Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Отчёт о лабораторной работе №4

Дисциплина: Базы данных

Тема: Язык SQL-DML

Выполнил студент гр. 43501/1 О.В. Горемыкина

Руководитель А.В. Мяснов

“ ” 2016 г.

Санкт -Петербург

2016

1. Цели работы

Познакомить студентов с языком создания запросов управления данными SQL-DML.

2. Программа работы

1. Изучите SQL-DML
2. Выполните все запросы из списка стандартных запросов. Продемонстрируйте результаты преподавателю.
3. Получите у преподавателя и реализуйте SQL-запросы в соответствии с индивидуальным заданием. Продемонстрируйте результаты
4. Выполненные запросы SELECT сохраните в БД в виде представлений, запросы INSERT, UPDATE или DELETE -- в виде ХП. Выложите скрипт в Subversion.

3. Ход работы

3.1. Язык SQL

Data Manipulation Language (DML) (язык управления (манипулирования) данными) — это семейство компьютерных языков, используемых в компьютерных программах или пользователями баз данныхдля получения, вставки, удаления или изменения данных в базах данных.

На текущий момент наиболее популярным языком DML является SQL, используемый для получения и манипулирования данными в РСУБД.

Функции языков DML определяются первым словом в предложении (часто называемом запросом), которое почти всегда является глаголом. В случае с SQL эти глаголы — «select» («выбрать»), «insert» («вставить»), «update» («обновить»), и «delete» («удалить»). Это превращает природу языка в ряд обязательных утверждений (команд) к базе данных.

3.2. Выполнение стандартных запросов

3.2.1. Выборка всех данных из каждой таблицы

```
CREATE VIEW v1 as SELECT * FROM ComplectationName;
CREATE VIEW v2 as SELECT * FROM Carcase;
CREATE VIEW v3 as SELECT * FROM EngineCapacity;
CREATE VIEW v4 as SELECT * FROM EnginePower;
CREATE VIEW v5 as SELECT * FROM Transmission;
CREATE VIEW v6 as SELECT * FROM Engine_Trans;
CREATE VIEW v7 as SELECT * FROM ModelType;
CREATE VIEW v8 as SELECT * FROM Complectation;
CREATE VIEW v9 as SELECT * FROM ModelComplectation;
CREATE VIEW v10 as SELECT * FROM Car_available;
CREATE VIEW v12 as SELECT * FROM Client;
CREATE VIEW v13 as SELECT * FROM Sale;
CREATE VIEW v14 as SELECT * FROM Warranty;
CREATE VIEW v15 as SELECT * FROM OptionList;
CREATE VIEW v16 as SELECT * FROM Comp_Option;
CREATE VIEW v17 as SELECT * FROM ExtraOption;
```

Выборка данных из таблицы ModelComplectation:

```
SELECT * FROM v9;
```

ID_MC	MODEL	ID_COMP	PRICE
1	Mazda3 hatchback	1	1259.0000
2	Mazda3 sedan	2	1249.0000
3	Mazda3 sedan	3	1344.0000
4	Mazda6 sedan	4	1304.0000
5	Mazda6 sedan	5	1355.0000
6	Mazda6 sedan	6	1576.0000
7	Mazda CX-5	7	1349.0000
8	Mazda CX-5	8	2012.0000
9	Mazda CX-9	9	2649.0000
10	Mazda BT-50	10	2345.0000
11	Mazda MX-5	11	2345.0000

3.2.2. Выборка данных из одной таблицы при нескольких условиях

Выведем названия и стоимость комплектаций Mazda3 sedan стоимостью от 1500 до 5000:

```
CREATE VIEW v25
    as SELECT model,price
    FROM ModelComplectation
    WHERE model in ('Mazda3 sedan', 'Mazda3 hatchback')
    AND price BETWEEN 1500 AND 5000;
```

MODEL	PRICE
Mazda3 sedan	2259.0000
Mazda3 sedan	2205.0000
Mazda3 sedan	1561.0000
Mazda3 hatchback	2859.0000
Mazda3 sedan	2857.0000
Mazda3 hatchback	2215.0000
Mazda3 hatchback	2595.0000
Mazda3 hatchback	1719.0000
Mazda3 hatchback	2644.0000

Выведем имена и контакты клиентов с именами на буквы «A» и «M»:

```
REATE VIEW v18
    as SELECT name,phone
    FROM Client
    WHERE name like 'M%'
    OR name like 'A%';
select * from v18;
```

NAME	PHONE
Alex Kolpakov	9112735684
Michael Pavlov	9112735684
Maria Kalugina	9112735684

3.2.3. Вычисляемое поле в запросе

Вычислим сумму всех заказов за 2015 год:

```
    SELECT SUM(totalPrice)
    FROM sale
    WHERE dat BETWEEN '01.01.2015' AND '31.12.2015';
```

