

Отчёт по лабораторной работе

Дисциплина: Базы данных

Тема: SQL-DML

Выполнил студент гр. 43501/1

(подпись) А.О. Перешеин

Руководитель

(подпись) А.В. Мяснов

“ ____ ” _____ 2014 г.

Санкт-Петербург

2014

1. Цели работы

Знакомство 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 преподавателю.
4. Выполненные запросы SELECT сохраните в БД в виде представлений, запросы INSERT, UPDATE или DELETE - в виде ХП. Выложите скрипт в Subversion.

3. Стандартные запросы

3.1. Сделайте выборку всех данных из каждой таблицы

```
SQL> select * from artists;
```

ART_ID	ARTIST
1	Pantera
2	Amorphis
3	Slayer
4	Ice Cube

```
SQL> select * from art_alb;
```

ART_ID	ALB_ID
1	1
2	2
2	3
3	4
4	5

```
SQL> select * from cds_alb;
```

CODE	ALB_ID
679137227	1
781676650	2
781676650	3
329751487	4
105520746	5

```
SQL> select * from cds;
```

CODE	RELEASE_YEAR	SOLD	STATUS
679137227	1990	100000000	3
781676650	2000	100000000	3
329751487	1986	200000000	3
105520746	2013	170000000	3

SQL> select * from albums;

ALB_ID	ALBUM	RELEASE_YEAR	LABEL	RATING	NUM_OF_RAT	ST_ID	TYPE	GENRE
1	Cowboys from Hell	1990	Atco	5	54	1	album	1
2	Tales From The Thousand Lakes	1994	Relapse	4	37	2	album	1
3	Black Winter Day	1994	Relapse	4	23	2	EP	1
4	Reign In Blood	1986	Def Jam Recordings	5	78	3	album	1
5	Icon	2013	Priority Records	3	48	4	compilation	2

SQL> select * from studios;

ST_ID	TITLE	COUNTRY
1	Pantego Sound Studios	USA
2	Sunlight Studios	Sweden
3	Hit City West	USA
4	Death Row Studio	USA

SQL> select * from playlists;

NAME	DATE_OF_CR
list1	2013-01-09
list3	2014-09-30
mp3	2012-05-13
list2	2014-03-21
home	2013-01-30

SQL> select * from pl_tr;

NAME	TR_ID
home	21
home	22
home	23
home	24
home	25
home	26
home	27
home	32
home	33
home	34
list1	1
list1	2
list2	3
list2	4
list2	5
mp3	35
mp3	36
mp3	37
mp3	38
mp3	50
NAME	TR_ID
mp3	51
mp3	52

SQL> select * from tracks;

TR_ID	TITLE	DURATION
-------	-------	----------

1	Cowboys From Hell	00:04:07.0000
2	Primal Concrete Sledge	00:02:13.0000
3	Psycho Holiday	00:05:19.0000
4	Heresy	00:04:46.0000
5	Cemetery Gates	00:07:02.0000
6	Domination	00:05:04.0000
7	Shattered	00:03:21.0000
8	Clash With Reality	00:05:16.0000
9	Medicine Man	00:05:14.0000
10	Message In Blood	00:05:10.0000
11	The Sleep	00:05:47.0000
12	The Art Of Shredding	00:04:20.0000
21	Thousand Lakes	00:02:03.0000
22	Into Hiding	00:03:46.0000
23	The Castaway	00:05:33.0000
24	First Doom	00:03:52.0000
25	Black Winter Day	00:03:51.0000
26	Drowned Maid	00:04:23.0000
27	In The Beginning	00:03:38.0000
28	Forgotten Sunrise	00:04:54.0000

