



**Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Пензенский государственный университет»
(ФГБОУ ВО «Пензенский государственный университет»)**

**Нижнеломовский филиал ФГБОУ ВО
«Пензенский государственный университет»
(НлФ ФГБОУ ВО «ПГУ»)**

Практическая работа

**по дисциплине: «Технология разработки и защиты баз данных»
на тему: «Создание таблиц и простейших запросов на языке SQL»**

Выполнили:
студент группы 23НФПО
Кириллов А.О.
Веденяпин Д.А.
Принял:
Преподаватель
Кайгородова В.О.

Нижний Ломов, 2025 г.

Цель работы: отработать навыки создания таблиц, добавления данных и выполнения простейших запросов на языке SQL.

Задачи:

1. Создать структуру базы данных (таблицы) для заданной предметной области.
2. Заполнить таблицы данными (не менее 5 записей в каждой).
3. Выполнить указанные запросы для получения данных.

Ход работы:

1. Делаем таблицы для предметных областей.

Задание 1 Библиотека. Предметная область: Управление библиотечным фондом.

```
1  DROP TABLE IF EXISTS authors CASCADE;
2  DROP TABLE IF EXISTS books CASCADE;
3
4  CREATE TABLE authors (
5      author_id SERIAL,
6      first_name TEXT,
7      country TEXT,
8      PRIMARY KEY (author_id)
9  );
10
11 CREATE TABLE books (
12     book_id UUID DEFAULT gen_random_uuid(),
13     author_id INT NOT NULL,
14     title TEXT NOT NULL,
15     year_public INT NOT NULL,
16     PRIMARY KEY (book_id),
17     FOREIGN KEY (author_id) REFERENCES authors(author_id)
18 );
19
```

```

20 ✓ INSERT INTO authors (first_name, country) VALUES
21     ('Артур', 'Нигерия'),
22     ('Артур', 'Эфиопия'),
23     ('Василиса', 'Узбекистан'),
24     ('Анастасия', 'Армения'),
25     ('Владислейв', 'ОАЭ');
26
27 ✓ INSERT INTO books (author_id, title, year_publick) VALUES
28     (1, 'Инструкция по сборке от двери запорожца', 1925),
29     (2, 'React для чайников', 2026),
30     (3, 'Расстрел святой водой', 2000),
31     (4, 'Как уместить все в себе и даже больше', 2025),
32     (5, 'Как бидуин нашел нефть', 800);

```

```

34 SELECT title FROM books;
35 SELECT title FROM books WHERE year_publick > 2000;
36 SELECT b.title AS title_book, a.first_name AS first_name_author| FROM books b INNER JOIN authors a USING (author_id);

```

	title text	🔒
1	Инструкция по сборке от двери запорожца	
2	React для чайников	
3	Расстрел святой водой	
4	Как уместить все в себе и даже больше	
5	Как бидуин нашел нефть	

	title text	🔒
1	React для чайников	
2	Как уместить все в себе и даже больше	

	title_book text	🔒	first_name_author text	🔒
1	Инструкция по сборке от двери запорожца		Артур	
2	React для чайников		Артур	
3	Расстрел святой водой		Василиса	
4	Как уместить все в себе и даже больше		Анастасия	
5	Как бидуин нашел нефть		Владислейв	

