**„EU peníze školám“**

**Projekt DIGIT – digitalizace výuky na ISŠTE Sokolov**

**reg. č. CZ.1.07/1.5.00/34.0496**

|  |  |
| --- | --- |
| **III/2 Inovace a zkvalitnění výuky prostřednictvím ICT** | **VY\_32\_INOVACE\_1\_6\_04** |
| Název vzdělávacího materiálu | Základy SQL – vazby mezi tabulkami |
| Jméno autora | Ing. Bohuslava ČEŽÍKOVÁ |
| Tématická oblast | Databázové systémy |
| Vzdělávací obor | 68-46-M/01 Veřejnosprávní činnost |
| Předmět | Informační a komunikační technologie |
| Ročník | 2., 3. |
| Rozvíjené klíčové kompetence | Kompetence k učení   * posoudí vlastní pokrok a určí překážky či problémy bránící učení; * kriticky zhodnotí výsledky;   Kompetence k řešení problémů   * osvědčené postupy aplikuje při řešení obdobných problémových situací; * sleduje vlastní pokrok při zdolávání problémů. |
| Průřezové téma | Informační a komunikační technologie |
| Časový harmonogram | 1 vyučovací hodina |
| Použitá literatura a zdroje | Databáze, modely dat, relační algebra, SŘBD, SQL, normální formy [online]. [cit. 2013-02-12]. Dostupné z: http://is.muni.cz/th/143327/fi\_b/bc\_prace\_vitxx.txt  Entity-relationship model. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-02-12]. Dostupné z: http://cs.wikipedia.org/wiki/Entity-relationship\_model  SKŘIVAN, Jaromír. SQL - tvorba tabulek. In: [online]. [cit. 2013-02-12]. Dostupné z: http://interval.cz/clanky/sql-tvorba-tabulek/ |
| Pomůcky a prostředky | pc, internet, sw |
| Anotace | Databázové systémy, základy jazyka SQL – vazby mezi tabulkami. |
| Způsob využití výukového materiálu ve výuce | Výklad, pracovní list |
| Datum (období) vytvoření vzdělávacího materiálu | 2/2013 |

*Tento výukový materiál je plně v souladu s Autorským zákonem (jsou zde dodržována všechna autorská práva). Pokud není uvedeno jinak, autorem textů   
a obrázků je Ing. Bohuslava Čežíková.*

**Vazby mezi tabulkami**

V praxi těžko budeme mít systém, jehož databáze by obsahovala jen jednoduchou tabulku. Naše firma bude mít různá oddělení, do kterých budou zaměstnanci rozděleni, budou plnit různé funkce, od kterých se budou odvíjet jejich platy, apod. Všechny tyto údaje nemůžeme vměstnat do jediné tabulky. Pro evidenci všech těchto údajů budeme potřebovat víc tabulek. Navrhněme si tedy následující tabulky. Tabulka ZAMESTNANEC bude obsahovat sloupce CISLO, JMENO, PRIJMENI, ROD\_CISLO, CIS\_ODD a FUNKCE. Druhá tabulka ODDELENI bude obsahovat CISLO a NAZEV. Třetí tabulka se bude jmenovat PLATY, která bude mít sloupce CISLO, FUNKCE a PLAT. Pro lepší pochopení následujícího výkladu si představme, že v tabulkách máme následující údaje:

ZAMESTNANEC:

CISLO JMENO PRIJMENI ROD\_CISLO CIS\_ODD FUNKCE

1 Jan Novák 751015/2352 10 127

2 Petr Nový 780401/4421 15 121

3 Jan Nováček 650906/1566 10 156

4 Jiří Novotný 40205/3566 20 127

ODDELENI:

CISLO NAZEV

10 studovna

15 centrála

20 počítačový sál

PLATY:

CISLO FUNKCE PLAT

121 vedoucí 31500

127 technik 27000

156 správce 21500

Nyní se podrobněji podíváme, jaké informace nám tyto tři tabulky poskytují. V první tabulce ZAMESTNANEC, na řádku s číslem zaměstnance 1, vidíme, že jeho funkce ve firmě má číslo 127. Z tabulky PLATY vyčteme, že funkce s číslem 127 je technik. Dále v prvním řádku tabulky ZAMĚSTNANEC vidíme, že Jan Novák pracuje na oddělení číslo 10. Z tabulky ODDĚLENÍ zjistíme, že oddělení číslo 10 je studovna.

