Kapitola 3

DML - Insert, Delete, Update

3.1 Zadanie cvičenia - Insert

Podľa úloh vložte nasledujúce dáta do tabuliek `os_udaje, student, zap_predmety`.
Hodnoty, ktoré nie sú zadané nastavte na NULL, príp. na hodnotu, aby boli zachované integritné obmedzenia.

rod_cislo	meno	priezvisko	os_cislo	rocnik	st_skupina	zameranie	má zapísané predmety v šk.roku
830722/6247	Karol	Novy	123	1	5ZI012	Inf.	BI11(2008), BI02(2008), BE01(2003)
860114/2462	Karol	Lempassky	90	2	5ZSA21	IS-AI	II08(2006), II07(2007)
831228/8896	Marek	Hyk	23	3	5ZP031	PI-bc.	BH01(2009), BF08(2009)
845902/0012	Petra	Dratova	8	3	5ZI032	Inf.	BI11(2006), BH18(2009), BF11(2009)
870412/7259	Filip	Petrovic	234	1	5ZSD13	IS-SPR	II07(2007)
820101/7725	Karol	Granak	13	2	5ZN021	IM	IE01(2007)

Poznámka:

1. Skratka št. odboru, zamerania - podľa nej si zistite číslo odboru a zamerania:
   - Inf – Informatika , bez zamerania - bakalárske štúdium
   - PI-bc. - Počítačové inžinierstvo, bez zamerania - bakalárske štúdium
   - IS-AI - Informačné systémy , Aplikovaná informatika - inžinierske štúdium
   - IS-SPR - Informačné systémy , Systémy pre podporu rozhodovania - inžinierske štúdium
   - IM - Informačný manažment, bez zamerania - inžinierske štúdium
2. Učiteľa a počet kreditov si zistite z tabuľky `predmet_bod`.

!

3.1.1 Úlohy insert

1. údaje z prvých dvoch riadkov vložte pomocou príkazov SQL – **Insert into ... values(...)**
2. údaje z ďalších riadkov vložte nasledovne:
   (a) údaje sú už uložené v dvoch tabuľkách – `osoba` (osobné a študentské údaje) a `skusky`(zapísané

predmety) v schéme KVET3.

(b) vložte údaje do tabuľky `os_udaje` z tabuľky `osoba`

```
insert into os_udaje(rod_cislo, meno, priezvisko)
  select rod_cislo, meno, priezvisko from kvet3.osoba;
```

(c) vložte údaje do tabuľky `študent` z tabuľky `osoba`

```
insert into student
        (rod_cislo, os_cislo, rocnik, st_skupina,
         st_odbor, st_zameranie)
  select rod_cislo, os_cislo, rocnik, st_skupina, st_odbor, st_zameranie
    from kvet3.osoba;
```

(d) vložte údaje do tabuľky `zap_predmety` z tabuľky `skusky`

```
insert into zap_predmety
         (os_cislo, cis_predm, skrok, prednasajuci, ects)
  select os_cislo, cis_predm, skrok, garant, ects
    from kvet3.skusky sk join  predmet_bod  using (cis_predm,skrok);
```

3.2 Zadanie cvičenia - Update

1. Zmeňte priezvisko u študenta `Novy` na `Stary`.
2. Zmeňte meno študenta, ktorý má osobné číslo `8` na Karolína.
3. Zmeňte len všetkým prvákom číslo predmetu z `BI11` na `BI01`.
4. Zmeňte všetkým študentom, ktorí nemajú nastavený žiadny stav, stav na `S`.
5. Zvýšte študentom, ktorí majú stav **S** a nie sú v poslednom ročníku, ročník o jedna (upravte i študijnú skupinu), pričom použijete len jeden príkaz.

Štruktúra študijnej skupiny:

fakulta	pracovisko	odbor	zameranie	rocnik	kruzok	
5	P	I	0	2	7	- Prievidza, Inf., 2.ročník, krúžok 7
5	Z	S	A	1	3	- Žilina, IS-AI, 1.ročník, krúžok 3

!

- výber z podreťazca -

  `substr(zcoho, odkadial, kolko) -substr( '5ZSA13', 4,1) = 'A'`

- spojenie reťazcov -

  `'Ahoj '||'Karol'= 'Ahoj Karol'`

- bakalári majú st_odbor z intervalu $< 100, 199 >$ a trojročné štúdium
- inžinieri majú st_odbor z intervalu $< 200, 299 >$ a dvojročné štúdium

3.3 Zadanie cvičenia - Delete

1. Zmažte študentovi s osobným číslom `123` predmet `BE01`
2. Zmažte všetkým študentom študijnej skupiny `5ZI022` predmet `BI01`
3. Zmažte všetky dáta vo všetkých tabuľkách (`os_udaje, student, zap_predmety`), ktoré sa týkajú študentov zapísaných (`dat_zapisu`) v roku `1999`.

