

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет безопасности информационных технологий

Направление подготовки: 10.03.01 Информационная безопасность

Образовательная программа: "Информационная безопасность / Information security"

Дисциплина:

«Информационная безопасность баз данных»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

«Манипулирование данными в БД на языке SQL»

Выполнил студент:

N3246 / ИББД N23 1.3

Суханкулиев Мухаммет / _____

ФИО

Подпись

Проверила:

Карманова Наталия Андреевна / _____

ФИО

Подпись

*Отметка о выполнении (один из вариантов:
отлично, хорошо, удовлетворительно, зачтено)*

Дата

Санкт-Петербург

2025 г.

СОДЕРЖАНИЕ

Введение	4
1 Манипулирование данными в БД на языке SQL	5
1.1 Ход работы	5
1.1.1 Создание 4 связанных таблиц.	5
1.1.2 Наполнение базы данных.....	5
1.1.3 Обновление записи в одной таблице на основании информации из другой таблицы.....	6
1.1.4 Удаление нескольких записей из одной таблицы на основании информации из другой таблицы.	6
1.1.5 Вывести часть столбцов из таблицы.....	6
1.1.6 Вывести несколько записей из таблицы, используя условие ограничения.	7
1.1.7 Декартово произведение двух таблиц.	7
1.1.8 Вывести записи из таблицы на основании условия ограничения, содержащегося в другой таблице.....	7
1.1.9 Функции агрегирования (sum, avg, min, max).....	8
1.1.10 Вывести записи из таблицы, используя сортировку от большего к меньшему.....	8
1.1.11 Вывести записи из таблицы, используя сортировку от меньшего к большему с ограничением количества выводимых строк.	8
1.1.12 Агрегирование выводимых записей по одному из полей (group by).....	9
1.1.13 Запрос, когда табличное выражение представляет собой другой запрос.	9
Заключение.....	10
Список использованных источников.....	11

ВВЕДЕНИЕ

Цель работы – получение навыков манипулирования данными в БД при помощи операторов SQL.

Для достижения поставленной цели необходимо решить следующие **задачи**:

- Создать по крайней мере 3 связанные таблицы. Должны быть определены первичные и внешние ключи для таблиц, т. е. по крайней мере для одной пары таблиц должна быть определена связь 1:M;
- Наполнить таблицы базы данных при помощи операторов INSERT. Каждая таблица должна иметь не менее 5 разных записей;
- Обновить записи в одной таблице на основании записи из другой (между таблицами должна быть связь);
- Удалить несколько записей из одной таблицы на основании информации из другой таблицы;
- Вывести часть столбцов из таблицы;
- Вывести несколько записей из таблицы, используя условие ограничения;
- Сделать декартово произведение двух таблиц;
- Вывести записи из таблицы на основании условия ограничения, содержащегося в другой таблице;
- Применить функции агрегирования к выводимым записям (sum, avg, min, max);
- Вывести записи из таблицы, используя сортировку от большего к меньшему;
- Вывести записи из таблицы, используя сортировку от меньшего к большему с ограничением количества выводимых строк;
- Произвести агрегирование выводимых записей по одному из полей (group by);
- Выполнить запрос, когда табличное выражение представляет собой другой запрос.

1 МАНИПУЛИРОВАНИЕ ДАННЫМИ В БД НА ЯЗЫКЕ SQL

База данных будет создана на том же хосте, что и в первой лабораторной работе.

```
postgres=# CREATE DATABASE marvel
postgres=# ;
CREATE DATABASE
postgres=# \c marvel
Вы подключены к базе данных "marvel" как пользователь "postgres".
```

1.1 Ход работы

1.1.1 Создание 4 связанных таблиц.

```
CREATE TABLE teams (
    id SERIAL PRIMARY KEY,
    name TEXT NOT NULL UNIQUE,
    base TEXT NOT NULL
);
CREATE TABLE Query returned successfully in 49 msec.

CREATE TABLE heroes (
    id SERIAL PRIMARY KEY,
    name TEXT NOT NULL UNIQUE,
    real_name TEXT NOT NULL,
    team_id INT,
    debut_year INT CHECK (debut_year >= 1900),
    FOREIGN KEY (team_id) REFERENCES teams(id) ON DELETE SET NULL
);
CREATE TABLE Query returned successfully in 53 msec.

CREATE TABLE powers (
    id SERIAL PRIMARY KEY,
    name TEXT NOT NULL UNIQUE
);
CREATE TABLE Query returned successfully in 42 msec.

CREATE TABLE hero_powers (
    hero_id INT NOT NULL,
    power_id INT NOT NULL,
    PRIMARY KEY (hero_id, power_id),
    FOREIGN KEY (hero_id) REFERENCES heroes(id) ON DELETE CASCADE,
    FOREIGN KEY (power_id) REFERENCES powers(id) ON DELETE CASCADE
);
CREATE TABLE Query returned successfully in 45 msec.
```

