**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Институт информационных технологий и управления**

**Кафедра компьютерных систем и программных технологий**

**ОТЧЕТ**

**по лабораторной работе №4**

**«Язык SQL-DML»**

**Информационное обеспечение систем управления**

Студент гр. 43501/1 Матлаш И.Г.

Преподаватель Мяснов А.В.

Санкт-Петербург

2015

# 1.Цель работы

Ознакомиться с языком создания запросов управления данными SQL-DML.

## Программа работы:

1. Изучить SQL-DML

2. Выполнить все запросы из списка стандартных запросов.

3. Реализовать SQL-запросы в соответствии с индивидуальным заданием.

## Индивидуальное задание:

БД хранит информацию об машинах, их характеристиках, состоянии, а также информацию о владельце, дате размещения объявления и действительно ли сейчас оно.

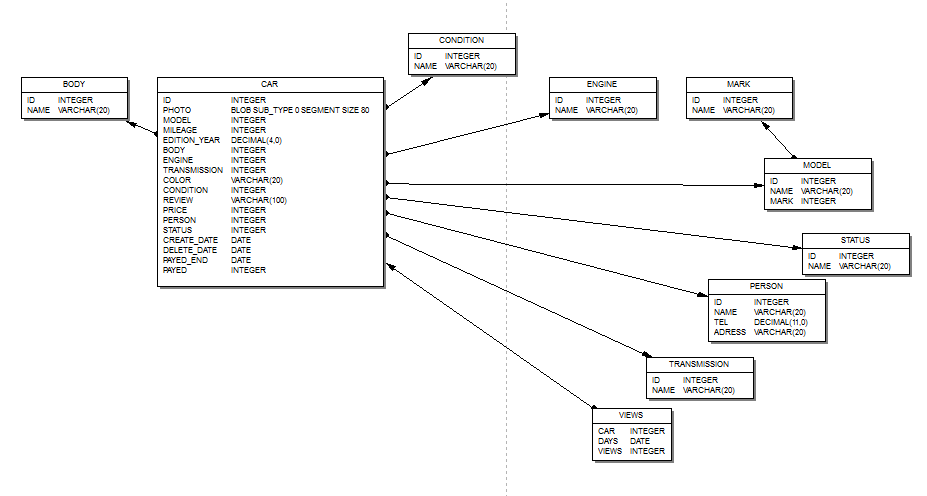


Рис 1. ER-диаграмма, построенная в IBExpert.

# 2. Стандартные запросы

## 1. Сделайте выборку всех данных из каждой таблицы

select \* from body;

ID NAME

1 SUV

2 Cabriolet

3 Sedan

4 Universal

select \* from condition;

ID NAME

1 new

2 good

3 average

4 bad

5 terrible

select \* from mark;

ID NAME

1 Mercedes-Benz

2 Porsche

3 Lada

select \* from person;

ID NAME TEL ADRESS

1 Ivan 89211234253 SPB,Nevski 11-32

2 Max 89114326789 SPB,2 line 23-9

3 Vitaliy 89219864365 SPB,Liteiniy 4-1

4 Andry 89219865423 SPB,Liteiniy 4-2

5 Ruslan 89219765365 SPB,Liteiniy 4-3

6 Igor 89219765332 SPB,3 line 2-7

7 Ruslan 89219654332 SPB,5 line 7-1

select \* from transmission;

ID NAME

1 Automatic

2 Manual

select \* from car;

ID PHOTO MODEL MILEAGE EDITION\_YEAR BODY ENGINE TRANSMISSION COLOR CONDITION REVIEW PRICE PERSON STATUS CREATE\_DATE DELETE\_DATE PAYED\_END PAYED

