/\* procedura nam vlozi riadok do tabulky customer order \*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `create\_customer\_order`(

IN identification\_no\_ CHAR(11),

IN status\_ VARCHAR(15),

OUT order\_id\_ INT)

BEGIN

INSERT INTO `customer\_order` (identification\_no, status, order\_date) values (identification\_no\_, status\_, sysdate());

SELECT LAST\_INSERT\_ID() into order\_id\_;

END

/\*procedura ktora pride rezervaciu \*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `create\_reservation`(

IN user\_id\_ INTEGER,

IN ico\_ CHAR(8),

IN service\_id\_ INTEGER,

IN date\_ CHAR(10),

IN hour\_ INTEGER

)

BEGIN

DECLARE order\_id\_ INTEGER;

DECLARE cust\_identificaton\_no\_ CHAR(11);

DECLARE emp\_identificaton\_no\_ CHAR(11);

DECLARE hire\_date\_ DATE;

DECLARE hour\_duration\_ INTEGER;

SELECT

   identification\_no

INTO cust\_identificaton\_no\_ FROM

   customer

WHERE

   user\_id = user\_id\_;

SELECT

   hour\_duration

INTO hour\_duration\_ FROM

   Services

WHERE

   service\_id = service\_id\_;

CALL create\_customer\_order(cust\_identificaton\_no\_,'prijata', order\_id\_);

SELECT

   hire\_date, ico, identification\_no

INTO hire\_date\_, ico\_, emp\_identificaton\_no\_  FROM

   employee

WHERE

   ico = ico\_

       AND (termination\_date IS NULL

       OR termination\_date >= (SELECT SYSDATE()))

       AND work\_position IN (SELECT

           type

       FROM

           services

       WHERE

           service\_id = service\_id\_);

INSERT INTO reservation (order\_id, hire\_date, ICO, identification\_no, service\_id, repair\_date, work\_hours) values (order\_id\_, hire\_date\_, ico\_, emp\_identificaton\_no\_, service\_id\_, ADDTIME(CONVERT(str\_to\_date(date\_, '%Y-%m-%d'), DATETIME), SEC\_TO\_TIME(hour\_\*3600)) , hour\_duration\_);

END

/\* procedura, ktora nam vytvori zakaznika  \*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `create\_customer`(

IN psc\_ CHAR(5),

IN country\_id\_ CHAR(3),

IN town\_name\_ VARCHAR(100),

IN identification\_no\_ CHAR(11),

IN first\_name\_ VARCHAR(15),

IN last\_name\_ VARCHAR(20),

IN street\_ VARCHAR(30),

IN orientation\_no\_ INTEGER,

IN user\_id\_ INTEGER)

BEGIN

DECLARE person\_ident\_no\_ CHAR(11);

DECLARE cust\_ident\_no\_ CHAR(11);

SELECT

   identification\_no

INTO person\_ident\_no\_ FROM

   person

WHERE

   identification\_no = identification\_no\_;

SELECT

   identification\_no

INTO cust\_ident\_no\_ FROM

   customer

WHERE

   identification\_no = identification\_no\_;

IF (person\_ident\_no\_ IS NULL) THEN

IF NOT EXISTS(SELECT \* FROM town WHERE town\_id = psc\_) THEN

INSERT INTO town (town\_id, country\_id, town\_name) VALUES (psc\_, country\_id\_, town\_name\_);

END IF;

INSERT INTO person (identification\_no, town\_id, first\_name, last\_name, street, orientation\_no) VALUES (identification\_no\_, psc\_, first\_name\_, last\_name\_, street\_, orientation\_no\_);

IF (user\_id\_ IS NULL) THEN

INSERT INTO customer (identification\_no) VALUES (identification\_no\_);

   ELSE

INSERT INTO customer (identification\_no, user\_id) VALUES (identification\_no\_, user\_id\_);

   END IF;

ELSEIF (cust\_ident\_no\_ IS NULL) THEN

IF (user\_id\_ IS NULL) THEN

INSERT INTO customer (identification\_no) VALUES (identification\_no\_);