Задание 2 Интернет-Магазин. Предметная область: Управление товарами интернет-магазина.

```
1 DROP TABLE IF EXISTS categories CASCADE;
2 DROP TABLE IF EXISTS products CASCADE;
3
4 CREATE TABLE categories (
5     category_id SERIAL,
6     title TEXT,
7     description TEXT,
8     PRIMARY KEY (category_id)
9 );
10
11 CREATE TABLE products (
12     product_id UUID DEFAULT gen_random_uuid(),
13     category_id INT NOT NULL,
14     title TEXT NOT NULL,
15     price MONEY NOT NULL,
16     PRIMARY KEY (product_id),
17     FOREIGN KEY (category_id) REFERENCES categories(category_id)
18 );
19
20 INSERT INTO categories (title, description) VALUES
21 ('Запчасти', 'Детали которые спасут вас от всего на свете'),
22 ('Игрушки', 'Обеспечат вам веселье'),
23 ('Электроника', 'Просто электроника лол'),
24 ('Бытовые товары', 'С ними не заскучаешь в любую погоду'),
25 ('Природные ископаемые', 'Сделает вас должником государству');
26
27 INSERT INTO products (category_id, title, price) VALUES
28 (1, 'Дверь от запорожца', 40000.00),
29 (2, 'Фиолетовый пазел вдупло', 250000),
30 (3, 'Супер про пупер меганупер райзен 7 7500f', 250000),
31 (4, 'Чайник ВАСИЛИСА', 0),
32 (5, 'Жижа от камаза', 9999999);
33
34 SELECT title FROM products;
35 SELECT title FROM products WHERE price > 1000;
36 SELECT p.title AS product_title, c.title AS category_title FROM products p INNER JOIN categories c USING (category_id);
```

	title text	🔒
1	Дверь от запорожца	
2	Фиолетовый пазел вдупло	
3	Супер про пупер мегапупер райзен 7 7500f	
4	Чайник ВАСИЛИСА	
5	Жижа от камаза	

	title text	🔒
1	Дверь от запорожца	
2	Фиолетовый пазел вдупло	
3	Супер про пупер мегапупер райзен 7 7500f	
4	Жижа от камаза	

	product_title text	🔒	category_title text	🔒
1	Дверь от запорожца		Запчасти	
2	Фиолетовый пазел вдупло		Игрушки	
3	Супер про пупер мегапупер райзен 7 7500f		Электроника	
4	Чайник ВАСИЛИСА		Бытовые товары	
5	Жижа от камаза		Природные ископаемые	

Задание 3 Университет. Предметная область: Управление учебным процессом.

```
1  DROP TABLE IF EXISTS courses CASCADE;
2  DROP TABLE IF EXISTS students CASCADE;
3
4  ✓ CREATE TABLE courses (
5      course_id SERIAL,
6      title TEXT,
7      teacher_first_name TEXT,
8      PRIMARY KEY (course_id)
9  );
10
11 ✓ CREATE TABLE students (
12     student_id UUID DEFAULT gen_random_uuid(),
13     course_id INT,
14     first_name INT NOT NULL,
15     group_title TEXT NOT NULL,
16     PRIMARY KEY (student_id),
17     FOREIGN KEY (course_id) REFERENCES courses(course_id)
18 );
19
20 ✓ INSERT INTO courses (title, teacher_first_name) VALUES
21 ('Дверовцы', 'Артур'),
22 ('Прогроберы', 'Андрей'),
23 ('Компьютерщики', 'Андрей'),
24 ('Подсчитыватели', 'Лиля'),
25 ('Адвокатчики', 'Яна');
26
27 ✓ INSERT INTO students (course_id, first_name, group_title) VALUES
28 (1, 'Артурио', '21ДВ'),
29 (2, 'Васельса', '15ПГ'),
30 (3, 'Александр аля Булочкин', '30КТ'),
31 (4, 'Андриана', '25ПС'),
32 (5, 'Любава', '69АК');
33
34 SELECT first_name FROM students;
35 SELECT first_name FROM students WHERE group_title = '69АК';
36 SELECT s.first_name AS student_name, c.title AS course_title FROM students s INNER JOIN courses c USING (course_id);
```

	first_name text	🔒
1	Артурио	
2	Васельса	
3	Александр аля Булочкин	
4	Андриана	
5	Любава	

	first_name text	🔒
1	Любава	

	student_name text	🔒	course_title text	🔒
1	Артурио		Дверовцы	
2	Васельса		Прогроберы	
3	Александр аля Булочкин		Компьютерщики	
4	Андриана		Подсчитыватели	
5	Любава		Адвокатчики	