**Reprezentace vazeb 1:N (1:1)**

Všechny tři tabulky spolu nějakým způsobem souvisí. Všechny tři nám dohromady uchovávají všechny informace, které potřebujeme. Primárními klíči v každé tabulce našeho příkladu je vždy sloupce CISLO. Hodnoty v tomto sloupci jsou v rámci každé tabulky jedinečné. Mezi těmito třemi tabulkami existují celkem dvě vazby. První vazbu můžeme najít v první tabulce ve sloupci CIS\_ODD, jehož hodnoty se odkazují do tabulky ODDELENI. Druhá vazba je dána v první tabulce sloupcem FUNKCE, jehož hodnoty se odkazují do tabulky PLATY. Pokud tedy chceme vytvořit správně všechny tři tabulky, musíme SQL serveru při jejich zakládání o těchto vazbách říct.

Sloupec CIS\_ODD v tabulce ZAMESTNANEC se nazývá *cizí klíč*. Cizí klíč nám (spolu s primárním klíčem) zprostředkovává vazbu mezi tabulkami. Jeho hodnota pak odpovídá v jiné tabulce (která je odkazovaná) hodnotě primárního klíče. Druhým cizím klíčem v tabulce ZAMESTNANEC je sloupec FUNKCE. Ten odkazuje na hodnoty sloupce CISLO v tabulce PLATY.

Obě vazby jsou *typu 1:N*. To znamená, že jeden zaměstnanec pracuje právě na jednom oddělení, ale na jednom oddělení pracuje více zaměstnanců (odtud tedy poměr *oddělení:zaměstnanec = 1:N*). Dále jeden zaměstnanec vykonává jednu funkci (ale jedna funkce může být vykonávána více zaměstnanci, např. můžeme mít ve firmě 10 programátorů).

Jak tedy pomocí příkazů jazyka SQL vytvořit tyto tři tabulky se správnou „provázaností“? Cizí klíč je speciální případ integritního omezení, neboť hodnotu položky samotné nám vůbec neomezuje, pouze popisuje vazbu mezi tabulkami.

Syntaxe zápisu cizího klíče je následující:

FOREIGN KEY (jména sloupců)

REFERENCES jméno\_odkazované\_tabulky (její primární klíče)

Na chvíli se zastavme nad postupem, kterou tabulku vytvořit jako první, kterou jako druhou a kterou jako poslední. Kdybychom se nejprve pokusili vytvořit tabulku ZAMESTNANEC, server by nám nahlásil chybu (není to úplně pravda, některé databázové systémy by nám umožnily vytvořit tabulku ZAMESTNANEC jako první). Odkazovali bychom totiž na tabulku, která ještě v systému neexistuje. Takže správný postup spočívá nejprve ve vytvoření tabulek, na které se budeme odkazovat a nakonec vytvoříme tabulky, které se odkazují na jiné. Je nutné, aby zúčastněné sloupce měly stejný datový typ. Následuje zápis v SQL pro vytvoření uvedených tří tabulek (včetně správného pořadí vytvoření):

CREATE TABLE oddeleni

(cislo INTEGER,

nazev VARCHAR(20),

PRIMARY KEY (cislo))

CREATE TABLE platy

(cislo INTEGER,

funkce VARCHAR(10),

plat FLOAT(10) DEFAULT 8000,

PRIMARY KEY (cislo))

CREATE TABLE zamestnanec

(cislo INTEGER,

jmeno VARCHAR(10),

prijmeni VARCHAR(20),

rodne\_cislo VARCHAR(11) NOT NULL,

cis\_odd INTEGER,

funkce INTEGER,

PRIMARY KEY (cislo),

FOREIGN KEY (cis\_odd) REFERENCES oddeleni (cislo),

FOREIGN KEY (funkce) REFERENCES platy (cislo))

Vazbu 1:1, i když je speciálním případem vazby 1:N, nemusíme reprezentovat pomocí cizích klíčů. Vezměme si např. člověka a jeho rodné číslo. Zde evidentně nastává vazba 1:1, neboť každý člověk má právě jedno rodné číslo a jedno rodné číslo náleží právě jednomu člověku. Tuto vlastnost jsme schovali rovnou do tabulky, tj. rodné číslo je přímo jedním z atributů zaměstnance a není třeba pro rodné číslo vytvářet samostatnou tabulku, která by měla sloupce CISLO\_ZAM a ROD\_CIS. Pokud byste takovou tabulku vytvořili, chyba by to nebyla, ale je to zbytečné.

