Федеральное государственное автономное образовательное учреждение

высшего образования «Национальный исследовательский университет

ИТМО»

*Факультет программной инженерии и компьютерной техники*

*Направление подготовки: 09.03.04 - Нейротехнологии и программирование*

*Дисциплина «Информатика»*

**Отчет**

**По лабораторной работе №4. «Введение в базы данных»**

Выполнил:

Мухин Никита Денисович

Группа: Р3123

Преподаватель:

Болдырева Елена Александровна

Г. Санкт-Петербург, 2023 г

# **Задание**

# 

1. Проделайте все шаги из примера выше. Удостоверьтесь, что Вы поняли, как

это работает.

1. Самостоятельно выберите тематику для своей собственной базы данных.

База данных должна содержать не менее четырех таблиц. Каждая таблица должна иметь минимум одну связь с другими таблицами для формирования сложных запросов. Каждая таблица содержит не менее 5 записей.

1. Нарисуйте схему базы данных как в примере с указанием связей между таблицами.
2. Создайте таблицы с помощью Python для SQLite и MySQL.
3. Добавьте по одной новой записи в каждую из таблиц Вашей базы данных.
4. Продемонстрируйте работу запросов на извлечение данных:

- выбрать все записи из таблиц,

- составить запрос по извлечению данных с использованием JOIN

- составить запрос по извлечению данных с использованием WHERE и GROUP BY

Составить два запроса, в которых будет вложенный SELECT-запрос (вложение с помощью WHERE.

- составить 2 запроса с использованием UNION (объединение запросов).

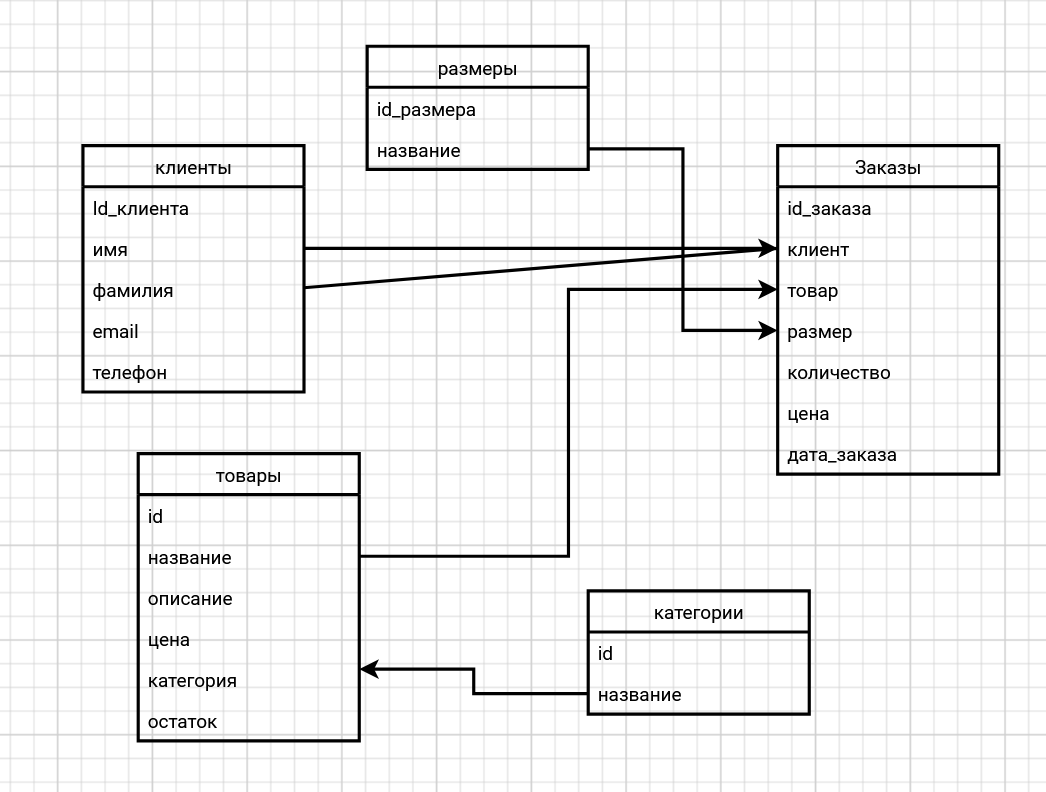
Составить 1 запрос с использованием DISTINCT. Если для демонстрации работы этого ключевого слова

недостаточно данных – предварительно дополните таблицу.

