



## 1. Databáze. Relace

### 1.1 Databáze

*Databáze*

*Databáze* je souhrn dat vztahujících se k určitému tématu nebo účelu. Databázi lze chápat jako množinu dat popisujících určitou část objektivní reality, udržovanou a využívanou prostřednictvím databázového systému.

*Databázový systém*

*Databázový systém* - systém řízení báze dat (SRBD) - Database Management System (DBMS) je programové vybavení pro vytvoření a využití databázi (bázi dat). Program Microsoft Access je systém řízení relační databáze (RDBMS), který ukládá a načítá informace na základě definovaných vztahů.



*Seznam v Excelu*

Jednoduché databázové operace lze provádět již v tabulkovém programu MS Excel formou seznamů. *Seznam* je skupina řádků s daty stejného typu. Seznam je tedy jednoduchá databázová tabulka. V každém řádku je uvedena jedna *věta* (záznam), ve sloupcích jsou jednotlivá *pole* (položky). V prvním řádku jsou uvedena záhlaví sloupců, tj. *struktura věty*. Řádek se záhlavím sloupců je vhodné odlišit jiným formátováním, např. kurzívou. Řádek se záhlavím neoddelujeme volným řádkem od dat v seznamu.<sup>1</sup>

*Omezení Excelu*

Tabulkový program Excel nabízí řadu databázových operací (řazení, filtrování, souhrny, kontingenční tabulky). Excel je však nevhodný pro:

- zpracování rozsáhlých tabulek,
- zpracování dat ve více dílčích tabulkách, mezi nimiž je souvislost.

 ENC.xls  
 Deník

Pro ukázkou jsme v Excelu připravili v sešitu *Enc* na listu *Deník* dvě tabulky, týkající se prodeje programů vytvářených a prodávaných fiktivní firmou Encián:

- tabulku o prodeji: Tabulka je připravena v rozsahu A1:D166 s názvem rozsahu *Prodeje* (viz obr. 1-1). Tabulka obsahuje v prvním řádku nadpisy sloupců:
    - *Datum*: Datum prodeje programů, tj. datum vystavení faktury, kterou byl prodej realizován.
    - *Program*: Program, který byl prodán. Předpokládáme, že na jednom řádku je uveden jen jediný program. Programy jsou označeny zkratkami (*BON* – Bonita, *FIN* – Finanční analýza, *INV* – Investice, *KAU* – Kauzální analýza, *POD* – Podnikatel, *SAF* – SAFI).
    - *Počet*: Počet prodaných licencí programu. (Na jedné faktuře může být uvedeno více licencí programu, který jsme prodali stejnému zákazníkovi.)
    - *Typ*: Typ zákazníka vyjádřený jednopísmenným kódem (*B* – bankovní, *F* – finanční, *J* – jiné, *P* – průmyslové organizace).
- Tabulka obsahuje 165 vět (tj. 165 řádků bez započítání řádku s názvy sloupců).

OBR. 1-1: PRVNÍCH DESET VĚT TABULKY PRODEJE V EXCELU

	A	B	C	D
1	<i>Datum</i>	<i>Program</i>	<i>Počet</i>	<i>Typ</i>
2	2.1.2001	BON	1	B
3	2.1.2001	FIN	1	B
4	2.1.2001	SAF	1	B
5	3.1.2001	INV	1	P
6	4.1.2001	FIN	1	P
7	4.1.2001	FIN	1	P
8	4.1.2001	INV	1	P
9	5.1.2001	BON	1	P
10	5.1.2001	BON	1	P
11	5.1.2001	SAF	1	P

- tabulku o programech: Tabulka je připravena v rozsahu K168:L174 s názvem rozsahu *Programy* (viz obr. 1-2). Tabulka obsahuje v prvním řádku nadpisy sloupců:

<sup>1</sup> Podrobný výklad seznamů MS Excelu je uveden ve skriptech Kubálek, T. – Čermáková, L.: *Manažerská informatika. Tabulkový program MS Excel verze 2000 CZ. VŠE, Praha 2001. ISBN 80-245-0137-6* v kapitole 8.



- *Program*: Zkratka programu (*BON, FIN, INV, KAU, POD, SAF*)
- *Cena*: Cena za licenci programu v Kč.

OBR. 1-2: TABULKA PROGRAMY V EXCELU

	<i>K</i>	<i>L</i>
<b>168</b>	<i>Program</i>	<i>Cena</i>
<b>169</b>	BON	3800
<b>170</b>	FIN	4000
<b>171</b>	INV	2400
<b>172</b>	KAU	10000
<b>173</b>	POD	1900
<b>174</b>	SAF	19000

*Spuštění  
Accessu*

Tabulky načteme do databázového systému MS Access. Databázový systém Access spustíme z Windows 95, 98, 2000 či NT klepnutím do nabídky START a výběrem PROGRAMY, MICROSOFT ACCESS. (Programy jsou uspořádány dle abecedy.) Často je možné Access také spustit poklepnutím na tlačítko zástupce Accessu na pracovní ploše.

Objeví se dialogové okno, které nabízí tři možnosti zahájení práce s Accessem:

- Vytvořit novou databázi použitím prázdné databáze.
- Vytvořit novou databázi použitím průvodce databázi.
- Otevřít existující databázi.

Vybereme první možnost a klepneme do tlačítka **OK**.

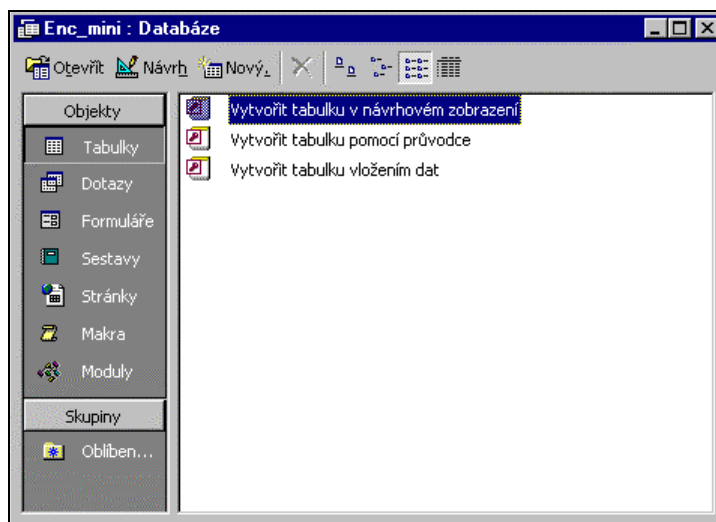


Enc\_mini.mdb

V dialogovém okně **Nový databázový soubor** vyplníme v poli *Název souboru* text *Enc\_mini*, zvolíme v poli *Uložit do* vhodnou složku (adresář) a klepneme do tlačítka **Vytvořit**.

Objeví se úvodní okno tvorby databáze (viz obr. 1-3).

OBR. 1-3: ÚVODNÍ OKNO TVORBY DATABÁZE



*Objekty  
databáze*

V okně databáze je zatím prázdný seznam *objektů* databáze, s nimiž se postupně budeme seznamovat<sup>2</sup>:

- *Tabulka* je souhrn dat týkajících se určitého objektu. Datové věty (záznamy) jsou uvedeny v řádcích, jednotlivá pole vět ve sloupcích.
- *Dotaz* je otázka, kterou se ptáme na výskyt určitých dat v databázi. Lze se ptát na obsah jedné či více tabulek spojených relacemi, výsledkem často bývá tzv. sada záznamů.

<sup>2</sup> Řádky *Vytvořit tabulku...* jsou pouze zástupci rychlé tvorby nových objektů. Zobrazování zástupců nových objektů je možné potlačit příkazem z menu NÁSTROJE, MOŽNOSTI v kartě **Zobrazit** zrušením zaškrtnutí pole *Zástupce nových objektů*.



- *Formulář* slouží k pohodlnému vkládání, opravě či prohlížení dat. V návrhu formuláře určujeme, v jaké formě chceme zobrazit data. Formulář může zobrazovat data z jedné či více tabulek.
- *Sestava* se využívá při prezentaci dat na obrazovce či na tiskárně. Do sestavy můžeme snadno zařadit souhrny skupin dat, např. dílčí a celkové součty.
- *Stránka* usnadňuje distribuci aplikací MS Accessu uživatelům, kteří mají nainstalován MS Office 2000 a nechtějí použít Access. Datové stránky zpřístupňují data z databáze Accessu formou WWW stránek.
- *Makro* je seznam akcí, které často provádíme v sekvenci za sebou. Makro umožňuje sekvenci uchovat a snadno opakovaně provádět.
- *Modul* obsahuje příkazy jazyka Visual Basic. Pomocí jazyka Visual Basic lze automatizovat mnohem složitější operace než v makrech.

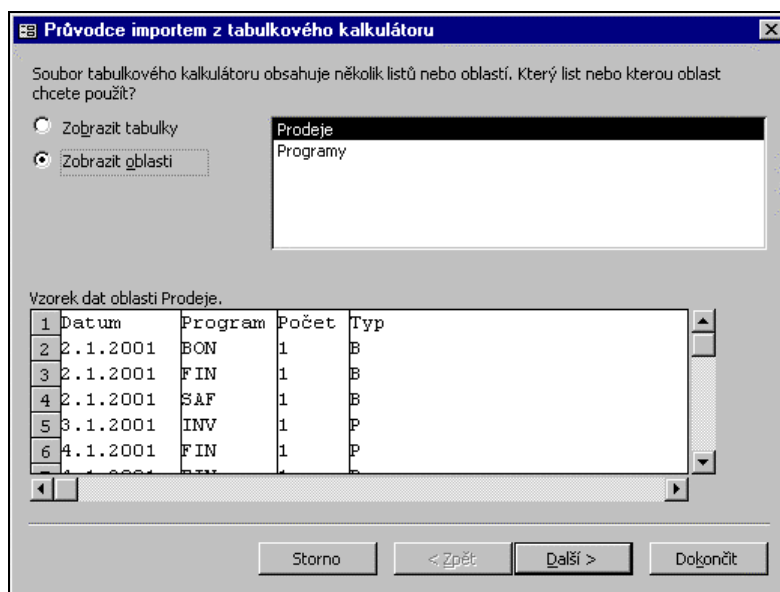
Klepnutím na zástupce typu objektu v panelu objektů v levé části databázového okna zobrazíme seznam příslušných objektů. Po vytvoření nové databáze je zobrazen seznam tabulek, později seznam, která byl aktivní při zavření databáze. V seznamu zatím žádná tabulka není uvedena.

