Санкт- Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Отчет по лабораторной работе
Курс: «Базы данных»
Тема: «Язык SQL-DML»

Выполнил:
Бояркин Н.С. группа 43501/3
Проверил:
Мяснов А.В.

Санкт – Петербург
2017

1. Цель 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.

2. Стандартные запросы

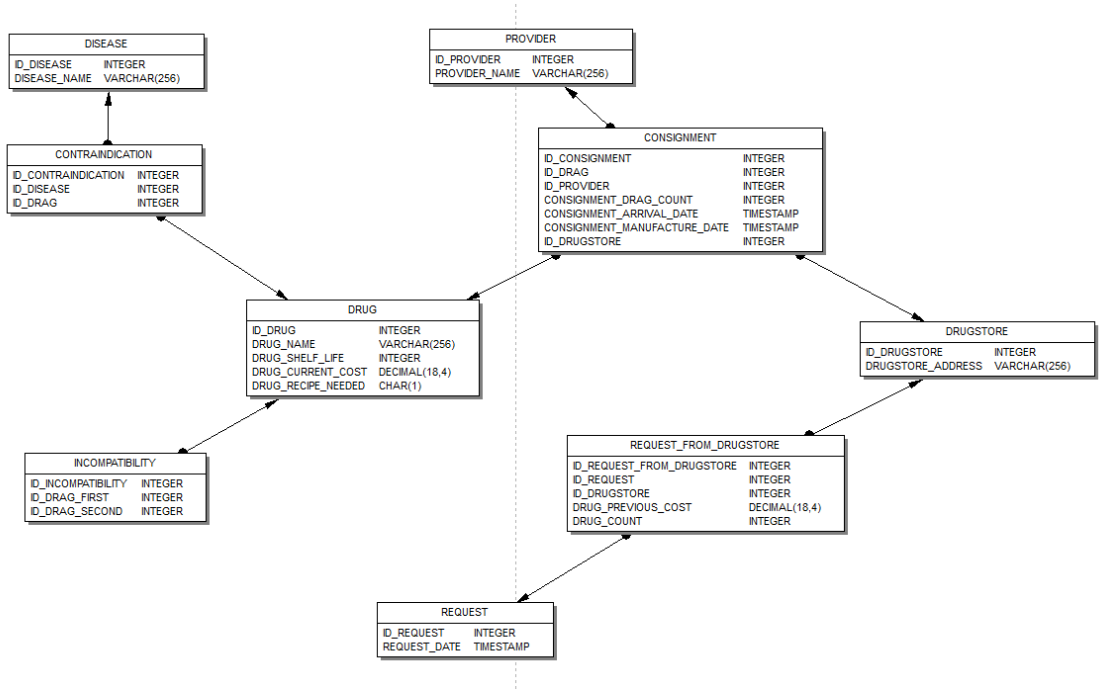
1. Сделайте выборку всех данных из каждой таблицы
2. Сделайте выборку данных из одной таблицы при нескольких условиях, с использованием логических операций, LIKE, BETWEEN, IN (не менее 3-х разных примеров)
3. Создайте в запросе вычисляемое поле
4. Сделайте выборку всех данных с сортировкой по нескольким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

3. Индивидуальные запросы

1. Вывести диагнозы, при которых назначаются препараты, которые являются несовместимыми.
2. Вывести 5 клиентов, совершивших закупки на максимальную сумму за заданный период.
3. Вывести 10 складов, на которые чаще всего приходит большая часть поставки лекарства.

4. Ход работы

SQL-Диаграмма



Выборка данных из каждой таблицы

```
SELECT * FROM CONSIGNMENT;
SELECT * FROM CONTRAINDICATION;
SELECT * FROM DISEASE;
SELECT * FROM DRUG;
SELECT * FROM DRUGSTORE;
SELECT * FROM INCOMPATIBILITY;
SELECT * FROM PROVIDER;
SELECT * FROM REQUEST;
SELECT * FROM REQUEST_FROM_DRUGSTORE;
```

Выборка данных из одной таблицы при нескольких условиях, с использованием логических операций, LIKE, BETWEEN, IN