**Vazba M:N**

Vazba M:N se realizuje opět pomocí cizích klíčů. Je tu ale jedno rozšíření oproti předchozímu případu. Vezměme si situaci, že každý zaměstnanec může vykonávat obecně více funkcí (a jedna funkce může být vykonávána více zaměstnanci). Máme tedy tabulku ZAMESTNANEC se sloupci JMENO, PRIJMENI a ROD\_CIS. Druhou tabulku nazvěme FUNKCE, která bude mít sloupce CISLO a NAZEV. K realizaci této vazby nám ovšem nebudou tyto dvě tabulky stačit. Potřebujeme ještě jednu, nazvěme si ji VYKON\_FUNKCE. Ta bude obsahovat dva atributy – prvním bude zaměstnanec a druhým bude funkce, kterou vykonává. Do takové tabulky jsme schopni zanést informace, že jeden zaměstnanec vykonává více funkcí (bude v tabulce VYKON\_FUNKCE uveden na více řádcích) a zároveň budeme schopni vyčíst, jací lidé vykonávají konkrétní funkci. Oba atributy v této pomocné tabulce budou reprezentovány pomocí čísla zaměstnance a čísla funkce. Mějme tyto tabulky:

ZAMESTNANEC:

CISLO JMENO PRIJMENI ROD\_CISLO

1 Jan Novák 751015/2352

2 Petr Nový 780401/4421

3 Jan Nováček 650906/1566

4 Jiří Novotný 740205/3566

VYKON\_FUNKCE:

CISLO\_ZAM CISLO\_FUN

1 127

2 121

3 156

4 127

2 127

FUNKCE:

CISLO NAZEV

121 vedoucí

127 technik

156 správce

Podívejme se na obsah těchto tabulek, jaké informace nám sdělují. V tabulce VYKON\_FUNKCE na druhém řádku vyčteme, že zaměstnanec číslo 2 (Petr Nový) vykonává funkci číslo 121 (vedoucí) a na posledním řádku vyčteme dále, že tentýž zaměstnanec vykonává také funkci číslo 127 (technik). Konkrétně zaměstnanec Petr Nový vykonává v naší firmě vedoucího a je zároveň technikem. Když se na tabulku podíváme z druhého pohledu, zjistíme, že funkci číslo 127 (technik) vykonávají celkem 3 lidé, a to zaměstnanec číslo 1, 2 a 4 (jejich jména bychom vyčetli z tabulky ZAMESTNANEC).

Sloupec CISLO\_ZAM, v tabulce VYKON\_FUNKCE, je cizím klíčem a odkazuje do tabulky ZAMESTNANEC, sloupec CISLO\_FUN je cizím klíčem odkazujícím do tabulky FUNKCE. Jak je to s primárním klíčem v tabulce VYKON\_FUNKCE? Jedinou možnost, kterou máme, je za primární klíč zvolit oba sloupce, tj. dvojici (CISLO\_ZAM, CISLO\_FUN). Samozřejmě bychom tabulku VYKON\_FUNKCE mohli rozšířit o sloupce např. DATUM\_OD, kde by na každém řádku pro zaměstnance a funkci byla informace, kdy začal zaměstnanec příslušnou funkci vykonávat. SQL příkazy pro vytvoření těchto tří tabulek jsou tyto:

CREATE TABLE zamestnanec

(cislo INTEGER,

jmeno VARCHAR(10),

prijmeni VARCHAR(20),

rodne\_cislo VARCHAR(11) NOT NULL,

PRIMARY KEY (cislo))

CREATE TABLE funkce

(cislo INTEGER,

nazev VARCHAR(10),

PRIMARY KEY (cislo))

CREATE TABLE vykon\_funkce

(cislo\_zam INTEGER,

cislo\_fun INTEGER,

PRIMARY KEY (cislo\_zam, cislo\_fun),

FOREIGN KEY (cislo\_zam) REFERENCES zamestnanec (cislo),

FOREIGN KEY (cislo\_fun) REFERENCES funkce (cislo))

Sloupce CISLO\_ZAM a CISLO\_FUN tvoří ve dvojici primární klíč a zároveň jsou cizími klíči. Chyba to samozřejmě není, obecně v SQL neexistuje žádný důvod, proč by sloupec, který tvoří primární klíč, nemohl být zároveň cizím klíčem.

Poznámka: *Při psaní SQL příkazů většinou nezáleží na velikosti písmen.*

**Pracovní list**

**Určete typ vazeb**

|  |  |
| --- | --- |
| **Situace** | **Vazba**  **(M:N, 1:N, 1:1)** |
| Student – předmět |  |
| Manžel – manželka (ČR) |  |
| Autor – kniha |  |
| Vstupenka (konkrétní akce) – návštěvník |  |
| Faktura – odběratel |  |
| Stavba – stavební firma |  |
| Člověk – rodné číslo |  |