Задание 4 Автосервис. Предметная область: Управление заказами автосервиса.

```
1 DROP TABLE IF EXISTS clients CASCADE;
2 DROP TABLE IF EXISTS orders CASCADE;
3
4 CREATE TABLE clients (
5     client_id SERIAL,
6     name TEXT,
7     phone text,
8     PRIMARY KEY (client_id)
9 );
10
11 CREATE TABLE orders (
12     order_id serial,
13     client_id INT,
14     date text NOT NULL,
15     price int NOT NULL,
16     PRIMARY KEY (order_id),
17     FOREIGN KEY (client_id) REFERENCES clients(client_id)
18 );
19
```

```
INSERT INTO clients (name, phone) VALUES
('Артур', '79624733115'),
('Данила', '79875241774'),
('Пабло', '78535422452'),
('Елена', '73256634693'),
('Таксиса', '78535422451');
```

```
INSERT INTO orders (client_id, date, price) VALUES
1, '03.09.78', 1),
2, '30.09.07', 3),
3, '23.04.25', 30000),
4, '24.12.24', 25000),
5, '04.03.02', 69034);
```

```
34 SELECT * FROM orders;
35 SELECT * FROM orders WHERE price > 5000;
36 SELECT o.order_id, o.date, o.price, c.name FROM orders o INNER JOIN clients c USING (client_id);
```


	order_id [PK] integer	client_id integer	date text	price integer
1	1	1	03.09.78	1
2	2	2	30.09.07	3
3	3	3	23.04.25	30000
4	4	4	24.12.24	25000
5	5	5	04.03.02	69034

	order_id [PK] integer	client_id integer	date text	price integer
1	3	3	23.04.25	30000
2	4	4	24.12.24	25000
3	5	5	04.03.02	69034

	order_id integer	date text	price integer	name text
1	1	03.09.78	1	Артур
2	2	30.09.07	3	Данила
3	3	23.04.25	30000	Пабло
4	4	24.12.24	25000	Елена
5	5	04.03.02	69034	Таксиса

Задание 5 Кинотеатр. Предметная область: Управление показами фильмов.

```

1  DROP TABLE IF EXISTS movies CASCADE;
2  DROP TABLE IF EXISTS showtimes CASCADE;
3
4  CREATE TABLE movies (
5      movie_id serial,
6      title TEXT NOT NULL,
7      genre TEXT NOT NULL,
8      PRIMARY KEY (movie_id)
9  );
10
11 CREATE TABLE showtimes (
12     showtime_id SERIAL,
13     movie_id INT,
14     time_show TIMESTAMP DEFAULT NOW(),
15     PRIMARY KEY (showtime_id),
16     FOREIGN KEY (movie_id) REFERENCES movies (movie_id)
17 );

```

```

19  ✓ INSERT INTO movies (title, genre) VALUES
20    ('Форсаж двери не летают', 'Боевик'),
21    ('Чайник летит', 'Комедия'),
22    ('Гусь свинье не товарищ', 'Драмма'),
23    ('Крещение Василисы. Начало зада', 'Документальное'),
24    ('Груз вакутагин (200)', 'Триллер');
25
26  ✓ INSERT INTO showtimes (movie_id) VALUES
27    (1),
28    (2),
29    (3),
30    (4),
31    (5);
32
33  SELECT title FROM movies;
34  SELECT title FROM movies WHERE genre = 'Документальное';
35  SELECT m.title, s.time_show FROM movies m INNER JOIN showtimes s USING (movie_id);

```


	title text
1	Форсаж двери не летают
2	Чайник летит
3	Гусь свинье не товарищ
4	Крещение Василисы. Начало зада
5	Груз вакутагин (200)


	title text
1	Крещение Василисы. Начало зада



	title text	time_show timestamp without time zone
1	Форсаж двери не летают	2025-05-29 09:51:30.060856
2	Чайник летит	2025-05-29 09:51:30.060856
3	Гусь свинье не товарищ	2025-05-29 09:51:30.060856
4	Крещение Василисы. Начало зада	2025-05-29 09:51:30.060856
5	Груз вакутагин (200)	2025-05-29 09:51:30.060856