Načtení  
tabulky Excelu

Tabulku Excelu načteme do databáze následujícím postupem:

- Z menu zadáme příkaz SOUBOR, NAČÍST EXTERNÍ DATA, IMPORT.
- V dialogovém okně **Import** v poli *Soubory typu* vybereme *Microsoft Excel* a ze souborů v příslušném adresáři vybereme soubor *Enc* (.xls) a klepneme do tlačítka **Import** nebo poklepeme na název souboru.
- V dialogovém okně **Průvodce importem z tabulkového kalkulátoru** vybereme *Zobrazit oblasti* (viz obr. 1-4), neboť budeme čerpat data z nazvaných oblastí sešitu Excelu.

OBR. 1-4: PRŮVODCE IMPORTEM Z TABULKOVÉHO KALKULÁTORU



Prodeje

V pravé části nám *Průvodce* nabízí pojmenované oblasti sešitu *Enc*. Ponecháme vybrané *Prodeje*. Ve spodní části okna je uveden vzorek oblasti *Prodeje*, abychom si ověřili, že vybíráme zamýšlenou oblast. Klepneme do tlačítka **Další**.

- V druhém dialogovém okně *Průvodce* je zaškrtnuto pole *První řádek obsahuje hlavičky sloupců*. Klepneme do tlačítka **Další**.
- Ve třetím dialogovém okně je zaškrtnuto pole *Do nové tabulky*, neboť data nedoplňujeme do stávající tabulky, ale vytváříme novou tabulku. Klepneme do tlačítka **Další**.
- Ve čtvrtém dialogovém okně můžeme upravit názvy načítaných sloupců tabulky či některé sloupce označit jako neimportované. Klepneme do tlačítka **Další**.
- V pátém dialogovém okně se nabízí možnost:
  - *Primární klíč přidá aplikace Access*.



– Vlastní primární klíč.

– Neurčovat primární klíč.

Význam primárních klíčů poznáme v další části této kapitoly. Vybereme poslední možnost *Neurčovat primární klíč*. Klepneme do tlačítka **Další**.

– V posledním dialogovém okně *Průvodce* ponecháme název importované tabulky dle názvu rozsahu *Prodeje* a nezaškrtneme pole *Analyzovat strukturu tabulky po importu dat* ani pole *Zobrazit nápovědu po ukončení průvodce*. Klepneme do tlačítka **Dokončit**.

– Objeví se dialogové okno s hlášením *Ukončen import souboru ... do tabulky Prodeje*. Klepneme do tlačítka **OK**.



Programy

Obdobně načteme tabulku *Programy* ze sešitu *Enc*. V pátém dialogovém okně vybereme možnost *Vlastní primární klíč* a ponecháme nabízený primární klíč *Program*. Primární klíč je sloupec (pole), jehož hodnota jednoznačně identifikuje řádek (větu) tabulky. Později poznáme, že primární klíč má velký význam pro práci s tabulkami a relacemi mezi nimi.

Přehled objektů

V přehledu objektů tabulek jsou již uvedeny dvě tabulky. Přehled může být zobrazen čtyřmi různými způsoby, které lze volit klepnutím do příslušných tlačítek v nabídce (prvním řádku) okna *Databáze*<sup>3</sup>:



– *Velké ikony*: Každý objekt (tabulka) je zastoupen ikonou, pod níž je vypsán jeho název. Pokud je název delší, je vypsán pouze jeho začátek. Umístění ikon lze měnit.



– *Malé ikony*: Každý objekt je zastoupen malou ikonou, vedle níž je vypsán jeho název. Ikony s názvy jsou rozmístěny v několika sloupcích v pořadí po řádcích. Umístění ikon lze měnit.



– *Seznam*: Každý objekt je zastoupen malou ikonou, vedle níž je vypsán jeho název. Ikony s názvy jsou rozmístěny v několika sloupcích v pořadí po sloupcích. Umístění ikon nelze měnit.



– *Detaily*: Každý objekt je vypsán na zvláštním řádku. Kromě ikony a názvu jsou uvedeny vlastnosti objektu: popis (popis lze zadat klepnutím pravým tlačítkem myši do objektu, příkazem *VLASTNOSTI* a vyplněním pole *Popis* v dialogovém okně **Vlastnosti objektu**, např. pro tabulku *Prodeje* zadáme popis *převzato z Excelu*), datum změny, datum vytvoření, typ objektu (tabulka)<sup>4</sup>. Šířku jednotlivých sloupců vlastností lze měnit tažením za hranice mezi sloupci v řádku jejich názvů, tj. obdobně jako např. měníme šířku sloupců v tabulce Wordu či Excelu.

Ponecháme zobrazení ve formě seznamu.

Datový list

Zkontrolujeme nyní, jak se tabulky načetly. Poklepáme na název objektu *Prodeje*. Tabulka *Prodeje* se zobrazí v datovém listu v samostatném okně (viz obr. 1-5). Sloupce jsou správně nadepsány. (Nadpisy sloupců tvoří voliče sloupců, tj. je možné označit sloupec klepnutím do nadpisu sloupce.) Poklepáním na hranice sloupců zoptimalizujeme šířku sloupců. V posledním řádku okna je vypsána poloha kurzoru (v obr. 1-5 je kurzor na 10. větě), na konci řádku je zrekapitulován celkový počet vět v tabulce. Tlačítka vedle pořadí vět mají běžný význam, zleva:

- na první větu,
- na předchozí větu,
- na následující větu,
- na poslední větu,
- přidat novou větu.

Aktuální větu poznáme také podle trojúhelníku v tzv. voliči záznamů, tj. v šedém sloupci na levém kraji okna. Klepnutím do uzavíracího tlačítka (v pravém horním rohu) zavřeme okno

OBR. 1-5: PRODEJE V DATOVÉM LISTU

Datum	Program	Počet	Typ
2.1.2001	BON	1	B
2.1.2001	FIN	1	B
2.1.2001	SAF	1	B
3.1.2001	INV	1	P
4.1.2001	FIN	1	P
4.1.2001	FIN	1	P
4.1.2001	INV	1	P
5.1.2001	BON	1	P
5.1.2001	BON	1	P
5.1.2001	SAF	1	P

<sup>3</sup> Způsoby zobrazení lze také měnit příkazem z menu *ZOBRAZIT* nebo z místní nabídky vyvolané klepnutím pravým tlačítkem myši do bílé plochy pod seznamem objektů a volbou *ZOBRAZIT*.

Ve všech způsobech zobrazení můžeme měnit pořadí objektů příkazem z menu *ZOBRAZIT*, *USPOŘÁDAT IKONY* nebo z místní nabídky volbou *USPOŘÁDAT IKONY*.

<sup>4</sup> Abychom viděli všechny vlastnosti objektu, bude možná nutné využít vodorovný posuvník okna *Databáze*.

Návrhové  
zobrazení

datového listu. Vzhledem ke změnám šířky sloupců jsme nejprve dotázáni, zda uložíme změny v rozvržení objektu tabulka *Prodeje*. Klepneme do tlačítka **Ano**.

Obdobně zkontrolujeme tabulku *Programy*. V datovém listu vidíme obsah tabulky. Každý sloupec má v databázovém systému své vlastnosti. Definici struktury tabulky a vlastností sloupců uvidíme v návrhovém zobrazení. Mezi datovým listem a návrhovým zobrazením se lze přepínat tlačítkem **Zobrazit**, které má různý vzhled v datovém listu (pro přepínání do návrhového zobrazení) a v návrhovém zobrazení (pro přepínání do datového listu). Okno datového listu či návrhového zobrazení obsahuje vlastní panely nástrojů (*Datový list tabulky*, *Návrh tabulky*).

Klepnutím do tlačítka **Zobrazit** zobrazíme okno návrhového zobrazení (viz obr. 1-6).

OBR. 1-6: NÁVRHOVÉ ZOBRAZENÍ TABULKY PROGRAMY

V návrhovém zobrazení si všimneme, že na začátku řádku *Program* je zobrazen obrázek klíče signalizující primární klíč. O vlastnostech polí se více dozvíme v kap. 2. Klepnutím do tlačítka **Zobrazit** se vrátíme do datového listu a poklepáním na konce označení sloupců (voličů sloupců) zoptimalizujeme šířky sloupců. Klepnutím do uzavíracího tlačítka a potvrzením uložení změn uzavřeme tabulku *Programy*.



Pro úplnost dodejme, že příslušné zobrazení tabulky označené v přehledu objektů tabulek můžeme získat také:

- klepnutím do tlačítka **Otevřít** (tabulka se zobrazí v datovém listu) nebo
- klepnutím do tlačítka **Návrh** (tabulka se zobrazí v návrhovém zobrazení).



## 1.2 Relace

Tabulky jsou v databázi zatím izolované bez vzájemné souvislosti. Rádi bychom ke každé faktuře vypsali cenu programu z tabulky *Programy*. Nerozšíříme tabulku *Prodeje* o sloupec *Cena*, neboť:

- Při aktualizaci ceny bychom museli cenu opravit ve všech řádcích tabulky *Prodeje*, v nichž je uveden program, jehož cena se mění.
- Pro nové prodeje bychom vždy museli vyhledávat a zapisovat cenu programu.
- Ceny by byly uvedeny v tabulce opakovaně, čímž by se zbytečně zvětšovala tabulka *Prodeje* a zvyšovala se pravděpodobnost chybného vyplnění dat.

Dotazy

Kombinování dat z více tabulek umožňují tzv. dotazy, které plní řadu dalších funkcí. Klepneme do zástupce *Dotazy*. V seznamu dotazů není zatím uveden žádný dotaz.

Nový objekt vytvoříme klepnutím do tlačítka **Nový**. (Tabulky jsme zatím připravovali méně obvyklou cestou načtením z jiného programu.) Nevyužijeme průvodce tvorbou dotazu, proto v dialogovém okně **Nový dotaz** ponecháme zvolenu možnost *Návrhové zobrazení* a klepneme do tlačítka **OK**.



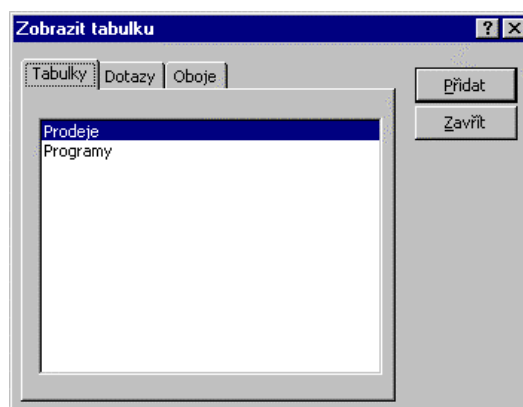


V dialogovém okně **Zobrazit tabulku** (viz obr. 1-7) vybereme poklepáním (nebo klepnutím do názvu tabulky a potom do tlačítka **Přidat**) tabulky, které budou zařazeny do dotazu, tj. v našem případě tabulky *Prodeje* a *Programy*.

