

Федеральное государственное автономное образовательное учреждение
высшего образования «Санкт-Петербургский политехнический университет
Петра Великого»
Институт компьютерных наук и технологий
Кафедра Компьютерных систем и программных технологий

Отчёт по лабораторной №4

Дисциплина: Базы данных

Тема: SQL-DML

Выполнил студент гр. 43501/4

(подпись) В.С. Филиппов

Руководитель

(подпись) А.В. Мяснов

“ ” _____ 2015 г.

Санкт -Петербург

2015

Цель работы

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

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.

Выполнение работы

Запросы из списка стандартных запросов

```
/*1 Сделайте выборку всех данных из каждой таблицы*/
CREATE VIEW playersSel AS SELECT * FROM players;
CREATE VIEW clubsSel AS SELECT * FROM clubs;
CREATE VIEW leaguesSel AS SELECT * FROM leagues;
CREATE VIEW seasonsSel AS SELECT * FROM seasons;
CREATE VIEW matchesSel AS SELECT * FROM matches;
CREATE VIEW eventsSel AS SELECT * FROM events;
CREATE VIEW standingsSel AS SELECT * FROM standings;
CREATE VIEW club_players_list_historySel AS SELECT * FROM
club_players_list_history;
CREATE VIEW current_club_players_listSel AS SELECT * FROM
current_club_players_list;
commit;

/*
select * from leaguesSel;
      LEAGUE_ID NAME                KIND
=====
1 RFPL                Country
*/

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

/*выборка нападающих*/
CREATE VIEW selectForwards AS SELECT * FROM players where pos like 'Forward';
/*выборка игроков родившихся в 90-х*/
CREATE VIEW selectBornIn90 AS SELECT * FROM players where birthdate between
'01-JAN-1990' and '31-DEC-1999';
/*выборка игроков, чей день рождения как у друзей*/
CREATE VIEW selectBornIn1607199412081994 AS SELECT * FROM players where
birthdate in ('16-JUL-1994', '12-AUG-1994');
commit;

/*
SQL> select first 5 * from selectBornIn90;

      PLAYER_ID NAME                BIRTHDATE POS
```

```
=====
17 dqoqzajphlfg 1997-07-27 Defender
68 cmtcoqtrdtla 1997-05-03 Defender
77 ioecerwhysgnasph 1998-03-11 Midfielder
133 jgw 1996-01-11 Defender
200 qzovywgevhmgrtu 1992-05-15 Goalkeeper
```

```
SQL> select first 5 * from selectBornIn1607199412081994;
```

```
=====
PLAYER_ID NAME BIRTHDATE POS
=====
43739 irfogvyumklawn 1994-08-12 Goalkeeper
46635 b 1994-07-16 Coach
92806 uqimuqtizlh 1994-08-12 Forward
```

```
SQL>
*/
```

```
/*3 Создайте в запросе вычисляемое поле*/
```

```
/*выборка таблицы сезона с альтернативным вариантом вычисления очков, где
разница мячей имеет больший вес ( 2*разницу_мячей + стандартные_очки )*/
CREATE VIEW alterPointsInRFPL15_16 as select club_id, (points+2*diff_goals)
as alterPoints from standings where season_id like 1;
commit;
```

```
/*
```

```
SQL> select * from alterPointsInRFPL15_16;
```

```
=====
CLUB_ID ALTERPOINTS
=====
1 8
2 1
3 -1
```

```
*/
```

```
/*4 Сделайте выборку всех данных с сортировкой по нескольким полям*/
```

```
/*выборка с=отсортирована по позиции, имени, и дате рождения*/
CREATE VIEW selectPlayersSort as select * from players order by pos asc, name
asc, birthdate;
commit;
```

```
/*
```

```
SQL> select * from selectPlayersSort;
```

```
=====
PLAYER_ID NAME BIRTHDATE POS
=====
5 Berezuzkiy V 1987-05-07 Defender
1 Dzuba 1988-08-22 Forward
2 Kerjakov 1982-06-10 Forward
6 Movsisyan 1991-10-30 Forward
3 Lodigin 1990-06-01 Goalkeeper
4 Natho 1983-07-02 Midfielder
```

```
*/
```

```
/*5 Создайте запрос, вычисляющий несколько совокупных характеристик таблиц*/
```

```
/*подсчёт голов в сезоне РФПЛ 15-16*/
CREATE VIEW selectSumOfGoalsRFPL15_16 as select sum(goals) as sumOfGoals from
standings where season_id like 1;
commit;
```

```
/*
```

```
SQL> select * from selectSumOfGoalsRFPL15_16;
```

```

SUMOFGOALS
=====
5
*/