Получение строк таблицы поставок, в которых количество лекарств равно 100:

```
CREATE VIEW S2P1 AS
    SELECT * FROM CONSIGNMENT
    WHERE CONSIGNMENT.CONSIGNMENT_DRAG_COUNT LIKE 100;

COMMIT;
```

Получение строк таблицы лекарств, в которых срок годности лекарств от 10 до 20 месяцев:

```
CREATE VIEW S2P2 AS
    SELECT * FROM DRUG
    WHERE DRUG.DRUG_SHELF_LIFE BETWEEN 10 AND 20;

COMMIT;
```

Получение строк таблицы запросов, в которых количество лекарств 15, 20, 35 или 22:

```
CREATE VIEW S2P3 AS
    SELECT * FROM REQUEST_FROM_DRUGSTORE
    WHERE REQUEST_FROM_DRUGSTORE.DRUG_COUNT IN (15, 20, 35, 22);

COMMIT;
```

Вычисляемое поле

Получение цены лекарства с 10% скидкой:

```
CREATE VIEW S3P1 AS

  SELECT DRUG.ID_DRUG, DRUG.DRUG_NAME, (DRUG.DRUG_CURRENT_COST * 9 / 10) AS DISCOUNT_PRICE FROM D
RUG;

COMMIT;
```

Вывод строки с префиксом:

```
CREATE VIEW S3P2 AS
    SELECT ('Prefix: ' || DRUG.DRUG_NAME) FROM DRUG;

COMMIT;
```

Выборка данных с сортировкой по нескольким полям

Вывод таблицы поставок с сортировкой по дате прибытия и изготовления:

```
CREATE VIEW S4P1 AS

  SELECT * FROM CONSIGNMENT ORDER BY CONSIGNMENT.CONSIGNMENT_ARRIVAL_DATE, CONSIGNMENT.CONSIGNMEN
T_MANUFACTURE_DATE;

COMMIT;
```

Вывод таблицы лекарств с сортировкой по названию и сроку годности:

```
CREATE VIEW S4P1 AS
    SELECT * FROM DRUG ORDER BY DRUG.DRUG_NAME, DRUG.DRUG_SHELF_LIFE;

COMMIT;
```

Вычисление нескольких совокупных характеристик таблиц

Вывод минимального, среднего и максимального срока годности лекарств:

```
CREATE VIEW S5 AS

  SELECT MIN(DRUG.DRUG_SHELF_LIFE), AVG(DRUG.DRUG_SHELF_LIFE), MAX(DRUG.DRUG_SHELF_LIFE) FROM DRU
G;

COMMIT;
```

Выборка данных из связанных таблиц

Вывод противопоказаний:

```
CREATE VIEW S6P1 AS
    SELECT ('Contraindication: ' || DISEASE.DISEASE_NAME || ' ' || DRUG.DRUG_NAME) FROM DISEASE
    JOIN CONTRAINDICATION ON DISEASE.ID_DISEASE = CONTRAINDICATION.ID_DISEASE
    JOIN DRUG ON CONTRAINDICATION.ID_DRAG = DRUG.ID_DRUG;

COMMIT;
```

Вывод количества лекарств, поставленных каждым поставщиком:

```
CREATE VIEW S6P2 AS
    SELECT PROVIDER.PROVIDER_NAME, SUM(CONSIGNMENT.CONSIGNMENT_DRAG_COUNT) FROM PROVIDER
    JOIN CONSIGNMENT ON PROVIDER.ID_PROVIDER = CONSIGNMENT.ID_PROVIDER
    GROUP BY PROVIDER.PROVIDER_NAME;

COMMIT;
```

Расчёт характеристики с использованием группировки, наложение ограничения на результат группировки

Вывод количества лекарств с одинаковым названием:

```
CREATE VIEW S7 AS
```

```
    SELECT DRUG.DRUG_NAME, COUNT(DRUG.DRUG_NAME) AS NAMES_COUNT FROM DRUG
    GROUP BY DRUG.DRUG_NAME
    ORDER BY NAMES_COUNT DESC;

COMMIT;
```