1. Обновить две записи в двух разных таблицах Вашей базы данных
2. Удалить по одной записи из каждой таблицы.
3. Удалите все записи в одной из таблиц.
4. Сформируйте отчет

# **Выполнение работы:**

**Схема базы данных:**



**SQLite**

import sqlite3  
  
# Подключение к базе данных  
conn = sqlite3.connect('siren.db')  
  
# Создание курсора  
cur = conn.cursor()  
  
  
# Создание таблиц  
cur.execute('''  
CREATE TABLE IF NOT EXISTS категории (  
 id INTEGER PRIMARY KEY,  
 название VARCHAR(255)  
)  
''')  
  
cur.execute('''  
CREATE TABLE IF NOT EXISTS товары (  
 id INTEGER PRIMARY KEY,  
 название INTEGER,  
 описание TEXT,  
 цена FLOAT,  
 категория INTEGER,  
 остаток INTEGER,  
 FOREIGN KEY (категория) REFERENCES категории (id)  
)  
''')  
  
cur.execute('''  
CREATE TABLE IF NOT EXISTS клиенты (  
 id\_клиента INTEGER PRIMARY KEY,  
 имя VARCHAR(255),  
 фамилия VARCHAR(255),  
 email VARCHAR(255),  
 телефон VARCHAR(255)  
)  
''')  
  
cur.execute('''  
CREATE TABLE IF NOT EXISTS размеры (  
 id\_размера INTEGER PRIMARY KEY,  
 название VARCHAR(255)  
)  
''')  
  
cur.execute('''  
CREATE TABLE IF NOT EXISTS заказы (  
 id\_заказа INTEGER PRIMARY KEY,  
 клиент INTEGER REFERENCES клиенты (id\_клиента),  
 товар INTEGER REFERENCES товары (id),  
 размер INTEGER REFERENCES размеры (id\_размера),  
 количество INTEGER,  
 цена FLOAT,  
 дата\_заказа DATE  
 )  
''')  
  
cur.execute('''  
CREATE TABLE IF NOT EXISTS магазины (  
 id INTEGER PRIMARY KEY,  
 название VARCHAR(255),  
 адрес VARCHAR(255),  
 телефон VARCHAR(255),  
 город VARCHAR(255)  
)  
''')  
  
  
# Заполнение таблицы sizes данными  
sizes\_data = [('XS',), ('S',), ('M',), ('L',), ('XL',)]  
  
cur.executemany('''  
INSERT INTO размеры (название)  
VALUES (?)  
''', sizes\_data)  
  
# Заполнение таблицы shops данными  
shops\_data = [  
 ('Сирень Москва', 'г. Москва, ул. Тверская, д. 12, стр. 2', '+7 (495) 123-45-67', 'Москва'),  
 ('Сирень', 'г. Санкт-Петербург, ул. Большая Конюшенная, д. 25', '+7 (812) 987-65-43', 'Санкт-Петербург'),  
 ('Сирень Казань', 'г. Казань, ул. Профсоюзная, д. 42', '+7 (843) 567-89-01', 'Казань')  
]  
  
cur.executemany('''  
INSERT INTO магазины (название, адрес, телефон, город)  
VALUES (?, ?, ?, ?)  
''', shops\_data)  
  
