**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Институт компьютерных наук и технологий**

**Кафедра Компьютерных Систем и Программных Технологий**

**Отчет по дисциплине**

**«Базы данных»**

[**Язык SQL-DML**](http://tiger.ftk.spbstu.ru/trac/edu-db-2016/ticket/33)

**Работу выполнил студент группы №:** 43501/3 Крутихин М.В.

**Работу принял преподаватель:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Мяснов А.В.

**Санкт-Петербург**

**2016 г.**

**Программа работы**

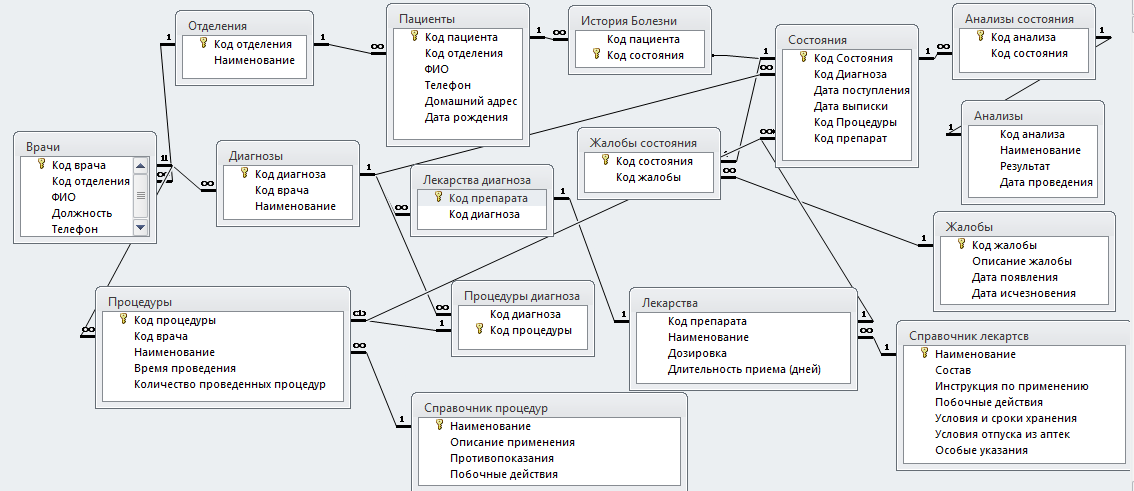
1. Изучите SQL-DML
2. Выполните все запросы из списка стандартных запросов. Продемонстрируйте результаты преподавателю.
3. Получите у преподавателя и реализуйте SQL-запросы в соответствии с индивидуальным заданием. Продемонстрируйте результаты преподавателю.
4. Выполненные запросы SELECT сохраните в БД в виде представлений, запросы INSERT, UPDATE или DELETE -- в виде ХП. Выложите скрипт в Subversion.

## Список стандартных запросов

* Сделайте выборку всех данных из каждой таблицы
* Сделайте выборку данных из одной таблицы при нескольких условиях, с использованием логических операций, LIKE, BETWEEN, IN (не менее 3-х разных примеров)
* Создайте в запросе вычисляемое поле
* Сделайте выборку всех данных с сортировкой по нескольким полям
* Создайте запрос, вычисляющий несколько совокупных характеристик таблиц
* Сделайте выборку данных из связанных таблиц (не менее двух примеров)
* Создайте запрос, рассчитывающий совокупную характеристику с использованием группировки, наложите ограничение на результат группировки
* Придумайте и реализуйте пример использования вложенного запроса
* С помощью оператора INSERT добавьте в каждую таблицу по одной записи
* С помощью оператора UPDATE измените значения нескольких полей у всех записей, отвечающих заданному условию
* С помощью оператора DELETE удалите запись, имеющую максимальное (минимальное) значение некоторой совокупной характеристики
* С помощью оператора DELETE удалите записи в главной таблице, на которые не ссылается подчиненная таблица (используя вложенный запрос)

**Ход работы:**

По итогам предыдущей работы имеется следующая SQL-схема:

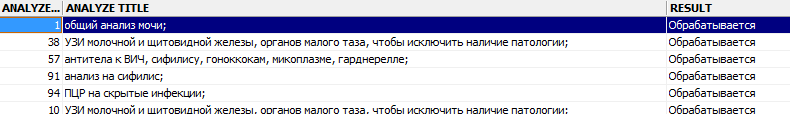


*Рис1. SQL-схема БД*

**1. Сделайте выборку всех данных из каждой таблицы**

|  |
| --- |
| select \* from analyzes;  select \* from "CONDITION ANALYZES";  select \* from conditions;  select \* from DEPARTMENTS;  select \* from DIAGNOSIS;  select \* from "DIAGNOSIS MEDICINE";  select \* from "DIAGNOSIS PROCEDURES";  select \* from doctors;  select \* from "MEDICATION GUIDE";  select \* from medicine;  select \* from "PATIENT CONDITION";  select \* from patients;  select \* from procedures; |

Пример выборки из таблицы «анализы»:

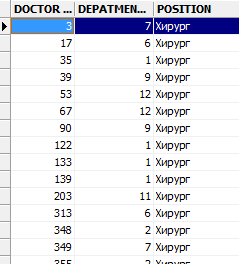


**2. Сделайте выборку данных из одной таблицы при нескольких условиях, с использованием логических операций, LIKE, BETWEEN, IN (не менее 3-х разных примеров)**

Выбор всех хирургов из таблицы Doctors^

|  |
| --- |
| select \* from DOCTORS where "POSITION" like 'Хирург%'; |

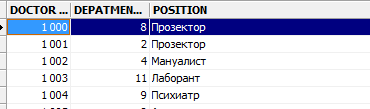
Пример выборки:



Выбор все докторов, идентификационные номера, которых заданы в следующем интервале:

|  |
| --- |
| select \* from doctors where "DOCTOR ID" between 1000 and 1150; |

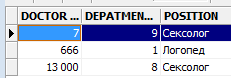
Пример выборки:



Выбор все докторов, идентификационные номера, которых соответствуют указанным:

|  |
| --- |
| select \* from doctors where "DOCTOR ID" IN (7, 666, 13000); |

Пример выборки:

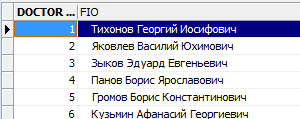


**3. Создайте в запросе вычисляемое поле**

Объединим инициалы в одно поле:

|  |
| --- |
| select "DOCTOR ID", (FAMILY || ' ' || NAME || ' ' || SURNAME) AS FIO from doctors; |

Пример выборки:

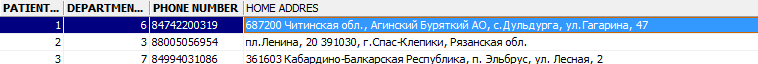


**4. Сделайте выборку всех данных с сортировкой по нескольким полям**

Сортировка пациентов по идентификационному номеру и дате рождения:

|  |
| --- |
| select \* from PATIENTS ordered by "PATIENT ID", "DATE OF BIRTH"; |

Пример выборки:



**5. Создайте запрос, вычисляющий несколько совокупных характеристик таблиц**

Запрос, определяющий первую по алфавиту фамилию пациента, а так же количество пациентов :

|  |
| --- |
| select Count(\*), MIN(FAMILY) from PATIENTS; |

Результат:

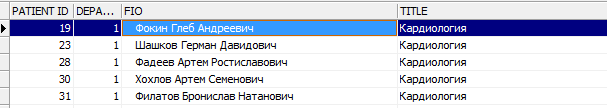


**6. Сделайте выборку данных из связанных таблиц (не менее двух примеров)**

Вывод всех пациентов, вместе с информацией об отделениях, в которых они проходят лечение:

|  |
| --- |
| select patients."PATIENT ID", patients."DEPARTMENT ID", (patients.family || ' ' || patients.name || ' ' || patients.surname ) as fio, DEPARTMENTS.TITLE from patients full join DEPARTMENTS on patients."DEPARTMENT ID" = DEPARTMENTS."DEPARTMENT ID"; |

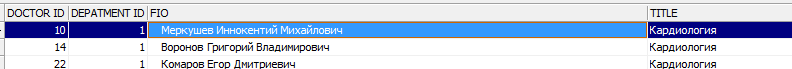
Результат:



Аналогичный запрос для докторов:

|  |
| --- |
| select doctors."DOCTOR ID", doctors."DEPATMENT ID", (doctors.family || ' ' || doctors.name || ' ' || doctors.surname ) fio,departments.title from doctors full join DEPARTMENTS on doctors."DEPATMENT ID" = DEPARTMENTS."DEPARTMENT ID"; |

Результат:



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

Выборка лекарств с дозировкой больше, чем 1:

|  |
| --- |
| select DOSAGE, COUNT (DOSAGE) from MEDICINE GROUP BY DOSAGE having COUNT (DOSAGE)>1; |

Результат:

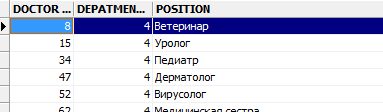


**8. Придумайте и реализуйте пример использования вложенного запроса**

Выбор всех врачей из приемного отделения:

|  |
| --- |
| select \* from DOCTORS where "DEPATMENT ID" = (select "DEPARTMENT ID" from DEPARTMENTS where TITLE like 'Приемное отделение%'); |

Результат:



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

|  |
| --- |
| Insert into DEPARTMENTS (TITLE, "DEPARTMENT ID") values (13, ‘Ортопедия’);  Insert into PATIENTS ("PATIENT ID", "DEPARTMENT ID", "PHONE NUMBER" ,"HOME ADDRES", "DATE OF BIRTH", FAMILY, NAME, SURNAME) values (87, 5, 84950400159, ‘186930, г.Костомукша, ул. Советская, д. 16, а/я 73’, ‘Харитонов’, ‘Глеб’, ‘Денисович’);  Insert into DOCTORS ("DOCTOR ID", "DEPATMENT ID", "POSITION", "PHONE NUMBER", FAMILY, NAME, SURNAME) values (33, 19, ‘Травматолог’, 84994031086, ‘Иванов’, ‘Глеб’, ‘Богданович’);  Insert into DIAGNOSIS ("DIAGNOSIS ID", "DOCTOR ID", TITLE) values (77, 195, ‘Грипп и пневмония’);  Insert into “PATIENT CONDITION” ("PATIENT ID", "CONDITION ID") values (5, 22778);  Insert into CONDITIONS ("CONDITION ID", "DIAGNOSIS ID", DATE\_OF\_RECEIT , DATE\_OF\_ISSUE, "PROCEDURE ID", "MEDICINE ID" ) values (1, 195, ‘19.03.2014', ‘20.04.2014’, 17, 85 );  Insert into “COMPAINTS CONDITIONS” ("COMPLAINTS ID", "CONDITION ID") values (17, 1);  Insert into COMPAINTS ("COMPLAINTS ID", DESCRIPTION, "DATE COMPLAINT START", "DATE COMPLAINT FINAL") values (17, ‘боль в груди’, ‘19.03.2014’, ‘20.04.2014’);  Insert into CONDITION ANALYZES ("ANALYZE ID", "CONDITION ID") values (18, 1);  Insert into ANALYZES ("ANALYZE ID", "ANALYZE TITLE", RESULT, "ANALYZ DATE") values (18, ‘Анализ крови’, ‘Отрицательный’, ‘20.04.2014’);  Insert into DIAGNOSIS MEDICINE ("DIAGNOSIS ID", "MEDICINE ID") values (77, 13);  Insert into MEDICINE ("MEDICINE ID", "TITLE MEDICINE", DOSAGE, "AMOUNT OF DAYS") values (13, ‘Нафтизин’, 5, 7);  Insert into DIAGNOSIS PROCEDURES ("DIAGNOSIS ID", "PROCEDURE ID") values (18, 35);  Insert into PROCEDURES ("PROCEDURE ID", "DOCTOR ID", "TITLE PROCEDURE", "TIME PROCEDURE", AMOUNT\_OF\_PROCEDURES) values (35, 19, ‘Промывание’, ’13-15’, 8); |

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

Единичный инкремент дозировки каждого лекарства:

|  |
| --- |
| update MEDICINE set DOSAGE=DOSAGE+1; |

**11. С помощью оператора DELETE удалите запись, имеющую максимальное (минимальное) значение некоторой совокупной характеристики**

Удаление записи врача с максимальным идентификационным номером:

|  |
| --- |
| delete from doctors where "DOCTOR ID" = (select MAX("DOCTOR ID") from doctors); |

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

|  |
| --- |
| delete from DEPARTMENTS where "DEPARTMENT ID" IN (select "DEPARTMENT ID" from DEPARTMENTS where "DEPARTMENT ID" not in (select "DEPATMENT ID" from DOCTORS)); |

**Выполнение индивидуального задания.**

Формулировка задания:

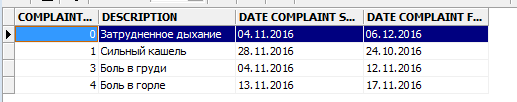
1. Вывести наиболее частые диагнозы для каждого вида жалоб. Виды жалоб ввести отдельной таблицей-справочником.
2. Вывести пациентов, у которых менялись диагнозы.
3. Вывести наиболее частый диагноз для каждого отделения.

