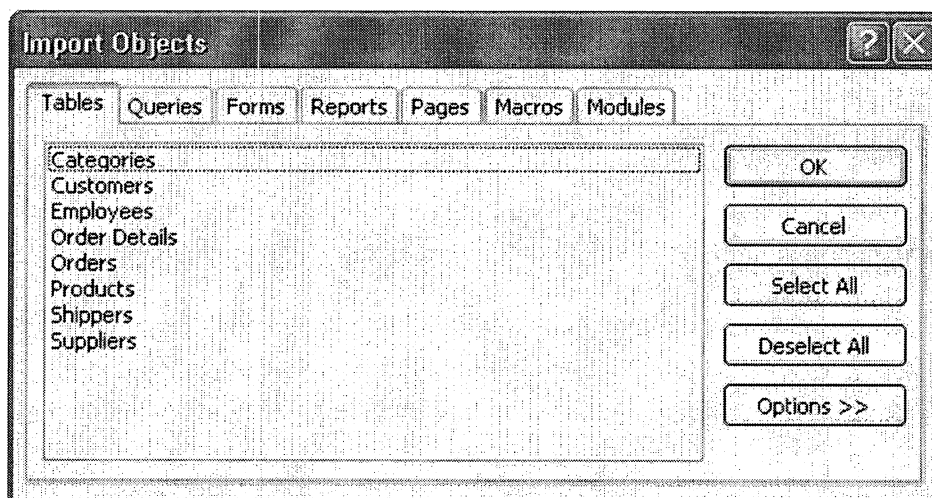


## Export do jiné databáze Accessu

Po vykonání příkazu pro export specifikujeme cílovou databázi, název tabulky v ní a určíme, zda exportujeme definici tabulky i data, nebo jen definici (v tom případě získáme v cílové databázi prázdnou tabulku stejných vlastností, ale bez dat).



## Import z jiné databáze Accessu



Provedeme příkaz **File/Get External Data/Import** (lze též použít příkazu **Import** kontextové nabídky okna databáze ). Po specifikaci databázového souboru lze v okně Import objektů vyznačit nejen tabulky, ale i všechny další objekty (viz karty tohoto okna), které chceme do naší databáze importovat.

## Import z tabulky Excelu

**Import Spreadsheet Wizard**

You can specify information about each of the fields you are importing in the area below. You can then modify field information in the 'Field Options' dialog box.

Field Options

Field Name: **Typ vozu** Data type: **Text**

Indexed: **No** ☐ Do not import

|   | Typ vozu | Značka        | SPZ       | Rok výroby |
|---|----------|---------------|-----------|------------|
| 1 | autobus  | Neoplan       | AZ-52-60  | 1985       |
| 2 | autobus  | Karosa LC 735 | AO-23-32  | 1992       |
| 3 | mikrobus | Fiat Ducato   | ASD-45-07 | 1992       |
| 4 | mikrobus | Mercedes 307  | ALK-55-56 | 1991       |
| 5 | mikrobus | Ford Transit  | ARR-12-67 | 1994       |
| 6 | nákladní | Avia 30       | AAZ-28-90 | 1987       |

Buttons: Cancel, < Back, Next >, Finish

**Import Spreadsheet Wizard**

Microsoft Access recommends that you define a primary key for your new table. A primary key is used to uniquely identify each record in your table. It allows you to retrieve data more quickly.

☒ Let Access add primary key.

☐ Choose my own primary key. **ID**

☐ No primary key.

| ID | Typ vozu | Značka        | SPZ       | Rok výroby |
|----|----------|---------------|-----------|------------|
| 1  | autobus  | Neoplan       | AZ-52-60  | 1985       |
| 2  | autobus  | Karosa LC 735 | AO-23-32  | 1992       |
| 3  | mikrobus | Fiat Ducato   | ASD-45-07 | 1992       |
| 4  | mikrobus | Mercedes 307  | ALK-55-56 | 1991       |
| 5  | mikrobus | Ford Transit  | ARR-12-67 | 1994       |
| 6  | nákladní | Avia 30       | AAZ-28-90 | 1987       |

Buttons: Cancel, < Back, Next >, Finish



## Import z tabulky Excelu

Importovanou oblast Excelu je výhodné nejdříve v Excelu pojmenovat. Potom provedeme příkaz **File/Get External Data/Import** (lze též použít příkaz **Import** kontextové nabídky okna databáze – objekt Tabulky). Po specifikaci excelovského souboru se spustí průvodce importem, který nám dá vybrat jména oblastí či listů sešitu Excelu. Pro každé pole importované tabulky lze volit, zda bude vůbec import pole proveden, nový název pole a indexování podle tohoto pole. Dále lze určit, jak bude nadefinován (a zda vůbec) primární klíč.

**Import Spreadsheet Wizard**

That's all the information the wizard needs to import your data.

Import to Table: **AUTA**

☒ I would like a wizard to analyze my data.

☐ Display Help after the wizard is finished.

Buttons: Cancel, < Back, Next >, Finish

**Table Analyzer Wizard**

Is the wizard grouping information correctly?  
If not, you can drag and drop fields to form groups that make sense.

What name do you want for each table?  
It's a good idea for the name to identify the kind of information the table contains.

| Table1                   | Generated Unique ID |
|--------------------------|---------------------|
| ID                       | Barva               |
| Značka                   |                     |
| SPZ                      |                     |
| Rok výroby               |                     |
| Počet najetých km (k 11) |                     |
| Lookup to Table3         |                     |
| Lookup to Table2         |                     |

Buttons: Cancel, < Back, Next >, Finish

Tabulka bude importována v původní formě. Lze si však navíc vyžádat analýzu struktury po importu dat. V tom případě se po importu spouští analyzátor, který testuje, zda není efektivnější původní excelovskou tabulku rozdělit na více tabulek o menším počtu sloupců, mezi nimiž se nadefinují relace. Tím je dosaženo snížení duplicit v importovaných datech a obvykle se zefektivní další práce s nimi.

## Proč a kdy převádět data z Excelu do Accessu



První zásada, kterou je nutno dodržovat při tvorbě tabulek v Accessu, je, že ukládáme-li do databáze data o novém typu objektů, nutno je dát do nové tabulky. Ukládáme-li např. data o zaměstnancích a jejich dětech, uděláme dvě tabulky, jednu pro zaměstnance, druhou pro jejich děti. V tabulce dětí vytvoříme pole Rodič, v němž ukládáme primární klíč rodiče dítěte z tabulky zaměstnanců. Pole rodič bude cizím klíčem vzhledem k primárnímu klíči zaměstnanců. Mezi oběma tabulkami tak vzniká relace 1:N.

Excel však neumí popsané vztahy mezi tabulkami dělat. Excelovskou databázi je pouze jedna tabulka. Pokud do tabulek Excelu ukládáme data o jednom typu objektů a nepotřebujeme mezi nimi dělat relace, stačí nám na to opravdu jen Excel. V opačném případě je zapotřebí přejít k Accessu. Náš příklad zaměstnanci - jejich děti bychom proto museli ukládat do jedné tabulky Excelu. Záznam pro zaměstnance by tam musel být v tolika výskytech, kolik má dětí, tedy pro každé jeho dítě. Takto bychom zatížili excelovskou databázi obrovským množstvím duplicit.

|   | Jméno     | Příjmení     | ZaměstnánOd | Plat  | Barva očí | ID |
|---|-----------|--------------|-------------|-------|-----------|----|
| ▶ | Daniela   | Černá        | 10.5.2001   | 28000 | zelená    | 1  |
|   | JménoDítě | PříjmeníDítě |             |       |           |    |
| ▶ | Dan       | Černý        |             |       |           |    |
|   | Martina   | Černá        |             |       |           |    |
|   | Jarmila   | Černá        |             |       |           |    |
| * |           |              |             |       |           |    |
| - | Martin    | Novák        | 1.1.2000    | 23000 | modrá     | 2  |
|   | JménoDítě | PříjmeníDítě |             |       |           |    |
|   | Karel     | Novák        |             |       |           |    |
|   | David     | Novák        |             |       |           |    |
|   | Martin    | Novák        |             |       |           |    |
|   | Lucie     | Nováková     |             |       |           |    |
| * |           |              |             |       |           |    |

|   | A       | B        | C           | D     | E         | F         | G            |
|---|---------|----------|-------------|-------|-----------|-----------|--------------|
| 1 | Jméno   | Příjmení | ZaměstnánOd | Plat  | Barva očí | JménoDítě | PříjmeníDítě |
| 2 | Martin  | Novák    | 1.1.2000    | 23000 | modrá     | Karel     | Novák        |
| 3 | Martin  | Novák    | 1.1.2000    | 23000 | modrá     | David     | Novák        |
| 4 | Martin  | Novák    | 1.1.2000    | 23000 | modrá     | Martin    | Novák        |
| 5 | Martin  | Novák    | 1.1.2000    | 23000 | modrá     | Lucie     | Nováková     |
| 6 | Daniela | Černá    | 10.5.2001   | 28000 | zelená    | Dan       | Černý        |
| 7 | Daniela | Černá    | 10.5.2001   | 28000 | zelená    | Martina   | Černá        |
| 8 | Daniela | Černá    | 10.5.2001   | 28000 | zelená    | Jarmila   | Černá        |
| 9 |         |          |             |       |           |           |              |

Export dotazu:

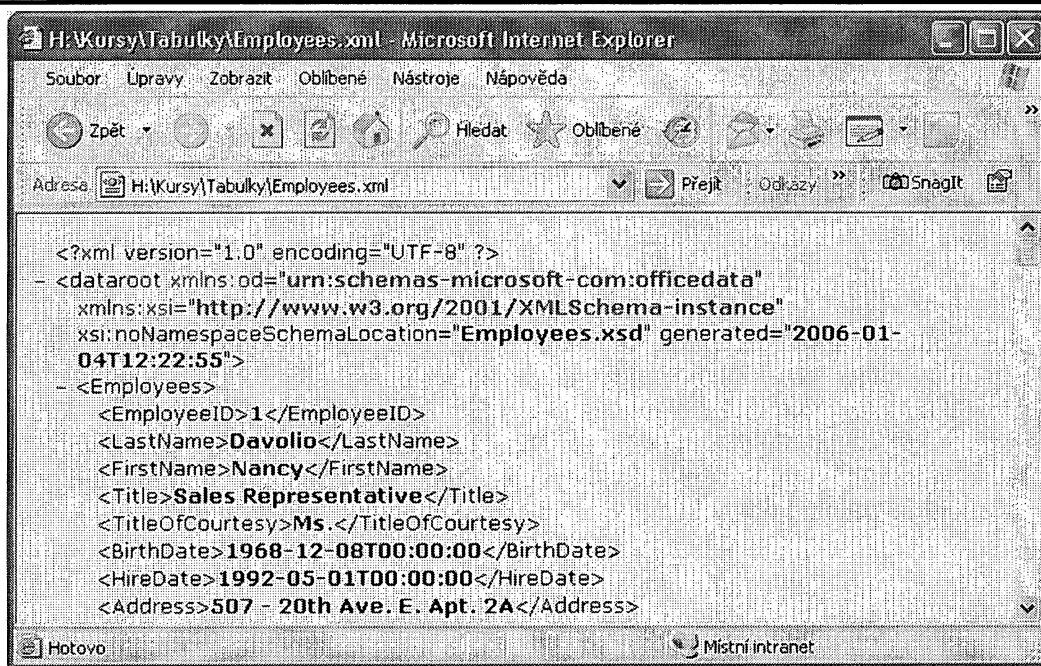
Exportovat do jiné databáze Accessu lze i ostatní objekty (dotazy, formuláře apod.). Protože dotazy a formuláře pracují nad tabulkami, musíme při exportu zvážit, zda není třeba exportovat i tyto tabulky. Pokud budete dotaz zvolený v databázi Accessu exportovat do Excelu, budou exportována **data** vybraná tímto dotazem.

poznámka





# Jazyk XML

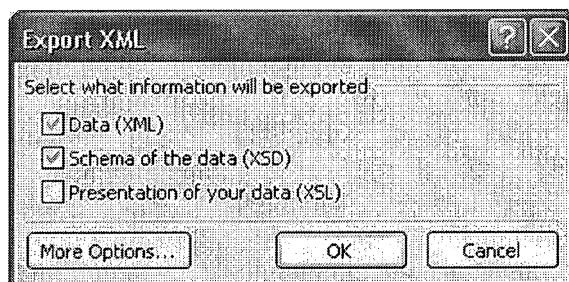


## Jazyk XML

XML (eXtensible Markup Language) je platformě nezávislý značkovací jazyk zaměřený na obsahovou stránku dokumentu. Oproti např. jazyku HTML neobsahuje XML soubor žádné značky definující vzhled/formát výsledného dokumentu, ale jen obsah/význam jednotlivých položek. Jednotlivé značky se vytvářejí podle potřeby. Například při exportu z Excelu se pro každý sloupec tabulky vytvoří jedna značka, do které jsou následně zapisovány hodnoty polí jednotlivých záznamů.

Jazyk XML se výborně hodí pro ukládání a popis dat. Jeho výhoda je v tom, že data jsou uložena standardním způsobem ve znakové formě. S formátem XML lze snadno pracovat na Internetu. Data zakódovaná do XML můžeme lehce načíst do jiné databáze.

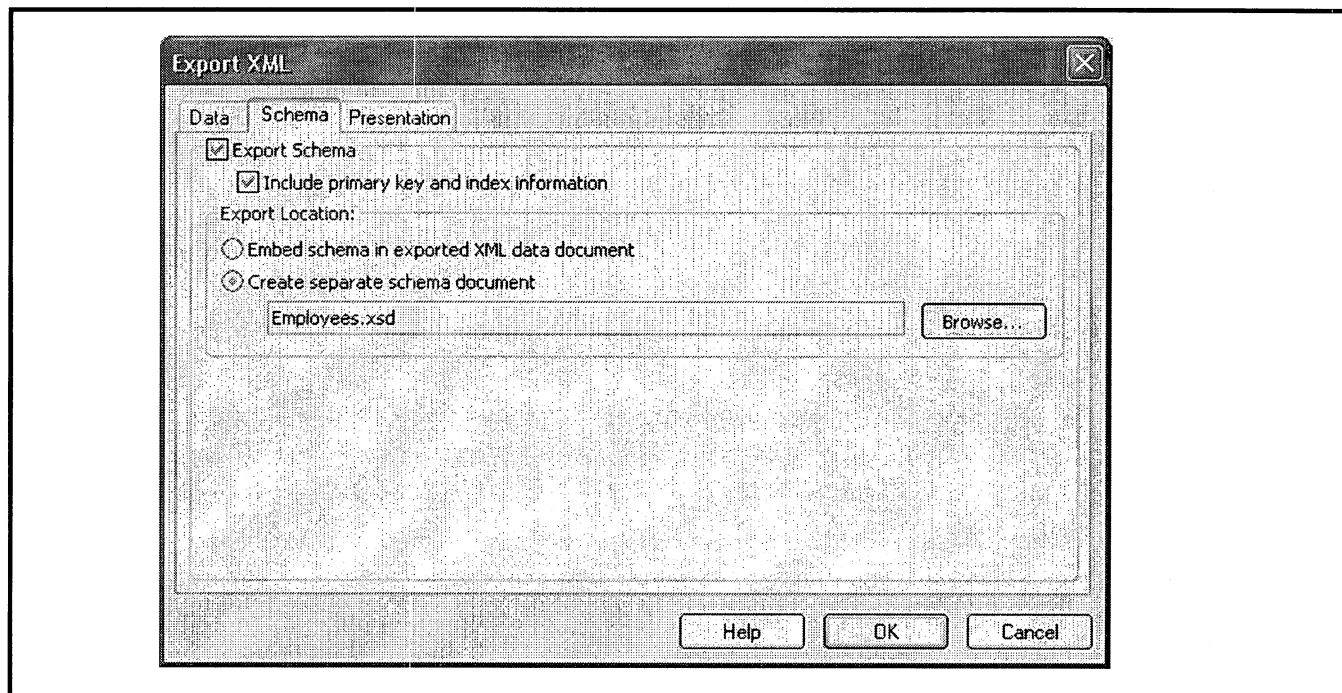
## Export do formátu XML



Export dat do formátu XML děláme standardním způsobem. Vyznačíme zdroj dat (tabulku Accessu v okně databáze) a provedeme příkaz pro export, přičemž volíme typ souboru XML a jeho název. Potom se objeví dialogové okno Export XML, v němž určíme, zda exportujeme pouze data (XML), nebo i datové schéma (XSD), či prezentaci dat ve formátu HTML.

Schéma dat popisuje jejich strukturu. V případě tabulky jsou to názvy a datové typy sloupců a jejich další vlastnosti.

# Jazyk XML



Vlastní data jsou exportována v souboru s příponou XML. Rozhodneme-li se exportovat **schéma dat**, vzniká další soubor s příponou XSD (pokud schéma neuložíme do souboru s daty).



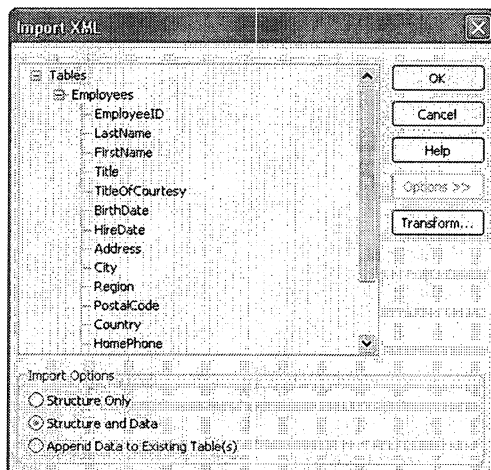
Volíme-li **prezentaci dat**, vznikají další dva soubory. Jeden s příponou HTM je určen pro zobrazování dat v Internet Exploreru, druhý, s příponou XSL, určuje formát zobrazení.

Okno pro nastavení upřesnění vyvoláte tlačítkem **More Options**.

Podobně jako tabulky lze do formátu XML exportovat dotazy, formuláře a sestavy.

## Import dat ve formátu XML

Vlastní import děláme standardním způsobem. Volíme-li k importu soubor s příponou XML, který vznikl exportem spolu se schématem (XSD), je třeba odpovídající schéma při importu použít. Pak máme možnosti uvedené na obrázku. Volíme-li k importu soubor schématu (XSD), dostaneme v případě tabulky pouze prázdnou tabulku bez dat, která má strukturu definovanou tímto souborem.





## Rozdělení databáze

Již na prvních stránkách této příručky jsme uvedli, že databázi Accessu lze rozdělit na dvě části. Část tabulkovou (back-end databázi) a to ostatní (front-end databázi). Tabulkovou část umísťujeme často na server a je sdílená více uživateli. Každý uživatel, který k ní má přístup, si vytvoří svoji „front-end“ databázi a jako její datový zdroj má diskutované tabulky umístěné na serveru. Přístup uživatelů k databázi „back-end“ se řídí oprávněními běžnými ve Windows, a platným zabezpečením v Accessu.

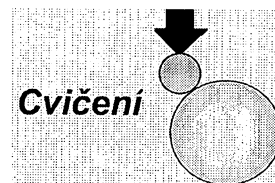
### Postup rozdělení

- Zálohujeme originální databázi.
- Vykonáme příkaz *Tools/Database Utilities/Database Splitter*.

Určíme umístění a název back-end databáze. Z původní databáze se stane databáze „front-end“, zůstanou v ní všechny objekty kromě tabulek. Místo originálních tabulek vzniknou propojené tabulky na databázi back-end.

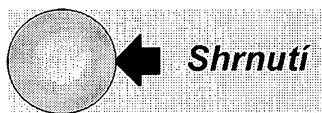
Uživatel, který chce pracovat s back-end databází na serveru si svoji front-end databázi vytvoří takto: udělá si novou databázi a na kartě Tabulky okna databáze vytvoří propojení na tabulky databáze back-end.





### *Úkoly k procvičování:*

- ❶ Exportujeme do formátu XML tabulku **Employees** z databáze Northwind. Při exportu volíme maximum možností. Vzniklé soubory s příponami XML, XSD, XSL a HTML si prohlédneme v Internet Exploreru.
- ❷ Importujeme do Accessu soubor XML vytvořený v předchozím cvičení. Importujeme do Accessu soubor XSD vytvořený v předchozím cvičení. Soubor XSD odstraníme a pokusíme se opět o import souboru XML.



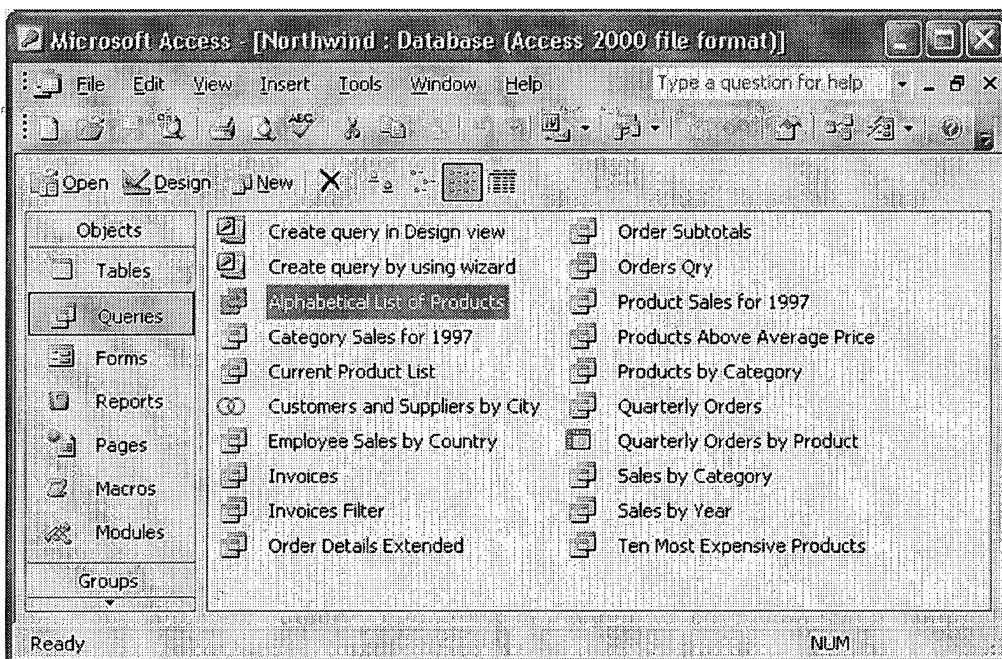
- Objekty Accessu exportujeme příkazem *File/Export*.
- Import dat provedeme příkazem *File/Get External Data/Import* nebo *Link Tables*.
- Access XP umí exportovat a importovat data ve formátu XML.

# Úvod do tvorby výběrových dotazů

## *Přehled témat*

- práce s dotazem v návrhovém zobrazení
- výrazy v dotazech
- důležité funkce Accessu
- vypočítávaná pole v dotazech

# Dotazy



## Dotazy (Queries)

Umožňují získávat specifické informace z tabulek a též tabulky modifikovat.

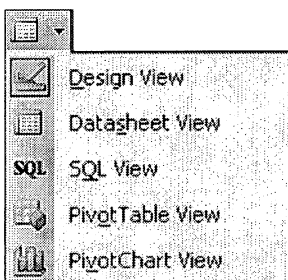
### Co je to dotaz

Objekt, obsahující příkazy (ne data) specifikující práci (výběr, modifikace) s daty umístěnými v tabulkách. Dotaz pracuje vždy s aktuálními uloženými daty a jím získané výsledky mohou být použity jako základ pro tvorbu dalších dotazů, formulářů a sestav či datových stránek.

### Základní druhy dotazů

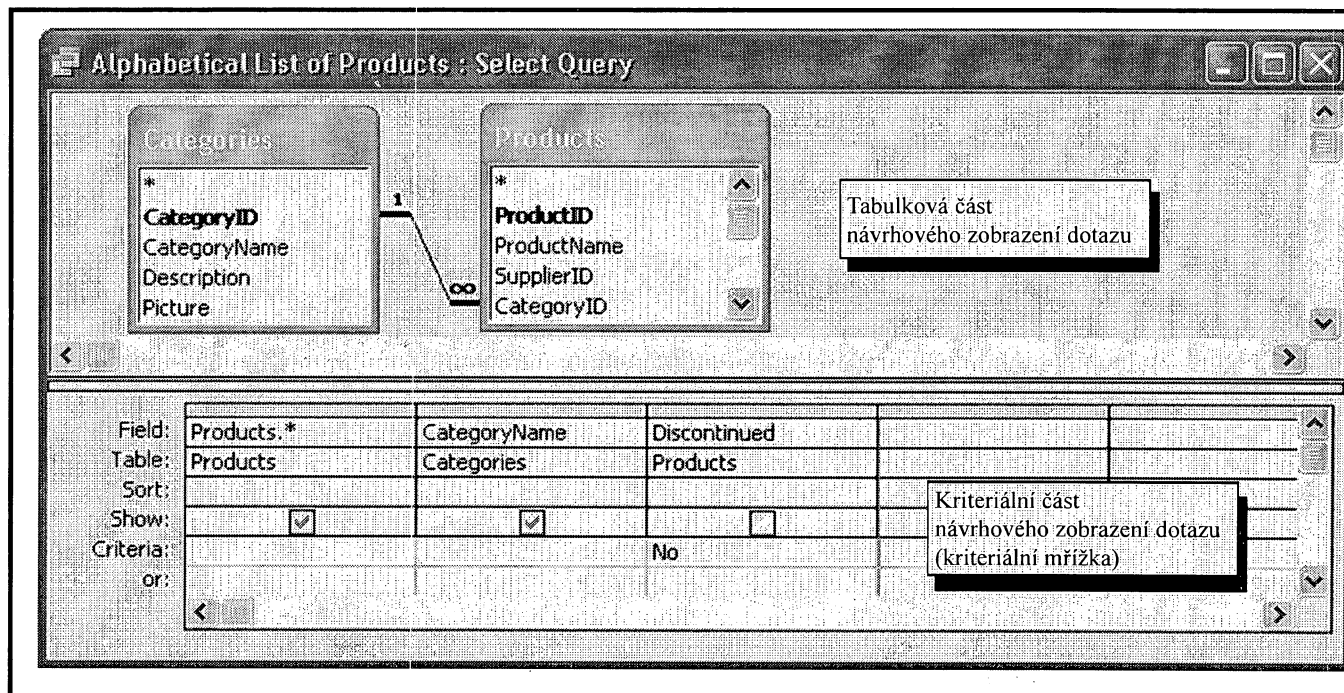
- Pasivní (výběrové (Select Queries)) vybírají a zobrazují data
- Aktivní (akční (Action Queries)) dělají změny v tabulkách (aktualizují hodnoty polí, mažou záznamy).

### Režimy zobrazení dotazu



- **Návrh (Design View).** V tomto zobrazení dotaz vytváříme a upravujeme.
- **Datový list (Datasheet View).** Slouží k prohlížení výsledku dotazu (dat vybraných dotazem).
- **SQL (SQL View).** Kóduje příkaz pro práci s daty jazykem SQL (Structured Query Language).

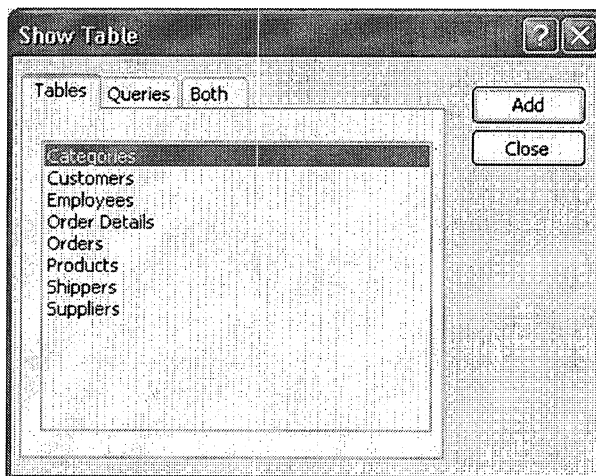
## Práce s dotazem v návrhovém zobrazení



## Práce s dotazem v návrhovém zobrazení



Vytvoříme-li nový dotaz v návrhovém zobrazení (například klepnutím na tlačítko **New** na kartě **Queries** a výběrem volby **Design View**, ihned se objeví v popředí dialogové okno **Show Table**, kterým vkládáme do tabulkové části dotazu datové zdroje (tabulky, dotazy).



Po zavření tohoto okna lze však kdykoli tlačítkem **Show Table** (příkaz **Query/Show Table**) další tabulky či dotazy do tabulkové části návrhového zobrazení dotazu vložit. Existující relace mezi tabulkami se přitom zobrazují spojnicemi, pro určitý dotaz můžeme relaci vytvořit dodatečně přetažením polí mezi tabulkami (případně změnit typ existující relace).

Do kriteriální mřížky umístíme pole,

- která chceme zobrazit ve výsledcích dotazu,
- či na která klademe kritéria.



## Zobrazování polí ve výsledku dotazu

Nechceme-li pole umístěné do kritériální mřížky zobrazit ve výsledku dotazu, deaktivujeme jeho vlastnost **Show** (viz řádek **Show**).

## Řazení

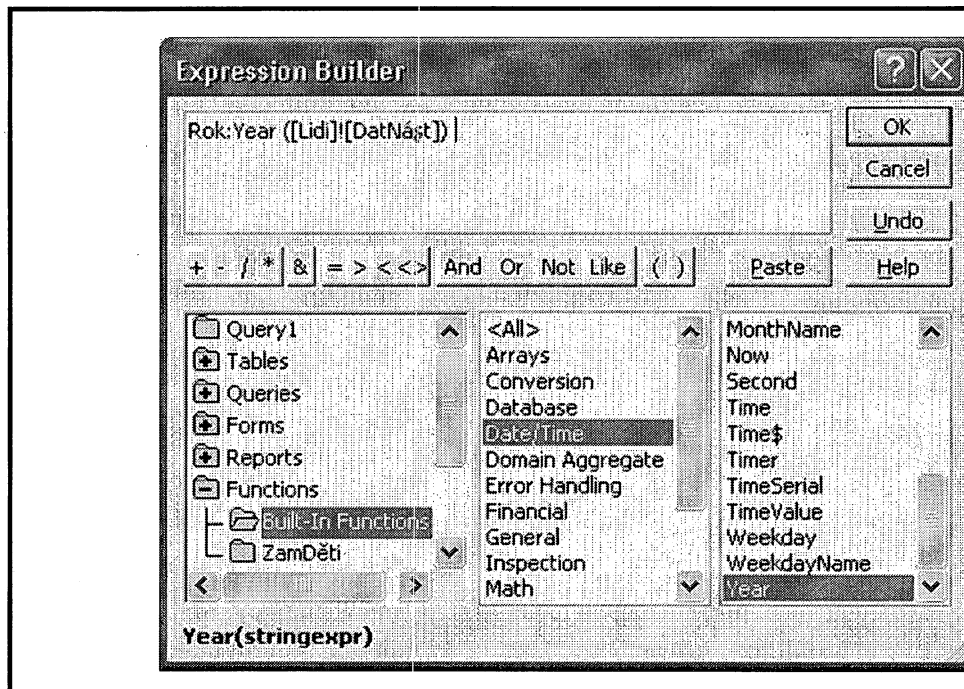
V řádku **Sort** volíme **Ascending** či **Descending**, chceme-li podle tohoto pole řadit záznamy výsledku. Aktivujeme-li řazení pro více polí, dělá se vícenásobně, tj. výsledky jsou primárně seřazeny podle prvního pole zleva, kde je řazení aktivováno. Pro stejné hodnoty v tomto poli jsou záznamy seřazeny podle hodnot v druhém poli zleva, kde je řazení aktivováno, atd. Lze řadit podle datových typů Text, Number, Date/Time, Yes/No.

## Zadávání kritérií dotazu

Kritéria pro určité pole zadáváme ve sloupci pro toto pole do řádků **Criteria** a **or** kritériální mřížky.

- Podmínky zapsané na stejném řádku jsou spojeny pomocí **AND**.
- Podmínky zapsané na různých řádcích jsou spojeny pomocí **OR**.

## Výrazy Accessu používané v dotazech



### Výrazy Accessu používané v dotazu

V dotazech často používáme výrazy. Výraz je platnou kombinací operátorů a operandů.



### Často používané operátory

Aritmetické : + - \* /\ mod ^

Relační :

| Operátor            | Význam                                      | Příklad                 |
|---------------------|---------------------------------------------|-------------------------|
| < <= >= > <> =      | <i>nerovnice</i>                            | >=#01/01/1958#          |
| between ... and ... | <i>specifikace intervalu včetně mezi</i>    | between 50000 and 75000 |
| Is                  | <i>test na hodnotu NULL</i>                 | Is Null, Is Not Null    |
| Like                | <i>test, zda pole vyhovuje zadané masce</i> | Like "*1988"            |



## Operátor Like v kriteriální mřížce dotazu

Operátor **Like** používáme v kriteriální mřížce dotazu takovým způsobem, že vpravo od něj zapíšeme tvar šablony (masky), jemuž musí vyhovovat hodnoty pole, na které tento operátor působí. Ve zmíněné masce používáme, kromě běžných znaků, též znaky, které mají speciální význam. Uvádíme je v následující tabulce. Připomeňme, že nerozlišujeme velká a malá písmena.

| Znak                 | Význam znaku v masce                                               | Příklad                                                                      |
|----------------------|--------------------------------------------------------------------|------------------------------------------------------------------------------|
| ?                    | <i>zastupuje jakýkoliv jeden znak</i>                              | Like "J?" vyhovují dva znaky, první musí být J                               |
| *                    | <i>zastupuje Null a nebo libovolnou skupinu znaků</i>              | Like "J*" vyhoví jakýkoliv text začínající na J                              |
| #                    | <i>zastupuje jakoukoliv číslici</i>                                | Like "#####/#####" vyhoví rodné číslo se čtyřmi znaky za lomítkem            |
| [ <i>charlist</i> ]  | <i>zastupuje jakýkoliv jeden znak uvedený v seznamu charlist</i>   | Like "###[56]###/#####" vyhoví rodné číslo ženy se čtyřmi znaky za lomítkem  |
| [! <i>charlist</i> ] | <i>zastupuje jakýkoliv jeden znak neuvedený v seznamu charlist</i> | Like "###[!56]###/#####" vyhoví rodné číslo muže se čtyřmi znaky za lomítkem |

Chceme-li testovat na výskyt znaků patřícího mezi tyto speciální znaky (např. ?, #, \*) nutno dát tyto znaky do hranatých závorek [].

Operátor Like možno použít nejen na testování textových polí, ale s výhodou i u polí typu Number a Date/Time. Vložíme-li např. na pole **OrderDate** z tabulky **Orders** databáze **Northwind** kritérium Like "\*"8", pak vypíšeme všechny objednávky z roku končícího na číslici 8.

## Často používané logické operátory

Rozlišujeme binární logické operátory **and** a **or** a unární operátor **not**. Binární operátory působí na logické výrazy umístěné vlevo a vpravo od operátoru (A and B), unární operátor působí na logický výraz umístěný vpravo od něj (not A).

| Operátor | Význam                                                                                        | Příklad                           |
|----------|-----------------------------------------------------------------------------------------------|-----------------------------------|
| and      | <i>Logický součin. A and B. Pro splnění podmínky musí platit zároveň A i B.</i>               | >=#01/01/1958# and <=#06/01/1961# |
| or       | <i>Logický součet. A or B. Pro splnění podmínky stačí, aby platila jedna z podmínek A, B.</i> | "Belgium" or "France" or "UK"     |
| not      | <i>Negace. not A. Je splněno, neplatí-li A.</i>                                               | not "France"                      |

## Operátor pro spojování řetězců

Konkatenační operátor (ampersand): &

poznámka



Funkce In(). Místo logického operátoru or lze použít funkci In(), např. In("Belgium", "France", "UK")



## Precedence vyhodnocování určuje pořadí vyhodnocování operátorů



### Pravidla

- Operace se vyhodnocují v pořadí klesající precedence
- V případě stejné precedence probíhá vyhodnocování zleva doprava
- Precedenci lze zvýšit závorkami - operace v závorkách se vyhodnocují nejdříve

### Precedence aritmetických operátorů a &

- Umocňování ^
- Unární mínus
- Násobení a dělení
- Celočíselné dělení \
- Zbytek celočíselného dělení mod
- Sčítání a odčítání
- Spojování řetězců &

Příklady  $-2^2 = -4$

$(-2)^2 = 4$  (Zvýšení precedence unárního mínus závorkami)

$A/B * C$  (A vydělíme B a výsledek násobíme C.)

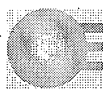
/ a \* mají stejnou precedenci, proto je vyhodnocujeme zleva doprava)

$A/(B * C)$  (Vypočítá se součin B a C a jeho hodnotou se dělí A.)

Precedence relačních operátorů je stejná.

