

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Отчёт о лабораторной работе №3

Дисциплина: Базы данных

Тема: Язык SQL-DML

Выполнил студент гр. 43501/1

Нагорнов А.А.

Руководитель

Мяснов А.В.

“ ” _____ 2016 г.

Санкт -Петербург

2016

1. Цели работы

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

2. Программа работы

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

3. Ход работы

DML — это семейство компьютерных языков, используемых в компьютерных программах или пользователями баз данных для получения, вставки, удаления или изменения данных в базах данных.

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»).

3.1. Выполнение стандартных запросов

- Сделайте выборку всех данных из каждой таблицы

```
select * from albums;
select * from album_track;
select * from groups;
select * from group_members;
select * from people;
select * from styles;
select * from tracks;
select * from album_artist;
select * from authors;
select * from group_accounting;
select * from languages;
select * from performers;
select * from style_track;
```

```
SQL> select * from styles;
```

	ID	STYLE
1		Punk
2		Classic
3		Rap

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

Операторы `IN` (равен любому из списка) и `NOT IN` (не равен ни одному из списка) используются для сравнения проверяемого значения поля с заданным списком. Этот список значений указывается в скобках справа от оператора `IN`.

```
select * from styles where ID in ( 2 );
```

```
SQL> select * from styles where ID in ( 2 );

ID STYLE
=====
2 Classic
```

Оператор `BETWEEN` используется для проверки условия вхождения значения поля в заданный интервал, то есть вместо списка значений атрибута этот оператор задает границы его изменения.

```
select * from people where ID between 1 and 4;
```

```
SQL> select * from people where ID between 1 and 4;

ID NAME
=====
1 Shnurov
2 Tsoi
3 Pistoletov
4 Yuri Kasparyan
```

Оператор `LIKE` применим только к символьным полям типа `CHAR` или `VARCHAR`

```
select * from languages where language like 'Ru%n';
```

```
SQL> select * from languages where language like 'Ru%n';

LANGUAGE
=====
Russian
Ru"wgBb2X6.n
```

- Создайте в запросе вычисляемое поле

```
select sum(ID) from group_members;
```

```
SQL> select sum(ID) from group_members;

SUM
=====
3
```

- Сделайте выборку всех данных с сортировкой по нескольким полям

```
SELECT ID, language FROM tracks where ID between 1 and 20 ORDER BY language, id;
```

```
SQL> SELECT ID, language FROM tracks where ID between 1 and 20 ORDER BY language, id;
```

ID	LANGUAGE
4	English
6	English
10	English
17	English
18	English
20	English
7	Latvian
9	Latvian
15	Latvian
16	Latvian
1	Russian
2	Russian
3	Russian
5	Russian
8	Russian
11	Russian
12	Russian
13	Russian
14	Russian
19	Russian

- Создайте запрос, вычисляющий несколько совокупных характеристик таблиц

Найдем минимальный и максимальный ID среди всех людей:

```
select min(ID) as Minimum, max(ID) as Maximum from people;
```

```
SQL> select min(ID) as Minimum, max(ID) as Maximum from people;
```

MINIMUM	MAXIMUM
1	100004

- Сделайте выборку данных из связанных таблиц

Выведем название трека и ID его стилей:

```
select tracks.name, style_track.style_id
from tracks, style_track
where tracks.ID = style_track.track_id;
```

```
SQL> insert into style_track values(2,2,1);
SQL> select tracks.name, style_track.style_id
CON> from tracks, style_track
CON> where tracks.ID = style_track.track_id;
```

NAME	STYLE_ID
War	1
War	2

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

Сгруппируем название трека и его ID, который меньше 20:

```
select name, min(id) as MIN_ID
from tracks group by name having min(id)<20;
```

NAME	MIN_ID
*njI:>p5Z#~J31A>(mpZNp-Y1;_}J,t8&Vqi;n	15
-\\J]zmDN8aXvdB'3CIuxqj5Xo4K/CWH>%4'Bi9[LQ%LRxJvi	18
0{7W3u{vQ?'	19
10;BM/6-A0jKR:l nNb*.I12>'F"kol]Phb79ULJ,%mK..Co#	11
Cuckoo	3
New russian pirate	2
PBk:'6=w5DQ}%l@L-HQ1wN[l+JnM@du}Or	8
War	1
XASagF1ob8&=sF]P8e:L7: K(BYn	16
YaH ~BxAUYR2Y\$}{5In?DQ= (b,<FmfN6'Ei /2a`&Vv	17
\\a!8* Ng2w+)m[z5Y]"{uc{xLCj:9'	6
\\dyhy:JDU/=RyGhDn~;DxGs+pf/I@RQMGYq_dSCb1	13
'>y"yvlgt>{j%9DNfR~aN?lPxj_x8}o	7
gM2g}C1C8(\\PoL2	10
i&YB+xNiqMORN.nrEU	5
lt;>Mt9D;aI@MnCPbBaTfo.~,nYF)!^N)P&3xYG:*S>u	12
q{U! ~Y^wb@!fKW_7^L1,~mM,BJ	9
rq;X&;Yz ^2pG,rx\\00Y(6_^0:xR□cUUmK5	4
} N[3<'>' \\))630prwLtmxNI=c`^/T0;I#+ki@PZe)K(0	14