1.1.2 Наполнение базы данных.

```
INSERT INTO teams (name, base) VALUES
('Avengers', 'Stark_Tower'),
('X-Men', 'Xavier_Institute'),
('Guardians_of_the_Galaxy', 'Space'),
('Fantastic_Four', 'Бакстер Билдинг'),
('S.H.I.E.L.D.', 'Helicarrier');
INSERT 0 5 Query returned successfully in 46 msec.
```

```

UPDATE teams
SET base = 'Baxter_Building'
WHERE base = 'Бакстеп Билдинг';
UPDATE 1 Query returned successfully in 121 msec.

INSERT INTO heroes (name, real_name, team_id, debut_year) VALUES
('Iron_Man', 'Tony_Stark', 1, 1963),
('Captain_America', 'Steeve_Rodgers', 1, 1941),
('Wolverine', 'Logan', 2, 1974),
('Spider-Man', 'Piter_Parker', NULL, 1962), -- Без команды
('Star-Lord', 'Piter_Quill', 3, 1976);
INSERT 0 5 Query returned successfully in 113 msec.

INSERT INTO powers (name) VALUES
('Genius'),
('Billionaire'),
('Playboy'),
('Superpower'),
('Regeneration'),
('Accurate'),
('Spider_senses');
INSERT 0 7 Query returned successfully in 73 msec.

INSERT INTO hero_powers (hero_id, power_id) VALUES
(1, 1), -- Железный человек -> Гений
(1, 2),
(1, 3),
(2, 4), -- Капитан Америка -> Суперсила
(3, 5), -- Росوماха -> Регенерация
(4, 7), -- Человек-паук -> Паучье чутье
(5, 6); -- Звёздный Лорд -> Мастер стрельбы
INSERT 0 7 Query returned successfully in 116 msec.

```

P.s.: Для уточнения ID можно использовать SELECT

1.1.3 Обновление записи в одной таблице на основании информации из другой таблицы.

```

UPDATE heroes
SET debut_year = 1960
FROM teams
WHERE heroes.team_id = teams.id AND teams.name = 'Avengers';
UPDATE 2 Query returned successfully in 46 msec.

```

1.1.4 Удаление нескольких записей из одной таблицы на основании информации из другой таблицы.

```

DELETE FROM heroes
WHERE team_id IN (
    SELECT id FROM teams WHERE name = 'X-Men'
);
DELETE 1 Query returned successfully in 68 msec.

```

1.1.5 Вывести часть столбцов из таблицы.

```

marvel=# SELECT name, real_name FROM heroes;
name          | real_name

```

```

-----+-----
Spider-Man      | Piter_Parker
Star-Lord       | Piter_Quill
Iron_Man        | Tony_Stark
Captain_America | Steeve_Rodgers
(4 строки)

```

1.1.6 Вывести несколько записей из таблицы, используя условие ограничения.

```

marvel=# SELECT name, real_name, debut_year FROM heroes
        WHERE team_id = (SELECT id FROM teams WHERE name = 'Avengers');
 name      | real_name | debut_year
-----+-----+-----
Iron_Man   | Tony_Stark |      1960
Captain_America | Steeve_Rodgers |      1960
(2 строки)

