Министерство науки и высшего образования РФ

Российский химико-технологический университет

имени Д. И. Менделеева

Кафедра информационных компьютерных технологий

Отчет по курсовой работе по дисциплине «Базы данных»

на тему:

«Прокат автомобилей внутренний учет.»

Исполнитель: гр. КС-30

Ф. И. О. студента: Терехова Оксана Игоревна

Ф. И. О. руководителя: Семенов Геннадий Николаевич

Москва, 2018г.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

ПО ДИСЦИПЛИНЕ «БАЗЫ ДАННЫХ

Студентке Тереховой Оксане Игоревне группы КС-30

Тема курсовой работы

11 Прокат автомобилей внутренний учет.

Автомобиль: идентификационный номер, марка, , компания-поставщик, число штук данного типа общее, в гараже, выдано в настоящее время, выдано всего, выдано в среднем за месяц, общий пробег; модель, длительность, мощность, год выпуска и производитель.

Клиенты: ФИО, паспортные данные, адрес, телефон; заказы, взятые автомобили сейчас и в прошлом: номер, дата выдачи, дата возвращения, общая стоимость заказа.

Этапы разработки работы

1. Инфологическое проектирование БД

2. Даталогическое проектирование БД для СУБД (Data Definition Language)

3. Наполнение и манипулирование данными в СУБД MySQL (Data Manipulation Language)

4. Запросы на выборку (Data Query Language)

|  |  |  |
| --- | --- | --- |
| Основные разделы | Удельный вес  раздела работы | Срок выполнения |
| Этап 1 |  |  |
| Этап 2 |  |  |
| Этап 3 |  |  |
| Этап 4 |  |  |
| Этап 5 |  |  |

Дата выдачи знания \_\_\_\_\_\_\_\_\_\_ 2018 г.

Дата сдача работы \_\_\_\_\_\_\_\_\_\_\_2018 г.

Дата защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

Руководитель работы

Оглавление

[1. Техническое задание 4](#_Toc533376941)

[2. Инфологическая модель 4](#_Toc533376942)

[3. Даталогическая модель 4](#_Toc533376943)

[3.1. Скрипты создания и заполнения 4](#_Toc533376944)

[3.1.1. Создание БД с именем auto 4](#_Toc533376945)

[3.1.2. Создание таблиц с помощью CREATE 4](#_Toc533376946)

[3.1.3. Вставка данных – INSERT 5](#_Toc533376947)

[3.2. Скриншоты таблиц 6](#_Toc533376948)

[3.2.1. Clients 6](#_Toc533376949)

[3.2.2. Companies 6](#_Toc533376950)

[3.2.3. Garage 6](#_Toc533376951)

[3.2.4. Cars 7](#_Toc533376952)

[3.2.5. Orders 7](#_Toc533376953)

[4. Результаты запросов 7](#_Toc533376954)

[4.1. Запросы 7](#_Toc533376955)

[4.2. Процедуры 9](#_Toc533376956)

[4.3. Представления 9](#_Toc533376957)

[4.4. Триггеры 11](#_Toc533376958)

[5. Заключение 12](#_Toc533376959)

[6. Литература 12](#_Toc533376960)