   ELSE

INSERT INTO customer (identification\_no, user\_id) VALUES (identification\_no\_, user\_id\_);

   END IF;

END IF;

END

/\*Procedura, ktora nam aktualizuje profil uzivatela \*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `update\_customer`(

IN country\_id\_ CHAR(3),

IN psc\_ CHAR(5),

IN town\_name\_ VARCHAR(100),

IN first\_name\_ VARCHAR(15),

IN last\_name\_ VARCHAR(20),

IN street\_ VARCHAR(30),

IN orientation\_no\_ INTEGER,

IN user\_id\_ INTEGER

)

BEGIN

DECLARE identification\_no\_ CHAR(11);

SELECT

   identification\_no

INTO identification\_no\_ FROM

   customer

WHERE

   user\_id = user\_id\_;

IF NOT EXISTS(SELECT \* FROM town WHERE town\_id = psc\_) THEN

INSERT INTO town VALUES (psc\_,country\_id\_,town\_name\_);

END IF;

UPDATE person

SET town\_id = psc\_, first\_name = first\_name\_, last\_name = last\_name\_, street = street\_, orientation\_no = orientation\_no\_

WHERE identification\_no = identification\_no\_;

END

/\* procedura ktora nam updatne zaznam shopping\_cart\*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `update\_shopping\_cart`(

IN user\_id\_ INTEGER,

IN car\_part\_id\_ INTEGER,

IN quantity\_ INTEGER

)

BEGIN

UPDATE shopping\_cart

SET quantity = quantity\_

WHERE user\_id = user\_id\_

AND car\_part\_id = car\_part\_id\_;

END

/\* Procedura, ktora maze veci z shopping\_cart \*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `delete\_from\_shopping\_cart`(

IN user\_id\_ INTEGER,

IN car\_part\_id\_ INTEGER

)

BEGIN

DELETE FROM shopping\_cart where user\_id = user\_id\_ and car\_part\_id = car\_part\_id\_;

END

/\*Procedura, ktora prida súčiastky do košíka \*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `add\_to\_shopping\_cart`(

IN user\_id\_ INTEGER,

IN car\_part\_id\_ INTEGER,

IN quantity\_ INTEGER

)

BEGIN

DECLARE stock\_ INTEGER;

DECLARE quantity\_in\_cart\_ INTEGER;

SELECT

   stock

INTO stock\_ FROM

   car\_parts

WHERE

   car\_part\_id = car\_part\_id\_;

SELECT

   quantity

INTO quantity\_in\_cart\_ FROM

   shopping\_cart

WHERE

   user\_id = user\_id\_

       AND car\_part\_id = car\_part\_id\_;

IF EXISTS(SELECT \* FROM shopping\_cart where user\_id = user\_id\_ and car\_part\_id = car\_part\_id\_) THEN

IF ( (quantity\_ + quantity\_in\_cart\_) > stock\_) THEN

UPDATE   shopping\_cart

SET quantity = stock\_

WHERE    user\_id = user\_id\_

       AND car\_part\_id = car\_part\_id\_;

ELSE

UPDATE   shopping\_cart

SET quantity = (quantity\_in\_cart\_ + quantity\_)

WHERE    user\_id = user\_id\_

       AND car\_part\_id = car\_part\_id\_;

END IF;

ELSE

INSERT INTO shopping\_cart (user\_id, car\_part\_id, quantity) VALUES (user\_id\_, car\_part\_id\_, quantity\_);

END IF;

END

/\*Procedúra, ktorá zoberie veci, čo sú v košíku, vytvorí objednávku a veci pridá do order\_items \*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `confirm\_shopping\_cart`(

IN user\_id\_ INTEGER,

IN cust\_order\_id\_ INTEGER)

BEGIN

DECLARE identification\_no\_ CHAR(11);

DECLARE order\_id\_ INTEGER;

DECLARE car\_part\_id\_ INTEGER;