```

/*6 Сделайте выборку данных из связанных таблиц (не менее двух примеров)*/

/*Команда - количество очков*/

```

CREATE VIEW selectClubsAndPoints as select Clubs.name, Standings.points from
Clubs, Standings where Clubs.club_id = Standings.club_id;
commit;
/*
SQL> select * from selectClubsAndPoints;

```

```

NAME          POINTS
=====
Zenit          4
Spartak        3
CSKA           1
*/

```

/*команда 1 команда 2 счёт*/

```

CREATE VIEW selectMatchesGoals as select c1.name as club1, c2.name as club2,
m.first_club_goals, m.second_club_goals
from Matches m inner join clubs c1 on c1.club_id = m.first_club_id
inner join clubs c2 on c2.club_id = m.second_club_id;
commit;
/*
SQL> select * from selectMatchesGoals;

```

```

CLUB1          CLUB2          FIRST_CLUB_GOALS SECOND_CLUB_GOALS
=====
Spartak        Zenit          0              2
Spartak        CSKA           1              0
CSKA           Zenit          1              1
*/

```

/*7 Создайте запрос, рассчитывающий совокупную характеристику с использованием группировки, наложите ограничение на результат группировки*/

/*количество игроков каждой позиции*/

```

CREATE VIEW selectCountOfPlayersByPos as select Players.pos as "position",
COUNT(Players.pos) as "count" from Players group by Players.pos;
commit;
/*
SQL> select * from selectCountOfPlayersByPos;

```

```

position       count
=====
Defender       1
Forward        3
Goalkeeper     1
Midfielder     1
*/

```

/*8 Придумайте и реализуйте пример использования вложенного запроса*/

/*показать игроков Зенита*/

```

CREATE VIEW selectPlayersOfZenit as select * from players where player_id in
(select player_id from current_club_players_list where club_id = (select
club_id from clubs where name = 'zenit'));

```

```
commit;
```

```
/*  
SQL> select * from selectPlayersOfZenit;
```

```
      PLAYER_ID NAME                BIRTHDATE POS  
=====
```

PLAYER_ID	NAME	BIRTHDATE	POS
1	Dzuba	1988-08-22	Forward
2	Kerjakov	1982-06-10	Forward
3	Lodigin	1990-06-01	Goalkeeper

```
*/
```

```
/*9 С помощью оператора INSERT добавьте в каждую таблицу по одной записи*/
```

```
create procedure ins_player(n VARCHAR(16), b DATE, p pos_type)  
as  
begin  
    insert into players (name, birthdate, pos) values (:n, :b, :p);  
end;  
  
create procedure ins_club(n VARCHAR(16), b DATE, ni VARCHAR(17), c  
VARCHAR(16))  
as  
begin  
    insert into clubs (name, birthdate, nickname, country) values (:n, :b,  
:ni, :c);  
end;  
  
create procedure ins_league(n VARCHAR(16), k league_type)  
as  
begin  
    insert into leagues (name, kind) values (:n, :k);  
end;  
  
create procedure ins_season(lid INTEGER, p VARCHAR(9))  
as  
begin  
    insert into seasons (league_id, period) values (:lid, :p);  
end;  
  
create procedure ins_match(d DATE, sid INTEGER, fcid INTEGER, scid INTEGER,  
fcg SMALLINT, scg SMALLINT)  
as  
begin  
    insert into matches (match_date, season_id, first_club_id,  
second_club_id, first_club_goals, second_club_goals) values (:d, :sid, :fcid,  
:scid, :fcg, :scg);  
end;  
  
create procedure ins_event(pid INTEGER, "mid" INTEGER, em SMALLINT, e  
event_type, ag_io CHAR(1))  
as  
begin  
    insert into events (player_id, match_id, event_minute, event, ag_io)  
values (:pid, :mid, :em, :e, :ag_io);  
end;  
  
create procedure ins_club_in_season(sid INTEGER, cid INTEGER, v SMALLINT, d  
SMALLINT, l SMALLINT, g SMALLINT, mg SMALLINT, dg SMALLINT, p SMALLINT)  
as  
begin  
    insert into standings (season_id, club_id, victories, draws, losses,  
goals, missed_goals, diff_goals, points) values (:sid, :cid, :v, :d, :l, :g,  
:mg, :dg, :p);
```

```

end;

create procedure ins_player_in_club_history(cid INTEGER, pid INTEGER, db
DATE, de DATE)
as
begin
    insert into club_players_list_history (club_id, player_id, date_begin,
date_end) values (:cid, :pid, :db, :de);
end;

create procedure ins_player_in_club_curr(cid INTEGER, pid INTEGER)
as
begin
    insert into current_club_players_list (club_id, player_id) values (:cid,
:pid);
end;

commit;

/*
execute procedure ins_player('Shatov', '22-AUG-1992', 'Midfielder');

execute procedure ins_league('EPL', 'Country');
execute procedure ins_season(2, '2015/2016');
execute procedure ins_event(2, 2, 90, 'INJURY', 0);
execute procedure ins_player_in_club_history(1, 7, '20-JUL-2015', NULL);
execute procedure ins_player_in_club_curr(1, 7);
*/
/*ins_event not work?*/

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

/* +100 очков Зениту во всех сезонах во всех лигах*/
create procedure plus100forZenit
as
begin
    update standings set points = points + 100 where club_id = (select
club_id from clubs where name = 'Zenit');
end;

/*
execute procedure plus100forZenit
*/

/*11 С помощью оператора DELETE удалите запись, имеющую максимальное
(минимальное) значение некоторой совокупной характеристики*/
/*удалим самого молодого игрока*/
create procedure deleteYoungPlayer
as
begin
    delete from Players where birthdate = (select max(birthdate) from
Players);
end;

/*
execute procedure ins_player('young_man', '2-AUG-2015', 'Midfielder');
execute procedure deleteYoungPlayer;
*/

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