Precedence logických operátorů Not, And, Or klesá v pořadí

- Not
- And
- Or



## Časté operandy výrazů používané v dotazech:

|                                 |                                                                                                                         |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| <b>Pole tabulky</b><br>znamená  | Specifikace <b>Lidi.Plat</b> resp. <b>[Lidi].[Plat]</b> či <b>[Lidi]![Plat]</b><br>tabulka <b>Lidi</b> pole <b>Plat</b> |
| <b>Volání funkce</b><br>znamená | <b>Year([Datum])</b><br>je funkce <b>Year()</b> použitá na pole <b>Datum</b>                                            |
| <b>Konstanty</b>                | <b>#01/01/1987#</b>                                                                                                     |
| <b>Parametry dotazu</b>         | (viz dále)                                                                                                              |

## Funkce

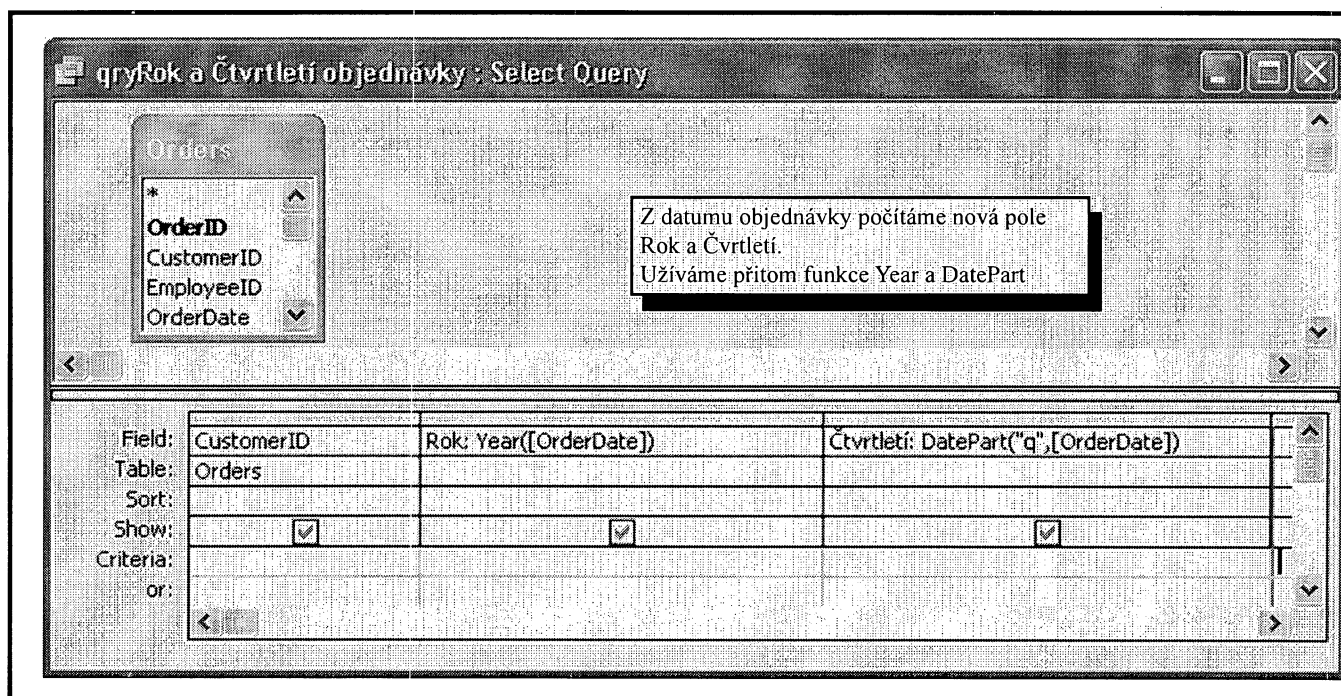
Access obsahuje celou řadu vestavěných funkcí, které se dělí do několika skupin. Jejich seznam včetně kategorizace je přehledně zobrazován tvůrcem výrazu. Přesný popis jednotlivých funkcí máme uveden v nápovědě. Uživatelsky důležité skupiny funkcí jsou podrobněji popsány v následující tabulce:

| Název skupiny funkcí | Popis a příklady                                                                                                                                                                                                                            |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Date/Time            | Pracují s daty typu Date/Time. Například funkce DatePart umí z datumu extrahovat jeho části (rok, čtvrtletí apod).                                                                                                                          |
| Math                 | Implementují základní matematické funkce - goniometrické, logaritmus, exponenciála, zaokrouhlovací funkce apod.                                                                                                                             |
| Text                 | Umožňují práci s daty typu Text. Slouží k získávání částí textových údajů (Left, Right, Mid), hledání určitého textu v jiném (InStr), porovnávání textů (StrComp) apod.                                                                     |
| SQL Aggregate        | Používáme je při tvorbě souhrnných dotazů (viz dále). Patří sem běžné funkce jako Sum, Count, Min, Max, Avg.                                                                                                                                |
| Domain Aggregate     | Jsou obdobou databázových funkcí Excelu. Slouží k provádění podmíněných výpočtů v nějaké doméně (tabulka, dotaz). Např. funkcí DAvg lze spočítat podmíněný průměr, DSum podmíněný součet. Doménu a podmínku zadáváme funkci jako argumenty. |
| Inspection           | Slouží k testování typu argumentu např. funkce IsNumeric vrací hodnotu True, je-li její argument číslo, jinak vrací False. Funkce IsNull obdobně testuje, zda má její argument hodnotu Null.                                                |

| Textové funkce                                                |                                                                                                                                                                                 |                                                                                                           |
|---------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| InStr([start,] string1, string2[, compare])                   | Hledá řetězec string2 v řetězci string1 a vrací pozici jeho prvního výskytu. Nenajde-li, vrací 0.                                                                               | InStr("790214/169","/")=7<br>InStr("790214/0169", "5")=0                                                  |
| Left(string, length)                                          | Vrací z řetězce string počet znaků zadaných argumentem length zleva.                                                                                                            | Left("790214/0169",6)="790214"                                                                            |
| Right(string, length)                                         | Vrací z řetězce string počet znaků zadaných argumentem length.                                                                                                                  | Right("790214/0169",4)="0169"                                                                             |
| Mid(string, start[, length])                                  | Z řetězce string od pozice start vrací počet znaků zadaný argumentem length.                                                                                                    | Mid("790214/0169",3,2)="02"                                                                               |
| Len(string)                                                   | Vrací délku zadaného textového řetězce ve znacích.                                                                                                                              | Len("790214/0169")=13                                                                                     |
| LCase(string)                                                 | Převádí všechny znaky v řetězci na malá písmena.                                                                                                                                | LCase("RobertPe")="robertpe"                                                                              |
| UCase(string)                                                 | Převádí všechny znaky v řetězci na velká písmena.                                                                                                                               | UCase("www")="WWW"                                                                                        |
| Trim(string)                                                  | V textovém řetězi zruší všechny mezery na začátku a na konci.                                                                                                                   | Trim(" robertpe ")="robertpe"                                                                             |
| LTrim(string)                                                 | V textovém řetězci zruší všechny počáteční mezery.                                                                                                                              | LTrim(" robertpe ")="robertpe "                                                                           |
| RTrim(string)                                                 | V textovém řetězci zruší všechny koncové mezery.                                                                                                                                | RTrim(" robertpe ")=" robertpe"                                                                           |
| Funkce pro práci s kalendárními daty a časem                  |                                                                                                                                                                                 |                                                                                                           |
| Date()                                                        | Vrací aktuální datum dle nastavení systému.                                                                                                                                     |                                                                                                           |
| DatePart(interval, date[,firstdayofweek[, first-weekofyear]]) | Vrací určitou část ze zadaného data.                                                                                                                                            | DatePart("q",#5.1.2004#)=1<br>DatePart("ww",Date()) vrací pořadové číslo týdne v roce pro aktuální datum. |
| Year(datum), Month(datum), Day(datum)                         | Ze zadaného data vrací odpovídající složku.                                                                                                                                     |                                                                                                           |
| Hour(čas), Minute(čas), Second(čas)                           | Ze zadaného času vrací odpovídající složku.                                                                                                                                     |                                                                                                           |
| DateSerial(year, month, day)                                  | Ze zadaných složek sestaví datum.                                                                                                                                               | DateSerial(2004,10,2)=#2.10.2004#                                                                         |
| Weekday(date, [firstdayofweek])                               | Ze zadaného data vrací pořadové číslo dne v rámci týdne. Volitelným parametrem firstdayofweek lze nastavit, který den má index 1 (pro týden od pondělí volíme jeho hodnotu =2). | Weekday(#04/28/2004#,2)=3<br>(28.4.2004 byla středa tzn. 3. den v týdnu od pondělka)                      |
| WeekdayName(weekday[, abbreviate, firstday-ofweek])           | Zadané pořadové číslo (v intervalu 1-7) převede na název dne.                                                                                                                   | WeekdayName(1)="Sunday"                                                                                   |
| Matematické funkce                                            |                                                                                                                                                                                 |                                                                                                           |
| Abs(number)                                                   | Vrací absolutní hodnotu zadaného čísla. Číselný datový typ zůstává stejný.                                                                                                      | Abs(-1500)=1500                                                                                           |
| Sgn(number)                                                   | Vrací 1 pro kladná čísla, -1 pro záporná čísla, 0 pro 0.                                                                                                                        | Sgn(-100)=-1; Sgn(100)=1;<br>Sgn(0)=0                                                                     |
| Int(number)                                                   | Vrací celou část čísla.                                                                                                                                                         | Int(100,56)=100                                                                                           |
| Round(expression [, numdecimalplaces])                        | Zaokrouhluje daný výraz na uvedený počet desetinných míst. Není-li počet des. míst zadán, funkce vrací celou část čísla.                                                        | Round(100.1234, 2)=100,12                                                                                 |

| Ověřovací funkce      |                                                                                 |                                                            |
|-----------------------|---------------------------------------------------------------------------------|------------------------------------------------------------|
| IsNumeric(expression) | Jestliže lze výraz vyhodnotit jako číslo, vrátí funkce True, jinak vrátí False. | IsNumeric(123)=True<br>IsNumeric("RobertPe")=False         |
| IsDate(expression)    | Jestliže lze výraz vyhodnotit jako datum, vrátí funkce True, jinak vrátí False. | IsDate(Date())=True<br>IsDate("13. února roku 2004")=False |
| IsNull(expression)    | Jestliže výraz obsahuje hodnotu Null, vrátí funkce True, jinak vrátí False.     |                                                            |
| Převodní funkce       |                                                                                 |                                                            |
| CInt(expression)      | Převede zadaný výraz na celé číslo.                                             |                                                            |
| CDate(expression)     | Převede zadaný výraz na datum.                                                  | CDate("January 04, 2006")=#01/04/2006#                     |
| CStr(expression)      | Převede zadaný výraz na text.                                                   |                                                            |
| CCur(expression)      | Převede zadaný výraz na datový typ Currency.                                    |                                                            |

## Vypočítávaná pole v dotazoch



## Tvorba nových polí v dotazoch

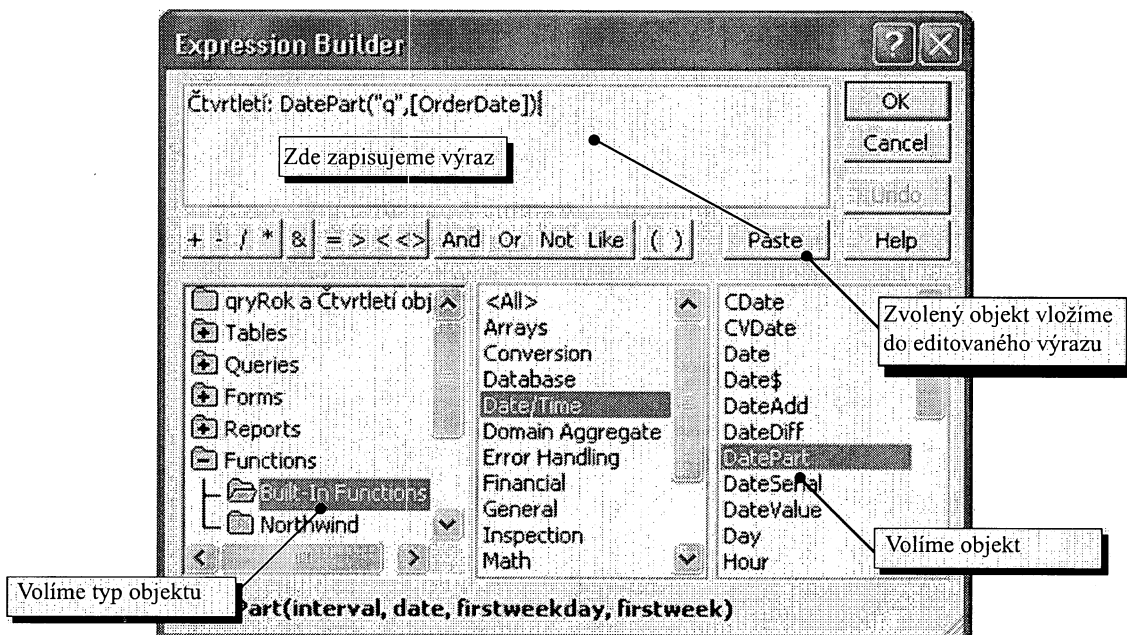
### Charakteristika

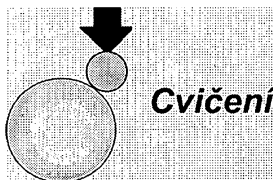
Môžeme počítať nové pole, ktoré sú výsledkom dotazu.

### Definície pole s počítanou hodnotou

Název pole: Výraz

Je-li ve sloupci s počítanou hodnotou složitější výraz, možno s výhodou využiť tvůrce výrazů. Jsou-li v názvech polí používaných ve výrazu povolené nealfanumerické znaky (např. mezery), nutno dát celý název do hranatých závorek.





## Úkoly k procvičování:

1

V databázi Northwind sestavme výběrový dotaz zobrazující názvy výrobků dodávaných z USA, Japonska či Austrálie, jejichž jednotková cena nepřevyšuje 20\$:

1. V okně databáze volíme **Queries** a klepneme na tlačítko **New**. Dotaz děláme v návrhovém zobrazení.
2. Do tabulkové části okna umístíme tabulky **Suppliers** a **Products**.
3. Do kritériální mřížky dáme pole **Country**, **ProductName**, **UnitPrice**.
4. Na pole **Country** klademe podmínku "USA" Or "Japan" Or "Australia".
5. Na pole **UnitPrice** klademe kritérium  $\leq 20$ .

Výsledky dotazu (přepněte do zobrazení datového listu nebo klepněte na tlačítko **Run**):

| Country   | Product Name                    | Unit Price |
|-----------|---------------------------------|------------|
| Japan     | Konbu                           | \$6.00     |
| Japan     | Genen Shouyu                    | \$15.50    |
| Australia | Pavlova                         | \$17.45    |
| USA       | Sasquatch Ale                   | \$14.00    |
| USA       | Steeleye Stout                  | \$18.00    |
| USA       | Boston Crab Meat                | \$18.40    |
| USA       | Jack's New England Clam Chowder | \$9.65     |
| Australia | Filo Mix                        | \$7.00     |
| USA       | Louisiana Hot Spiced Okra       | \$17.00    |
| USA       | Laughing Lumberjack Lager       | \$14.00    |
| Australia | Outback Lager                   | \$15.00    |
| Japan     | Longlife Tofu                   | \$10.00    |

2

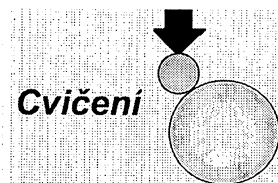
V databázi Northwind sestavme dotaz zobrazující výrobky a celkové ceny výrobků objednaných z Německa, Brazílie, nebo Francie. Celkové ceny počítáme jako součin jednotkové ceny a množství. Dotaz uložíme jako **qryCelkCeny**

Postup:

1. Do tabulkové část návrhového zobrazení dotazu dáme tabulky **Customers**, **Orders**, **Order Details**, **Products**.
2. Do kritériální mřížky vložíme pole **Country** z tabulky **Customers**, pole **ProductName** z tabulky **Products** a pole **OrderDate** z tabulky **Orders**.
3. Vytvoříme vypočítávané pole

**CelkCeny:** [Order Details].[UnitPrice]\*[Order Details].[Quantity]

4. Na pole **Country** klademe kritérium "Germany" Or "Brazil" Or "France"
5. Na pole **OrderDate** klademe kritérium **Like** "\*1996"



## Úkoly k procvičování:

Výsledek:

| qryCelkCeny : Select Query |         |                                 |             |            |
|----------------------------|---------|---------------------------------|-------------|------------|
|                            | Country | Product Name                    | Order Date  | CelkCeny   |
| ▶                          | France  | Alice Mutton                    | 25-Jul-1996 | \$936.00   |
|                            | France  | Outback Lager                   | 25-Jul-1996 | \$240.00   |
|                            | France  | Chartreuse verte                | 04-Sep-1996 | \$864.00   |
|                            | France  | Mozzarella di Giovanni          | 04-Sep-1996 | \$556.00   |
|                            | France  | Rössle Sauerkraut               | 22-Nov-1996 | \$1,092.00 |
|                            | France  | Thüringer Rostbratwurst         | 22-Nov-1996 | \$3,465.00 |
|                            | France  | Côte de Blaye                   | 22-Nov-1996 | \$2,108.00 |
|                            | France  | Maxilaku                        | 22-Nov-1996 | \$560.00   |
|                            | France  | Tourtière                       | 22-Nov-1996 | \$165.20   |
|                            | France  | Tourtière                       | 16-Oct-1996 | \$88.50    |
|                            | France  | Carnarvon Tigers                | 29-Oct-1996 | \$1,000.00 |
|                            | France  | Jack's New England Clam Chowder | 29-Oct-1996 | \$92.40    |
|                            | France  | Ipoh Coffee                     | 29-Oct-1996 | \$1,472.00 |
|                            | France  | NuNuCa Nuß-Nougat-Creme         | 25-Nov-1996 | \$560.00   |
| Record: 1 of 134           |         |                                 |             |            |

- 3 V nápovědě zjistěte, k čemu lze použít a jak se chovají funkce **DateAdd** a **DateDiff**.



- Dotazy umožňují získávat informace z tabulek a tabulky modifikovat.
- V návrhovém zobrazení dotaz vytváříme, v zobrazení datového listu si prohlížíme jeho výsledky.
- Složitější výrazy Accessu lze dělat pomocí tvůrce výrazů.
- Dotaz umožňuje vytvářet nová vypočítávaná pole z polí existujících.





# Základní typy výběrových dotazů

## *Přehled témat*

- parametrické dotazy
- dotazy vracející několik největších a nejmenších hodnot
- vyčlenění duplicit z výsledku dotazu
- souhrnné dotazy
- vlastnosti spojení tabulek

# Parametrické dotazy

qryVýrobky firmy objednané v určitém roce : Select Query

Customers: CustomerID, CompanyName, ContactName, ContactTitle

Orders: OrderID, CustomerID, EmployeeID, OrderDate

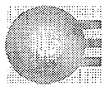
Order Details: OrderID, ProductID, UnitPrice, Quantity

Products: ProductID, ProductName, SupplierID, CategoryID

| Field:    | CompanyName                         | OrderDate                           | ProductName                         |                          |                          |
|-----------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|
| Table:    | Customers                           | Orders                              | Products                            |                          |                          |
| Sort:     |                                     |                                     |                                     |                          |                          |
| Show:     | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Criteria: | Like [Zadejte firmu] & '*'          | Like '*' & [Rok]                    |                                     |                          |                          |
| or:       |                                     |                                     |                                     |                          |                          |

Zadáme tolik počátečních písmen firmy, kolik známe

Zadáme poslední číslice roku



## Parametrické dotazy

### Charakteristika

Po spuštění parametrického dotazu zadáváme konkrétní hodnoty parametrů, což umožňuje spouštět dotaz pokaždé s jinými vstupními daty, aniž měníme jeho návrh.

### Postup tvorby

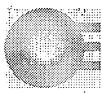
Do výrazu v kritériální mřížce dotazu zapíšeme parametr, což je text v hranatých závorkách. Tento text se nesmí shodovat s názvem pole žádné tabulky, která je součástí dotazu.

### Spuštění dotazu

- Po spuštění dotazu se Access ptá na hodnoty jednotlivých parametrů.
- Každému běhu parametrického dotazu lze zadat jinou hodnotu parametru, aniž upravujeme kritériální mřížku dotazu.

[illegible]

Pořadí parametrů v tomto okně také určuje, v jakém pořadí bude Access požadovat zadání jejich hodnot. Jinak je pořadí dáno návrhovou mřížkou zleva doprava.



## Reakce Accessu na nedodržení typu parametru při jeho zadávání

gryVýrobky firmy objednané v určitém roce : Select Query

| Field:    | CompanyName                         | OrderDate                           | ProductName                         |                          |                          |
|-----------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|
| Table:    | Customers                           | Orders                              | Products                            |                          |                          |
| Sort:     |                                     |                                     |                                     |                          |                          |
| Show:     | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Criteria: | Like [Zadejte firmu] & "*"          | Like "*" & [Rok]                    |                                     |                          |                          |
| or:       |                                     |                                     |                                     |                          |                          |

Enter Parameter Value

Rok

p

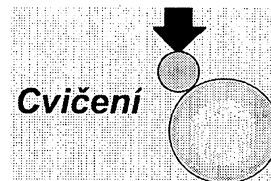
OK Cancel

Microsoft Office Access

**The value you entered isn't valid for this field.**

For example, you may have entered text in a numeric field or a number that is larger than the FieldSize setting permits.

OK



## Úkoly k procvičování:

- 1 Zkopírujme dotaz **qryCelkCeny** jako **qryCelkCenyParam**. Zadávejme do něj potom jako parametry počáteční písmeno země a rok data objednávky.

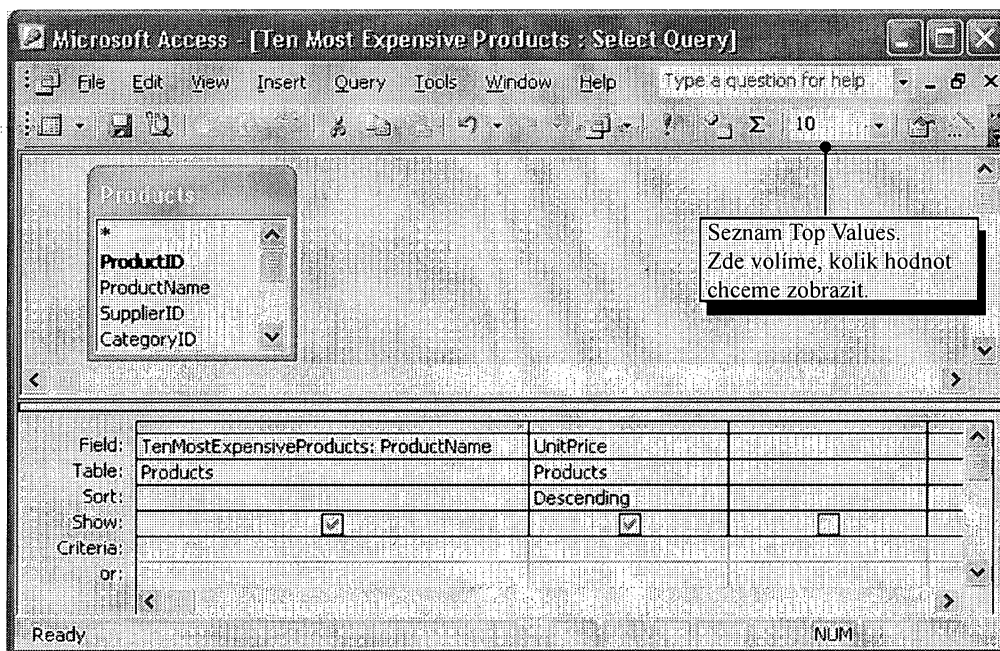
Dále vypíšeme jen záznamy, pro které je **CelkCena** menší, než zadaná hodnota (parametr). Dotaz má tedy celkem tři parametry.

U číselných parametrů zadejme kontrolu datových typů.

Postup:

1. Na pole **Country** klademe kritérium **Like [Název země] & "\*"**
2. Pro pole **OrderDate** máme podmínku **Like "\*" & [Rok Objednávky]**
3. Hodnoty pole **CelkCeny** jsou omezeny vztahem **<[Horní mez pro Cenu]**.
4. Zobrazíme dialogové okno příkazu **Query/Parameters**. Zde specifikujeme typy parametrů. Na parametr **[Rok Objednávky]** klademe podmínky na celočíselnost (Integer), parametr **[Horní mez pro Cenu]** je měna (Currency).

## Dotaz vracející několik největších či nejmenších hodnot



## Dotaz vracející několik největších či nejmenších hodnot

### Charakteristika

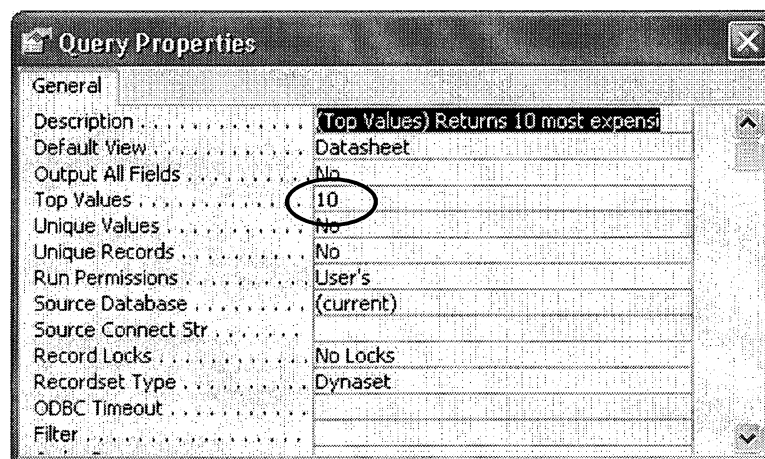
Dotazem získáme záznamy obsahující několik největších či nejmenších hodnot v určitých polích.

### Postup tvorby:

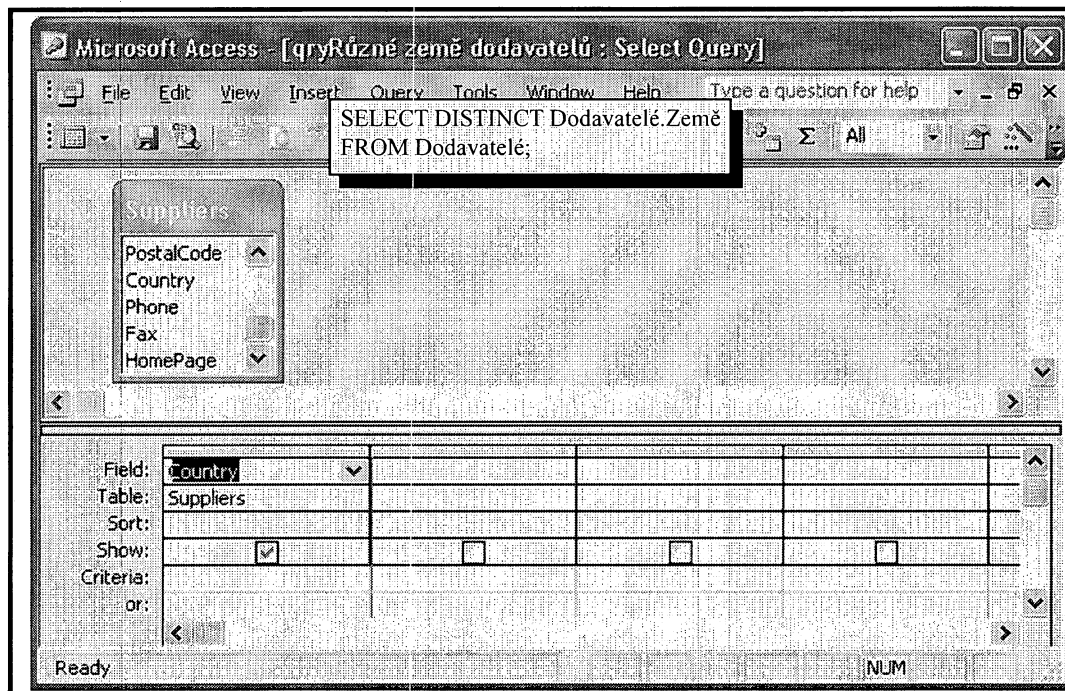
Podle sloupce, pro který chceme vrátit největší či nejmenší hodnoty seřadíme

- v případě největších hodnot sestupně
- v případě nejmenších hodnot vzestupně.

V seznamu **Top Values** volíme, kolik hodnot (procent) chceme zobrazit (lze sem i vpisovat). Zadaný počet hodnot se uchovává ve vlastnostech dotazu.



## Vyloučení duplicit z výsledku dotazu



## Vyloučení duplicit z výsledku dotazu

### Postup tvorby:

Vlastnost **Unique Values** dotazu nastavíme na **Yes**. V příkaze SELECT, jímž se tento dotaz kóduje, je toto nastavení zapsáno slovem DISTINCT.

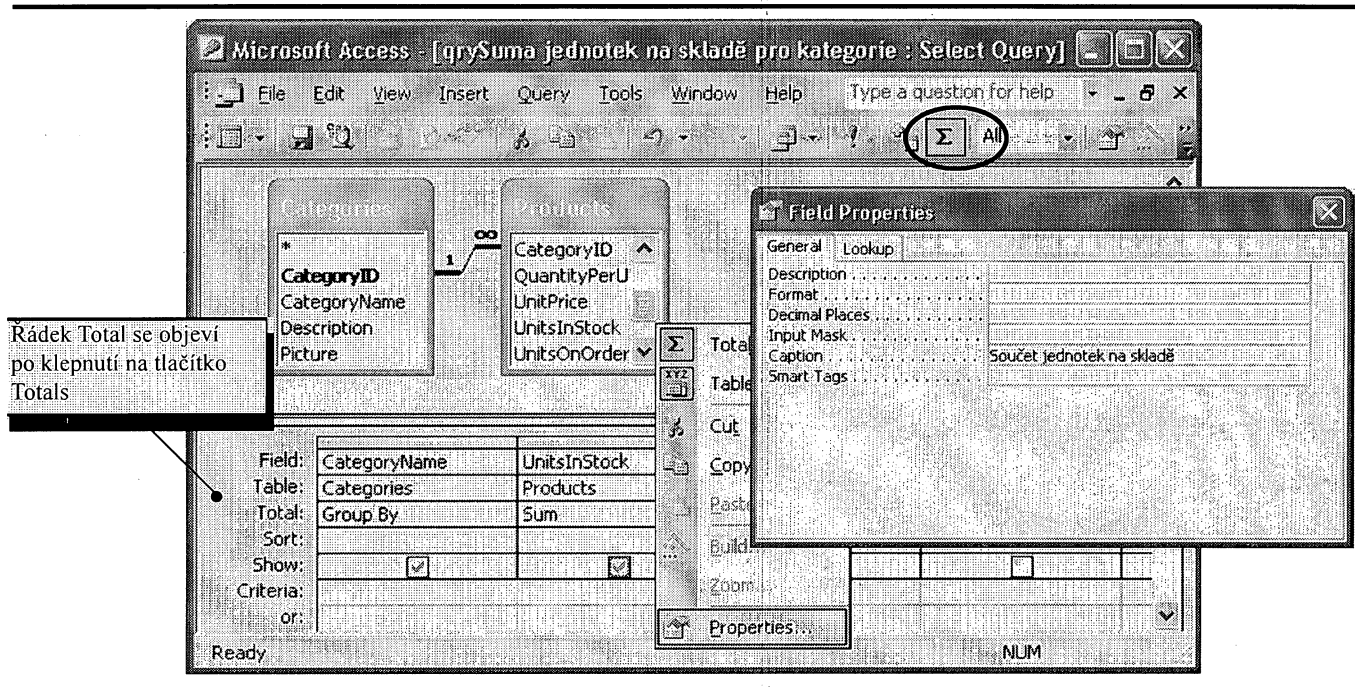


*Vlastnosti dotazu získáme z místní nabídky tabulkové části návrhového zobrazení dotazu.*





# Souhrnné dotazy



## Souhrnné dotazy

### Charakteristika

Záznamy se stejnými hodnotami v určitém poli seskupíme a pro tyto položky počítáme souhrnné hodnoty z jiných polí.

### Postup tvorby



- Klepneme na tlačítko **Totals**. V kritériální mřížce se objeví řádek **Total**.
- Volíme **Group By** pro pole, podle nichž seskupujeme záznamy.
- Volíme **souhrnnou funkci** pro pole, v nichž souhrny počítáme.
- V okně vlastností sumarizovaného pole můžeme změnit jeho **Caption**.

| Category Name  | Součet jednotek na skladě |
|----------------|---------------------------|
| Beverages      | 559                       |
| Condiments     | 507                       |
| Confections    | 386                       |
| Dairy Products | 393                       |
| Grains/Cereals | 308                       |
| Meat/Poultry   | 165                       |
| Produce        | 100                       |
| Seafood        | 701                       |

Record: 1 of 8



## Definice kritérií pro souhrnné dotazy

The screenshot displays the Microsoft Access interface. The main window shows the 'qrySumy objednaného množství pro firmy : Select Query' in Design View. It includes three tables: Customers, Orders, and Order Details. The Customers table has fields: CustomerID, CompanyName, ContactName, and ContactTitle. The Orders table has fields: OrderID, CustomerID, EmployeeID, and OrderDate. The Order Details table has fields: OrderID, ProductID, UnitPrice, and Quantity. The query is set to group by CompanyName and OrderDate, and sum the Quantity. The preview window shows the results of the query, which is a summary of quantities ordered by company and year.

| Company Name          | Rok  | Součet množství |
|-----------------------|------|-----------------|
| Santé Gourmet         | 1996 | 48              |
| Save-a-lot Markets    | 1996 | 567             |
| Seven Seas Imports    | 1996 | 321             |
| Simons bistro         | 1996 | 17              |
| Split Rail Beer & Ale | 1996 | 174             |
| Suprêmes délices      | 1996 | 185             |

## Definice kritérií pro souhrnné dotazy

### Všeobecné kritérium (Where)

#### Charakteristika

- Všeobecné kritérium stanoví, které záznamy se použijí pro tvorbu souhrnů.
- Vlastní seskupování a další výpočet proběhne až po výběru (vyfiltrování) vhodných záznamů.

#### Postup tvorby

- Záznamy pro seskupování vybíráme podle určitých kritérií.
- V poli kritériální mřížky, na které klademe kritérium nastavíme na řádku **Total** volbu **Where**.
- Na řádcích **Criteria** a **or** tohoto pole uvedeme podmínky.
- Dále postupujeme jako u běžného souhrnného dotazu.

## Kritérium pro skupiny

Chceme jen ty skupiny, kde je součet  $\geq 100$

| Field:    | CompanyName                         | Rok: Year([OrderDate])              | Součet množství: Quantity           | CompanyName                      | OrderDate                          |
|-----------|-------------------------------------|-------------------------------------|-------------------------------------|----------------------------------|------------------------------------|
| Table:    | Customers                           |                                     | Order Details                       | Customers                        | Orders                             |
| Total:    | Group By                            | Group By                            | Sum                                 | Where                            | Where                              |
| Sort:     |                                     |                                     |                                     |                                  |                                    |
| Show:     | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/>         | <input type="checkbox"/>           |
| Criteria: |                                     |                                     | $\geq 100$                          | Like [Začátek názvu firmy] & "*" | Like "*" & [Poslední číslice roku] |
| or:       |                                     |                                     |                                     |                                  |                                    |

| Company Name          | Rok  | Součet množství |
|-----------------------|------|-----------------|
| Save-a-lot Markets    | 1996 | 567             |
| Seven Seas Imports    | 1996 | 321             |
| Split Rail Beer & Ale | 1996 | 174             |
| Suprêmes délices      | 1996 | 185             |

Record: 1 of 4

### Kritérium pro skupiny

#### Charakteristika

Ze skupin vytvořených při souhrnném dotazu nechceme vybrat všechny, ale vybíráme na základě kritéria pouze některé.

#### Postup tvorby

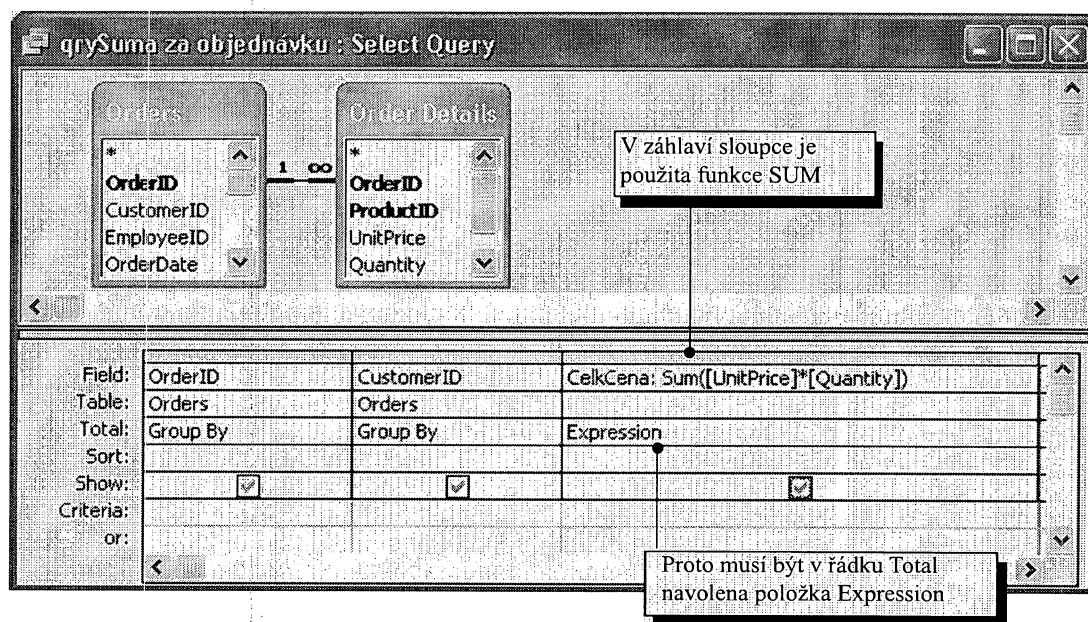
Do řádků **Criteria** a **or** polí, na která kritéria klademe, píšeme podmínky.

### Volba Expression v řádku Total souhrnného dotazu

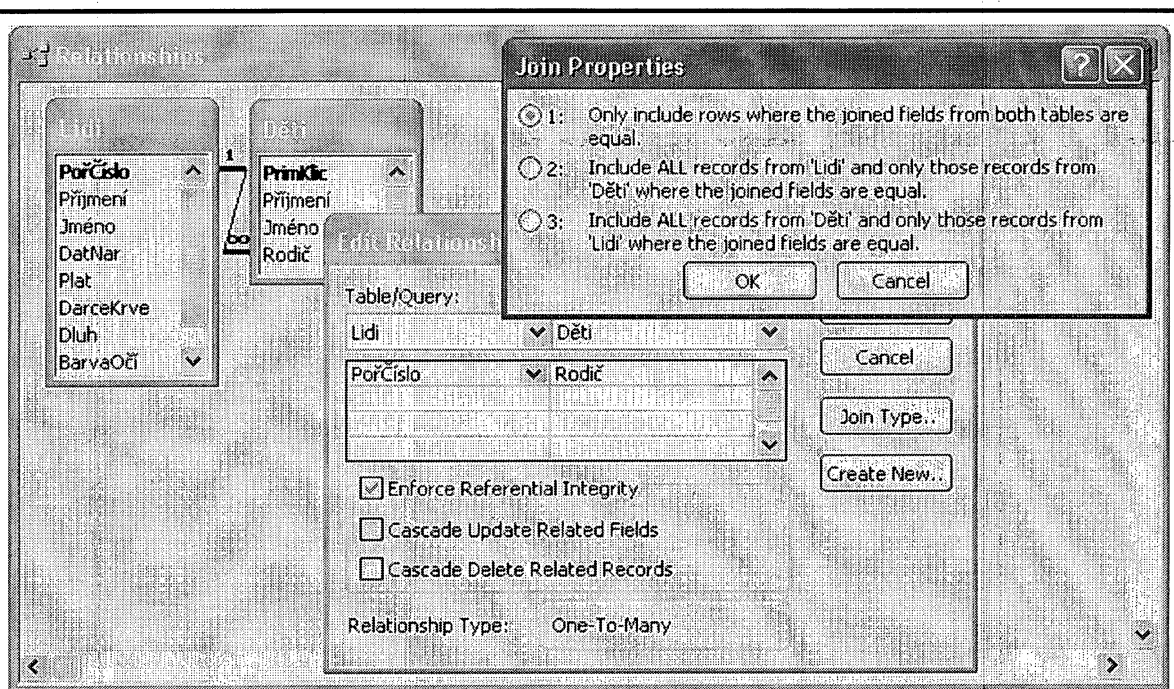
Obsahuje-li sloupec s počítanou hodnotou ve výrazu záhlaví souhrnnou funkci a jde-li o souhrnný dotaz, musí být v řádku **Total** navolena položka **Expression**.