1 1 200000 2002 2 1 1 Pink 2 best car ever 14999 1 1 28.01.2014 29.12.2015 1

2 2 500000 2005 1 1 1 Red 4 best car maybe 25000 2 1 27.01.2014 03.03.2016 1

3 3 30000 2007 3 1 2 Green 3 awesome car 2000 3 2 16.01.2014 26.01.2014 0

4 3 1000 2008 3 1 2 Silver-Blue 5 not your car 1500 4 1 18.01.2014 0

5 1 220000 2003 2 1 1 Black 1 my car 18700 5 1 02.01.2014 14.06.2015 1

6 4 100 2014 3 1 2 Black 3 good car 8700 6 1 02.01.2014 0

7 5 25000 2014 3 1 2 Black-Grey 2 my good car 7700 7 1 03.01.2014 0

8 6 360000 2007 4 1 2 Black-Silver 4 1700 1 1 04.01.2014 0

9 6 80000 2003 4 1 2 Red 3 2400 2 1 05.01.2014 0

10 4 5000 2012 3 1 2 Yellow 2 6600 3 1 06.01.2014 0

11 4 56000 2011 3 1 2 Green 2 4300 4 1 07.01.2014 0

12 5 0 2013 3 1 2 Purple 1 5600 5 1 08.01.2014 0

select \* from engine;

ID NAME

1 Gasoline

2 Gas

select \* from model;

ID NAME MARK

1 911 Carrera 2

2 GL450 1

3 2110 3

4 Priora 3

5 Granta 3

6 2104 3

select \* from status;

ID NAME

1 on sale

2 saled

select \* from views;

CAR DAYS VIEWS

1 28.01.2014 3

2 28.01.2014 2

1 29.01.2014 1

2 29.01.2014 5

1 30.01.2014 4

3 30.01.2014 8

4 30.01.2014 2

5 30.01.2014 1

6 28.01.2014 3

7 28.01.2014 2

8 29.01.2014 1

9 29.01.2014 5

10 30.01.2014 4

11 30.01.2014 8

12 30.01.2014 2

12 28.01.2014 1

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

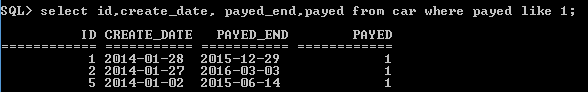
select id,create\_date, payed\_end,payed from car where payed like 1;

ID CREATE\_DATE PAYED\_END PAYED

1 28.01.2014 29.12.2015 1

2 27.01.2014 03.03.2016 1

5 02.01.2014 14.06.2015 1



select id,create\_date from car where create\_date between '2014-01-01' and '2014-01-19';

ID CREATE\_DATE

3 16.01.2014

4 18.01.2014

5 02.01.2014

6 02.01.2014

7 03.01.2014

8 04.01.2014

9 05.01.2014

10 06.01.2014

11 07.01.2014

12 08.01.2014

select id,name from person where name in ('Ivan','Ruslan');

ID NAME

1 Ivan

5 Ruslan

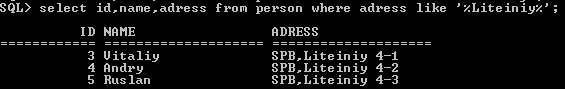
7 Ruslan

select id,name,adress from person where adress like '%Liteiniy%';

ID NAME ADRESS

3 Vitaliy SPB,Liteiniy 4-1

4 Andry SPB,Liteiniy 4-2

5 Ruslan SPB,Liteiniy 4-3

## 3. Создайте в запросе вычисляемое поле

select id,price,price\*65 as price\_rub from car;

ID PRICE PRICE\_RUB

1 14999 974935

2 25000 1625000

3 2000 130000

4 1500 97500

5 18700 1215500

6 8700 565500

7 7700 500500

8 1700 110500

9 2400 156000

10 6600 429000

11 4300 279500

12 5600 364000

## 4. Сделайте выборку всех данных с сортировкой по нескольким полям

select \* from person order by name,tel;

ID NAME TEL ADRESS

4 Andry 89219865423 SPB,Liteiniy 4-2

6 Igor 89219765332 SPB,3 line 2-7

1 Ivan 89211234253 SPB,Nevski 11-32

2 Max 89114326789 SPB,2 line 23-9

7 Ruslan 89219654332 SPB,5 line 7-1

5 Ruslan 89219765365 SPB,Liteiniy 4-3

3 Vitaliy 89219864365 SPB,Liteiniy 4-1

## 5. Создайте запрос, вычисляющий несколько совокупных характеристик таблиц

select avg(price),max(price) from car;