TR_ID	TITLE	DURATION
-------	-------	----------

29	To Fathers Cabin	00:03:50.0000
30	Magic And Mayhem	00:04:28.0000
32	Folk Of The North	00:01:20.0000
33	Moon And Sun	00:03:36.0000
34	Moon And Sun Part II	00:05:10.0000
35	Angel Of Death	00:04:51.0000
36	Piece By Piece	00:02:02.0000
37	Necrophobic	00:01:40.0000
38	Altar Of Sacrifice	00:02:50.0000
39	Jesus Saves	00:02:54.0000
40	Criminally Insane	00:02:23.0000
41	Reborn	00:02:11.0000
42	Epidemic	00:02:23.0000
43	Postmortem	00:02:44.0000
44	Raining Blood	00:04:57.0000
45	Amerikkas Most Wanted	00:04:08.0000
46	Check Yo Self	00:04:16.0000
47	It Was A Good Day	00:04:20.0000
48	Wicked	00:03:56.0000
49	Pushin Weight	00:04:35.0000

TR_ID	TITLE	DURATION
-------	-------	----------

50	You Know How We Do It	00:03:53.0000
51	Steady Mobbin	00:04:10.0000
52	You Can Do It	00:04:19.0000
53	Really Doe	00:04:28.0000
54	Bop Gun (One Nation)	00:04:47.0000
55	What Can I Do?	00:04:26.0000

SQL> select * from tracklists;

ALB_ID	TR_ID
--------	-------

1	1
1	2
1	3
1	4
1	5
1	6

1	7
1	8
1	9
1	10
1	11
1	12
2	21
2	22
2	23
2	24
2	25
2	26
2	27
2	28

ALB_ID	TR_ID
2	29
2	30
3	25
3	32
3	33
3	34
4	35
4	36
4	37
4	38
4	39
4	40
4	41
4	42
4	43
4	44
5	45
5	46
5	47
5	48

ALB_ID	TR_ID
5	49
5	50
5	51
5	52
5	53
5	54

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

SQL> select * from tracks where duration between '00:03:30' and '00:04:00';

TR_ID	TITLE	DURATION
22	Into Hiding	00:03:46.0000
24	First Doom	00:03:52.0000
25	Black Winter Day	00:03:51.0000
27	In The Beginning	00:03:38.0000
29	To Fathers Cabin	00:03:50.0000
33	Moon And Sun	00:03:36.0000
48	Wicked	00:03:56.0000
50	You Know How We Do It	00:03:53.0000

SQL> select * from albums where rating in (3, 4);

ALB_ID	ALBUM	RELEASE_YEAR	LABEL	RATING	NUM_OF_RAT	ST_ID	TYPE	GENRE
2	Tales From The Thousand Lakes	1994	Relapse	4	37	2	album	1
3	Black Winter Day	1994	Relapse	4	23	2	EP	1
5	Icon	2013	Priority Records	3	48	4	compilation	2

SQL> select * from tracks where title like '%Blood';

TR_ID	TITLE	DURATION
10	Message In Blood	00:05:10.0000
44	Raining Blood	00:04:57.0000

3.3. Создайте в запросе вычисляемое поле:

SQL> select 'title and year:', album, release_year from albums where num_of_rat > (200/5);

CONSTANT	ALBUM	RELEASE_YEAR
title and year:	Cowboys from Hell	1990
title and year:	Reign In Blood	1986
title and year:	Icon	2013

3.4. Сделайте выборку всех данных с сортировкой по нескольким полям:

SQL> select * from albums order by release_year desc, album asc;

ALB_ID	ALBUM	RELEASE_YEAR	LABEL	RATING	NUM_OF_RAT	ST_ID	TYPE	GENRE
5	Icon	2013	Priority Records	3	48	4	compilation	2
3	Black Winter Day	1994	Relapse	4	23	2	EP	1
2	Tales From The Thousand Lakes	1994	Relapse	4	37	2	album	1
1	Cowboys from Hell	1990	Atco	5	54	1	album	1
4	Reign In Blood	1986	Def Jam Recordings	5	78	3	album	1

3.5. Создайте запрос, вычисляющий несколько совокупных характеристик таблиц:

SQL> select 'all sold copy:', sum(sold) from cds;

CONSTANT	SUM
all sold copy:	480000000

SQL> select 'number of all tracks', count (*) from tracks;

CONSTANT	COUNT
number of all tracks	46

3.6. Сделайте выборку данных из связанных таблиц (не менее двух примеров):

SQL> select albums.album, studios.title from albums, studios where albums.st_id = studios.st_id;