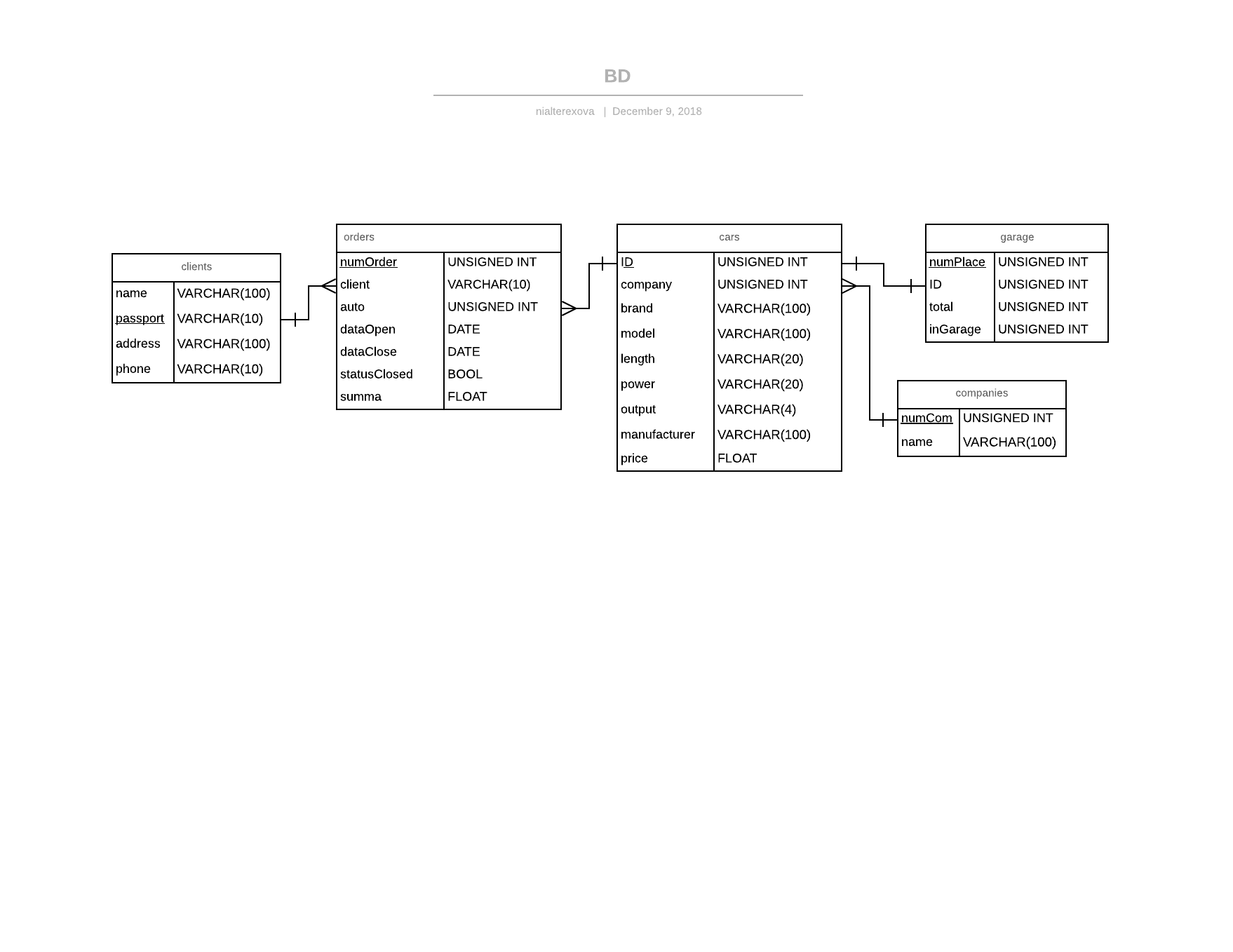
# Техническое задание

**Прокат автомобилей внутренний учет.**

**Автомобиль**: идентификационный номер, марка, , компания-поставщик, число штук данного типа общее, в гараже, выдано в настоящее время, выдано всего, выдано в среднем за месяц, общий пробег; модель, длительность, мощность, год выпуска и производитель.

**Клиенты**: ФИО, паспортные данные, адрес, телефон; заказы, взятые автомобили сейчас и в прошлом: номер, дата выдачи, дата возвращения, общая стоимость заказа.

# Инфологическая модель



# Даталогическая модель

## Скрипты создания и заполнения

### Создание БД с именем auto

CREATE DATABASE auto CHARACTER SET utf8 COLLATE utf8\_general\_ci;