Použitie funkcie pre prácu s dátumom:

```
select dat_zapisu,
       to_char(dat_zapisu,'DD') as den,
       to_char(dat_zapisu,'MM') as mesiac,
       to_char(dat_zapisu,'YYYY') as rok
from student;
```

Kapitola 4

Pomôcky

4.1 Insert

```
INSERT INTO tabulka[(zoznam_stlpcov)]
{
  VALUES (zoznam_hodnot)
  |
  SELECT-prikaz
}
```

- Príkaz Insert má dve formy:
  - VALUES - vloženie LEN JEDNÉHO riadku pomocou priameho zadania nových hodnôt.
  - SELECT - vloženie viacerých riadkov predpripravených dát z pomocných tabuliek.
- Je potrebné zadať unikátne hodnoty stĺpcov primárneho kľúča.
- Je potrebné zadať hodnoty všetkých NOT NULL stĺpcov.
- Aby bolo možné znovu pospájať súvisiace dáta z rôznych tabuliek, je potrebné zadať hodnotu cudzieho kľúča.

4.1.1 INSERT - VALUES

```
SQL> desc os_udaje
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ROD_CISLO                                 NOT NULL CHAR(11)
 MENO                                      NOT NULL VARCHAR2(15)
 PRIEZVISKO                                NOT NULL VARCHAR2(15)
 ULICA                                              VARCHAR2(20)
 PSC                                                CHAR(5)
 OBEC                                               VARCHAR2(50)
```

- **Počet a poradie hodnôt je rovnaké ako je vytvorená tabuľka. Neznáme hodnoty nastavte na NULL, ak sa nejedná o NOT NULL stĺpec:**

```
INSERT into os_udaje
VALUES ('810502/5511','Peter', 'Maly',  null, null, null );
```

- **Hodnoty sú v poradí v akom sú vymenované stĺpce:**

```
INSERT INTO os_udaje(meno, priezvisko, rod_cislo)
VALUES ('Peter', 'Maly', '810502/5511');
```

4.1.2 INSERT - SELECT

- **Vloženie hodnôt z pomocnej tabuľky:**

```
INSERT INTO os_udaje( rod_cislo, priezvisko, meno )
SELECT rc, priezvisko, meno
  FROM prijimacky
  WHERE prijaty = 'A';
```

- **Kombinácia hodnôt z tabuľky a konštánt.**

```
INSERT INTO student( os_cislo, rod_cislo, st_odbor, st_zameranie, rocnik, stav, dat_zapisu )
SELECT os_cislo, rod_cislo, st_odbor, st_zameranie,  1, 'S', sysdate
  FROM prijimacky
  WHERE prijaty = 'A';
```

Príklad 4.1: Vložte prvákom (Inf.) všetky povinné predmety prvého ročníka roku 2009

1. Budeme potrebovať získať všetky hodnoty pre NOT NULL stĺpce.

```
SQL> desc zap_predmety
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OS_CISLO                                  NOT NULL NUMBER(38)
 CIS_PREDM                                 NOT NULL CHAR(4)
 SKROK                                     NOT NULL NUMBER(4)
 PREDNASAJUCI                              NOT NULL CHAR(5)
 ECTS                                      NOT NULL NUMBER(3)
 ZAPOCET                                            DATE
 VYSLEDOK                                           CHAR(1)
 DATUM_SK                                           DATE
```

2. Získanie množiny študentov prvého ročníka Informatiky.

```
select st.os_cislo
  from student st
  where st.rocnik = 1
    and st.st_odbor = 100
    and st.st_zameranie = 0;
```

3. Získanie množiny predmetov, ktoré máme študentom zapísať.

```
select pb.cis_predm, pb.skrok, pb.garant, pb.ects
  from st_program stp JOIN predmet_bod pb ON ( stp.cis_predm = pb.cis_predm
           AND stp.skrok = pb.skrok )
  where stp.rocnik = 1
    and stp.skrok = 2009
    and stp.st_odbor = 100
    and stp.st_zameranie = 0
    and stp.typ_povin = 'P';
```

4. Kartézsky súčin medzi predpripravenými množinami.

```
select st.os_cislo,cis_predm, skrok, pb.garant, pb.ects
  from student st,
       st_program stp JOIN predmet_bod pb USING ( cis_predm, skrok )
  where st.rocnik = 1
    and st.st_odbor = 100
    and st.rocnik = stp.rocnik
    and stp.skrok = 2009
    and st.st_odbor = stp.st_odbor
    and st.st_zameranie = stp.st_zameranie
    and stp.typ_povin = 'P';
```