Задание 6 Спортивный клуб. Предметная область: Управление тренировками.

```
1  DROP TABLE IF EXISTS members CASCADE;
2  DROP TABLE IF EXISTS trainings CASCADE;
3
4  CREATE TABLE members (
5      member_id serial,
6      first_name TEXT NOT NULL,
7      age INT NOT NULL,
8      PRIMARY KEY (member_id)
9  );
10
11 CREATE TABLE trainings (
12     training_id SERIAL,
13     type TEXT NOT NULL,
14     member_id INT,
15     PRIMARY KEY (training_id),
16     FOREIGN KEY (member_id) REFERENCES members (member_id)
17 );
18
19 INSERT INTO members (first_name, age) VALUES
20 ('Артур', 17),
21 ('Васильеса', 17),
22 ('Анастасия', 16),
23 ('Любава', 2),
24 ('Руслан', 18);
25
26 INSERT INTO trainings (type, member_id) VALUES
27 ('Жим двери от запорожца лежа', 1),
28 ('Заплыв 50км по Волге', 2),
29 ('Подъем термоса с чаем', 3),
30 ('Дыхательные тренировки для родов', 4),
31 ('Становая тяга седла', 5);
32
33 SELECT first_name FROM members;
34 SELECT first_name FROM members WHERE age > 18;
35 SELECT m.first_name, t.type FROM trainings t INNER JOIN members m USING (member_id);
```

	first_name text 
1	Артур
2	Васильеса
3	Анастасия
4	Любава
5	Руслан

	first_name text 

	first_name text 	type text 
1	Артур	Жим двери от запорожца лежа
2	Васильеса	Заплыв 50км по Волге
3	Анастасия	Подъем термоса с чаем
4	Любава	Дыхательные тренировки для родов
5	Руслан	Становая тяга седла

Задание 7 Ресторан. Предметная область: Управление заказами ресторана.

```
1  DROP TABLE IF EXISTS customers CASCADE;
2  DROP TABLE IF EXISTS orders CASCADE;
3
4  CREATE TABLE customers (
5      customer_id serial,
6      first_name TEXT NOT NULL,
7      email TEXT NOT NULL,
8      PRIMARY KEY (customer_id)
9  );
10
11 CREATE TABLE orders (
12     order_id SERIAL,
13     customer_id INT,
14     date TIMESTAMP DEFAULT NOW(),
15     amount MONEY,
16     PRIMARY KEY (order_id),
17     FOREIGN KEY (customer_id) REFERENCES customers (customer_id)
18 );
19
20 INSERT INTO customers (first_name, email) VALUES
21 ('Артур', 'dwdwawdwa@gmail.com'),
22 ('Васильеса', 'vefdwdwa@gmail.com'),
23 ('Анастасия', '142dwdwa@gmail.com'),
24 ('Любава', 'jtyytdwdwa@gmail.com'),
25 ('Руслан', 'sfefeeqwedwdwa@gmail.com');
26
27 INSERT INTO orders (customer_id, amount) VALUES
28 (1, 14646),
29 (2, 2124241),
30 (3, 3525),
31 (4, 423),
32 (5, 5253);
33
34 SELECT order_id, amount, date FROM orders;
35 SELECT order_id, amount, date FROM orders WHERE amount > 1000;
36 SELECT o.order_id, o.amount, o.date, c.first_name FROM orders o INNER JOIN customers c USING (customer_id);
```


	order_id [PK] integer	amount money	date timestamp without time zone
1	1	14 646,00 ?	2025-05-29 10:17:25.05385
2	2	2 124 241,00 ?	2025-05-29 10:17:25.05385
3	3	3 525,00 ?	2025-05-29 10:17:25.05385
4	4	423,00 ?	2025-05-29 10:17:25.05385
5	5	5 253,00 ?	2025-05-29 10:17:25.05385

