

Санкт-Петербургский государственный политехнический университет
Кафедра компьютерных систем и программных технологий

Отчёт по лабораторной работе

Дисциплина: Базы данных

Тема: Язык SQL-DML

Выполнил студент гр. 43501/3

(подпись) В.Е. Бушин

Руководитель

(подпись) А.В. Мяснов

“ _ ” _____ 2015 г.

Санкт-Петербург

2015

1. Цель работы

Познакомить студентов с основами проектирования схемы БД, языком описания сущностей и ограничений БД SQL-DDL.

2. Язык SQL

Язык SQL (Structured Query Language) - язык структурированных запросов. Он позволяет формировать весьма сложные запросы к базам данных. В SQL определены два подмножества языка:

- SQL-DDL (Data Definition Language) - язык определения структур и ограничений целостности баз данных. Сюда относятся команды создания и удаления баз данных; создания, изменения и удаления таблиц; управления пользователями и т.д.
- SQL-DML (Data Manipulation Language) - язык манипулирования данными: добавление, изменение, удаление и извлечение данных, управления транзакциями.

3. 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.

4. Ход работы

- 1) Был самостоятельно изучен язык SQL-DML.
- 2) Запросы из списка стандартных запросов:

- Выборка всех данных из каждой таблицы:

```
create view ClubSelect as select * from Club;  
create view PlayerSelect as select * from Player;  
create view MatchSelect as select * from Match;  
create view ActSelect as select * from Act;  
create view MatchstatSelect as select * from Matchstat;  
create view ChampionshipSelect as select * from Championship;  
create view TransferSelect as select * from Transfer;  
commit;
```

Пример выборки данных из таблицы клубов:

ID	CLUB_NAME	CITY
1	Arsenal	London
2	Brighton & Hove Albion FC	Brighton
3	Reading	Reading
4	Cardiff City	Cardiff

Рис. 1. Выборка данных из таблицы клубов

- Выборка данных из одной таблицы при нескольких условиях, с использованием логических операций, LIKE, BETWEEN, IN:

```
create view selArsenal as select * from club where CLUB_NAME like 'Arsenal';  
create view selMatch as select * from match where STAGE between 32 and 8;  
create view selPlayer as select * from player where NUMBER in (7,8,9,10);
```

Пример выборки с использованием оператора IN:

ID	PLAYER_NAME	NUMBER	CLUBID	COUNTRY
2	Samuel Baldock	9	2	England
3	Pavel Pogrebnyak	7	3	Russia
4	Kenwyne Jones	9	4	Trinidad and Tobago

Рис. 2. Выборка игроков с номерами 7,8,9,10

- Вычисляемое поле в запросе:

Для примера сложим значения полей clubID и number из таблицы player

```
create view vichisl as select clubID, number, clubID+number as summa from player;
commit;
```

Результат:

CLUBID	NUMBER	SUMMA
1	14	15
2	9	11
3	7	10
4	9	13

Рис. 3. Сумма 2 полей из таблицы игроков

- Выборка всех данных с сортировкой по нескольким полям:

```
create view playerSort as select * from player order by clubID asc, number asc;
commit;
```

Пример сортировки:

ID	PLAYER_NAME	NUMBER	CLUBID	COUNTRY
6	Laurent Koscielny	6	1	France
5	Mesut Özil	11	1	Germany
1	Theo Walcott	14	1	England
2	Samuel Baldock	9	2	England
7	Danny Holla	22	2	Netherlands
8	Christopher Gunter	2	3	Wales
3	Pavel Pogrebnyak	7	3	Russia
4	Kenwyne Jones	9	4	Trinidad and Tobago
9	Federico Macheda	14	4	Wales

Рис. 4. Игроки, отсортированные по клубам и по номерам

- Выборка данных из связанных таблиц:

```
create view allPlayers as select Club.club_name as Team, Player.player_name as
Player_name, Player.number as Number from Club, Player where Club.id =
Player.clubid;
```

```
create view allActs as select Player.player_name as name, Act.acttype,
Act.match_minute from Player, act where Player.id = Act.playerid;
```