## Úkoly k procvičování:

- 1 Sestavme souhrnný dotaz, který zobrazuje pro každou objednávku zákazníka celkovou cenu. Dotaz nazveme **qrySumaZaObjednávku**.



# Vlastnosti spojení tabulek



## Vlastnosti spojení tabulek

### Základní typy spojení tabulek v Accessu

Rozeznáváme

**vnitřní (přímé) vazby (INNER JOIN)** – v dialogovém okně vlastností 1

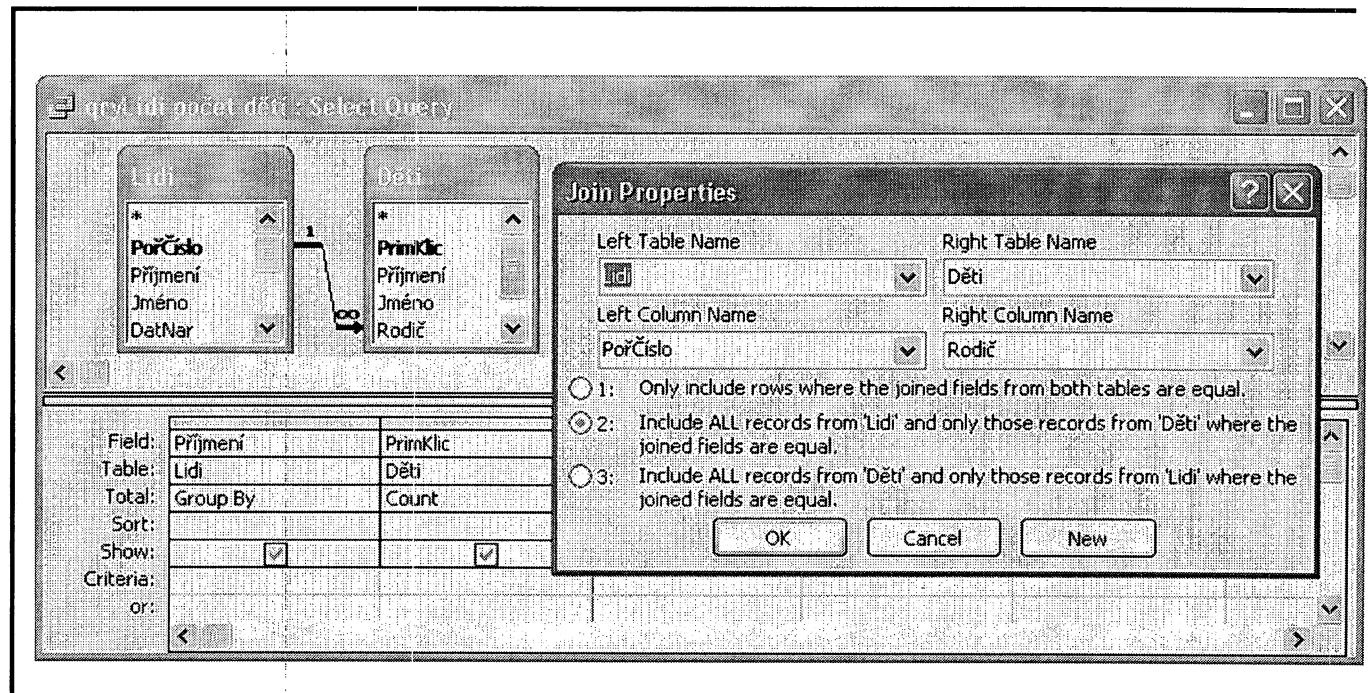
**vnější vazby (LEFT JOIN, RIGHT JOIN)** – v dialogovém okně vlastností 2,3

**Změna spojení:**

Typ spojení můžeme měnit

- v okně **Relationship** databáze
- pro konkrétní dotaz v tabulkové části návrhového zobrazení dotazu.

## Příklad: Databáze ZamDěti, počet dětí rodičů z tabulky Lidi

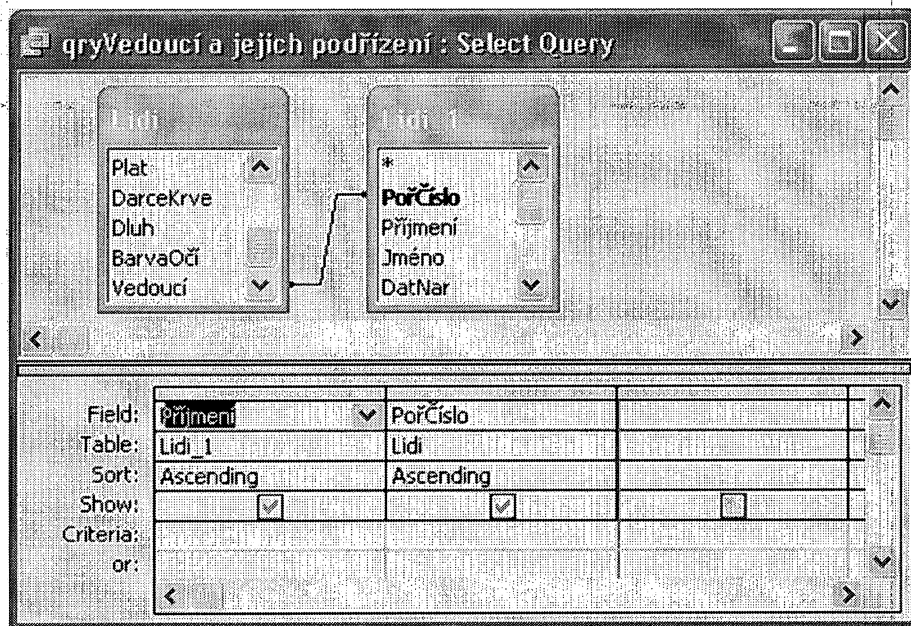


qryLidi počet dětí : Select ...

|   | Příjmení | CountOfPrimKlic |
|---|----------|-----------------|
| ► | Bouchal  | 0               |
|   | Fousková | 2               |
|   | Klímová  | 0               |
|   | Kouba    | 1               |
|   | Kousal   | 1               |

Record: 1 of 5

## Zrcadlové spojení – spojení tabulky do sebe sama



### Spojení tabulky do sebe sama

#### Postup

- V návrhovém režimu umístíme tabulku do tabulkové části dotazu dvakrát.
- Vytvoříme odpovídající spojení a nastavíme další potřebné atributy dotazu.
- Spustíme dotaz.



## Příklad: Databáze ZamDěti, vedoucí a počet podřízených

The screenshot displays the Microsoft Access interface. The main window is titled 'qryVedoucí a jejich podřízení : Select Query'. It shows a query design grid with two tables: 'Lidi' and 'Lidi\_1'. The 'Lidi' table has fields: Plat, DarceKrye, Dluh, BarvaOči, and Vedoucí. The 'Lidi\_1' table has fields: \* (primary key), PorČíslo, Příjmení, Jméno, and DatNar. A line connects the 'Vedoucí' field in 'Lidi' to the '\*' field in 'Lidi\_1'. Below the design grid is a table showing the query results:

| Field:    | Příjmení                            | PorČíslo                            |
|-----------|-------------------------------------|-------------------------------------|
| Table:    | Lidi_1                              | Lidi                                |
| Total:    | Group By                            | Count                               |
| Sort:     | Ascending                           | Ascending                           |
| Show:     | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Criteria: |                                     |                                     |
| or:       |                                     |                                     |

Below the table is a smaller window titled 'qryVedoucí a jejich podříze...' showing the results of the query:

| Příjmení | CountOfPorČíslo |
|----------|-----------------|
| Bouchal  | 3               |
| Kouba    | 2               |

At the bottom of this window, it says 'Record: 1 of 2'.

V tabulce **Lidi** databáze **ZamDěti** vytvoříme pole **Vedoucí**, v němž uchovááme pro každého zaměstnance primární klíč vedoucího. Ve sloupci však zobrazujeme příjmení vedoucího. Použijeme k tomu průvodce vyhledáváním.

Potom vytvoříme dotaz zobrazující k vedoucímu počet jeho podřízených.



- ❶ V dotaze **qryCelkCeny** zobrazme záznamy s desíti nejvyššími cenami.

Postup:

U pole **CelkCeny** nastavíme v řádku **Sort Descending**

Do okna **Top Values** vložíme číslo **10**.

- ❷ V databázi **Northwind** sestavme dotaz zobrazující všechny země z tabulky **Suppliers**, každou však právě jednou.

Postup:

1. Do tabulkové části návrhového zobrazení dotazu dáme tabulku **Suppliers**.
2. Do kritériální mřížky umístíme pole **Country**.

V dialogu pro vlastnosti dotazu nastavíme **Unique Values** na **Yes**.

- ❸ V databázi **Northwind** sestavme parametrický dotaz, který určí souhrnné množství a názvy výrobků objednaných od dodavatelů ze zadané země (parametr). Dotaz uložíme jako **qrySumaVýrobkůZemě**.

Postup:

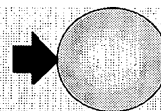
1. V dotaze použijeme tabulky **Suppliers**, **Products** a **Order Details**.
2. Do kritériální mřížky dáme pole **Quantity** z tabulky **Order Details** pole **ProductName** z tabulky **Products** a pole **Country** z tabulky **Suppliers**.
3. Klepneme na tlačítko **Totals**. V řádku **Total** potom volíme **Sum** pro pole **Quantity**, **Group By** pro pole **ProductName** a **Where** pro pole **Country**.
4. Pole **Country** omezíme parametrem dotazu.

- ❹ V dotaze **qrySumaVýrobkůZemě** vypisujeme jen záznamy, pro které je počítaná suma větší nebo rovna zadané hodnotě (parametr).

Postup:

Na pole **Quantity** klademe kritérium **>=[Dolní mez součtu]**





- Parametrický dotaz lze spouštět pokaždé s jinými hodnotami parametrů.
- Dotaz umožňuje vypočítávat také souhrnné informace.
- Výsledek dotazu nad více tabulkami je ovlivněn nastaveným typem spojení.



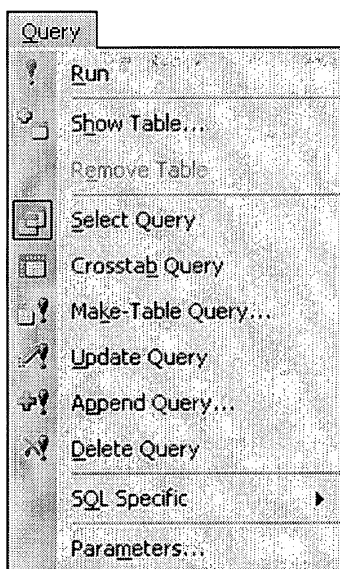


# Akční dotazy

## *Přehled témat*

- charakteristika akčních dotazů
- vytvářecí dotaz
- aktualizací dotaz
- přidávací dotaz
- odstraňovací dotaz

# Akční dotazy



## Akční dotazy (Action Queries)

### Charakteristika

Mění data v tabulkách.

### Klady

Efektivně sestavený akční dotaz udělá v databázi obrovský kus práce, která by byla často ručně těžko proveditelná.

### Zápory

Jejich účinek **nelze** vracet zpět. Spustíme-li akční dotaz, který je chybně sestaven, dochází k nenávratné změně dat.

### Obrana proti poškození databáze špatným akčním dotazem

Zálohování dat, pečlivé otestování dotazu se záložními daty dříve, než jej naostro spustíme.

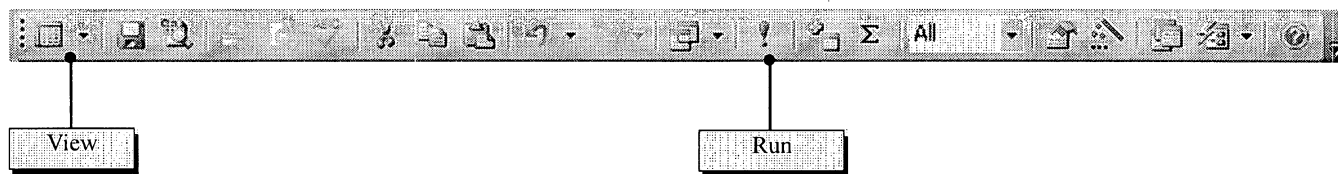
**Zásada:** Před provedením akčního dotazu vždy mít zálohovaná původní data.

### Testování akčního dotazu při jeho tvorbě

Často je možné simulovat akční dotaz dotazem výběrovým. Získáme tak data, na která bude náš akční dotaz působit. Jsou-li správná, transformujeme tento výběrový dotaz na akční.



Ve výběrových dotazech nebylo rozdílu v použití tlačítek **View** a **Run** panelu s nástroji. Obě dvě spouští z návrhového režimu výběrového dotazu tento dotaz.



U akčních dotazů však tlačítkem **View** pouze testujeme běh dotazu (zobrazí se pouze data, která by dotaz modifikoval, kdyby byl opravdu spuštěn), kdežto tlačítkem **Run** děláme dotaz naostro. Naneštěstí však data zobrazovaná po stisknutí tlačítka **View** nejsou často dostatečně průkazná pro otestování dobré funkce dotazu.

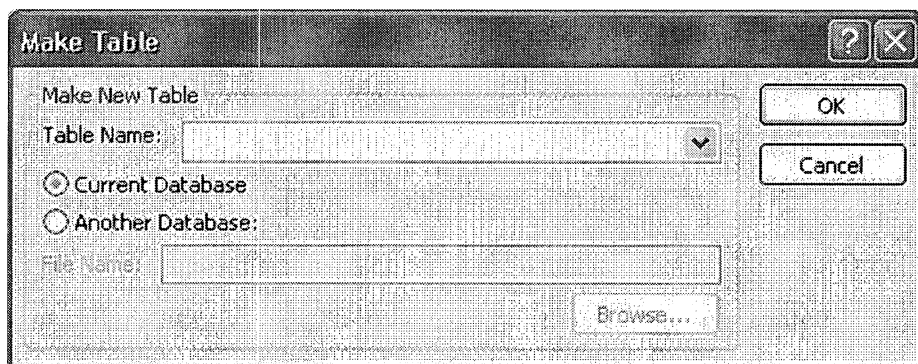
## Vytvářecí dotaz (Make-table Query)

### Charakteristika

Udělá novou tabulku. Je ze všech akčních dotazů nejméně nebezpečný. Sloupce této nové tabulky přejímají datové typy původních sloupců, ne však všechny další vlastnosti. Nová tabulka nemá primární klíč, vlastnosti sloupce získané aplikací průvodce vyhledáváním se též nepřenášejí do nové tabulky.

### Postup tvorby

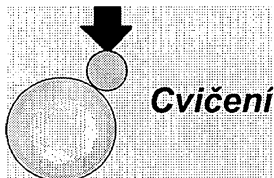
Vytvoříme výběrový dotaz, specifikujeme potřebná kritéria, ověříme jeho chování spuštěním. Nakonec jej změním na dotaz vytvářecí. Objeví se dialog vytvořit tabulku, v němž vyplníme název tabulky, kterou dotaz vytvoří.



*Existuje-li již v naší databázi tabulka určeného jména, je tato nejdříve odstraněna a vzápětí znovu vytvořena.*

poznámka





## Cvičení

### Úkoly k procvičování:

- 1 Sestavme v databázi Northwind vytvářecí dotaz, který udělá tabulku **Suma kategorií na skladě**. Tabulka obsahuje sloupec **CategoryName** a sloupec součtů jednotek na skladě pro každou kategorii.

qrySuma jednotek na skladě pro kategorie : Make Table Query

Categories

- \* CategoryID
- CategoryName
- Description
- Picture

Products

- QuantityPerUnit
- UnitPrice
- UnitsInStock
- UnitsOnOrder
- ReorderLevel

| Field:    | CategoryName                        | SoučetJednotekNaSkladě: UnitsInStock |
|-----------|-------------------------------------|--------------------------------------|
| Table:    | Categories                          | Products                             |
| Total:    | Group By                            | Sum                                  |
| Sort:     |                                     |                                      |
| Show:     | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/>  |
| Criteria: |                                     |                                      |
| or:       |                                     |                                      |

Query Properties

General

Description . . . . .

Output All Fields . . . . . No

Top Values . . . . . All

Unique Values . . . . . No

Unique Records . . . . . No

Run Permissions . . . . . User's

Source Database . . . . . (Current)

Source Connect Str . . . . .

Destination Table . . . . . Suma kategorií na skladě

Destination DB . . . . . (current)

Dest Connect Str . . . . .

Use Transaction . . . . . Yes

Record Locks . . . . . Edited Record

Suma kategorií na skladě : Table

| CategoryName   | SoučetJednotekNaSkladě |
|----------------|------------------------|
| Beverages      | 559                    |
| Condiments     | 507                    |
| Confections    | 386                    |
| Dairy Products | 393                    |
| Grains/Cereals | 308                    |
| Meat/Poultry   | 165                    |
| Produce        | 100                    |
| Seafood        | 701                    |

Record: 1 of 8

## Aktualizační dotazy (Update Queries)

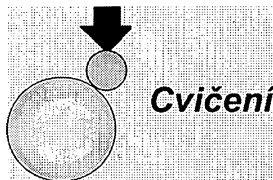


### Charakteristika

Aktualizují data v tabulkách.

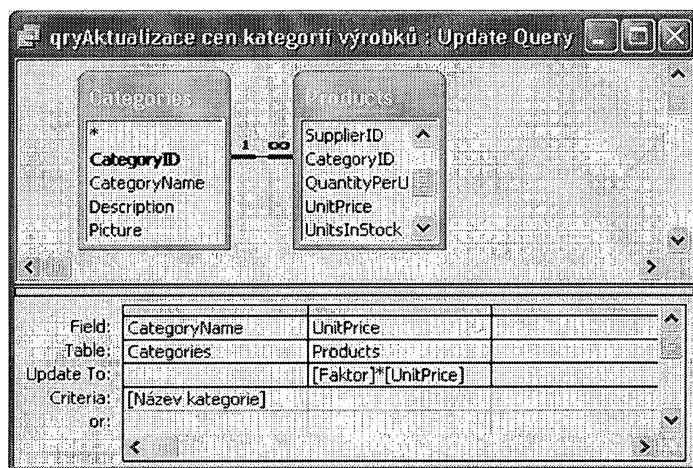
### Postup tvorby

Po změně typu dotazu na aktualizací se v kritériální mřížce objeví řádek **Update To**. Do tohoto řádku píšeme v příslušném sloupci mřížky vzorec, který mění hodnoty odpovídajícího pole. Kritéria přitom formulujeme běžným způsobem v řádcích **Criteria**.

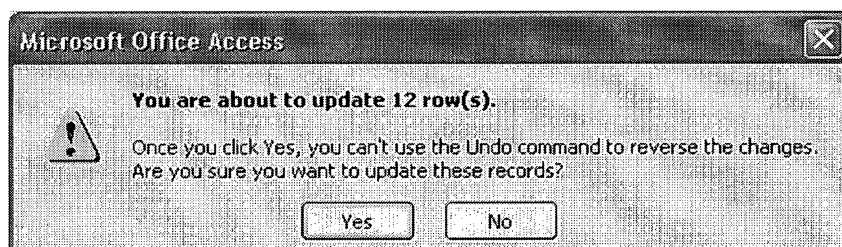


## Úkoly k procvičování:

- 1 Sestavme v databázi Northwind aktualizací dotaz na změnu cen výrobků vybrané kategorie. Cenu měníme násobením původní ceny určitým číslem (faktorem), což je parametr dotazu. Vznikne tedy aktualizací a zároveň parametrický dotaz:



Poslední zpráva před vlastní aktualizací



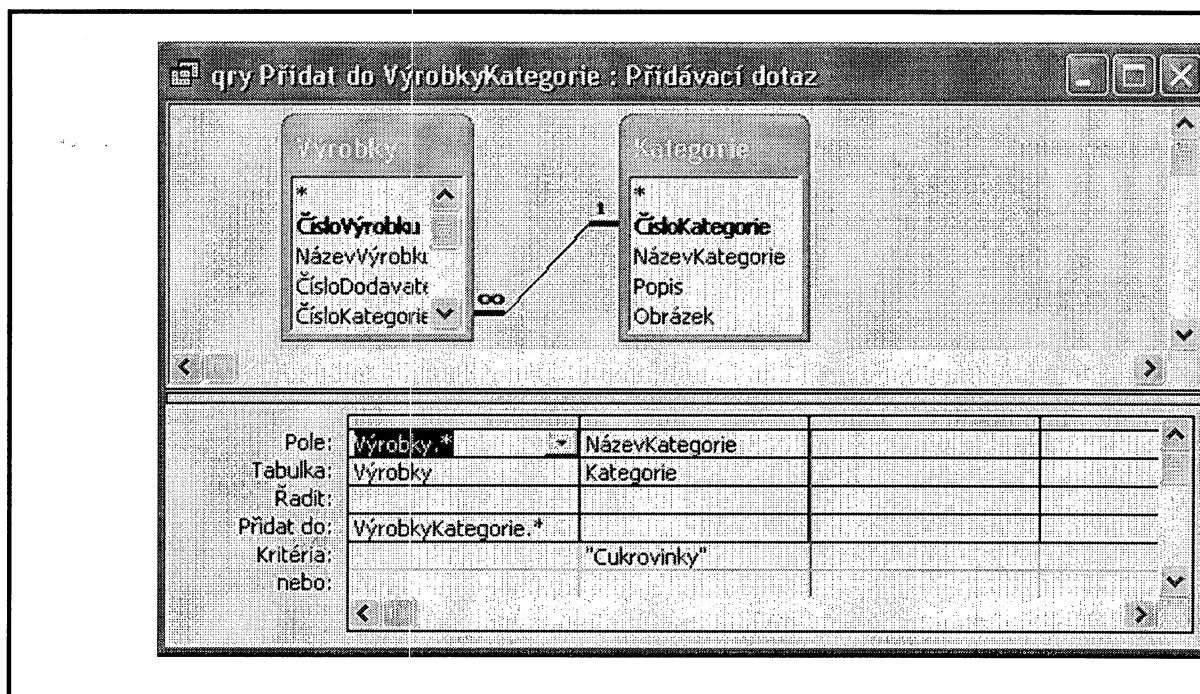
- 2 Sestavme aktualizací dotaz, který v databázi **ZamDěti** zvýší zaměstnancům narozeným před určitým datem (zadejme jako parametr) plat o určité procento (opět zadáme jako parametr):

1. Vytvoříme nový dotaz, který má v tabulkové části návrhového zobrazení tabulku **Lidi**.
2. Příkazem **Query/Update Query** změníme jeho typ na aktualizací dotaz.
3. Do kritériální mřížky dotazu vložíme pole obsahující datum narození **DatNar** a pole **Plat**.
4. Na pole **DatNar** vložíme kritérium **<=[Zadejte datum narození]**
5. Do řádku **Update To** do pole **Plat** vložíme výraz **[Koeficient navýšení]\*[Plat]**

Dotaz uložíme, tlačítkem **View** zkontrolujeme běh dotazu, tlačítkem **Run** dotaz spustíme.



## Přídávací dotazy



### Přídávací dotazy (Append Queries)

#### Charakteristika

Umožňují kopírovat záznamy či jejich části ze zdrojové tabulky do tabulky cílové. Kritériem přitom určíme, o které záznamy a která pole záznamů zdrojové tabulky jde.

#### Postup provedení

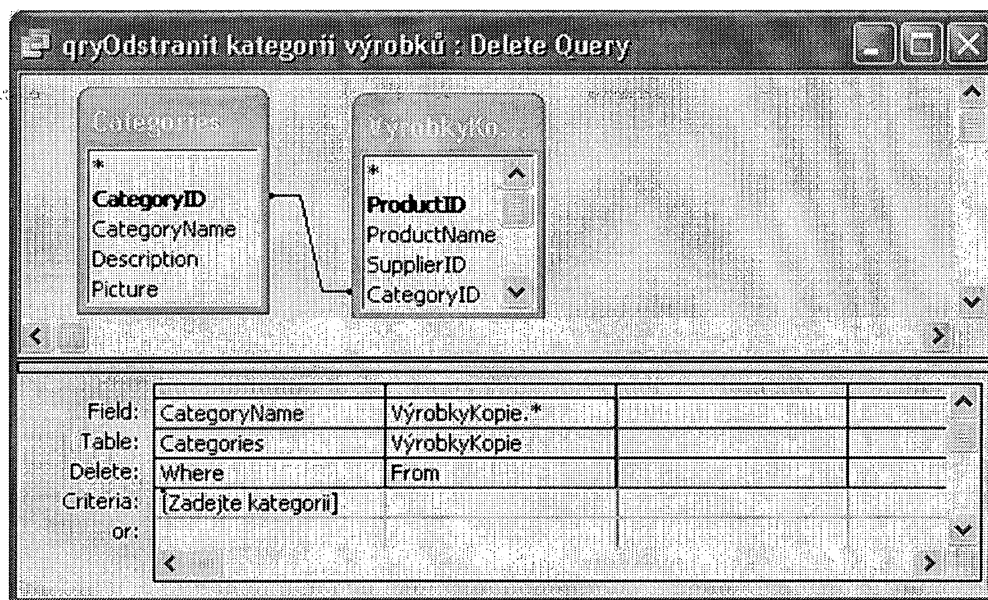
Do tabulkové části návrhového zobrazení dotazu umístíme zdrojovou tabulku a tabulky potřebné pro tvorbu kritérií. Přídávacím dotazem nelze v cílové tabulce vytvořit nové pole. Vytvoříme nový dotaz a změníme ho na přídávací. Přitom budeme muset zadat specifikaci cílové tabulky.

Shodují-li se všechna pole zdrojové a cílové tabulky, stačí do kritériální mřížky přidat pole \* (pracujeme-li však v replice databáze, nutno přidat každé pole jednotlivě). V řádku **Append To** se objeví název cílové tabulky, tečka a \*. Do mřížky dále zapíšeme příslušná kritéria, ověříme správnost dotazu a spustíme jej.

Přidáváme-li jen některá pole, určíme u každého pole zdrojové tabulky v řádku **Append To**, do kterého pole tabulky cílové jej přidáme. Datové typy těchto polí by se buď měly shodovat, nebo by se typ zdrojového pole měl dát převést na datový typ pole cílového (číselný datový typ se dá převést na text, ne však naopak).

Chceme-li při převodu zachovat hodnoty pole typu **AutoNumber**, přidáme jej do cílové tabulky. Nepřidáme-li naopak do pole typu **AutoNumber** cílové tabulky nic, vyplní Access toto pole v cílové tabulce běžným způsobem sám.

# Odstraňovací dotazy



## Odstraňovací dotazy (Delete Queries)

### Charakteristika

Na základě specifikovaného kritéria odstraňují záznamy. Lze odstraňovat z jedné tabulky i z více tabulek v relaci 1:1 současně. Při odstraňování záznamů z tabulky na straně jedna relace 1:N se může projevit efekt kaskádového odstraňování (viz dále).

### Postup provedení

V kritériální mřížce odstraňovacího dotazu je řádek **Delete**. Do mřížky nutno přidat pole \* z tabulky, z níž chceme odstraňovat. V řádku **Delete** se pro toto pole objeví text **From**. Do mřížky přidáme ještě pole ke specifikaci kritérií. V nich se v řádku **Delete** objeví nápis **Where**. Poté otestujeme a provedeme dotaz.

tip

*Odstraňovací dotaz nejdříve realizujeme jako výběrový, abychom se přesvědčili o správnosti nadefinovaných kritérií.*

tip

### **Pro tabulky v relaci 1:N**

*Obsahuje-li dotaz pouze tabulku na straně jedna relace 1:N, a je-li mezi oběma tabulkami nadefinována referenční integrita, nelze z tabulky na straně jedna odstranit záznamy, které mají partnerské záznamy v tabulce na straně N. Povolíme-li v tomto případě odstraňování souvisejících polí v kaskádě, odstraní se spolu se záznamem ze strany jedna tabulky všechny partnerské záznamy z tabulky na straně N. Jsou-li dvě tabulky v relaci 1:N a obsahuje-li dotaz obě tyto tabulky a my odstraňujeme z tabulky na straně jedna, nutno nastavit vlastnost **Unique Records** na **Yes**. Blíže o této vlastnosti viz nápověda Accessu.*

tip

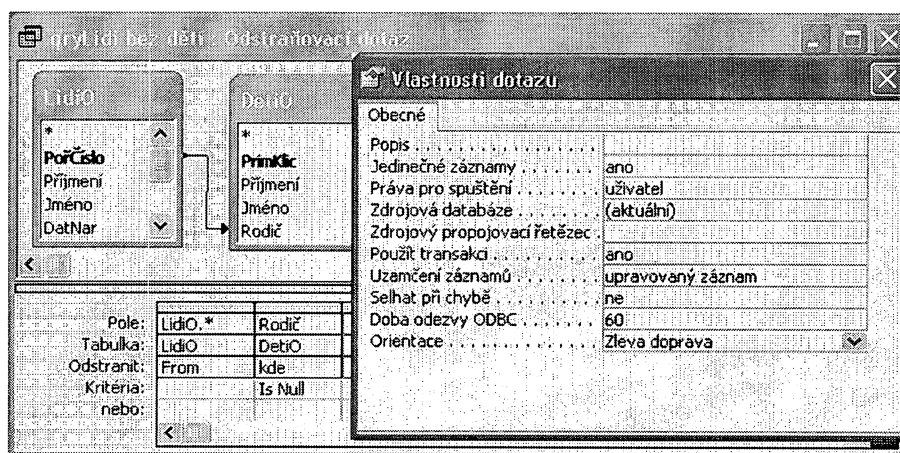
### **Odstranění záznamů s duplicitními poli přidávacím dotazem**

*Z tabulky s duplicitami zkopírujeme strukturu do nové tabulky. Z polí, která obsahují duplicity vytvoříme v nové tabulce primární klíč. Spustíme do nové tabulky přidávací dotaz ze staré. Vzhledem k tomu, že pole, která slouží jako primární klíč, nemohou obsahovat duplicitní hodnoty, bude výsledkem procedury tabulka bez duplicit v těchto sloupcích.*

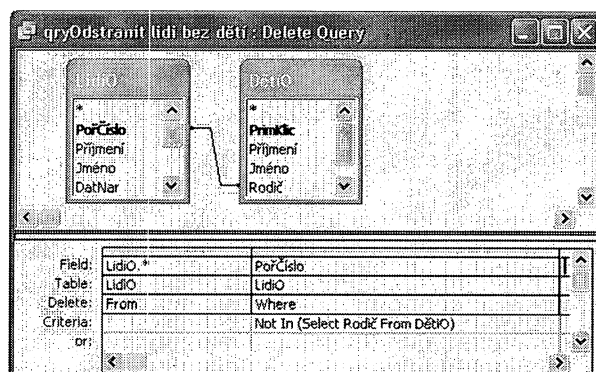
## Úkoly k procvičování:

- 1 V databázi **ZamDěti** odstraníme zaměstnance bez dětí. Zaměstnanci jsou v tabulce **LidiO**, jejich děti v tabulce **DětiO**, relace mezi tabulkami je 1:N.

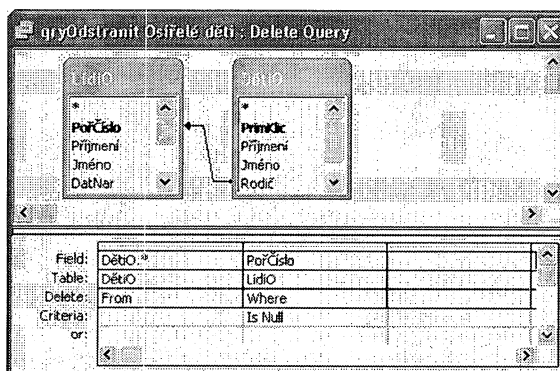
1. způsob:



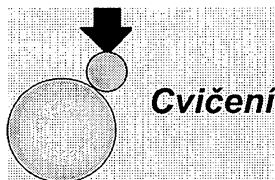
2. způsob



- 2 Jsou dvě tabulky **LidiO** a **DětiO** v relaci 1:N. Příklad řeší odstranění osiřelých záznamů z tabulky na straně N (**DětiO**).



Postup v tomto příkladě uvedený bychom měli aplikovat na tabulku na straně N relace 1:N v případě, že chceme mezi tabulkami nadefinovat referenční integritu a osiřelé záznamy v tabulce na straně N tomu brání.



## Úkoly k procvičování:

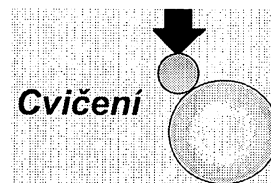
- 3 V databázi **ZamDěti** vytvoříme tabulku obsahující zaměstnance (pole primární klíč a příjmení) a počet jejich dětí.

1. V okně databáze vyznačíme objekt **Queries** a klepneme na tlačítko **New**. Volíme Design View dotazu.
2. Do tabulkové části okna s dotazem přidáme tabulky **Lidi** a **Děti**.
3. Do kritériální mřížky dotazu dáme z tabulky **Lidi** pole primárního klíče a pole příjmení. Z tabulky **Děti** sem vložíme pole primárního klíče.
4. Z dotazu uděláme souhrnný dotaz (klepnutím na tlačítko **Totals**). V řádku **Souhrn** volíme pro primární klíč z tabulky **Děti** funkci **Count**.
5. Klepnutím na tlačítko **View** či **Run** zkontrolujeme výsledky.
6. Příkazem **Query/Make-Table Query** uděláme z doposud výběrového dotazu dotaz vytvářecí a zadáme název tabulky, která z dotazu vznikne.
7. Tlačítkem **View** zobrazíme výsledky dotazu.
8. Tlačítkem **Run** dotaz provedeme.

Vzniklý dotaz uložíme a zkontrolujeme druh ikony, který je dotazu v okně Database přiřazen.

- 4 V databázi **ZamDěti** zkopírujeme strukturu tabulky **Lidi** do nové tabulky **LidiN**. Pak sestavíme přidávací dotaz, který do **LidiN** přidá záznamy z tabulky **Lidi** s platem větším, než zadaná hodnota (parametr):

1. Na kartě tabulky okna databáze vyznačíme tabulku **Lidi**.
2. Příkazem CTRL+C ji vložíme do schránky
3. Příkazem CTRL+V ze schránky vyjmeme její strukturu (volíme v dialogovém okně příkazu), jako název nové tabulky píšeme **LidiN**.
4. Přejdeme na kartu **Dotazy**, vytvoříme nový dotaz v návrhovém zobrazení a do jeho tabulkové části přidáme tabulku **Lidi**.
5. Typ dotazu měníme příkazem **Query/Append Query** dotaz na přidávací. Objeví se dialogové okno, v němž volíme název tabulky, do které přidáváme – tabulka **LidiN**.



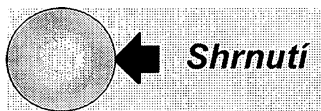
## Úkoly k procvičování:

6. Do kritériální mřížky dotazu vložíme pole \* a **Plat**.
7. Zkontrolujeme obsah řádku **Append To**. V poli **Plat** musí být tento řádek prázdný.
8. V poli **Plat** definujeme kritérium  $\geq$ [**Dolní mez platu**].

Dotaz uložíme, otestujeme a spustíme.

- 5** V databázi **Northwind** vytvoříme kopie tabulek **Customers** a **Orders** a nazvěme je **CustomersK** a **OrdersK**. Udělejme mezi nimi stejnou relaci, jako mezi tabulkami **Customers** a **Orders**. Povolme však kaskádové odstraňování. Potom zhotovme odstraňovací dotaz pro zákazníky v tabulce **CustomersK**, odstraňující zákazníky z určité země (parametr dotazu). Sledujme efekt kaskádového odstraňování v tabulce **OrdersK**.

1. Otevřeme databázi **Northwind** a aktivujeme objekt **Tabulky**.
2. Zkopírujeme tabulky **Customers** a **Orders** na nové tabulky **CustomersK** a **OrdersK**. Kopírujeme strukturu i data.
3. Aktivujeme okno databáze a klepnutím na tlačítko **Relationships** zobrazíme okno relací
4. Klepneme na tlačítko **Show Table** a přidáme do okna **Relationships** tabulky **CustomersK** a **OrdersK**.
5. Vytvoříme mezi nimi stejnou relaci, jaká je mezi tabulkami **Customers** a **Orders**. Navíc však povolíme odstraňování souvisejících polí v kaskádě.
6. Zavřeme okno **Relationships** a přitom uložíme jeho nový obsah.
7. V okně databáze aktivujeme dotazy, uděláme nový dotaz v návrhovém zobrazení a do jeho tabulkové části dáme tabulku **ZákazníciK**.
8. Příkazem **Query/Delete Query** měníme typ na odstraňovací.
9. Do kritériální mřížky dáme z tabulky **CustomersK** pole \* a pole **Country**. Zkontrolujeme obsah řádku **Delete**.
10. Na pole **Country** vkládáme kritérium [**Zadejte zemi**].
11. Dotaz uložíme, otestujeme a spustíme.
12. Porovnáme počet záznamů v tabulce **Orders** a **OrdersK**.