ALBUM	TITLE
Cowboys from Hell	Pantego Sound Studios
Tales From The Thousand Lakes	Sunlight Studios

Black Winter Day	Sunlight Studios
Reign In Blood	Hit City West
Icon	Death Row Studio

SQL> select artists.artist, albums.album from artists, albums, art_alb where art_alb.art_id = artists.art_id and art_alb.alb_id=albums.alb_id;

ARTIST	ALBUM
Pantera	Cowboys from Hell
Amorphis	Tales From The Thousand Lakes
Amorphis	Black Winter Day
Slayer	Reign In Blood
Ice Cube	Icon

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

SQL> select studios.title, count(albums.album) from studios, albums where studios.st_id = albums.st_id group by title;

TITLE	COUNT
Death Row Studio	1
Hit City West	1
Pantego Sound Studios	1
Sunlight Studios	2

SQL> select studios.title, count(albums.album) from studios, albums where studio s.st_id = albums.st_id group by title having count(*) > 1;

TITLE	COUNT
Sunlight Studios	2

3.8. Придумайте и реализуйте пример использования вложенного запроса:

SQL> select albums.album from albums where genre in (select gen_id from genres where genre = 'Rock');

ALBUM
Cowboys from Hell
Tales From The Thousand Lakes
Black Winter Day
Reign In Blood

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

```
insert into artists values (5, 'Edge of Sanity');
insert into cds values (714679272, 1996, 600000, 1);
insert into studios values (5, 'Cutting Room', 'Sweden');
insert into albums values (6, 'Crimson', 1996, 'Black Mark Production', 5, 18, 5, 'album', 1);
insert into cds_alb values (714679272, 6);
insert into art_alb values (5, 6);
insert into tracks values (13, 'Crimson', '00:40:00');
insert into tracklists values (6, 13);
insert into playlists values ('list4', '12-15-2014');
insert into pl_tr values ('list4', 13);
```

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

```
SQL> update cds set sold = 1700000 where code = 105520746;
SQL> select * from cds;
```

CODE	RELEASE_YEAR	SOLD	STATUS
679137227	1990	100000000	3
781676650	2000	100000000	3
329751487	1986	200000000	3
105520746	2013	1700000	3
714679272	1996	600000	1

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

```
SQL> delete from tracks where tr_id in (select max(tr_id) from tracks);
SQL> select * from tracks;
```

TR_ID	TITLE	DURATION
1	Cowboys From Hell	00:04:07.0000
2	Primal Concrete Sledge	00:02:13.0000
3	Psycho Holiday	00:05:19.0000
4	Heresy	00:04:46.0000
5	Cemetery Gates	00:07:02.0000
6	Domination	00:05:04.0000
7	Shattered	00:03:21.0000
8	Clash With Reality	00:05:16.0000
9	Medicine Man	00:05:14.0000
10	Message In Blood	00:05:10.0000
11	The Sleep	00:05:47.0000
12	The Art Of Shredding	00:04:20.0000
21	Thousand Lakes	00:02:03.0000
22	Into Hiding	00:03:46.0000
23	The Castaway	00:05:33.0000
24	First Doom	00:03:52.0000
25	Black Winter Day	00:03:51.0000
26	Drowned Maid	00:04:23.0000
27	In The Beginning	00:03:38.0000
28	Forgotten Sunrise	00:04:54.0000

TR_ID	TITLE	DURATION
29	To Fathers Cabin	00:03:50.0000
30	Magic And Mayhem	00:04:28.0000
32	Folk Of The North	00:01:20.0000
33	Moon And Sun	00:03:36.0000
34	Moon And Sun Part II	00:05:10.0000
35	Angel Of Death	00:04:51.0000
36	Piece By Piece	00:02:02.0000
37	Necrophobic	00:01:40.0000
38	Altar Of Sacrifice	00:02:50.0000
39	Jesus Saves	00:02:54.0000
40	Criminally Insane	00:02:23.0000
41	Reborn	00:02:11.0000
42	Epidemic	00:02:23.0000
43	Postmortem	00:02:44.0000
44	Raining Blood	00:04:57.0000
45	Amerikkas Most Wanted	00:04:08.0000

