Санкт-Петербургский Политехнический Университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**ОТЧЕТ**

**по лабораторной работе**

**«Язык SQL-DML»**

По дисциплине «Базы данных»

Работу выполнил студент

Группа 3530901/60202

Курякин Д. А.

Работу принял преподаватель

Мяснов А.В.

Санкт-Петербург

2020

**Содержание**

[1. Цель работы 2](#_Toc38812630)

[2. Программа работы 3](#_Toc38812631)

[3. Стандартные запросы 3](#_Toc38812632)

[3.1. Выборка с использованием логических операции 3](#_Toc38812633)

[3.2. Запрос с вычисляемым полем 4](#_Toc38812634)

[3.3. Выборка с использованием сортировки 4](#_Toc38812635)

[3.4. Запрос с вычисление совокупных характеристик таблицы 5](#_Toc38812636)

[3.5. Выборка из связанных таблиц 5](#_Toc38812637)

[3.6 Запрос с использованием группировки 6](#_Toc38812638)

[3.7 Вложенный запрос 7](#_Toc38812639)

[3.8 Вставка записей 7](#_Toc38812640)

[3.9 Изменение записей 8](#_Toc38812641)

[3.10 Удаление записей по условию 9](#_Toc38812642)

[4. Запросы по заданию преподавателя 10](#_Toc38812643)

[4.1. Вывести 10 водителей, которые имеют наибольшее отношение количества нарушений к количеству ДТП, где они были виновниками. 10](#_Toc38812644)

[4.2. Вывести количество нарушений каждого вида для каждого года. 10](#_Toc38812645)

[4.3. Вывести 10 моделей автомобилей, владельцы которых чаще всего попадают в аварии. 11](#_Toc38812646)

[5. Представления и хранимые процедуры 12](#_Toc38812647)

[Вывод 14](#_Toc38812648)

**1. Цель работы**

Познакомиться с языком создания запросов управления данными SQL-DML.

**2. Программа работы**

1. Изучение SQL-DML.

2. Выполнение всех запросов из списка стандартных запросов. Демонстрация результатов представилю.

3. Получение у преподавателя и реализация SQL-запросов в соответствии с индивидуальным заданием. Демонстрация результатов преподавателю.

4. Сохранение в БД выполненных запросов SELECT в виде представлений, запросов INSERT, UPDATE или DELETE — в виде ХП. Выкладывание скрипта в GitLab.

**3. Стандартные запросы**

**3.1. Выборка с использованием логических операции**

Сделайте выборку данных из одной таблицы при нескольких условиях, с использованием логически операции LIKE, BETWEEN, IN (не менее 3-х разных примеров).

Выборка из первых 4 автомобилей у которых категория управления B(1):

SELECT \*

FROM car

WHERE id IN (1, 2, 3, 4)

AND categories = 1;



Выборка 20 первых инспекторов у которых people\_id > 100:

SELECT \*

FROM inspector

WHERE id BETWEEN 0 AND 20

AND people\_id > 100

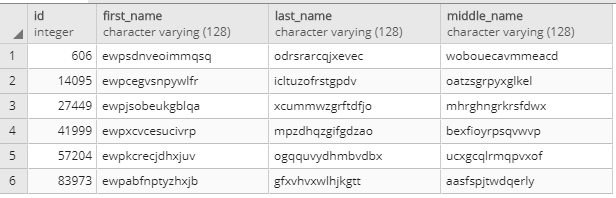


Выборка всех у которых имя начинается на ewp:

SELECT \*

FROM people

WHERE first\_name LIKE 'ewp%'



**3.2. Запрос с вычисляемым полем**

Создайте в поле вычисляемое поле.

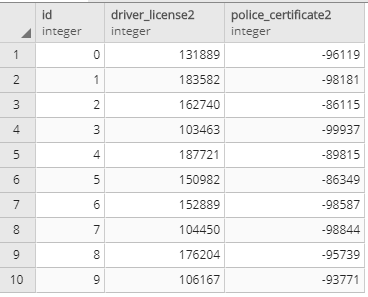
Выборка штрафов у которых driver\_license + 100000 и police\_certificate - 100000

SELECT id, driver\_license + 100000 AS driver\_license2,

police\_certificate - 100000 AS police\_certificate2

FROM fine

LIMIT 10;

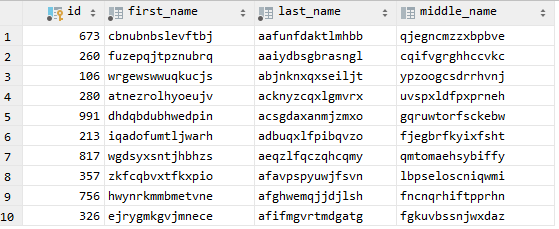


**3.3. Выборка с использованием сортировки**

Сделайте выборку данных с сортировкой по нескольким полям.