# Вставка данных в таблицу items  
items\_data = [  
 ('Футболка Сирень', 'Фирменная оверсайз футболка с нашивкой на груди. Состав:100% хлопок', 2999.0, 1, 12),  
 ('Футболка Красная Лилия', 'Оверсайз футболка с рисунком на спине. Состав:100% хлопок', 2999.0, 1, 43),  
 ('Футболка Темное прошлое', 'Оверсайз футболка с готическими элементами. Состав:100% хлопок', 2999.0, 1, 35),  
 ('Лонгслив Сирень', 'Фирменный оверсайз лонгслив с нашивкой на груди. Состав:100% хлопок', 4999.0, 2, 2),  
 ('Лонгслив Футуризм', 'Оверсайз лонгслив в стиле футуризма. Состав:60% хлопок 40% полиэстр', 4999.0, 2,13),  
 ('Лонгслив Восстание машин', 'Лимитированный оверсайз лонгслив. Состав: 100% полиэстер', 4999.0, 2, 9),  
 ('Рубашка Сирень', 'Фирменная оверсайз рубашка с нашивкой на груди. Состав:100% хлопок', 4499.0, 3, 23),  
 ('Рубашка Лотос', 'Фирменная оверсайз рубашка с нашивкой на груди. Состав:100% хлопок', 4499.0, 3, 40),  
 ('Рубашка Вечная ночь', 'Лимитированная оверсайз рубашка с вышиакой на спине. Состав:100% хлопок', 4499.0, 3, 54),  
 ('Свитшот Сирень', 'Фирменный оверсайз свитшот с нашивкой на груди. Состав:100% хлопок', 5699.0, 4, 4),  
 ('Свитшот Покой', 'Готический оверсайз свитшот. Состав: 55% хлопок 45% полиэстер', 5699.0, 4, 6),  
 ('Свитшот Препятствие', 'Оверсайз свитшот из старой коллекции. Количество ограничено. Состав: 100% хлопок', 5699.0, 4,1),  
 ('Джемпер Сирень', 'Фирменный оверсайз джемпер с нашивкой на груди. Состав:100% шерсть', 6699.0, 5, 17),  
 ('Джемпер Холод', 'Оверсайз джемпер в холодных оттенках. Состав: 100% хлопок', 6699.0, 5, 23),  
 ('Джемпер Изморозь', 'Альтернативная версия джемпера "Холод"(более теплая). Состав: 30% шерсть 70% полиэтстер', 6699.0, 5, 15),  
 ('Худи Сирень', 'Фирменнре оверсайз худи с нашивкой на груди. Состав:100% хлопок', 4999.0, 6, 5),  
 ('Худи Соната заката', 'Худи, выполненное в коллаборации с брендом Волчок. Limited Edition. Состав: 100% хлопок', 4999.0, 6, 14),  
 ('Худи Призрак', 'Худи из новой колекции "Сновидения". Состав: 55% хлопок 45% полиэстер', 4999.0, 6, 2),  
 ('Брюки Сирень', 'Фирменные оверсайз брюки с нашивкой на карманах. Состав:100% хлопок', 5699.0, 7, 19),  
 ('Брюки Черные', 'Повседневные оверсайз брюки. Состав:95% хлопок 5% полиэстер', 5699.0, 7, 12),  
 ('Брюки Карго', 'Обычные штаны для работяг. Состав: 100% хлопок', 5699.0, 7, 30),  
]  
  
for item\_data in items\_data:  
 cur.execute('''  
 INSERT INTO товары (название, описание, цена, категория, остаток)  
 VALUES (?, ?, ?, ?, ?)  
 ''', item\_data)  
  
# Вставка данных в таблицу categories  
categories = ['ФУТБОЛКИ', 'ЛОНГСЛИВЫ', 'РУБАШКИ', 'СВИТШОТЫ', 'ДЖЕМПЕРЫ', 'ХУДИ', 'БРЮКИ / ДЖИНСЫ / ШОРТЫ']  
for category\_name in categories:  
 cur.execute('''  
 INSERT INTO категории (название)  
 VALUES (?)  
 ''', (category\_name,))  
  