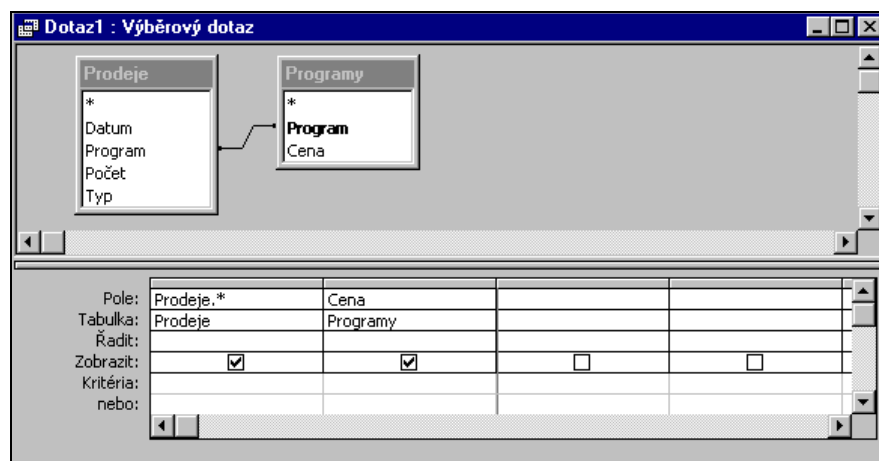
Klepnutím do tlačítka **Zavřít** zavřeme okno **Zobrazit tabulku**. Návrhové zobrazení dotazu se skládá ze dvou částí (viz obr. 1-8, kde je již návrh dotazu dokončen):

- V horní části je přehled tabulek, z nichž dotaz čerpá data, včetně relací mezi tabulkami.
- V dolní části je návrhová mřížka dotazu, do níž budeme vybírat z horní části pole tabulek, popř. upřesňovat kritéria výběru, řazení a další.

OBR. 1-7: DIALOGOVÉ OKNO  
ZOBRAZIT TABULKU



OBR. 1-8: OKNO NÁVRHU DOTAZU



Access sám vytypoval souvislost mezi tabulkami prostřednictvím pole *Program*. Souvislost vztahu (tzv. relace) je zakreslena v horní části čarou spojující příslušná pole tabulek. Z tabulky *Prodeje* vybereme do výsledku všechna pole poklepáním na první řádek (s hvězdičkou). V tabulce *Programy* v horní části poklepeme pouze na pole *Cena*. Výsledky poklepání jsou patrné v návrhové mřížce ve spodní části návrhu dotazu (viz obr. 1-8).

Pokud by související pole nebyla stejně nazvána, Access by nepoznal souvislost, tj. nepřipravil by relaci mezi tabulkami *Prodeje* a *Ceny*. Předvedeme si tuto situaci. Klepneme do relace mezi tabulkami (bude vyznačena tučně) a relaci klávesou **Delete** odstraníme.

Klepnutím do tlačítka **Zobrazit** zobrazíme datový list výsledku dotazu. Výsledkem je tzv. sada záznamů uspořádaná stejně jako běžná tabulka<sup>5</sup>. Access zobrazil veškeré kombinace 165 vět tabulky *Prodeje* a 6 vět tabulky *Programy*, např. k první větě z tabulky *Prodeje* postupně připojoval 6 vět tabulky *Programy* bez ohledu na souvislost vět. Sadu tvoří  $165 * 6 = 990$  záznamů.

Nás však zajímají pouze kombinace, v nichž se shoduje pole *Program* z tabulky *Prodeje* i z tabulky *Programy*. Obnovíme proto relaci mezi tabulkami. Klepnutím do tlačítka **Zobrazit** se vrátíme do návrhového zobrazení. Ukazatelem myši se přiblížíme poli *Program* v tabulce *Programy* v horní části okna návrhu, stiskneme levé tlačítko myši a tažením myši (tj.

Datový list  
výsledku



Definice relace  
v dotazu



<sup>5</sup> Později poznáme, že sady záznamů jsou dvou typů: dynamické sady, které lze editovat, a snímky, v nichž data nelze upravovat. V našem případě se jedná o snímek.





pohybem myši při držení levého tlačítka) přetáhneme pole *Program* z tabulky *Programy* na pole *Program* v tabulce *Prodeje*. Mezi tabulkami se vytvoří relace.

Klepnutím do tlačítka **Zobrazit** zobrazíme datový list výsledku dotazu. Výsledná sada záznamů má nyní 165 záznamů, neboť relací jsme omezili výsledné kombinace. Ke každé faktuře jsou vybrány z tabulky *Programy* pouze věty se stejnou hodnotou pole *Program*. Vzhledem k tomu, že pole *Program* je v tabulce *Programy* primárním klíčem, existuje ke každé větě z tabulky *Prodeje* nanejvýš jedna věta v tabulce *Programy*. Sadu tvoří  $165 * 1 = 165$  záznamů.

#### Uložení dotazu



12 Ceny

Klepnutím na uzavírací tlačítko končíme práci s dotazem. Access se dotáže, zda chceme změny do návrhu objektu dotaz uložit. Klepneme do tlačítka **Ano**. (Klepnutím do tlačítka **Ne** by se dotaz uzavřel bez uložení. Klepnutím do tlačítka **Storno** bychom se vrátili do dotazu.) Access nás vyzve k vyplnění názvu dotazu. Nabídnutý název *Dotaz1* přepíšeme na *S12 Ceny* a klepneme do tlačítka **OK**. V přehledu objektů dotazů je uveden první dotaz s názvem *S12 Ceny*.<sup>6</sup>

Access umožňuje definovat relace souhrnně pro celou databázi tak, abychom relace nemuseli upřesňovat v jednotlivých dotazech. (V našem případě úprava vzhledem k stejnému názvu propojujících polí nebyla nutná. Uvidíme však, že při definování relací v databázi můžeme doplnit další vlastnosti relací nedefinovatelné v návrhu dotazu.) Relace jsou společné pro celou databázi, nemohou existovat ve více variantách, proto nemají vyhrazenou svou záložku v přehledu objektů Accessu.

#### Relace



Relace v databázi zobrazíme zadáním příkazu z menu **NÁSTROJE**, **RELACE** nebo klepnutím do tlačítka **Relace**. V dialogovém okně **Zobrazit tabulku** (viz obr. 1-7) vybereme obě dvě tabulky *Prodeje* a *Programy*. Práci s dialogovým oknem ukončíme klepnutím do tlačítka **Zavřít**. Tažením myši přesuneme pole *Program* z tabulky *Programy* na pole *Program* v tabulce *Prodeje*. Zobrazí se dialogové okno **Upravit relace** (viz obr. 1-9).

V horní části dialogového okna je zrekapitulována konstrukce relace. Obecně může být požadována shoda polí s různými názvy a shoda více dvojic polí, proto horní tabulka obsahuje více řádků.

#### Referenční integrita

V rámci definice relace nejen stanovíme spojení (souvislost) tabulek, ale také zajistíme tzv. referenční integritu.

Referenční integrita je systém pravidel, která se využívají k zajištění platnosti relací mezi záznamy vázaných tabulek. Referenční integrita slouží k zamezení nechtěného odstranění nebo změny souvisejících dat. Referenční integritu lze nastavit pouze tehdy, jsou-li splněny následující podmínky:

- Odpovídající pole primární tabulky je primárním klíčem nebo má jedinečný index<sup>7</sup>.
- Odpovídající pole jsou shodného datového typu<sup>8</sup>.
- Obě tabulky jsou součástí stejné databáze.

Zaškrtnutím pole *Zajistit referenční integritu* zajistíme následující omezení editace databáze:

OBR. 1-9: DIALOGOVÉ OKNO UPRAVIT RELACE

<sup>6</sup> Název objektů s výjimkou tabulek budeme zahajovat písmenem S (skripta), číslem kapitoly a podkapitoly, tj. v kap. 1.2 vytváříme objekt *S12 Ceny*. Bude-li v kapitole objektů více, odlišíme je malým písmenem za číslem podkapitoly, např. *S31a Dotaz*.

<sup>7</sup> Výklad jedinečného indexu je zařazen do kap. 2.

<sup>8</sup> Datový typ jsme poznali při prohlížení struktury tabulky. Např. pole *Program* je typu *Text*. Existují dvě výjimky. Relace může být definována mezi polem typu *Automatické číslo* a polem typu *Číslo*, jehož vlastnost *Velikost pole* je nastavena na hodnotu *Dlouhé celé číslo*. Druhou výjimkou je relace mezi polem typu *Automatické číslo* a polem typu *Číslo*, pokud je vlastnost *Velikost pole* obou polí nastavena na hodnotu *Replikační identifikátor*.



- Do tabulky *Prodeje* nebude možné zapsat prodej programu, který není uveden v tabulce *Programy*. (Bude nanejvýš možné ponechat pole *Program* nevyplněné.)
- V tabulce *Programy* nebude možné opravit hodnotu pole *Program* pro program, který se vyskytuje v tabulce *Prodeje*.
- V tabulce *Programy* nebude možné odstranit větu s programem, který se vyskytuje v tabulce *Prodeje*.

Druhé dvě podmínky lze modifikovat:

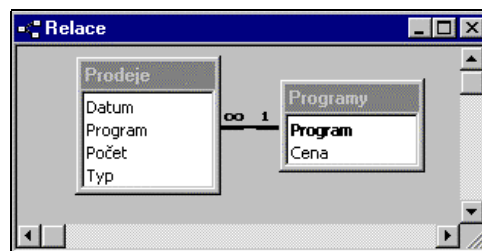
- Po zaškrtnutí pole *Aktualizace souvisejících polí v kaskádě* se změnou pole *Program* v tabulce *Programy* změní automaticky všechny související hodnoty pole *Program* v tabulce *Prodeje*, např. změníme-li značku programu BON v tabulce *Programy* na BONI, změní se značky programů Bonity v celé tabulce *Prodeje* na BONI.
- Po zaškrtnutí pole *Odstranění souvisejících polí v kaskádě* se odstraněním věty z tabulky *Programy* odstraní všechny prodeje tohoto programu.

V našem případě zaškrtneme pouze pole *Aktualizace souvisejících polí v kaskádě*.

Klepnutím do tlačítka **Vytvořit** vytvoříme relaci. Tažením za název tabulky v okně relací můžeme tabulku přesouvat za účelem zpřehlednění schématu relací. Vidíme, že referenční integrita je vyznačena ve schématu symboly 1:∞ (viz obr. 1-10). Access na základě primárního klíče v tabulce *Programy* poznal, že se jedná o relaci typu 1:N, tj. k jednomu programu může existovat více prodejů, avšak k jednomu prodeji existuje jediný program.

Klepnutím do uzavíracího tlačítka uzavřeme okno **Relace**. Potvrdíme klepnutím do tlačítka **Ano**, že chceme uložit změny rozvržení relací. K úpravám relací se můžeme později vrátit příkazem **NÁSTROJE, RELACE** nebo klepnutím do tlačítka **Relace**.