Выборка людей с отсортированными фамилиями по возрастанию.

SELECT *\**FROM people  
ORDER BY last\_name ASC, first\_name ASC  
LIMIT 10;



**3.4. Запрос с вычисление совокупных характеристик таблицы**

Создайте запрос, вычисляющий несколько совокупных характеристик таблицы.

Запрос среднее значение номеров водительского удостоверения, последний номер полицейского удостоверения, максимальную дату и минимальную дату.

SELECT COUNT(\*) AS count,

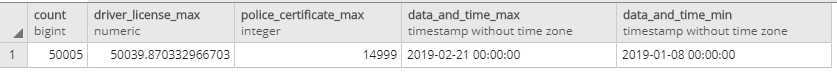
AVG (driver\_license) AS driver\_license\_max,

MAX(police\_certificate) AS police\_certificate\_max,

MAX(data\_and\_time) AS data\_and\_time\_max,

MIN(data\_and\_time) AS data\_and\_time\_min

FROM fine;



**3.5. Выборка из связанных таблиц**

Сделайте выборку из связанных таблиц не менее двух примеров.

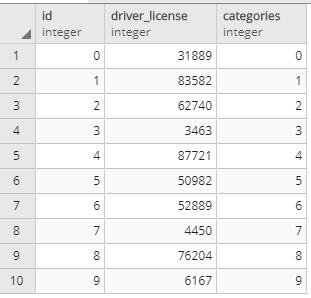
Соединение штрафов и водительских удостоверений:

SELECT fine.id, driver\_license, driver\_license.categories

FROM fine

JOIN driver\_license ON fine.id = driver\_license.id

LIMIT 10;



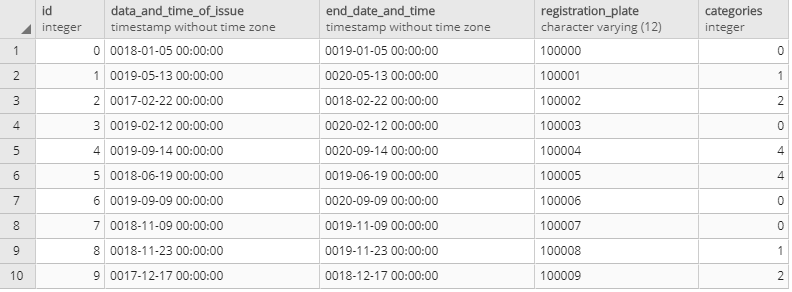
Соединение полиса ОСАГО и автомобилей:

SELECT osago.id, osago.data\_and\_time\_of\_issue, osago.end\_date\_and\_time, car.registration\_plate, car.categories

FROM osago

JOIN car ON osago.car = car.id

LIMIT 10;



**3.6 Запрос с использованием группировки**

Создайте запрос, рассчитывающий совокупную характеристику с использованием группировки, наложите ограничение на результат группировки.

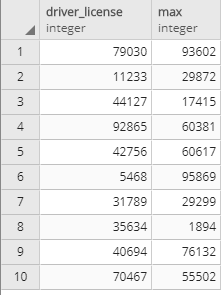
Выборка показывающая последний номер штрафа у водителя:

SELECT driver\_license, MAX(id\_violation)

FROM fine

GROUP BY driver\_license

LIMIT 10;



**3.7 Вложенный запрос**

Придумайте пример использования вложенного запроса.

Выборка показывающая максимальное количество штрафов

SELECT MAX(c.cnt) AS maxFine

FROM(SELECT COUNT(driver\_license) AS cnt

FROM fine

GROUP BY driver\_license) c



**3.8 Вставка записей**

С помощью оператора INSERT добавьте в каждую таблицу по одной записи.

