**Zadanie: Tvorba základných programových modulov**

1. **Vytvorte 1-3 trigre, ktoré budú kontrolovať správnosť zadávaných údajov**

/\* tento trigger zabezpeci, ze aj ked do tabulky FAKTURA budeme \*/

/\* chciet vlozit zaznam s prazdnym datumom objednania, nastavi \*/

/\* sa nam na aktualny den \*/

DELIMITER $$

CREATE TRIGGER faktura\_skontroluj\_datum

BEFORE INSERT ON faktura

FOR EACH ROW BEGIN

IF NEW.datum\_objednania = null THEN

SET NEW.datum\_objednania = CURDATE();

END IF;

END$$

DELIMITER ;

/\* dany trigger skontroluje vkladane data, ak ich hodnoty \*/

/\* su zaporne, vymeni ich za hodnoty nulove, podla ktorych sa \*/

/\* nasledne daju dohladat nekorektne zaznamy \*/

DELIMITER $$

CREATE TRIGGER objednane\_skontroluj\_data

BEFORE INSERT ON objednane

FOR EACH ROW BEGIN

IF NEW.id\_faktura < 0 THEN

SET NEW.datum\_objednania = 0;

END IF;

IF NEW.id\_vyr < 0 THEN

SET NEW.id\_vyr = 0;

END IF;

IF NEW.id\_faktura < 0 THEN

SET NEW.mnozstvo = 0;

END IF;

END$$

DELIMITER ;

/\* predpokladajme, ze si chceme udrzat aj povodne data o odberateloch \*/

/\* tento trigger zabezpeci, ze ked sa updatne zaznam o odberatelovi, \*/

/\* povodne hodnoty sa prenesu do tabulky ODBERATEL\_ZALOHA \*/

/\* predpokladame, ze tabulka ODBERATEL\_ZALOHA ma rovnaku strukturu ako tabulka ODBERATEL \*/

/\* plus obsahuje stlpec ZMENENE, ktory je typu date \*/

DELIMITER $$

CREATE TRIGGER update\_odberatel

BEFORE UPDATE ON odberatel

FOR EACH ROW BEGIN

INSERT INTO odberatel\_zaloha

SET action = 'update',

id\_odberatel = OLD.c\_odb,

ico\_rc = OLD.ico\_rc,

nazov = OLD.nazov,

mesto = OLD.mesto,

ulica = OLD.ulica,

cislo = OLD.cd,

zmenene = NOW();

END$$

DELIMITER ;

1. **Vytvorte 1-3 jednoduché procedúry**

/\* Procedura pre zistenie mnozstva vyrobkov na sklade podla id vyrobku \*/

DROP PROCEDURE IF EXISTS getQuantity;

DELIMITER //

CREATE PROCEDURE getQuantity( IN itemID INTEGER )

BEGIN

SELECT vyroba.mnozstvo FROM vyroba

WHERE vyroba.id\_mat=itemID;

END//

DELIMITER ;

/\* Procedura pre vypocet ceny vyrobku s DPH na zaklade zadanej sadzby DPH \*/

DROP PROCEDURE IF EXISTS getProductPriceWithTax;

DELIMITER //

CREATE PROCEDURE getProductPriceWithTax( IN itemID INTEGER, IN taxRate INTEGER )

BEGIN

SELECT vyrobky.nazov\_vyr AS Nazov, SUM(vyrobky.cena\_v\*(1+(taxRate/100))) FROM vyrobky

WHERE vyrobky.id\_vyr=itemID;

END//

DELIMITER ;

/\* Procedura pre vypisanie faktur po splatnosti \*/

DROP PROCEDURE IF EXISTS getOverDueInvoices;

DELIMITER //

CREATE PROCEDURE getOverDueInvoices()

BEGIN

SELECT faktura.id\_faktura AS 'ID objednavky', odberatel.nazov AS Odberatel,

DATE\_ADD( faktura.datum\_vystavenia, INTERVAL 14 DAY) AS Splatnost

FROM faktura

JOIN odberatel ON odberatel.c\_odb = faktura.id\_odberatel

WHERE faktura.datum\_vystavenia <= DATE\_ADD( faktura.datum\_vystavenia, INTERVAL 14 DAY );

END//

DELIMITER ;

1. **Vytvorte 1-3 jednoduché funkcie**

/\* funckcia zaokruhli zadanu sumu ak dostane ako parameter kladne cislo \*/

/\* inak vrati 0 \*/

DROP FUNCTION IF EXISTS zaokruhli\_sumu;

DELIMITER //

CREATE FUNCTION zaokruhli\_sumu(suma FLOAT)

RETURNS INT

BEGIN

IF suma > 0 THEN

RETURN ROUND(suma, 0);

END IF;

RETURN 0;

END //

DELIMITER ;

DELIMITER //

CREATE FUNCTION addTaxToPrice(price FLOAT(5,2))

RETURNS FLOAT(6,2)

BEGIN

declare outPrice FLOAT(6,2);

SET outPrice = price\*(1.2);

RETURN outPrice;

END //

DELIMITER ;

1. **Vytvorte aspoň 1-3 transakcie, ktoré budú obsahovať priebežné body, body návratu**

/\* demonstracia pouzitia priebeznych bodov navratu \*/

CREATE TABLE t\_test (id INT NOT NULL PRIMARY KEY) ENGINE=InnoDB;

START TRANSACTION;

INSERT

INTO t\_test

VALUES (1);

SELECT \*

FROM t\_test;

id

---

1

SAVEPOINT tran2;

INSERT

INTO t\_test

VALUES (2);

SELECT \*

FROM t\_test;

id

---

1

2

ROLLBACK TO tran2;

SELECT \*

FROM t\_test;

id

---

1

ROLLBACK;

SELECT \*

FROM t\_test;

id

---

1. **Trigger pre tabulku t1**

CREATE TABLE IF NOT EXISTS t1(a INT(3),b FLOAT(5,2));

/\* ak vkladane cislo do stlpca je delitelne piatimi, \*/

/\* do stlpca b sa zapise hodnota 100 \*/

DELIMITER //

CREATE TRIGGER devideFive

BEFORE INSERT ON t1

FOR EACH ROW BEGIN

IF (NEW.a % 5) = 0 THEN

SET NEW.b = 100;

END IF;

END//