**„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\_09** |
| Název vzdělávacího materiálu | Základy SQL – vkládání dat do tabulky |
| 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.txtEntity-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\_modelSKŘIVAN, 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.*

**Vkládání dat do tabulky**

Máme-li kompletně vytvořenou strukturu databázových tabulek, můžeme začít s jejich naplněním daty. Pro vkládání jednoho řádku dat do tabulky se používá příkaz INSERT, jehož syntaxe je následující:

INSERT INTO jméno\_tabulky [(jména sloupců)] VALUES (seznam hodnot)

Za klíčovými slovy INSERT INTO se udává jméno tabulky, do které chceme data vkládat (je-li tabulka v jiném databázovém schématu, použijeme tvar jméno\_schématu.jméno\_tabulky). Seznam sloupců uvedený v hranatých závorkách je nepovinný, použijeme jej jenom ve speciálním případě. Za klíčovým slovem je uveden seznam hodnot, které vkládáme. Pořadí hodnot musí přesně odpovídat pořadí sloupců v tabulce. Pokud toto pořadí hodnot nechceme dodržet, pak musíme uvést část jména\_sloupců za klíčovým slovem INTO, ve které budeme vlastně specifikovat, v jakém pořadí vkládáme jednotlivé hodnoty. Pořadí sloupců tedy musí odpovídat pořadí hodnot.

**Vkládání dat do tabulky**

Vytvoříme tabulku ZAMESTNANEC. Ta bude obsahovat jednoduché sloupce a nebude mít žádný cizí klíč. Tabulku vytvoříme následujícím příkazem:

CREATE TABLE zamestnanec

 (cislo INTEGER PRIMARY KEY,

 jmeno VARCHAR(10),

 prijmeni VARCHAR(20) NOT NULL,

 pocet\_hodin INTEGER,

 vzdelani VARCHAR(12) DEFAULT 'neuvedeno',

 ridicsky\_p CHAR(1) DEFAULT 'N')

Pro úspěšné vložení řádku do tabulky musíme dodržet několik zásad, které se týkají uvedených integritních omezeních. V prvé řadě musíme zajistit, aby vkládaná hodnota sloupce CISLO, která je primárním klíčem tabulky, byla jedinečná. Dále musí být vždy zadána hodnota PRIJMENI. A samozřejmě musí platit, že typ zadávaných hodnot bude totožný (nebo i kompatibilní, např. celé číslo lze uložit do položky reálného čísla, ale ne naopak) s typem sloupců, do kterých hodnoty vkládáme. Následuje výčet příkazů INSERT, které se provedou úspěšně. Předpokládejme, že v tabulce ZAMESTNANEC nemáme ještě žádná data.

INSERT INTO zamestnanec

 VALUES (1, 'Jan', 'Novák', 60, 'SŠ', 'A')

INSERT INTO zamestnanec (cislo, jmeno, prijmeni, pocet\_hodin)

 VALUES (2, 'Petr', 'Nový', 50)

INSERT INTO zamestnanec (cislo, prijmeni, jmeno, ridicsky\_p)

 VALUES (7, 'Novacek', 'Jan', 'A')

INSERT INTO zamestnanec (cislo, prijmeni)

 VALUES (4, 'Novotný')

První příkaz pro vkládání dat do tabulky je jeho uvedení v základním tvaru, pořadí hodnot odpovídá přesně pořadí sloupců, jak jsme je vytvořili v tabulce.

V druhém příkazu vkládáme další řádek do tabulky, vložíme hodnoty položek CISLO, JMENO, PRIJMENI a POCET\_HODIN, do položky VZDELANI se uloží implicitní hodnota ‚neuvedeno‘, a do položky RIDICSKY\_P se vloží předdefinovaná hodnota ‚N‘.

Třetí INSERT zanechá položku POCET\_HODIN jako nedefinovanou, ostatní budou mít hodnotu zadanou, nebo defaultní.

Poslední příkaz nám vkládá hodnoty do sloupců, které z definice tabulky nesmí zůstat prázdné, ostatní položky buď budou mít defaultní hodnotu, nebo nedefinovanou. Většina databázových systémů neumožňuje nezadat všechny sloupce, jak je tomu u druhého až čtvrtého příkazu. Tuto vlastnost podporuje např. systém Oracle. Tabulka po provedení těchto čtyř příkazů bude mít následující obsah:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CISLO | JMENO | PPRJMENI | POCET\_HODIN | VZDELANI | RIDICSKY\_P |
| 1 | Jan | Novák | 60 | SŠ | A |
| 2 | Petr | Nový | 50 | neuvedeno | N |
| 7 | Jan | Nováček |  | neuvedeno | A |
| 4 |  | Novotný |  | neuvedeno | N  |

**Pracovní list**

Zapište příkaz SQL, který do tabulky ZAMESTNANEC zapíše 4 řádky (data):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **cislo** | **jmeno** | **prijmeni** | **rodne\_cislo** | **vzdelani** | **ridicske\_opravneni** |
| 1 | Jan | Novák | 123456/0123 | VŠ | NE |
| 2 | Petr | Nový | 234567/1234 | SŠ | B |
| 3 | Jiří | Nováček | 345678/2345 | SŠ | NE |
| 4 | Jana | Nádvorníková | 456789/3456 | VŠ | B |

Poznámka: strukturu tabulky ZAMESTNANEC známe z minulých hodin:

`cislo` int AUTO\_INCREMENT,

 `jmeno` varchar(20),

 `prijmeni` varchar(20),

 `rodne\_cislo` varchar(11),

 `vzdelani` varchar(10),

 `ridicske\_opravneni` varchar(3),

 PRIMARY KEY (`cislo`)