### Создание таблиц с помощью CREATE

CREATE TABLE clients (

name\_cl varchar(100) NOT NULL ,

passport bigint(10) NOT NULL PRIMARY KEY,

address varchar(100) NOT NULL,

phone VARCHAR(10) NOT NULL

) DEFAULT CHARSET = utf8

CREATE TABLE garage (

num\_place int(10) unsigned NOT NULL PRIMARY KEY AUTO\_INCREMENT,

id int(10) unsigned NOT NULL REFERENCES cars(id),

total int(10) unsigned,

ingarage int(10) unsigned

) DEFAULT CHARSET = utf8

CREATE TABLE companies (

num\_com int(10) unsigned NOT NULL PRIMARY KEY AUTO\_INCREMENT,

name\_com varchar(100) NOT NULL

) DEFAULT CHARSET = utf8

CREATE TABLE cars (

id int(10) unsigned NOT NULL PRIMARY KEY AUTO\_INCREMENT,

company int(10) unsigned NOT NULL REFERENCES companies(num\_com),

brand varchar(100),

model varchar(100),

power int(5),

year\_out int(4),

manufacturer varchar(100),

price float unsigned

) DEFAULT CHARSET = utf8

CREATE TABLE orders (

numOrder int(10) unsigned NOT NULL PRIMARY KEY AUTO\_INCREMENT,

client bigint(10) unsigned NOT NULL REFERENCES clients(passport),

auto int(10) unsigned NOT NULL REFERENCES cars(id),

dataOpen date NOT NULL,

dataClose date NOT NULL,

statusClosed bit DEFAULT 0,

summa float unsigned

) DEFAULT CHARSET = utf8

### Вставка данных – INSERT

INSERT INTO clients (name, passport, address, phone) VALUES

( 'Иванов Вася', 4516928454, 'Москва, ул Панфилова, 12, 122', '9256139767'),

( 'Киркоров', 2348693576, 'Москва, Звездный переулок, 69, 696', '9666666666'),

( 'Дед Мороз', 1234567890, 'Полюс, город Великий Устюг, дом Деда Мороза', '9123456789'),

( 'Геральд Ведьмак', 4565438864, 'Долина Каэр Морхен, замок', '9252050505'),

( 'Сэм Уитвики', 5657465536, 'Tranquillity, 76', '9257657687'),

( 'Пол Уокер', 4564767876, 'Калифорния, 869', '9547658791')

INSERT INTO companies (name\_com) VALUES

( 'Автоцентр на перекрестке дорог'),

( 'Подделки из палок'),

( 'Конюшня "Три коня и Вася"')

INSERT INTO cars (company, brand, model, power, year\_out, manufacturer, price) VALUES

( 1, 'BMW', 'X7', 280, 2016, 'BMW official', 15000),

( 1, 'Лада', 'Калина', 100, 2007, 'Лада', 1500),

( 1, 'Лимузин', 'Золотой в блестках', 200, 2018, 'Королевская гавань', 1000000),

( 2, 'Сани', 'NewYear', 800, 1258, 'Cristmas New', 65000),

( 3, 'Лошадь', 'Плотва', 1, 2010, 'Мама и Папа', 500),

( 1, 'Chevrolet Camaro', 'Bibi', 500, 2013, 'Кибертрон', 34000),

( 1, 'Audi', 'A8', 300, 2016, 'Audi official', 14000)

INSERT INTO garage (id, total, ingarage) VALUES

( 1, 10, 10),

( 7, 10, 10),

( 2, 30, 30),

( 6, 1, 1),

( 3, 3, 3),

( 4, 2, 2),

( 5, 1, 1)

INSERT INTO orders (client, auto, dataOpen, dataClose, statusClosed, summa) VALUES

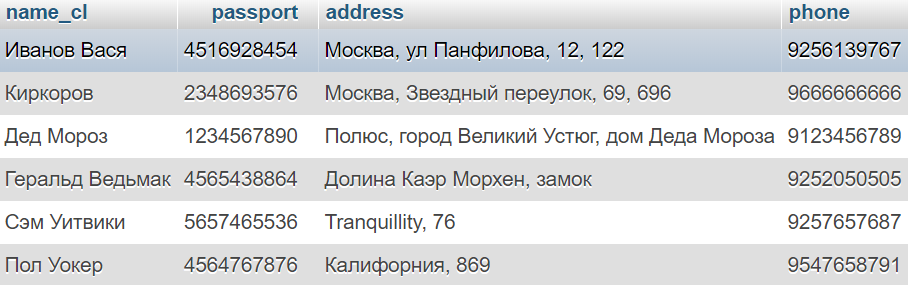
( 1234567890, 4, '2018-12-31','2019-01-02', 1, 130000),

( 4565438864, 5, '2018-05-01','2018-05-11', 1, 5000),

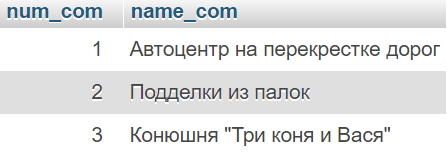
( 4516928454, 2, '2018-12-04','2018-12-06', 1, 3000)