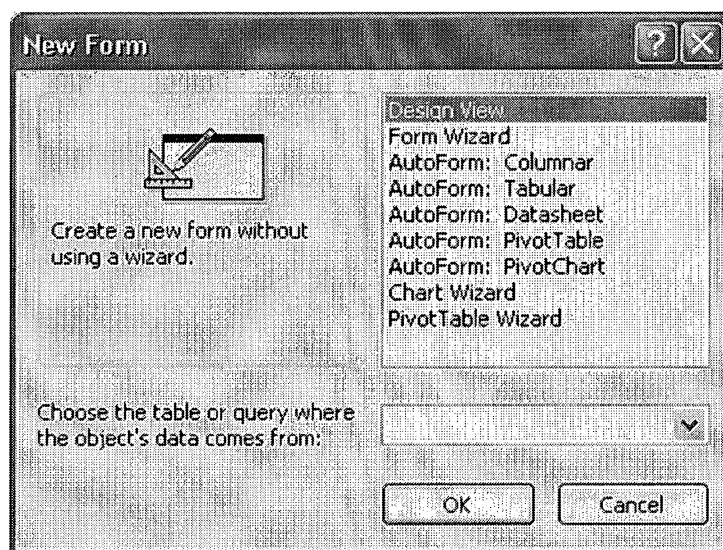
- Akční dotazy mění data v tabulkách. Změny, které provedou, jsou ovšem nevratné.
- Make-Table Query vytvoří novou tabulku z dat vybraných z tabulek existujících.
- Update Query mění hodnoty polí v tabulce.
- Append Query kopíruje záznamy z jedné tabulky do jiné.
- Delete Query odstraňuje z tabulky záznamy splňující určená kritéria.



# Formuláře

## *Přehled témat*

- použití formulářů
- návrhové zobrazení formuláře
- vlastnosti formulářů
- vlastnosti ovládacích prvků



## Použití formulářů

- Zobrazení a úpravy dat. Běžný uživatel málokdy pracuje přímo s tabulkami a dotazy. Autor databáze pro něj často dělá formuláře, které práci s daty usnadní.
- Provádění dodatečných výpočtů s daty. Často lze na formuláři dělat další výpočty s daty, které bychom jinak realizovali velice obtížně.
- Tvorba dialogových oken.

## Tvorba formulářů

- **Návrhové zobrazení (Design View)**

Autor dělá celý formulář od začátku sám. Je to nejpracnější způsob tvorby formuláře, ale zároveň umožňuje využít všech možností Accessu a celého spektra tvůrčích schopností autora.

- **Průvodce formulářem (Form Wizard)**

Vede autora tvorbou formuláře z jednoho či více datových zdrojů. Nabízí i možnosti vzniku podformulářů.



- **Automatické formuláře (AutoForms)**

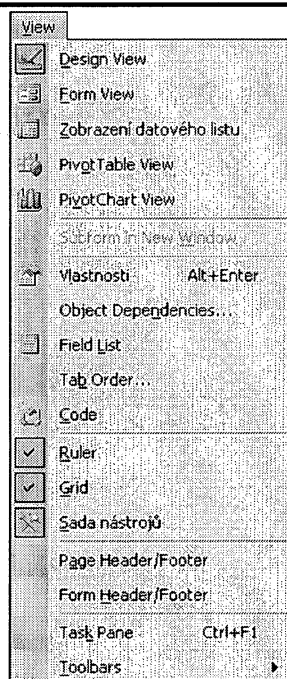
Autor zvolí zdroj dat a Access vytvoří formulář příslušného typu. Automatickému formuláři tabelárnímu (**AutoForm Tabular**) říkáme také nekonečný formulář, protože zobrazuje najednou více záznamů (v řádcích pod sebou). Automatický formulář sloupcový (**AutoForm Columnar**) zobrazuje vždy pouze jeden (aktivní) záznam. Na další záznamy přepínáme pomocí navigační lišty.

## **Rozdělení formulářů podle toho, zda jsou či nejsou napojeny na datový zdroj**

**Vázané formuláře** mají určen zdroj dat - záznamů (**Record Source**). Jeho pole se zobrazují v seznamu polí (**Field List**) formuláře (to je speciální okno, o kterém budeme dále diskutovat). Zdrojem dat pro formulář jsou tabulky či dotazy. Zdroj dat formuláře najdeme či upravíme pod vlastností **Record Source** na kartě **Data** okna **Properties** (viz dále).

**Nevázané formuláře** nemají zdroj dat. Užíváme je jako dialogová okna, či k navigaci po aplikaci (pomocí ovládacích prvků – tlačítek apod. umístěných na formulář). Takovým typickým formulářem je formulář nazvaný **Main Switchboard** v databázi **Northwind**.

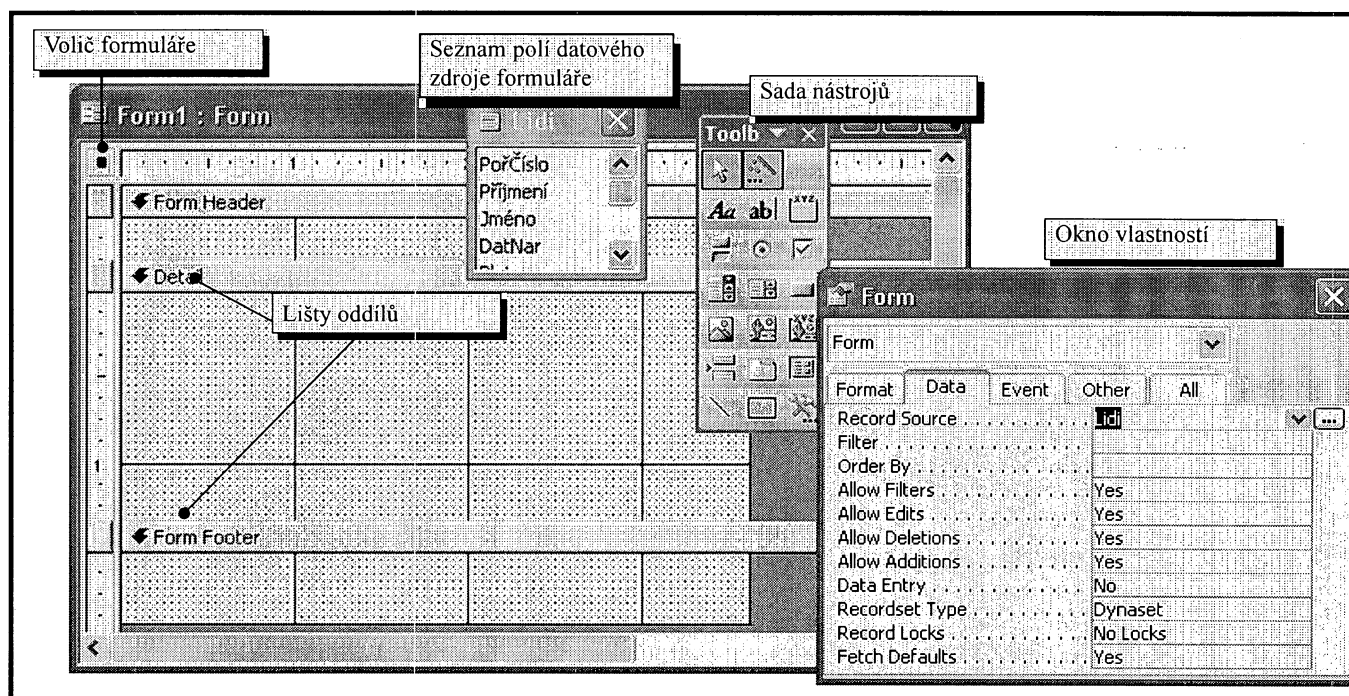
# Způsoby zobrazení formuláře



## Způsoby zobrazení formuláře

- **Návrhové zobrazení (Design View)** - tvorba a editace návrhu formuláře.
- **Formulářové zobrazení (Form View)** - zobrazení dat ve formuláři spolu s příslušnými sekcemi formuláře = záhlaví, zápatí – a dalšími ovládacími prvky.
- **Zobrazení datového listu (Datasheet View)** - zobrazení dat formuláře „excelovským způsobem“, bez sekcí a ovládacích prvků vložených na formulář.
- **Zobrazení kontingenční tabulky a kontingenčního grafu (Pivot Table/Pivot Chart View)** - viz dále.

# Návrhové zobrazení formuláře



## Základní součásti návrhového zobrazení formuláře

### Formulář (Form) a jeho oddíly (Sections)

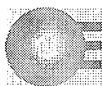
- **Záhlaví a zápatí (Form Header/Footer) formuláře** - zobrazují informace, které se nemění záznam od záznamu - např. nadpisy nebo obrázky, tlačítka apod.
- **Tělo (Detail)** (zobrazuje záznamy datového zdroje)
- **Záhlaví a zápatí stránky (Page Header/Footer)** - objeví se až v režimu **Print Preview** či po vytisknutí formuláře. Proto nebudeme-li formulář tisknout, je zbytečné záhlaví a zápatí stránky dělat.

**Seznam polí (View/Field List).** Obsahuje seznam polí datového zdroje. Z tohoto ovládacího prvku se pole vkládají do formuláře. Pomocí kláves **Shift** a **Ctrl** se dá vyznačovat a poté vkládat do těla formuláře více polí najednou.

Při vložení pole na formulář (a podobně i na sestavu) se s ním automaticky vkládá i jeho popis. Do něj se vloží obsah vlastnosti **Caption** z návrhu tabulky. Proto je vhodné při tvorbě tabulky tuto vlastnost pečlivě vyplňovat. Pokud **Titulek** není zadán, vloží se název pole.

**Okno vlastností (View/Properties)** obsahuje vlastnosti formuláře a jeho součástí (ovládacích prvků, polí), rozdělené podle druhu na jednotlivé karty. Je-li zobrazeno okno vlastností, je vždy navrchu. Klepneme-li potom na volič formuláře, objeví se v okně vlastnosti formuláře. Klepneme-li na jiný ovládací prvek na formuláři, objeví se v tomto okně vlastnosti aktivního prvku. Je-li okno vlastností zavřeno, lze poklepáním na libovolný prvek formuláře zobrazit okno vlastností pro tento prvek. U formuláře samotného poklepeme na volič formuláře.

**Sada nástrojů (View/Toolbox).** Ovládací prvky (**Controls**) zde umístěné vkládáme do formuláře.



## Příklad

Tvorba formuláře, který zobrazuje celkovou cenu za každou objednávku:

1. Sestavíme souhrnný dotaz **qrySumaZaObjednávku**, který bude sloužit jako zdroj záznamů pro formulář.

2. Na kartě **Forms** okna databáze klepneme na tlačítko **New** a v dialogovém okně **New Form** vyznačíme **AutoForm: Tabular** a zvolíme právě vytvořený dotaz jako zdroj dat.

3. Po klepnutí na tlačítko OK získáme příslušný formulář.

Volič záznamů

vySumaZaObjednávku

| Order ID | Customer               | Zahlavi formuláře | Celk.Cena  |
|----------|------------------------|-------------------|------------|
| 10248    | Wilman Kala            |                   | \$440.00   |
| 10249    | Tradição Hipermercados |                   | \$1,863.40 |
| 10250    | Hanari Carnes          |                   | \$1,813.00 |
| 10251    | Victualles en stock    |                   | \$670.80   |
| 10252    | Suprêmes délices       |                   | \$3,730.00 |
| 10253    | Hanari Carnes          |                   | \$1,444.80 |
| 10254    | Chop-suey Chinese      |                   | \$625.20   |
| 10255    | Richter Supermarkt     |                   | \$2,490.50 |
| 10256    | Wellington Importadora |                   | \$517.80   |
| 10257    | HILARIÓN-Abastos       |                   | \$1,119.90 |

Record: 1 of 830

Navigační systém

4. Tentýž datový zdroj, ale formulář vytvořen jako **AutoForm: Columnar**:

Volič záznamů

vySumaZaObjednávku

Order ID: 10248

Customer: Wilman Kala

Celk.Cena: \$440.00

Record: 1 of 830

Navigační systém



## Základní součásti formuláře ve formulářovém zobrazení

- Titulní pruh okna s titulkem (Caption) a tlačítka okna (Control Box, Min/Max Buttons))
- Záhloví, zápatí, tělo formuláře (Form Header/Footer, Detail)
- Navigační ovládací prvek - navigační systém formuláře (Navigation Buttons)
- Volič záznamů - s černou trojúhelníkovou šipkou na aktivním záznamu (Record Selectors)
- Dělicí čáry (Dividing Lines)

## Základní techniky práce v návrhovém zobrazení formuláře

### Zobrazení záhlaví a zápatí formuláře (stránky)

- Příkazem *View/Form Header/Footer* resp. *View/Page Header/Footer*

### Vyznačení formuláře

- Klepnutím na volič formuláře

### Vyznačení oddílu formuláře

- Klepnutím na jeho lištu

### Změna velikosti oddílu formuláře

- Tahem za horní hranu lišty pod oddílem.

### Změna velikosti podkladové mřížky formuláře

- Tahem za její hranu

### Vložení pole ze seznamu polí do formuláře

- Tahem položky pole ze seznamu polí (Field List) do příslušného oddílu formuláře

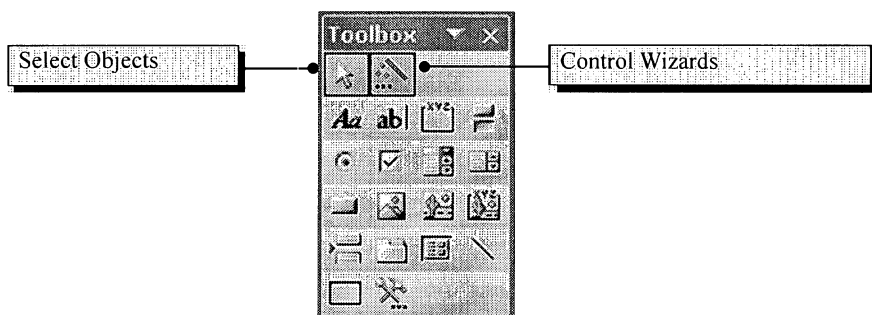
### Vložení více polí do formuláře najednou

- Pomocí kláves **Shift** a **Ctrl** pole v seznamu polí (Field List) vyznačíme a tahem vložíme do formuláře.

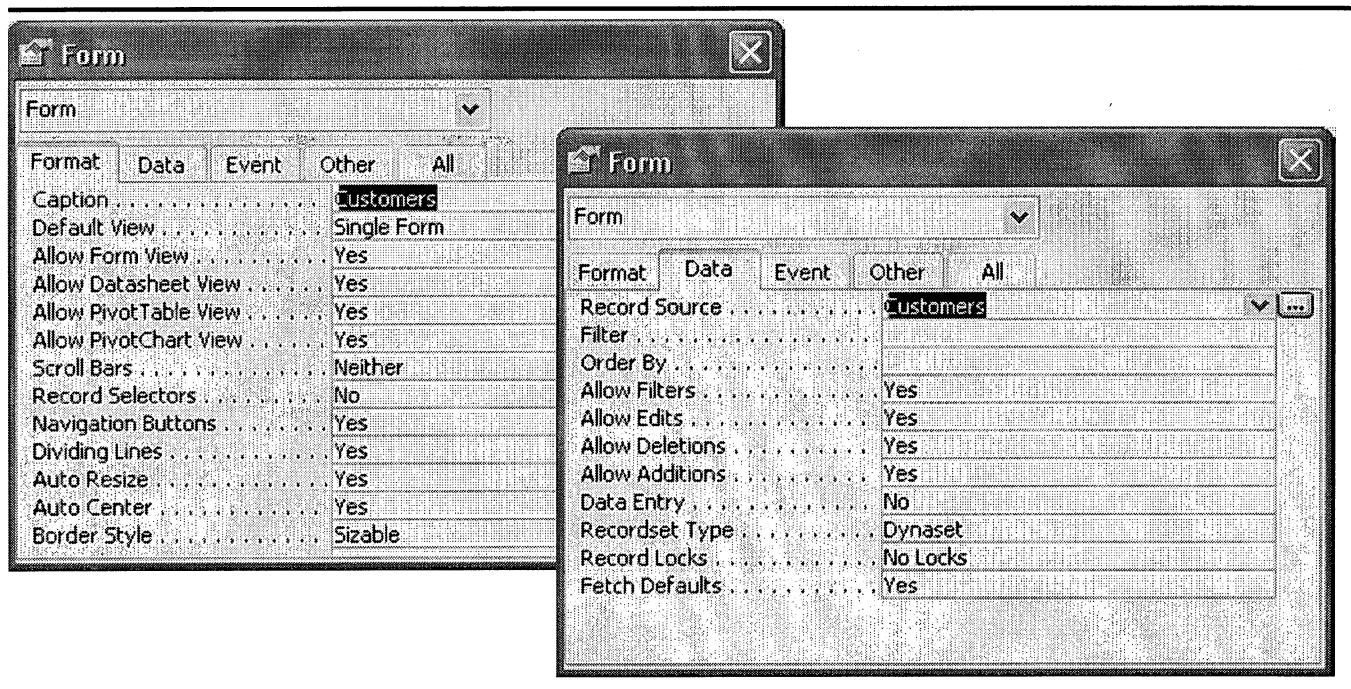
## Vložení ovládacího prvku ze soupravy nástrojů (Toolbox) do formuláře



- Klepneme na tlačítko s ikonou ovládacího prvku a potom tahem nakreslíme ovládací prvek jako obdélník na formuláři.
- Chceme-li kreslit více ovládacích prvků stejného typu, poklepeme v soupravě nástrojů na ikoně prvku a kreslíme. Specifikaci odvoláme klepnutím na tlačítko **Select Objects**.
- Některé ovládací prvky (seznamy, tlačítka) mají průvodce, který se ihned po nakreslení ovládacího prvku spustí. Průvodce pomáhá nastavit funkci ovládacího prvku. Spuštění průvodce zapínáme/vypínáme tlačítkem **Control Wizards**.



# Základní vlastnosti formuláře



## Základní vlastnosti formuláře

Okno vlastnosti formuláře snadno vyvoláme poklepáním na volič formuláře. Okno má celkem pět karet.

### Karta Format

Určujeme základní formátovací vlastnosti formuláře.

**Caption:** text v titulním pruhu formuláře.

**Default View:** způsob zobrazení dat po otevření formuláře.

Vlastnosti typu **Allow**.....: povolují či zakazují určité zobrazení.

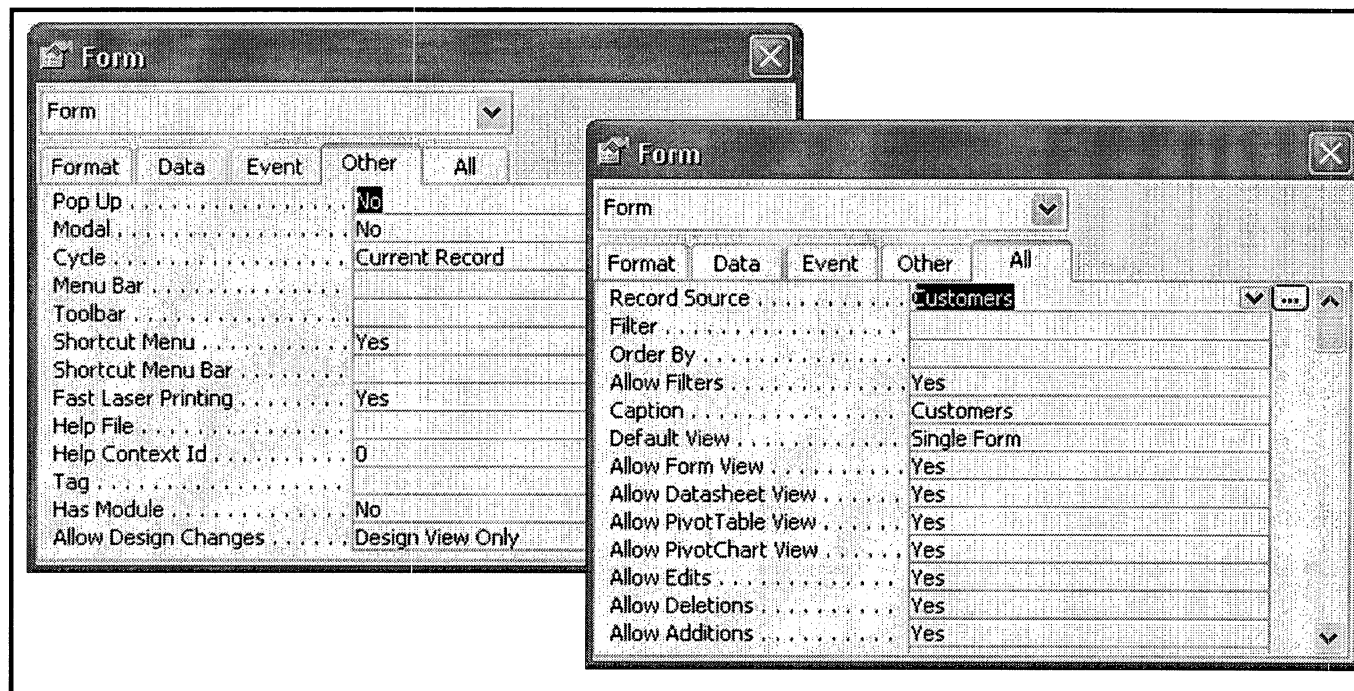
**Scroll Bars, Records Selector, Navigation Buttons, Dividing Lines:** určují, zda zobrazit či ne příslušné prvky na formuláři.

**Border Style:** má hodnoty Sizable, Thin, None, Dialog. Standardní nastavení této vlastnosti je Sizable, kdy můžeme měnit velikost okna formuláře.

**Dialogové okno.** Chceme-li vytvořit z formuláře dialogové okno, volíme nastavení vlastnosti Border Style na Dialog. **Modální dialogové okno** musí být zavřeno, jinak nelze v aplikaci dále pracovat. Tuto vlastnost získá formulář nastavením **Modal** na **Yes** (karta **Other**). Je-li třeba, aby toto okno bylo neustále nad ostatními okny v Accessu, volíme nastavení jeho vlastnosti **Pop Up okno** na **Yes** (karta **Other**). V tom případě není po otevření formuláře na obrazovce panel **Form Design**, ale panel **Database**.



# Základní vlastnosti formuláře



## Karta Data

**Record Source:** určuje u vázaného formuláře zdroj záznamů.

**Filter a Order By:** uchovávají poslední použitý filtr a řazení. Záznamy zobrazované na formuláři lze totiž podobně jako v tabulce filtrovat a řadit. Vlastnosti typu **Allow...** povolují či zakazují příslušné akce na formuláři. Povolení úprav přitom znamená úpravy v existujících záznamech kromě odstranění záznamů. Tato akce se povoluje/zakazuje vlastností **Allow Deletions**.

**Data Entry:** formulář zobrazí pouze nový záznam pro zadání dat do něj. Kromě vlastnosti **Data Entry** přepnuté na **Yes** je třeba nastavit i vlastnost **Allow Additions** na **Yes**.

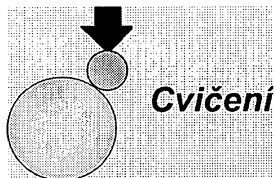
## Karta Event

Jednotlivé vlastnosti na této kartě uvedené jsou přiřazeny odpovídajícím událostem na formuláři. Jako hodnotu vlastnosti obvykle zadáváme název makra, či procedury ve Visual Basicu. Tento kód je automaticky vyvolán, nastane-li příslušná událost formuláře. Například otevření okna formuláře (**On Open**) můžeme obsloužit kódem, změnu aktivního záznamu (**On Current**) lze doprovázet vykonáním programu či makra apod.

## Karta Other

Obsahuje vlastnosti, které nebylo možno zařadit do předchozích karet. **Menu Bar, Shortcut Menu Bar, Toolbar:** V Accessu lze podobně jako v jiných aplikacích MS Office konstruovat vlastní nabídky a panely nástrojů. Zadáme-li do těchto vlastností název takové nabídky či panelu, zobrazí se po zobrazení formuláře. Řádek místní nabídky se zobrazí po klepnutí pravým tlačítkem na formuláři. Vlastnost **Shortcut Menu** musí být přitom nastavena na **Yes**.

**Allow Design Changes:** nastavíme-li na hodnotu **Design View Only**, je okno vlastností zobrazováno pouze v návrhovém zobrazení formuláře.



## Úkoly k procvičování:

- 1 V databázi **ZamDěti** vytvoříme sloupcový a tabulkový formulář pro tabulku **Lidi**. Každý z nich necht' má odpovídající záhlaví. **Formuláře** nazvěme **frmLidiS** a **frmLidiT**. Na obou formulářích přejdeme do návrhového a formulářového zobrazení a vyzkoušíme si práci s jednotlivými prvky zobrazení popsanými výše.

Postup:

1. V okně databáze **ZamDěti** aktivujeme objekt **Forms** a klepneme na tlačítko **New**.
2. V dialogovém okně pro nový formulář volíme jako zdroj dat tabulku **Lidi** a děláme **AutoForm: Columnar**.
3. V hotovém formuláři se přepneme do návrhového zobrazení a tahem za horní okraj lišty těla formuláře uděláme záhlaví cca 1 cm vysoké.
4. Zobrazíme soupravu nástrojů (**View/Toolbox** a klepneme na tlačítko **Caption**.
5. V záhlaví formuláře uděláme tahem myši popisek a vepíšeme sem text **Zaměstnanci**.
6. Klepneme na okraj popisku, abychom jej vyznačili, a jeho obsah vhodně formátujeme.
7. Okno hotového formuláře zavřeme a přitom jej uložíme jako **frmLidiS**.

Body 1 až 7 opakujeme pro **AutoForm: Tabular**.

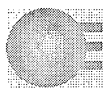
- [illegible]



# Ovládací prvky formuláře

## *Přehled témat*

- textové pole
- pole se seznamem
- tlačítko
- kalendář



## Vlastnosti ovládacích prvků

Je-li zobrazeno okno vlastností (*View/Properties*), pak se klepnutím na ovládací prvek zobrazí v okně vlastností vlastnosti tohoto prvku.

### Ovládací prvky (Controls)

- **Nevázané** nemají datový zdroj.
- **Vázané** mají datový zdroj (je určen v okně vlastností ve vlastnosti **Control Source** na kartě **Data** jako název pole datového zdroje).
- **Vypočítávané** ovládací prvky (ve vlastnosti **Control Source** na kartě **Data** je uveden vzorec pro výpočet hodnoty ovládacího prvku).

### Vyznačení jednoho ovládacího prvku

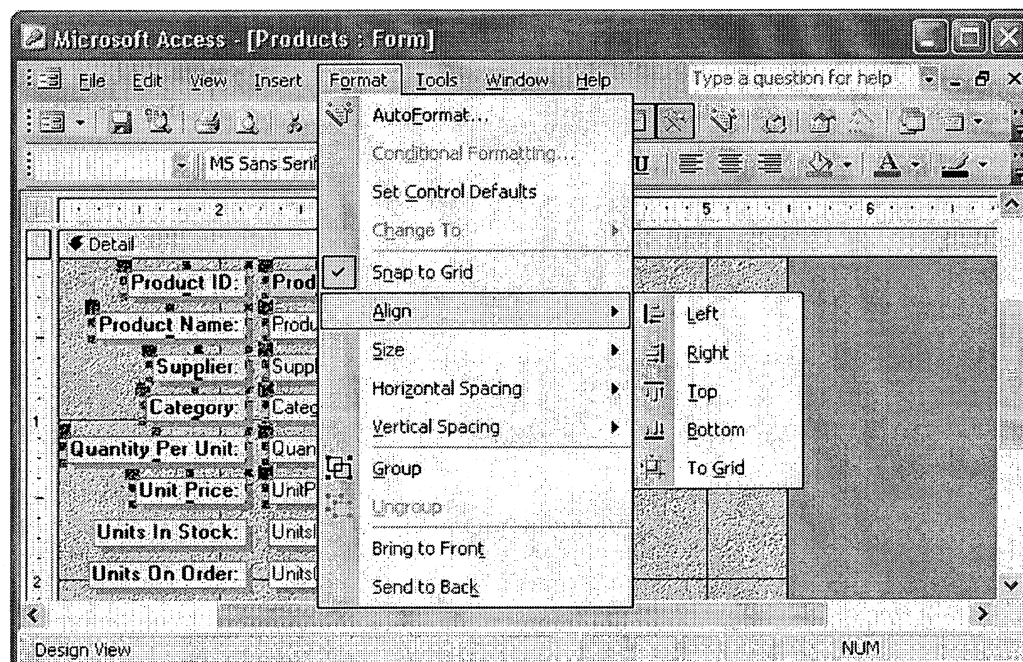
- Klepnutím na prvek.

### Vyznačení více ovládacích prvků najednou

- Na prvcích postupně klepeme se stisknutou klávesou **Shift**.
- Při zapnutém tlačítku **Select Objects** lze též kolem prvků opsat tahem myši obdélník.



Přichycení prvků k mřížce zapneme/vypneme příkazem **Format/Snap to Grid**. Po zapnutí se při pohybu a změně velikosti prvku tento automaticky přichycuje k podkladové mřížce formuláře. Doporučuje se mít tuto vlastnost zapnutou. Dočasné vypnutí provedeme tahem za hranu prvku se stisknutou klávesou **Ctrl**.

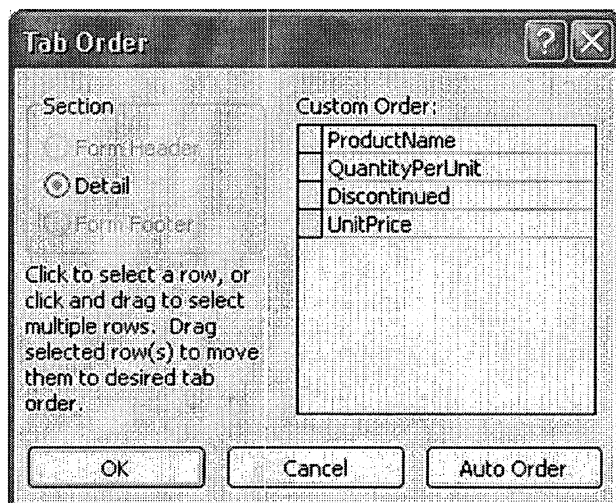


Mezery mezi vyznačenými ovládacími prvky měníme příkazy *Format/Vertical Spacing*, *Format/Horizontal Spacing*.

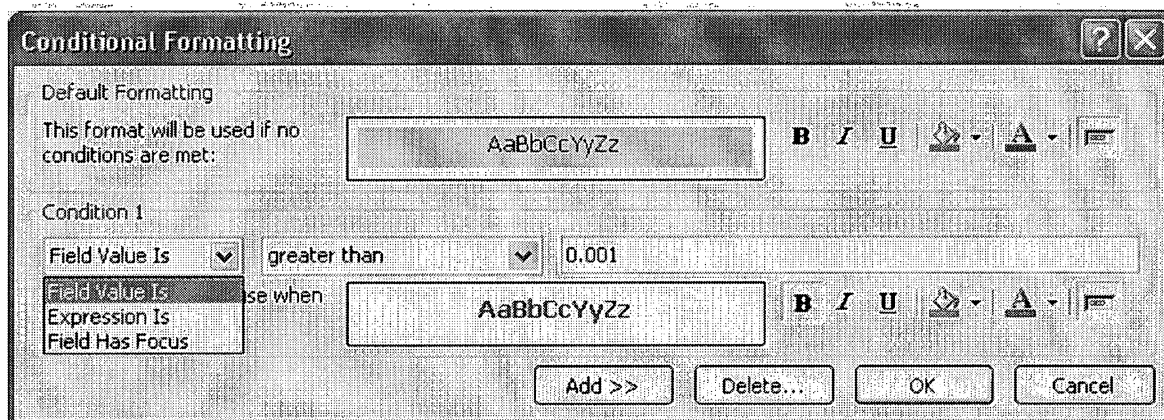


**Změna pořadí aktivace prvků ve formulářovém zobrazení klávesou Tab**

- Příkaz *View/Tab Order*.



# Podmíněné formátování



## Některé důležité ovládací prvky formuláře



### Textové pole (Text Box)



- Vkládáme-li na formulář textové pole ze soupravy nástrojů či ze seznamu polí, vloží se sem spolu s **popiskem (Label)** - text do popisku je přebrán z vlastnosti **Caption** daného pole nebo z názvu pole.

- Umístíme-li ukazatel myši na hranu textového pole, má tvar otevřené dlaně s roztaženými prsty. Nyní lze tahem myši posouvat obě dvě součásti prvku najednou (textové pole spolu s popiskem).

- Umístíme-li ukazatel myši na levé horní táhlo libovolného člena dvojice, má ukazatel tvar dlaně se vztyčeným ukazovákem. V tom případě posouváme jen tento člen dvojice.

- Chceme-li oddělit popisek přiřazený textovému oknu, vyznačíme jej klepnutím a **Ctrl+X** jej vyjmeme do schránky. Vyznačíme-li lištu oddílu formuláře, pak повеlem **Ctrl+V** vložíme neasociovaný popisek do tohoto oddílu. Vyznačíme-li samotné textové okno, pak повеlem **Ctrl+X** sdružíme (asociujeme) popisek tomuto oknu.

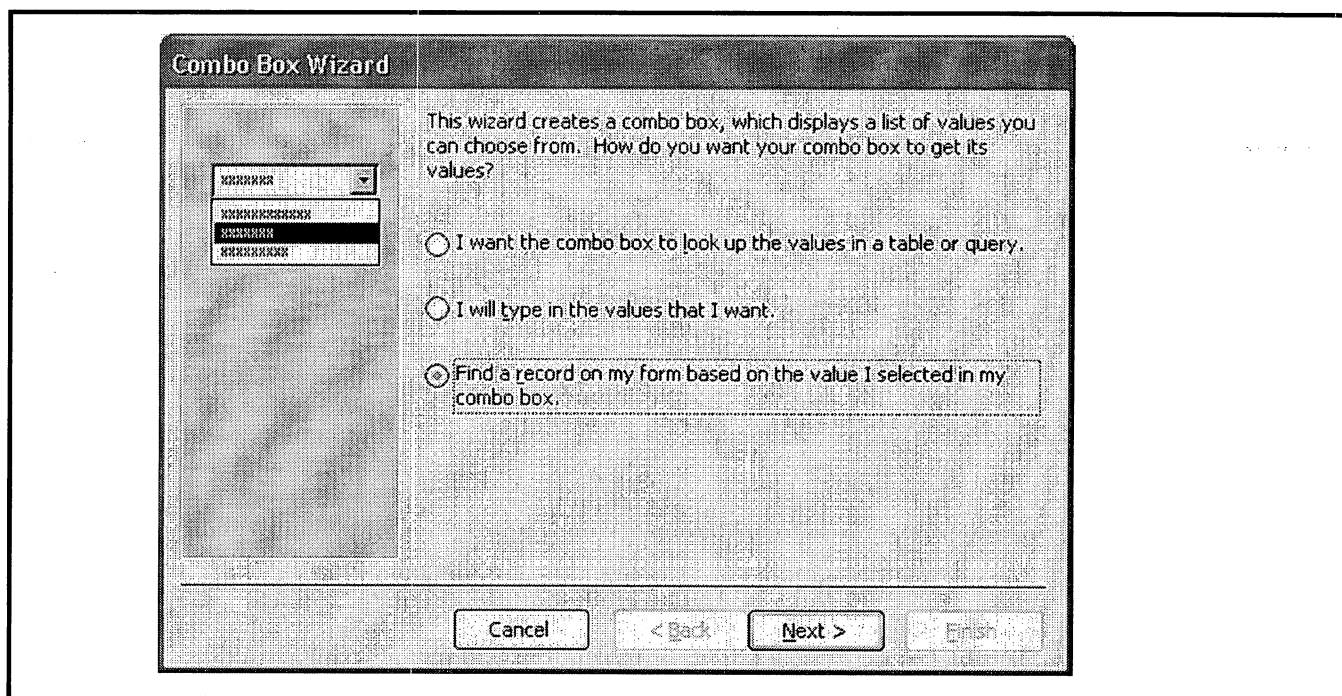
### Podmíněné formátování (*Format/Conditional Formatting*)

Hodnoty textových polí formulářů lze formátovat podmíněně. Jejich formát se potom může měnit v závislosti na

- obsahu tohoto pole (**hodnota pole je**)
- obsahu jiných polí (**výraz je**). V tomto případě zadáváme k vyhodnocení logický výraz (jeho hodnota je true/false).



# Seznamy a pole se seznamem



## Seznamy a pole se seznamem (List Box, Combo Box)



K ovládacímu prvku **List Box** či **Combo Box** je přiřazen průvodce, který pomůže specifikovat funkci seznamu.



Průvodce se aktivuje po vložení ovládacího prvku do formuláře, je-li zapnuto tlačítko **Control Wizards**.

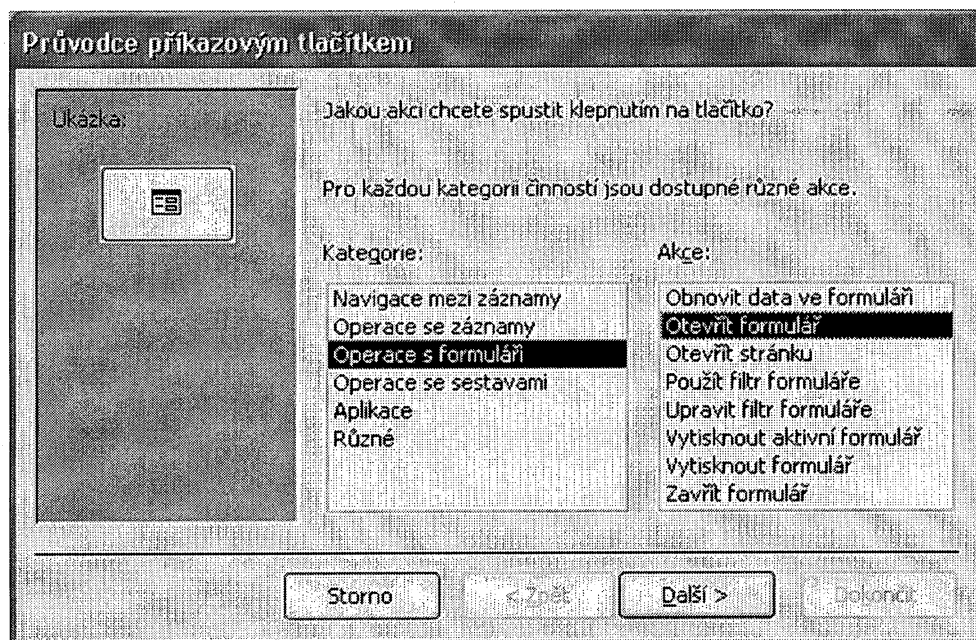
## Jak se liší seznam (List Box) od pole se seznamem (Combo Box)

Zabírá na obrazovce více místa, často je však výběr v něm přehlednější – vidíme okolní položky.