Реализация вложенного запроса

Вывод строк таблицы запросов с определенной датой:

```
CREATE VIEW S8 AS
    SELECT * FROM REQUEST

  WHERE REQUEST.ID_REQUEST = (SELECT REQUEST.ID_REQUEST FROM REQUEST WHERE REQUEST.REQUEST_DATE =
 '09.05.2014 0:15:01');

COMMIT;
```

Добавление в каждую таблицу по одной записи с помощью INSERT

```
CREATE PROCEDURE S9 AS
BEGIN
    INSERT INTO DISEASE (ID_DISEASE, DISEASE_NAME)
    VALUES (100001, 'ОРЗ');
    COMMIT;

    INSERT INTO DRUG (ID_DRUG, DRUG_NAME, DRUG_SHELF_LIFE, DRUG_CURRENT_COST, DRUG_RECIPE_NEEDED)
    VALUES (100001, 'Супрадин', 30, 25.0, 0);
    COMMIT;

    INSERT INTO INCOMPATIBILITY (ID_INCOMPATIBILITY,ID_DRAG_FIRST, ID_DRAG_SECOND)
    VALUES (100001, 3, 4);
    COMMIT;

    INSERT INTO CONTRAINDICATION (ID_CONTRAINDICATION,ID_DISEASE, ID_DRAG)
    VALUES (100001, 2, 5);
    COMMIT;

    INSERT INTO PROVIDER (ID_PROVIDER,PROVIDER_NAME)
    VALUES (100001, 'Докторсилач');
    COMMIT;

    INSERT INTO DRUGSTORE (ID_DRUGSTORE,DRUGSTORE_ADDRESS)
    VALUES (100001, 'Санкт-Петербург, ул. Парашютная 12к1');
    COMMIT;


  INSERT INTO CONSIGNMENT (ID_CONSIGNMENT,ID_DRAG, ID_PROVIDER, CONSIGNMENT_DRAG_COUNT, CONSIGNME
NT_ARRIVAL_DATE, CONSIGNMENT_MANUFACTURE_DATE, ID_DRUGSTORE)
    VALUES (100001, 8, 3, 1000, '02-10-2011', '02-09-2011', 4);
    COMMIT;

    INSERT INTO REQUEST (ID_REQUEST,REQUEST_DATE)
    VALUES (100001, '03-02-2011');
    COMMIT;


  INSERT INTO REQUEST_FROM_DRUGSTORE (ID_REQUEST_FROM_DRUGSTORE,ID_REQUEST, ID_DRUGSTORE, DRUG_PR
EVIOUS_COST, DRUG_COUNT)
    VALUES (100001, 8, 4, 10.0, 40);
    COMMIT;
END;
```

Изменение значений нескольких полей у всех записей, отвечающих заданному условию с помощью UPDATE

Увеличение срока годности всех лекарств на один месяц:

```
CREATE PROCEDURE S10 AS
BEGIN
    UPDATE DRUG SET DRUG.DRUG_SHELF_LIFE = DRUG.DRUG_SHELF_LIFE + 1;
    COMMIT;
END;
```

Удаление записи, имеющей максимальное (минимальное) значение некоторой совокупной характеристики

Создание лекарства с максимальной ценой и его удаление, с помощью вложенного запроса:

```
CREATE PROCEDURE S11 AS
BEGIN
    INSERT INTO DRUG (ID_DRUG, DRUG_NAME, DRUG_SHELF_LIFE, DRUG_CURRENT_COST, DRUG_RECIPE_NEEDED)
    VALUES (100001, 'Аспирин', 9999999.0, 9999999.0, 1);
    COMMIT;

    DELETE FROM DRUG
    WHERE DRUG.DRUG_CURRENT_COST = (SELECT MAX(DRUG.DRUG_CURRENT_COST) FROM DRUG);
    COMMIT;
END;
```

Удаление записей в главной таблице, на которые не ссылается подчиненная таблица (используя вложенный запрос)