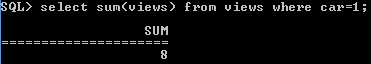
AVG MAX

8266 25000

select sum(views) from views where car=1;

SUM

8



## 6. Сделайте выборку данных из связанных таблиц (не менее двух примеров)

select person.name,model.name from person,model,car where car.person = person.id and car.model = model.id;

NAME NAME1

Ivan 911 Carrera

Ruslan 911 Carrera

Max GL450

Vitaliy 2110

Andry 2110

Igor Priora

Vitaliy Priora

Andry Priora

Ruslan Granta

Ruslan Granta

Ivan 2104

Max 2104

select mark.name,model.name from model,mark where mark.id = model.mark;

NAME NAME1

Mercedes-Benz GL450

Porsche 911 Carrera

Lada 2110

Lada Priora

Lada Granta

Lada 2104

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

select car,sum(views) from views group by car having sum(views)>5;

CAR SUM

1 8

2 7

3 8

11 8

## 8. Придумайте и реализуйте пример использования вложенного запроса

select car.id,car.person,car.model from car where car.model in (select model.id from model where model.name like '%911%');

ID PERSON MODEL

1 1 1

5 5 1

## 9. Оператор insert

Добавить в таблицы по одной записи

insert into mark values (4,'Fiat');

insert into body values (5,'Van');

insert into model values (7,'Ducato',5);

insert into person values (8,'Nikita','89215544332','SPB,7 line 8-3');

insert into car values (13,NULL,7,0,2014,5,1,2,'White',1,NULL,10400,8,1,

'2014-01-13',NULL,NULL,0);

insert into views values (12,'2014-01-28',4);

insert into views values (12,'2014-01-30',2);

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

select id,payed,payed\_end from car;

ID PAYED PAYED\_END

1 1 29.12.2015

2 1 03.03.2016

3 0

4 0

5 1 14.06.2015

6 0

7 0

8 0

9 0

10 0

11 0

12 0

13 0

update car set payed = 0, payed\_end = NULL where payed\_end<'2015-07-01';

1 record(s) was(were) updated in CAR

select id,payed,payed\_end from car;

ID PAYED PAYED\_END

1 1 29.12.2015

2 1 03.03.2016

3 0

4 0

5 0

6 0

7 0

8 0

9 0

10 0

11 0

12 0

13 0

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

delete from views where car =(select first 1 car from views group by car order by sum(views) desc);

## 12. С помощью оператора DELETE удалите записи в главной таблице, на которые не ссылается подчиненная таблица (используя вложенный запрос)

select \* from engine;

ID NAME

1 Gasoline

2 Gas

select id,name from engine where not exists(select \* from car where car.engine=engine.id);

ID NAME

2 Gas

delete from engine where not exists(select \* from car where car.engine=engine.id);

select \* from engine;

ID NAME

1 Gasoline

# 3. Индивидуальное задание

## Индивидуальные задания:

## 1. Вывести десять наиболее популярных объявлений за заданный период времени.

select first 10 car,sum(views) from views group by car order by sum(views) desc;

CAR SUM

1 8

3 8

11 8

2 7

12 6

9 5

10 4

6 3

4 2

7 2

## 2. Удалить неиспользуемые типы двигателей.

select \* from engine;

ID NAME

1 Gasoline

2 Gas

select id,name from engine where not exists(select \* from car where car.engine=engine.id);

ID NAME

2 Gas

delete from engine where not exists(select \* from car where car.engine=engine.id);

select \* from engine;

ID NAME

1 Gasoline

## 3. Вывести Список пользователей с количеством объявлений, поданных каждым.

select car.person,count(car.id) from car group by car.person;

PERSON COUNT

1 2

2 2

3 2

4 2

5 2

6 1

7 1

8 1

# 4. Вывод

В ходе работы были изучены необходимые для работы с реляционными базами данных запросы языка SQL-DML: insert, update, delete. Изучены внутреннее и внешнее объединение таблиц, группировки и агрегатные функции, вложенные запросы, а также представления. Полученные навыки работы с SQL-запросами будут использованы в дальнейшей работе с базой данных.DML запросы имеют довольно большие возможности для манипуляции базами данных.