OBR. 1-10: RELACE TYPU 1:N



### 1.3 Databáze Per\_mini a Personal

Nová databáze



Per\_mini.mdb

Ukažme si tvorbu relací na jiném příkladu. Připravíme databázi *Personal* s evidencí zaměstnanců firmy Encián. Vytvoříme novou databázi příkazem **SOUBOR, NOVÝ** nebo klepnutím do tlačítka **Nový** nebo kombinací kláves **Ctrl+N**. V dialogovém okně **Nová** vybereme z nabízených šablon šablonu *Databáze z karty Obecné*. V dialogovém okně **Nový databázový soubor** vybereme v poli *Uložit do* vhodný adresář a v poli *Název souboru* zadáme název *Per\_mini*. Klepneme do tlačítka **Vytvořit**.

Tabulky pro databázi *Personal* nepřevzeme ze sešitu Excelu na rozdíl od předchozí databáze *Enc\_mini*. Jejich strukturu i obsah vytvoříme v Accessu. Definováním struktury tabulky se budeme podrobně zabývat v kap. 2. Zatím tabulky zjednodušíme:

- uvedením jen některých polí ve struktuře tabulek,
- zjednodušením datového typu všech polí na textový typ,
- zapsáním jen ilustrativních vět do tabulek, tj. redukcí obsahu tabulek.



Personal

Jádrem databáze *Per\_mini* (kterou později přejmenujeme na *Personal*) bude tabulka *Personal*. Název tabulky se může shodovat s názvem databáze. Připravíme nejprve strukturu tabulky. V rámci objektů *Tabulky* klepneme do tlačítka **Nový**. V dialogovém okně **Nová tabulka** vybereme možnost *Návrhové zobrazení* a klepneme do tlačítka **OK**.

Strukturu tabulky definujeme ve speciální tabulce o třech sloupcích:

- **Název pole:** Tabulka *Personal* bude obsahovat tato pole:
  - *E\_mail:* Obsahem pole bude elektronická adresa zaměstnance. Každý zaměstnanec má v rámci firmy jednoznačnou elektronickou adresu. (Nebudeme sem zapisovat doménu @encian.cz.) Elektronická adresa bude jednoznačně identifikovat zaměstnance, bude tedy primárním klíčem tabulky.
  - *Příjmení.*
  - *Kancelář:* Kancelář, v níž zaměstnanec sídlí. U externích zaměstnanců nemusí být kancelář vyplněna.





- *Pracoviště*: Firma Encián se člení na několik pracovišť. Každé pracoviště má kromě názvu jednoznačnou třípísmennou zkratku, kterou budeme zapisovat do tohoto pole.
- *Datový typ*: Pro všechna pole ponecháme typ text, vlastnosti ve spodní části tabulky ponecháme dle návrhu Accessu.
- *Popis*: Vyplnění sloupce je nepovinné, zatím jej ponecháme prázdný.



Vyplníme nejprve název prvního pole *E\_mail*. Odešleme klávesou **Enter**. Klepnutím do tlačítka **Primární klíč** definujeme první pole jako primární klíč. Odešleme na nabízeném datovém typu text a odešleme i prázdný popis. Obdobně (již bez definování primárního klíče) zapíšeme ostatní pole struktury (viz obr. 1-11).

OBR. 1-11: VÝCHOZÍ STRUKTURA TABULKY PERSONAL

	Název pole	Datový typ	Popis
	E_mail	text	
	Příjmení	text	
	Kancelář	text	
	Pracoviště	text	

Vlastnosti pole

Obsah:

Obecné | Vyhledávání

Velikost pole: 50

Formát:

Vstupní maska:

Titulek:

Výchozí hodnota:

Ověřovací pravidlo:

Ověřovací text:

Je nutno zadat: ne

Povolit nulovou délku: ne

Indexovat: ne

Kompresce kódu Unicode: ano

Název pole může být dlouhý nejvýše 64 znaků včetně mezer. Chcete-li získat informace o názvech polí, stiskněte klávesu F1.

Klepneme do uzavíracího tlačítka a na dotaz *Chcete uložit změny do návrhu objektu tabulka Tabulka1?* klepneme do tlačítka **Ano**. Accessem nabízený pracovní návrh názvu tabulky *Tabulka1* v dialogovém okně **Uložit jako** přepíšeme na *Personal*.

OBR. 1-12: VÝCHOZÍ OBSAH TABULKY PERSONAL

E_mail	Příjmení	Kancelář	Pracoviště
Novak.Petr	Novák	31	RED
Sladka.Marie	Sladká	32	RED
Kolinska.Eva	Kolínská	23	PRG
Pospisilova.Alena	Pospíšilová	25	PRG
Svetly.Ondrej	Světlý		PRG

Záznam: 5 z 5

V přehledu objektů tabulek se již objevila tabulka *Personal*. Poklepáme na její název a otevřeme tak její zobrazení v datovém listu. Dle obr. 1-12 vyplníme obsah tabulky. Poklepáním na hranice sloupců zoptimalizujeme jejich šířku.

Klepneme do uzavíracího tlačítka a na dotaz *Chcete uložit změny v rozvržení objektu tabulka Personal?* klepneme do tlačítka **Ano**. Access ukládá obsah každé věty ihned po přechodu na další větu. Ukládací dotaz se vztahoval k uložení změn šířek sloupců pro zobrazení v datovém listu. Znovu poklepeme na tabulku *Personal*. Access si pamatuje upravené šířky sloupců. Navíc tabulku zobrazuje v řazení dle primárního klíče, tj. v pořadí vět Kolinska.Eva, Novak.Petr, Pospisilova.Alena, Sladka.Marie, Svetly.Ondrej. Zavřeme tabulku *Personal*.



Telefony

K tabulce *Personal* připravíme vysvětlující tabulky *Telefony* a *Pracoviště*. V tabulce *Telefony* bude ke každé kanceláři uvedena nanejvýš jedna místní telefonní linka, v tabulce *Pracoviště* budou slovně vysvětleny kódy pracovišť z tabulky *Personal*. Vysvětlující tabulky nám nejen umožní vypsát ve výsledcích dotazů vysvětlující texty, ale také vytvoří předpoklad pro definici referenční integrity tak, abychom nezapsali číslo kanceláře, která není uvedena v tabulce *Telefony*, či abychom nepoužili v tabulce *Personal* kód neuvedený ve vysvětlující tabulce *Pracoviště*.

Tabulku *Telefony* připravíme jiným způsobem než tabulku *Personal*. Vložíme nejprve její obsah a potom upřesníme strukturu. V rámci objektů *Tabulky* klepneme do tlačítka **Nový**. V dialogovém okně **Nová tabulka** vybereme možnost *Zobrazení datového listu* a klepneme do tlačítka **OK**. Access připraví datový list o 21 řádcích a 10 sloupcích. Do datového listu zapíšeme obsah dle obr. 1-13. (Do kanceláře 25 není zaveden telefon.)

OBR. 1-13: VÝCHOZÍ OBSAH TABULKY TELEFONY

Pole1	Pole2	Pole3	Pole4
23	809		
25			
31	812		
32	813		

Klepneme pravým tlačítkem myši do nadpisu *Pole1*. Z místní nabídky vybereme volbu **PŘEJMENOVAT SLOUPEC**. Kurzor se umístí do nadpisu *Pole1*, který přepíšeme na *Kancelář* a odešleme. Druhý sloupec přejmenujeme jiným způsobem. Poklepáme na nadpis *Pole2*, přepíšeme jej na *Telefon* a odešleme.



Chceme povýšit pole *Kancelář* na primární klíč. Klepnutím do tlačítka **Zobrazit** chceme přejít do návrhového zobrazení. V dialogovém okně **Uložit jako** upřesníme název tabulky *Telefony*. Access zobrazí okno s varováním, že není definován primární klíč. Na dotaz *Chcete teď vytvořit primární klíč?* odpovíme klepnutím do tlačítka **Ne** (jinak by Access sám doplnil nové pole primárního klíče). V návrhovém zobrazení doplníme k poli *Kancelář* primární klíč klepnutím do tlačítka **Primární klíč**. Práci s tabulkou ukončíme klepnutím do uzavíracího tlačítka a uložením změn v návrhu. Poklepáním na tabulku *Telefony* se můžeme přesvědčit, že Access již zobrazí jen vyplněné sloupce a řádky. Tabulku uzavřeme.



Pracoviště

Obdobně dle obrázku obr. 1-14 připravíme strukturu a obsah tabulky *Pracoviště*. Záměrně opomineme nadefinovat primární klíč a zapsat pracoviště *RED*.

OBR. 1-14: VÝCHOZÍ STAV TABULKY PRACOVISTĚ

Pracoviště	Pracoviště_v
PRG	programování

Vidíme, že název pole se může shodovat s názvem tabulky. Název *Pracoviště\_v* vypovídá, že v poli je vysvětlení kódu pracoviště.

Relace



Nyní budeme definovat relace a referenční integritu v databázi. Z menu zadáme **NÁSTROJE, RELACE** nebo klepneme do tlačítka **Relace**. V dialogovém okně **Zobrazit tabulku** klepneme do tabulky *Personal*, potom za držení klávesy **Ctrl** klepneme do tabulky *Telefony*. Označili jsme tak dvě tabulky. (Opakovaným klepnutím za držení klávesy **Ctrl** bychom mohli označení tabulky zrušit. Klepnutím s držením klávesy **Shift** bychom mohli označit souvislý rozsah tabulek.) Klepnutím do tlačítka **Přidat** přidáme vybrané tabulky do schématu relací a zavřeme okno **Zobrazit tabulku** tlačítkem **Zavřít**.



Uvědomili jsme si, že jsme opomněli přidat tabulku *Pracoviště*. Klepneme do tlačítka **Přidat tabulku**. Vybereme v dialogovém okně **Zobrazit tabulku** opomenutou tabulku *Pracoviště*. Klepneme do tlačítka **Přidat** a **Zavřít**.

Definujeme relace mezi tabulkami. Nejprve táhneme pole *Kancelář* z tabulky *Telefony* na pole *Kancelář* v tabulce *Personal*. V dialogovém okně **Upravit relace** zaškrtneme pole *Zajistit referenční integritu* a *Aktualizace souvisejících polí v kaskádě* a klepneme do tlačítka **Vytvořit**. Vypíše se hlášení *Relace musí být ve stejném počtu polí, se stejnými typy dat*. Klepnutím do tlačítka **OK** potvrdíme hlášení. Tlačítkem **Storno** ukončíme práci s dialogovým oknem **Upravit relace**.