Удаление записей в таблице лекарств, на которые не ссылается ни одна подчиненная таблица (противопоказания, поставки, несовместимость лекарств):

```
CREATE PROCEDURE S12 AS
BEGIN
    DELETE FROM DRUG
    WHERE DRUG.ID_DRUG IN
    (
        SELECT DRUG.ID_DRAG FROM DRUG
        WHERE DRUG.ID_DRUG NOT IN (SELECT CONTRAINDICATION.ID_DRAG FROM CONTRAINDICATION)
        AND DRUG.ID_DRUG NOT IN (SELECT CONSIGNMENT.ID_DRAG FROM CONSIGNMENT)
        AND DRUG.ID_DRUG NOT IN (SELECT INCOMPATIBILITY.ID_DRAG FROM INCOMPATIBILITY)
    );
    COMMIT;
END;
```

Вывод диагнозов, при которых назначаются препараты, которые являются несовместимыми

Ввиду того, что в базе данных отсутствует информации о показаниях лекарств для болезней (присутствует только информация о противопоказаниях), была создана еще одна таблица, которая реализует связь многие ко многим для таблиц лекарств и болезней:

```
CREATE TABLE INDICATION
(
    ID_INDICATION   INTEGER     NOT NULL,
    ID_DISEASE      INTEGER     NOT NULL REFERENCES DISEASE,
    ID_DRUG         INTEGER     NOT NULL REFERENCES DRUG,
    CONSTRAINT PK_INDICATION PRIMARY KEY (ID_INDICATION)
);

COMMIT;
```

Результирующий запрос:

```
CREATE VIEW I1 AS
    SELECT DISEASE.ID_DISEASE, DISEASE.DISEASE_NAME FROM DISEASE
    JOIN INDICATION ON DISEASE.ID_DISEASE = INDICATION.ID_DISEASE
    WHERE INDICATION.ID_DRUG IN (SELECT INCOMPATIBILITY.ID_DRAG_FIRST FROM INCOMPATIBILITY)
       OR INDICATION.ID_DRUG IN (SELECT INCOMPATIBILITY.ID_DRAG_SECOND FROM INCOMPATIBILITY)
    GROUP BY DISEASE.ID_DISEASE, DISEASE.DISEASE_NAME;
```

Вывод 5-и клиентов, совершивших закупки на максимальную сумму за заданный период

```
CREATE VIEW I2 AS

  SELECT FIRST 5 SKIP 0 REQUEST.ID_REQUEST, (REQUEST_FROM_DRUGSTORE.DRUG_PREVIOUS_COST * REQUEST_
FROM_DRUGSTORE.DRUG_COUNT) AS DRUG_WEIGHT FROM REQUEST
    JOIN REQUEST_FROM_DRUGSTORE ON REQUEST.ID_REQUEST = REQUEST_FROM_DRUGSTORE.ID_REQUEST
    WHERE REQUEST.REQUEST_DATE > '01.01.2013' AND REQUEST.REQUEST_DATE < '01.01.2014'
    ORDER BY DRUG_WEIGHT DESC;
```

Вывод 10-и складов, на которые чаще всего приходит большая часть поставки лекарства

```
CREATE VIEW I3 AS

  SELECT FIRST 10 SKIP 0 CONSIGNMENT.ID_DRUGSTORE, SUM(CONSIGNMENT.CONSIGNMENT_DRAG_COUNT) AS DRU
G_WEIGHT FROM CONSIGNMENT
    GROUP BY CONSIGNMENT.ID_DRUGSTORE
    ORDER BY DRUG_WEIGHT DESC;
```

5. Вывод

В ходе работы я познакомился с языком SQL-DML, который позволяет производить операции над данными, хранящимися в базу.

Операции языка SQL-DML, которые не влияют на данные в таблицах:

1. SELECT – выборка данных из таблицы.
2. JOIN – объединение таблиц для выборки.
3. WHERE – наложение условия на выборку.
4. ORDERBY – сортировка данных.
5. GROUPBY – группировка данных.

Операции языка SQL-DML, которые влияют на данные в таблицах:

1. INSERT–добавление записи в таблицу.
2. UPDATE – обновление данных, находящихся в таблице.
3. DELETE – удаление данных из таблицы.

Использование SQL-DMLоткрывает большие возможности по созданию запросов, а именно объединение таблиц, группировку данных, вложенные запросы, операторы условий и т. д.