

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Институт компьютерных наук и технологий
Кафедра Компьютерных Систем и Программных Технологий**

**Отчет по дисциплине
«Базы данных»**

Язык SQL-DML

**Работу выполнил студент группы №: 43501/3
Работу принял преподаватель: _____**

**Крутихин М.В.
Мяснов А.В.**

**Санкт-Петербург
2016 г.**

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

Ход работы.

1. Для каждого врача вычислить процент ошибочных диагнозов.

Ошибочным следует считать диагноз, если он отличается от последнего диагноза пациента.

Данная процедура заполняет таблицу информацией об:

- общем количестве диагнозов каждого врача
- количестве ложных диагнозов
- проценте ложных диагнозов

```
create procedure task1
as
/*Вспомогательные переменные для хранения данных*/
declare variable total_amount DOUBLE PRECISION;
declare variable false_amount DOUBLE PRECISION;
declare variable sol DOUBLE PRECISION;
declare variable help_id integer;
begin

/*Заполнение колонки общего количества диагнозов*/
FOR select "DOCTOR ID", COUNT ("DIAGNOSIS ID")
from DIAGNOSIS
group by "DOCTOR ID"
into :help_id, :total_amount
DO BEGIN
    update or insert into "FALSE DIAGNOSIS"
    (DOCTOR_ID, TOTAL_AMOUNT) values (:help_id, :total_amount);
END

/*Заполнение колонки с количеством ложных диагнозов*/
FOR select "DOCTOR ID", COUNT ("DIAGNOSIS ID")
from DIAGNOSIS where "FALSE DIAGNOSIS">0
group by "DOCTOR ID" into :help_id, :false_amount
DO BEGIN
```

```

update or insert into "FALSE DIAGNOS"
(DOCTOR_ID, FALSE_AMOUNT) values (:help_id, :FALSE_AMOUNT);
END

/*Вычисление процента ложных диагнозов для каждого врача*/
FOR select "DOCTOR ID", TOTAL_AMOUNT, FALSE_AMOUNT from
"FALSE DIAGNOS"
into :help_id, :total_amount, :false_amount
DO BEGIN
:sol = (:false_amount / :total_amount) * 100;
update or insert into "FALSE DIAGNOS"
(DOCTOR_ID, TOTAL_AMOUNT, FALSE_AMOUNT, PERSENTS_FALSE )
values (:help_id, :total_amount, :FALSE_AMOUNT, :sol);
END
END

```

Пример результата:

DOCTOR...	TOTAL_AMOUNT	FALSE_AMOUNT	PERSENTS_FALSE
1	12,000	10,000	83,333
2	6,000	6,000	100,000
3	11,000	10,000	90,909
4	8,000	8,000	100,000
5	15,000	12,000	80,000
6	4,000	3,000	75,000
7	15,000	11,000	73,333
8	8,000	4,000	50,000
9	12,000	10,000	83,333
10	8,000	6,000	75,000
11	10,000	5,000	50,000
12	9,000	9,000	100,000
13	11,000	6,000	54,545
14	7,000	6,000	85,714

2. Для каждого диагноза вычислить долю пациентов, поступающих повторно и вылечивающихся за одно поступление.

В ходе выполнения данного задания выяснилось, что в разрабатываемой базе данных есть недоработка – отсутствует таблица справочник с информацией по каждому конкретному виду диагноза.

Скрипт, добавляющий данную таблицу:

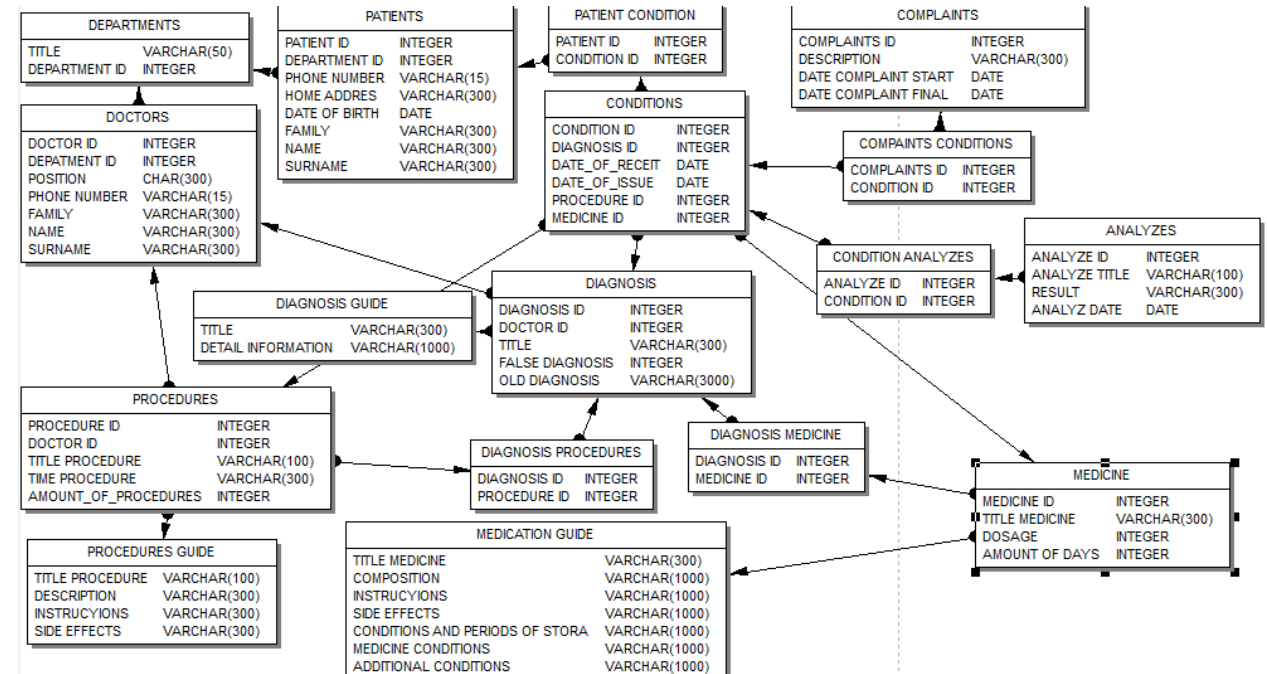
```

CREATE TABLE "DIAGNOSIS GUIDE" (
    TITLE          VARCHAR(300) NOT NULL,
    "DETAIL INFORMATION" VARCHAR(1000)
);
ALTER TABLE "DIAGNOSIS GUIDE" ADD CONSTRAINT "PK_DIAGNOSIS GUIDE"
PRIMARY KEY (TITLE);
ALTER TABLE DIAGNOSIS ADD CONSTRAINT FK_DIAGNOSIS_2 FOREIGN KEY

```

(TITLE) REFERENCES "DIAGNOSIS GUIDE" (TITLE);

С учетом этих изменений ER-диаграмма будет выглядеть следующим образом:



Код процедуры:

```
create procedure task2
as
/*Вспомогательные переменные для хранения данных*/
declare variable total_amount DOUBLE PRECISION;
declare variable false_amount DOUBLE PRECISION;
declare variable true_amount DOUBLE PRECISION;
declare variable sol_first DOUBLE PRECISION;
declare variable sol_second DOUBLE PRECISION;
declare variable TITLE varchar (300);
declare variable help integer;
begin
/* Цикл, в котором вычисляются характеристики отдельно для каждого
диагноза */
FOR select title
from "DIAGNOSIS GUIDE"
order by title
into :TITLE
DO BEGIN
```

```

/* Вычисление общего количества пациентов для каждого диагноза */
FOR select "PATIENT ID", count ("CONDITION ID") from "PATIENT
CONDITION"
where "CONDITION ID" in
(select "CONDITION ID" from CONDITIONS where "DIAGNOSIS ID" in
(select "DIAGNOSIS ID" from DIAGNOSIS where TITLE in
(select TITLE from "DIAGNOSIS GUIDE" where TITLE like :TITLE)
)) group by "PATIENT ID"
into :help, :total_amount
DO BEGIN
update or insert into HELP_TABLE
  (ID , "COUNT" ) values (:help, :total_amount);
END