5. Teraz môžeme pridať hlavičku na vloženie dát do tabuľky `zap_predmety`.

```
insert into zap_predmety
      ( os_cislo, cis_predm, skrok, prednasajuci, ects)
select st.os_cislo,cis_predm, skrok, pb.garant, pb.ects
  from student st,
       st_program stp JOIN predmet_bod pb USING ( cis_predm, skrok )
  where st.rocnik = 1
    and st.st_odbor = 100
    and st.rocnik = stp.rocnik
    and stp.skrok = 2009
    and st.st_odbor = stp.st_odbor
    and st.st_zameranie = stp.st_zameranie
    and stp.typ_povin = 'P';
```

4.2 Update

```
UPDATE tabulka
SET stlpec = hodnota
   [, stlpec = hodnota ]
WHERE podmienky;
```

- Príkazom Update môžeme zmeniť viaceré hodnoty LEN jednej tabuľky.
- Ak je podmienka na zmenu riadkov z inej tabuľky, je potrebné použiť vnorený Select.
- Ak máme nové hodnoty v pomocnej tabuľke, je potrebný select príkaz v časti SET. Aby nedošlo k strate dát, je potrebné doplniť aj podmienku, aby boli menené len riadky, ku ktorým máme hodnoty v pomocnej tabuľke.
- **Zmena hodnôt na základe podmienky z rovnakej tabuľky:**

```
UPDATE student
SET stav = 'K',
    ukoncenie = sysdate
where os_cislo = 5204;
```

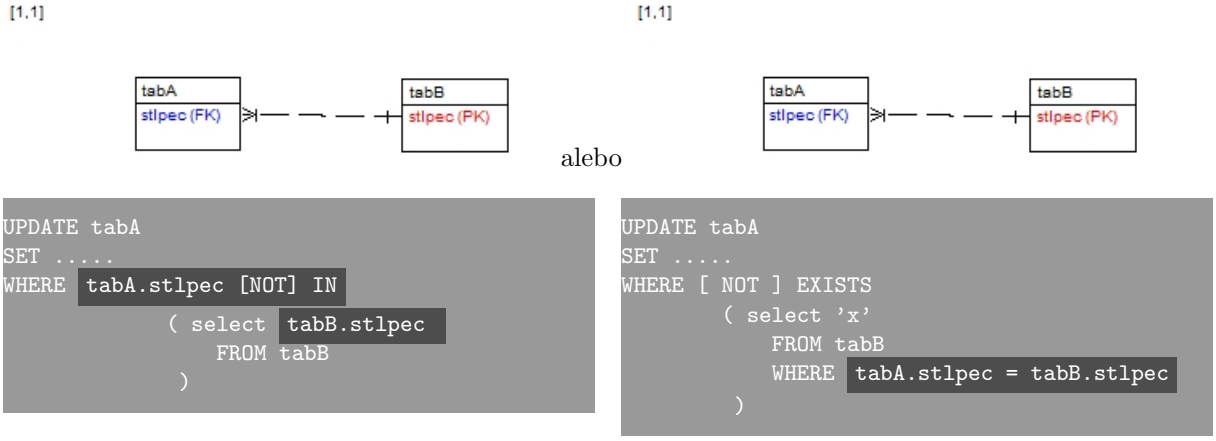
- **Zmena hodnôt na základe podmienky z druhej tabuľky:**

```
UPDATE student
SET stav = 'K',
    ukoncenie = sysdate
where os_cislo IN ( select os_cislo from ABSOLVENTI
                    where skrok = to_char( sysdate, 'YYYY') -1 );
```

- **Zmena hodnôt na hodnoty, ktoré sú v pomocnej tabuľke:**

```
UPDATE student
SET  st_skupina = ( select skupina from st_pom
                    where student.os_cislo = st_pom.os_cislo)
where EXISTS ( select skupina from st_pom
                      where student.os_cislo = st_pom.os_cislo);
```

4.2.1 Množina udaná pomocou vnoreného selectu - Jednoduchý primárny kľúč.



Príklad 4.2: Množina udaná pomocou vnoreného selectu.

- IN

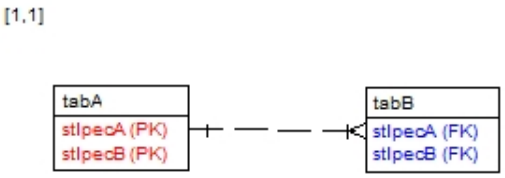
```
select ou.meno, ou.priezvisko
  from os_udaje ou JOIN student st USING (rod_cislo)
  where st.os_cislo NOT IN
       ( select zp.os_cislo
           from zap_predmety zp
           where zp.cis_predm = 'BI06'
       );
```

- EXISTS

```
select ou.meno, ou.priezvisko
  from os_udaje ou JOIN student st USING (rod_cislo)
  where  NOT EXISTS
       ( select 'x'
           from zap_predmety zp
           where zp.cis_predm = 'BI06'
             and zp.os_cislo = st.os_cislo
       );
```

4.2.2 Množina udaná pomocou vnoreného selectu - Kompozitný primárny kľúč.

Príklad 4.3: Množina udaná pomocou vnoreného selectu - Kompozitný primárny kľúč.