SUM
89316.00000000000

3.2.4. Выборка всех данных с сортировкой по нескольким полям

Отсортируем типы комплектаций сначала по корпусам, затем по названиям комплектаций:

```
    CREATE VIEW v19  AS
    SELECT compName, carcaseType FROM Complectation ORDER BY carcaseType,compName;
    select * from v19;

COMPNAME                       CARCASETYPE
============================== ===========
supreme                        cabriolet
active                         crossover
drive                          crossover
supreme                        crossover
active                         hatchback
supreme                        picap
supreme                        picap
active                         sedan
active                         sedan
active                         sedan
comfort                        sedan
comfort                        sedan
drive                          sedan

COMPNAME                       CARCASETYPE
============================== ===========
drive                          sedan
drive                          sedan
exclusive                      sedan
exclusive                      sedan
gti                            sedan
gti                            sedan

COMPNAME                       CARCASETYPE
============================== ===========
gti                            sedan
gti                            sedan
supreme                        sedan
supreme                        sedan
```

3.2.5. Запрос, вычисляющий несколько совокупных характеристик таблиц

Найдем минимальную, максимальную и среднюю цены среди всех моделей и комплектаций:

```
CREATE VIEW v20  AS
    SELECT MIN(price) as minimum, AVG(price) as average, MAX(price) as maximum
    FROM ModelComplectation;
select * from v20;

       MINIMUM                AVERAGE        MAXIMUM
============== ====================== ==============
     1023.0000      1934.065573770492      2940.0000
```

3.2.6. Выборка данных из связанных таблиц

Выведем модели, стоимость и vin номера автомобилей в наличии в комплектации «supreme» и двигателем 3.2DT:

```
CREATE VIEW v21 AS
    SELECT ModelComplectation.model, ModelComplectation.price, Car_available.vin
    FROM ModelComplectation,Complectation,Car_available,Engine_Trans
    WHERE ModelComplectation.price > 2000
    AND Engine_Trans.transType = 'DT'
    AND Engine_Trans.capacity = 3.2
    AND ModelComplectation.id_comp = Complectation.id_comp
    AND Car_available.id_mc = ModelComplectation.id_mc
    AND Engine_Trans.id_engine_trans = Complectation.id_engine_trans;
select * from v21;
```

```
MODEL                                PRICE VIN
============================== ============== =================
Mazda MX-5                         2138.0000 80607382067083842
Mazda3 hatchback                   2853.0000 88935945329699469
Mazda CX-5                         2153.0000 86162207312007650
Mazda3 hatchback                   2387.0000 89529092415541287
```

Отобразим имена клиентов, стоимость их заказа и гарантийные случаи за последние 10 дней.

```
CREATE VIEW v22 AS
    SELECT DISTINCT Client.name, sale.totalPrice, Warranty.problem, sale.dat
    FROM Client,sale,Warranty
    WHERE sale.dat  BETWEEN '20.11.2016' AND '30.11.2016'
    AND sale.id_client = Client.id_client
    AND Warranty.id_sale = sale.id_sale;
select * from v22;


NAME                                               TOTALPRICE PROBLEM
================================================== ========== ========================
Heraclides                                          4857.0000 W*
Kailash                                             2939.0000 DmCM
Kailash                                             2939.0000 x!f|'s{v}P0Z5nm5oA1r
```

3.2.7. Запрос, рассчитывающий совокупную характеристику с использованием группировки

Сгруппируем модели автомобилей стоимостью не менее 1100:

```
CREATE VIEW v23(color, numb) AS
    SELECT model, MIN(price)
    FROM ModelComplectation GROUP BY model HAVING MIN(price) >1100;
select * from v23;

COLOR                                NUMB
============================== ==========
Mazda BT-50                        1179.0000
Mazda3 sedan                       1162.0000
Mazda6 sedan                       1161.0000
```

3.2.8. Использование вложенного запроса

Выберем все комплектации, в которых есть в наличии `'Mazda3 sedan:`

```
CREATE VIEW v24 AS
    SELECT compName
    FROM Complectation
    WHERE Complectation.id_comp IN
        (SELECT id_comp
        FROM ModelComplectation
        WHERE model = 'Mazda3 sedan');
select * from v24;

COMPNAME
================
gti
supreme
supreme
active
exclusive
comfort
exclusive
comfort
gti
active
exclusive
supreme
```

3.2.9. Добавление записей в таблицы

С помощью оператора INSERT добавим в таблицы по одной записи:

```
INSERT INTO ComplectationName VALUES ('sport');
INSERT INTO Carcase VALUES ('coupe');
INSERT INTO EngineCapacity VALUES (3.5);
INSERT INTO EnginePower VALUES (270);
INSERT INTO Complectation VALUES (101,'supreme','cabriolet',2.0,160,'MT');
INSERT INTO ModelComplectation VALUES (131,'Mazda MX-5',11,2345);
```

