

Санкт-Петербургский Политехнический Университет Петра Великого  
Институт компьютерных наук и технологий  
Кафедра компьютерных систем и программных технологий

## Базы данных

Отчет по лабораторной работе №4  
SQL-DML

**Работу выполнила:**

Темнова А.С.

Группа: 43501/3

**Преподаватель:**

Мяснов А.В.

Санкт-Петербург  
2016

# 1 Цель работы

Познакомить студентов с языком создания запросов управления данными SQL-DML.

# 2 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.

# 3 Список стандартных запросов

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)

# 4 Ход выполнения работы

## 4.1 Выполнение всех запросов из списка стандартных запросов

1. Сделайте выборку всех данных из каждой таблицы

```
1  SELECT * FROM BUYER;  
2  SELECT * FROM COLOR;  
3  SELECT * FROM COUNTRY;  
4  SELECT * FROM COUNTRY_PRODUCER;  
5  SELECT * FROM DELIVERY;  
6  SELECT * FROM PRODUCER;  
7  SELECT * FROM PRODUCER_WINE;  
8  SELECT * FROM PURCHASE;  
9  SELECT * FROM SHOP;  
10 SELECT * FROM STORAGE;  
11 SELECT * FROM SUPPLIER;  
12 SELECT * FROM TYPE;  
13 SELECT * FROM WINE;  
14
```

Листинг 1: SELECT

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

```
1 SELECT * FROM WINE WHERE NAME LIKE 'B%';
2 SELECT * FROM DELIVERY WHERE PRICE BETWEEN 1000 AND 2000;
3 SELECT * FROM DELIVERY WHERE PRICE IN (890, 900, 1604);
4
```

Листинг 2: SELECT LIKE/BETWEEN/IN

3. Создайте в запросе вычисляемое поле

Вычисляемым полем будет выдержка вина TIME\_EXPOSURE, которая вычисляется как разница текущего года и WINE\_YEAR.

```
1 SELECT WINE_ID, (2016 - WINE_YEAR) AS TIME_EXPOSURE FROM WINE;
2
```

Листинг 3: SELECT AS

4. Сделайте выборку всех данных с сортировкой по нескольким полям

```
1 SELECT * FROM WINE ORDER BY NAME, WINE_YEAR;
2 SELECT * FROM SHOP ORDER BY NAME, ADDRESS;
3
```

Листинг 4: SELECT ORDER

5. Создайте запрос, вычисляющий несколько совокупных характеристик таблиц

Вычисляем минимальный год изготовления вина и максимальный.

```
1 SELECT MIN(WINE_YEAR), MAX(WINE_YEAR) FROM WINE;
2
```

Листинг 5: SELECT MIN/MAX

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

```
1 /* Вино и цвет */
2 SELECT COLOR.NAME AS COLOR_NAME, WINE.NAME AS WINE_NAME FROM WINE LEFT JOIN COLOR ON
3   ↳ WINE.COLOR_ID = COLOR.COLOR_ID ORDER BY COLOR_NAME;
4
5 /* Дата поставки и поставщик */
6 SELECT DELIVERY.DATE_DELIVERY AS DATE_DELIVERY, SUPPLIER.SUPPLIER_ID AS SUPPLIER FROM
7   ↳ DELIVERY LEFT JOIN SUPPLIER ON DELIVERY.SUPPLIER_ID = SUPPLIER.SUPPLIER_ID ORDER
8   ↳ BY DATE_DELIVERY;
```

Листинг 6: JOIN

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

```
1 /*Поставщики, у которых широкий ассортимент вин */
2 SELECT SUPPLIER_ID, COUNT(I_PR_WI) AS WINE_NUMBER FROM DELIVERY GROUP BY SUPPLIER_ID
3   ↳ HAVING COUNT(ID_PR_WI) > 1;
4
5 /* Покупатели, которые приобрели более 100 единиц товара*/
6 SELECT BUYER_ID, SUM(NUMBER) AS NUMBER_OF_BOTTLES FROM PURCHASE GROUP BY BUYER_ID
7   ↳ HAVING SUM(NUMBER) > 100;
```

