

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

## Отчет по лабораторной работе

Дисциплина: «Базы данных»

Тема: «Язык SQL-DML»

**Работу выполнила:**

Мартюшева Надежда

Группа: 43501/3

**Преподаватель:**

Мяснов Александр Владимирович

Санкт-Петербург  
2017

## 1 Цели работы

Познакомиться с языком создания запросов управления данными SQL-DML.

## 2 Программа работы

1. Самостоятельное изучение 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 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).

## 4 Индивидуальное задание

1. Вывести 10 клубов, члены которых чаще всего занимали призовые места за всю историю наблюдений.
2. Вывести 10 танцоров, которые чаще всего участвовали в соревнованиях.
3. Вывести 10 мероприятий с наименьшим отношением суммарных спонсорских взносов к количеству участников.

## 5 Ход работы

SQL-диаграмма базы данных приведена на рисунке 1.

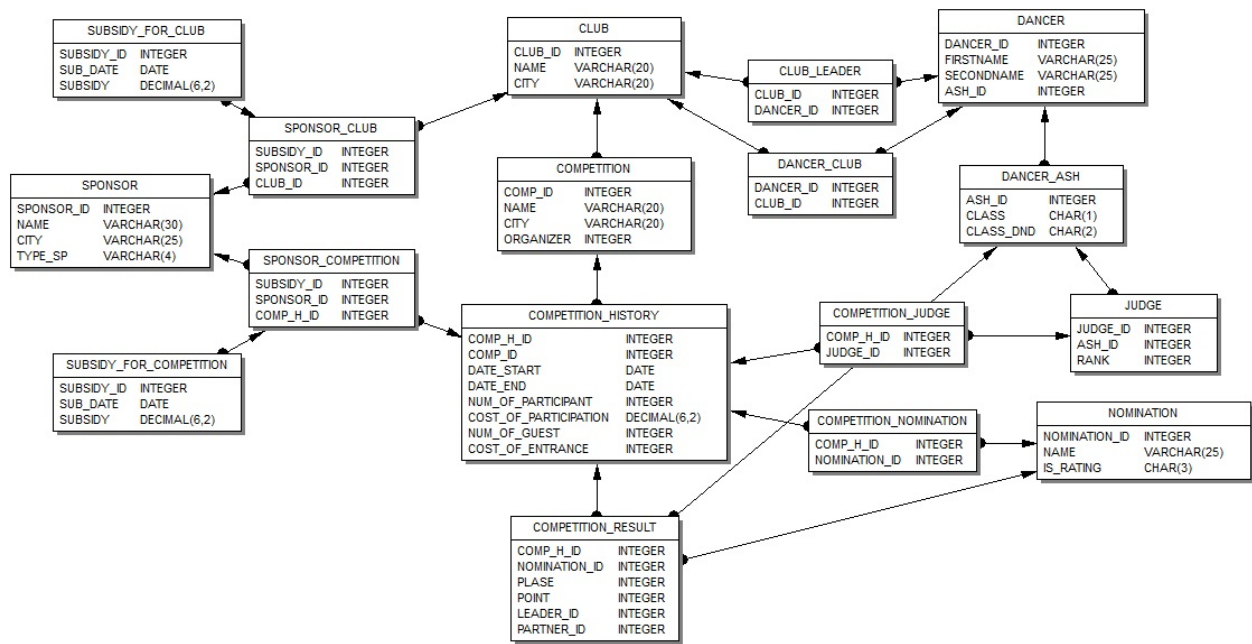


Рис. 1: SQL-диаграмма базы данных после внесения изменений

## 5.1 Выполнение стандартных запросов

Выборка всех данных из каждой таблицы

```

1 select * from dancer;
2 select * from dancer_club;
3 select * from dancer_ASH;
4 select * from judge;
5 select * from club;
6 select * from club_leader;
7 select * from competition;
8 select * from nomination;
9 select * from competition_history;
10 select * from competition_result;
11 select * from competition_nomination;
12 select * from competition_judge;
13 select * from sponsor;
14 select * from sponsor_club;
15 select * from subsidy_for_club;
16 select * from sponsor_competition;
17 select * from subsidy_for_competition;