3.2.10. Измените значений нескольких полей у всех записей, отвечающих заданному условию

Поднимем стоимость всех Mazda CX-5 на 100:

```
SQL> select * from ModelComplectation where model = 'Mazda CX-5';
       ID MC MODEL                          ID COMP          PRICE
============ ============================== ============ ==============
           0 Mazda CX-5                             35       2741.0000
           2 Mazda CX-5                             20       1435.0000
          23 Mazda CX-5                             13       1663.0000
          30 Mazda CX-5                             32       2542.0000
          33 Mazda CX-5                             77       1675.0000
          40 Mazda CX-5                              8       2296.0000
          53 Mazda CX-5                             88       1002.0000
          77 Mazda CX-5                             97       2153.0000

SQL> UPDATE ModelComplectation SET price = price + 100 WHERE model = 'Mazda CX-5';
SQL> select * from ModelComplectation where model = 'Mazda CX-5';

       ID_MC MODEL                          ID_COMP          PRICE
============ ============================== ============ ==============
           0 Mazda CX-5                             35       2841.0000
           2 Mazda CX-5                             20       1535.0000
          23 Mazda CX-5                             13       1763.0000
          30 Mazda CX-5                             32       2642.0000
          33 Mazda CX-5                             77       1775.0000
          40 Mazda CX-5                              8       2396.0000
          53 Mazda CX-5                             88       1102.0000
          77 Mazda CX-5                             97       2253.0000
```

3.2.11. Удаление записи, имеющей максимальное (минимальное) значение некоторой совокупной характеристики

Удалим максимальную мощность двигателя:

```
SQL> select * from EnginePower;
 ENGINEPOWER
============
         104
         120
         150
         160
         175
         192
         200
         250
         270

SQL> DELETE FROM EnginePower WHERE enginePower = (select max(enginePower) from EnginePower);
SQL> select * from EnginePower;
 ENGINEPOWER
============
         104
         120
         150
         160
         175
         192
         200
         250
```

3.2.12. Удаление записи в главной таблице, на которые не ссылается подчиненная таблица

Удалим из списка неиспользуемый кузов:

```
DELETE FROM Carcase WHERE carcaseType NOT IN
    (SELECT carcaseType FROM Complectation);
```

```
CARCASETYPE
===========
sedan
crossover
hatchback
picap
wagon
cabriolet
coupe
```

```
SQL> DELETE FROM Carcase WHERE carcaseType NOT IN
CON>     (SELECT carcaseType FROM Complectation);
SQL> select * from Carcase;
```

```
CARCASETYPE
===========
sedan
crossover
hatchback
picap
wagon
cabriolet
```

Был удален тип кузова coupe.

3.3. Индивидуальное задание

3.3.1. Вывести количество заказанных дополнительных опций по месяцам заданного года

```
--  Вывести количество заказанных дополнительных опций по месяцам заданного года
CREATE VIEW year_options AS
    SELECT  EXTRACT(Year FROM dat) AS Y,
            EXTRACT(Month FROM dat) AS M,
            COUNT(2) AS ordered_options
    FROM sale
    JOIN optionlist USING (id_sale)
    WHERE dat BETWEEN '01.01.2015' AND '31.12.2015'
    GROUP BY 1,2;
select * from year_options;
```

```
      Y       M ORDERED_OPTIONS
======= ======= ================
   2015       3                1
   2015       6                2
   2015       7                2
   2015      10                1
   2015      11                1
   2015      12                5
```

3.3.2. Вывести 5 наиболее популярных двигателей за заданный период

```
CREATE VIEW top5engine_id AS
    SELECT first 5  id_engine_trans as id_engine, count(id_engine_trans) as count_engines
    from Engine_Trans
    join complectation using (id_engine_trans)
    join modelcomplectation using (id_comp)
    join car_available using (id_mc)
    join sale using (id_car)
    GROUP BY id_engine_trans
    ORDER BY count_engines DESC;
CREATE VIEW top5engine AS
    SELECT id_engine, capacity,transType,enginePower,  count_engines
    FROM top5engine_id, Engine_Trans
    WHERE top5engine_id.id_engine = Engine_Trans.id_engine_trans;
select * from top5engine;
```

ID_ENGINE	CAPACITY	TRANSTYPE	ENGINEPOWER	COUNT_ENGINES
94	2.5	AT	200	19
46	1.6	MT	175	12
93	1.5	MT	250	8
39	3.2	DT	160	7
61	3.2	DT	120	7