# Вставка данных в таблицу orders  
order\_data = [  
 (1, 1, 2, 1, 1, 4999, '2023-11-01'),  
 (2, 2, 1, 5, 2, 11398, '2023-11-04'),  
 (3, 3, 3, 2, 1, 6699, '2023-11-10'),  
 (4, 4, 13, 1, 2, 9998, '2023-11-12'),  
 (5, 5, 18, 4, 2, 11398, '2023-11-23'),  
 (6, 6, 7, 4, 2, 13398, '2023-11-21'),  
 (7, 7, 19, 5, 3, 14997, '2023-11-17'),  
 (8, 8, 15, 2, 2, 9998, '2023-11-14'),  
 (9, 9, 14, 2, 2, 11398, '2023-11-07'),  
 (10, 10, 21, 1, 2, 11998, '2023-11-15'),  
 (11, 11, 5, 4, 2, 13398, '2023-11-29')  
]  
  
cur.executemany('''  
INSERT INTO заказы (id\_заказа, клиент, товар, размер, количество, цена, дата\_заказа)  
VALUES (?, ?, ?, ?, ?, ?, ?)  
''', order\_data)  
  
  
clients\_data = [  
 ('Анастасия', 'Петрова', 'lostlost14124@mail.ru', '+79434567892'),  
 ('Дмитрий', 'Смирнов', 'rachedlife.io@mail.ru', '+79534567893'),  
 ('Екатерина', 'Иванова', 'gdldfgbbhh@yandex.ru', '+79634567894'),  
 ('Константин', 'Волков', 'lifeof23sas@mail.ru', '+79734567895'),  
 ('Мария', 'Смирнова', 'strisland2@mail.ru', '+79834567896'),  
 ('Олег', 'Петров', 'mailmyma1l@yandex.ru', '+79934567897'),  
 ('Ольга', 'Семенова', 'randoofran342@mail.ru', '+78034567898'),  
 ('Павел', 'Смирнов', 'buybuybuy2452@mail.ru', '+78134567900'),  
 ('Татьяна', 'Иванова', 'tatyIvo@yandex.ru', '+78234567901'),  
 ('Федор', 'Волков', 'VolkoVolco@yandex.ru', '+78334567902'),  
 ('Юлия', 'Петрова', 'july312dJuly@mail.ru', '+78434567903')  
]  
  
cur.executemany('''  
INSERT INTO клиенты (имя, фамилия, email, телефон)  
VALUES (?, ?, ?, ?)  
''', clients\_data)  
  
# Выполнение запроса  
cur = conn.cursor()  
cur.execute('''  
UPDATE заказы  
SET клиент = (SELECT имя || " " || фамилия FROM клиенты WHERE заказы.клиент = клиенты.id\_клиента)  
''')  
  
# Выполнение запроса  
cur = conn.cursor()  
cur.execute('''  
UPDATE заказы  
SET товар = (SELECT название FROM товары WHERE заказы.товар = товары.id)  
''')  
  
# Выполнение запроса  
cur = conn.cursor()  
cur.execute('''  
UPDATE заказы  
SET размер = (SELECT название FROM размеры WHERE размер = размеры.id\_размера)  
''')  
  
# Выполнение запроса  
cur = conn.cursor()  
cur.execute('''  
UPDATE товары  
SET категория = (SELECT название FROM категории WHERE товары.категория = категории.id)  
''')  
  
  
  
# Выбор всех записей из таблиц  
cur.execute('''  
SELECT \*  
FROM товары;  
''')  
print('Товары:')  
for row in cur:  
 print(row)  
  
cur.execute('''  
SELECT \*  
FROM категории;  
''')  
print('Категории:')  
for row in cur:  
 print(row)  
  
cur.execute('''  
SELECT \*  
FROM магазины;  
''')  
print('Магазины:')  
for row in cur:  
 print(row)  
  
cur.execute('''  
SELECT \*  
FROM клиенты;  
''')  
print('Клиенты:')  
for row in cur:  
 print(row)  
  
# Выбор данных с использованием JOIN  
cur.execute('''  
SELECT товары.название, категории.название  
FROM товары  
JOIN категории  
ON товары.категория = категории.id;  
''')  
print('Товары и категории:')  
for row in cur:  
 print(row)  
  
