

## PL/SQL - Trigger, Procedúra, Funkcia

### 9.1 Zadanie cvičenia - Trigger

Okrem vytvorenia a skompilovania triggra si nezabudnite aj **skontrolovať**, či a ako daný trigger **funguje**. !

- (a) Rozšírte tabuľku `zap_predmety` o dva stĺpce - `uzivatel`, `datum`.  
(b) Vytvorte trigger, ktorý pri zmene naplní tieto stĺpce do všetkých zmenených riadkov aktuálnym užívateľom a dátumom a časom tejto zmeny. (Tak, aby ste zabránili prepísaniu užívateľom.)
- Pomocou triggra (triggrov) zabezpečte, že žiaden študent nemôže predmet opakovať viac ako jeden krát. (Pri operácii `insert`)
- Spravte trigger, ktorý vám zabezpečí kaskádu na operáciu `Update` pre zmenu osobného čísla.
- Vytvorte tabuľku pre logovanie operácií vykonávaných nad tabuľkou `zap_predmety`. Potrebné je logovať – kto, kedy, akú operáciu vykonal a ako vyzeral PK pôvodného záznamu (v prípade `Delete`, `Update`) alebo nového záznamu (v prípade `Insert`).

### 9.2 Zadanie cvičenia - Procedúry, funkcie

- Vytvorte procedúru `Vyskladaj_skupinu`, ktorá má 6 parametrov:
  - pracovisko - vstupný parameter - skratka pracoviska
  - odbor - vstupný parameter - číslo štúdiijného odboru
  - zameranie - vstupný parameter - číslo zamerania
  - rocnik - vstupný parameter - číslo ročníka
  - kruzok - vstupný parameter - poradové číslo krúžku (1,2, ..., 9, A, B, ...)
  - st\_skupina - **výstupný** parameter - vyskladaná skupina.

**Príklad:**

vstupy: Z, 100, 0, 1, 2      výstup: 5ZI012  
          Z, 101, 0, 3, A      5ZP03A

Skratky odborov a zameraní je možné nájsť v tabuľke `priklad_db2.st_odbory`. !

2. Predchádzajúcu procedúru prepíšte na funkciu `f_vyskladaj_skupinu`, kde pôvodný výstupný parameter bude výstupom funkcie.
3. Pokúste sa použiť funkciu `f_vyskladaj_skupinu` v `selecte`.
4. Vytvorte procedúru `Vloz_predmet`, ktorá vykoná `INSERT` operáciu nového predmetu do tabuľky `PREDMET`.  
Zabezpečte aby boli vyplnené všetky `NOT NULL` stĺpce ešte pred pokusom o vloženie.  
Skompilujte procedúru a vložte nasledujúce predmety:
  - Číslo predmetu: BI14, názov: Databázové systémy
  - Číslo predmetu: BI12, názov: Úvod do inžinierstva
  - Číslo predmetu: BI12, názov: Úvod do inžinierstva( Čo sa stalo pri vkladaní tohto záznamu? Nastala chyba? Prečo? )
5. Vytvorte príslušnú `exception` na ošetrovanie predchádzajúcej chyby v procedúre `Vloz_predmet`

### 9.3 Domáca úloha - Trigger

1. Napíšte trigger `zp_nemaz`, ktorým zakážete vymazávať komukoľvek riadky z tabuľky `zap_predmety`.  
Následne sa pokúste vymazať niektorý riadok.
2. Vyskúšajte si odstavenie a znovu aktivovanie triggeru z predchádzajúceho príkladu.
3. Dronpnite trigger `zp_nemaz` z príkladu číslo 1.

### 9.4 Domáca úloha - Procedúry a funkcie

1. Upravte procedúru `Vloz_predmet`, tak aby ste skontrolovali skôr ako to tam vložíte, či je možné vložiť a ak nie vypíšete hlášku, inak tam vložte nové údaje.
2. Prepíšte procedúru `Vloz_predmet` na funkciu, tak aby návratová hodnota bola:
  - 1 - ak riadok bol úspešne vložený,
  - 0 - ak sa nepodarilo riadok vložiť.
3. Vytvorte procedúru `Zmen_predmet`, ktorá vykoná `UPDATE` operáciu názvu predmetu v tabuľke `PREDMET`.
  - Ako vstupný parameter odovzdajte číslo predmetu a nový názov predmetu.
  - Vytvorte `exception` pre ošetrovanie prípadu, keď zadaný predmet neexistuje, t.j. nenastane žiaden `UPDATE`.
  - Skompilujte procedúru a otestujte.
4. Vytvorte funkciu `Zrus_predmet`, ktorá vykoná `DELETE` operáciu predmetu v tabuľke `PREDMET`. Ako vstupný parameter odovzdajte číslo predmetu. Návratová hodnota funkcie bude počet vymazaných riadkov.