Оператор SQL HAVING является указателем на результат выполнения агрегатных функций. Агрегатной функцией в языке SQL называется функция, возвращающая какое-либо одно значение по набору значений столбца.

Таковыми функциями

являются: SQL COUNT(), SQL MIN(), SQL MAX(), SQL AVG(), SQL SUM().

- Придумайте и реализуйте пример использования вложенного запроса

Выберем те треки, название стилей которых начинается с буквы 'P':

```
select name from tracks
where tracks.ID in
(select track_id
from style_track
where style_track.style_id in
(select ID
from styles
where styles.style like ('P%')) );
```

```
SQL> select name from tracks
CON> where tracks.ID in
CON> (select track_id
CON> from style_track
CON> where style_track.style_id in
CON> (select ID
CON> from styles
CON> where styles.style like ('P%')) );
```

NAME

=====

War

- С помощью оператора `INSERT` добавьте в каждую таблицу по одной записи

```
insert into Languages values ('Ukrainian');
insert into tracks values (100004,' We Are Waiting for Change', 'Russian');
```

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

```
select * from tracks where id between 1 and 5;
update tracks set language='Latvian' where id in (4);
select * from tracks where id between 1 and 5;
```

```
SQL> select * from tracks where id between 1 and 5;
```

ID	NAME	LANGUAGE
1	War	Russian
2	New russian pirate	Russian
3	Cuckoo	Russian
4	rq;X&;Yz ^2pG,rx\00Y(6_^0:xR□cUUmKs	English
5	i&YB+xNiqMORN.nrEU	Russian

```
SQL> update tracks set language='Latvian' where id in (4);
SQL> select * from tracks where id between 1 and 5;
```

ID	NAME	LANGUAGE
1	War	Russian
2	New russian pirate	Russian
3	Cuckoo	Russian
4	rq;X&;Yz ^2pG,rx\00Y(6_^0:xR□cUUmKs	Latvian
5	i&YB+xNiqMORN.nrEU	Russian

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

```
select * from people where id>1000000;
delete from people where id = (select max(id) from people);
select * from people where id>1000000;
```

```
SQL> select * from people where id>1000000;
```

ID	NAME
100001	u_bw^QDaTqt1
100002	qaY^uHea]10JkhfaydAj\m^iBJ0mGymetqh
100003	OqYB1B[qtC0sWJySQkejruLBxHtZGlC
100004	bfPMtdwLHE\s\g

```
SQL> delete from people where id = (select max(id) from people);
SQL> select * from people where id>1000000;
```

ID	NAME
100001	u_bw^QDaTqt1
100002	qaY^uHea]10JkhfaydAj\m^iBJ0mGymetqh
100003	OqYB1B[qtC0sWJySQkejruLBxHtZGlC

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

Удалим из таблицы треков запись у которой в качестве языка песни указан Ukrainian:

```
insert into tracks values (100005,' We', 'Ukrainian');

select * from tracks where id>100000;

delete from tracks where language in
(select language from languages where language like('Ukr%'));
select * from languages where language like('Ukr%');

select * from tracks where id>100000;
```

```
SQL> select * from tracks where id>100000;
```

ID	NAME	LANGUAGE
100005	We	Ukrainian
100001	Kd"jc-G7ok"\$clq0m3T*Z,&q^sf4B("p	Russian
100002	tE)0'j.i2C\h"^]"eMH:JTSsJ0x~!Y7/	English
100003	"?GoqL;b7n?	English
100004	We Are Waiting for Change	Russian

```
SQL> delete from tracks where language in
CON> (select language from languages where language like('Ukr%'));
SQL> select * from tracks where id>100000;
```

ID	NAME	LANGUAGE
100001	Kd"jc-G7ok"\$clq0m3T*Z,&q^sf4B("p	Russian
100002	tE)0'j.i2C\h"^]"eMH:JTSsJ0x~!Y7/	English
100003	"?GoqL;b7n?	English
100004	We Are Waiting for Change	Russian