DECLARE quantity\_ INTEGER;

DECLARE v\_last\_row\_fetched BOOLEAN;

DECLARE shopping\_cart\_cursor CURSOR FOR

SELECT

   car\_part\_id, quantity

FROM

   shopping\_cart

WHERE

   user\_id = user\_id\_;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET  v\_last\_row\_fetched = 1;

IF (cust\_order\_id\_ IS NULL) THEN

SELECT

identification\_no

INTO identification\_no\_ FROM

customer

WHERE

user\_id = user\_id\_;

CALL create\_customer\_order(identification\_no\_,'zrealizovana',order\_id\_);

END IF;

OPEN shopping\_cart\_cursor;

shopping\_cart: LOOP

FETCH shopping\_cart\_cursor INTO car\_part\_id\_,quantity\_;

   IF (v\_last\_row\_fetched = 1) THEN

           LEAVE shopping\_cart;

END IF;

IF (cust\_order\_id\_ IS NOT NULL) THEN

INSERT INTO order\_items VALUES (cust\_order\_id\_,car\_part\_id\_,quantity\_);

   ELSE

INSERT INTO order\_items VALUES (order\_id\_,car\_part\_id\_,quantity\_);

   END IF;

   DELETE FROM shopping\_cart where user\_id = user\_id\_ AND car\_part\_id = car\_part\_id\_;

END LOOP shopping\_cart;

CLOSE shopping\_cart\_cursor;

SET  v\_last\_row\_fetched = 0;

END

/\*Procedúra, ktorá nám odstráni rezerváciu podľa id reservation\_id \*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `delete\_from\_reservation`(

IN reservation\_id\_ INTEGER

)

BEGIN

DELETE FROM reservation where reservation\_id = reservation\_id\_;

END

/\* Procedura, ktora prida zamestnanca \*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `create\_employee`(

IN country\_id\_ CHAR(3),

IN town\_id\_ CHAR(5),

IN town\_name\_ VARCHAR(100),

IN identification\_no\_ CHAR(11),

IN first\_name\_ VARCHAR(15),

IN last\_name\_ VARCHAR(20),

IN street\_ VARCHAR(30),

IN orientation\_no\_ INTEGER,

IN hire\_date\_ CHAR(10),

IN ico\_ CHAR(8),

IN work\_position\_ VARCHAR(20),

IN hour\_start\_ INTEGER,

IN hour\_end\_ INTEGER,

IN price\_per\_hour\_ FLOAT,

IN termination\_date\_ CHAR(10)

)

BEGIN

IF EXISTS(select \* from person where identification\_no = identification\_no\_) THEN

IF NOT EXISTS (select \* from town where town\_id = town\_id\_) THEN

INSERT INTO Town VALUES (town\_id\_,country\_id\_,town\_name\_);

   END IF;

   UPDATE Person

   SET town\_id = town\_id\_,

       first\_name = first\_name\_,

       last\_name = last\_name\_,

       street = street\_,

       orientation\_no = orientation\_no\_

WHERE identification\_no = identification\_no\_;

   INSERT INTO Employee values (hire\_date\_,ico\_,identification\_no\_,work\_position\_,hour\_start\_,hour\_end\_,price\_per\_hour\_,termination\_date\_);

ELSE

IF NOT EXISTS (select \* from town where town\_id = town\_id\_) THEN

INSERT INTO Town VALUES (town\_id\_,country\_id\_,town\_name\_);

   END IF;

   INSERT INTO Person values (identification\_no\_,town\_id\_,first\_name\_,last\_name\_,street\_,orientation\_no\_);

INSERT INTO Employee values (hire\_date\_,ico\_,identification\_no\_,work\_position\_,hour\_start\_,hour\_end\_,price\_per\_hour\_,termination\_date\_);

END IF;

END

/\* Procedura, ktora aktualizuje termination\_date pri zamestnancovi \*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `terminate\_employee`(

IN identification\_no\_ CHAR(11),

IN term\_date\_ CHAR(10)

)