INSERT INTO people ( id, first\_name, last\_name, middle\_name)

VALUES (1, 'Александр', 'Сидоров', 'Александрович');

INSERT INTO driver\_license ( id, numer, categories, data\_and\_time\_of\_issue, end\_date\_and\_time, unit\_gipdd, people\_id)

VALUES (1, 782212, 1, '2010-05-01', '2020-05-01', 'ГИБДД 7816', 1);

INSERT INTO inspector ( id, police\_certificate, rank, people\_id)

VALUES (1, 12345, 'Лейтенант', 1);

INSERT INTO machine\_directory ( id, brand, model)

VALUES (1, 'LADA', 'Vesta');

INSERT INTO violation ( id, title, punishment)

VALUES (1, 1000, 'Превышение скорости');

INSERT INTO dir\_categories ( id, name)

VALUES (1, 'A');

INSERT INTO categories ( categories, id\_categories)

VALUES (1, 1);

INSERT INTO car ( id, registration\_plate, brand\_and\_monel, categories, people\_id)

VALUES (1, 'B123KX178', 1, 2, 1);

INSERT INTO fine ( id, driver\_license, police\_certificate, data\_and\_time, id\_violation, people\_id)

VALUES (1, 1, 2, '2018-12-10', 1, 1);

INSERT INTO dtp ( id, injured, guilty, damage\_car)

VALUES (1, 1, 1, 1);

INSERT INTO details ( id, name)

VALUES (1, 'Левое крыло');

INSERT INTO damage\_car (damage\_car, damage)

VALUES (1, 1);

INSERT INTO damage (damage\_id, damage)

VALUES (1, 1);

INSERT INTO injured (injured, injured\_id)

VALUES (1, 1);

INSERT INTO guilty (guilty, guilty\_id)

VALUES (2, 1);

INSERT INTO osago (id, data\_and\_time\_of\_issue, end\_date\_and\_time, drivers)

VALUES (1, '2018-05-01', '2019-05-01', 1);

INSERT INTO drivers (drivers, drivers\_id)

VALUES (1, 1);

**3.9 Изменение записей**

С помощью оператора UPDATE измените значения нескольких полей у всех записей, отвечающих заданному условию.

Запишем все записи о штрафах водителя 72849 на водитела 72850:

SELECT \*

FROM fine

WHERE driver\_license = 72849 or driver\_license = 72850;

UPDATE fine

SET driver\_license = 72850

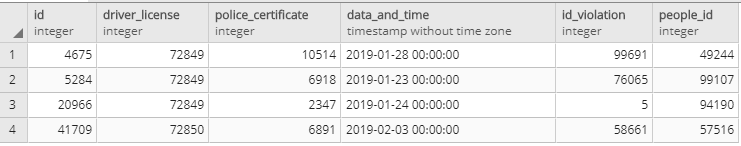
WHERE driver\_license = 72849;

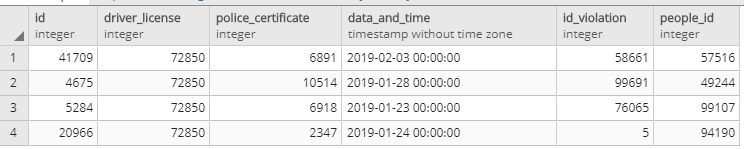
SELECT \*

FROM fine

WHERE driver\_license = 72849 or driver\_license = 72850;

До UPDATE



После UPDATE 

**3.10 Удаление записей по условию**

С помощью оператора DELETE удалите запись, имеющую максимальное (минимальное) значение некоторой совокупной характеристики.

Удаляем все штрафы водителя 72850:

SELECT \*

FROM fine

WHERE driver\_license = 72850;

DELETE

FROM fine

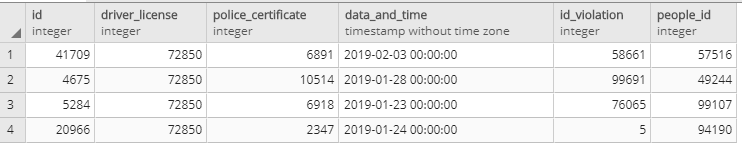
WHERE driver\_license = 72850;

SELECT \*

FROM fine

WHERE driver\_license = 72850;

До DELETE



После DELETE

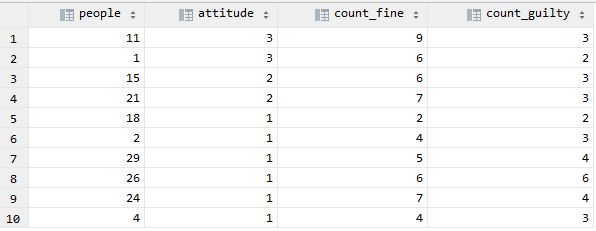


# **4. Запросы по заданию преподавателя**

## **4.1. Вывести 10 водителей, которые имеют наибольшее отношение количества нарушений к количеству ДТП, где они были виновниками.**

С вызываем внешние ключи таблицы guilty с внешними ключами таблиц dtp и car, группируем по столбцу people\_id таблицы car и выводим id водителя и количество нарушений водителя человека. С вызываем внешний ключ driver\_license таблицы fine с внешним ключом id таблицы driver\_license, группируем по столбцы people\_id табицы driver\_license и выводим id водителя и количество его штрафов.