```

```

select count (*) from HELP_TABLE into :total_amount ;
delete from HELP_TABLE;

```

```

/* Вычисление количества пациентов, вылечивающихся за одно поступление,
для каждого диагноза */
FOR select "PATIENT ID", count ("CONDITION ID") from "PATIENT
CONDITION"
where "CONDITION ID" in
(select "CONDITION ID" from CONDITIONS where "DIAGNOSIS ID" in
(select "DIAGNOSIS ID" from DIAGNOSIS where TITLE in
(select TITLE from "DIAGNOSIS GUIDE" where TITLE like :TITLE)
)) group by "PATIENT ID" having count ("CONDITION ID")=1
into :help, :true_amount
DO BEGIN
update or insert into HELP_TABLE
  (ID , "COUNT" ) values (:help, :true_amount);
END

```

```

select count (*) from HELP_TABLE into :true_amount ;
delete from HELP_TABLE;

```

```

/* Вычисление количества пациентов, поступающих повторно */
FOR select "PATIENT ID", count ("CONDITION ID") from "PATIENT
CONDITION"
where "CONDITION ID" in
(select "CONDITION ID" from CONDITIONS where "DIAGNOSIS ID" in
(select "DIAGNOSIS ID" from DIAGNOSIS where TITLE in
(select TITLE from "DIAGNOSIS GUIDE" where TITLE like :TITLE)
)) group by "PATIENT ID" having count ("CONDITION ID")>1
into :help, :false_amount
DO BEGIN

```

```

update or insert into HELP_TABLE
  (ID , "COUNT" ) values (:help, :false_amount);
END

select count (*) from HELP_TABLE into :false_amount ;
delete from HELP_TABLE;

/*Вычисление необходимых процентных соотношений*/
:sol_first = (:false_amount / :total_amount) * 100;
:sol_second = (:true_amount / :total_amount) * 100;

update or insert into STATISTIC_DIAGNOS
  (TITLE, total_amount,true_amount, FALSE_AMOUNT, PERSENTS_FIRST,
  PERSENTS_SECOND )
  values (:TITLE, :total_amount, :true_amount, :false_amount, :sol_first,
:sol_second);

END
END

```

Пример результата:

TITLE	TOTAL_AMOUNT	TRUE_AMOUNT	FALSE_AMOUNT	PERSENTS_FIRST	PERSENTS_SECOND
Анемии, связанные с питанием	331,000	331,000	0,000	0,000	100,000
Болезни наружного уха	454,000	452,000	2,000	0,441	99,559
▶ Болезни сосудистой оболочки и сетчатки	401,000	401,000	0,000	0,000	100,000
Болезни стекловидного тела и глазного яблока	406,000	405,000	1,000	0,246	99,754
Болезнь, вызванная вирусом иммунодефицита...	401,000	401,000	0,000	0,000	100,000
Вирусные лихорадки, передаваемые членист...	394,000	393,000	1,000	0,254	99,746
Вирусный гепатит	406,000	406,000	0,000	0,000	100,000
Гельминтозы	377,000	377,000	0,000	0,000	100,000
Гемолитические анемии	376,000	376,000	0,000	0,000	100,000
Глаукома	404,000	402,000	2,000	0,495	99,505
Грипп и пневмония	396,000	396,000	0,000	0,000	100,000
Невротические, связанные со стрессом и сома...	414,000	414,000	0,000	0,000	100,000
Некоторые бактериальные зоонозы	386,000	386,000	0,000	0,000	100,000

Вывод:

В ходе данной работы были получены навыки работы с хранимыми процедурами. ХП дают возможность значительно ускорить процесс работы с базами данных, за счет того, что позволяют сохранить часто используемые однотипные операции сложной выборки данных из базы.

Кроме того, использование ХП позволяет повысить безопасность, путем предоставления пользователю доступа только к ним, а не непосредственно к самим таблицам базы данных.