3.3.3. Вывести 10 клиентов, которые совершили повторный заказ на большую сумму

```
--  Вывести 10 клиентов, которые совершили повторный заказ на большую сумму
CREATE VIEW top10idclients AS
    SELECT s1.id_client, s1.totalPrice, s1.dat
    FROM sale as s1, sale as s2
    WHERE s1.id_client = s2.id_client
    AND s1.totalPrice>s2.totalPrice
    AND s1.dat > s2.dat
    ORDER BY s1.id_client, s1.dat;
--select * from top10idclients;

CREATE VIEW top10clients AS
    SELECT FIRST 10 c.id_client, c.name
    FROM top10idclients as top, client as c
    WHERE top.id_client = c.id_client
    GROUP BY c.id_client,c.name;
select * from top10clients;
```

ID_CLIENT	NAME
1	Rigoberto
2	Meiryan
5	Augusto
7	Slawomir
10	Buenaventura
11	Boulus
12	Iris
13	Berenice
14	Martin
16	Eloy

Проверим правильность выборки:

```
SQL> select id_client, totalprice,dat from sale order by id_client,dat;
```

ID_CLIENT	TOTALPRICE	DAT
0	2151.0000	2008-12-14
1	8552.0000	2009-12-15
1	9645.0000	2010-09-18
2	5465.0000	2013-02-09
2	7964.0000	2013-05-14
2	5473.0000	2016-09-11
3	4813.0000	2012-07-31
3	2991.0000	2016-03-22
5	5665.0000	2008-09-08
5	1103.0000	2008-11-01
5	7544.0000	2013-01-10

```
             6      6238.0000 2009-04-19
             6      2174.0000 2011-11-11
             7      5315.0000 2010-07-28
             7      9926.0000 2011-04-06
             8      4552.0000 2015-03-16
             8      1053.0000 2015-12-31
             9      1084.0000 2012-06-02
            10      1153.0000 2008-10-15
            10      7029.0000 2010-07-27
            11      8674.0000 2010-08-11
            11      6062.0000 2012-09-28
            11      7377.0000 2015-12-01
            12      5142.0000 2009-11-12
            12      9012.0000 2013-12-01
            12      7222.0000 2013-12-26
            13      1346.0000 2011-10-10
            13      7638.0000 2014-06-07
            14      8734.0000 2009-06-10
            14      9246.0000 2015-05-03
            14      8424.0000 2016-03-05
            15      1364.0000 2014-10-10
            16      1373.0000 2009-11-26
            16      2536.0000 2012-02-23
            16      3209.0000 2012-08-08
...
```

3.4. Сохранение выполненных запросов в виде хранимых процедур

3.4.1. Insert

```
SET TERM ^ ;
CREATE PROCEDURE insert_value ( n VARCHAR(10))
AS
BEGIN
    INSERT INTO Carcase VALUES( :n );
END
^
SET TERM ; ^
```

3.4.2. Update

```
CREATE PROCEDURE up price(model VARCHAR(30))
AS BEGIN
    UPDATE ModelComplectation
    SET price = price + 100
    WHERE model = :model;
END
^
```

```
SQL> select * from ModelComplectation where model = 'Mazda CX-5';

       ID_MC MODEL                           ID_COMP           PRICE
============ ============================== ============ ==============
           0 Mazda CX-5                             35      2741.0000
           2 Mazda CX-5                             20      1435.0000
          23 Mazda CX-5                             13      1663.0000
          30 Mazda CX-5                             32      2542.0000
          33 Mazda CX-5                             77      1675.0000

SQL> execute procedure up_price('Mazda CX-5 ');
SQL> select * from ModelComplectation where model = 'Mazda CX-5';

       ID_MC MODEL                           ID_COMP           PRICE
============ ============================== ============ ==============
           0 Mazda CX-5                             35      2841.0000
           2 Mazda CX-5                             20      1535.0000
          23 Mazda CX-5                             13      1763.0000
          30 Mazda CX-5                             32      2642.0000
          33 Mazda CX-5                             77      1775.0000
```

3.4.3. Delete

```
CREATE PROCEDURE rem_EnginePower(enginePower INT)
AS BEGIN
    DELETE FROM EnginePower
    WHERE enginePower =
        (select max(:enginePower)
        from EnginePower);
END
^

CREATE PROCEDURE del_unused_Carcase
AS BEGIN
    DELETE FROM Carcase
    WHERE carcaseType
    NOT IN
        (SELECT carcaseType
        FROM Complectation);
END
^
```

4. Выводы

Язык DML позволяет достаточно просто выполнять простейшие запросы на выборку данных, группировку и вычисление совокупных характеристик. Однако при написании более сложных запросов у неопытного пользователя может возникнуть ряд трудностей.

Поскольку представленная база данных имеет сложную нормализованную структуру, проводить операции добавления, удаления и редактирования данных вручную слишком трудоемко. Поэтому были реализованы хранимые процедуры, упрощающие эти действия.