Při tvorbě tabulky *Telefony* v zobrazení datového listu Access odhadl datový typ polí v tabulce na číselný, pole *Kancelář* musí být však v tabulkách *Personal* i *Telefony* stejného typu. Z okna relací můžeme přímo upravovat strukturu zobrazených tabulek. Klepneme na tabulku *Telefony* pravým tlačítkem. Z místní nabídky vybereme **NÁVRH TABULKY**. V prvním řádku návrhu tabulky *Telefony* změníme datový typ pole *Kancelář* na text. Klepneme do uzavíracího tlačítka a odsouhlasíme uložení změn. Nyní již můžeme úspěšně zopakovat přípravu relace tabulek *Telefony* a *Personal* včetně referenční integrity a aktualizace souvisejících polí v kaskádě.

Pokračujeme definováním relace mezi tabulkami *Pracoviště* a *Personal*. Táhneme pole *Pracoviště* z tabulky *Pracoviště* na pole *Pracoviště* v tabulce *Personal*. V dialogovém okně **Upravit relace** zaškrtneme pole *Zajistit referenční integritu* a *Aktualizace souvisejících polí v kaskádě* a klepneme do tlačítka **Vytvořit**. Vypíše se hlášení *Nenalezen žádný jedinečný index pro odkazované pole primární tabulky*. Klepnutím do tlačítka **OK** potvrdíme hlášení. Tlačítkem **Storno** ukončíme práci s dialogovým oknem **Upravit relace**.

Odpovídající pole primární tabulky (*Pracoviště*) není primárním klíčem, ani nemá jedinečný index, proto nelze zavést referenční integritu. Klepneme na tabulku *Pracoviště* pravým tlačítkem. Z místní nabídky vybereme **NÁVRH TABULKY**. Hned v prvním řádku nadefinujeme pole *Pracoviště* jako primární klíč klepnutím do tlačítka **Primární klíč**. Klepneme do uzavíracího tlačítka a odsouhlasíme uložení změn.

Znovu se pokusíme připravit relaci včetně referenční integrity. Tentokrát se vypíše hlášení *Program Microsoft Access není schopen vytvořit tento vztah a zajistit referenční integritu. Data v tabulce Personal poškozuji pravidla referenční integrity*. Access kontroluje referenční integritu již při přípravě relací. V tabulce *Personal* jsou uvedeni zaměstnanci z pracoviště *RED*, které není uvedeno v tabulce *Pracoviště*. Tlačítkem **OK** uzavřeme chybového hlášení. Pracoviště *RED* doplníme do tabulky. Klepneme na tabulku *Pracoviště* pravým tlačítkem myši. Z místní nabídky vybereme **NÁVRH TABULKY**. Tlačítkem **Zobrazit** přejdeme do zobrazení datového listu. Doplníme pracoviště *RED* s názvem *ředitelství*. Tabulku *Pracoviště* uzavřeme. Nyní již úspěšně doplníme relaci včetně referenční integrity a aktualizace souvisejících polí v kaskádě.

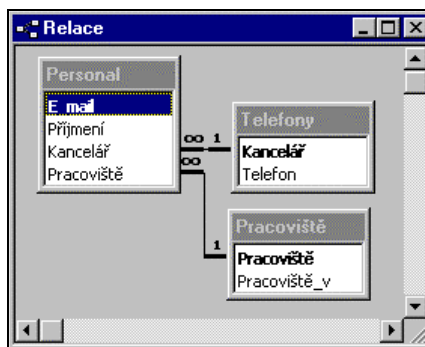
Tažením za názvy tabulek uspořádáme tabulky v relacích přehledně dle obr. 1-15. (Tažením za spodní okraj můžeme měnit velikost seznamu polí tabulek, což je někdy vhodné pro tabulky se složitou strukturou.) Tažením za pravý dolní roh zmenšíme okno relací na minimum. Klepneme na ukončovací tlačítko a potvrdíme uložení změn rozvržení relací.

Pokud bychom chtěli schéma relací zjednodušit, můžeme klepnout na seznam polí některé tabulky, např. *Telefony*, a zobrazení tabulky v schématu odstranit klávesou **Delete**. Tím neodstraníme dříve připravené relace. Po klepnutí do seznamu polí tabulky (která byla v relaci s tabulkou smazanou v relaci) můžeme její přímé relace zobrazit klepnutím do tlačítka **Zobrazit přímé relace**. Zobrazí se všechny přímé relace z dané tabulky včetně seznamu polí tabulek zařazených do těchto relací. Tlačítkem **Zobrazit všechny relace** zobrazíme všechny dříve definované relace (a současně také seznamy polí tabulek zařazených do těchto relací). Pro úplnost dodejme, že pokud je tabulka obsažena v relacích (byť není zobrazena v schématu relací), Access nás upozorní na odstraňování relací před odstraněním tabulky.





OBR. 1-15: VÝCHOZÍ RELACE V DATABÁZI PER\_MINI



Ověřme si, že se relace uplatní při návrhu dotazu. Chceme k zaměstnancům doplnit telefonní čísla. V přehledu objektů klepneme do zástupce *Dotazy* a do tlačítka *Nový*. Z nabídky tvorby dotazů vybereme *Návrhové zobrazení*. V dialogovém okně **Zobrazit tabulku** vybereme poklepáním tabulky *Personal* a *Telefony*. Dialogové okno **Zobrazit tabulku** zavřeme klepnutím do tlačítka **Zavřít**. Poklepáním vybereme z horní části návrhu dotazu pole *Příjmení* z tabulky *Personal* a *Telefon* z tabulky *Telefony*. Klepnutím do tlačítka **Zobrazit** zobrazíme datový list dotazu. Výsledkem jsou pouze čtyři věty, u pátého zaměstnance (Ondřeje Světlého) neexistuje propojení tabulek *Personal* a *Telefony*, neboť zaměstnanec nemá vyplněno pole *Kancelář*. Referenční integrita zabránila zapsat do tabulky *Personal* kancelář neexistující v tabulce *Telefony*, připouští však nevyplnění pole *Kancelář*, což se u jedné věty stalo.



Typ spojení

V relaci můžeme změnit tzv. typ spojení. Vrátime se do návrhového zobrazení dotazu a poklepeme na relaci mezi tabulkami. V dialogovém okně **Vlastnosti spojení** (viz obr. 1-16) vybereme vlastnost *Zahrnout všechny záznamy z tabulky Personal a z tabulky Telefony pouze ty záznamy, ve kterých jsou spojená pole shodná*. V datovém listu již bude pět vět. Do relace se doplnila šipka směrem k tabulce *Telefony*. Klepneme na uzavírací tlačítko. Dotaz neuložíme.

Abychom nemuseli typ spojení měnit v každém dotazu, upravíme jej přímo v relacích. Klepneme do tlačítka **Relace**. Poklepáním na relaci mezi tabulkami *Personal* a *Telefony* se zobrazí v schématu relací dialogové okno **Upravit relace** (viz obr. 1-9). V něm klepneme do tlačítka **Typ spojení**. Potom se objeví dialogové okno **Vlastnosti spojení**, v němž upravíme typ spojení tak, aby vypsal všechny věty z tabulky *Personal*. Ve schématu relací se doplní šipka směrem k tabulce *Telefony* dle obr. 1-17.

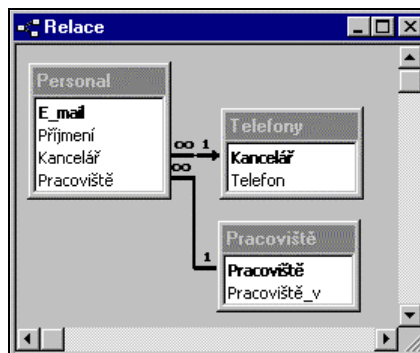


Cesty

Zaměstnanci firmy Encián jezdí na služební cesty. Jejich evidenci zajistíme v tabulce *Cesty*. Tabulku vytvoříme v zobrazení datového listu. Obsah vět a názvy polí vložíme dle obr. 1-18.

OBR. 1-16: DIALOGOVÉ OKNO VLASTNOSTI SPOJENÍ

OBR. 1-17: RELACE DATABÁZE PER\_MINI PO ÚPRAVĚ TYPU SPOJENÍ





OBR. 1-18: VÝCHOZÍ STAV TABULKY CESTY

Cesty : Tabulka				
Rok	Cesta	Zahájení	Ukončení	Místo
2001	1	14.1.2001	16.1.2001	Kralupy nad Vltavou
2001	2	1.2.2001	6.2.2001	Poprad
2001	4	7.3.2001	10.3.2001	Brno
2001	6	23.3.2001	25.3.2001	České Budějovice
* Záznam: 4 z 4				

Po vyplnění obsahu a změně názvu položek přejdeme do návrhového zobrazení. Tabulku uložíme pod názvem *Cesty* a Access nenecháme doplnit primární klíč. Primární klíč doplníme sami v návrhovém zobrazení. Poprvé primární klíč neurčuje pouze jedna položka. *Cesty* totiž v položce *Cesta* číslujeme každý rok znovu. Jednotlivou větu v tabulce *Cesty* identifikuje jednoznačně dvojice polí *Rok* a *Cesta*. V návrhovém zobrazení tažením v prvním šedém nevyplněném sloupci (voliči řádků) označíme první dvě pole *Rok* a *Cesta* a klepneme do tlačítka **Primární klíč**. Symbol klíče se objevil ve voliči prvních dvou řádků.

V návrhu ponecháme datové typy dle návrhu Accessu (*Rok* a *Cesta* typu číselné, *Zahájení* a *Ukončení* typu datum a čas, *Místo* typu text). Klepnutím do ukončovacího tlačítka uzavřeme tabulku *Cesty*. Změny ve struktuře uložíme.

Nyní se pokusíme vytvořit relaci a referenční integritu mezi tabulkami *Personal* a *Cesty*. Řešíme dva nové problémy:

- Mezi tabulkami *Personal* a *Cesty* není vztah 1:N (k jedné větě z první tabulky existuje žádná, jedna či více vět z druhé tabulky), ani N:1 (žádná, jedna či více vět z první tabulky existuje k jedné větě z druhé tabulky).
- V tabulce *Cesty* jsou dvě klíčová pole.

Jeden zaměstnanec může vykonat více cest. Nemůžeme proto doplnit do tabulky *Personal* pole *Rok* a *Cesta*, jejichž obsah by nebyl jednoznačný. Není tedy možné definovat relaci N:1 mezi tabulkami *Personal* a *Cesty*.

Na jednu cestu mohlo jet i více zaměstnanců než jeden. Nemůžeme proto do tabulky *Cesty* doplnit pole *E\_mail*. Není tedy možné definovat relaci 1:N mezi tabulkami *Personal* a *Cesty*.

Poprvé se setkáváme s vazbou M:N (k jedné větě z první tabulky může být více vět z druhé tabulky a k jedné větě z druhé tabulky může být více vět z první tabulky). Ilustrujme si vazbu na konkrétním zjednodušeném obsahu tabulek zaměstnanců, kteří jeli na stejnou služební cestu.

