

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 goalsSel AS SELECT * FROM goals;
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;
```

```
/*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;
```

```
/*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;
```

```
/*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;
```

```
/*команда 1 команда 2 счёт*/
/*CREATE VIEW selectMatchesResults as select Clubs.name as club1, Clubs.name
as club2, Matches.first_club_goals, Matches.second_club_goals from Clubs,
Matches where Matches.first_club_id = Clubs.club_id or Matches.second_club_id
= Clubs.club_id;
commit;*/
```

```
/*голы: игрок дата матча минута матча*/
CREATE VIEW selectGoalsPlayers as select Players.name, Matches.match_date,
Goals.goal_minute from Players, Matches, Goals where Goals.player_id =
Players.player_id and Goals.match_id = Matches.match_id;
commit;
```

```
/*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;

/*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;

/*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_goal(pid INTEGER, "mid" INTEGER, gm SMALLINT, ag
CHAR(1))
as
begin
    insert into goals (match_id, player_id, goal_minute, autogoal) values
(:pid, :"mid", :gm, :ag);
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_club('CSKA', '25-MAY-1950', 'Armetci', 'Russia');
execute procedure ins_league('EPL', 'Country');
execute procedure ins_season(2/*lid*/, '2015/2016');
execute procedure ins_match('25-OCT-2015',1/*Zenit*/,100003/*CSKA*/,1,7,0);
execute procedure ins_goal(2, 100004, 87, 0);
execute procedure ins_club_in_season(1,100003,0,1,0,2,2,0,1);
execute procedure ins_player_in_club_history(1, 100004/*Shatov*/, '20-JUL-
2015', NULL);
execute procedure ins_player_in_club_curr(1, 100004);
*/

/*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 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 deleteNotUseLeagues;
*/

```

Запросы индивидуального задания

```

/* 1 Вывести 10 самых результативных матчей за всю историю. */
create view selectMoreResultsMatches as select first 10 * from matches order
by (first_club_goals+second_club_goals) desc;

```

```

/*
SQL> select * from selectMoreResultsMatches;

```

```

MATCH ID MATCH DATE SEASON ID FIRST CLUB ID SECOND CLUB ID
FIRST CLUB GOALS SECOND CLUB GOALS FIRST CLUB YELLOW SECOND CLUB YELLOW
FIRST CLUB RED SECOND
 CLUB RED FIRST CLUB PERCENT FIRST CLUB PRESSURE SECOND CLUB PRESSURE
FIRST CLUB SHORTS SECOND CLUB SHORTS FIRST CLUB SHORTS IN TARGET
SECOND CLUB SHORTS IN TAR
GET

```

```

=====
=====
=====
=====
=====
=====
=====
=====
=====
=====
=====

```

```

=====
=====
=====
=====
=====
=====
=====
=====
=====
=====
=====

```

```

=====
=====
=====
=====
=====
=====
=====
=====
=====
=====
=====

```

```

/* 2 Вывести 5 команд с наибольшим количеством побед за выбранный период. */
create view selectTopClubs as select name from clubs where club_id in (select
first 5 club_id from standings order by points);

```

```

/*
SQL> select * from selectTopClubs;

```

```

NAME
=====
Zenit
Spartak
CSKA

```

```

SQL>
*/

```

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

Заключение

В данной работе была осуществлена работа с базой данных посредством языка SQL-DML. Освоены операции выборки, вставки и обновления данных как с условиями, так и в зависимости от связей между таблицами.

Были созданы представления и ХП. Они позволяют хранить данные внутри БД. Некоторые задачи были достигнуты с помощью вложенных запросов.