Пример выборки игроков:

TEAM	PLAYER_NAME	NUMBER
Arsenal	Theo Walcott	14
Arsenal	Mesut Özil	11
Arsenal	Laurent Koscielny	6
Brighton & Hove Albion FC	Samuel Baldock	9
Brighton & Hove Albion FC	Danny Holla	22
Reading	Pavel Pogrebnyak	7
Reading	Christopher Gunter	2
Cardiff City	Kenwyne Jones	9
Cardiff City	Federico Macheda	14

Рис. 5. Выборка всех игроков

- Запрос, рассчитывающий совокупную характеристику с использованием группировки, наложите ограничение на результат группировки:

```
create view countActs as select Act.acttype as act_type,
count(Act.acttype) as numb from Act group by Act.acttype;
```

Результат запроса:

ACT_TYPE	NUMB
goal	3

Рис. 6. Количество типов действий для игроков

- Пример использования вложенного запроса:

```
create view playersWithoutActs as select * from player where id not in (select
playerid from Act);
```

Результат запроса:

ID	PLAYER_NAME	NUMBER	CLUBID	COUNTRY
3	Pavel Pogrebnyak	7	3	Russia
5	Mesut Ozil	11	1	Germany
6	Laurent Koscielny	6	1	France
7	Danny Holla	22	2	Netherlands
8	Pavel Pogrebnyak	2	3	Wales
9	Federico Macheda	14	4	Italy

Рис. 7. Игроки без полезных действий

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

```
create procedure ins_club (i integer, name varchar(25), city varchar(25)) as begin
insert into Club values (:i, :name, :city);
end;
```

```
create procedure ins_player (i integer, name varchar(50), numb integer,
clubid integer, country varchar(25), price integer) as begin
insert into Player values (:i, :name, :numb, :clubid, :country, :price);
end;
```

```
create procedure ins_match (i integer, d date, stage stagedomain, home integer,
away integer, result varchar(25), win integer) as begin
insert into Match values (:i, :d, :stage, :home, :away, :result, :win);
end;
```

```
create procedure ins_act (i integer, playerid integer, acttype acttypedomain,
m integer) as begin
insert into Act values (:i, :playerid, :acttype, :m);
end;
```

```
create procedure ins_matchstat (i integer, matchid integer, hS integer,
awS integer, actid integer) as begin
insert into Matchstat values (:i, :matchid, :hS, :awS, :actid);
end;
```

```
create procedure ins_tranfer (i integer, playerid integer, price integer,
source integer, dest integer) as begin
insert into Transfer values (:i, :playerid, :price, :source, :dest);
end;
```

```
create procedure ins_champ (i integer, clubid integer, played integer, won integer,
drawn integer, lost integer, points integer, gf integer, ga integer) as begin
insert into Championship values (:i, :clubid, :played, :won, :drawn, :lost, :points,
:gf, :ga);
end;
```

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

```
create procedure upd_price (val integer) as begin
update Player set transferprice = :val + transferprice where Player.clubid = 3;
end;
```

Используем процедуру:

```
execute procedure upd_price (10);
```

Результат:

3	Pavel Pogrebnyak	7	3	Russia	1 000 010
4	Kenwyne Jones	9	4	Trinidad and Tobago	2 500 000
5	Mesut Ozil	11	1	Germany	40 000 000
6	Laurent Koscielny	6	1	France	20 000 000
7	Danny Holla	22	2	Netherlands	1 250 000
8	Christopher Gunter	2	3	Wales	2 500 010

Рис. 8. Выполнение операции UPDATE

- С помощью оператора DELETE удалите запись, имеющую максимальное (минимальное) значение некоторой совокупной характеристики
Удалим запись с минимальным значением transferprice и с параметром clubid i из таблицы player:

```
create procedure del_min_pr (i integer) as begin
delete from player where clubid=:i and transferprice=(select min(transferprice)
from player where clubid=:i);
end;
```

- С помощью оператора DELETE удалите записи в главной таблице, на которые не ссылается подчиненная таблица (используя вложенный запрос)
Удалим матчи, на которые нет статистики:

```
create procedure del_match as begin
delete from Match where id not in (select matchid from Matchstat);
end;
```

3) Индивидуальные задания, полученные у преподавателя:

- Отобразить суммарную трансферную стоимость каждого футболиста за его карьеру.

