

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

Отчёт по лабораторной работе

Дисциплина: Базы данных

Тема: Хранимые процедуры

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 возможностями реализации более сложной обработки данных на стороне сервера с помощью хранимых процедур.

2. Программа работы

- 1) Изучить возможности языка PSQL
- 2) Создать две хранимые процедуры в соответствии с индивидуальным заданием, полученным у преподавателя
- 3) Выложить скрипт с созданными сущностями в svn
- 4) Продемонстрировать результаты преподавателю

3. Ход работы

- 1) Был самостоятельно изучен язык PSQL.
- 2) В соответствии с индивидуальным заданием были созданы 2 хранимые процедуры:
 - Для заданного игрока рассчитать статистику по показателям за сезон. Вывести отличие результатов от средних по всем игрокам аналогичного амплуа. Чтобы выполнить требуемое задание, нужно в таблицу Player добавить поле «позиция», а также добавить в БД домен значений для этого поля:

```
connect 'D:\database\FOOTBALL.fdb' user 'SYSDBA' password 'masterkey';

create domain PlayerPositionDomain
AS varchar(20)
check (value in ("goalkeeper", "central defender", "fullback", "central midfielder",
"defensive midfielder", "attacking midfielder", "winger", "centre forward",
"striker"));

alter table Player add PlayerPosition PlayerPositionDomain;

commit;
```

Процедура:

```
create procedure playerstat(plid integer)
returns (name varchar(50), goals float, assists float, yellowCards float, redCards float,
raznGoals float, raznAssists float, raznYellowCards float, raznRedCards float) as
declare variable pos varchar(20);
declare variable cnt int;
declare variable avGoals float;
declare variable avAssists float;
declare variable avYellowCards float;
declare variable avRedCards float;
declare variable tmpGoals float;
declare variable tmpAssists float;
declare variable tmpYellow float;
declare variable tmpRed float;
begin
select Player.player_name, Player.playerposition from Player, Act where
:plid=Player.id
and :plid=Act.playerid into :name, :pos;
select count(Act.acttype) from Act where :plid=Act.playerid and Act.acttype="goal"
into :goals;
```

```

select count(Act.acttype) from Act where :plid=Act.playerid and
Act.acttype="assist" into :assists;
select count(Act.acttype) from Act where :plid=Act.playerid and
Act.acttype="yellow card" into :yellowCards;
select count(Act.acttype) from Act where :plid=Act.playerid and Act.acttype="red
card" into :redCards;

select count(Player.playerposition) from Player where Player.playerposition=:pos
and Player.id<1000 into :cnt;
select count(Act.acttype) from Act, Player where Player.id=Act.playerid
and Act.acttype="goal" and Player.playerposition=:pos into :tmpGoals;
select count(Act.acttype) from Act, Player where Player.id=Act.playerid
and Act.acttype="assist" and Player.playerposition=:pos into :tmpAssists;
select count(Act.acttype) from Act, Player where Player.id=Act.playerid
and Act.acttype="yellow card" and Player.playerposition=:pos into :tmpYellow;
select count(Act.acttype) from Act, Player where Player.id=Act.playerid
and Act.acttype="red card" and Player.playerposition=:pos into :tmpRed;

avGoals=:tmpGoals/:cnt;
avAssists=:tmpAssists/:cnt;
avYellowCards=:tmpYellow/:cnt;
avRedCards=:tmpRed/:cnt;
raznGoals=:goals-:avGoals;
raznAssists=:assists-:avAssists;
raznYellowCards=:yellowCards-:avYellowCards;
raznRedCards=:redCards-:avRedCards;
end;

```

- Обновить турнирную таблицу с учетом результатов очередного тура.
Добавлены две новые таблицы для чемпионата:

```

connect 'D:\database\FOOTBALL.fdb' user 'SYSDBA' password 'masterkey';

create table Match_Champ(
    id int primary key,
    matchDate DATE,
    tour int,
    home int not null,
    away int not null,
    result varchar(25)
);

alter table MatchChamp add constraint match_champ_to_home
foreign key (home) references Club(id);
alter table MatchChamp add constraint match_champ_to_away
foreign key (away) references Club(id);

create table MatchStat_Champ (
    id int primary key,
    matchID int not null,
    home_scored integer,
    away_scored integer,