# Выбор данных с использованием WHERE и GROUP BY  
cur.execute('''  
SELECT товары.название, SUM(заказы.количество) AS количество\_проданных\_товаров  
FROM товары  
JOIN заказы  
ON товары.id = заказы.товар  
GROUP BY товары.название;  
''')  
print('Проданные товары:')  
for row in cur:  
 print(row)  
  
# Вложенные SELECT-запросы  
cur.execute('''  
SELECT заказы.дата\_заказа,  
 (SELECT товары.название  
 FROM товары  
 WHERE товары.id = заказы.товар) AS название\_товара,  
 заказы.количество  
FROM заказы;  
''')  
print('Заказы:')  
for row in cur:  
 print(row)  
  
# Объединение запросов (UNION)  
cur.execute('''  
SELECT \*  
FROM товары  
WHERE товары.цена < 5000;  
''')  
print('Товары дешевле 5000 рублей:')  
for row in cur:  
 print(row)  
  
cur.execute('''  
SELECT \*  
FROM товары  
WHERE товары.цена >= 5000;  
''')  
print('Товары дороже 5000 рублей:')  
for row in cur:  
 print(row)  
  
cur.execute('''  
SELECT \*  
FROM товары  
UNION  
SELECT \*  
FROM товары;  
''')  
print('Все товары:')  
for row in cur:  
 print(row)  
  
# DISTINCT  
cur.execute('''  
SELECT DISTINCT товары.название  
FROM товары;  
''')  
print('Уникальные названия товаров:')  
for row in cur:  
 print(row)  
  
# Обновление данных  
cur.execute('''  
UPDATE товары  
SET цена = 4599  
WHERE id = 1;  
''')  
print('Цена товара с ID 1 обновлена до 4599 рублей.')  
  
cur.execute('''  
UPDATE клиенты  
SET телефон = "+79341233252"  
WHERE id\_клиента;  
''')  
print('Номер телефона клиента с ID 1 обновлен на "+79341233252".')  
  
# Удаление данных  
cur.execute('''  
DELETE FROM товары  
WHERE id = 1;  
''')  
print('Товар с ID 1 удален.')  
  
cur.execute('''  
DELETE FROM клиенты  
WHERE id\_клиента = 1;  
''')  
print('Клиент с ID 1 удален.')  
  
cur.execute('''  
DELETE FROM магазины;  
''')  
print('Все магазины удалены.')  
  
# Сохранение изменений  
conn.commit()  
  
# Закрытие соединения  
conn.close()

**MYSQL**

import mysql.connector  
from mysql.connector import Error  
  
# Создание базы данных  
def create\_database():  
 connection = mysql.connector.connect(  
 host="localhost",  
 user="root",  
 password="password"  
 )  
  
 try:  
 cursor = connection.cursor()  
 cursor.execute("CREATE DATABASE attaadddataa")  
 except Error as e:  
 print(f"The error '{e}' occurred")  
 finally:  
 if connection.is\_connected():  
 connection.close()  
  
create\_database()  
  
# Подключение к базе данных  
conn = mysql.connector.connect(  
 host="localhost",  
 user="root",  
 password="password",  
 database="attaadddataa"  
)  
  
# Создание курсора  
cur = conn.cursor()  
  
  
  
# Создание таблиц  
cur.execute('''  
CREATE TABLE IF NOT EXISTS категории (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 название VARCHAR(255)  
)  
''')  
  
cur.execute('''  
CREATE TABLE IF NOT EXISTS товары (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 название VARCHAR(255),  
 описание TEXT,  
 цена FLOAT,  
 категория INT,  
 остаток INT  
)  
''')  
  
cur.execute('''  
CREATE TABLE IF NOT EXISTS клиенты (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 имя VARCHAR(255),  
 фамилия VARCHAR(255),  
 почта VARCHAR(255),  
 телефон VARCHAR(255)  
)  
''')  
  