Personal		Cesty		
<i>E_mail</i>	<i>Příjmení</i>	<i>Rok</i>	<i>Cesta</i>	<i>Místo</i>
Novak.Petr	Novák	2001	1	Kralupy nad Vltavou
Sladka.Marie	Sladká	2001	2	Poprad
		2001	6	České Budějovice

Zaměstnanci mohli jet na více služebních cest. Doplnění cesty (přesněji roku a cesty) do struktury tabulky *Personal* by proto vedlo k duplicitě vět v tabulce *Personal*. Museli bychom rozšířit primární klíč na položky *E\_mail*, *Rok* a *Cesta*. Příjmení by se opakovalo ve více větách a těžko by se aktualizovalo (Museli bychom příjmení, např. po provdání, opravovat ve více větách.):

Personal			
<i>E_mail</i>	<i>Příjmení</i>	<i>Rok</i>	<i>Cesta</i>
Novak.Petr	Novák	2001	1
Novak.Petr	Novák	2001	2
Novak.Petr	Novák	2001	6
Sladka.Marie	Sladká	2001	2

Na jednu cestu mohlo vyjet více zaměstnanců. Doplnění e-mailu zaměstnance do struktury tabulky *Cesty* by proto vedlo k duplicitě vět v tabulce *Cesty*. Museli bychom rozšířit primární klíč na položky *Rok*, *Cesta* a *E\_mail*. *Rok* a číslo cesty by se opakovaly ve více větách a těžko by se aktualizovaly. Bylo by nutné opravovat jej ve více větách.





Cesty

Rok	Cesta	Místo	E mail
2001	1	Kralupy nad Vltavou	Novak.Petr
2001	2	Poprad	Novak.Petr
2001	2	Poprad	Sladka.Marie
2001	6	České Budějovice	



Výjezdy

Normalizace

Vazbu typu M:N převedeme na dvě vazby typu 1:N vytvořením „můstkové“ tabulky *Výjezdy* dle obr. 1-19. Na primární klíč povýšíme všechna tři pole tabulky, abychom zabránili duplicitnímu zadání stejné služební cesty stejného zaměstnance.

OBR. 1-19: VÝJEZDY

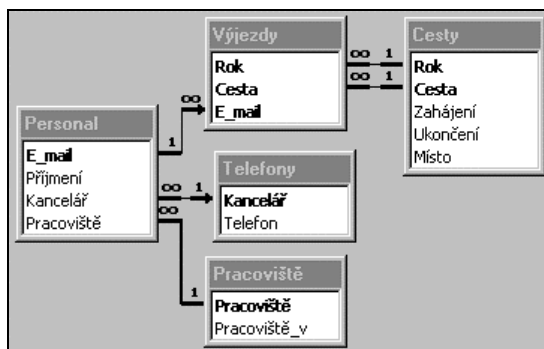
Rok	Cesta	E mail
2001	1	Novak.Petr
2001	2	Novak.Petr
2001	2	Sladka.Marie
2001	4	Pospisilova.Alena
2001	4	Svetly.Ondrej
2001	6	Novak.Petr

Uspořádání dat do více tabulek za účelem jedinečnosti polí se nazývá *normalizace*. Normalizace má svá pravidla:

- Každé pole v tabulce by mělo představovat jedinečný typ informace.
- Každá tabulka musí mít jednoznačný primární klíč, který je vytvořen z jednoho nebo více polí v této tabulce.
- Pro každou jedinečnou hodnotu primárního klíče se musí hodnoty datových polí týkat předmětu tabulky a musí tento předmět úplně popisovat.<sup>9</sup>

Po vytvoření „můstkové“ tabulky již můžeme doplnit definici relací v databázi dle obr. 1-20. V relaci mezi tabulkami *Personal* a *Výjezdy* upravíme typ spojení tak, aby se vypisovali všechny věty z tabulky *Personal*, protože ne každý zaměstnanec musí jet na služební cestu.

OBR. 1-20: RELACE V DATABÁZI PER\_MINI S „MŮSTKOVOU“ TABULKOU VÝJEZDY



Novým prvkem je vytvoření relace mezi tabulkami *Výjezdy* a *Cesty* s propojením dvou polí. Tažením myši přesuneme pole *Rok* z tabulky *Cesty* na pole *Rok* v tabulce *Výjezdy*. V dialogovém okně **Upravit relace** doplníme druhý řádek relace. Klepnutím do prvního sloupce druhého řádku se zobrazí vyklápěcí šipka, do níž klepneme. Z nabídky polí tabulky *Cesty* vybereme pole *Cesta*. Stejně pole vybereme do druhého sloupce z tabulky *Výjezdy*. Zaškrtneme pole *Zajistit referenční integritu* a *Aktualizace souvisejících polí v kaskádě*. Dále upravíme typ spojení.

OBR. 1-21: DVOJITÁ RELACE

Upravit relace

Tabulka či dotaz: Cesty Související tabulka či dotaz: Výjezdy

Rok Rok

Cesta Cesta

☒ Zajistit referenční integritu

☒ Aktualizace souvisejících polí v kaskádě

☐ Odstranění souvisejících polí v kaskádě

OK Storno Typ spojení Vytvořit novou

Pro úplnost ještě dodejme, že relaci můžeme také vytvořit poklepáním do šedivé plochy okna **Relace** a klepnutím do tlačítka **Vytvořit novou** v okně **Upravit relaci**. Stejně tlačítko můžeme využít i po poklepání na libovolnou relaci.

<sup>9</sup> Normalizaci provádí Access v *Průvodci analýzou tabulky* dostupném volbami v menu NÁSTROJE, ANALÝZA, TABULKA (viz kap. 10.1).





Abychom získali pro práci v dalších kapitolách větší množství dat bez jejich vkládání, převezmeme je ze sešitu Excelu. Dosavadní stav databáze ponecháme v souboru *Per\_mini.mdb* a vytvoříme novou databázi do souboru *Personal.mdb*. Každá tabulka je zastoupena v sešitu *Enc.xls* jedním listem s názvem shodným s názvem tabulky (*Personal*, *Telefony*, *Pracoviště*, *Cesty*, *Výjezdy*). Tabulky obsahují větší množství vět, tabulka *Personal* je obohacena o další datová pole.



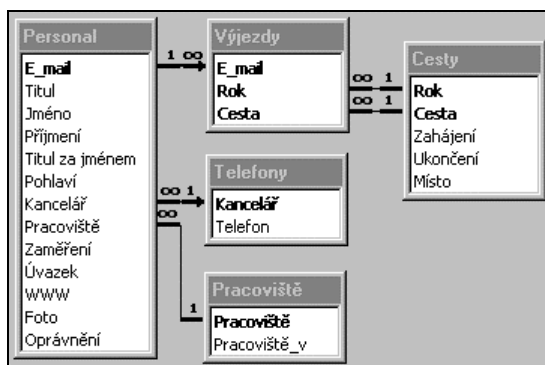
Personal.mdb

Novou databázi založíme příkazem z menu SOUBOR, NOVÝ, v kartě **Obecné** vybereme položku *Databáze*. Databázi vytvoříme ve vhodném adresáři a nazveme *Personal*. Postupně importujeme jednotlivé tabulky, pro každou tabulku aplikujeme následující postup:

- Z menu zadáme SOUBOR, NAČÍST EXTERNÍ DATA, IMPORT.
- V dialogovém okně **Import** v poli *Soubory typu* vybereme *Microsoft Excel* a najdeme sešit *Enc*. Klepneme do tlačítka **Import**.
- V prvním okně *Průvodce importem z tabulkové kalkulátoru* ponecháme vybráno pole *Zobrazit tabulky* a vpravo vybereme příslušný list.
- V druhém okně zaškrtneme pole *První řádek obsahuje hlavičky sloupců*.
- Ve třetím okně ponecháme zaškrtnuté pole *Do nové tabulky*.
- Ve čtvrtém okně ponecháme výběr všech polí pro import beze změny názvů polí.
- V pátém poli zvolíme *Vlastní primární klíč* v případě, že je tvořen jediným polem (tj. pro tabulky *Personal*, *Telefony* a *Pracoviště*), jinak zaškrtneme *Neurčovat primární klíč*.
- V posledním poli ponecháme navržený název importované tabulky.

V tabulkách, kde jsme neurčili při importu primární klíč, definujeme primární klíč v návrhovém zobrazení (v tabulce *Cesty* pole *Rok* a *Cesta*, v tabulce *Výjezdy* pole *E\_mail*, *Rok*, *Cesta*). V databázi nadefinujeme relace dle obr. 1-22.

OBR. 1-22: RELACE V DATABÁZI PERSONAL



(Ve schématu je prvotně v tabulce *Personal* zobrazeno pouze prvních pět polí. Tažením za spodní hranu zobrazíme všechna pole tabulky.)

## 1.4 Databáze Registr



Registr.mdb

Přípravu relací a referenční integrity procvičíme v databázi registru organizací, které obchodují s firmou Encián. Téměř všechny tabulky databáze (začínající písmeny *R\_*) jsou převzaty z Registru organizací. Údaje o podnicích jsou reálné, prodeje programů firmy Encián těmto podnikům jsou však smyšlené.

Založíme novou databázi *Registr*, do níž importujeme ze sešitu *Enc.xls* několik tabulek, zastoupených v sešitu samostatnými listy: *R\_Podniky*, *R\_Vzniky*, *R\_Vlastnictví*, *R\_SNA*, *R\_Právní\_formy*, *R\_Okresy*, *R\_Pracovníci*, *Typy*, *R\_OKEČ5*, *R\_Kraje*, *Burza*, *Burza\_převod*. Ve všech tabulkách je první pole primárním klíčem a je jediný.

Poznámky k tabulkám:

- *R\_Podniky*: základní tabulka, obsahující identifikační číslo organizace (IČO) a řadu údajů o podniku, z nichž většina je zaznamenána kódy. Kódy jsou slovně popsány ve vysvětlujících tabulkách *R\_Vzniky*, *R\_Vlastnictví*, *R\_Právní\_formy*, *R\_SNA*, *R\_Okresy*, *R\_Pracovníci*.



- *Typy*: tabulka vysvětlující kódy podniků využívané ve firmě Encián (jediné pole nepřebírané z Registru organizací).
- *R\_OKEČ5*: tabulka obsahuje vysvětlení pětímístných kódů odvětvové klasifikace ekonomických činností. Pro každou organizaci může být uvedeno až šest různých kódů OKEČ (*OKEČ5A*, *OKEČ5B*, *OKEČ5C*, *OKEČ5D*, *OKEČ5E*, *OKEČ5F*).