```

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

```

1  /*Клубы СанктПетербурга*/
2  select name
3  from club
4  where city = 'СанктПетербург';
5
6  /*Клубы, в названии которых есть слово Dance*/
7  select name
8  from club
9  where name like '%Dance%';
10
11 /*Соревнования, проходившие летом 2016 года*/
12 select comp_id, date_start, date_end
13 from competition_history
14 where date_start between '1.6.2016' and '31.8.2016';
15
16 /*Пары, занявшие с 1 по 6 места за всю историю наблюдений*/
17 select comp_h_id, nomination_id, plase, leader_id, partner_id
18 from competition_result

```

```

19| where plase in (1, 6);
20|

```

### Вычисляемое поле в запросе

Вывести, сколько денег с участников и гостей было получено клубом-организатором за каждое соревнование 2016 года.

```

1| select
2|   c.name,
3|   b.name as organisator ,
4|   h.num_of_guest as visitors ,
5|   h.cost_of_participation*h.num_of_participant+h.cost_of_entrance*h.num_of_guest as income
6| from competition c
7|   join competition_history h on h.comp_id = c.comp_id
8|   join club b on b.club_id = c.organizer
9| where
10|  h.date_start between '1.1.2016' and '31.12.2016';

```

### Выборка всех данных с сортировкой по нескольким полям

```

1| /*Отсортировать танцоров по убыванию номера ACX*/
2| select first 20
3|   firstname ,
4|   secondname ,
5|   ASH_id
6| from dancer
7| order by ASH_id desc;
8|
9| /*Отсортировать клубы по городу, а затем по алфавиту*/
10| select first 40
11|   name,
12|   city
13| from club
14| where club_id < 200
15| order by city , name;
16|

```

### Запрос, вычисляющий несколько совокупных характеристик таблиц

Вывести, сколько лет проводится чемпионат, среднее количество участников и гостей, а также среднюю стоимость участия в чемпионате и цену за входной билет.

```

1| select
2|   count(*) ,
3|   avg(num_of_participant) as avg_num_of_participants ,
4|   avg(cost_of_participation) as avg_cost_of_participation ,
5|   avg(num_of_guest) as avg_num_of_guest ,
6|   avg(cost_of_entrance) as avg_cost_of_entrance
7| from
8|   competition_history
9| group by comp_id;

```

### Выборка данных из связанных таблиц

```

1| /*Вывести название, даты проведения и организатора питерских турниров*/
2| select
3|   c.name,
4|   h.date_start ,
5|   h.date_end ,
6|   c.organizer
7| from competition c
8|   join competition_history h on h.comp_id = c.comp_id
9| where c.city = 'СанктПетербург';
10|
11| /*Вывести фамилию, имя и ранг для каждого судьи*/
12| select
13|   d.firstname ,
14|   d.secondname ,

```

```

15 | j.rank
16 | from dancer d
17 | join dancer_ASH a on a.ASH_id = d.ASH_id
18 | join judge j on a.ASH_id = j.ASH_id;
19 |

```

**Запрос, рассчитывающий совокупную характеристику с использованием группировки с ограничением на результат группировки**

Подсчитать количество танцоров с номером АСХ меньшим 10000 в каждом классе. Вывести в порядке увеличения класса и только те классы, где танцоров больше 5.

```

1 | select class, count(*)
2 | from dancer_ASH
3 | where ASH_id < 10000
4 | group by class
5 | having count(*) > 5;

```

**Вложенный запрос**

Выведите id, фамилию и имя танцоров из питерских клубов.

```

1 | select
2 | dancer_id,
3 | (select firstname from dancer d where c.dancer_id = d.dancer_id) as firstname,
4 | (select secondname from dancer d where c.dancer_id = d.dancer_id) as secondname
5 | from dancer_club c
6 | where
7 | club_id in (select club_id from club where city = 'СанктПетербург-');
8 |