	order_id [PK] integer	amount integer	date timestamp without time zone
1	1	14646	2025-05-29 10:19:57.218774
2	2	2124241	2025-05-29 10:19:57.218774
3	3	3525	2025-05-29 10:19:57.218774
4	5	5253	2025-05-29 10:19:57.218774

	order_id integer	amount integer	date timestamp without time zone	first_name text
1	1	14646	2025-05-29 10:19:57.218774	Артур
2	2	2124241	2025-05-29 10:19:57.218774	Васильеса
3	3	3525	2025-05-29 10:19:57.218774	Анастасия
4	4	423	2025-05-29 10:19:57.218774	Любава
5	5	5253	2025-05-29 10:19:57.218774	Руслан

Задание 8 Медицинский центр. Предметная область: Управление записями на прием.

```


1 DROP TABLE IF EXISTS patients CASCADE;
2 DROP TABLE IF EXISTS appointments CASCADE;
3
4 CREATE TABLE patients (
5     patient_id SERIAL,
6     ferst_name TEXT,
7     birthday DATE,
8     PRIMARY KEY (patient_id)
9 );
10
11 CREATE TABLE appointments (
12     appointment_id UUID DEFAULT gen_random_uuid(),
13     patient_id INT NOT NULL,
14     date_visit DATE NOT NULL,
15     PRIMARY KEY (appointment_id),
16     FOREIGN KEY (patient_id) REFERENCES patients (patient_id)
17 );
18


```

```

19  ✓ INSERT INTO patients (ferst_name, birthday) VALUES
20    ('Артур', '1980-05-15'),
21    ('Любава', '1990-07-22'),
22    ('Джони аля порк', '1975-12-03'),
23    ('Андрей', '1975-12-03'),|
24    ('Борисыч', '1975-12-03');
25
26  ✓ INSERT INTO appointments (patient_id, date_visit) VALUES
27    (1, '2024-11-01'),
28    (2, '2024-11-02'),
29    (3, '2024-11-03'),
30    (4, '2024-11-10'),
31    (5, '2024-11-10');
--
47  SELECT ferst_name FROM patients;
48  SELECT ferst_name FROM patients WHERE birthday <= CURRENT_DATE - INTERVAL '30 years';
49  SELECT p.ferst_name AS patient_first_name, a.date_visit AS date_patient_visit FROM appointments a INNER JOIN patients p USING (patient_id);
--

```

	ferst_name text 
1	Артур
2	Любава
3	Джони аля порк
4	Андрей
5	Борисыч

	ferst_name text 
1	Артур
2	Джони аля порк
3	Андрей
4	Борисыч

	patient_first_name text	date_patient_visit date
1	Артур	2024-11-01
2	Любава	2024-11-02
3	Джони аля порк	2024-11-03
4	Андрей	2024-11-10
5	Борисыч	2024-11-10

Задание 9 Транспортная компания. Предметная область: Управление доставками.

```

1  DROP TABLE IF EXISTS deliveries CASCADE;
2  DROP TABLE IF EXISTS drivers CASCADE;
3
4  CREATE TABLE deliveries (
5      deliverie_id SERIAL,
6      date DATE,
7      prise int,
8      PRIMARY KEY (deliverie_id)
9  );
10
11 CREATE TABLE drivers (
12     drivers_id UUID DEFAULT gen_random_uuid(),
13     deliverie_id INT NOT NULL,
14     name TEXT NOT NULL,
15     number_lisenzia int NOT NULL,
16     PRIMARY KEY (drivers_id),
17     FOREIGN KEY (deliverie_id) REFERENCES deliveries (deliverie_id)
18 );
19
20 INSERT INTO deliveries (date, prise) VALUES
21 ('2024-04-01', 150),
22 ('2024-05-02', 200),
23 ('20220-04-02', 500),
24 ('2026-05-05', 700),
25 ('1245-04-03', 55555);
26
27 INSERT INTO drivers (deliverie_id, name, number_lisenzia) VALUES
28 (1, 'Артур', 123456),
29 (2, 'Русллана', 654321),
30 (3, 'Андрей', 522511),
31 (4, 'Борисыч', 641212),
32 (5, 'Любава', 112233);
33
34
35 SELECT * FROM deliveries;
36 SELECT * FROM deliveries WHERE prise > 2000;
37 SELECT dl.deliverie_id, dl.date, dl.prise, dr.name FROM deliveries dl INNER JOIN drivers dr USING (deliverie_id);
38