Nemá textové pole. U **pole se seznamem** se dají zápisem do textového pole vkládat nové položky. Toto vlastnost lze povolit/zakázat.

Může být konfigurován na výběr více položek najednou.

# Tlačítko

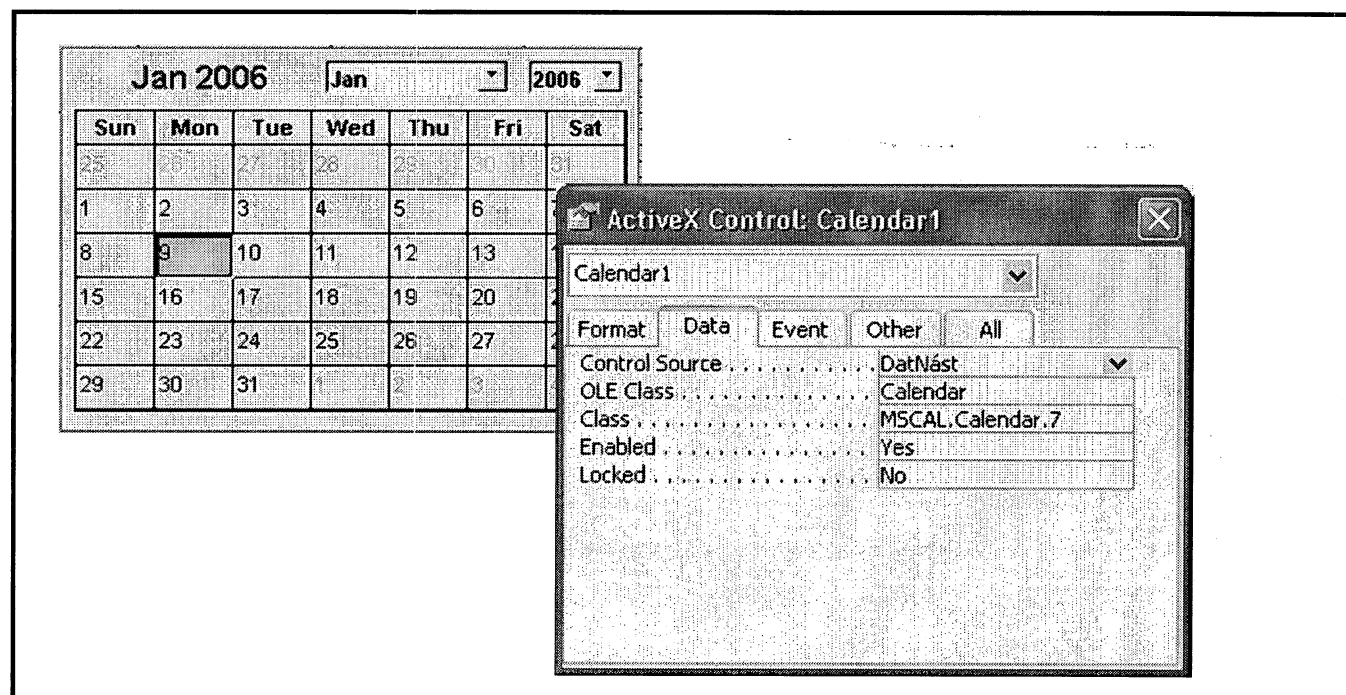


## Tlačítko (Command Button)



Je k němu opět k dispozici průvodce, který poskytuje celou škálu možností obsluhy události klepnutí na tlačítko.

# Kalendář



## Kalendář (Calendar)

Určíme-li pro tento ovládací prvek jako zdroj záznamů pole datového typu Date/Time, zobrazují se pak jeho hodnoty na kalendáři, který tvoří okno prvku. Z kalendáře lze i do pole sloužícího jako zdroj záznamů data zadávat.

### Jak získáme prvek Calendar:

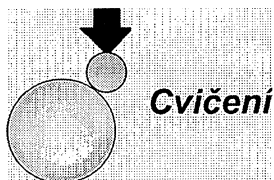


Přejdeme do návrhového zobrazení formuláře. Klepneme na tlačítko **More Controls** v **Toolboxu**. V nabídce tlačítka zvolíme položku **Calendar Control 11.0**.

### Jak napojíme prvek Calendar na datové pole:

V okně vlastností ovládacího prvku Calendar určíme jako zdroj ovládacího prvku pole z datového zdroje vázaného na formulář.

Podobně jako ovládací prvek Calendar funguje i prvek **Microsoft Date and Time Picker Control**. Najdeme ho opět mezi dalšími ovládacími prvky. Na formuláři zabírá méně místa (kalendář se rozbálí až klepnutím na šipku textového pole). Propojení s datovým polem formuláře nastavíme opět v datových vlastnostech prvku.



## Úkoly k procvičování:

1. Do formuláře **grySumaZaObjednávku** z předchozího příkladu doplníme celkovou sumu za všechny objednávky a pro každou objednávku vypočítáme procenta z celkového součtu:
  1. V návrhovém zobrazení formuláře vytvoříme tahem myši za dolní část lišty formuláře asi 1 cm vysoké zápatí.
  2. Ze soupravy nástrojů vložíme do zápatí **Text Box** a nazveme jej **txtCelkSuma**. Tento název vytvoříme zapsáním do řádku **Name** okna vlastností textového pole.
  3. Do vlastnosti **Control Source** na kartě **Data** okna vlastností textového pole zadáme vzorec **=Sum([CelkCena])**. Formát tohoto ovládacího prvku nastavíme na **Currency** (karta **Format** okna vlastností).
  4. Rozšíříme podkladovou mřížku formuláře směrem doprava a vložíme do ní nové textové pole. Jeho popisek umístíme do záhlaví formuláře. Do textového pole napíšeme vzorec **=[CelkCena]/[txtCelkSuma]** (vlastnost **Control Source** na kartě **Data**). Jeho vlastnost **Format** na kartě **Format** nastavíme na **Percent**.
  5. Upravíme velikost okna formuláře podle podkladové mřížky a formulář uložíme jako **frmSumaZaObjednávku**. Potom přepneme do formulářového zobra-

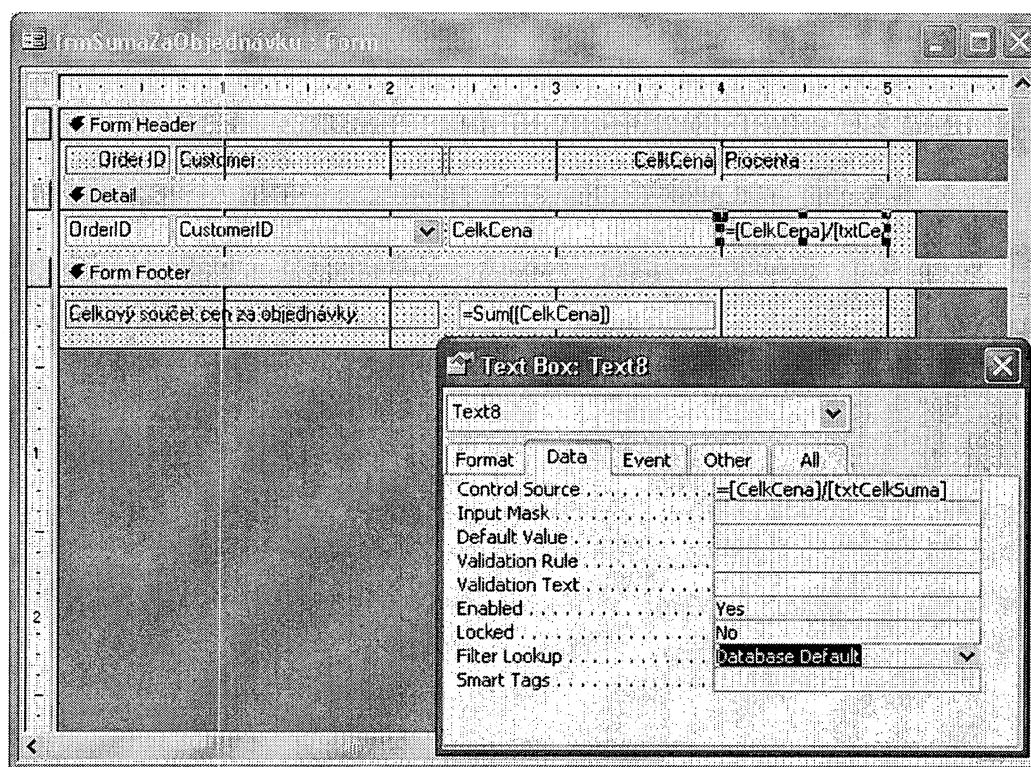
| Order ID | Customer               | CelkCena   | Procenta |
|----------|------------------------|------------|----------|
| 10248    | Wilman Kala            | \$440.00   | 0.03%    |
| 10249    | Tradição Hipermercados | \$1,863.40 | 0.14%    |
| 10250    | Hanari Carnes          | \$1,813.00 | 0.13%    |
| 10251    | Victuailles en stock   | \$670.80   | 0.05%    |
| 10252    | Suprêmes délices       | \$3,730.00 | 0.28%    |
| 10253    | Hanari Carnes          | \$1,444.80 | 0.11%    |

Celkový součet cen za objednávky: \$1,354,458.59

Record: 1 of 830

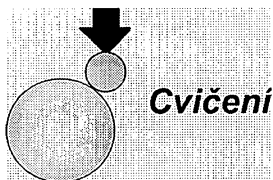
txtCelkSuma

## Úkoly k procvičování:



- 2 V zápatí formuláře **frmLidiT** vytvoříme vypočítávané textové pole, v němž se zobrazuje průměrný plat. Pro každý záznam uděláme pole počítající odchylku od průměrného platu:
1. Na kartě **Forms** okna databáze **ZamDěti** vyznačíme **frmLidiT** a klepnutím na tlačítko **Design** přejdeme do návrhového zobrazení formuláře **frmLidiT**.
  2. Tahem za dolní okraj lišty **Form Footer** vytvoříme cca 1 cm vysoké zápatí.
  3. Není-li zobrazena souprava nástrojů, zadáme příkaz **View/Toolbox**.
  4. Klepneme na tlačítko **Text Box** soupravy nástrojů a v zápatí vytvoříme tahem myši nevázané textové pole s odpovídajícím popisem.
  5. Poklepáním na hraně textového pole zobrazíme jeho okno vlastností. Zde na kartě **Other** vyplníme hodnotu vlastnosti **Name** jako **AvgPlat**.
  6. Do textového pole vepíšeme vzorec **=Avg([Plat])** a do popisku text **Průměrný plat**.
  7. Zvětšíme šířku okna a podkladové mřížky formuláře tak, aby se vpravo vešlo ještě jedno textové pole a toto pole sem vložíme. Jeho popisek přemístíme do záhlaví formuláře a vepíšeme do něj text **Odchylky od průměrného platu**.
  8. Do nového textového pole dáme vzorec **=([Plat]-[AvgPlat])**
- Pro obě textová pole nastavíme v okně **Properties** na kartě **Format** formát **Currency**. Je-li třeba, učiníme to i pro pole **Plat**.





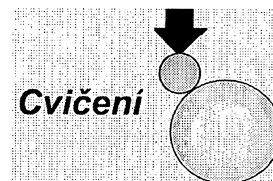
## Úkoly k procvičování:

- 3 V databázi **Northwind** vytvoříme z tabulky zákazníků formulář, který má v záhlaví **Combo Box** s názvy firem. Po vyznačení položky v seznamu se ve formuláři zobrazí podrobnosti o firmě zákazníků:

Vytvoříme sloupcový formulář (lze použít AutoForm). Do jeho záhlaví vložíme **Combo Box** a nastavíme jeho vlastnosti pomocí průvodce. Ze tří možností, které průvodce nabízí, volíme poslední.

Průvodce Combo Boxu provede dvě důležité akce.

- Zajistí, že po aktualizaci seznamu bude spuštěna procedura ve Visual Basicu, která k položce v seznamu vybrané najde odpovídající záznam.



## Úkoly k procvičování:

Pro ilustraci uvádíme text generované událostní procedury.

```
Private Sub Combo49_AfterUpdate()
```

```
    ' Find the record that matches the control.
```

```
    Dim rs As Object
```

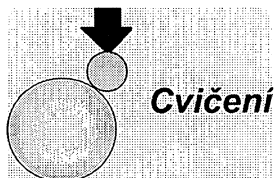
```
    Set rs = Me.Recordset.Clone
```

```
    rs.FindFirst "[CustomerID] = '" & Me![Combo49] & "'"
```

```
    If Not rs.EOF Then Me.Bookmark = rs.Bookmark
```

```
End Sub
```

V případě potřeby můžeme ještě položky v rozvíracím seznamu setřídít podle jména firmy namísto kódu. Učiníme tak změnou dotazu, jehož výsledek je zdrojem pro Combo Box. Ve vlastnostech prvku najdeme vlastnost **Row Source** a dotaz zeditujeme.



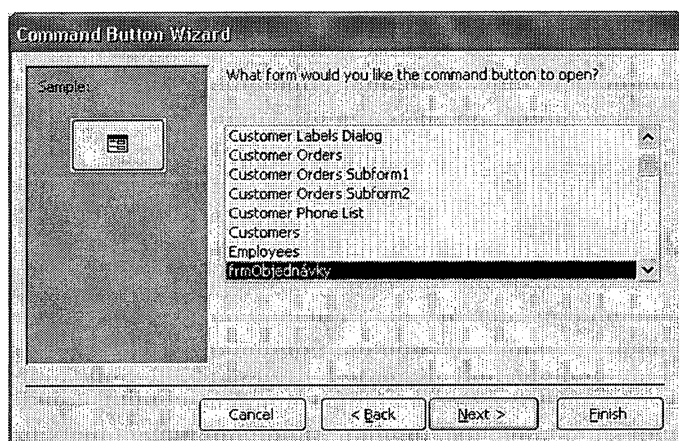
## Cvičení

### Úkoly k procvičování:

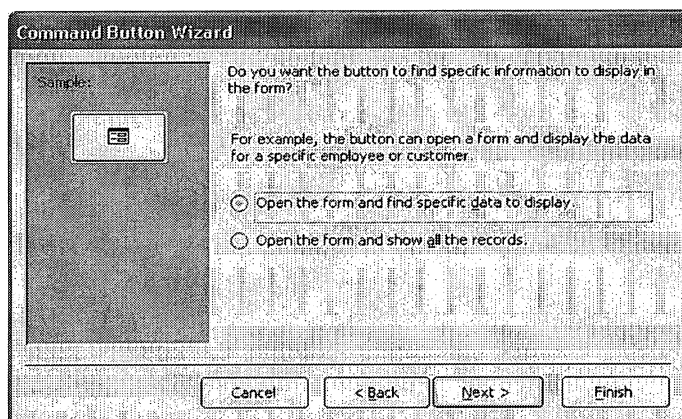
- 4 Vytvořme AutoForm: Tabular **frmObjednávky** zobrazující záznamy z tabulky **Orders**:

Do formuláře zákazníků z předchozího příkladu dodejme tlačítko, kterým zobrazíme objednávky zvoleného zákazníka.

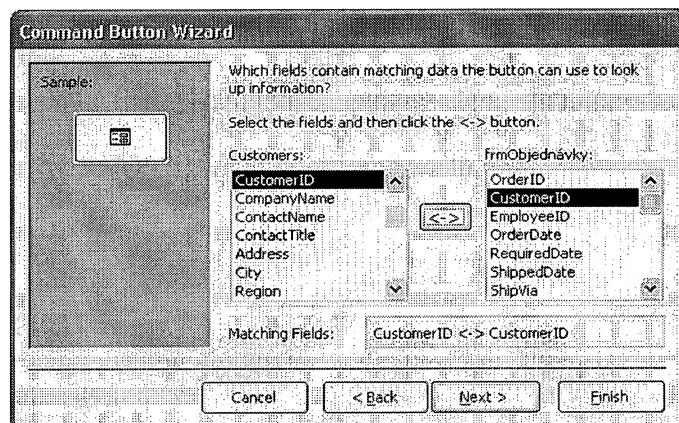
Po umístění tlačítka do zápatí formuláře zadáme, že se má tlačítkem otevřít formulář.



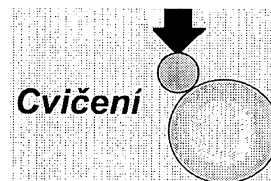
V dalším kroku volíme název formuláře.



Potom určíme, zda se mají zobrazit všechny, nebo jen určité záznamy tohoto formuláře. Volíme první možnost.







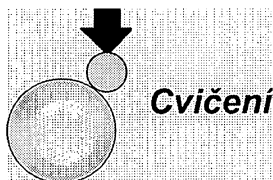
## Úkoly k procvičování:

Dále je třeba učit propojení polí formuláře se zákazníky a formuláře s objednávkami. V každém z těchto formulářů to je pole **CustomerID**.

Nakonec určíme, zda na tlačítku bude nápis, či obrázek.

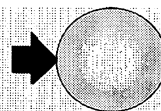
Průvodce opět obslouží událost klepnutí na tlačítko procedurou, která otevírá formulář s objednávkami, ale zobrazí v něm jen objednávky vybraného zákazníka.

- 5 V databázi **ZamDěti** uděláme k tabulce **Děti** tabulkový formulář **frmDěti**. Do formuláře **frmLidiS** dejme tlačítko, kterým se pro aktuální záznam ve formuláři otevře formulář **frmDěti** a zobrazí pouze děti příslušného zaměstnance.
  1. Formulář **frmDěti** uděláme k tabulce **Děti** stejným postupem, jako jsme dělali formulář **frmLidiT**.
  2. Otevřeme formulář **frmLidiS** v návrhovém zobrazení a uděláme v něm cca 1 cm zápatí.
  3. V soupravě nástrojů zkontrolujeme, zda je zapnutý Control Wizards. Ne-li, zapneme jej.
  4. V soupravě nástrojů klepneme na **Command Button** a vložíme tento objekt do zápatí **frmLidiS**.
  5. Spustí se průvodce příkazovým tlačítkem. V něm volíme **Form Operations/ Open Form**.
  6. Název otevíraného formuláře bude **frmDěti**.
  7. Dále volíme **Open the form and show all the records**.
  8. Definujeme vztah mezi poli primárního klíče z tabulky **Lidi** a **Rodič** z tabulky **Děti**.
  9. Potom volíme, zda na tlačítku bude nápis, či obrázek, a běh průvodce se ukončí.Vyzkoušíme funkci tlačítka a je-li třeba, navolíme vhodnou velikost obou formulářů.



## Úkoly k procvičování:

- 6 Ve formuláři **frmSumaZaObjednávku** databáze **Northwind** formátujeme pole obsahující procenta červeně, pokud je jeho hodnota větší než 0,001.
- Postup:
- Přejdeme do návrhového zobrazení pro **frmSumaZaObjednávku** a pole obsahující procenta vyznačíme
1. Vydáme příkaz **Format/Conditional Formatting**.
  2. V podmínce 1 volíme v prvním textovém okně **Field Value Is** a v seznamu vpravo vybereme operátor **greater than**.
  3. Do textového okna vpravo zapíšeme **0.001**.
  4. Formát textu volíme červený a tlačítkem **OK** vše potvrdíme.
- Vyzkoušíme chování formuláře ve formulářovém zobrazení
- 7 Ve formuláři **frmLidiT** databáze **ZamDěti** formátujeme červeně příjmení těch, kteří mají záporné odchylky platu od průměru.
- Postup:
1. Přejdeme do návrhového zobrazení pro **frmLidiT** a pojmenujeme pole zobrazující odchylky od průměrného platu jako **Odchylka**.
  2. Aktivujeme pole **Příjmení**.
  3. Vydáme příkaz **Format/Conditional Formatting**.
  4. V podmínce 1 volíme v prvním textovém okně **Expression Is** a do druhého textového okna píšeme vzorec **[Odchylka]<0**
  5. Formát textu volíme červený a tlačítkem **OK** zavřeme okno **Conditional Formatting**.
- Vyzkoušíme chování formuláře ve formulářovém zobrazení.



- Nejčastěji používanými ovládacími prvky jsou Text Box a Label, Combo Box a Check Box.
- Text Box můžeme použít pro výpočet nové položky, která se bude zobrazovat ve formuláři.
- Vkládáme-li List Box nebo Combo Box z Toolboxu, je výhodné použít průvodce (Control Wizards).
- Průvodce použijme také při vkládání Command Button, umožní snadno nastavit požadovanou akci.



420 100

100 100

420 100

100 100



# Podformuláře

## *Přehled témat*

- formulář s podformulářem
- vypočítávané položky v podformulářích
- závislosti objektů a dědičnost vlastností

# Formulář s podformulářem

Suppliers

Supplier ID: 8 Company Name: Specialty Biscuits, Ltd. Country: UK

frmVýrobky podformulář

|   | Product Name               | Quantity Per Unit    | Unit Price | Units In Stock | Units On Order |
|---|----------------------------|----------------------|------------|----------------|----------------|
| ▶ | Teatime Chocolate Biscuits | 10 boxes x 12 pieces | \$9.20     | 25             | 0              |
|   | Sir Rodney's Marmalade     | 30 gift boxes        | \$81.00    | 40             | 0              |
|   | Sir Rodney's Scones        | 24 pkgs. x 4 pieces  | \$10.00    | 3              | 40             |
|   | Scottish Longbreads        | 10 boxes x 8 pieces  | \$12.50    | 6              | 10             |
| * |                            |                      | \$0.00     | 0              | 0              |

Record: 1 of 4

Record: 8 of 29

Toolbox

Subform/Subreport



## Podformuláře (Subforms)

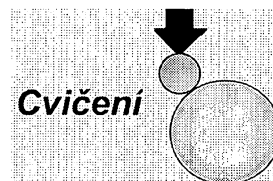
Podformuláře umožňují realizovat ve formuláři vztah 1:N. Záznamy tabulky na straně jedna se zobrazují ve formuláři, záznamy tabulky na straně N relace se zobrazují v podformuláři, který funguje jako ovládací prvek formuláře. Po vytvoření je však podformulář samostatným formulářem, jako každý jiný formulář. Naopak, z každého formuláře zobrazujícího záznamy tabulky na straně N lze udělat podformulář formuláře se záznamy tabulky na straně jedna relace. Podformulář má ve formuláři vlastní navigační prvek pro práci se záznamy.

Podformulář děláme ve formuláři pomocí prvku **Subform/Subreport** ze soupravy nástrojů. Tento prvek má vestavěného průvodce, který funkci podformuláře pomůže nastavit. Průvodce často nabízí pro propojení polí formuláře a podformuláře více variant. Je-li však mezi výchozími tabulkami relace 1:N, respektuje vždy první nabízená varianta tuto relaci.

poznámka

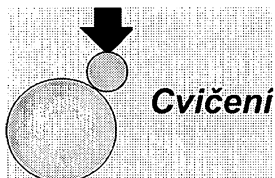


Děláme-li formulář z více tabulek či dotazů pomocí průvodce formulářem, nabízí tento často variantu formulář s podformulářem. Též automatické formuláře z dotazů aplikovaných na více tabulek dělají často formuláře s podformuláři.



## Úkoly k procvičování:

- 1 Vytvořme podle úvodního obrázku v kapitole Podformuláře formulář s podformulářem. Ve formuláři jsou pole **SupplierID**, **CompanyName**, **Country** z tabulky **Suppliers**. Formulář vytvořme v návrhovém zobrazení. Do jeho těla přidejme prvek podformulář či podsestava a jeho funkci nadefinujme pomocí průvodce.  
Postup:  
Vytvoříme **AutoForm: Columnar** pro tabulku **Suppliers** a uložíme jej jako **frmDodavateléVýrobky**.  
V jeho návrhovém zobrazení odstraníme nadbytečná pole z těla formuláře a zbylá pole umístíme do horní části těla. Upravíme též rozměry okna (viz obrázek na předešlé stránce).  
Přesvědčíme se, zda je zapnutá volba **Control Wizards** na soupravě nástrojů. Klepneme na tlačítko **Subform/Subreport** a v těle formuláře vytvoříme tahem myši podformulář. Spustí se průvodce podformulářem.  
V něm volíme, že použijeme existující tabulky a dotazy, tabulku **Products**. Potom zvolíme do podformuláře příslušná pole z této tabulky (viz zmíněný obrázek). Je potřeba zvolit také propojovací pole **SupplierID**.  
Průvodce dále navrhne správné propojení formuláře s podformulářem.  
V závěru běhu průvodce volíme název podformuláře **frmVýrobky podformulář** a vše uložíme.  
V okně databáze se přesvědčíme, že vznikly dva formuláře **frmDodavateléVýrobky** a **frmVýrobky podformulář**.  
Zobrazíme-li **frmVýrobky podformulář** v návrhovém režimu a na kartě **Format** okna vlastností změníme výchozí zobrazení, pak se toto nově zvolené zobrazení bude používat i v případě, že zobrazíme formulář **frmDodavateléVýrobky** s tímto podformulářem.



## Úkoly k procvičování:

- 2 Vytvořme pomocí průvodce formulářem formulář pro zobrazení zákazníků. Tento formulář má podformulář s objednávkami každého zákazníka. Ke každé objednávce je v hlavním formuláři ještě jeden podformulář s rozpisem objednávek.

**Form Wizard**

How do you want to view your data?

- ☒ by Customers
- ☐ by Orders
- ☐ by Order Details

CompanyName, City, Country

Orders\_OrderID, OrderDate, ShipName, ShipCity

Order Details\_OrderID, ProductID, UnitPrice, Quantity

☒ Form with subform

**Form Wizard**

What layout would you like for each subform?

☒ Tabular

☐ Datasheet

☐ PivotTable

☐ PivotChart

☒ Tabular

☒ Datasheet

☐ PivotTable

☐ PivotChart

**frmZákazníci s podformulářem**

Company Name: Alfreds Futterkiste

City: Berlin

Country: Germany

Orders:

| Order ID | Order Date  | Ship Name           | Ship City |
|----------|-------------|---------------------|-----------|
| 10643    | 25-Aug-1997 | Alfreds Futterkiste | Berlin    |
| 10692    | 03-Oct-1997 | Alfreds Futterkiste | Berlin    |

Record: 1 of 6

Order Details:

| Order ID | Product           | Unit Price | Quantity |
|----------|-------------------|------------|----------|
| 10643    | Rössle Sauerkraut | \$45.60    | 15       |
| 10643    | Chartreuse verte  | \$18.00    | 21       |
| 10643    | Spearsild         | \$12.00    | 7        |

Record: 1 of 3

Record: 1 of 91

- 3 Do podformuláře pro rozpis objednávek z předchozího příkladu přidejme pole obsahující součin jednotkové ceny a množství.

**Order Details Subform: Form**

Form Header

Detail

| Order ID   | OrderID   |
|------------|-----------|
| Product    | ProductID |
| Unit Price | UnitPrice |
| Quantity   | Quantity  |

Form Footer

Součet za objednávku:  $=\text{Sum}([\text{UnitPrice}] * [\text{Quantity}])$

**Text Box: Text8**

Text8

Format Data Event Other All

Control Source:  $=\text{Sum}([\text{UnitPrice}] * [\text{Quantity}])$

Input Mask:

Default Value:

Validation Rule:

Validation Text:

Enabled: Yes

Locked: No

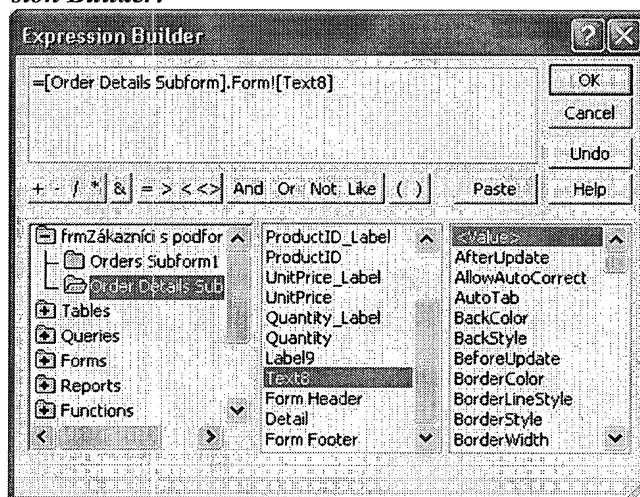
Filter Lookup: Database Default

Smart Tags:



## Úkoly k procvičování:

Dále vytvoříme v zápatí hlavního formuláře pole, zobrazující součet za objednávku celkem. Pro vytvoření tohoto pole dáme do zápatí podformuláře pro rozpis objednávek pomocné pole sčítající jednotkovou cenu násobenou množstvím. Pomocné pole nebude viditelné, protože se podformulář pro rozpis objednávek zobrazuje v režimu datového listu. Na toto pole se potom odkážeme z našeho textového pole umístěného v zápatí hlavního formuláře, čímž obsah pomocného pole zviditelníme. K vytvoření odkazu s výhodou poslouží **Expression Builder**.



Výsledný formulář vypadá takto:

**frmZákazníci s podformulářem**

Company Name:

City:

Country:

Orders

| Order ID | Order Date  | Ship Name           | Ship City |
|----------|-------------|---------------------|-----------|
| 10643    | 25-Aug-1997 | Alfreds Futterkiste | Berlin    |
| 10692    | 03-Oct-1997 | Alfreds Futterkiste | Berlin    |

Record: 1 of 6

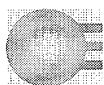
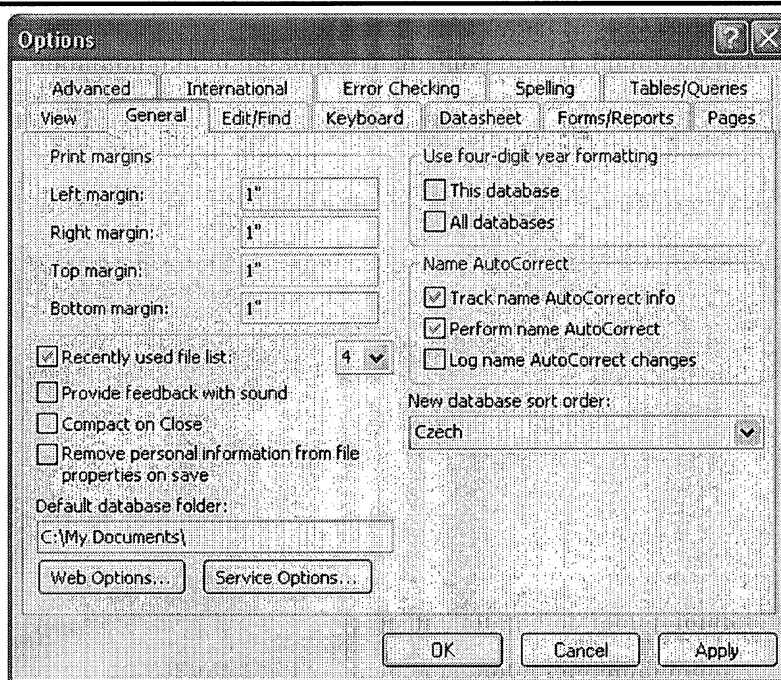
Order Details

| Order ID | Product           | Unit Price | Quantity |
|----------|-------------------|------------|----------|
| 10643    | Rössle Sauerkraut | \$45.60    | 15       |
| 10643    | Chartreuse verte  | \$18.00    | 21       |

Součet za objednávku celkem:

Record: 1 of 91

# Závislosti objektů



## Závislosti objektů

Okno relací zobrazuje přehledně **závislosti tabulek** v databázi Access. Pokud byste chtěli některou tabulku odstranit nebo rozdělit, máte možnost prohlédnout si i tabulky související a rozhodnout o bezpečných úpravách.

V aplikaci Access 2003 lze již zobrazit **informace o vzájemných závislostech většiny databázových objektů**. Možnost zobrazit seznam objektů, které využívají určitý objekt, pomáhá při dlouhodobém udržování databáze a brání vzniku chyb souvisejících s chybějícími zdroji dat.

## Objekty, pro které nelze zobrazit závislosti

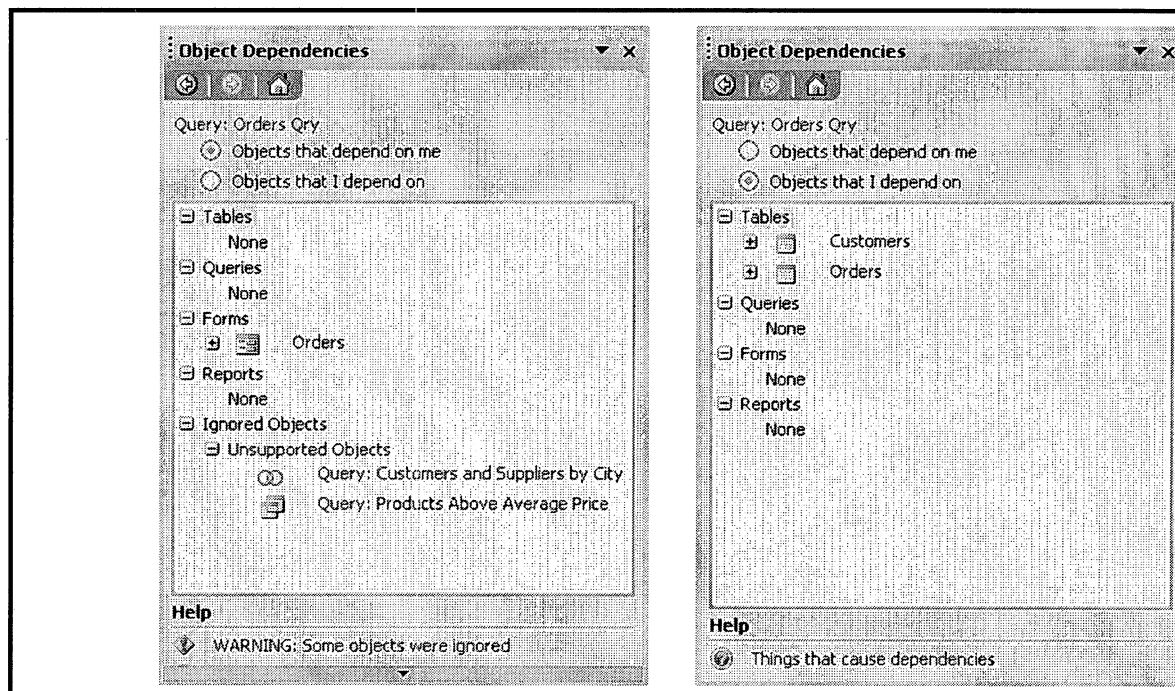
- datové stránky, makra a moduly v databázi
- objekty nepodporované funkcí automatických oprav názvů, jako jsou akční dotazy a dotazy jazyka SQL včetně sjednocovacích dotazů, definičních dotazů, přidávacích dotazů a poddotazů.

Informace o závislosti nejsou k dispozici ani tehdy, nemáte-li oprávnění k otevření objektu v návrhovém zobrazení.

## Zobrazení závislostí

Informace o závislostech lze zobrazit pouze v případě, že je povoleno **sledování informací o automatických opravách názvů (Track name AutoCorrect info)**. To nastavíte volbou **Tools/Options**, na kartě **General** v sekci **Name AutoCorrect**. Samotné provádění automatických oprav názvů (**Perform name AutoCorrect**), které nastavujete na téže kartě, může zůstat vypnuté.

# Závislosti objektů



Samotné zobrazení závislostí provedete z místní nabídky zvoleného objektu (dotazu, formuláře, tabulky) volbou **Object Dependencies**.



V okně **Object Dependencies** přepínáme mezi volbami

- Objects that depend on me
- Objects that I depend on.

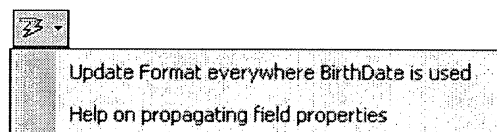
Lze zobrazit celkem čtyři úrovně podrobností.

## Dědičnost vlastností

Kromě automatické opravy názvů polí a objektů v jiných objektech, které je využívají, nabízí Access 2003 i automatickou aktualizaci zděděných vlastností.

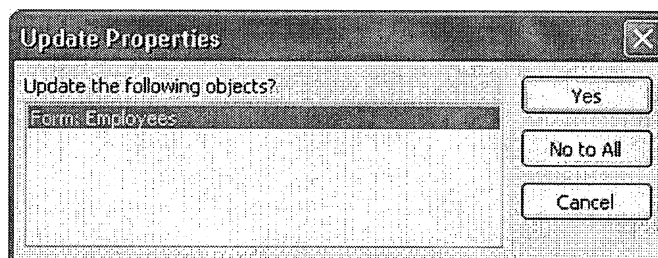
Vlastnosti, které se u různých ovládacích prvků dědí při přetažení pole ze seznamu polí do formuláře nebo sestavy, můžete najít v nápovědě. Tak například textové pole "zdědí" z vlastností pole formát.

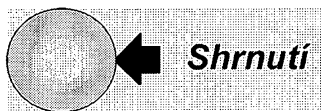
Po úpravě zvolené vlastnosti například u pole v tabulce



- rozbalíte seznam značky vedle řádku s vlastností
- vyberete **Update Format ....**

a v nabídnutém seznamu souvisejících objektů vyberete objekty k aktualizaci.