3.2. Индивидуальные задания

Необходимо:

1. Вывести 5 людей, которые одновременно были авторами и исполнителями наибольшего количества композиций.

```
create or alter view topauthors as
select first 5 artist_id as artist, count(track_id) as cnt
from authors,performers
where authors.track_id = performers.id_track
group by artist
order by cnt desc;
```

```
SQL> create or alter view topauthors as
CON> select first 5 artist_id as artist, count(track_id) as cnt
CON> from authors,performers
CON> where authors.track_id = performers.id_track
CON> group by artist
CON> order by cnt desc;
SQL>
SQL> select * from topauthors;
```

ARTIST	CNT
1	4
2	3
14	3
24	3
25	3

2. Вывести 10 людей, которые поучаствовали в записи наибольшего количества композиций.

```
select first 10 name, count(id_ispolnit) as cnt
from people join performers
on people.id = performers.id_ispolnit
group by name, id_ispolnit
order by cnt desc;
```

```
SQL> select first 10 name, count(id_ispolnit) as cnt
CON> from people join performers
CON> on people.id = performers.id_ispolnit
CON> group by name, id_ispolnit
CON> order by cnt desc;

NAME                                     CNT
=====
Shnurov                                 4
Tsoi                                    3
Timati                                  3
Butusov                                 3
Visockij                                3
Lolita                                  2
Bjanka                                  2
Agutin                                  2
Laima Vajkule                           2
Filip Kirkorov                          2
```

3. Вывести группы, все альбомы которых состоят из более чем 10 композиций.

```
create or alter view topalbums as
select album_id, count(track_id) as cnt
from album_track
group by album_id
order by cnt desc;

create view topgroups as
select group_id from group_members
where artist_id in
(
select people_id from album_artist, topalbums
where album_artist.album_id = topalbums.album_id
and topalbums.cnt > 10
)
group by group_id;
```

```
SQL> create or alter view topalbums as
CON> select album_id, count(track_id) as cnt
CON> from album_track
CON> group by album_id
CON> order by cnt desc;
SQL>
SQL> create view topgroups as
CON> select group_id from group_members
CON> where artist_id in
CON> (
CON> select people_id from album_artist, topalbums
CON> where album_artist.album_id = topalbums.album_id
CON> and topalbums.cnt > 10
CON> )
CON> group by group_id;
SQL>
SQL> select * from topalbums;

- ALBUM_ID      CNT
-----
2              11
3              11
1               3
4               1

SQL> select * from topgroups;

- GROUP_ID
-----
2
3
```


3.3. Сохранение выполненных запросов в виде хранимых процедур

Insert

```
SET TERM ! ;  
CREATE PROCEDURE insert_value ( i INT, n VARCHAR(10))  
AS  
BEGIN  
    INSERT INTO groups (id, name) VALUES( :i, :n );  
END  
!  
EXECUTE PROCEDURE insert_value (5, 'azaza')!
```

```
SQL>  
SQL> EXECUTE PROCEDURE insert_value (5, 'azaza')!  
SQL> select *from groups!  
-  
      ID NAME  
-----  
1 Kino  
2 Leningrad  
3 Black Star Mafia  
4 U-Piter  
5 azaza
```

Update

```
CREATE PROCEDURE gr_ch(id_ch int, name_ch VARCHAR(50))  
AS BEGIN  
    UPDATE groups  
    SET name = :name_ch  
    WHERE id = :id_ch;  
END  
!  
EXECUTE PROCEDURE gr_ch (5, 'New name')!
```

```
SQL>  
SQL> EXECUTE PROCEDURE gr_ch (5, 'New name')!  
SQL> select * from groups!  
-  
      ID NAME  
-----  
1 Kino  
2 Leningrad  
3 Black Star Mafia  
4 U-Piter  
5 New name
```

Delete

```
CREATE PROCEDURE rm_gr(id_r INT)
AS BEGIN
    DELETE FROM groups
    WHERE id = :id_r;
END
!
Set term ; !
EXECUTE PROCEDURE rm_gr (5);
```

```
SQL> EXECUTE PROCEDURE rm_gr (5);
SQL> select * from groups;

+-----+-----+
      ID NAME
+-----+-----+
      1 Kino
      2 Leningrad
      3 Black Star Mafia
      4 U-Piter
```

4. Выводы

С помощью языка DML можно достаточно просто выполнять простейшие запросы на модификацию записей и выборку данных, а также группировку и вычисление совокупных характеристик. Хранимые процедуры позволяют сохранить часто используемые однотипные операции.