Za účelem vyjádření relací k tabulkám *R\_Kraje* a *Burza* připravíme v databázi dva dotazy, jejichž výsledné datové sady jsou využity v relacích.



R\_Kraje\_převod

V prvním dotazu (viz obr. 1-23) vybereme z vysvětlující tabulky *R\_Okresy* kód okresu a z něho zjistíme *Kraj*. Kódy okresů jsou tvořeny kódem kraje (1. a 2. znak) a upřesněním okresu (3. a 4. znak). Kód kraje tak získáme z kódu okresu jako levé dva znaky okresu.<sup>10</sup>

OBR. 1-23: DOTAZ R\_KRAJE\_PŘEVOD

Pole:	Okres	Kraj: Left([Okres];2)
Tabulka:	R_Okresy	
Řadit:		
Zobrazit:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kritéria:		
nebo:		



R\_OKEČ2\_převod

Ve druhém dotazu (viz obr. 1-24) vybereme z tabulky *R\_Podniky* pětímístný *OKEČ5A* a k němu vypočítáme dvoumístný *OKEČ2*. Dvoumístný *OKEČ2* počítáme pouze z první činnosti uváděné pro podnik. Pětímístný OKEČ je systematický: první dvě číslice vyjadřují odvětví, první tři číslice podrobnější členění, první čtyři číslice ještě podrobnější členění, všech pět číslic detailní členění. Dvoumístný *OKEČ2* potřebujeme pro napojení na převodní tabulku *Burza\_převod*, v níž k dvoumístným *OKEČ2* jsou uváděny kódy skupin burzy. Skupiny burzy obsahují většinou několik odvětví dle OKEČ.

OBR. 1-24: DOTAZ R\_OKEČ2\_PŘEVOD

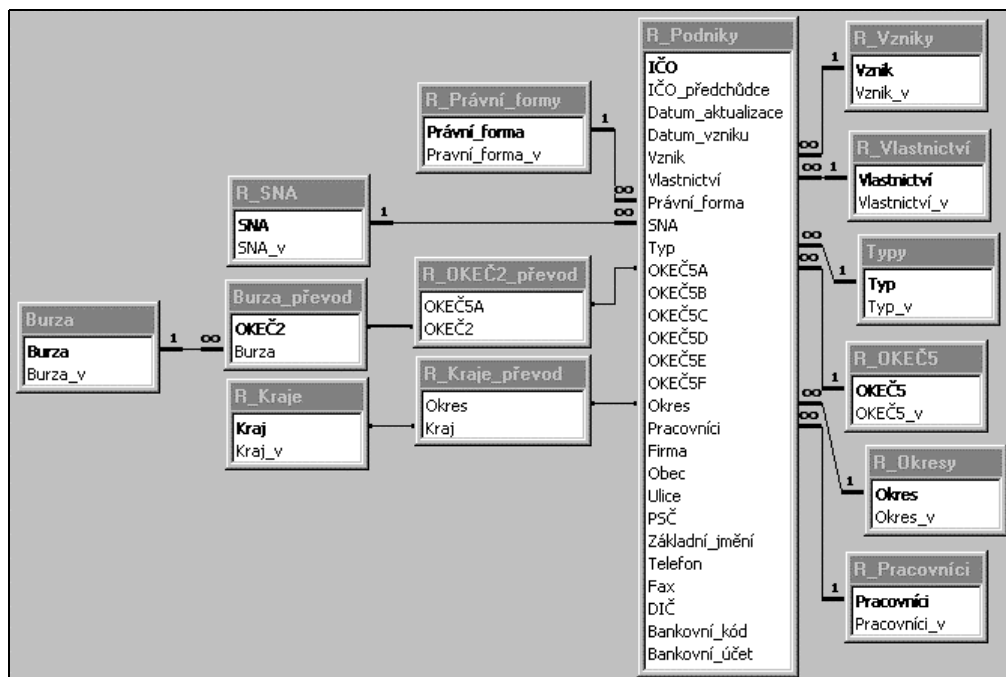
Pole:	OKEČ5A	OKEČ2: Left([OKEČ5A];2)
Tabulka:	R_Podniky	
Řadit:		
Zobrazit:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kritéria:		
nebo:		

V databázi *Registr* definujeme relace. V dialogovém okně **Zobrazit tabulku** (viz obr. 1-7) klepneme do záložky **Oboje** a vybereme všechny datové sady (tabulky i dotazy) klepnutím do prvního názvu a následným klepnutím se současným držením klávesy **Shift** do posledního názvu. Klepnutím do tlačítka **Přidat** přidáme všechny tabulky do schématu relací. Tabulky ve schématu uspořádáme dle obr. 1-25, podle nějž doplníme také relace a referenční integritu.

<sup>10</sup> Podrobnější výklad dotazů a funkce *Left* obsahuje kap. 3.1.



OBR. 1-25: RELACE V DATABÁZI REGISTR



Pro relace obsahující dotazy nelze definovat referenční integritu.

## 1.5 Databáze Enc\_maxi



Enc\_maxi.mdb

Vraťme se ještě k databázi *Enc\_mini* z kap. 1.1 a 1.2. Dosavadní stav databáze ponecháme v souboru *Enc\_mini.mdb* a vytvoříme novou databázi do souboru *Enc\_maxi.mdb*. Každá tabulka je zastoupena v sešitu *Enc.xls* jedním listem s názvem shodným s názvem tabulky (*Faktury*, *Licence*, *Programy*). Tabulky obsahují větší množství vět, tabulka *Faktury* je obohacena o další datová pole, tabulka *Licence* upřesňuje jednotlivé licence nakoupené na samostatné faktury.

Založíme novou databázi *Enc\_maxi*, do níž importujeme ze sešitu *Enc.xls* tři tabulky, zastoupené v sešitu samostatnými listy: *Faktury*, *Licence*, *Programy*. V tabulkách *Faktury* a *Programy* je první pole primárním klíčem a je jediný. V tabulce *Licence* jsou primárními klíči dvě první pole (*Program*, *Sériové\_číslo*). V databázi připravíme relace a referenční integritu dle obr. 1-26.

OBR. 1-26: VÝCHOZÍ RELACE V DATABÁZI ENC\_MAXI



V tabulce *Faktury* (102 vět) jsou prodeje sdruženy do faktur (tabulka *Prodeje* z databáze *Enc\_mini* obsahovala 165 vět). Na jednu fakturu mohlo být prodáno více programů, občas bylo prodáno i více licencí stejného programu na jedné fakturě. V rámci jednotlivých programů jsou licence jednoznačně identifikovány sériovým číslem. Celkem bylo prodáno 178 licencí. (Prodejů bylo pouze 165, protože existovaly prodeje více licencí stejného programu). V tabulce *Programy* je ke kódu programu doplněna *Cena* a vysvětlující text *Program\_v*.

Propojení  
tabulky

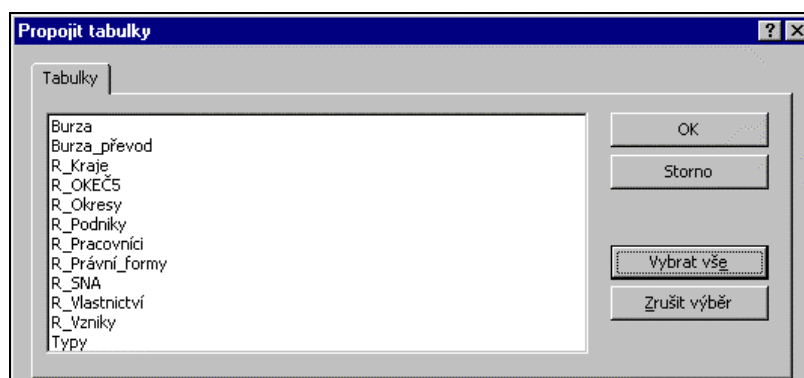
V tabulce *Faktury* je upřesněn:

- v poli *IČO* podnik (organizace), kterému jsme programy z faktury prodali,
  - v poli *E\_mail* zaměstnanec, který fakturu připravil a realizoval prodej.
- Díky těmto upřesněním můžeme propojit relacemi další tabulky o podnicích a zaměstnancích.

Access neumožňuje otevřít souběžně dvě databáze.<sup>11</sup> Můžeme však čerpat informace z tabulek jiné databáze tzv. *propojením*. Propojením tabulky zůstává tabulka v původní databázi. Můžeme upravovat pouze obsah tabulky, nemůžeme měnit strukturu tabulky.

Databázi *Enc\_maxi* propojíme s tabulkami databázi *Registr* a *Personal*. Z menu zadáme příkaz SOUBOR, NAČÍST EXTERNÍ DATA, PROPOJIT TABULKY. V dialogovém okně **Propojit** upřesníme adresář a název databáze, z níž chceme tabulku připojovat. Nejprve to bude databáze *Registr*. Po výběru databáze se zobrazí seznam tabulek v databázi (viz obr. 1-27). Klepnutím do tlačítka **Vybrat vše** vybereme všechny tabulky a klepneme do tlačítka **OK**.

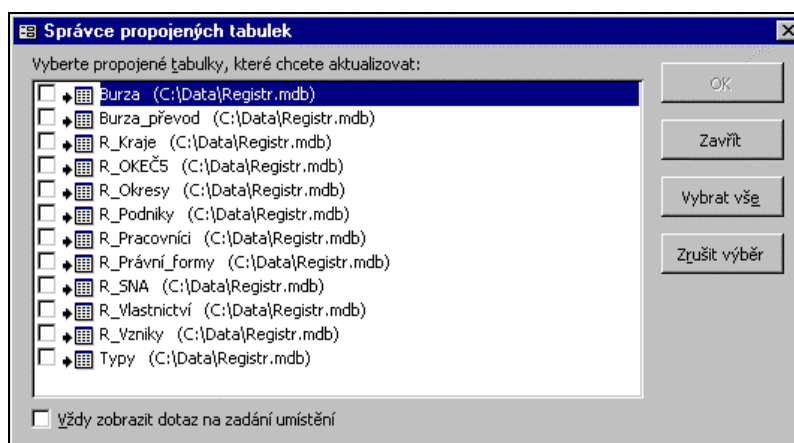
OBR. 1-27: DIALOGOVÉ OKNO PROPOJIT TABULKY



- Licence
- Programy
- R\_Kraje
- R\_OKEČ5

Tabulky se přidají do seznamu tabulek. Před symbolem tabulky je šipka, která signalizuje, že tabulka není součástí databáze, že je propojena. Propojenou tabulku můžeme použít i při konstrukci dotazů.