Листинг 7: COUNT

8. Придумайте и реализуйте пример использования вложенного запроса

Будем выводить год и название вина, которое является сладким. И отсортируем относительно года.

```
1 SELECT WINE_YEAR, NAME AS WINE_NAME FROM WINE WHERE TYPE_ID=( SELECT TYPE_ID FROM
2   ↳ TYPE WHERE NAME = 'sweet' ) ORDER BY WINE_YEAR;
```

Листинг 8: SELECT ( SELECT ... )

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

```

1  INSERT INTO COLOR (COLOR_ID, NAME) VALUES (4, 'blue');
2  INSERT INTO TYPE (TYPE_ID, NAME) VALUES (5, 'hot');
3  INSERT INTO WINE (WINE_ID, NAME, WINE_YEAR, COLOR_ID, TYPE_ID) VALUES (100008, 'Bla_
  ↳ bla_bla', 2016, 4, 5);
4  INSERT INTO SUPPLIER (SUPPLIER_ID, ADDRESS, PHONE) VALUES (100003, 'СПб_Невский_пр._
  ↳ д.15', 1111111);
5  INSERT INTO SHOP (SHOP_ID, ADDRESS, PHONE, NAME) VALUES (103, 'улКамышова._д.30_
  ↳ корп.1', 2034740, 'Леонардо_да_Винчик');
6  INSERT INTO PRODUCER (PRODUCER_ID, NAME, ADDRESS, PHONE) VALUES (108, 'Best_one', '
  ↳ гСПб._ул._Ленинградская_д.92', 7585027);
7  INSERT INTO COUNTRY (COUNTRY_ID, NAME) VALUES (100004, 'Russia');
8  INSERT INTO COUNTRY_PRODUCER (ID_CO_PR, COUNTRY_ID, PRODUCER_ID) VALUES
  ↳ (108,100004,17);
9  INSERT INTO PRODUCER_WINE (ID_PR_WI, PRODUCER_ID, WINE_ID) VALUES (108,108,1);
10 INSERT INTO DELIVERY (DELIVERY_ID, SUPPLIER_ID, ID_PR_WI, DATE_DELIVERY, NUMBER,
  ↳ PRICE) VALUES (108,100003,7,22.11.2016,15,3400);
11 INSERT INTO STORAGE (STORAGE_ID, SHOP_ID, DELIVERY_ID, NUMBER) VALUES (108,13,1,50);
12 INSERT INTO BUYER (BUYER_ID, NAME) VALUES (101, 'AA');
13 INSERT INTO PURCHASE (PURCHASE_ID, BUYER_ID, SHOP_ID, ID_PR_WI, DATE_BUY, NUMBER)
  ↳ VALUES (101,101,13,16,22.11.2016,1);
14

```

Листинг 9: INSERT

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

Увеличим цену на 100, где поставщик имеет ID = 1.

```

1  UPDATE DELIVERY SET PRICE=PRICE+100 WHERE SUPPLIER_ID=1;
2

```

Листинг 10: UPDATE

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

```

1  INSERT INTO WINE (WINE_ID, NAME, WINE_YEAR, COLOR_ID, TYPE_ID) VALUES (100009, 'Bla_
  ↳ bla_bla', 3016, 1, 1);
2
3  DELETE FROM WINE WHERE WINE_ID=( SELECT WINE_ID FROM WINE WHERE WINE_YEAR=( SELECT
  ↳ MAX(WINE_YEAR) FROM WINE ) );
4

```

Листинг 11: DELETE

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

```

1  DELETE FROM PRODUCER_WINE WHERE ID_PR_WI IN (
2  SELECT ID_PR_WI FROM PRODUCER_WINE WHERE ID_PR_WI NOT IN (
3  SELECT ID_PR_WI FROM PURCHASE) );
4