ZLE:

```
UPDATE tabA
SET .....
WHERE tabA.stlpecA [NOT] IN
            ( select tabB.stlpecA
                FROM tabB
             )
   AND tabA.stlpecB [NOT] IN
            ( select tabB.stlpecB
                FROM tabB
             )
```

DOBRE:

```
UPDATE tabA
SET .....
WHERE [ NOT ] EXISTS
       ( select 'x'
           FROM tabB
           WHERE tabA.stlpecA = tabB.stlpecA
             AND tabA.stlpecB = tabB.stlpecB
        )
```

4.3 Delete

```
DELETE FROM tabulka
WHERE podmienky;
```

- Slúži na vymazanie hodnôt LEN z jednej tabuľky.
- Ak je podmienka z inej tabuľky, musíme použiť vnorený select.
- **Vymazanie všetkých riadkov tabuľky.**

  ```
  DELETE FROM zap_predmety;
  ```

- **Vymazanie riadkov na základe podmienky z rovnakej tabuľky.**

  ```
  DELETE FROM zap_predmety
  WHERE cis_predm = 'BI06';
  ```

- **Vymazanie riadkov na základe podmienky z inej tabuľky.**

  ```
  DELETE FROM zap_predmety
  WHERE cis_predm IN ( select cis_predm from predmet
                       where nazov like 'Zaklady data%');
  ```

4.4 Poradie operácií

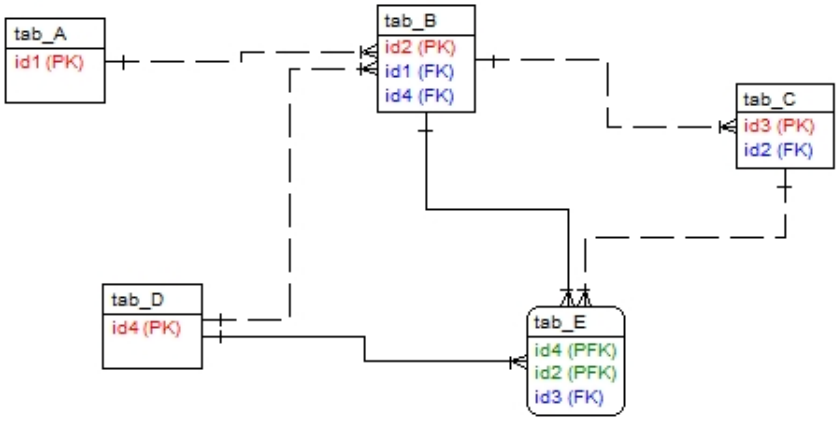
Foreign key - cudzí kľúč Hodnota Foreign key (FK - cudzí kľúč) je rovná hodnote primary key (PK - primárny kľúč), alebo hodnotu NULL.

!

- Insert, Load - Najprv musí byť zadaná hodnota primárneho kľúča.
- Delete - Najprv vymazávame hodnoty cudzieho kľúča.

Príklad 4.4: Poradie operácií

[1,1]



Poradie	Insert	Delete
tab_A,tab_B,tab_C,tab_D, tab_E	nie	nie
tab_A,tab_D,tab_B,tab_C, tab_E	ano	nie
tab_E,tab_C,tab_B,tab_D, tab_A	nie	áno
tab_D,tab_A,tab_B,tab_C, tab_E	áno	nie

4.5 Transakcie

4.5.1 DIS - Príkazy ukončenia transakcie

```
COMMIT - potvrdenie transakcie - t.j. zmeny budú natrvalo uložené.

ROLLBACK - neúspešné ukončenie transakcie - t.j. zmeny budú vrátené.

Všetky DDL-príkazy sú potvrdením transakcie.
```

Príklad 4.5: DIS - príkazy

```
SQL> create table pom ( cislo integer );

Table created.

SQL> insert into pom values ( 1 );
SQL> insert into pom values ( 2 );

SQL> commit;

Commit complete.
```

```
SQL> insert into pom values ( 3 );

SQL> select * from pom;

     CISLO
----------
         1
         2
         3

SQL> rollback;

Rollback complete.
```

```
SQL> select * from pom;

     CISLO
----------
         1
         2
```

Príklad 4.6: Ukončenie transakcií

```
SQL> create table pom1 ( cislo integer );

SQL> insert into pom1 values ( 11 );
SQL> insert into pom1 values ( 12 );

SQL> create table pom2 ( text varchar2(50));

SQL> rollback;

Rollback complete.
```

```
SQL> select * from pom1;

     CISLO
----------
        11
        12
```