# Create the размеры table if it doesn't exist  
cur.execute('''  
CREATE TABLE IF NOT EXISTS размеры (  
 id\_размера INT PRIMARY KEY,  
 название VARCHAR(255)  
)  
''')  
  
cur.execute('''  
CREATE TABLE IF NOT EXISTS заказы (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 клиент INT REFERENCES клиенты (id\_клиента),  
 товар INTEGER,  
 размер INTEGER REFERENCES размеры (id\_размера),   
 количество INT,  
 цена FLOAT,  
 дата\_заказа DATE,  
 FOREIGN KEY (товар) REFERENCES товары(id)  
)  
''')  
  
cur.execute('''  
CREATE TABLE IF NOT EXISTS магазины (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 название VARCHAR(255),  
 адрес VARCHAR(255),  
 телефон VARCHAR(255),  
 город VARCHAR(255)  
)  
''')  
  
  
# Insert data into the размеры table  
sizes\_data = [('XS', '1'), ('S', '2'), ('M', '3'), ('L', '4'), ('XL', '5')]  
  
cur.executemany('''  
INSERT INTO размеры (название, id\_размера)  
VALUES (%s, %s)  
''', sizes\_data)  
  
# Заполнение таблицы shops данными  
shops\_data = [  
 ('Сирень Москва', 'г. Москва, ул. Тверская, д. 12, стр. 2', '+7 (495) 123-45-67', 'Москва', 1),  
 ('Сирень', 'г. Санкт-Петербург, ул. Большая Конюшенная, д. 25', ': +7 (812) 987-65-43', 'Санкт-Петербург', 2),  
 ('Сирень Казань', 'г. Казань, ул. Профсоюзная, д. 42', '+7 (843) 567-89-01', 'Казань', 3),  
]  
  
cur.executemany('''  
 INSERT INTO магазины (название, адрес, телефон, город, id)  
 VALUES (%s, %s, %s, %s, %s)  
''', shops\_data)  
  
  
# Вставка данных в таблицу items  
items\_data = [  
 ('Футболка Сирень', 'Фирменная оверсайз футболка с нашивкой на груди. Состав:100% хлопок', 2999.0, 1, 12,1),  
 ('Футболка Красная Лилия', 'Оверсайз футболка с рисунком на спине. Состав:100% хлопок', 2999.0, 1, 43,2),  
 ('Футболка Темное прошлое', 'Оверсайз футболка с готическими элементами. Состав:100% хлопок', 2999.0, 1, 35,3),  
 ('Лонгслив Сирень', 'Фирменный оверсайз лонгслив с нашивкой на груди. Состав:100% хлопок', 4999.0, 2, 2,4),  
 ('Лонгслив Футуризм', 'Оверсайз лонгслив в стиле футуризма. Состав:60% хлопок 40% полиэстр', 4999.0, 2,13,5),  
 ('Лонгслив Восстание машин', 'Лимитированный оверсайз лонгслив. Состав: 100% полиэстер', 4999.0, 2, 9,6),  
 ('Рубашка Сирень', 'Фирменная оверсайз рубашка с нашивкой на груди. Состав:100% хлопок', 4499.0, 3, 23,7),  
 ('Рубашка Лотос', 'Фирменная оверсайз рубашка с нашивкой на груди. Состав:100% хлопок', 4499.0, 3, 40,8),  
 ('Рубашка Вечная ночь', 'Лимитированная оверсайз рубашка с вышиакой на спине. Состав:100% хлопок', 4499.0, 3, 54,9),  
 ('Свитшот Сирень', 'Фирменный оверсайз свитшот с нашивкой на груди. Состав:100% хлопок', 5699.0, 4, 4,10),  
 ('Свитшот Покой', 'Готический оверсайз свитшот. Состав: 55% хлопок 45% полиэстер', 5699.0, 4, 6,11),  
 ('Свитшот Препятствие', 'Оверсайз свитшот из старой коллекции. Количество ограничено. Состав: 100% хлопок', 5699.0, 4,1,12),  
 ('Джемпер Сирень', 'Фирменный оверсайз джемпер с нашивкой на груди. Состав:100% шерсть', 6699.0, 5, 17,13),  
 ('Джемпер Холод', 'Оверсайз джемпер в холодных оттенках. Состав: 100% хлопок', 6699.0, 5, 23,14),  
 ('Джемпер Изморозь', 'Альтернативная версия джемпера "Холод"(более теплая). Состав: 30% шерсть 70% полиэтстер', 6699.0, 5, 15,15),  
 ('Худи Сирень', 'Фирменнре оверсайз худи с нашивкой на груди. Состав:100% хлопок', 4999.0, 6, 5,16),  
 ('Худи Соната заката', 'Худи, выполненное в коллаборации с брендом Волчок. Limited Edition. Состав: 100% хлопок', 4999.0, 6, 14,17),  
 ('Худи Призрак', 'Худи из новой колекции "Сновидения". Состав: 55% хлопок 45% полиэстер', 4999.0, 6, 2,18),  
 ('Брюки Сирень', 'Фирменные оверсайз брюки с нашивкой на карманах. Состав:100% хлопок', 5699.0, 7, 19,19),  
 ('Брюки Черные', 'Повседневные оверсайз брюки. Состав:95% хлопок 5% полиэстер', 5699.0, 7, 12,20),  
 ('Брюки Карго', 'Обычные штаны для работяг. Состав: 100% хлопок', 5699.0, 7, 30, 21),  
]  
  