## Скриншоты таблиц

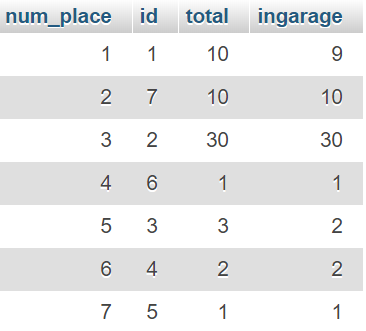
### Clients



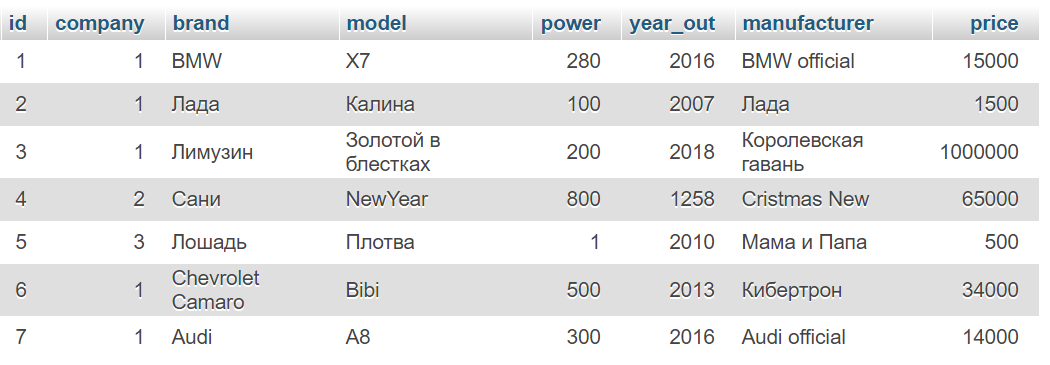
### Companies



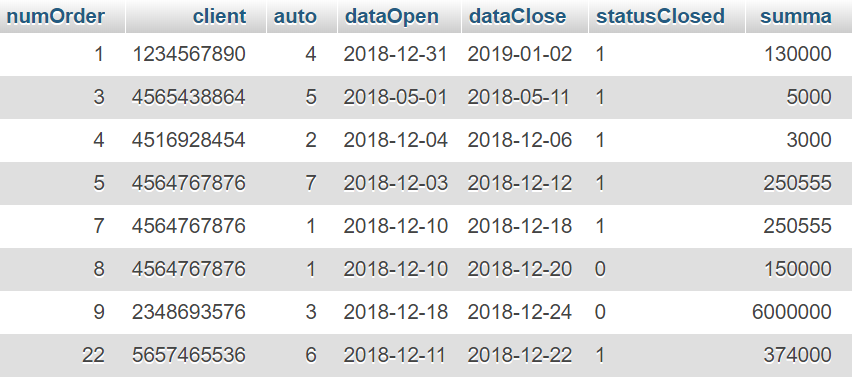
### Garage



### Cars



### Orders



# Результаты запросов

## Запросы

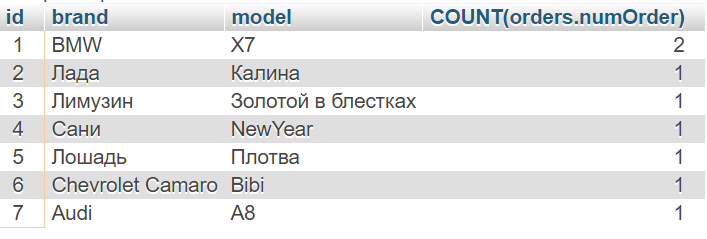
1. Общее количество выданных машин каждой модели за все время

SELECT cars.id, cars.brand, cars.model, COUNT(orders.numOrder)

FROM cars, orders

WHERE cars.id=orders.auto

GROUP BY cars.id



1. Выдано и заработано в среднем за последний месяц, группируя по каждому авто

SELECT avg(stat.col), AVG(stat.summ)

from (SELECT cars.brand, cars.model, COUNT(orders.numOrder)as col, SUM(orders.summa)as summ FROM cars, orders WHERE cars.id=orders.auto AND month(dataClose) = month(CURDATE()) AND year(dataOpen)=Year(CURDATE())

GROUP BY cars.model) as stat



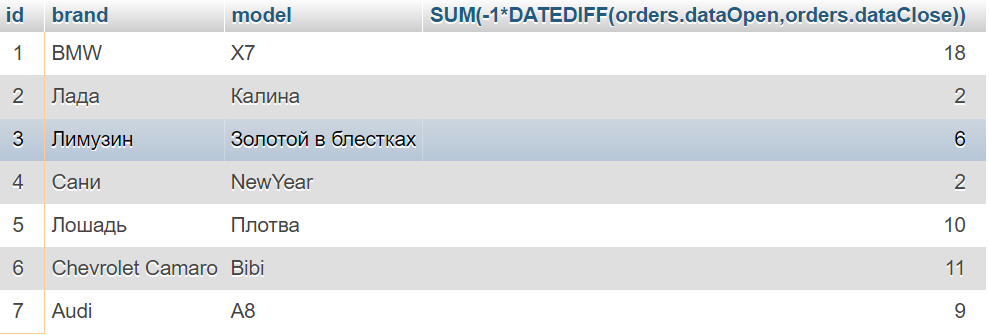
1. Общее количество дней проведенное на выдаче авто каждой модели