```

```

create procedure deleteNotUseLeagues
as
begin
    delete from Leagues where league_id not in (select league_id from
Seasons);
end;
/*
execute procedure ins_league('FPL', 'Country');
execute procedure deleteNotUseLeagues;
*/

```

Запросы индивидуального задания

```

/* 1 Вывести 10 самых результативных матчей за всю историю. */
/* При условии что таблица goals и таблица matches соответствуют друг другу
*/

```

```

create or alter view selectMoreResultsMatches as
select c1.name as club1, c2.name as club2, m.first_club_goals,
m.second_club_goals, m.match_date
from (select * from
    (
        select first 10 match_id as mid, count(*) goal_count
        from events where event = 'GOAL' group by match_id
        order by goal_count desc) as tmp
    join matches on matches.match_id = tmp.mid
    ) m

inner join clubs c1 on c1.club_id = m.first_club_id
inner join clubs c2 on c2.club_id = m.second_club_id;

```

```

/*
SQL> select * from selectMoreResultsMatches;

```

CLUB1	CLUB2	FIRST_CLUB_GOALS	SECOND_CLUB_GOALS	MATCH_DATE
Spartak	Zenit	0	2	2015-09-26
CSKA	Zenit	1	1	2015-11-30
Spartak	CSKA	1	0	2015-11-30

```

/* 2 Вывести 5 команд с наибольшим количеством побед за выбранный период. */

```

```

create or alter view selectTopClubs as
select clubs.name, wins from
    (
        select first 5 cid, count(cid) wins from
            (
                select first_club_id cid from matches
                where matches.match_date between '01-SEP-2015' and '31-DEC-
2015'
                and matches.first_club_goals > matches.second_club_goals
            union all
                select second_club_id cid from matches
                where matches.match_date between '01-SEP-2015' and '31-DEC-
2015'
                and matches.first_club_goals < matches.second_club_goals
            )
        group by cid order by wins desc
    )
, clubs
where clubs.club_id = cid;

```

```

/*
SQL> select * from selectTopClubs;

NAME                               WINS
=====
Zenit                             1
Spartak                           1
*/

/* 3 Удалить неиспользуемые лиги. */
/* см. 12 пункт выше */

```

Вывод

В данной работе была осуществлена работа с базой данных посредством языка SQL-DML. Язык SQL-DML легко позволяет обрабатывать данные. Имеет операции работы со множествами, работу с локальными переменными.

По ходу работы было сложно адаптироваться к синтаксису и семантике языка. Это связано с тем, что (по сравнению с обычными языками программирования) приходится держать в голове много данных и связей между ними. Новой была работа с множествами.

Освоены операции выборки, вставки и обновления данных как с условиями, так и в зависимости от связей между таблицами.

Были созданы представления. Они позволяют хранить написанные ранее запросы внутри БД. В последствии из них можно осуществлять выборку. Так же были использованы ХП, которые позволяют сохранить проделанные ранее действия по имени и входным параметрам. ХП имеют ещё больше возможностей, которые мы рассмотрим в следующей работе.