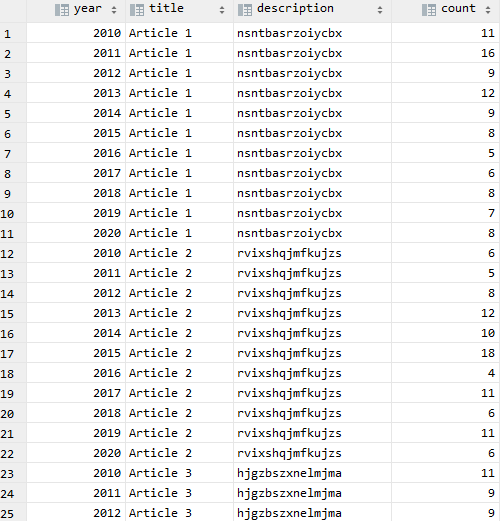
WITH dtp\_new AS (SELECT car.people\_id AS people, *count*(car.people\_id) AS count\_guilty  
 FROM guilty  
 JOIN dtp ON guilty.guilty = dtp.guilty  
 JOIN car ON guilty.guilty\_id = car.id  
 GROUP BY car.people\_id),  
 fine\_new AS (SELECT driver\_license.people\_id as people\_id\_fine, *count*(fine.driver\_license) AS count\_fine  
 FROM fine  
 JOIN driver\_license ON fine.driver\_license = driver\_license.id  
 GROUP BY driver\_license.people\_id)  
SELECT dtp\_new.people, fine\_new.count\_fine/dtp\_new.count\_guilty AS attitude,  
 fine\_new.count\_fine, dtp\_new.count\_guilty  
FROM dtp\_new  
JOIN fine\_new ON dtp\_new.people = fine\_new.people\_id\_fine  
ORDER BY attitude DESC  
LIMIT 10



## **4.2. Вывести количество нарушений каждого вида для каждого года.**

Связываем внешние ключи таблицы fine и таблицы violation. Группируем id\_violation, title, punishment, year и сортируем по id\_violation, year. Выводим year год, title название штрафа, description описание штрафа, *count* количество нарушений.

SELECT *EXTRACT*(YEAR FROM data\_and\_time) as year,violation.title as title, violation.punishment as description, *count*(*EXTRACT*(YEAR FROM data\_and\_time))  
FROM fine  
JOIN violation ON fine.id\_violation = violation.id  
GROUP BY id\_violation, title, punishment, year  
ORDER BY id\_violation ASC, year

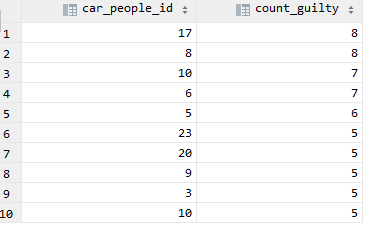


В качестве примера вывода запроса было приведено 25 из 550 строк.

## **4.3. Вывести 10 моделей автомобилей, владельцы которых чаще всего попадают в аварии.**

Связываем внешние ключи таблицы guilty с таблицами dtp и car. Группируем car.id и сортируем по count\_guilty. Выводим car\_people\_id идентификатор водителя, где он был виновным в ДТП и count\_guilty количество аварий водителя, где он был виновным в ДТП.

SELECT car.people\_id AS car\_people\_id, *count*(car.people\_id) AS count\_guilty  
FROM guilty  
JOIN dtp ON guilty.guilty = dtp.guilty  
JOIN car ON guilty.guilty\_id = car.id  
GROUP BY car.id  
ORDER BY count\_guilty DESC  
limit 10



# **5. Представления и хранимые процедуры**

Оформим стандартные запросы и запросы, заданные преподавателем, в виде представлений. Представления – виртуальные именованные таблицы, создаваемые с помощью запроса SELECT .