```

1.1.7 Декартово произведение двух таблиц.

```

marvel=# SELECT heroes.name AS name, teams.name AS team FROM heroes CROSS
        JOIN teams;
 name      | team
-----+-----
Spider-Man | Avengers
Star-Lord  | Avengers
Iron_Man   | Avengers
Captain_America | Avengers
Spider-Man | X-Men
Star-Lord  | X-Men
Iron_Man   | X-Men
Captain_America | X-Men
Spider-Man | Guardians_of_the_Galaxy
Star-Lord  | Guardians_of_the_Galaxy
Iron_Man   | Guardians_of_the_Galaxy
Captain_America | Guardians_of_the_Galaxy
Spider-Man | S.H.I.E.L.D.
Star-Lord  | S.H.I.E.L.D.
Iron_Man   | S.H.I.E.L.D.
Captain_America | S.H.I.E.L.D.
Spider-Man | Fantastic_Four
Star-Lord  | Fantastic_Four
Iron_Man   | Fantastic_Four
Captain_America | Fantastic_Four
(20 строк)

```

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

```

marvel=# SELECT name, debut_year
marvel=# FROM heroes
marvel=# WHERE debut_year > 1959
marvel=# AND team_id = (SELECT id FROM teams WHERE name = 'Avengers');
 name      | debut_year
-----+-----
Iron_Man   |      1960
Captain_America |      1960

```

(2 строки)

1.1.9 Функции агрегирования (sum, avg, min, max).

```
marvel=# SELECT SUM(debut_year) AS total_debut_years FROM heroes;
total_debut_years
-----
          7858
(1 строка)

marvel=# SELECT AVG(debut_year) AS average_debut_year FROM heroes;
average_debut_year
-----
1964.5000000000000000
(1 строка)

marvel=# SELECT MIN(debut_year) AS earliest_debut_year FROM heroes;
earliest_debut_year
-----
          1960
(1 строка)

marvel=# SELECT MAX(debut_year) AS latest_debut_year FROM heroes;
latest_debut_year
-----
          1976
(1 строка)
```

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

```
marvel=# SELECT name, real_name, debut_year FROM heroes ORDER BY debut_year
DESC;
name      | real_name | debut_year
-----+-----+-----
Star-Lord | Piter_Quill |          1976
Spider-Man | Piter_Parker |          1962
Iron_Man  | Tony_Stark  |          1960
Captain_America | Steeve_Rodgers |          1960
(4 строки)
```

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

```
marvel=# SELECT name, real_name FROM heroes ORDER BY name ASC LIMIT 3;
name      | real_name
-----+-----
Captain_America | Steeve_Rodgers
Iron_Man      | Tony_Stark
Spider-Man     | Piter_Parker
(3 строки)
```

1.1.12 Агрегирование выводимых записей по одному из полей (group by).

```
marvel=# SELECT teams.name AS team, COUNT(heroes.id) AS hero_count
        FROM heroes, teams WHERE heroes.team_id = teams.id
        GROUP BY teams.name;
   team          | hero_count 
-----+-----
Guardians_of_the_Galaxy |          1
Avengers          |          2
(2 строки)
```

1.1.13 Запрос, когда табличное выражение представляет собой другой запрос.

```
marvel=# SELECT name, debut_year FROM heroes,
        (SELECT AVG(debut_year) AS avg_debut_year FROM heroes) AS avg_year
        WHERE debut_year > avg_year.avg_debut_year;
   name          | debut_year 
-----+-----
Star-Lord        |        1976
(1 строка)
```


ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы были получены практические навыки манипулирования данными в базе данных с использованием языка SQL. Были созданы взаимосвязанные таблицы с первичными и внешними ключами.

Работа позволила закрепить базовые навыки работы с SQL, полезные для администрирования и анализа данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. НИУ ИТМО Методические указания по проведению лабораторной работы №2 – Манипулирование данными в БД на языке SQL: электронный ресурс – 2025. – URL: <https://docs.google.com/document/d/1XX9LogkHxCNUUVp638h2FpBt0Ojlsqva>
2. Основы технологий баз данных: учебное пособие / Б. А. Новиков, Е. А. Горшкова, Н. Г. Графеева; под ред. Е. В. Рогова. — 2-е изд. — М.: ДМК Пресс, 2020. — 582 с. — URL: https://drive.google.com/file/d/1TjYbunEjxsbovBiHeYOzBuZrOFonlIRk/view?usp=drive_link
3. Базы данных: Учебник для высших учебных заведений / Под ред. проф. А. Д. Хомоненко. — 6-е изд., доп. - СПб.: КОРОНА-Век, 2009. — 736 с. — URL: https://drive.google.com/file/d/1zIOuO6vdQvb_aVUHGAiucK5MPciUswGf/view?usp=drive_link