  
  
for item\_data in items\_data:  
 cur.execute('''  
 INSERT INTO товары (название, описание, цена,категория, остаток, id)  
 VALUES (%s, %s, %s, %s, %s, %s)  
 ''', item\_data)  
  
  
# Вставка данных в таблицу categories  
categories = [('ФУТБОЛКИ', 1), ('ЛОНГСЛИВЫ', 2), ('РУБАШКИ', 3), ('СВИТШОТЫ', 4), ('ДЖЕМПЕРЫ', 5), ('ХУДИ', 6), ('БРЮКИ / ДЖИНСЫ / ШОРТЫ', 7)]  
for category\_name, category\_id in categories:  
 cur.execute('''  
 INSERT INTO категории(название, id)  
 VALUES (%s, %s)  
 ''', (str(category\_name), category\_id))  
  
# Вставка данных в таблицу orders  
order\_data = [  
 (1, 1, 2, 1, 1, 4999, '2023-11-01'),  
 (2, 2, 1, 5, 2, 11398, '2023-11-04'),  
 (3, 3, 3, 2, 1, 6699, '2023-11-10'),  
 (4, 4, 13, 1, 2, 9998, '2023-11-12'),  
 (5, 5, 18, 4, 2, 11398, '2023-11-23'),  
 (6, 6, 7, 4, 2, 13398, '2023-11-21'),  
 (7, 7, 19, 5, 3, 14997, '2023-11-17'),  
 (8, 8, 15, 2, 2, 9998, '2023-11-14'),  
 (9, 9, 14, 2, 2, 11398, '2023-11-07'),  
 (10, 10, 21, 1, 2, 11998, '2023-11-15'),  
 (11, 11, 5, 4, 2, 13398, '2023-11-29')  
]  
  
  
  
cur.executemany('''  
INSERT INTO заказы (id, клиент, товар, размер, количество, цена, дата\_заказа)  
VALUES (%s, %s, %s, %s, %s, %s, %s)  
''', order\_data)  
  
  
# Выбор всех записей из таблиц  
cur.execute('''  
SELECT \*  
FROM товары;  
''')  
print('Товары:')  
for row in cur:  
 print(row)  
  
cur.execute('''  
SELECT \*  
FROM категории;  
''')  
print('Категории:')  
for row in cur:  
 print(row)  
  
cur.execute('''  
SELECT \*  
FROM магазины;  
''')  
print('Магазины:')  
for row in cur:  
 print(row)  
  
cur.execute('''  
SELECT \*  
FROM клиенты;  
''')  
print('Клиенты:')  
for row in cur:  
 print(row)  
  