46 Check Yo Self	00:04:16.0000
47 It Was A Good Day	00:04:20.0000
48 Wicked	00:03:56.0000
49 Pushin Weight	00:04:35.0000

TR_ID	TITLE	DURATION
50	You Know How We Do It	00:03:53.0000
51	Steady Mobbin	00:04:10.0000
52	You Can Do It	00:04:19.0000
53	Really Doe	00:04:28.0000
54	Bop Gun (One Nation)	00:04:47.0000
13	Crimson	00:40:00.0000

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

```
SQL> delete from genres where genres.genre not in (select genres.genre from albums, genres where albums.genre = genres.gen_id);
SQL> select * from genres;
```

GEN_ID	GENRE
1	Rock
2	Rap

4. Индивидуальное задание

4.1. Вывести список из десяти наиболее популярных треков в плейлистах с количеством появлений этих треков:

```
select first 10 tracks.title, count(pl_tr.tr_id) as cnt_track from tracks, pl_tr
where pl_tr.tr_id = tracks.tr_id group by tracks.title order by cnt_track desc;
```

TITLE	CNT_TRACK
dqnjf	3
eekzvpvaajsld	3
aterfilxzjcywferd	3
cwklaflxfnssqgwaw	3
ckhesndzqhpxanllegnaaw	3
cvawekygywoxritquarmpqigvc	3
igsxeju	3
hictyeey	3
iuviohjzwwgudpnuvylg	3
lrfoos	3

4.2. Вывести пять исполнителей, диски с альбомами которых наиболее часто удостоивались каких-либо наград:

```
select first 5 artists.artist from artists, cds, cds_alb, albums, art_alb, statuses
where cds.status in (1, 2, 3) and artists.art_id = art_alb.art_id and
art_alb.alb_id = albums.alb_id and cds_alb.alb_id = albums.alb_id and
cds_alb.code = cds.code and cds.status = statuses.stat_id
group by artists.artist order by count(*) desc;
```

ARTIST
thuaubudlesufoeqtlz
ipqmgxbhticazksylmle

fxbzyrpt
tkofjq
lwatwkgqxeqyb

4.3. Удалить плейлисты, состоящие не более, чем из пяти треков:

```
delete from pl_tr where pl_tr.name in  
(select playlists.name from playlists, pl_tr where playlists.name = pl_tr.name  
group by playlists.name having count(pl_tr.name) < 5);
```

```
delete from playlists where playlists.name not in (select pl_tr.name from pl_tr);
```

```
select * from playlists;
```

NAME	DATE_OF_CR
mp3	2012-05-13
home	2013-01-30

```
SQL> select * from pl_tr;
```

NAME	TR_ID
home	21
home	22
home	23
home	24
home	25
home	26
home	27
home	32
home	33
home	34
mp3	35
mp3	36
mp3	37
mp3	38
mp3	50
mp3	51
mp3	52
mp3	36926

5. Представления и хранимые процедуры

5.1. Сделайте выборку всех данных из каждой таблицы

```
create view t1_1 as select * from artists;  
create view t1_2 as select * from cds_alb;  
create view t1_3 as select * from cds;  
create view t1_4 as select * from albums;  
create view t1_5 as select * from studios;  
create view t1_6 as select * from playlists;  
create view t1_7 as select * from pl_tr;  
create view t1_8 as select * from tracks;  
create view t1_9 as select * from tracklists;
```

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

```
create view t2_1 as select * from tracks where duration
```

```
between '00:03:30' and '00:04:00';
create view t2_2 as select * from albums where rating in (3, 4);
create view t2_3 as select * from tracks where title like '%Blood';
```

5.3. Создайте в запросе вычисляемое поле:

```
create view t3_1 (string, album, release_year)
as select 'title and year:', album, release_year from albums
where num_of_rat > (200/5);
```

5.4. Сделайте выборку всех данных с сортировкой по нескольким полям:

```
create view t4_1 as select * from albums order by release_year desc, album asc;
```

5.5. Создайте запрос, вычисляющий несколько совокупных характеристик таблиц:

```
create view t5_1 (string, sold) as select 'all sold copy:', sum(sold) from cds;
create view t5_2 (string, cnt) as select 'number of all tracks', count (*) from tracks;
```