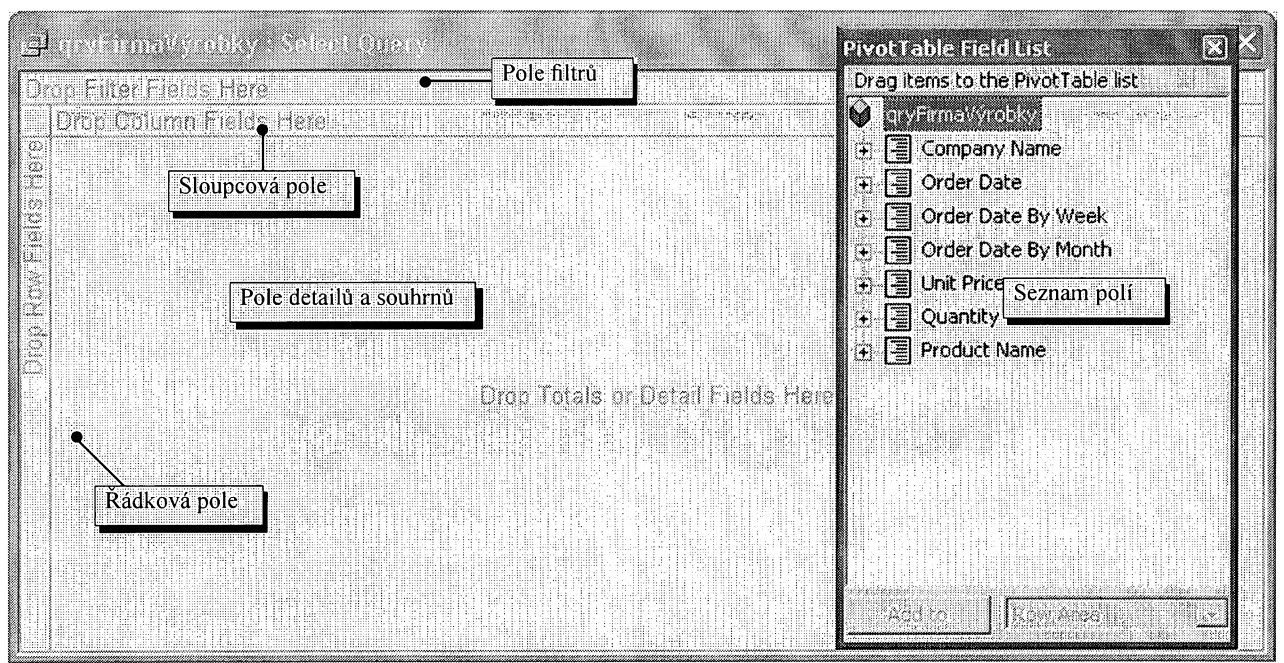
- Formulář s podformulářem umožňuje současné zobrazení záznamů z tabulek, které jsou ve vztahu 1:N.
- Podformulář vložíme do formuláře pomocí ovládacího prvku **Subform/Subreport**. Nezapomeňme před vkládáním zapnout průvodce ovládacími prvky.
- Pokud k tabulce na straně 1 použijete funkci AutoForm, vytvoří se formulář i s podformulářem pro tabulku na straně N.
- Pokud použijete průvodce formulářem a vyberete pole z tabulek spojených relací 1:N, průvodce sám nabídne zobrazení pomocí formuláře s podformulářem.

# Kontingenční tabulky

## *Přehled témat*

- vytvoření kontingenční tabulky
- souhrny a filtrování dat
- nové vypočítávané položky
- kontingenční graf

# Kontingenční tabulky Access 2003



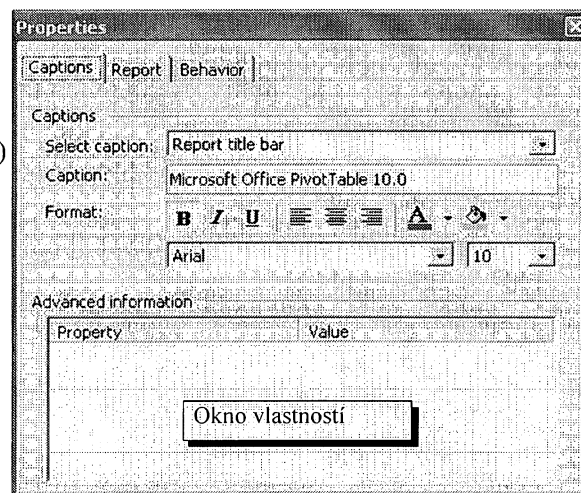
## Kontingenční tabulky (Pivot Tables) Access 2003

### Charakteristika

Kontingenční tabulky umožňují udělat klasickou tabulku s názvy řádků i sloupců, která obsahuje souhrnné informace získané z databázových tabulek a dotazů. Jejich obrovskou výhodou je snadné sestrojení a snadná modifikace. Formou kontingenční tabulky lze zobrazit tabulky, dotazy a formuláře databáze.

Zobrazení kontingenční tabulky (Pivot Table View) obsahuje čtyři základní oblasti, které jsou popsány na obrázku

- Oblast pro řádková pole (Row Fields)
- Oblast pro sloupcová pole (Column Fields)
- Oblast pro pole detailů a souhrnů  
(Totals or Detail Fields)
- Oblast pro pole filtrů (Filter Fields)



V každé této oblasti může být i více polí datového zdroje.

## Postup tvorby



### Přechod do Pivot Table View

- Otevřeme datový zdroj (tabulku či dotaz), z něhož chceme udělat kontingenční tabulku. V nabídce **View** volíme **Pivot Table View**.
- Nebo v dialogovém okně **New Form** volíme **AutoForm: Pivot Table** nebo **Pivot Table Wizard**.

V zobrazení kontingenční tabulky otevřeme okno **Pivot Table Field List (View/Field List)**. Toto okno obsahuje pole datového zdroje tabulky. Tahem myši umístíme zvolené pole ze seznamu polí do příslušné základní oblasti kontingenční tabulky. V každé této oblasti může být více polí.

Do oblasti pro sloupcová pole umístíme první pole, podle kterého chceme souhrnná data seskupovat (první úhel pohledu na souhrnná data). Do oblasti pro řádková pole umístíme druhé pole, podle kterého chceme souhrnná data seskupovat (druhý úhel pohledu na souhrnná data). Do oblasti součtů umístíme pole, v němž agregujeme data (počítáme v něm průměr, součet apod.). Do oblasti pro filtry umístíme pole, podle něhož filtrujeme záznamy, které vůbec budou v kontingenční tabulce zpracovány (agregovány).



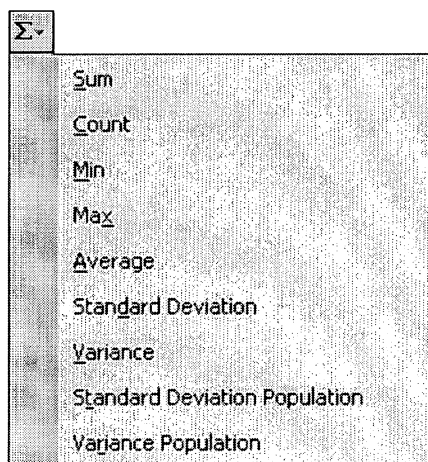
## Změna struktury kontingenční tabulky

- Umístěné pole lze také přemístit do jiné základní oblasti kontingenční tabulky. Děláme to tahem myši za záhlaví (obsahuje název pole) ze zdrojové do cílové oblasti kontingenční tabulky.
- Pole lze odstranit z tabulky tahem za záhlaví mimo tabulku.
- Pole lze přidat do základní oblasti tabulky tahem z okna **Pivot Table Field List**.

## Získání souhrnů v kontingenční tabulce

V oblasti pro pole detailů a souhrnů získáme nejdříve detaily – hodnoty uložené v záznamech tohoto pole. Pro získání souhrnů

- klepneme do buňky pole v oblasti detailů a souhrnů, pro které chceme získat souhrny
- rozbalíme nabídku tlačítka **AutoCalc** a volíme z ní vhodnou souhrnnou funkci. V tabulce obdržíme jak původní detaily, tak i nové souhrny.



- chceme-li skrýt detaily, vyznačíme příslušnou část oblasti detailů a provedeme příkaz **Hide Details** z místní nabídky (příkaz lze provést pouze pro jedno, či všechna pole v oblasti detailů)
- chceme-li zobrazit detaily, uděláme stejným způsobem příkaz **Show Details**
- skrývat a zobrazovat podrobnosti pro řádky a sloupce tabulky můžeme i tlačítky se znaménkem mínus a plus umístěnými v záhlaví řádku/sloupce.



## Úkoly k procvičování:

- 1 Sestavme kontingenční tabulku, z níž vyčteme pro zadanou firmu množství výrobku objednaného v určitém roce.

Postup práce:

Nejdříve vytvoříme podkladový dotaz pro kontingenční tabulku

| Field:    | CompanyName                         | OrderDate                           | UnitPrice                           | Quantity                            | ProductName                         |
|-----------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Table:    | Customers                           | Orders                              | Order Details                       | Order Details                       | Products                            |
| Sort:     |                                     |                                     |                                     |                                     |                                     |
| Show:     | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Criteria: | Like [Název firmy] & '*%*'          |                                     |                                     |                                     |                                     |
| or:       |                                     |                                     |                                     |                                     |                                     |

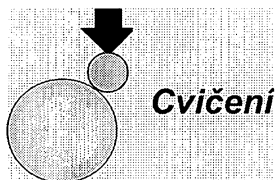
Potom buď v návrhovém režimu dotazu přepneme do zobrazení kontingenční tabulky, nebo dotaz zavřeme a kontingenční tabulku uděláme jako samostatný formulář.

This wizard automatically creates a form in PivotTable view.

Choose the table or query where the object's data comes from: qryFirmaVýrobky

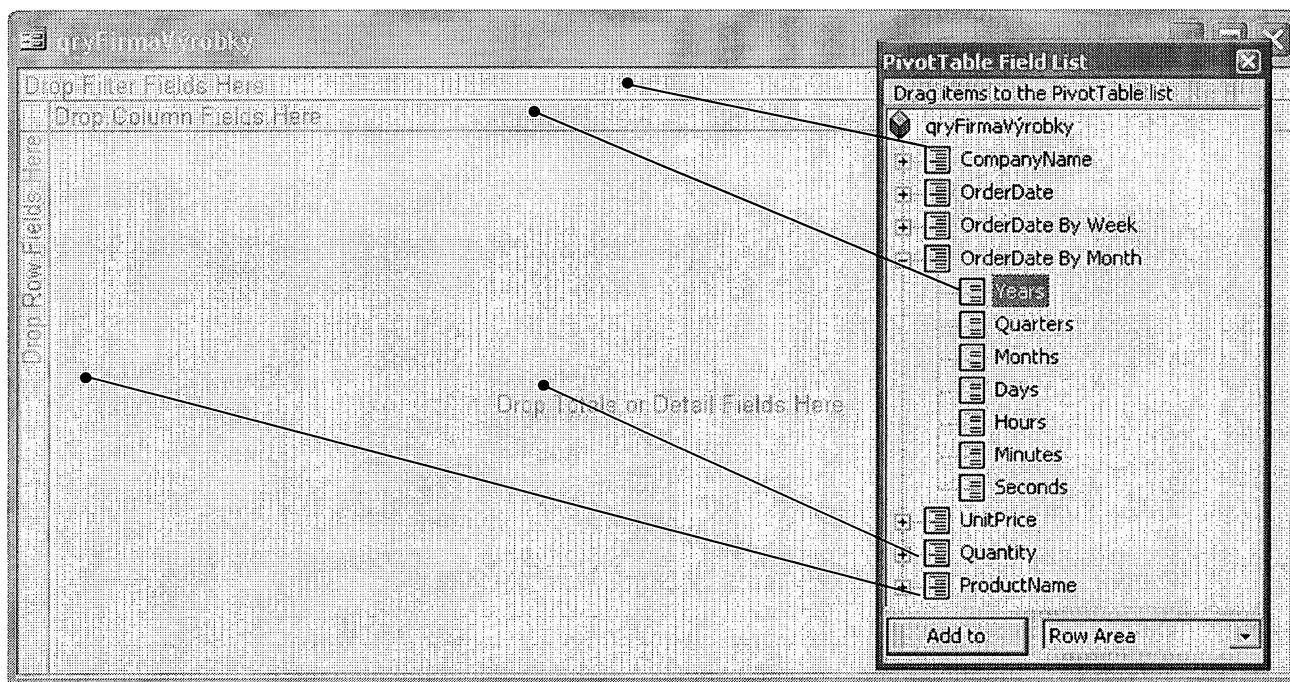
- Design View
- Form Wizard
- AutoForm: Columnar
- AutoForm: Tabular
- AutoForm: Datasheet
- AutoForm: PivotTable**
- AutoForm: PivotChart
- Chart Wizard
- PivotTable Wizard

OK Cancel



## Úkoly k procvičování:

V zobrazení kontingenční tabulky otevřeme okno **Seznam polí**. Z něj přesuneme příslušná pole do oblastí tabulky, v nichž je potřebujeme.



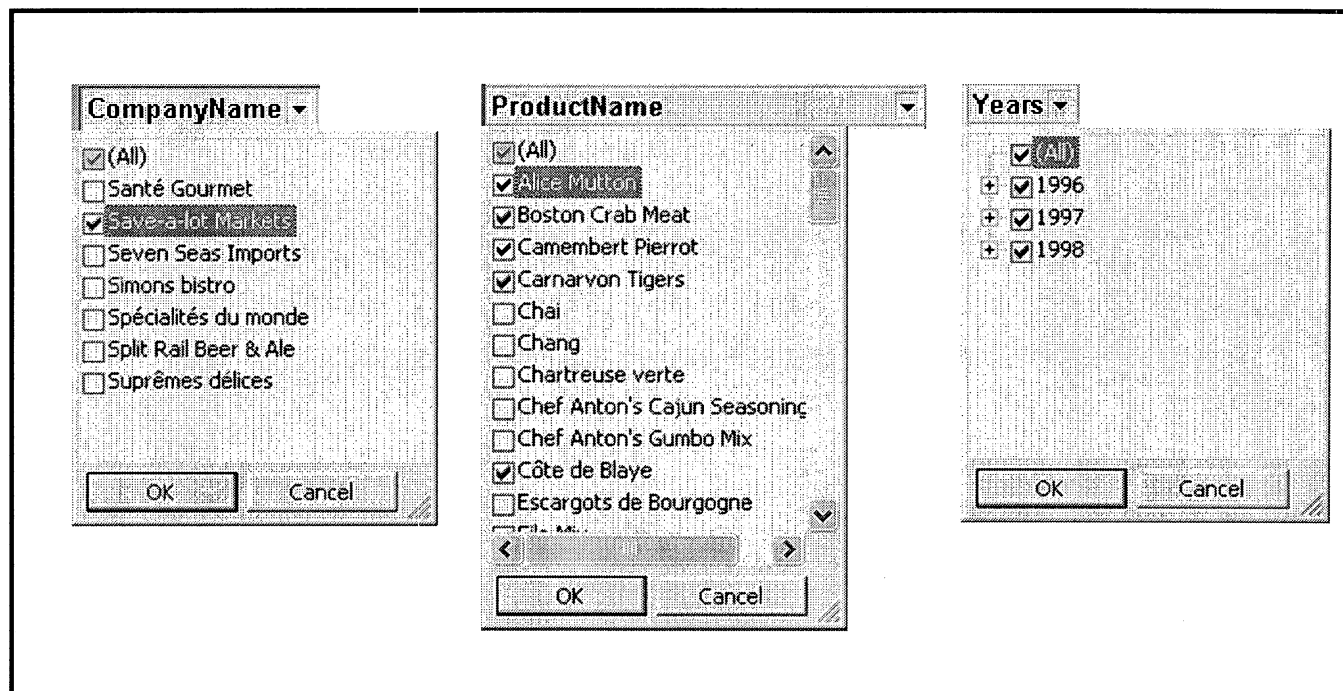
Získáme tabulku bez souhrnů v oblasti detailů.

| Company Name      |          | Years    |          |      |  | Grand Total |
|-------------------|----------|----------|----------|------|--|-------------|
| All               |          | 1996     | 1997     | 1998 |  |             |
| Product Name      | Quantity | Quantity | Quantity | No T |  |             |
| Alice Mutton      | 40       | 100      | 77       |      |  |             |
| Boston Crab Meat  |          | 27       | 25       |      |  |             |
| Camembert Pierrot | 80       | 40       | 45       |      |  |             |
| Carnarvon Tigers  | 20       | 30       |          |      |  |             |
| Chai              | 40       | 8        | 6        |      |  |             |
|                   |          | 21       |          |      |  |             |
|                   |          | 5        | 80       |      |  |             |
|                   |          |          | 45       |      |  |             |
|                   |          |          | 40       |      |  |             |
|                   |          |          | 10       |      |  |             |

### Získání souhrnů

Klepeme do pole **Quantity** v oblasti detailů a z nabídky tlačítka **AutoCalc** volíme příkaz **Sum**. Potom skryjeme podrobnosti pro množství.

## Filtrování dat v kontingenční tabulce



### Filtrování dat v kontingenční tabulce

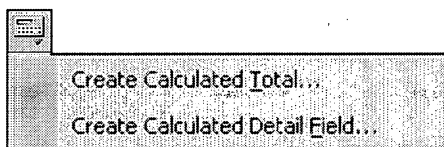
V záhlaví oblastí pro řádky, sloupce a filtry lze rozbalovat dialogová okna, v nichž lze volit podmínky pro zobrazení/nezobrazení určité položky.

Kontingenční tabulka z výše uvedeného příkladu má tento tvar (pro filtrování navolené výše):

| CompanyName ▼      |                 |                 |                 |
|--------------------|-----------------|-----------------|-----------------|
| Save-a-lot Markets |                 |                 |                 |
| Years ▼            |                 |                 |                 |
| ☑ 1997             |                 |                 | ☑ 1998          |
| + -                |                 |                 | + -             |
| ProductName ▼      | Sum of Quantity | Sum of Quantity | Sum of Quantity |
| Alice Mutton       | 127             | 77              | 204             |
| Boston Crab Meat   | 82              |                 | 82              |
| Camembert Pierrot  | 70              | 45              | 115             |
| Grand Total        | 279             | 122             | 401             |



## Počítání nových sloupců v kontingenční tabulce



Z nabídky tlačítka **Calculated Totals and Fields** lze vydávat příkazy pro výpočet nových sloupců

- Ze souhrnných polí
- Z polí podrobných dat – detailů.

Na následujícím obrázku je výpočítaný souhrnný sloupec **Predikce množství**. Jeho hodnoty jsou o 10% vyšší, než ve sloupci **Sum of Quantity**.

| CompanyName       | Years | Predikce množství | Sum of Quantity |
|-------------------|-------|-------------------|-----------------|
| Alice Mutton      | 1997  | 139.7             | 127             |
| Boston Crab Meat  | 1997  | 90.2              | 82              |
| Camembert Pierrot | 1997  | 77                | 70              |
| Grand Total       | 1997  | 306.9             | 279             |

## Zobrazení několika prvních či posledních položek

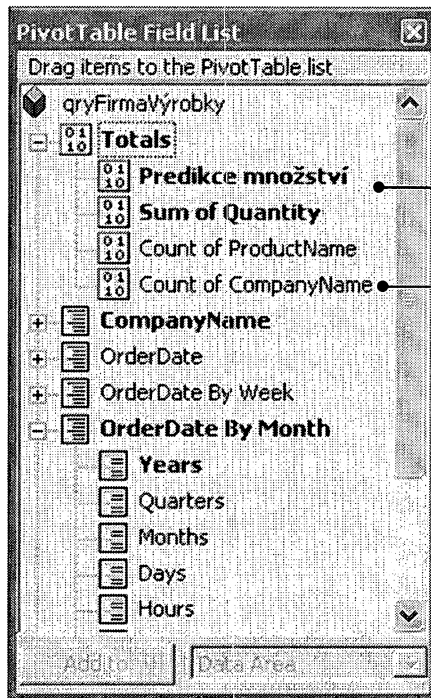
| Firma             | Roky | NázevVýrobku | Součet z Množství | Součet z Množství | Součet z Množství |
|-------------------|------|--------------|-------------------|-------------------|-------------------|
| Alice Mutton      | 1997 | 127          | 77                |                   |                   |
| Boston Crab Meat  | 1997 | 82           |                   |                   |                   |
| Camembert Pierrot | 1997 | 70           | 45                |                   |                   |
| Celkový součet    | 1997 | 279          | 122               |                   |                   |

poznámka



Odstraníme-li některé souhrnné údaje z kontingenční tabulky, zůstávají ještě v seznamu polí, odkud je lze zpět do tabulky přemístit. Z tohoto seznamu je však můžeme odstranit (příkaz **Delete** místní nabídky). Potřebujeme-li je potom, nutno provést jejich nový výpočet.

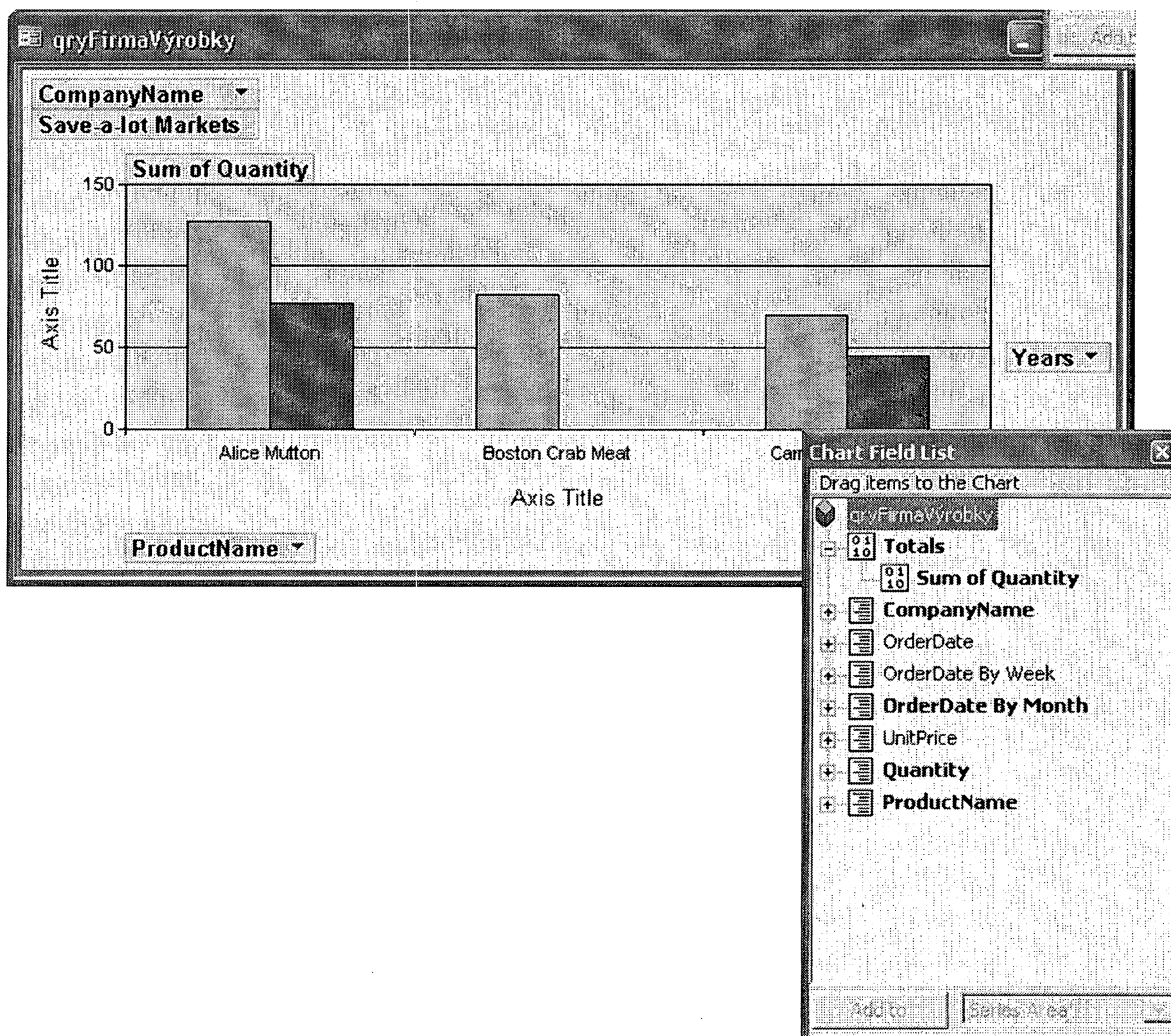


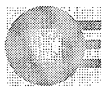


Tato souhrnná pole jsou v tabulce (tučný text)

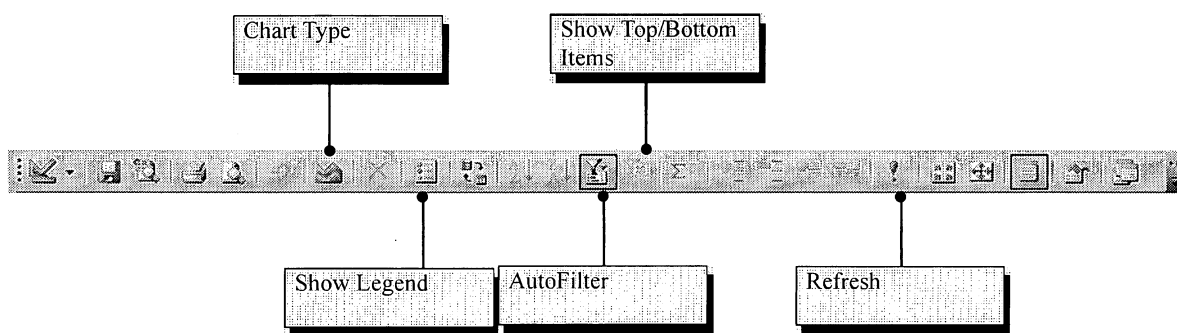
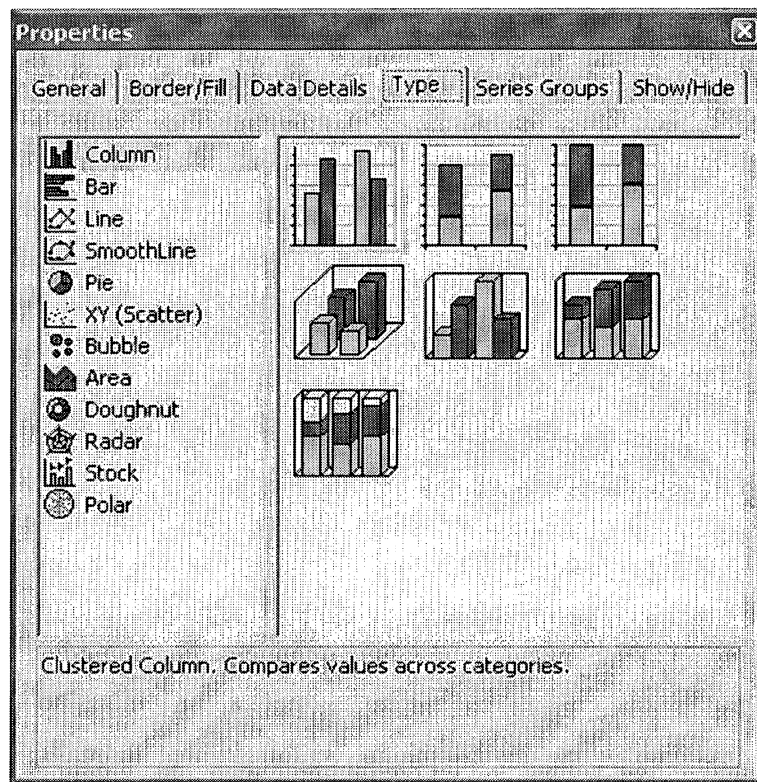
Tato souhrnná pole byla odstraněna (není tučné), ale je možné je tam myší opět umístit

## Kontingenční graf





Podobným způsobem jako kontingenční tabulky lze dělat i kontingenční grafy. Máme-li hotovou kontingenční tabulku, lze ji převést na graf (*View/PivotChart View*). V grafu lze podobně jako v kontingenční tabulce pracovat se seznamem polí (přidávání a odebírání polí). Změny, které v grafu takto uděláme, se projeví v kontingenční tabulce a naopak. Pomocí místní nabídky jednotlivých součástí grafu lze tento formátovat. Též v okně vlastností lze volit typ grafu a další formátovací akce.

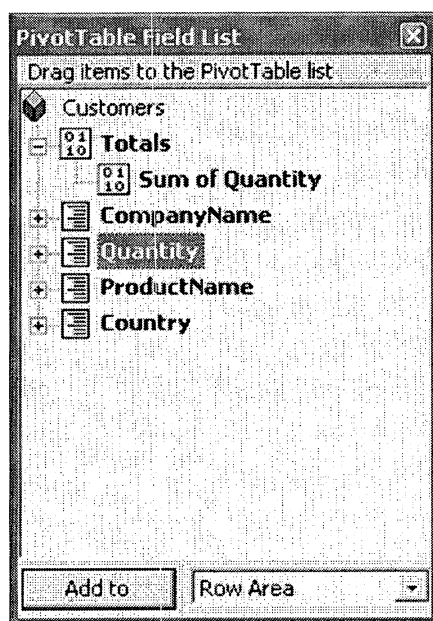


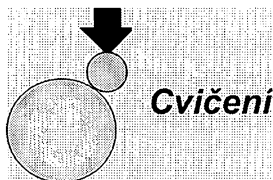
## Úkoly k procvičování:

- 1 V databázi Northwind sestavme kontingenční tabulku, která ukazuje pro zvoleného zákazníka součet množství všech výrobků objednaných z každé země. V okně databáze děláme nový formulář pomocí PivotTable Wizard. Přitom z tabulky **Customers** vybereme pole **CompanyName**, z tabulky **OrderDetails** pole **Quantity**, z tabulky **Products** **ProductName** a z tabulky **Suppliers** pole **Country**.

Kontingenční tabulku vytvoříme podle následujícího obrázku.

| CompanyName                     | Country         |                 |                 |                 | Grand Total     |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|                                 | France          | Germany         | Norway          | USA             |                 |
| Product Name                    | Sum of Quantity | Sum of Quantity | Sum of Quantity | Sum of Quantity | Sum of Quantity |
| Escargots de Bourgogne          | 40              |                 |                 |                 | 40              |
| Fløtemysost                     |                 |                 |                 | 20              | 20              |
| Grandma's Boysenberry Spread    |                 |                 |                 | 16              | 16              |
| Original Frankfurter grüne Soße |                 |                 | 2               |                 | 2               |
| Raclette Courdavault            | 15              |                 |                 |                 | 15              |
| Rössle Sauerkraut               |                 |                 | 17              |                 | 17              |
| Grand Total                     | 55              | 19              | 20              | 16              | 110             |



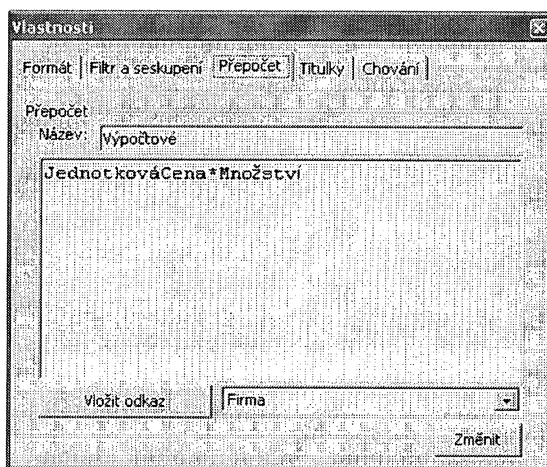


## Úkoly k procvičování:

- 2 Vytvoříme podobnou kontingenční tabulku jako v předchozím cvičení, přidejme do ní ještě pole **UnitPrice** z tabulky **OrderDetails**. V kontingenční tabulce vypočítáme součin jednotkové ceny a množství a ze vzniklého pole (**Calculated**) zobrazíme součty.

Postup:

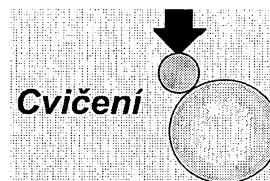
1. Stejně jako v předchozím příkladě získáme pole nutná pro konstrukci kontingenční tabulky.
2. Klepneme na tlačítko **Calculated Totals and Fields** a použijeme příkaz jeho nabídky **Create Calculated Detail Field**. V dialogovém okně **Properties** uděláme výraz pro počítané pole **Calculated** a klepneme na tlačítko **Change**.



3. Kontingenční tabulku znázorněnou na následujícím obrázku potom vytvoříme běžným způsobem.

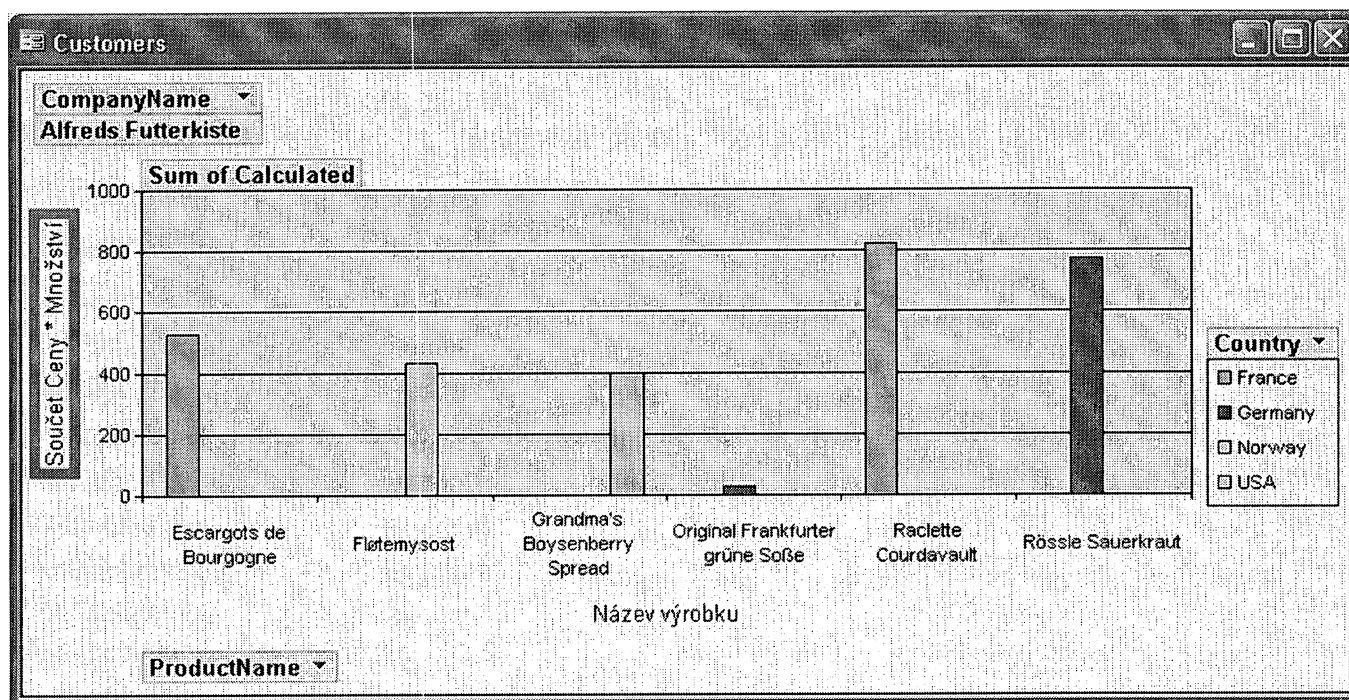
| CompanyName                     | Country | Sum of Calculated | Sum of Calculated |
|---------------------------------|---------|-------------------|-------------------|
| Alfreds Futterkiste             | France  | 530               |                   |
|                                 | Germany |                   |                   |
| Escargots de Bourgogne          |         |                   |                   |
| Fløtemysost                     |         |                   |                   |
| Grandma's Boysenberry Spread    |         |                   |                   |
| Original Frankfurter grüne Soße |         |                   |                   |
| Raclette Courdavault            |         | 825               |                   |
| Rössle Sauerkraut               |         |                   | 775.2             |
| Grand Total                     |         | 1355              | 801               |

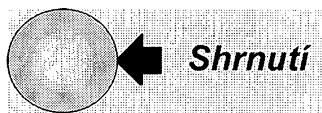




## Úkoly k procvičování:

- 3 K tabulce z předchozího cvičení zhotovme kontingenční graf. Vyzkoušejme si na něm přidávání a odebírání polí, filtrování a formátování. Zkontrolujme, zda se změny učiněné ve struktuře polí grafu promítnou do kontingenční tabulky a naopak.
  1. Zobrazme kontingenční tabulku a vydejme příkaz **View/PivotChart View**.
  2. Z místní nabídky pro popis os volme příkaz **Properties** a upravme popis os podle následujícího obrázku.
  3. Z místní nabídky plochy grafu volme příkaz **Chart Type** a měňme typ grafu.
  4. Z plochy grafu odstraňme pole **Sum of Calculated** a nahradíme jej polem **Quantity** ze seznamu polí.
  5. Obnovme opět zobrazení kontingenční tabulky.





- kontingenční tabulky (grafy) zobrazují souhrnné údaje vypočtené z polí tabulek či dotazů
- způsob seskupení souhrnných údajů určují pole sloupec a řádek
- polem filtru určujeme, které všechny záznamy budou seskupovány
- ve všech základních oblastech kontingenční tabulky lze použít více polí

# Datové stránky

## *Přehled témat*

- tvorba datové stránky
- návrhové zobrazení datové stránky
- seskupování a souhrny
- kontingenční tabulky v datových stránkách



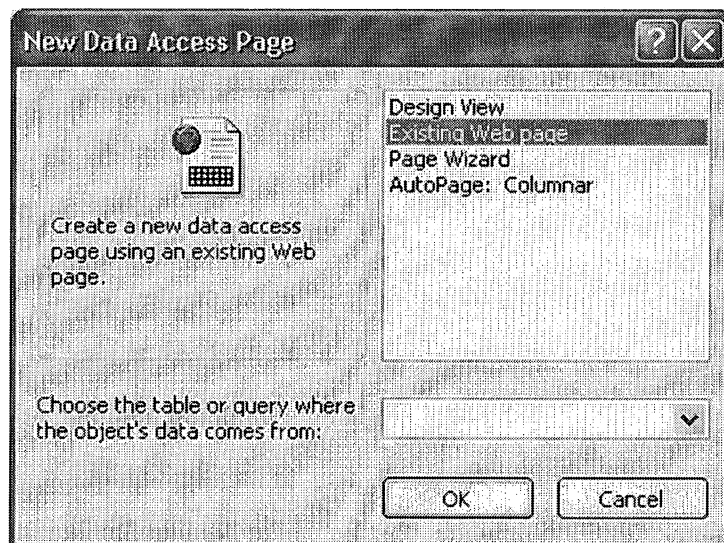
## Datové stránky (Pages)

Datová stránka je zvláštní typ internetové stránky, určené k prohlížení a úpravě dat v databázi prostřednictvím sítě WWW. K práci s publikovanou datovou stránkou stačí uživateli nainstalovaný program Internet Explorer verze 5 a vyšší. Stránku lze též odesílat elektronickou poštou.

