

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Факультет безопасности информационных технологий

Дисциплина:

«Информационная безопасность баз данных»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

«Манипулирование данными в БД на языке SQL»

Выполнили:

Чу Ван Доан, студент группы N3247



(подпись)

Проверил:

Волков А.Г.

(отметка о выполнении)

(подпись)

Санкт-Петербург

2024 г.

СОДЕРЖАНИЕ

Содержание	3
1 Манипулирование данными в БД на языке SQL	4
1.1 Цель работы.....	4
1.2 Задание.....	4
1.3 Ход работы.....	5
Заключение	11

1 Манипулирование данными в БД на языке SQL

1.1 Цель работы

Получение навыков создания базовых структурных элементов базы данных.

1.2 Задание

1.2.1 Наполнить таблицы базы данных при помощи операторов INSERT. Каждая таблица должна иметь не менее 5 разных записей.

1.2.2 Обновить записи в одной таблице на основании записи из другой (между таблицами должна быть связь).

1.2.3 Удалить несколько записей из одной таблицы на основании информации из другой таблицы.

1.2.4 Вывести часть столбцов из таблицы.

1.2.5 Вывести несколько записей из таблицы, используя условие ограничения.

1.2.6 Сделать декартово произведение двух таблиц.

1.2.7 Вывести записи из таблицы на основании условия ограничения, содержащегося в другой таблице.

1.2.8 Применить функции агрегирования к выводимым записям (sum, avg, min, max).

1.2.9 Вывести записи из таблицы, используя сортировку от большего к меньшему.

1.2.10 Вывести записи из таблицы, используя сортировку от меньшего к большему с ограничением количества выводимых строк.

1.2.11 Произвести агрегирование выводимых записей по одному из полей (group by).

1.2.12 Выполнить запрос, когда табличное выражение представляет собой другой запрос.

1.3 Ход работы

1.3.1 **Задание 1:** Наполнить таблицы базы данных при помощи операторов INSERT. Каждая таблица должна иметь не менее 5 разных записей.

Создадим таблицы, в которых хранится информация студентов, используя команду «create table»:

```
create table works(id_work int primary key, work_name varchar(20) not null,
student_name varchar not null);
```

```
create table students(id_student int primary key, student_name varchar not null,
student_group varchar(20) not null);
```

```
n3247_22=# create table works(id_work int primary key, work_name varchar(20) not null, student_name varchar not null);
CREATE TABLE
n3247_22=# select * from works;
 id_work | work_name | student_name
-----+-----+-----
(0 rows)

n3247_22=# create table students(id_student int primary key, student_name varchar not null, student_group varchar(20) not null);
CREATE TABLE
n3247_22=# select * from students;
 id_student | student_name | student_group
-----+-----+-----
(0 rows)
```

Рисунок 1 – Создание таблицы БД

Добавим записи в таблицы. Для этого воспользуемся командой «insert into».

```
n3247_22=# insert into works values (1, 'work_1', 'student_1');
INSERT 0 1
n3247_22=# insert into works values (2, 'work_2', 'student_1');
INSERT 0 1
n3247_22=# insert into works values (3, 'work_3', 'student_4');
INSERT 0 1
n3247_22=# insert into works values (4, 'work_4', 'student_2');
INSERT 0 1
n3247_22=# insert into works values (5, 'work_5', 'student_3');
INSERT 0 1
n3247_22=# insert into works values (6, 'work_6', 'student_2');
INSERT 0 1
n3247_22=# select * from works;
 id_work | work_name | student_name
-----+-----+-----
      1 | work_1    | student_1
      2 | work_2    | student_1
      3 | work_3    | student_4
      4 | work_4    | student_2
      5 | work_5    | student_3
      6 | work_6    | student_2
(6 rows)
```

Рисунок 2 – Добавление информации в таблицу works

```
n3247_22=# insert into students values (1, 'student_1', 'group_1');
INSERT 0 1
n3247_22=# insert into students values (2, 'student_2', 'group_3');
INSERT 0 1
n3247_22=# insert into students values (3, 'student_3', 'group_3');
INSERT 0 1
n3247_22=# insert into students values (4, 'student_4', 'group_2');
INSERT 0 1
n3247_22=# insert into students values (5, 'student_5', 'group_4');
INSERT 0 1
n3247_22=# insert into students values (6, 'student_6', 'group_2');
INSERT 0 1
n3247_22=#
n3247_22=# select * from students;
 id_student | student_name | student_group
-----+-----+-----
          1 | student_1    | group_1
          2 | student_2    | group_3
          3 | student_3    | group_3
          4 | student_4    | group_2
          5 | student_5    | group_4
          6 | student_6    | group_2
(6 rows)
```

Рисунок 3 – Добавление информации в таблицу students

1.3.2 **Задание 2:** Обновить записи в одной таблице на основании записи из другой (между таблицами должна быть связь).