OBR. 1-28: DIALOGOVÉ OKNO SPRÁVCE PROPOJENÝCH TABULEK



Access si zapsal odkaz, odkud má propojenou tabulku čerpat. Pokud umístění změníme (např. databázi prohlédneme na jiném počítači v jiném adresáři), Access při práci s tabulkou (např. při pokusu o prohlížení) bude vypisovat chybu *Nelze nalézt soubor ... Registr.mdb*. Odkaz na připojenou tabulku lze v databázi změnit příkazem NÁSTROJE, SPRÁVA DATABÁZE, SPRÁVCE PROPOJENÝCH TABULEK.<sup>12</sup>

<sup>11</sup> Pokud potřebujeme souběžně pracovat se dvěma databázemi na úrovni návrhů objektů či jiných operací, spustíme ve Windows znovu Access a v něm otevřeme jinou databázi.

<sup>12</sup> Při instalaci Accessu musí být *Správce propojených tabulek* zahrnut do instalace.



V dialogovém okně **Správce propojených tabulek** (viz obr. 1-28) klepnutím do prázdného čtverečku na začátku řádku zaškrtneme tabulky, jejichž umístění se změnilo (popř. vybereme všechny klepnutím do tlačítka **Vybrat vše**). Po zaškrtnutí tabulek klepneme do již aktivního tlačítka **OK** a zadáme formou běžného dialogového okna otevírání souboru umístění souboru databáze. Pokud se pokoušíme opravit správná propojení, Access přímo vypíše hlášení *Všechny označené propojené tabulky byly úspěšně obnoveny*. Někdy potřebujeme změnit odkaz na jinou verzi databáze umístěnou v jiném adresáři. Potom zaškrtneme pole *Vždy zobrazit dotaz na zadání umístění* a Access nám dá možnost upřesnit umístění databáze i v případě, že databáze se stejným názvem v původním adresáři existuje.

V propojených tabulkách můžeme editovat data, která se mění v propojené databázi. Tabulku *R\_Podniky* doplníme do relací. Přetáhneme pole *IČO* z tabulky *R\_Podniky* na pole *IČO* v tabulce *Faktury*. Pole *Zajistit referenční integritu* není dostupné. Mezi dvěma databázemi nemůžeme udržovat referenční integritu.

Pokusme se ještě změnit strukturu tabulky *R\_Podniky*. V seznamu objektů tabulek vybereme tabulku *R\_Podniky* a klepneme na tlačítko **Návrh**. Access nás upozorní, že *Tabulka R\_Podniky je propojena. Některé vlastnosti nemohou být změněny*. Tlačítkem **Ne** ani neotevřeme návrhové zobrazení tabulky.

Propojení tabulky můžeme odstranit klepnutím do názvu tabulky, stisknutím klávesy **Delete** a potvrzením varujícího hlášení. Odstraní se propojení, nikoliv propojená tabulka v jiné databázi. Odstraníme propojení tabulky *R\_Podniky* včetně relace a propojení všech dalších tabulek databáze *Registr*.

#### Import tabulek

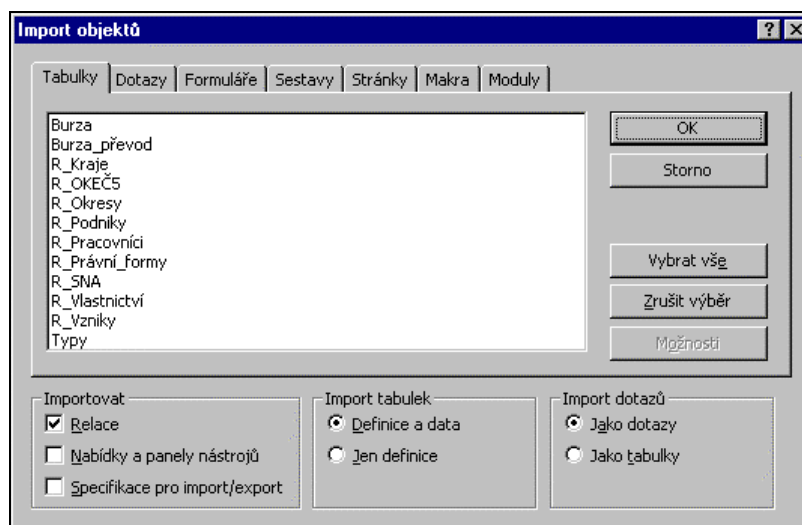
V našem případě chceme tabulky z databáze *Registr* umístit do databáze *Enc\_maxi*. Z menu zadáme příkaz **SOUBOR, NAČÍST EXTERNÍ DATA, IMPORT**. V dialogovém okně **Import** upřesníme adresář a název databáze *Registr*, z níž chceme tabulku připojovat. Po poklepání na název databáze se zobrazí seznam tabulek v databázi (viz obr. 1-29).

Klepnutím do tlačítka **Možnosti** můžeme rozšířit dialogové okno. (V obr. 1-29 je již stav po rozšíření.) Na rozdíl od propojování můžeme importovat i jiné druhy objektů (dotazy, formuláře a další). Je-li zaškrtnuto pole *Relace*, importujeme také relace mezi importovanými tabulkami z původní databáze. V případě importu tabulek můžeme importovat:

- strukturu tabulky i obsah tabulky (*Definice a data*) nebo
- jen strukturu tabulky (*Jen definice*).

V našem případě klepneme do tlačítka **Vybrat vše**. Klepneme na záložku **Dotazy**, opět klepneme do tlačítka **Vybrat vše** a potom do tlačítka **OK**.<sup>13</sup>

OBR. 1-29: DIALOGOVÉ OKNO IMPORT OBJEKTŮ PO KLEPNUTÍ DO TLAČÍTKA MOŽNOSTI

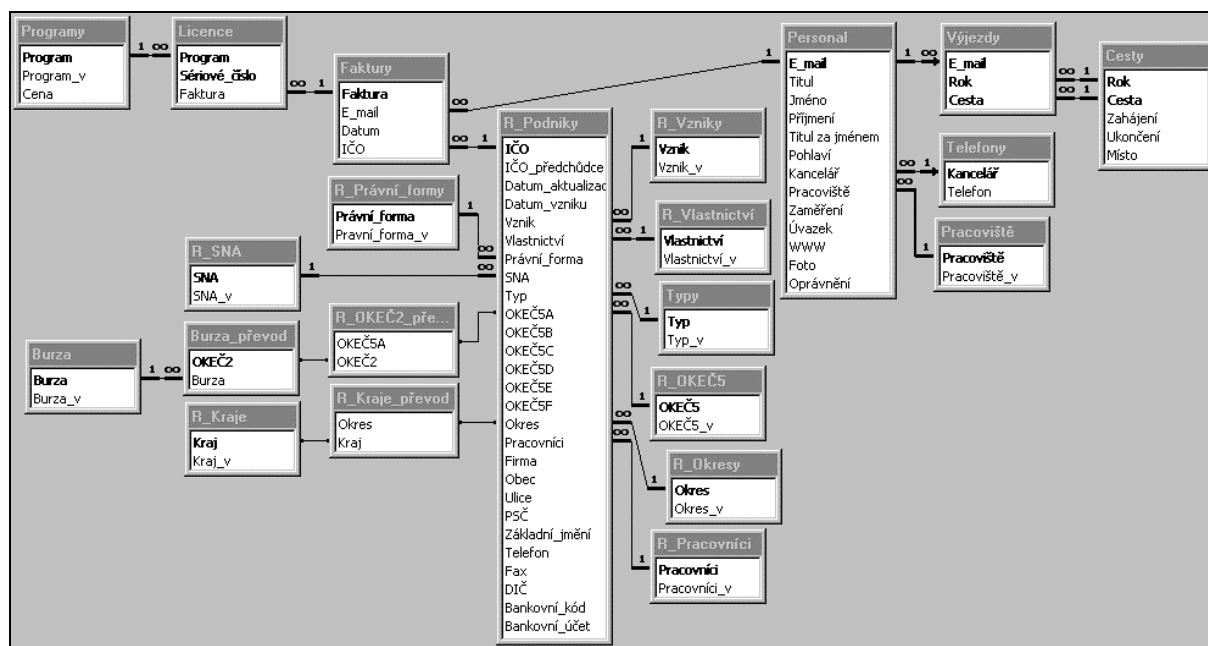


<sup>13</sup> Další informace o importu objektů z jiné databáze Accessu jsou uvedeny v kap. 9.5.



Obdobně importujeme všech pět tabulek včetně relací z databáze *Personal*. Podívejme se, jak se převzaly z původních databází relace a pravidla referenční integrity. Zobrazíme relace. Klepnutím do tlačítka **Zobrazit všechny relace** zobrazíme v schématu neuvedené relace včetně seznamu polí tabulek, k nimž se relace vztahují. Doplníme relaci mezi tabulkami *Faktury* – *R\_Podniky* (prostřednictvím pole *IČO*) a *Faktury* – *Personal* (prostřednictvím pole *E\_mail*) včetně referenční integrity a aktualizace souvisejících polí v kaskádě. Tabulku *R\_Kraje*, dotazy a relace k výsledkům dotazů musíme do schématu znovu přidat. Vhodně uspořádáme objekty v seznamu relací (viz obr. 1-30).

OBR. 1-30: RELACE V DATABÁZI ENC\_MAXI



## Shrnutí

1. Data jsou v databázích Accessu uložena ve formě *tabulek*.
2. Tabulky lze do databáze Accessu *načíst*, např. *ze sešitu Excelu*. Rozsahy dílčích tabulek v sešitu je vhodné nazvat.
3. Kromě tabulek obsahuje databáze Accessu další *objekty*: dotazy, formuláře, sestavy, stránky, makra, moduly.
4. *Seznamy objektů* lze zobrazovat ve formě velkých ikon, malých ikon, seznamu či detailů.
5. *Primární klíč* je pole (či více polí), jehož hodnota jednoznačně identifikuje větu tabulky.
6. Obsah tabulky lze prohlížet ve formě *datových listů*, v nichž je možné data upravovat, přidávat a odstraňovat věty. Strukturu tabulky lze měnit v *návrhovém zobrazení*.
7. Strukturu nové tabulky lze vytvořit i v zobrazení datového listu.
8. Mezi tabulkami je možné definovat relace. *Relace* jsou východiskem pro přípravu dotazů.
9. Relace je vhodné připravovat v *schématu relací*, z něhož můžeme přecházet i do návrhu struktury tabulek a do datového listu.
10. V rámci definice relace je možné stanovit také referenční integritu. *Referenční integrita* je systém pravidel, která se využívají k zajištění platnosti relací mezi záznamy provázaných tabulek.
11. Fungování referenční integrity lze modifikovat vlastnostmi *Aktualizace souvisejících polí v kaskádě* a *Odstranění souvisejících polí v kaskádě*.
12. V případě, že v relaci bývá často nevyplněno propojující pole, bývá vhodné změnit *typ spojení*.
13. Vazby typu M:N převádíme *normalizací* na dvě vazby typu 1:N.
14. Tabulky můžeme do databáze *propojovat* či *importovat* z jiné databáze.