CREATE OR REPLACE VIEW select\_categories AS  
 SELECT *\** FROM car  
 WHERE id IN (31, 32, 33, 34)  
 AND categories = 1;  
  
CREATE OR REPLACE VIEW select\_Between AS  
 SELECT *\** FROM inspector  
 WHERE id BETWEEN 0 AND 20  
 AND people\_id < 10;  
  
CREATE OR REPLACE VIEW select\_Like AS  
 SELECT *\** FROM people  
 WHERE first\_name LIKE 'e%';  
  
CREATE OR REPLACE VIEW select\_Сalculate AS  
 SELECT id, driver\_license + 100000 AS driver\_license2,  
 police\_certificate - 100000 AS police\_certificate2  
 FROM fine  
 LIMIT 10;  
  
CREATE OR REPLACE VIEW select\_OrderBy AS  
 SELECT *\** FROM people  
 ORDER BY last\_name ASC, first\_name ASC  
 LIMIT 10;  
  
CREATE OR REPLACE VIEW select\_aggregate AS  
SELECT *COUNT*(*\**) AS count,  
 *AVG*(driver\_license) AS driver\_license\_max,  
 *MAX*(police\_certificate) AS police\_certificate\_max,  
 *MAX*(data\_and\_time) AS data\_and\_time\_max,  
 *MIN*(data\_and\_time) AS data\_and\_time\_min  
FROM fine;  
  
CREATE OR REPLACE VIEW select\_Join1 AS  
 SELECT fine.id, driver\_license, driver\_license.categories  
 FROM fine  
 JOIN driver\_license ON fine.id = driver\_license.id  
 LIMIT 10;  
  
CREATE OR REPLACE VIEW select\_group AS  
 SELECT driver\_license, *MAX*(id\_violation)  
 FROM fine  
 GROUP BY driver\_license  
 LIMIT 10;  
  
CREATE OR REPLACE VIEW select\_maxNumberOfFines AS  
 SELECT *MAX*(c.cnt) AS maxFine  
 FROM(SELECT *COUNT*(driver\_license) AS cnt  
 FROM fine  
 GROUP BY driver\_license) c;  
  
CREATE OR REPLACE VIEW select1\_attitude AS  
 WITH dtp\_new AS (SELECT car.people\_id AS people, *count*(car.people\_id) AS count\_guilty  
 FROM guilty  
 JOIN dtp ON guilty.guilty = dtp.guilty  
 JOIN car ON guilty.guilty\_id = car.id  
 GROUP BY car.people\_id),  
 fine\_new AS (SELECT driver\_license.people\_id as people\_id\_fine, *count*(fine.driver\_license) AS count\_fine  
 FROM fine  
 JOIN driver\_license ON fine.driver\_license = driver\_license.id  
 GROUP BY driver\_license.people\_id)  
 SELECT dtp\_new.people, fine\_new.count\_fine/dtp\_new.count\_guilty AS attitude,  
 fine\_new.count\_fine, dtp\_new.count\_guilty  
 FROM dtp\_new  
 JOIN fine\_new ON dtp\_new.people = fine\_new.people\_id\_fine  
 ORDER BY attitude DESC  
 LIMIT 10;  
  
CREATE OR REPLACE VIEW select2\_CountFineInYear AS  
 SELECT *EXTRACT*(YEAR FROM data\_and\_time) as year,violation.title as title, violation.punishment as description, *count*(*EXTRACT*(YEAR FROM data\_and\_time))  
 FROM fine  
 JOIN violation ON fine.id\_violation = violation.id  
 GROUP BY id\_violation, title, punishment, year  
 ORDER BY id\_violation ASC, year;  
  
CREATE OR REPLACE VIEW select3\_CarInMaxDTP AS  
 SELECT car.people\_id AS car\_people\_id, *count*(car.people\_id) AS count\_guilty  
 FROM guilty  
 JOIN dtp ON guilty.guilty = dtp.guilty  
 JOIN car ON guilty.guilty\_id = car.id  
 GROUP BY car.id  
 ORDER BY count\_guilty DESC  
 limit 10;

Оформим запросы изменения и удаления данных как хранимые процедуры.

Изменение данных.

drop procedure if exists *procedureUpdate*;  
create procedure *procedureUpdate*()  
as $$  
 begin  
 UPDATE fine  
 SET driver\_license = 72850  
 WHERE driver\_license = 72849;  
  
 SELECT *\**  
 FROM fine  
 WHERE driver\_license = 72849 or driver\_license = 72850;  
 end  
$$  
 language plpgsql;

Удаленние данных.

drop procedure if exists *procedureDelete*;  
create procedure *procedureDelete*()  
as $$  
 begin  
 DELETE  
 FROM fine  
 WHERE driver\_license = 72850;  
 end  
$$  
 language plpgsql;

**Вывод**

В процессе выполнения данной работы было изучены с условиями, вычисляемыми полями, совокупными характеристиками, с соединениями, с использованием группировки и сортировки; запросы вставки тестовых данных; запросы удаления данных по различным условиям; были поверхностно изучены процедуры и приобретён навык создания представлений запросов. Также с помощью заданий преподавателя был приобретён опыт написания сложных запросов.