Для того, чтобы обновить запись в таблице, воспользуемся командой «update»

```
n3247_22=# update works set student_name = (select student_name from students where id_student=4) where id_work=5;
UPDATE 1
n3247_22=# update works set student_name = (select student_name from students where id_student=1) where id_work=6;
UPDATE 1
n3247_22=# select * from works;
 id_work | work_name | student_name
-----+-----+-----
        1 | work_1    | student_1
        2 | work_2    | student_1
        3 | work_3    | student_4
        4 | work_4    | student_2
        5 | work_5    | student_4
        6 | work_6    | student_1
(6 rows)
```

Рисунок 4 – Обновление записи в таблице works

1.3.3 **Задание 3:** Удалить несколько записей из одной таблицы на основании информации из другой таблицы.

Для того, чтобы удалить записи из таблицы, воспользуемся командой «delete»

```
n3247_22=# delete from students using works where students.student_name=works.student_name and works.id_work=4;
DELETE 1
n3247_22=# select * from students;
 id_student | student_name | student_group
-----+-----+-----
          1 | student_1   | group_1
          3 | student_3   | group_3
          4 | student_4   | group_2
          5 | student_5   | group_4
          6 | student_6   | group_2
(5 rows)
```

Рисунок 5 – Удаление записей из таблицы students

1.3.4 Задание 4: Вывести часть столбцов из таблицы.

```
n3247_22=# select id_student, student_name from students;
 id_student | student_name
-----+-----
          1 | student_1
          3 | student_3
          4 | student_4
          5 | student_5
          6 | student_6
(5 rows)
```

Рисунок 6 – Выборка части столбцов из таблицы

1.3.5 Задание 5: Вывести несколько записей из таблицы, используя условие ограничения.

Выведем несколько записей из таблицы, используя условие ограничения «id_work < 4».

```
n3247_22=# select * from works where id_work <= 3;
 id_work | work_name | student_name
-----+-----+-----
        1 | work_1   | student_1
        2 | work_2   | student_1
        3 | work_3   | student_4
(3 rows)
```

Рисунок 7 – Выборка из таблицы записей, удовлетворяющей условию

1.3.6 Задание 6: Сделать декартово произведение двух таблиц.

Сделаем декартово произведение двух таблиц с помощью команды «inner join».

```
n3247_22=# select * from works inner join students on works.student_name=students.student_name;
 id_work | work_name | student_name | id_student | student_name | student_group
-----+-----+-----+-----+-----+-----
      1 | work_1    | student_1    |          1 | student_1    | group_1
      2 | work_2    | student_1    |          1 | student_1    | group_1
      3 | work_3    | student_4    |          4 | student_4    | group_2
      5 | work_5    | student_4    |          4 | student_4    | group_2
      6 | work_6    | student_1    |          1 | student_1    | group_1
(5 rows)
```

Рисунок 8 – Декартово произведение двух таблиц

1.3.7 **Задание 7:** Вывести записи из таблицы на основании условия ограничения, содержащегося в другой таблице.

```
n3247_22=# select * from students where student_name in (select student_name from works);
 id_student | student_name | student_group
-----+-----+-----
          1 | student_1    | group_1
          4 | student_4    | group_2
(2 rows)
```

Рисунок 9 – Выборка записей из таблицы на основании условия ограничения,...

1.3.8 **Задание 8:** Применить функции агрегирования к выводимым записям (sum, avg, min, max).