**1. Вывести наиболее частые диагнозы для каждого вида жалоб. Виды жалоб ввести отдельной таблицей-справочником.**

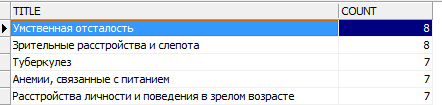
Запрос представленный ниже выводит 5 самых часто встречающихся диагнозов для заданного типа жалобы. Так же выводится количество этих диагнозов.

|  |
| --- |
| select first 5 TITLE, count (TITLE) from DIAGNOSIS where "DIAGNOSIS ID" in (select "DIAGNOSIS ID" from CONDITIONS where "CONDITION ID" in (select "CONDITION ID" from "COMPAINTS CONDITIONS" where "COMPLAINTS ID" = (select "COMPLAINTS ID" from COMPLAINTS where DESCRIPTION like 'Затрудненное дыхание%'))) group by TITLE order by count (TITLE) Desc; |

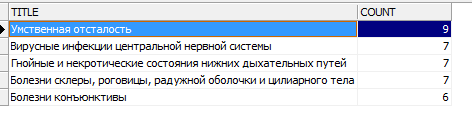
На момент выполнения этого пункта, в справочнике жалоб находились следующие значения:



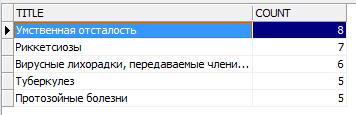
Пример выполнения для жалобы 0:



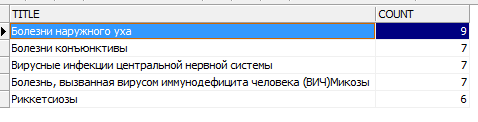
Пример выполнения для жалобы 1:



Пример выполнения для жалобы 2:



Пример выполнения для жалобы 3:

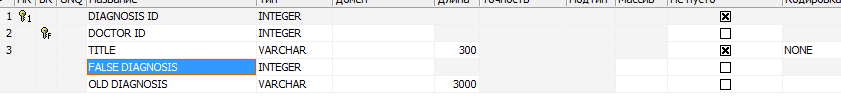


**2. Вывести пациентов, у которых менялись диагнозы.**

Во время выполнения данного пункта работы, выяснилось, что в БД не предусмотрена возможность отследить возможную замену одного диагноза на другой. Поэтому в таблицу были добавлены два поля для реализации данной возможности:

FALSE DIAGNOSIS – может быть пустым, показывает, сколько раз переписывался диагноз.

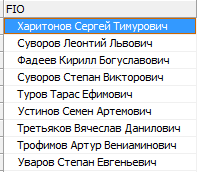
OLD DIAGNOSIS – поле, в которое записываются все сведения о «старых» диагнозах.



Запрос:

|  |
| --- |
| select (FAMILY || ' ' || NAME || ' ' || SURNAME) AS FIO from PATIENTS where "PATIENT ID" in (select "PATIENT ID" from "PATIENT CONDITION" where "CONDITION ID" in (select "CONDITION ID" from CONDITIONS where "DIAGNOSIS ID" in(select "DIAGNOSIS ID" from DIAGNOSIS where "FALSE DIAGNOSIS">2))); |

Результат:



**3. Вывести наиболее частый диагноз для каждого отделения.**

Запрос:

|  |
| --- |
| select first 1 TITLE,count (TITLE) from DIAGNOSIS where "DOCTOR ID" in (select "DOCTOR ID" from DOCTORS where "DEPATMENT ID" in  (select "DEPARTMENT ID" from DEPARTMENTS where TITLE like 'Кардиология%'))group by TITLE order by count (TITLE) Desc; |

Пример результата:



**Вывод:**

В ходе данной работы были изучены особенности языка SQL-DML, который позволяет производить различные операции над данным. Был получен опыт работы с такими операторами, как insert, select, update, delete, join, where, order by. Использование данных операторов позволяет производить объединение таблиц, создание вложенных запросов группировку данных и т.д.