```

Листинг 12: DELETE

## 4.2 Индивидуальное задание:

1. Вывести 5 стран, которые производят больше всего вина заданного цвета.

```

1  /* Выбираем из таблицы вин вина нужного цвета, при этом присоединяем страну, имеющую
  ↳ производителей,
2  которые производят это вино. Это все группируем по стране и складываем колво— вин. Сортируем
  ↳ по убыванию.*/
3
4  SELECT FIRST 5 country.country_id, COUNT(wine.wine_id) AS wine_count
5  FROM wine
6  INNER JOIN country ON country.country_id IN (
7  SELECT country_producer.country_id FROM country_producer
8  WHERE country_producer.producer_id IN (
9  SELECT producer_wine.producer_id FROM producer_wine

```

```

10 WHERE producer_wine.wine_id = wine.wine_id))
11 WHERE wine.color_id=1
12 GROUP BY country.country_id
13 ORDER BY wine_count DESC
14

```

Листинг 13: indiv 1

2. Вывести 5 сортов вин, по которым увеличивались закупки.

```

1  /* Первое поле — это id вина из таблицы вин. Дальше мы берем сумму из delivery.number
   ↪ поставок нужного нам вина в даты меньше сегодня
2  и больше сегодня(-30 дней). От этого всего ISNULL ( то что получилось, 0) — если это будет
   ↪ null, то у нас будет 0. И берем то же самое
3  только от -60 до -30 дней. И так же от этого ISNULL. Потом из первого вычитаем второе и
   ↪ получаем на сколько за последние 30 дней
4  больше закупили, чем за предыдущие 30 дней. Это все сортируем по убыванию и берем первые 5
   ↪ записей. */
5
6  SELECT FIRST 5 wine.wine_id, (
7      ISNULL(( SELECT SUM(delivery.number) FROM delivery WHERE ( delivery.date_delivery <
   ↪ CURRENT_DATE
8      AND delivery.date_delivery > DATEADD (-30 DAY TO CURRENT_DATE ) )
9      AND delivery.id_pr_wi IN (
10     SELECT producer_wine.id_pr_wi FROM producer_wine WHERE producer_wine.wine_id=wine
   ↪ .wine_id ), 0) -
11     ISNULL(( SELECT SUM(delivery.number) FROM delivery WHERE ( delivery.date_delivery <
   ↪ DATEADD (-30 DAY TO CURRENT_DATE)
12     AND delivery.date_delivery > DATEADD (-60 DAY TO CURRENT_DATE ) )
13     AND delivery.id_pr_wi IN (
14     SELECT producer_wine.id_pr_wi FROM producer_wine WHERE producer_wine.wine_id=wine
   ↪ .wine_id ), 0))
15 AS delivery_sum FROM wine ORDER BY delivery_sum DESC
16

```

Листинг 14: indiv 2

3. Удалить неиспользуемые склады.

```

1  /* Удаляем склады, на которые никогда не поступали вина */
2
3  DELETE FROM storage
4  WHERE storage.delivery_id NOT IN (
5      SELECT delivery.delivery_id FROM delivery );
6

```

Листинг 15: indiv 3

## 5 Выводы

В ходе данной работы я изучила язык SQL-DML. С помощью него можно делать выборки данных из таблиц (SELECT) с условиями (LIKE, BETWEEN, IN) и без, создавать в запросе вычисляемые поля, делать выборки данных с сортировкой по нескольким полям (ORDER BY), создавать запросы, вычисляющие несколько совокупных характеристик таблиц (MIN, MAX, etc), делать выборки данных из связанных таблиц (JOIN, LEFT JOIN, RIGHT JOIN), создавать запросы, рассчитывающие совокупные характеристики, реализовывать вложенные запросы, добавлять в таблицы записи (INSERT), изменять значения полей у записей (UPDATE), удалять записи (DELETE).

Кроме стандартных запросов, были выполнены запросы в соответствии с индивидуальным заданием. Эти запросы в разы сложнее тех, что реализованы в первой части лабораторной работы. И на них было потрачено большинство времени.