5.6. Сделайте выборку данных из связанных таблиц (не менее двух примеров):

```
create view t6_1 (album, studio) as select albums.album, studios.title from albums, studios
where albums.st_id = studios.st_id;
create view t6_2 (artist, album) as select artists.artist, albums.album from artists, albums, art_alb
where art_alb.art_id = artists.art_id and art_alb.alb_id = albums.alb_id;
```

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

```
create view t7_1 (studio, cnt) as select studios.title, count(albums.album) from studios, albums
where studios.st_id = albums.st_id group by title;
create view t7_2 (studio, cnt) as select studios.title, count(albums.album) from studios, albums
where studios.st_id = albums.st_id group by title having count(*) > 1;
```

5.8. Придумайте и реализуйте пример использования вложенного запроса:

```
create view t8_1 as select albums.album from albums where genre in
(select gen_id from genres where genre = 'Rock');
```

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

```
create or alter procedure P1
as
begin
insert into artists values (5, 'Edge of Sanity');
insert into cds values (714679272, 1996, 600000, 1);
insert into studios values (5, 'Cutting Room', 'Sweden');
insert into albums values (6, 'Crimson', 1996, 'Black Mark Production', 5, 18, 5, 'album', 1);
insert into cds_alb values (714679272, 6);
insert into art_alb values (5, 6);
insert into tracks values (13, 'Crimson', '00:40:00');
insert into tracklists values (6, 13);
```

```
insert into playlists values ('list4', '12-15-2014');
insert into pl_tr values ('list4', 13);
end
```

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

```
create or alter procedure P2
as
begin
update cds set sold = 1700000 where code = 105520746;
end
```

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

```
create or alter procedure P3
as
begin
delete from tracks where tr_id in (select max(tr_id) from tracks);
end
```

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

```
create or alter procedure P4
as
begin
delete from genres where genres.genre not in (select genres.genre from albums, genres where
albums.genre = genres.gen_id);
end
```

5.13. Индивидуальные задания:

```
create view t9 (tracks, cnt) as
select first 10 tracks.title, count(pl_tr.tr_id) as cnt_track from tracks, pl_tr
where pl_tr.tr_id = tracks.tr_id group by tracks.title order by cnt_track desc;
```

```
create view t10 as
select first 5 artists.artist from artists, cds, cds_alb, albums, art_alb, statuses
where cds.status in (1, 2, 3) and artists.art_id = art_alb.art_id and
art_alb.alb_id = albums.alb_id and cds_alb.alb_id = albums.alb_id and
cds_alb.code = cds.code and cds.status = statuses.stat_id
group by artists.artist order by count(*) desc;
```

```
create or alter procedure P5
as
begin
delete from pl_tr where pl_tr.name in
(select playlists.name from playlists, pl_tr where playlists.name = pl_tr.name
group by playlists.name having count(pl_tr.name) < 5);
```

```
delete from playlists where playlists.name not in (select pl_tr.name from pl_tr);  
end
```

6. Выводы

В ходе выполнения работы познакомились с основами языка манипулирования данными. Язык SQL-DML является одним из подмножеств языка структурированных запросов SQL. В его состав входят команды модификации, выборки, группировки, сортировки данных и др. В том числе в рамках SQL-DML возможно использование вычисляемых полей. Основные команды были продемонстрированы в ходе выполнения работы.

Все запросы типа select в итоге были сохранены в виде представлений. Представления – это таблицы, содержимое которых берется из других таблиц посредством запроса. При этом новые данные не создаются. Эти таблицы реально в БД не хранятся. Представления позволяют облегчить работу с данными в БД при многократном вызове тех или иных повторяющихся запросов.

Команды модификации данных были сохранены в виде хранимых процедур. Хранимые процедуры – это ряд операций над данными, которые реализуют общую для всех пользователей логику и не связаны с пользовательским интерфейсом. В связи с этими особенностями хранимые процедуры обычно выносятся на сервер, а пользователь только обращается к ним, при этом экономя время на ввод запроса и проверку его правильности.