Samotná datová stránka není součástí databázového souboru. Je to samostatný soubor ve formátu HTML, který obsahuje odkazy na data v databázovém souboru, je na databázový soubor napojen.

Datovou stránku lze konfigurovat jak pro změnu dat v databázi, tak pouze pro jejich zobrazování.

## Tvorba datové stránky

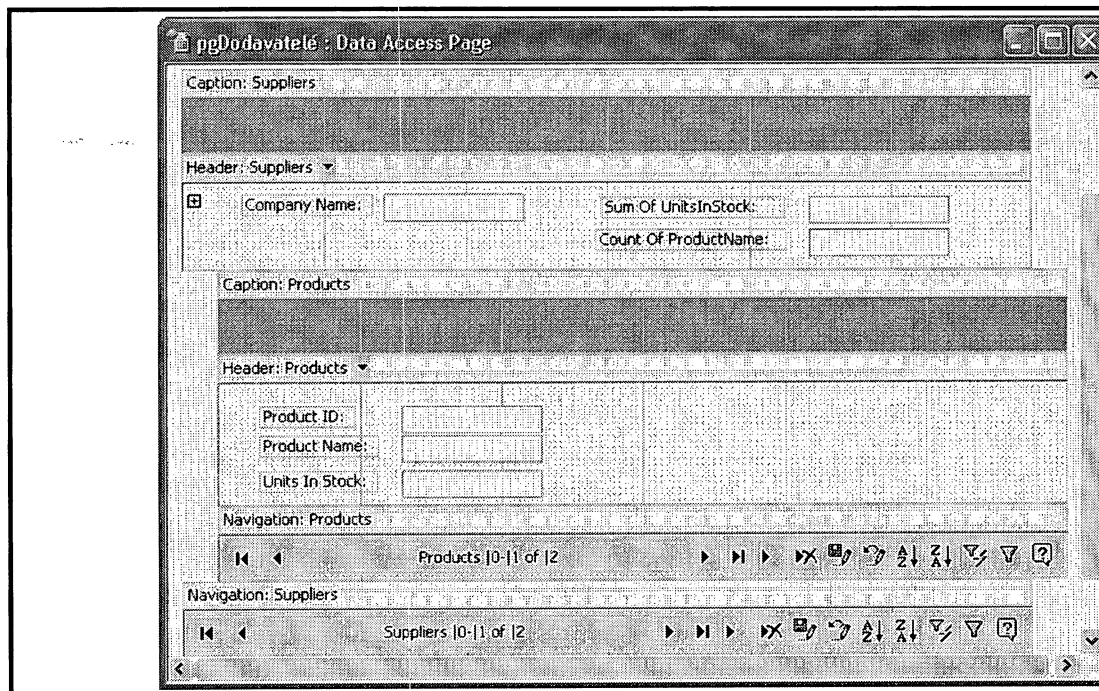


Na obrázku vidíme možnosti tvorby datové stránky. Návrhové zobrazení opět skýtá nejvíce možností, jak stránku vynalézavě vytvořit. Průvodce udělá stránku podle našich odpovědí na jeho dotazy. Automat vytvoří sloupcovou datovou stránku (vidíme pouze jeden záznam, pole záznamu jsou pod sebou) ze zvoleného zdroje.

## Způsoby zobrazení datové stránky

- Návrhové (Design View) - vytváříme a modifikujeme návrh stránky
- Celá stránka (Page View) - zobrazujeme výsledek v Accessu
- Náhled webové stránky (Web Page Preview) - zobrazení v internetovém prohlížeči.

## Návrhové zobrazení datové stránky



### Návrhové zobrazení datové stránky

Datová stránka se skládá z hlavičky pod kterou může obecně být více do sebe vnořených oddílů. Každý oddíl má tyto části

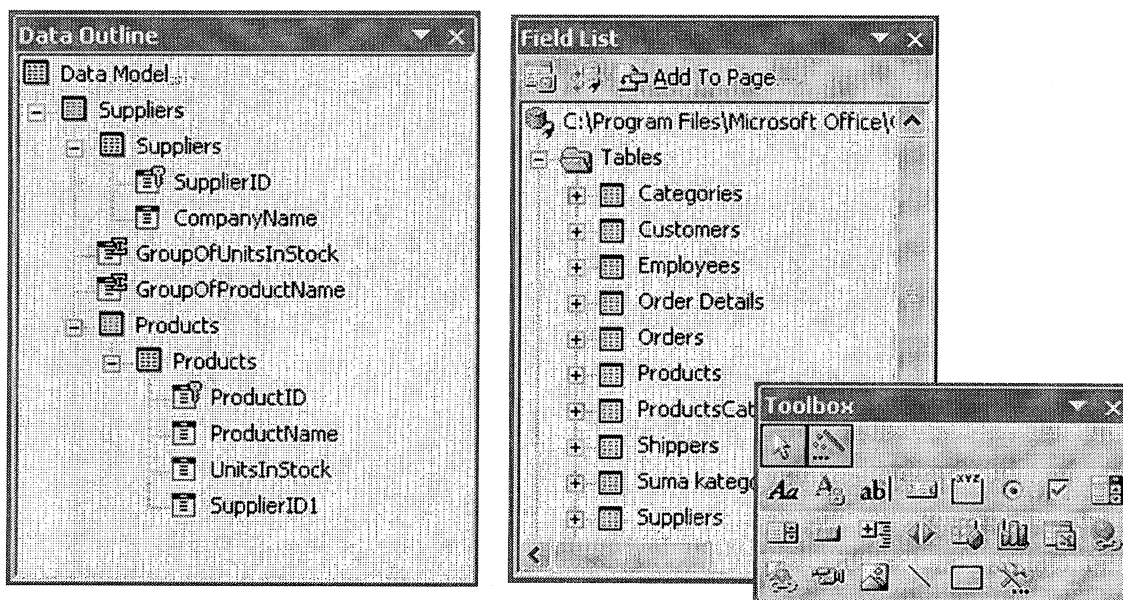
- **Caption** slouží k umístění titulku oddílu.
- **Header/Footer** může obsahovat pole zobrazovaných záznamů a souhrnné údaje.
- **Navigation** obsahuje ovládací prvky pro posun mezi záznamy a pro práci se záznamy.

Samozřejmě, že některé části oddílu mohou být skryty.

V návrhovém zobrazení datové stránky lze ještě otevřít okna **Data Outline**, **Field List**, zobrazit panel **Toolbox** a okno **Properties**.



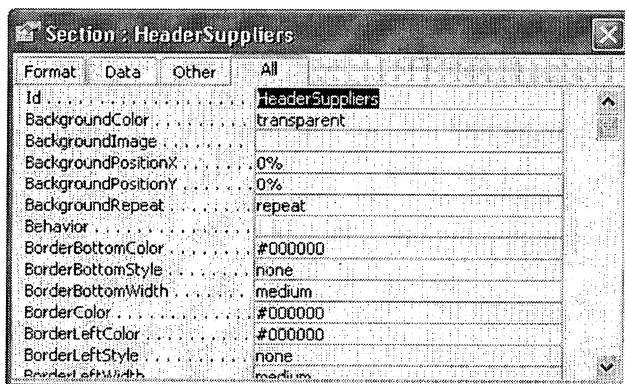
# Návrhové zobrazení datové stránky



V **datové osnově (Data Outline)** je stromovým diagramem vyjádřen datový model stránky. Jsou zde hierarchicky umístěny všechny tabulky a jejich pole zobrazovaná ve stránce a též pole použitá pro definici relací mezi tabulkami a dále i vypočítávané souhrnné ovládací prvky. Vyznačíme-li pole v datové osnově, vyznačí se i odpovídající pole ve stránce. V datové osnově lze upravovat relace, měnit vlastnosti zde zobrazených objektů a odstraňovat je ze stránky. Příslušné příkazy najdeme v místní nabídce objektu datové osnovy.

**Seznam polí (Field List)** datové stránky obsahuje stromový diagram zdrojů záznamů a odpovídajících polí v podkladové databázi. V horní části seznamu polí je ikona **Page connection properties** (zobrazí vlastnosti spojení s podkladovou databází, umožňuje jejich změnu a test spojení). Zde je v případě umístění databáze na síťovém prostředku vhodné změnit absolutní cestu na cestu síťovou ve formátu UNC (Universal naming convention), tedy např. \\dbserver\access\databaze.mdb. Dále jsou zde možnosti **Refresh field list** (aktualizuje seznam polí v podkladové databázi) a **Add To Page** (přidá vyznačená pole do datové stránky).

**Souprava nástrojů (Toolbox)** obsahuje tlačítka, jimiž do stránky vkládáme ovládací prvky.



Jednotlivé součásti datové stránky mají obvykle celou škálu vlastností, které najdeme v okně **vlastností (Properties)** - získáme z místní nabídky prvku.

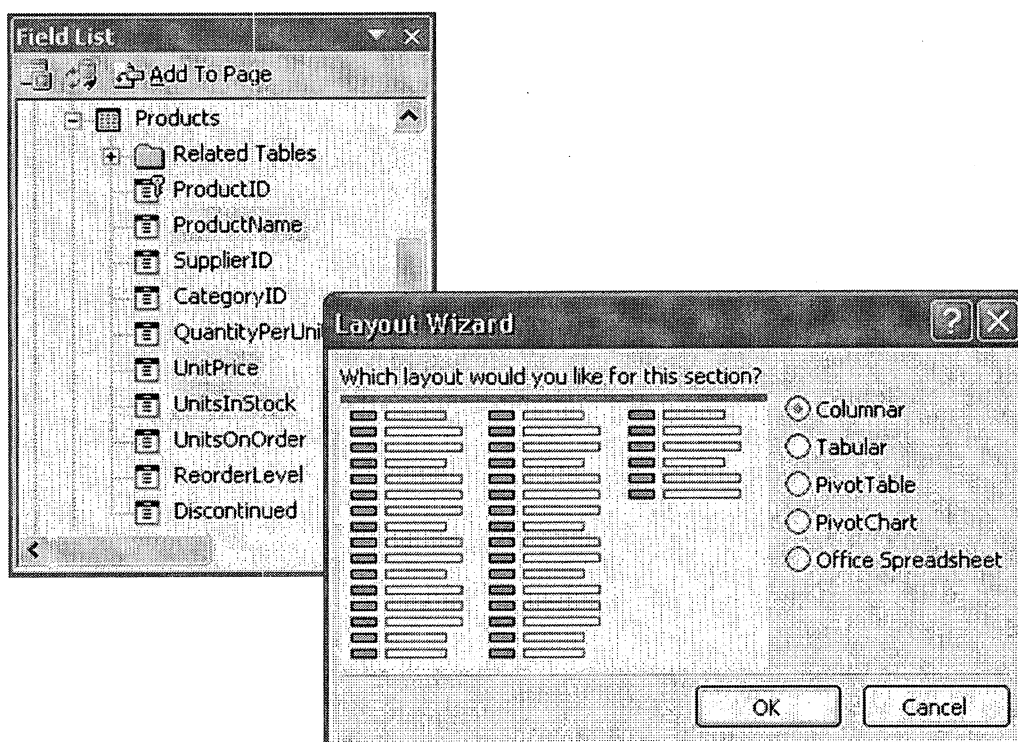
## Tvorba datové stránky v návrhovém zobrazení



Práci v návrhovém zobrazení ukážeme na příkladě sestavení datové stránky obsahující záznamy z tabulek ve vzájemné relaci 1:N (z databáze Northwind to budou tabulky **Suppliers** a **Products**).

V okně databáze aktivujeme objekt **Pages**. Klepneme na tlačítko **New** a ze seznamu vybereme **Design View**. Datový zdroj zatím netřeba specifikovat. V nabídce **View** zajistíme zobrazení **Field List**, **Data Outline** a **Toolbox**.

V datové stránce se objeví nevázaný oddíl. Nyní do něj budeme vkládat z **Field List** pole. Začneme tabulkou na straně N relace (v našem případě to bude tabulka **Products**). Ve **Field List** rozbalíme tuto tabulku a do mřížky zatím nevázaného oddílu datové stránky přetáhneme žádaná pole (**ProductName**, **UnitPrice**, **UnitsInStock**). Pole lze ve **Field List** vyznačit najednou (užijeme k tomu kláves **Ctrl** a **Shift**). Po přetažení do mřížky se objeví **Layout Wizard**. Určíme jedno z nabízených rozvržení (sloupcové).



Tím na stránce vzniká z původního **nevázaného** oddílu **vázaný** oddíl. Objeví se v něm záhlaví a navigace, v záhlaví jsou již naše pole.

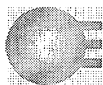
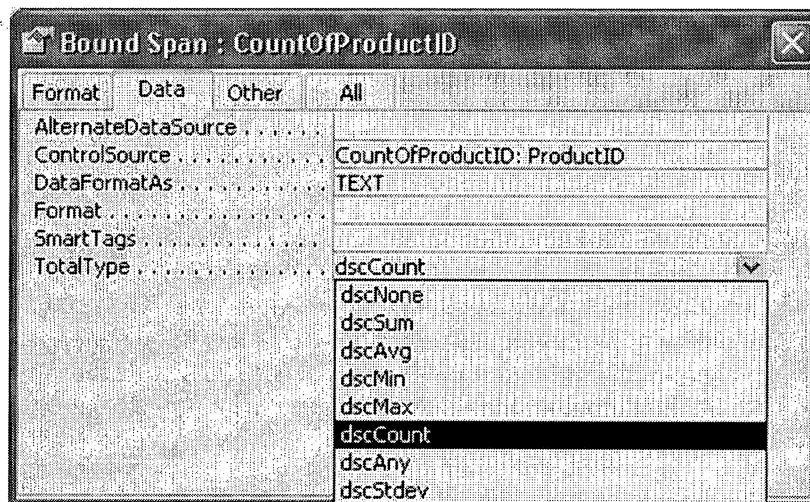
## Vytvoření nadřazeného oddílu

V seznamu polí rozbalíme tabulku na straně jedna v relaci 1:N (**Suppliers**) a táhneme zvolená pole (**CompanyName**, **City**, **Country**) nad záhlaví oddílu existujícího – tabulky na straně N (**Products**). Vytvoří se nový nadřazený oddíl pro tabulku na straně jedna (**Suppliers**) obsahující v záhlaví přetahovaná pole (**CompanyName**, **City**, **Country**). Původní oddíl tabulky na straně N relace je v tomto novém oddílu vnořený.

*Do datové stránky lze ze seznamu polí přesunout celou sbalenou tabulku, tj. všechna pole najednou.*



## Tvorba polí obsahujících souhrnné údaje



### Tvorba polí obsahujících souhrnné údaje

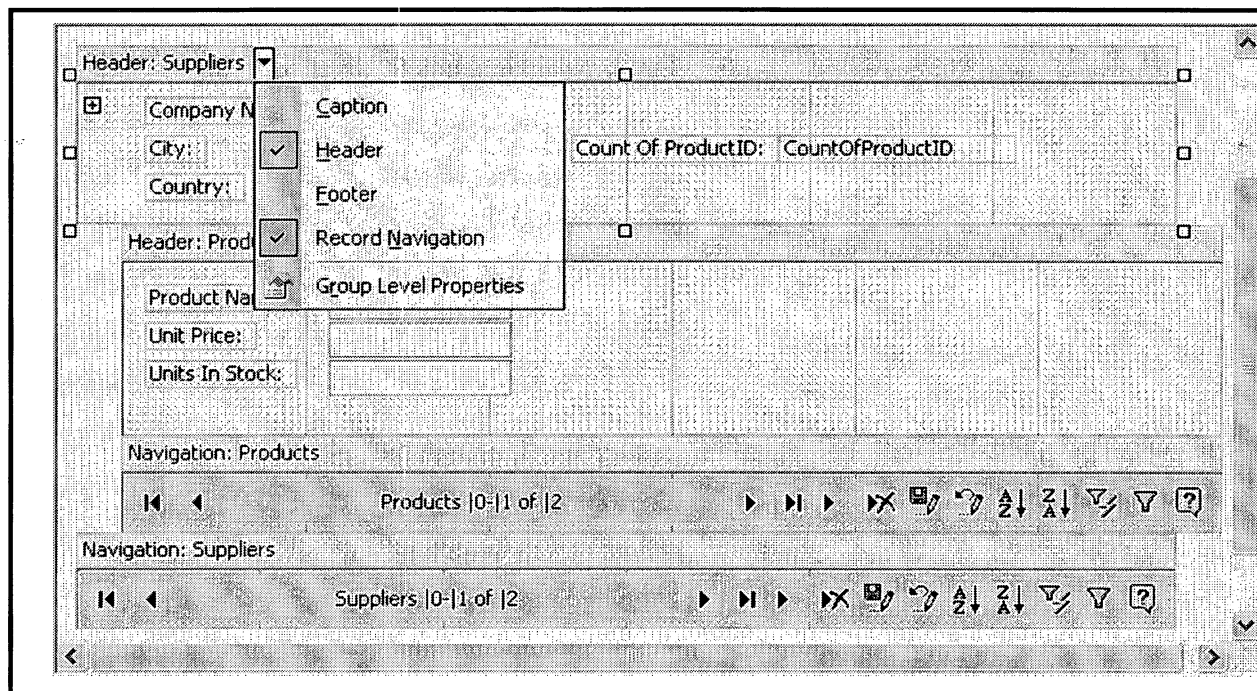
Do nadřazeného oddílu táhneme ze seznamu polí pole tabulky z **vnořného** oddílu (**ProductID** z tabulky **Products**). V nadřazeném oddílu se objeví **vázaný prvek Span** (**Bound Span**), který obsahuje souhrnnou funkci (v našem případě Count) aplikovanou na tažené pole. V okně vlastností tohoto prvku lze měnit typ souhrnné funkce.



Prvek **Bound Span** lze do stránky vložit i ze soupravy nástrojů. V tom případě však musíme jeho datový zdroj (**ControlSource**), typ souhrnné funkce (**TotalType**) a případné další vlastnosti určit sami.



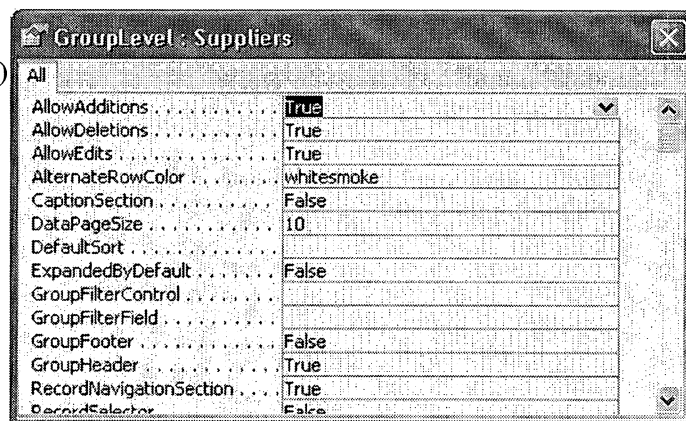
## Vlastnosti seskupování



## Vlastnosti seskupování

V liště záhlaví oddílu je rozbalovací seznam, v němž lze volit zobrazení/skrytí jednotlivých součástí oddílu. Poslední je zde příkaz **Group Level Properties**. Navolíme-li tento příkaz, objeví se v okně vlastností řada příslušných vlastností. Uvedme alespoň některé:

- vlastnosti ovlivňující změny v datovém zdroji (AllowAdditions – přidávání záznamů, AllowDeletions – odstraňování záznamů, AllowEdits – editace záznamů)
- počet záznamů na stránce (DataPageSize)

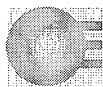


- záznamy jsou sbalené či rozbalené (ExpandedByDefault)
- přítomnost zápatí (GroupFooter) a záhlaví (GroupHeader)
- přítomnost sekce navigace (RecordNavigationSection)

Textové pole v datové stránce má vlastnost **ReadOnly** (True/False) a **Disabled** (True/False). Oběma těmito vlastnostmi lze velice jemně nastavovat přístup k datům v těchto polích.

poznámka





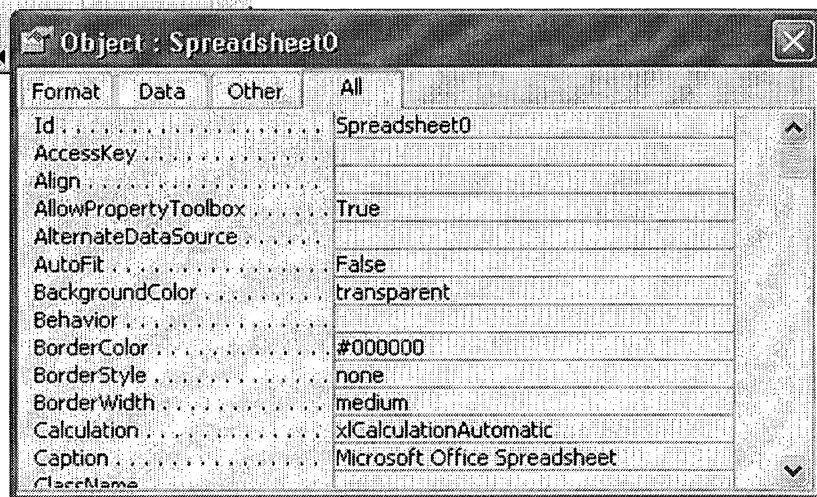
## Důležité součásti panelu Navigation



Na panelu navigace lze kromě navigačních tlačítek určených pro **posun mezi záznamy**, najít ještě tlačítka (sledujte na obrázku v rámečku zleva doprava) pro **vlořádání a odstraňování záznamů**, tlačítko pro **ukládání**, tlačítko **Undo**, tlačítka pro **třídění a filtrování podle výběru**. Poslední je tlačítko **Help**. Zmíněné ovládací prvky užíváme běžným způsobem jako v klasických tabulkách či formulářích přímo v Accessu.

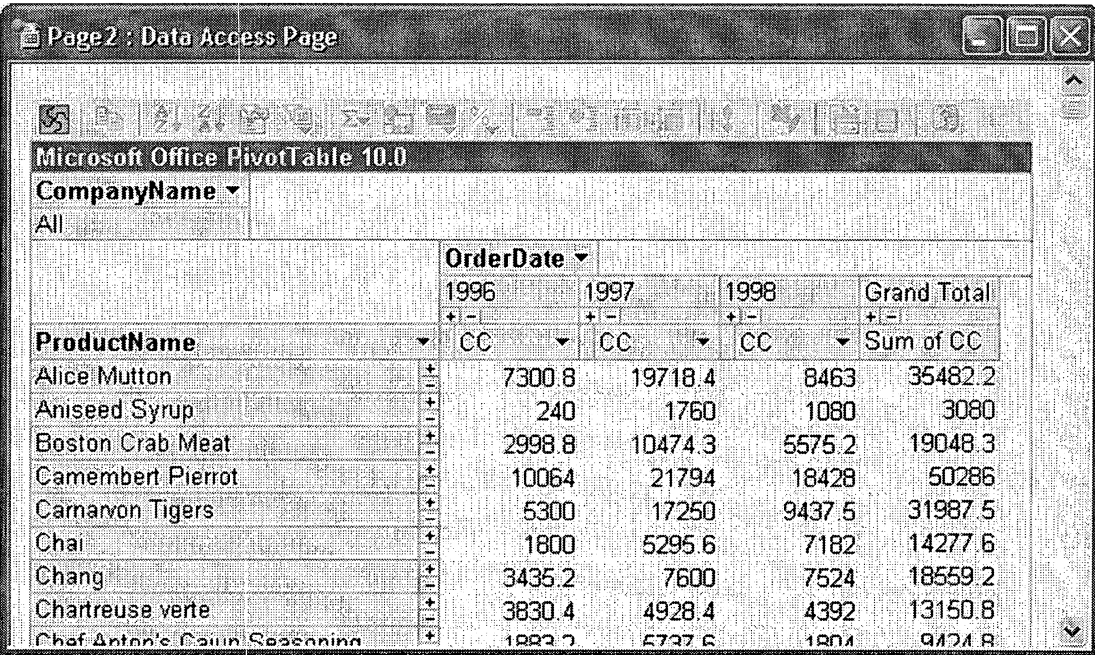
## Tabulka sady Office (Office Spreadsheet)

|   | A   | B | C | D | E |
|---|-----|---|---|---|---|
| 1 | 120 | 6 |   |   |   |
| 2 | 125 |   |   |   |   |
| 3 | 138 |   |   |   |   |
| 4 | 95  |   |   |   |   |
| 5 |     |   |   |   |   |



Do datové stránky lze přidat i **tabulku sady Office**, což je ve své podstatě datový list Excelu. Lze v něm dělat běžné výpočty včetně použití excelovského adresování. Je-li třeba odkazu na pole vně tabulky, děláme to podle předpisu **=host.NázevPole.Value**.

## Kontingenční tabulky v datových stránkách



The screenshot shows a window titled "Page2 : Data Access Page" containing a "Microsoft Office PivotTable 10.0". The PivotTable is set up with "CompanyName" as the filter (set to "All") and "OrderDate" as the column field. The row field is "ProductName". The data is summarized by the sum of "CC" (likely Cost of Goods Sold) for each product across the years 1996, 1997, and 1998, with a "Grand Total" column.

|                              | 1996   | 1997    | 1998   | Grand Total |
|------------------------------|--------|---------|--------|-------------|
|                              | + -    | + -     | + -    | + -         |
| ProductName                  | CC     | CC      | CC     | Sum of CC   |
| Alice Mutton                 | 7300.8 | 19718.4 | 8463   | 35482.2     |
| Aniseed Syrup                | 240    | 1760    | 1080   | 3080        |
| Boston Crab Meat             | 2998.8 | 10474.3 | 5575.2 | 19048.3     |
| Camembert Pierrot            | 10064  | 21794   | 18428  | 50286       |
| Carnarvon Tigers             | 5300   | 17250   | 9437.5 | 31987.5     |
| Chai                         | 1800   | 5295.6  | 7182   | 14277.6     |
| Chang                        | 3435.2 | 7600    | 7524   | 18559.2     |
| Chartreuse verte             | 3830.4 | 4928.4  | 4392   | 13150.8     |
| Chef Anton's Gumbo Seasoning | 1882.2 | 5727.6  | 1804   | 9414.8      |

### Kontingenční tabulky (Pivot Tables) v datových stránkách

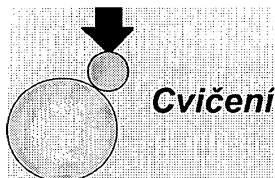
Do datových stránek lze též umísťovat kontingenční tabulky. Při jejich tvorbě postupujeme v podstatě stejně, jako při vzniku formulářové kontingenční tabulky.



V návrhovém zobrazení nové datové stránky lze odstranit výchozí nevázaný oddíl (není to však nutné) a do stránky vložit prvek **Office PivotTable** nebo **Office Chart**.

Do základních oblastí této tabulky potom přemístíme ze seznamu polí pole, která chceme zobrazit. Tato pole se objeví v seznamu polí kontingenční tabulky. Nutno rozlišovat **Field List** (je v každé datové stránce a je dostupný pouze v režimu návrhu) a **PivotTable Field List** (je součástí kontingenční tabulky a je dostupný uživateli při práci s tabulkou v Internet Exploreru např. pomocí místní nabídky). Všechny změny a výpočty, které v tabulce provedeme v návrhovém režimu se s ní ukládají a příslušná pole jsou pak dostupná v režimu náhledu či v Internet Exploreru.

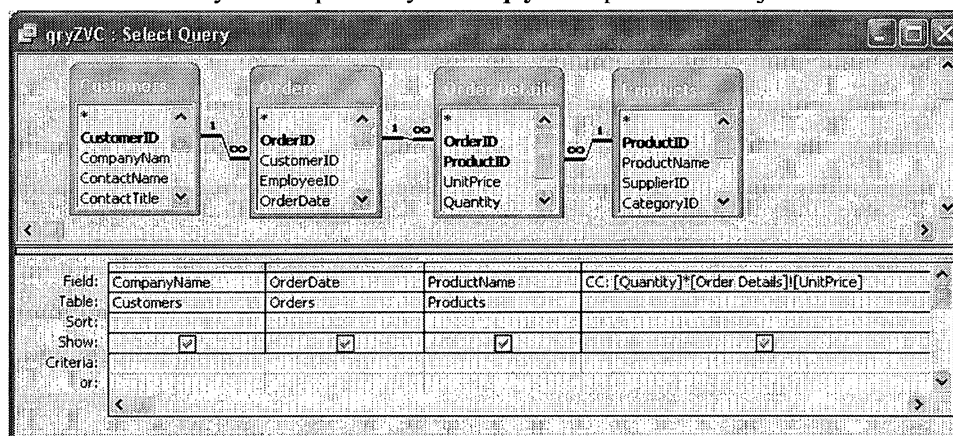
Uživatel může v Internet Exploreru přesouvat pole kontingenční tabulky mezi základními pozicemi tabulky, nebo je z tabulky odstraňovat. Může též dělat souhrnné výpočty stejně jako v případě formulářových kontingenčních tabulek. V panelu nástrojů nad tabulkou jsou k dispozici tlačítka s běžnými funkcemi. Změny udělané ve struktuře kontingenční tabulky v Internet Exploreru ani další výpočty se však do stránky neukládají.



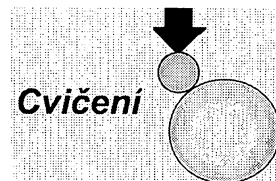
## Úkoly k procvičování:

- 1 V databázi **Northwind** sestavme datovou stránku zobrazující pro každého dodavatele jeho výrobky a celkový součet jednotek na skladě pro dodavatele. Vyzkoušejme si, že dáваме-li do stránky tabulky v relaci 1:N, lze začít přidávat pole z tabulky jak na straně jedna, tak z tabulky na straně N. V každém případě nám bude nabídnuta tvorba podřízeného či nadřazeného oddílu.
- 2 Sestavme v datové stránce kontingenční tabulku, která obsahuje tato pole: **CompanyName** z tabulky **Customers**, **OrderDate** z tabulky **Orders**, **ProductName** z tabulky **Products** a celkovou cenu **CC**, což je součin **Quantity** z tabulky **OrderDetails** a **UnitPrice** z téže tabulky.

Za tímto účelem vytvořme pomocný dotaz **qryZVC** podle následujícího obrázku



1. V okně databáze aktivujeme **Pages** a klepneme na tlačítko **New**. Stránku vytvořme v návrhovém zobrazení.
2. V okně datové stránky klepneme na mřížku a klávesou **Delete** ji odstráňme.
3. V soupravě nástrojů klepneme na tlačítko **Office PivotTable** a v datové stránce vytvořme tahem myši obdélník pro kontingenční tabulku.
4. V seznamu polí najďme dotaz **qryZVC**.
5. Pole **CompanyName** dejme do oblasti filtrů, pole **ProductName** do oblasti řádků, pole **OrderDate** umístěme do oblasti sloupců. Do datové oblasti kontingenční tabulky vlořme pole **CC**.
6. Datum objednávky seskupíme podle roků. V místní nabídce pole **OrderDate** klepneme na **Commands and Options**. V dialogovém okně příkazu na kartě **Filter and Group** volíme seskupení podle roků (v české verzi Accessu ještě na kartě **Formát** změníme číselný formát z "RRRR" na "YYYY").

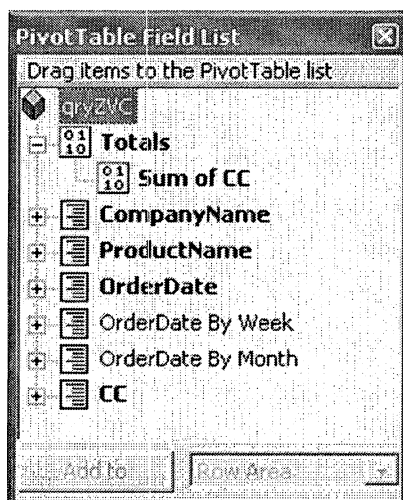


## Úkoly k procvičování:

7. Klepnutím na záhlaví **CC** vyznačíme ceny a tlačítkem **AutoCalc** vytvoříme jejich součty.
8. Po klepnutí na tlačítko **Hide Details** zůstanou zobrazeny pouze součty.
9. Stránku uložíme. Pak příslušný soubor otevřeme Internet Explorerem a ověříme si práci s kontingenční tabulkou.

Microsoft Office PivotTable 10.0

| Company Name                       |   | Order Date |           |           |             |
|------------------------------------|---|------------|-----------|-----------|-------------|
| Ana Trujillo Emparedados y helados |   | 1996       | 1997      | 1998      | Grand Total |
|                                    |   | + -        | + -       | + -       | + -         |
| Product Name                       |   | Sum of CC  | Sum of CC | Sum of CC | Sum of CC   |
| Camembert Pierrot                  | + |            | 340       |           | 340         |
| Gudbrandsdalsost                   | + | 28.8       |           |           | 28.8        |
| Konbu                              | + |            |           | 60        | 60          |
| Mascarpone Fabioli                 | + |            | 320       |           | 320         |
| Mozzarella di Giovanni             | + |            |           | 348       | 348         |
| Outback Lager                      | + | 60         |           |           | 60          |
| Queso Cabrales                     | + |            |           | 42        | 42          |
| Singaporean Hokkien Fried Mee      | + |            | 70        |           | 70          |
| Teatime Chocolate Biscuits         | + |            |           | 64.4      | 64.4        |
| Tofu                               | + |            | 69.75     |           | 69.75       |
| Grand Total                        | + | 88.8       | 799.75    | 514.4     | 1402.95     |





- V návrhovém zobrazení datové stránky pracujeme podobně jako při navrhování formulářů nebo sestav.
- Kromě Field List a Toolbox si můžeme zobrazit okno Data Outline. V datové osnově je stromovým diagramem vyjádřen datový model stránky.
- Pomocí prvků Bound Span můžeme do stránek vkládat souhrnné funkce.

# Sestavy

## *Přehled témat*

- použití sestav
- vytáření sestav pomocí průvodce
- vytváření sestav v návrhovém zobrazení
- řazení a seskupování
- tipy a triky pro práci se sestavami





## Sestavy (Reports)

rptDodavatelé-Výrobky

**rptDodavatelé-Výrobky**      Záhloví sestavy

| Country   | Company Name                                     | City   | Product Name          | Quantity Per Unit   | Záhloví stránky         |
|-----------|--------------------------------------------------|--------|-----------------------|---------------------|-------------------------|
| Australia |                                                  |        |                       |                     |                         |
|           | Pavlova, Ltd                                     | Melbo  | Záhloví skupiny Firma |                     |                         |
|           |                                                  |        | Pavlova               | 32 - 500 g boxes    | 29                      |
|           |                                                  |        | Alice Mutton          | 20 - 1 kg tins      | 0                       |
|           |                                                  |        | Camaronon Tigers      | 16 kg pkg.          | 42                      |
|           |                                                  |        | Outback Lager         | 24 - 355 ml bottles | 15                      |
|           |                                                  |        | Vegie-spread          | 15 - 625 g jars     | 24                      |
|           | Summary for 'SupplierID' = 7 (3 detail records)  |        |                       |                     |                         |
|           | Sum                                              |        |                       |                     | Zápatí skupiny Firma 10 |
|           | G'day, Mate                                      | Sydney |                       |                     |                         |
|           |                                                  |        | Perth Pasties         | 48 pieces           | 0                       |
|           |                                                  |        | Marjilup Dried Apples | 50 - 300 g pkgs.    | 20                      |
|           |                                                  |        | Filo Mix              | 16 - 2 kg boxes     | 38                      |
|           | Summary for 'SupplierID' = 24 (3 detail records) |        |                       |                     |                         |

Page: 1

rptDodavatelé-Výrobky

Jack's NewEngland Cla 12 - 12 oz cans 85

Summary for 'SupplierID' = 19 (2 detail records)

|                                                 |                     |
|-------------------------------------------------|---------------------|
| Sum                                             | 208                 |
| Summary for 'Country' = USA (12 detail records) |                     |
| Sum                                             | 666                 |
| Grand Total                                     | Zápatí sestavy 3119 |

Wednesday, January 11, 2006

Zápatí stránky Page 6 of 6

Page: 6



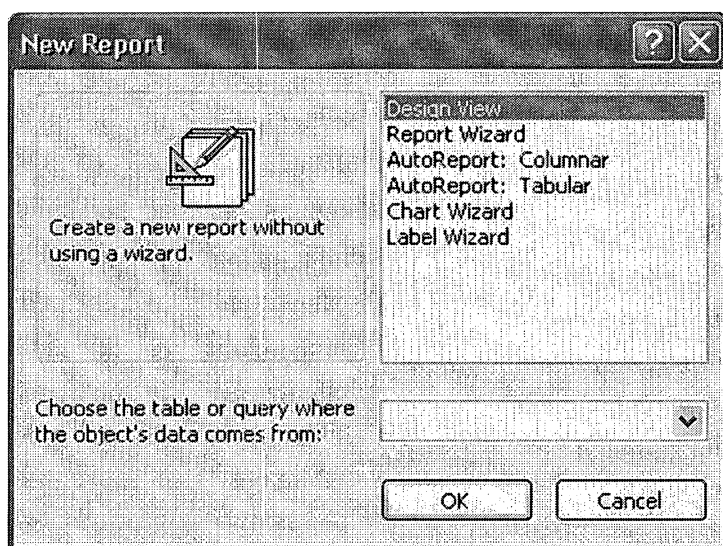
## Charakteristika



Sestavy slouží k tisknutí záznamů databáze. Jejich podstatnou vlastností je, že záznamy lze seskupovat podle zadanych požadavků polí - polí tisknutých záznamů (Firmy, roky, apod.). V sestavě rozlišujeme součásti uvedené v následující tabulce.

| <i>Součást sestavy</i>          | <i>Obvyklý minimální obsah</i> |
|---------------------------------|--------------------------------|
| Záhlaví sestavy (Report Header) | Název sestavy                  |
| Záhlaví stránky (Page Header)   | Názvy polí v sestavě           |
| Záhlaví skupiny (Group Header)  | Název skupiny                  |
| Zápatí skupiny (Group Footer)   | Souhrny pro skupinu            |
| Zápatí stránky (Page Footer)    | Číslo stránek                  |
| Zápatí sestavy (Report Footer)  | Souhrny pro sestavu            |

## Tvorba sestavy



V **Design View** uděláme sestavu od začátku sami. Velice pracný způsob tvorby, ale sestavu zhotovíme tak, jak si přejeme.