```

	deliverie_id [PK] integer	date date	prise integer
1	1	2024-04-01	150
2	2	2024-04-02	200
3	3	2024-04-03	180
4	4	2024-04-01	150
5	5	2024-05-02	200

	deliverie_id [PK] integer	date date	prise integer
1	8	1245-04-03	55555

	deliverie_id integer	date date	prise integer	name text
1	1	2024-04-01	150	Артур
2	2	2024-04-02	200	Руслана
3	3	2024-04-03	180	Любава
4	1	2024-04-01	150	Артур
5	2	2024-04-02	200	Руслана

Задание 10 Турфирма. Предметная область: Управление турами.

```

1  DROP TABLE IF EXISTS tours CASCADE;
2  DROP TABLE IF EXISTS clients CASCADE;
3
4  CREATE TABLE tours (
5      tour_id SERIAL,
6      direction TEXT,
7      prise int,
8      PRIMARY KEY (tour_id)
9  );
10
11 CREATE TABLE clients (
12     clients_id UUID DEFAULT gen_random_uuid(),
13     tour_id INT NOT NULL,
14     ferst_name TEXT NOT NULL,
15     pasport_date TEXT NOT NULL,
16     PRIMARY KEY (clients_id),
17     FOREIGN KEY (tour_id) REFERENCES tours (tour_id)
18 );
19

```



```

20 ▾ INSERT INTO tours (direction, prise) VALUES
21   ('Москва - Санкт-Петербург', 500000),
22   ('Краснодар - Сочи', 3000),
23   ('Новосибирск - Омск', 4000),
24   ('Казань - Нижний Новгород', 3500),
25   ('Екатеринбург - Челябинск', 4500);
26
27 ▾ INSERT INTO clients (tour_id, ferst_name, pasport_date) VALUES
28   (1, 'Артур', '2020-05-10'),
29   (2, 'Артурио', '2019-08-15'),
30   (3, 'Васидильдо', '2018-11-20'),
31   (4, 'Борисыч', '2021-02-05'),
32   (5, 'Любава', '2017-07-30');
33
35 SELECT * FROM tours;
36 SELECT * FROM tours WHERE prise > 50000;
37 SELECT t.tour_id, t.direction, t.prise, c.ferst_name FROM clients c INNER JOIN tours t USING (tour_id);
38

```

	tour_id [PK] integer	direction text	prise integer
1	1	Москва - Санкт-Петербу...	5000
2	2	Краснодар - Сочи	3000
3	3	Новосибирск - Омск	4000
4	4	Казань - Нижний Новгор...	3500
5	5	Екатеринбург - Челябинск	4500

	tour_id [PK] integer	direction text	prise integer
1	1	Москва - Санкт-Петербург	500000

	tour_id integer	direction text	prise integer	ferst_name text
1	1	Москва - Санкт-Петербу...	500000	Артур
2	2	Краснодар - Сочи	3000	Артурио
3	3	Новосибирск - Омск	4000	Васидильдо
4	4	Казань - Нижний Новгор...	3500	Борисыч
5	5	Екатеринбург - Челябинск	4500	Любава

Вывод: в ходе работы мы отработали навыки создания таблиц на языке запросов SQL, заполнение их данными, а также создали и выполнили простые запросы по выборке из них данных.