SELECT cars.id, cars.brand, cars.model, SUM(-1\*DATEDIFF(orders.dataOpen,orders.dataClose))

FROM cars, orders

WHERE cars.id=orders.auto

GROUP BY cars.id

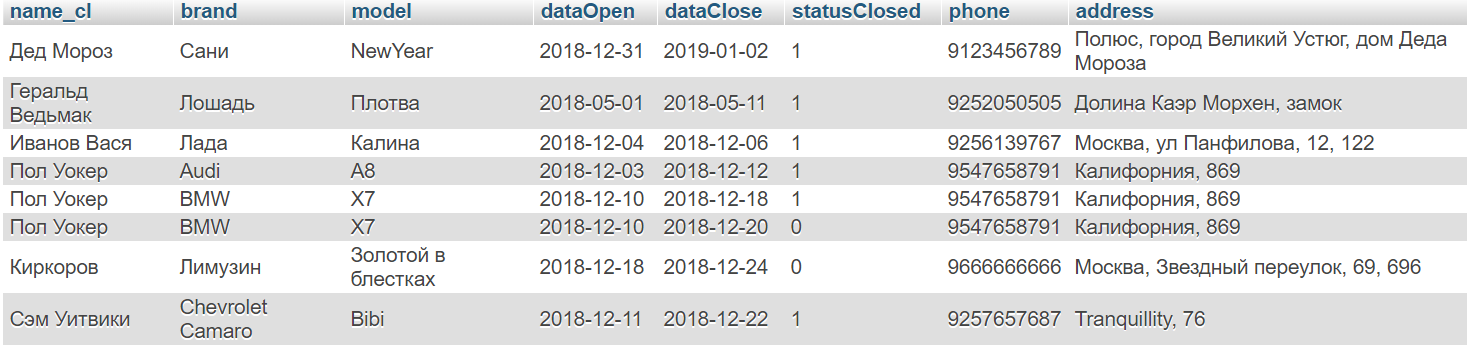


1. История заказов клиентов и информация о них

SELECT clients.name\_cl, cars.brand, cars.model, orders.dataOpen, orders.dataClose, orders.statusClosed, clients.phone, clients.address

FROM clients, cars, orders

WHERE clients.passport=orders.client AND orders.auto=cars.id



1. Самый дорогой заказ

SELECT clients.name\_cl, orders.summa

FROM clients, orders

WHERE clients.passport=orders.client

ORDER BY orders.summa DESC LIMIT 1



1. Рейтинг прибыльных клиентов (в сумме по заказам)

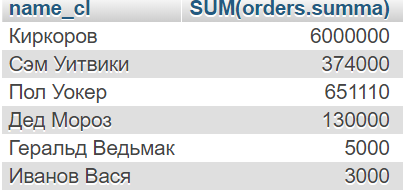
SELECT clients.name\_cl, SUM(orders.summa)

FROM clients, orders

WHERE clients.passport=orders.client

GROUP BY clients.name\_cl

ORDER BY orders.summa DESC



## Процедуры

1. Информация о клиенте и истории его заказов по номеру паспорта

DELIMITER //

CREATE PROCEDURE Client\_Info(n bigint(10))

BEGIN

SELECT clients.name\_cl, cars.brand, cars.model,orders.summa, orders.dataOpen, orders.dataClose, orders.statusClosed, clients.phone, clients.address

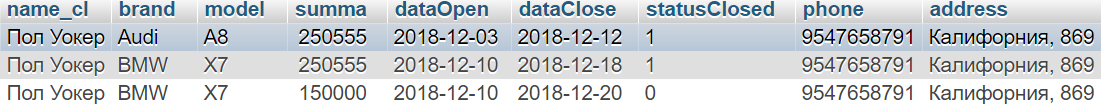
FROM clients, cars, orders

WHERE clients.passport=orders.client AND cars.id=orders.auto AND clients.passport=n;

END//

DELIMITER ;

CALL client\_Info(4564767876)



