МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

Кафедра компьютерных технологий и программной инженерии

ОТЧЁТ
ЗАЩИЩЁН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

старший преподаватель		Путилова Н. В.
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

Отчёт по лабораторной работе №6

по курсу: Проектирование баз данных

СТУДЕНТ ГР. № 4932 (номер группы) _____ (подпись, дата) С. И. Коваленко (инициалы, фамилия)

Санкт-Петербург
2021

Цель работы: создать в БД хранимые процедуры, реализующие:
— вставку с пополнением справочников (получаем ссылку на внешний ключ по значению данных из родительской таблицы, если данных нет- добавляем в родительскую, затем вставляем в дочернюю);
— удаление с очисткой справочников – удаление данных из родительской таблицы, если после удаления данных из дочерней у строки родительской больше нет зависимых (удаляется информация о студенте, если в его группе нет больше студентов, запись удаляется из таблицы с перечнем групп);
— каскадное удаление (удаление всех зависимых данных);
— вычисление и возврат значения агрегатной функции (т.к. агрегатная функция дает единственный результат) (задача- вернуть данные из процедуры/функции);
— формирование статистики во временной таблице. (задача- работа с временными таблицами).

```
вставку с пополнением справочников (получаем ссылку на внешний ключ по значению данных из родительской таблицы, если данных нет- добавляем в родительскую, затем вставляем в дочернюю)

DELIMITER //

CREATE PROCEDURE insert_car(
            new_car_number INT,
            owner_name VARCHAR(20),
            owner_surname VARCHAR(20),
            owner_middle_name VARCHAR(20),
            brand_name_ VARCHAR (20))
 BEGIN
   DECLARE id_new_owner INT;
   DECLARE id_new_brand INT;

   IF EXISTS (
      SELECT *
      FROM owner
      WHERE owner.Name = owner_name
        AND owner.Surname = owner_surname
        AND owner.Middle_name = owner_middle_name
      )
   THEN
      SELECT owner.id_owner INTO id_new_owner
      FROM owner
      WHERE owner.Name = owner_name
        AND owner.Surname = owner_surname
        AND owner.Middle_name = owner_middle_name;
   ELSE
      BEGIN
         INSERT INTO owner(Name, Surname, Middle_name)
         VALUES (owner_name, owner_surname, owner_middle_name);
         SET id_new_owner = last_insert_id();
```

```
    END;
  END IF;

  IF EXISTS (
    SELECT *
    FROM brand
    WHERE brand_name_ = brand.brand_name
    )
  THEN
    SELECT brand.id_brand_car INTO id_new_brand
    FROM brand
    WHERE brand_name_ = brand.brand_name;
  ELSE
    BEGIN
      INSERT INTO brand(brand_name)
      VALUES (brand_name_);
      SET id_new_brand = last_insert_id();
    END;
  END IF;

  INSERT INTO car (car_number, id_owner, id_brand_car)
  values (new_car_number, id_new_owner, id_new_brand);
END; //

CALL insert_car(10, 'аа', 'бб', 'вв','Уаз');
CALL insert_car(12, 'Пётр', 'Петров', 'Пертрович', 'Уаз')
```

удаление с очисткой справочников – удаление данных из родительской таблицы, если после удаления данных из дочерней у строки родительской больше нет зависимых (удаляется информация о студенте, если в его группе нет больше студентов, запись удаляется из таблицы с перечнем групп);

```
DELIMITER //

CREATE PROCEDURE delete_car(
            del_car_number INT)
 BEGIN
   DECLARE id_proc_owner INT;
   DECLARE id_proc_brand INT;

   SELECT car.id_brand_car INTO id_proc_brand
   FROM car
   WHERE car.car_number = del_car_number;
   SELECT car.id_owner INTO id_proc_owner
   FROM car
   WHERE car.car_number = del_car_number;

   DELETE FROM car
   WHERE car.car_number = del_car_number;

   IF NOT EXISTS(
      SELECT *
      FROM car
      WHERE car.id_owner = id_proc_owner
      )
   THEN
      DELETE FROM owner
         WHERE id_owner = id_proc_owner;
   END IF;
   IF NOT EXISTS(
      SELECT *
      FROM car
      WHERE car.id_brand_car = id_proc_brand
      )
   THEN
      DELETE FROM brand
         WHERE id_brand_car = id_proc_brand;
   END IF;

 END; //

CALL delete_car(10);
CALL delete_car(12)
```

```
каскадное удаление (удаление всех зависимых данных);

DELIMITER //

CREATE PROCEDURE cascade_delete_owner(del_id_owner INT)
 BEGIN
    DELETE FROM `check`
    WHERE `check`.car_number in
    (
       SELECT car.car_number
       FROM car
       WHERE car.id_owner = del_id_owner
    );

    DELETE FROM car
    WHERE id_owner = del_id_owner;

    DELETE FROM owner
    WHERE owner.id_owner = del_id_owner;
 END//

 CALL cascade_delete_owner(1)
```

```
 вычисление и возврат значения агрегатной функции (т.к. агрегатная функция дает единственный результат)
( задача- вернуть данные из процедуры/функции);

DELIMITER //

CREATE PROCEDURE count_car(OUT car_count INT)
BEGIN
   SELECT ifnull(count(car_number),0) into car_count
   from car;
END;

CALL count_car(@c);
SELECT @c
```

```
/* формирование статистики во временной таблице. (задача- работа с временными таблицами ). */

DELIMITER //

DROP PROCEDURE owner_statistic;

CREATE PROCEDURE owner_statistic()
 BEGIN
   DROP TABLE IF EXISTS statistic;
   CREATE TEMPORARY TABLE statistic(
      id_stat INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
```

```
    id_owner INT,
    count_car INT,
    avg_count_check DOUBLE DEFAULT 0
  )
  AUTO_INCREMENT = 1;

  INSERT INTO statistic(id_owner, count_car)
  SELECT owner.id_owner, count(car.car_number) as count_clothes
  FROM owner
  LEFT JOIN car ON owner.id_owner = car.id_owner
  GROUP BY id_owner;

  UPDATE statistic
        SET avg_count_check =
        (
                SELECT avg(t.count_clothes)
                FROM
                (
                        SELECT count(`check`.id_check) as count_clothes
      FROM `check`
                        LEFT JOIN car ON `check`.id_check = car.id_owner
                        GROUP BY id_owner
                ) AS t
        );

  SELECT DISTINCT
    statistic.id_owner,
    statistic.count_car,
    statistic.avg_count_check
  FROM statistic;

  DROP TABLE statistic;
END;//

CALL owner_statistic();
```