```

**Добавление в каждую таблицу одной записи**

```

1 | insert into club values
2 | (100, 'Smile', 'Москва');
3 | insert into dancer values
4 | (13, 'Мавлатов', 'Юрий', 1450);
5 | insert into dancer_ASH values
6 | (1450, 'A', 'S');
7 | insert into dancer_club values
8 | (13, 100);
9 | insert into club_leader values
10 | (100, 13);
11 | insert into judge values
12 | (8, 1450, 1, '12.6.2011');
13 | insert into nomination values
14 | (8, 'ABC', 'yes');
15 | insert into competition values
16 | (5, 'Кубок морской славы', 'СанктПетербург-', 0);
17 | insert into competition_history values
18 | (5, 4, '29.10.2016', '29.10.2016', 200, 300, 230, 300);
19 | insert into competition_judge value
20 | (5, 6);
21 | insert into competition_result (comp_h_id, nomination_id, plase, point, partner_id) values
22 | (5, 4, 8, 2, 8595);
23 | insert into competition_nomination values
24 | (5, 2);
25 | insert into sponsor values
26 | (2, 'YouDance', 'Москва', 'club');
27 | insert into sponsor_club values
28 | (1, 2, 144);
29 | insert into sponsor_competition values
30 | (3, 1, 3);
31 | insert into subsidy_for_club values
32 | ('15.8.2015', 240000, 1, 2);
33 | insert into subsidy_for_competition values
34 | ('24.9.2016', 5000, 2, 2);
35 |

```

**Изменение значения нескольких полей у всех записей, отвечающих заданному условию**

```
1 update competition_history
2 set
3     num_of_guest = 50,
4     cost_of_entrance = 300
5 where num_of_guest is null and cost_of_entrance is null;
6
7 update competition_history
8 set
9     date_end = date_start
10 where comp_h_id > 10;
```

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

Удалить минимальную из всех дотаций соревнованиям.

```
1 delete from subsidy_for_competition
2 where subsidy = (select min(subsidy) from subsidy_for_competition);
```

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

Удалить из списка клубов те клубы, в которых не числится не одного танцора.

```
1 delete from club c
2 where not exists (select * from dancer_club d where c.club_id = d.club_id);
```

## 5.2 Выполнение запросов по индивидуальному заданию

**10 клубов, члены которых чаще всего занимали призовые места за всю историю наблюдений**

```
1 select first 10
2 count(*) as medalists, c.name as club, c.city
3 from competition_result r
4 join dancer d on r.partner_id = d.ASH_id or r.leader_id = d.ASH_id
5 join dancer_club dc on d.dancer_id = dc.dancer_id
6 join club c on dc.club_id = c.club_id
7 where r.plase between 1 and 3
8 group by c.club_id, c.name, c.city
9 order by medalists desc;
```

**10 танцоров, которые чаще всего участвовали в соревнованиях**

```
1 select first 10
2 count(*) as num_of_comp, d.firstname, d.secondname
3 from competition_result r
4 join dancer d on r.partner_id = d.ASH_id or r.leader_id = d.ASH_id
5 group by d.ASH_id, d.firstname, d.secondname
6 order by num_of_comp desc;
```

**10 мероприятий с наименьшим отношением суммарных спонсорских взносов к количеству участников**

```
1 select first 10
2 sum(sub.subsidy)/h.num_of_participant as sub_TO_num,
3 c.name, c.city, h.date_start
4 from competition_history h
5 join sponsor_competition sc on h.comp_h_id = sc.comp_h_id
6 join subsidy_for_competition sub on sc.subsidy_id = sub.subsidy_id
7 join competition c on h.comp_id = c.comp_id
8 group by c.name, c.city, h.date_start, h.num_of_participant
9 order by subTONum;
```

## 6 Вывод

В ходе работы я ознакомилась с языком SQL-DDL, который позволяет манипулировать данными, хранящимися в базе.

Команда SELECT языка SQL-DML не влияет на данные в базе. Это команда используется для выборки данных из таблицы. Извлекаемые данные формируются при помощи следующих предложений:

- FIRST N – выбрать первые N записей;
- JOIN ... ON ... – выбирать из объединенных (join) по некоторому условию (on) таблиц;
- WHERE – выбирать только те строки, где выполняется заданное условие;
- GROUP BY – сгруппировать по данному столбцу/столбцам;
- ORDER BY – отсортировать строки по данному столбцу/столбцам.

Команды INSERT (вставка), DELETE (удаление), UPDATE (изменения) служат для изменения данных в базе.

Скорость выполнения запросов зависит от:

- объема базы данных;
- степени ее нормализации (для нормализованной БД требуется большее количество объединений);
- количества накладываемых условий.

Особенностью языка SQL-DML является его декларативность, то есть при создании запросов указывается лишь цель, которую необходимо достичь, но не указывается способ достижения этой цели.