```

```

        actsID int not null
    );

alter table MatchStatChamp add constraint matchStat_champ_to_match
    foreign key (matchID) references MatchChamp(id);
alter table MatchStatChamp add constraint matchstat_champ_to_act
    foreign key (actsID) references Act(id);

commit;

```

Процедура:

```

create procedure update_champ(t integer)
as
declare variable home_sc int;
declare variable away_sc int;
declare variable home_id int;
declare variable away_id int;
declare variable i int;
begin
    for select Match_champ.id from Match_champ where Match_champ.tour=:t into :i
    do
        begin
            select Matchstat_champ.home_scored, Matchstat_champ.away_scored,
                Match_champ.home, Match_champ.away from Matchstat_champ,
Match_champ
                where Match_champ.id=:i and Matchstat_champ.matchid=Match_champ.id
                into :home_sc, :away_sc, :home_id, :away_id;
            if(home_sc>away_sc) then
                begin
                    /*обновление статистики для победившего*/
                    update Championship set goalsFor=goalsFor+:home_sc
                        where clubid = :home_id;
                    update Championship set goalsAgainst=goalsAgainst+:away_sc
                        where clubid = :home_id;
                    update Championship set played=played+1 where clubid = :home_id
                        and clubid = :away_id;
                    update Championship set won=won+1 where clubid = :home_id;
                    update Championship set points=points+3 where clubid = :home_id;

                    /*обновление статистики для проигравшего*/
                    update Championship set goalsFor=goalsFor+:away_sc
                        where clubid = :away_id;
                    update Championship set goalsAgainst=goalsAgainst+:home_sc
                        where clubid = :away_id;
                    update Championship set lost=lost+1 where clubid = :away_id;
                end

            if(home_sc<away_sc) then
                begin
                    /*обновление статистики для победившего*/
                    update Championship set goalsFor=goalsFor+:away_sc

```

```

        where clubid = :away_id;
        update Championship set goalsAgainst=goalsAgainst+:home_sc
        where clubid = :away_id;
        update Championship set played=played+1 where clubid = :home_id
        and clubid = :away_id;
        update Championship set won=won+1 where clubid = :away_id;
        update Championship set points=points+3 where clubid = :away_id;

        /*обновление статистики для проигравшего*/
        update Championship set goalsFor=goalsFor+:home_sc
        where clubid = :home_id;
        update Championship set goalsAgainst=goalsAgainst+:away_sc
        where clubid = :home_id;
        update Championship set lost=lost+1 where clubid = :home_id;
    end

    if(home_sc=away_sc) then
    begin
        update Championship set goalsFor=goalsFor+:home_sc
        where clubid = :home_id and clubid = :away_id;
        update Championship set goalsAgainst=goalsAgainst+:away_sc
        where clubid = :home_id and clubid = :away_id;
        update Championship set played=played+1 where clubid = :home_id
        and clubid = :away_id;
        update Championship set drawn=drawn+1 where clubid = :home_id
        and clubid = :away_id;
        update Championship set points=points+1 where clubid = :home_id
        and clubid = :away_id;
    end
    end
end;

```

4. Вывод

Во время работы изучены хранимые процедуры. Хранимые процедуры позволяют хранить какие-либо сложные запросы в БД и выполнять их на стороне сервера. Это позволяет увеличить скорость работы базы, т.к. серверы обладают большей производительностью. В хранимых процедурах могут быть входные и выходные параметры и локальные переменные, в которых могут производиться числовые вычисления и операции над символьными данными, результаты вычислений могут присваиваться переменным и параметрам. В хранимых процедурах могут выполняться стандартные операции с БД (как DDL, так и DML). Это расширяет возможности работы с базами данных.