**Report Wizard** nám během tvorby klade otázky a sestavu udělá za nás. Zbývá jen opravit drobné detaily.

**AutoReports.** Po zadání zdroje dat je sestava ihned bez dalších dotazů zhotovena. Nakonec opět jen opravíme nedokonalosti. Sestava **Columnar** obsahuje názvy a hodnoty polí vedle sebe. Sestava **Tabular** obsahuje názvy polí v záhlaví stránky.



## Průvodce sestavou (Report Wizard)

Which fields do you want on your report?  
You can choose from more than one table or query.

Tables/Queries:  
Table: Products

Available Fields:

|              |
|--------------|
| ProductID    |
| SupplierID   |
| CategoryID   |
| UnitPrice    |
| UnitsOnOrder |
| ReorderLevel |
| Discontinued |

Selected Fields:

|                 |
|-----------------|
| CompanyName     |
| City            |
| Country         |
| ProductName     |
| QuantityPerUnit |
| UnitsInStock    |

Buttons: Cancel, < Back, Next >, Finish

Nejdříve volíme zdroj dat (Tables/Queries). Z vybraného zdroje potom volíme jednotlivá pole do sestavy. V jedné sestavě lze volit i více zdrojů po sobě (tabulek dotazů). Průvodce sám navrhne seskupení podle existujících relací mezi tabulkami.

Do you want to add any grouping levels?

Fields:

|                 |
|-----------------|
| CompanyName     |
| City            |
| ProductName     |
| QuantityPerUnit |
| UnitsInStock    |

Priority: [v] [v] [v] [v] [v]

Grouping Options ...

Buttons: Cancel, < Back, Next >, Finish

Do sestavy můžeme přidávat úrovně seskupení (grouping levels). Pro každé seskupení lze volit možnosti seskupení (Grouping Options).

What grouping intervals do you want for group-level fields?

Group-level fields:

|         |
|---------|
| Country |
|---------|

Grouping intervals:

|                   |
|-------------------|
| Normal            |
| Normal            |
| 1st Letter        |
| 2 Initial Letters |
| 3 Initial Letters |
| 4 Initial Letters |
| 5 Initial Letters |

Buttons: OK, Cancel

Pro textové pole možno seskupovat podle jednotlivých položek či několika prvních písmen.

Pro číselné položky podle intervalů číselných (10, 50, 100....), u položek typu datum/čas podle intervalů datumových a časových (roky, měsíce, hodiny....).



Pro číselná pole lze volit možnosti souhrnu (Summary Options). Tyto souhrny se objeví v zápatí skupiny. Zobrazení **Detail and Summary** znamená, že ve skupině sestavy budou jednotlivé datové záznamy a v zápatí skupiny zvolené souhrny. Zobrazení **Summary Only** obsahuje pouze souhrny bez konkrétních záznamů.

Dále volíme způsob rozvržení (Layout) sestavy a potom formát (Style) sestavy. Nakonec určíme název sestavy. Průvodce pod tímto názvem sestavu uloží.



## Základní zobrazení sestavy

### Návrhové zobrazení (Design View)

Děláme v něm návrh a konečné úpravy sestavy.

### Náhled (Print Preview)

Prohlížíme sestavu před vytisknutím.

### Ukázka (Layout Preview)

Zobrazujeme ukázková data sestavy způsobem náhledu.

### Návrhové zobrazení – časté operace

The screenshot shows the Design View of a report named 'rptDodavatelé-Výrobky'. The report is organized into several sections:

- Report Header:** Contains the report title 'rptDodavatelé-Výrobky'.
- Page Header:** Contains the fields 'Country', 'Company Name', 'City', 'Product Name', 'Quantity Per Unit', and 'Units In Stock'.
- Country Header:** Contains the field 'Country'.
- SupplierID Header:** Contains the fields 'Company Name' and 'City'.
- Detail:** Contains the fields 'Product Name', 'Quantity Per Unit', and 'Units In Stock'.
- SupplierID Footer:** Contains a summary field 'Sum' with the formula `=Sum([JednotkyNaSkladě])`.
- Country Footer:** Contains a summary field 'Sum' with the formula `=Sum([JednotkyNaSkladě])`.
- Page Footer:** Contains a page number field 'Page' with the formula `=Page`.
- Report Footer:** Contains a grand total field 'Grand Total' with the formula `=Sum([JednotkyNaSkladě])`.

V návrhovém zobrazení sestavy máme stejně jako u formulářů k dispozici seznam polí, soupravu nástrojů a okno vlastností.

Chceme-li do návrhového zobrazení přidat souhrny polí, vytvoříme v příslušné sekci zápatí vypočítávané textové pole se vzorcem, např. `=Sum([JednotkyNaSkladě])`.

Bude-li toto pole v zápatí skupiny **SupplierID**, bude sumarizovat přes každého dodavatele.

Bude-li toto pole v zápatí skupiny **Country**, bude sumarizovat přes každou zemi.

Bude-li toto pole v zápatí sestavy, bude sumarizovat přes celou sestavu.

Obecně: provedeme-li v zápatí/záhlaví skupiny výpočet pomocí souhrnné SQL funkce (sum, average, min, max, ...), počítáme souhrn pro celou skupinu, ke které zápatí/záhlaví patří.

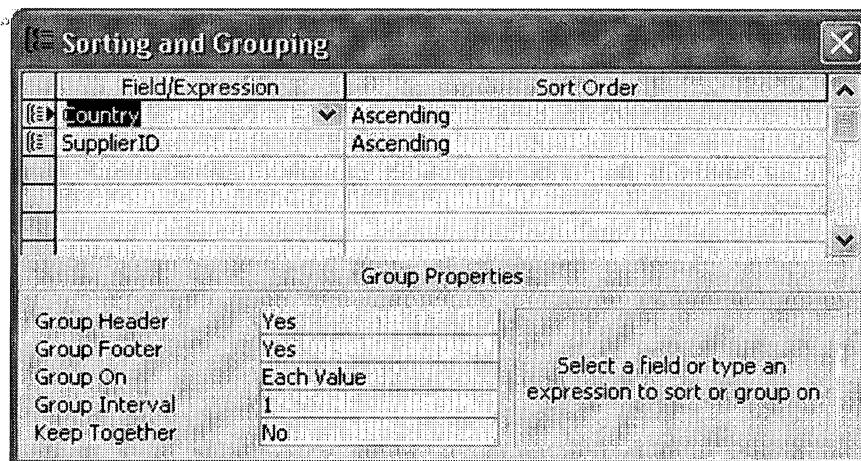


Datum a číslo stránky přidáme do příslušné sekce příkazem ***Insert/Page Numbers*** a ***Insert/Date and Time***. Hodnotu datumu a času poskytuje též funkce **Now**, hodnotu čísla stránky udává pole [Page], počet stránek udává pole [Pages].

Vzorec `= "Page " & [Page] & " of " & [Pages]` vypíše jak číslo stránky, tak i celkový počet stránek.

Je-li zapotřebí oddělit jednotlivé skupiny vodorovnými čarami, použijeme k tomu účelu prvek **Line** z **Toolboxu**. Vodorovné čáry děláme snadno se stisknutou klávesou **Shift**.

# Řazení a seskupování v sestavě



## Řazení a seskupování v sestavě



Skupiny v sestavě lze dělat a odstraňovat v dialogovém okně **Sorting and Grouping** které získáme klepnutím na stejnojmenné tlačítko panelu nástrojů. Pokud děláme sestavu pomocí průvodce, nalezneme v tomto dialogovém okně položky odpovídající úrovním seskupení, které jsme v průvodci volili.

Položka umístěná v okně **Sorting and Grouping** vytvoří skupinu, nadefinujeme-li jí buď záhlaví či zápatí, nebo obojí, což je asi nejčastější případ. Vlevo od názvu položky se objeví ikona – symbol skupiny.

Položka umístěná v tomto okně bez symbolu skupiny indikuje řazení záznamů ve skupině podle této položky.

Chceme-li odstranit skupinu ze sestavy, odstraníme příslušnou položku ze seznamu okna **Sorting and Grouping**.

Chceme-li přidat skupinu do sestavy, vyplníme v okně **Sorting and Grouping** nový řádek a přidáme pro něj záhlaví či zápatí skupiny.

## Vlastnost Group On

U **textových polí** volíme **Each Value**, nebo **Prefix Characters**. První volba znamená, že seskupujeme podle celé hodnoty v poli, druhá, že seskupujeme podle několika prvních znaků. Jejich počet nastavíme jako vlastnost **Group Interval**.

U **číselných polí** volíme **Each Value**, nebo **Interval**. První volba znamená, že seskupujeme podle celé hodnoty v poli, druhá, že seskupujeme do intervalů, jejichž délku určíme jako vlastnost **Group Interval**.

Podobně postupujeme i u polí typu **Date/Time**.

## Vlastnost Keep Together

Má tři hodnoty: **No**, **Whole Group**, **With First Detail**.

Hodnota **Whole Group** znamená, že celá skupina bude vytištěna na téže stránce (je-li to ovšem možné).

Hodnota **With First Detail** znamená, že záhlaví skupiny se na stránku nevytiskne, nevejde-li se na tuto stránku ještě minimálně jeden záznam patřící do této skupiny.



## Tipy a triky pro práci se sestavami

### Odstránkování u sekcí

Měníme pomocí vlastnosti **Force New Page** (vlastnost má hodnoty **None**, **Before Section**, **After Section**, **Before & After**).

### Tisk záhlaví a zápatí stránek v sestavě

Někdy nechceme tisknout záhlaví či zápatí stránek sestavy na první nebo poslední stránce sestavy. To lze ovlivňovat nastavením vlastnosti sestavy **Page Header (Page Footer)** na příslušnou hodnotu (**All Pages**, **No with Rpt Hdr**, **No with Rpt Ftr**, **No with Rpt Hdr/Ftr**).

### Tisk záhlaví skupiny(sekce) na každé straně

Obsahuje-li skupina tolik záznamů, že se tisknou na více stránek, je někdy vhodné tisknout záhlaví skupiny na každé této stránce. Za tím účelem nastavíme vlastnost skupiny **Repeat Section** na **Yes**.

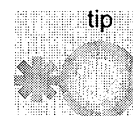
### Tisk pořadového čísla záznamu

Do sekce **Detail** umístíme nevázané textové pole a jeho vlastnost **Control Source** nastavíme na **=1**.

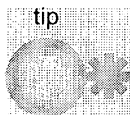
Jeho vlastnost **Running Sum** lze přepínat na hodnoty **No**, **Over Group** a **Over All**. Nastavíme-li přes skupinu, pak se záznamy v nové skupině číslují vždy od jedné. Nastavíme-li přes všechno, pak se záznamy číslují průběžně v celé sestavě.

### Tisk polí s proměnnou délkou

Pro pole s možnou velkou délkou (memo) nastavíme vlastnost **Can Grow** na **Yes**, pro pole s možnou malou délkou nastavíme vlastnost **Can Shrink** na **Yes**.







## Sestavy s parametrem

Chceme-li před tiskem zadat konkrétní hodnotu do sestavy, možno to učinit pomocí parametru. Do sestavy dáme textové pole a jeho vlastnost **Zdroj ovládacího prvku** nastavíme na hodnotu **=[Nápověda při zadání parametru]**, kde text uvnitř hranatých závorek musí být různý od názvů všech polí v sestavě. Zobrazuje se v okně pro zadání parametru jako nápověda. Toto okno se objeví před náhledem či vtištěním sestavy.

## Podsestavy (Subreports) – využití

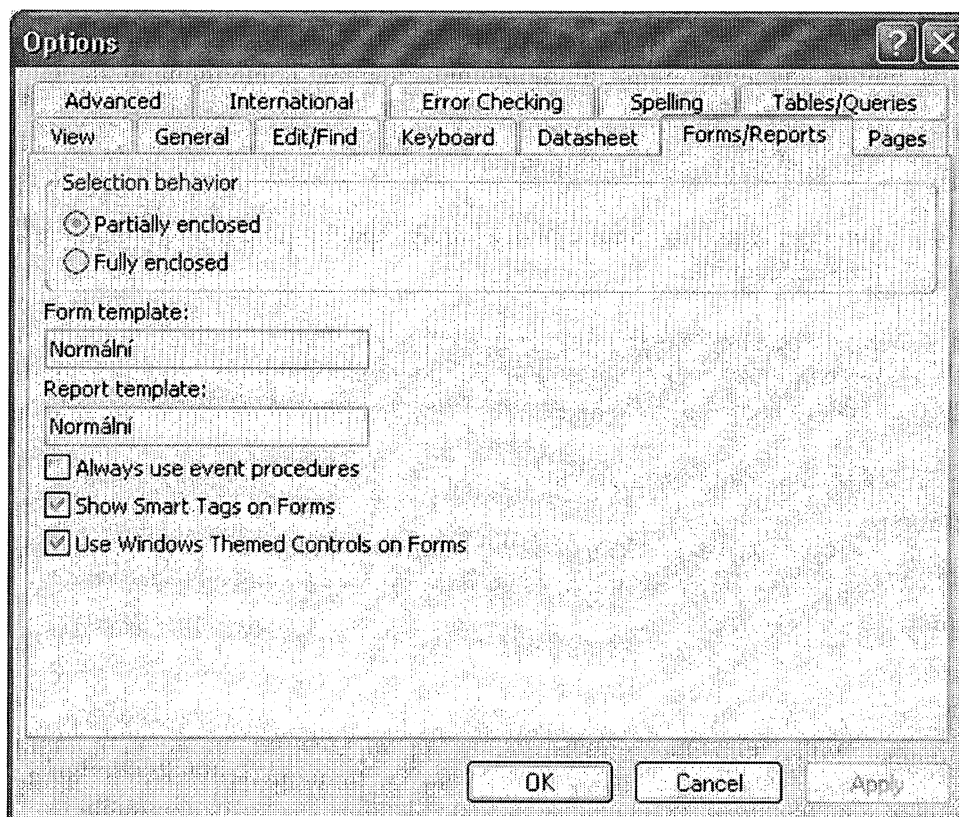
Podsestavy lze v sestavě využít podobně jako u formulářů, tj. k realizaci relace 1:N.

Pomocí podsestavy lze do sestavy vložit i jiná data, např. výsledky souhrnného dotazu.

Vložíme-li do těla sestavy podsestavu s více sloupci, lze dosáhnout efektu tisku více sloupců v těle sestavy a jednoho sloupce v ostatních částech sestavy.

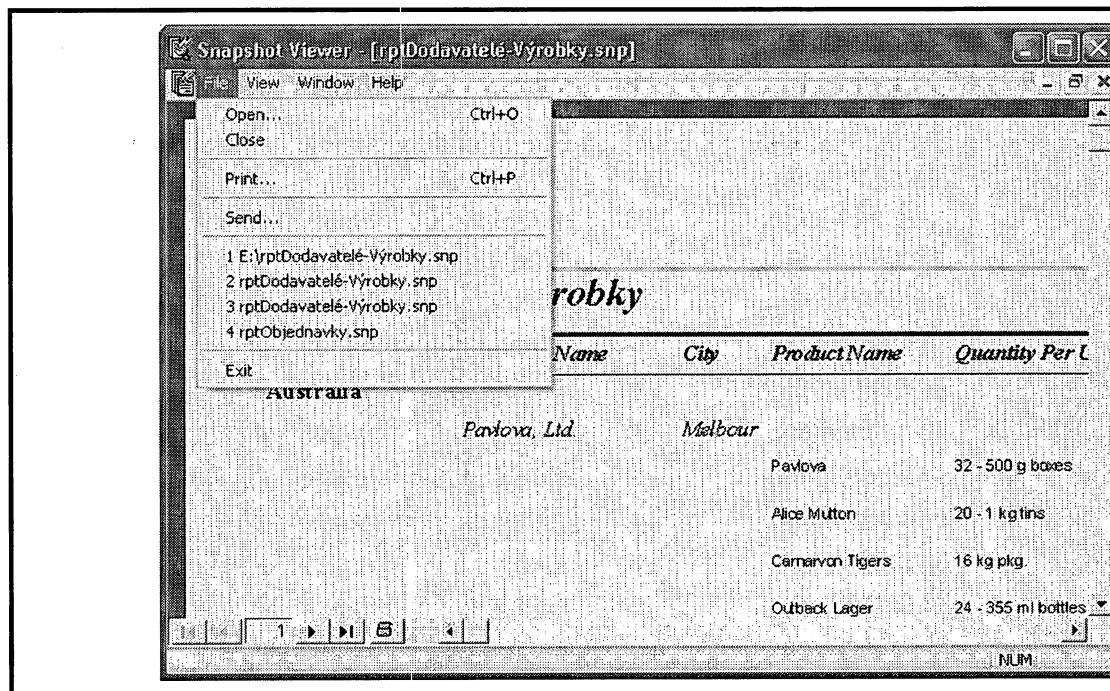
## Šablony formulářů a sestav

Děláme-li velké množství formulářů či sestav se stejnými sekcemi a ovládacími prvky, stačí vytvořit první formulář či sestavu. Tento objekt potom nadefinujeme jako šablonu podle které budou vznikat další formuláře či sestavy. Definici provedeme v dialogovém okně příkazu **Tools/Options**, karta **Forms/Reports**. Zde do okna **Form Template** či **Report Template** napíšeme název příslušné šablony.





## Snímky sestavy



## Snímky sestavy

Snímky sestavy jsou soubory s příponou SNP, které obsahují kopii sestavy. K práci s hotovými snímky slouží program **Snapshot Viewer**, který je bezplatně k dispozici (stažení) na Internetu (netřeba mít aplikaci Access). Program zobrazuje, tiskne a odesílá (E-mailem) hotové snímky.

Snímky lze též vkládat do internetových stránek, buď pomocí odkazu na ně, nebo pomocí ActiveX prvku snapview.ocx.

## Výhody používání snímků

Distribuce snímků sestav vede ke značným úsporám a přispívá i k ochraně dat v databázi. Příjemce snímku nemusí mít nainstalován program Access, ani nemusí mít k dispozici celý databázový soubor. Stačí mu jen relativně velice malý soubor se snímkem a program Snapshot Viewer. Ve snímcích se zobrazují i obrázky vložené do sestavy.

## Zhotovení snímku

V okně databáze vyznačíme objekt **Reports**. Z místní nabídky zvolené sestavy vykonáme příkaz **Export**. Jako typ souboru nadefinujeme **Snapshot Format (\*.snp)** a specifikujeme složku a název souboru.



## Práce s hotovým snímkem

Nainstalujeme program Snapshot Viewer. Lze to učinit jako součást instalace Accessu, nebo jej stáhneme z WWW stránky Microsoftu. Při standardní instalaci Accessu je úplný název souboru s programem

c:\Program Files\Common Files\Microsoft Shared\Snapshot Viewer\SNAPVIEW.EXE.

Program bud' spustíme a standardně (***File/Open***) otevřeme soubor se snímkem, nebo na souboru se snímkem poklepeme.

Po otevření souboru se snímkem lze z nabídky ***File*** programu volit příkazy ***Print*** nebo ***Send*** k odeslání souboru snímku E-mailem.

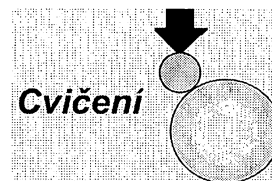
Soubor se snímkem lze samozřejmě standardním způsobem vložit do poštovní zprávy jako přílohu.

## Publikování snímku na WWW stránce

Na stránku lze umístit HTML odkaz na snímek

(tag <A HREF="http://CestaKSouboru/sestava.snp">Sestava Katalog</A>),

nebo ActiveX objekt snapview.ocx, např. programem Front Page. Zmíněný ActiveX objekt nám snímek zobrazí. Objekt je součástí instalace programu Snapshot Viewer. Při prohlížení snímku na stránce lze z místní nabídky snímku volit jeho vytištění.



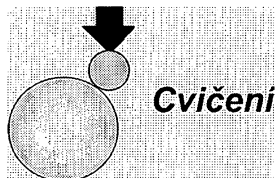
## Úkoly k procvičování:

- 1 V databázi **Northwind** vytvoříme pomocí průvodce sestavou sestavu pro tisk dodavatelů a jimi dodávaných výrobků. V sestavě seskupíme podle země dodavatele a firmy dodavatele. V zápatí oddílů sčítáme pole **UnitsInStock**.

Postup:

1. V okně databáze aktivujeme **Reports**, klepneme na tlačítko **New** a volíme **Report Wizard**.
2. Z tabulky **Suppliers** volme pole **CompanyName** a **Country**, z tabulky **Products** vezmeme pole **ProductName**, **UnitPrice** a **UnitsInStock**.
3. V průvodci sestavou navolíme seskupování podle **Country** a potom podle **CompanyName**, sčítat budeme **UnitsInStock**. Záznamy budeme třídit podle názvu výrobku.

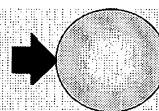
Sestavu uložíme jako **rptDodavatelé** a zkontrolujeme výsledek.



## Úkoly k procvičování:

- 2** V databázi **ZamDěti** vytvoříme sestavu obsahující tabulku **Lidi** s podsestavou obsahující děti zaměstnanců.
- Postup:
1. Vytvoříme nejprve z tabulky **Lidi** automatickou sestavu sloupcovou a uložíme ji jako B.
  2. Přejdeme do návrhového režimu sestavy a z těla odstraníme všechna pole kromě primárního klíče, příjmení a jména.
  3. Potom vložíme do těla sestavy podsestavu (tlačítkem **Subform/Subreport** soupravy nástrojů). Měla by v těle ležet pod zbylými poli, aby se šířka sestavy nezměnila (jinak by šířka sestavy překročila šířku papíru). Funkci podsestavu specifikujeme pomocí průvodce, podobně jako při tvorbě podformulářů. Jako zdroj záznamů podsestavu volíme tabulku **Děti**.
  4. U těla sestavy a u podsestavu nastavíme **Can Shrink** na **Yes**. Potom se na stránku vejde více záznamů lidí bez dětí.
  5. Jak zabráníme tomu, aby se v případě, kdy rodič nemá děti zobrazoval popisek podsestavu?
    - a. V návrhovém režimu hlavní sestavy tento popisek odstraníme.
    - b. Otevřeme samotnou sestavu dětí (která je podsestavou) v návrhovém režimu.
- Do záhlaví sestavy dětí vložíme textové pole, které bude sloužit jako titulek sestavy, a odstraníme jeho popisek. Jako zdroj ovládacího prvku pro toto textové pole zapíšeme výraz **=IIf(Count(\*)>0;"Děti";"" )**. Funkce Count(\*) určí celkový počet záznamů. Funkce IIF pracuje podobně jako funkce If Excelu. Vyhodnotí podmínku (Count(\*)>0). Je-li tato pravdivá, vypíše hodnotu druhého argumentu (Děti), jinak vypíše hodnotu třetího argumentu (prázdný řetězec). U diskutovaného textového pole nastavíme též **Can Shrink** na **Yes**.

## Shrnutí



- Sestavy slouží k tisknutí informací vybraných z databáze.
- Sestavu můžeme jednoduše vytvořit pomocí průvodce, a potom upravit v návrhovém zobrazení.
- Záznamy v sestavě lze seskupovat podle zadaných požadavků (polí tisknutých záznamů).
- Sestavu lze exportovat ve formátu .snp, což je přesná obrazová kopie sestavy. Tyto soubory si pak můžete prohlížet pomocí programu Snapshot Viewer.

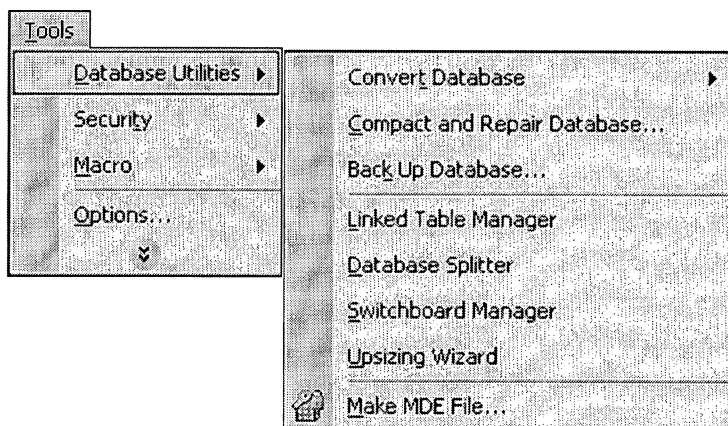


# Elementy správy databáze

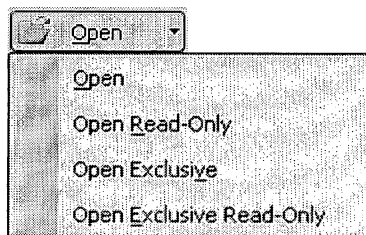
## *Přehled témat*

- možnosti otevření databáze
- ochrana databáze heslem
- komprimace a oprava databáze

# Otevření databáze



V nabídce **Tools/Database Utilities** je řada příkazů určených pro správu a údržbu databáze. V dalším zmíníme nejdůležitější z nich. Dříve si však ještě povíme o různých způsobech otevření databáze.



## Otevření databáze

Šipka tlačítka **Open** dialogového okna pro otevření databáze zobrazí nabídku. Volíme-li v ní příkaz

- **Open**, pak se databáze otevírá jak pro čtení, tak pro zápis.
- **Open Read-Only**, otevíráme si databázi jen pro čtení. V tomto režimu nelze měnit její obsah.
- **Open Exclusive**, pak nemohou takto otevřenou databázi otevřít další uživatelé.
- **Open Exclusive Read-Only**, pak je zabráněno ostatním uživatelům databázi otevřít a tato je otevřena jen pro čtení.



Nevolíme-li výhradní přístup k databázi, otevírá se ve sdíleném režimu, to znamená, že s ní může pracovat více uživatelů současně. Toto je běžný způsob otevření databáze realizovaný též klepnutím na větší část tlačítka **Open**.

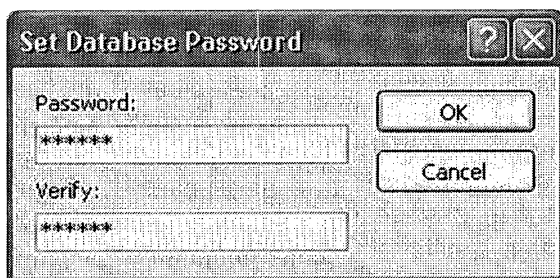


Výchozí režim otevření realizovaný větší částí tlačítka **Open** je běžně nastaven na **Shared**. V dialogovém okně **Tools/Options**, karta **Advanced**, lze však přepnout výchozí režim na **Exclusive**.



Otevíráme-li databázi jako některý z naposledy otevřených souborů, jejichž názvy jsou umístěny v dolní části nabídky **File**, pak je otevřena v takovém režimu přístupu, v němž jsme ji otevřeli naposledy.

## Vstupní heslo



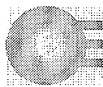
Příkazem **Tools/Security/Set Database Password** můžeme opatřit otevřenou databázi heslem. Při jejím novém otevírání nutno toto heslo zadat, jinak nelze databázi otevřít.

### Postup při zadání hesla

1. Databázi otevřeme ve výhradním režimu
2. Vykonáme příkaz **Tools/Security/Set Database Password** a do jeho dialogového okna zadáme dvakrát heslo (podruhé pro ověření).

### Postup při zrušení hesla

1. Databázi otevřeme ve výhradním režimu
2. Vykonáme příkaz **Tools/Security/Unset Database Password** a do jeho dialogového okna zadáme platné heslo.



## Převod databáze do různých verzí Accessu

Vlastní převod děláme z nabídky příkazu *Tools/Database Utilities/Convert Database*.

Databázi verze 97 lze ve vyšších verzích Accessu, otevřít, ale nelze provádět změny databázových objektů (tabulky, dotazy, apod.). To znamená, že změny udělané v návrhovém režimu existujících objektů nelze uložit, nové objekty nelze do databáze přidat. Chceme-li v této databázi dělat změny u objektů, musíme ji napřed převést do formátu vyšší verze, tj. 2000 resp. 2002. Vlastní převod uděláme příkazem *Tools/Database Utilities/Convert Database/To Access 2000 File Format*, resp. *Tools/Database Utilities/Convert Database/To Access 2002 - 2003 File Format*.

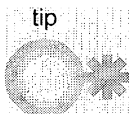
Databázi převedenou do formátu vyšší verze aplikace Access však již neotevřeme v nižší verzi Accessu.

Do formátu 2002 - 2003 nutno převádět databázi pouze ve výjimečných případech (tvorba souboru MDE či ADE ve verzi 2002). Víme-li však, že s databází budou pracovat pouze uživatelé verze 2002, můžeme ji do tohoto formátu převést.

## Komprimace a oprava databáze

Po větších změnách v databázi doporučujeme provést příkaz *Tools/Database Utilities/Compact and Repair Database*. Velikost databázového souboru se po provedení toho příkazu může významně zmenšit. Při komprimaci však nejde o aplikaci žádné zipovací techniky na databázový soubor. Jeho formát se nemění. Jde jen o jeho eventuální defragmentaci a efektivnější využití diskového prostoru při uložení databáze.

Je-li databázový soubor poškozen, nabízí při otevírání Access jeho opravu. Komprimace a oprava databáze je nyní v Accessu prováděna současně. Probíhající příkaz komprimace a opravy lze předčasně ukončit povel **Ctrl+Break** či klávesou **Esc**.

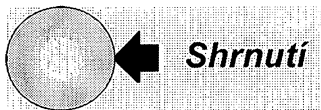


*Převádíme-li databázi verze 97 do formátu vyšší verze, volme spíše verzi 2000. Pracujeme-li s touto databází v Accessu 2002 nebo 2003, máme zde k dispozici i nové schopnosti této verze a databázi mohou otevřít i uživatelé Accessu 2000.*

### **Ztrátě dat v důsledku poškození databáze zabráníme zejména**



1. Pravidelnou aplikací příkazu *Compact and Repair Database*.
2. Ukončujeme-li aplikaci Access standardním způsobem (ne např. náhlým vypnutím počítače).
3. Nepřistupujeme-li k síťové databázi při výskytu problémů na síti.
4. Pravidelným zálohováním databázového souboru kopírováním nebo z okna databáze příkazem *Tools/Database Utilities/Back Up Database*.

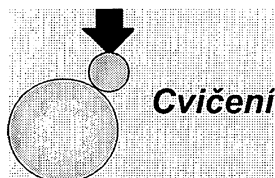


- Při otvírání databáze ji můžeme otevřít s výhradním přístupem.
- Pokud je databáze otevřena s výhradním přístupem, můžeme ji příkazem **Tools/Security/Set Database Password** opatřit heslem.
- Převod databáze do jiné verze Accessu děláme z nabídky příkazu **Tools/Database Utilities/Convert Database**.
- Nezapomínejme databáze komprimovat. Umožní nám to příkaz **Tools/Database Utilities/Compact and Repair Database**.

# Úkoly k procvičování

## *Přehled témat*

- závěrečná cvičení



## Úkoly k procvičování:

- 1 V databázi **Northwind** vytvoříme pomocí průvodce formulářem formulář **frmDodavatelé**, který zobrazuje dodavatele a jimi dodávané výrobky (jsou v podformuláři). Do formuláře přidejme seznam zobrazující všechny různé země dodavatelů. Potom udělejme parametrický dotaz, který zobrazuje dodavatele a jimi dodávané výrobky pro zvolenou zemi (parametr) v seznamu zemí na formuláři. K tomuto dotazu vytvoříme sestavu **rptDodavateléZeZemě** seskupující podle firmy dodavatele ze zvolené země. V zápatí pro firmu sčítáme **UnitsInStock**.

Na formulář **frmDodavatelé** dejme tlačítko, které zobrazí náhled sestavy pro zemi vybranou v seznamu formuláře.

Výsledný formulář:

## Úkoly k procvičování:

Po klepnutí na tlačítko **Náhled sestavy** se objeví náhled sestavy pro zemi vyznačenou v seznamu formuláře.

| Company Name                                                  | Product Name            | Unit Price | Units In Stock |
|---------------------------------------------------------------|-------------------------|------------|----------------|
| G'day, Mate                                                   | Perth Pasties           | \$22.50    | 0              |
|                                                               | Filo Milk               | \$7.00     | 38             |
|                                                               | New! In up Dried Apples | \$53.00    | 20             |
| Summary for 'Company Name' = G'day, Mate (3 detail records)   |                         |            |                |
| Sum                                                           |                         |            | 58             |
| Pavlova, Ltd.                                                 | Onback Lager            | \$15.00    | 15             |
|                                                               | Vegete spread           | \$43.90    | 24             |
|                                                               | Canarian Tiger          | \$62.50    | 42             |
|                                                               | Albe Minibla            | \$39.00    | 0              |
|                                                               | Pavlova                 | \$17.45    | 29             |
| Summary for 'Company Name' = Pavlova, Ltd. (5 detail records) |                         |            |                |
| Sum                                                           |                         |            | 110            |
| Jednoty na skladě celkem pro zemi Australia                   |                         |            | 168            |

### Postup:

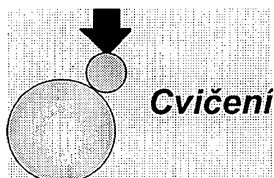
Pomocí průvodce formulářem uděláme formulář zobrazený výše.

V jeho návrhovém zobrazení vložíme do zápatí seznam, který nazveme **lst**.

### Naplnění seznamu položkami – země dodavatelů:

1. Při vkládání seznamu využijeme průvodce seznamem a hodnoty pro seznam načteme z tabulky **Suppliers**, pole **Country**.
2. Průvodce naplní seznam hodnotami pole **Country**, v poli jsou však duplicity.

Zdrojem položek pro náš seznam je dotaz, jehož tvar musíme upravit. Na kartě **Data** okna vlastností seznamu klepneme v řádku nazvaném **Row Source** na tlačítko s výpustkem. Objeví se tvůrce dotazů zobrazující zdrojový dotaz, v němž odstraníme z kritériální mřížky pole **SupplierID** a pro odstranění duplicit z výsledku dotazu nastavíme vlastnost dotazu **Unique Values** na **Yes**. Dále upravíme vlastnost seznamu **lst** nazvanou **Column Widths** podle následujícího obrázku, který zachycuje konečný stav vlastností seznamu **lst**.



## Úkoly k procvičování:

| Property        | Value                                                                        |
|-----------------|------------------------------------------------------------------------------|
| Control Source  |                                                                              |
| Row Source Type | Table/Query                                                                  |
| Row Source      | SELECT DISTINCT Suppliers.Country FROM Suppliers ORDER BY Suppliers.Country; |
| Bound Column    | 1                                                                            |
| Default Value   |                                                                              |
| Validation Rule |                                                                              |
| Validation Text |                                                                              |
| Enabled         | Yes                                                                          |
| Locked          | No                                                                           |
| Smart Tags      |                                                                              |

Formulář **frmDodavatelé** prozatím zavřeme a sestavíme dotaz **qryDV** podle následujícího obrázku:

| Field:    | Country                             | CompanyName                         | ProductName                         | UnitPrice                           | UnitsInStock                        |
|-----------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Table:    | Suppliers                           | Suppliers                           | Products                            | Products                            | Products                            |
| Sort:     |                                     |                                     |                                     |                                     |                                     |
| Show:     | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Criteria: | [Forms]![frmDodavatelé]![lst]       |                                     |                                     |                                     |                                     |

Dotaz má parametr **[forms]![frmDodavatelé]![lst]** (v české verzi lze psát místo **Forms** i **Formuláře**). Tento parametr identifikuje vyznačenou položku v seznamu zemí **lst** na našem formuláři. **Forms** je souhrn všech otevřených formulářů, **frmDodavatelé** je náš formulář a **lst** seznam různých zemí dodavatelů. V tomto kontextu však **lst** přímo identifikuje vyznačenou položku v něm.

Po zhotovení dotazu **qryDV** vytvoříme pomocí průvodce sestavou sestavu založenou na tomto dotazu. Seskupujeme v ní pro firmu. Sestavu zhotovenou průvodcem upravíme podle následujícího obrázku.



## Úkoly k procvičování:

The screenshot shows the 'rptDodavateléZeZemě' report in Microsoft Access. The report is designed with a header section containing the title 'Dodavatelé ze země' and a parameter [Forms]!frmDodavatelé!lst. Below the header is a table with columns: CompanyName, Product Name, Unit Price, and Units In Stock. The table has a detail section with a footer section. The footer section contains a summary row with the formula =Sum([UnitsInStock]) and a page footer with the formula =Page & [Page] & ' of ' & [Pages]. The report footer contains the formula =Sum([UnitsInStock]) and the text 'Jednotky na skladě celkem pro zemi'.

Budeme-li při tvorbě sestavy chtít zobrazit záznamy sestavy, spustí se podkladový dotaz a zeptá se na parametr **[Forms]![frmDodavatelé]![lst]**, neboť formulář **frmDodavatelé** není otevřen a není tedy v souhrnu **Forms**. V odpověď zadáme prozatím nějakou zemi, např. **Francie**. Všimněte si odkazu na prvek **lst** formuláře v záhlaví sestavy. V zápatí sestavy jsou dvě textová pole. V prvním je zdrojem ovládacího prvku

=**"Jednotky na skladě celkem pro zemi" & Forms!frmDodavatelé!lst**

Ve druhém je zdrojem ovládacího prvku výraz

=**Sum([UnitsInStock])**

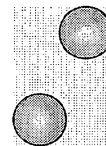
Zhotovenou sestavu uložíme jako **rptDodavateléZeZemě**.

Po zhotovení sestavy opět otevřeme formulář **frmDodavatelé** a vložíme do zápatí návrhového zobrazení tlačítko a obslužíme jeho průvodce. Události klepnutí na toto tlačítko přiřadíme průvodcem náhled sestavy **rptDodavateléZeZemě**.

Co se stane, vybereme-li v seznamu formuláře zemi a klepneme-li na tlačítko pro náhled sestavy? Před zobrazením záznamů sestavy se spustí její podkladový dotaz **qryDV** a jako hodnota jeho parametru bude dosazen název vyznačené země v seznamu **lst**. Záznamy získané jako výsledek dotazu budou poté zobrazeny v náhledu sestavy.



# Kontakty



**Počítačová škola Praha**  
Kodaňská 46, 101 00 Praha 10  
Tel: 234 064 900-3, Fax: 234 064 949

**Počítačová škola Brno**  
Josefská 8-10, 602 00 Brno  
Tel: 542 219 935-6, Fax: 542 219 938

**Internet**  
<http://www.gopas.cz>  
e-mail: [info@gopas.cz](mailto:info@gopas.cz)