В таблицах находится по 100000 записей, время выполнения – 47 мс:

```
create view showSumPrice as select Player.player_name as name,
sum(Transfer.price) as Sum_price from Transfer, Player where
Transfer.playerid=Player.id group by Player.player_name;
```

Результат запроса:

NAME	SUM_PRICE
Christopher Gunter	280 260 536
Danny Holla	323 311 379
Federico Macheda	96 704 934
Kenwyne Jones	416 098 814
Laurent Koscielny	148 392 741
Mesut Ozil	142 520 474
Pavel Pogrebnyak	253 531 200
Samuel Baldock	160 988 049
Theo Walcott	172 764 465

Рис. 9. Суммарная трансферная стоимость для каждого игрока

- Отобразить всех победителей кубковых турниров за выбранный период
Процедура запускалась при большом объёме данных (100000 записей), время выполнения – 32 мс:

```
create view showAllWinners as select Club.club_name as name, Match.matchdate as data
from club, Match where Club.id=Match.winner and Match.stage=1 and
Match.matchdate between '2014-12-01' and '2015-02-01';
```

Результат запроса:

NAME	DATA
x%	10.01.2015 00:00
O%C(Iv}FywV]G/:N)Z\mX	02.12.2014 00:00
{ca<Q9c^/Z{:c7	20.01.2015 00:00
L_`c(?X[_]	25.01.2015 00:00
IFH'7nFX>sk)coX	13.01.2015 00:00
OPz_\$_+EJF4O3L(#p	20.01.2015 00:00
Np	17.12.2014 00:00
DOBz2>()HE*N1F	14.12.2014 00:00
5~5V1A6N}b@2sS}kbI? =	28.12.2014 00:00
vSY!I- p	23.12.2014 00:00
~b"zdN*WL'~aa@?hn7L%-m...	26.01.2015 00:00
s8q-sF	21.01.2015 00:00
aSb4.F?M	18.01.2015 00:00
,?yO!IvBd3!3fdpc2!gk	13.01.2015 00:00
NcTg+C*B?	17.01.2015 00:00
2\&g?3\h	16.01.2015 00:00
!U.?r"uUPJ_AB?3I-R	23.12.2014 00:00
XX	25.01.2015 00:00
D~H: %@^\ 2F5lw{V<oX,8 6	17.12.2014 00:00
~Na#)couePyD^p	01.02.2015 00:00
{+(G(OD\Txe<@	22.12.2014 00:00
+dgPae[{	07.01.2015 00:00
`uYU*2C~SL16L]i2N0k>.>	22.12.2014 00:00
B_`Dk8KTrk&r;mM	07.12.2014 00:00
c!qtMwahrNvYL+.za{c=M	09.01.2015 00:00
YKI	03.01.2015 00:00
OQsHk	02.12.2014 00:00
'\ ~	09.01.2015 00:00
Ten	05.12.2014 00:00
"LchF<4 n/jR >2a	19.01.2015 00:00

Рис. 10. Победители кубковых турниров за определённое время

- Удалить неиспользуемые типы действий игроков за выбранный период

```
create procedure delUnusedActs as begin
delete from Act where Act.id not in(select actsid from Matchstat);
end
```

Процедура запускалась при большом объёме данных (100000 записей), время выполнения скрипта 1110 мс.

5. Вывод

Во время работы был изучен язык SQL-DML. Использовались следующие команды языка DML: insert (добавить), update(обновить), delete(удалить), select(выбрать). Созданы запросы извлечения данных из БД в соответствии с индивидуальным заданием. Также изучены представления и хранимые процедуры SQL. Эти сущности позволяют хранить запросы и скрипты в самой БД. С помощью представлений удобно реализовывать запросы добавления и удаления данных. С помощью хранимых процедур удобно реализовывать запросы добавления и удаления данных.