Добавляем один столбец “student_point” в таблицу students:

```
n3247_22=# alter table students add student_point int;
ALTER TABLE
n3247_22=# select * from students;
 id_student | student_name | student_group | student_point
-----+-----+-----+-----
          1 | student_1    | group_1       |
          3 | student_3    | group_3       |
          4 | student_4    | group_2       |
          5 | student_5    | group_4       |
          6 | student_6    | group_2       |
(5 rows)
```

Рисунок 10 – Добавление столбца в таблицу students

Добавляем данные для столбца «student_point», воспользуемся командой «update»:

```

n3247_22=# update students set student_point=64 where student_name='student_1';
UPDATE 1
n3247_22=# update students set student_point=99 where student_name='student_3';
UPDATE 1
n3247_22=# update students set student_point=75 where student_name='student_4';
UPDATE 1
n3247_22=# update students set student_point=100 where student_name='student_5';
UPDATE 1
n3247_22=# update students set student_point=69 where student_name='student_6';
UPDATE 1
n3247_22=# select * from students;
 id_student | student_name | student_group | student_point
-----+-----+-----+-----
      1 | student_1    | group_1       |          64
      3 | student_3    | group_3       |          99
      4 | student_4    | group_2       |          75
      5 | student_5    | group_4       |         100
      6 | student_6    | group_2       |          69
(5 rows)

```

Рисунок 11 – Добавление данных в столбец «student_point»

```

n3247_22=# select sum(student_point) from students;
 sum
-----
  407
(1 row)

n3247_22=# select avg(student_point) from students;
 avg
-----
81.400000000000000000000000000000
(1 row)

n3247_22=# select min(student_point) from students;
 min
-----
   64
(1 row)

n3247_22=# select max(student_point) from students;
 max
-----
  100
(1 row)

```

Рисунок 12 – Применение функций агрегирования к выводимым записям

1.3.9 **Задание 9:** Вывести записи из таблицы, используя сортировку от большего к меньшему.

```
n3247_22=# select * from students order by student_point desc;
id_student | student_name | student_group | student_point
-----+-----+-----+-----
          5 | student_5    | group_4       |          100
          3 | student_3    | group_3       |           99
          4 | student_4    | group_2       |           75
          6 | student_6    | group_2       |           69
          1 | student_1    | group_1       |           64
(5 rows)
```

Рисунок 13 – Выборка записей из таблицы, используя сортировку от большего к меньшему

1.3.10 **Задание 10:** Вывести записи из таблицы, используя сортировку от меньшего к большему с ограничением количества выводимых строк.

```
n3247_22=# select * from students order by student_point limit 3;
id_student | student_name | student_group | student_point
-----+-----+-----+-----
          1 | student_1    | group_1       |           64
          6 | student_6    | group_2       |           69
          4 | student_4    | group_2       |           75
(3 rows)
```

Рисунок 14 – Выборка записей из таблицы, используя сортировку от меньшего к большему с ограничением количества выводимых строк

1.3.11 **Задание 11:** Произвести агрегирование выводимых записей по одному из полей (group by).

```
n3247_22=# select student_name, count(student_name) from works group by student_name;
student_name | count
-----+-----
student_1    |      3
student_2    |      1
student_4    |      2
(3 rows)
```

Рисунок 15 – Агрегирование выводимых записей по одному из полей (group by)

1.3.12 **Задание 12:** Выполнить запрос, когда табличное выражение представляет собой другой запрос.

- good_students – таблица, содержащая student_name и количества вхождений в таблицу works (count).

- top_students – таблица, показывающая student_name, которые появляются более 1 раза в таблице good_students.

*with good_students as (select student_name, count(student_name) from works group by student_name), top_students as (select student_name from good_students where count > 1) select * from works where student_name in (select student_name from top_students);*

```
n3247_22=# with good_students as (select student_name, count(student_name) from works group by student_name),
top_students as (select student_name from good_students where count > 1)
select * from works where student_name in (select student_name from top_students);
 id_work | work_name | student_name
-----+-----+-----
      1 | work_1    | student_1
      2 | work_2    | student_1
      3 | work_3    | student_4
      5 | work_5    | student_4
      6 | work_6    | student_1
(5 rows)
```

Рисунок 16 – Выполнение запроса, когда табличное выражение представляет собой другой запрос

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы был изучен теоретический материал по манипулированию данными в БД на языке SQL. Приобретенные знания были применены на практике в СУБД PostgreSQL.