# Выбор данных с использованием JOIN  
cur.execute('''  
SELECT товары.название, категории.название  
FROM товары  
JOIN категории  
ON товары.категория = категории.id;  
''')  
print('Товары и категории:')  
for row in cur:  
 print(row)  
  
# Выбор данных с использованием WHERE и GROUP BY  
cur.execute('''  
SELECT товары.название, SUM(заказы.количество) AS количество\_проданных\_товаров  
FROM товары  
JOIN заказы  
ON товары.id = заказы.id  
GROUP BY товары.название;  
''')  
print('Проданные товары:')  
for row in cur:  
 print(row)  
  
# Вложенные SELECT-запросы  
cur.execute('''  
SELECT заказы.дата\_заказа,  
 (SELECT товары.название  
 FROM товары  
 WHERE товары.id = заказы.id) AS название\_товара,  
 заказы.количество  
FROM заказы;  
''')  
print('Заказы:')  
for row in cur:  
 print(row)  
  
# Объединение запросов (UNION)  
cur.execute('''  
SELECT \*  
FROM товары  
WHERE товары.цена < 5000;  
''')  
print('Товары дешевле 5000 рублей:')  
for row in cur:  
 print(row)  
  
cur.execute('''  
SELECT \*  
FROM товары  
WHERE товары.цена >= 5000;  
''')  
print('Товары дороже 5000 рублей:')  
for row in cur:  
 print(row)  
  
cur.execute('''  
SELECT \*  
FROM товары  
UNION  
SELECT \*  
FROM товары;  
''')  
print('Все товары:')  
for row in cur:  
 print(row)  
  
# DISTINCT  
cur.execute('''  
SELECT DISTINCT товары.название  
FROM товары;  
''')  
print('Уникальные названия товаров:')  
for row in cur:  
 print(row)  
  
# Обновление данных  
cur.execute('''  
UPDATE товары  
SET цена = 4599  
WHERE id = 1;  
''')  
print('Цена товара с ID 1 обновлена до 4599 рублей.')  
  
cur.execute('''  
UPDATE клиенты  
SET телефон = "+79341233252"  
WHERE id = 1;  
''')  
print('Номер телефона клиента с ID 1 обновлен на "+79341233252".')  
  
# Удаление данных  
# Удаление заказов, связанных с товаром с ID 1  
cur.execute('''  
SELECT id  
FROM заказы  
WHERE товар = 1;  
''')  
  
order\_ids = cur.fetchall()  
  
for order\_id in order\_ids:  
 cur.execute('''  
 DELETE FROM заказы  
 WHERE id = %s;  
 ''', order\_id)  
  
# Удаление товара с ID 1  
cur.execute('''  
DELETE FROM товары  
WHERE id = 1;  
''')  
print('Товар с ID 1 удален.')  
  
cur.execute('''  
DELETE FROM магазины;  
''')  
print('Все магазины удалены.')  
  
# Сохранение изменений  
conn.commit()  
  
# Закрытие соединения  
conn.close()

# **Заключение**

В ходе лабораторной работы я познакомился с основами работы с базами данных в Python. Я научился создавать таблицы, добавлять и удалять записи, выполнять различные запросы к данным. Я самостоятельно разработал базу данных для интернет-магазина одежды.

Для работы с базами данных в Python я использовал две системы управления базами данных: SQLite и MySQL.

Я успешно создал таблицы в обеих системах управления базами данных и выполнил различные запросы к данным. Я также обновил и удалил записи в таблицах.

# Список литературы:

1. SQL за 20 минут https://proglib.io/p/sql-for-20-minutes

6 бесплатных ресурсов для практики в SQL https://robotdreams.cc/blog/178-6-besplatnyh-

resursov-dlya-praktiki-v-sql.

1. 20 вопросов и задач по SQL на собеседовании с ответами - https://vc.ru/life/443626-20-

voprosov-i-zadach-po-sql-na-sobesedovanii-s-otvetami.

1. SQL. Занимательные задачки - https://habr.com/ru/articles/461567/.
2. SQL. Задачки для тренировки - https://sql-academy.org/ru/trainer.