BEGIN

UPDATE employee

   SET termination\_date = term\_date\_

   WHERE identification\_no = identification\_no\_;

END

/\* Procedura zmaze absenciu \*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `delete\_from\_absence`(

IN absence\_id\_ INTEGER

)

BEGIN

DELETE FROM absence WHERE absence\_id = absence\_id\_;

END

/\* Procedure prida novu absenciu zamestnancovi \*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `add\_absence`(

IN identification\_no\_ CHAR(11),

IN absence\_from\_ CHAR(10),

IN absence\_to\_ CHAR(10)

)

BEGIN

DECLARE hire\_date\_ DATE;

DECLARE ico\_ CHAR(8);

SELECT hire\_date, ico INTO hire\_date\_,ico\_ FROM employee where identification\_no = identification\_no\_ AND (termination\_date is null or termination\_date >= (select sysdate()));

INSERT INTO absence (hire\_date, ico, identification\_no, absence\_from, absence\_to) VALUES (hire\_date\_,ico\_,identification\_no\_,absence\_from\_,absence\_to\_);

END

/\*Procedura nastavy objednavku na zrealizovanu \*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `realize\_reservation`(

IN reservation\_id\_ INTEGER,

IN work\_hours\_ INTEGER

)

BEGIN

DECLARE order\_id\_ INTEGER;

SELECT

   order\_id

INTO order\_id\_ FROM

   reservation

WHERE

   reservation\_id = reservation\_id\_;

UPDATE reservation

SET work\_hours = work\_hours\_

WHERE reservation\_id = reservation\_id\_;

UPDATE customer\_order

SET status = 'zrealizovana'

WHERE  order\_id = order\_id\_;

END

/\* Procedure aktualizuje mnozstvo na sklade \*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `update\_car\_parts`(

IN car\_part\_id\_ INTEGER,

IN stock\_ INTEGER

)

BEGIN

UPDATE car\_parts

SET stock = stock\_

WHERE car\_part\_id = car\_part\_id\_;

END

/\*Procedura vytvorí novú car brand \*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `create\_car\_brand`(

IN brand\_name\_ VARCHAR(30)

)

BEGIN

INSERT INTO car\_brand (brand\_name) VALUES (brand\_name\_);

END

/\*Procedura vytvorí novú car type\*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `create\_car\_type`(

IN brand\_id\_ INTEGER,

IN car\_type\_name\_ VARCHAR(100)

)

BEGIN

INSERT INTO car\_type (brand\_id, car\_type\_name) VALUES (brand\_id\_,car\_type\_name\_);

END

/\*Procedura vytvorí novú car part\*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `create\_car\_part`(

IN car\_type\_id\_ INTEGER,

IN part\_name\_ VARCHAR(100),

IN part\_price\_ FLOAT,

IN stock\_ INTEGER,

IN image\_ VARCHAR(255)

)

BEGIN

INSERT INTO car\_parts (car\_type\_id, part\_name, part\_price, stock, image) VALUES (car\_type\_id\_, part\_name\_, part\_price\_, stock\_, image\_);

END

/\*Procedure aktualizuje tabulku services \*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `update\_services`(

IN service\_id\_ INTEGER,

IN hour\_duration\_ INTEGER,

IN price\_per\_hour\_ INTEGER

)

BEGIN

UPDATE services

SET hour\_duration = hour\_duration\_,

price\_per\_hour = price\_per\_hour\_

WHERE service\_id = service\_id\_;

END

/\* Prida novy service do tabulky \*/

CREATE DEFINER=`root`@`localhost` PROCEDURE `create\_service`(

IN town\_name\_  VARCHAR(200) CHARACTER SET utf8,

IN type\_ VARCHAR(50),

IN hour\_duration\_ INTEGER,

IN price\_per\_hour\_ INTEGER

)

BEGIN

INSERT INTO services (name, type,hour\_duration, price\_per\_hour) VALUES (town\_name\_,type\_,hour\_duration\_,price\_per\_hour\_);

END