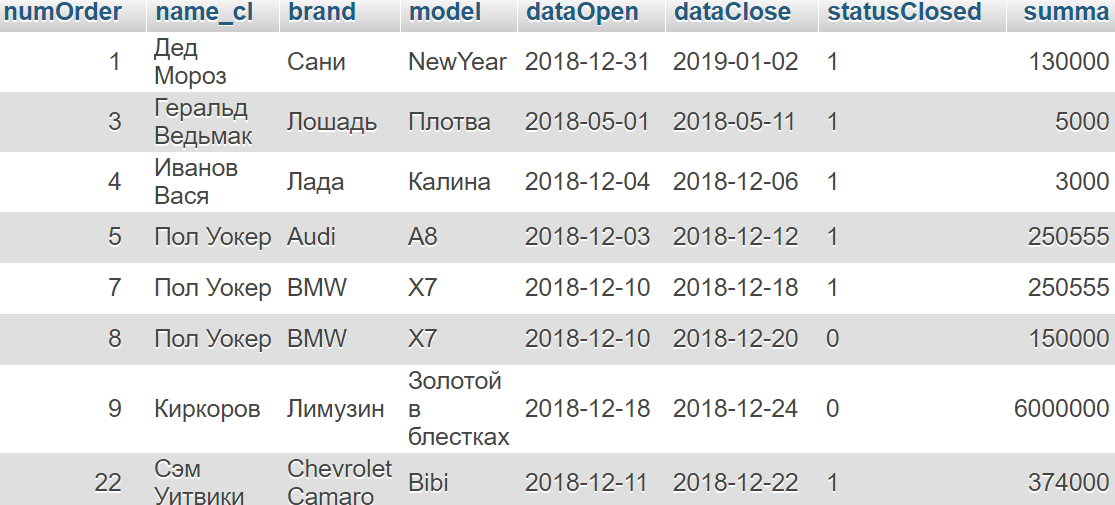
## Представления

CREATE VIEW orders\_all

AS SELECT orders.numOrder, clients.name\_cl, cars.brand, cars.model, orders.dataOpen, orders.dataClose, orders.statusClosed, orders.summa

FROM orders, clients, cars

WHERE orders.client=clients.passport AND orders.auto=cars.id



CREATE VIEW cars\_all

AS SELECT cars.id, companies.name\_com, cars.brand, cars.model, cars.power, cars.year\_out, cars.manufacturer, cars.price, garage.num\_place, garage.total, garage.total-garage.ingarage as issued\_now

FROM cars, companies, garage

WHERE garage.id=cars.id AND companies.num\_com=cars.company



## Триггеры

1. Триггер после добавления строки заказа проверяет статус заказа, если открыт, то изменяет количество авто в гараже

CREATE TRIGGER After\_Insert\_Order AFTER INSERT ON orders FOR EACH ROW

IF NEW.statusClosed=0 THEN

UPDATE garage

SET ingarage=ingarage-1

WHERE garage.id = NEW.auto;

END if;

1. Триггер следит за изменением статуса заказа и соответствующим образом изменяет количество авто в гараже

CREATE TRIGGER After\_Update\_Order AFTER UPDATE ON orders FOR EACH ROW

IF NEW.statusClosed=0 THEN

UPDATE garage

SET ingarage=ingarage-1

WHERE garage.id = NEW.auto;

ELSE

UPDATE garage

SET ingarage=ingarage+1

WHERE garage.id = NEW.auto;

END if;

1. Триггер возвращает авто в гараж, при удалении заказа

CREATE TRIGGER Before\_Delete\_Order BEFORE DELETE ON orders

FOR EACH ROW IF OLD.statusClosed=0 THEN

UPDATE garage

SET ingarage=ingarage+1

WHERE garage.id = OLD.auto;

END if

1. Триггер автоматически вычисляет сумму заказа, если она не была указана на этапе заполнения

CREATE TRIGGER Before\_Insert\_Order BEFORE INSERT ON orders

FOR EACH ROW IF NEW.summa IS NULL THEN

SET NEW.summa = -1 \* DATEDIFF(NEW.dataOpen,NEW.dataClose)\*(SELECT price FROM cars WHERE NEW.auto=cars.id);

END if

# Заключение

В процессе работы был освещен принцип действия реляционных баз данных, особенности их построения и структуры. Так же были затронуты основы структурированного языка запросов SQL на примере диалекта MySQL. Получено представление об основных конструкциях языка.

# Литература

«Руководство по MySQL»

Тахагхогхи С., Вильямс Х. Е., O'REILLY, 2007