

**ФЕДЕРАЛЬНОЕ АГЕНСТВО ЖЕЛЕЗНОДОРОЖНОГО
ТРАНСПОРТА**

**Федеральное государственное бюджетное образовательное
учреждение высшего образования**

**«ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ Императора
Александра I»**

**Кафедра «ИНФОРМАЦИОННЫЕ И ВЫЧИСЛИТЕЛЬНЫЕ
СИСТЕМЫ»**

Дисциплина «Базы данных и управление данными»

ОТЧЁТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

«Разработка и наполнение схем баз данных»

Выполнил студент
Факультет: АИТ
Группа: ИВБ-211

Шефнер А.

Проверил: проф.

Хомоненко А. Д.

Санкт-Петербург

2024

Задание:

Выберите предметную область и процесс, для автоматизации которого будет разрабатываться база данных. Для выбранного процесса разработайте ER-модель и для неё напишите набор SQL-команд для создания и наполнения базы данных.

Требования к сложности разрабатываемой схемы:

- не меньше 4 entity-таблиц
- минимум одно отношение "многие-ко-многим"
- использование типа данных timestamp для описания процесса

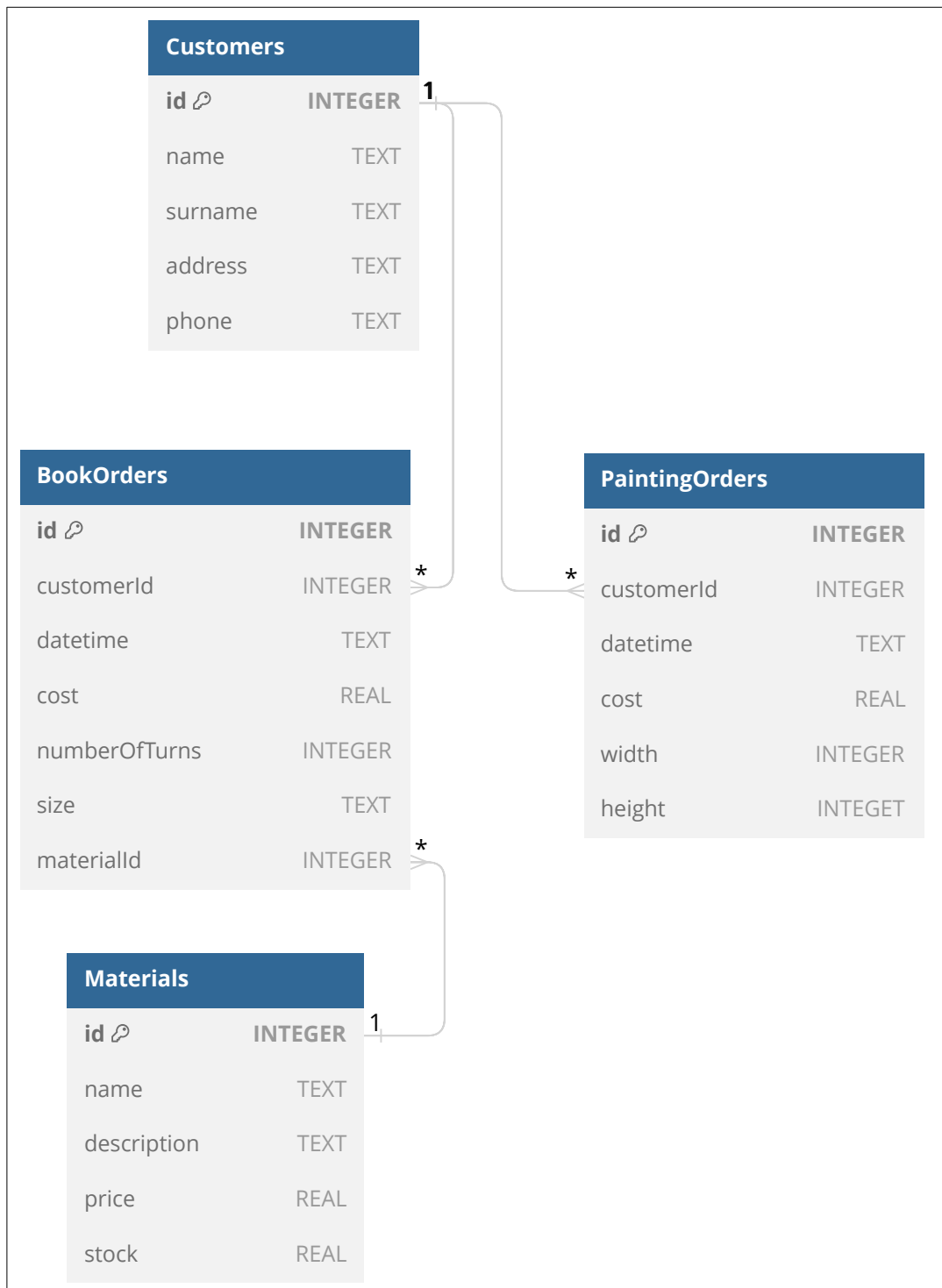
Отчёт по лабораторной работе должен содержать:

- краткое описание предметной области и моделируемого бизнес-процесса;
- ER-диаграмму схемы БД;
- DDL-команды создания и удаления объектов схемы.
- DML-команды заполнения, модификации и удаления данных из полученной базы.

Описание моделируемого процесса:

В данной модели представлены процессы взаимодействия клиентов и их заказов в магазине полиграфии. Клиент может заказать фотокнигу или картину. Фотокниги могут быть выполнены из разных материалов. Так же каждый заказ имеет свои размеры и стоимость.

ER-диаграмма:



Разработка базы данных:

Для работы с базой данных был выбран модуль sqlite3 языка программирования Python. Ниже представлена функция для создания базы данных.

```
def create_tables() → None:
    with get_connection() as conn:
        cur = conn.cursor()

        sql = """
        CREATE TABLE IF NOT EXISTS Customers (
            id INTEGER NOT NULL PRIMARY KEY,
            name TEXT NOT NULL,
            surname TEXT NOT NULL,
            address TEXT NOT NULL,
            phone TEXT NOT NULL
        );

        CREATE TABLE IF NOT EXISTS BookOrders (
            id INTEGER NOT NULL PRIMARY KEY,
            customerId INTEGER NOT NULL,
            datetime TEXT NOT NULL,
            cost REAL NOT NULL,
            numberOfTurns INTEGER NOT NULL,
            size TEXT NOT NULL,
            materialId INTEGER NOT NULL,
            FOREIGN KEY(customerId) REFERENCES Customers(id),
            FOREIGN KEY(materialId) REFERENCES Materials(id)
        );

        CREATE TABLE IF NOT EXISTS PaintingOrders (
            id INTEGER NOT NULL PRIMARY KEY,
            customerId INTEGER NOT NULL,
            datetime TEXT NOT NULL,
            cost REAL NOT NULL,
            width INTEGER NOT NULL,
            height INTEGER NOT NULL,
            FOREIGN KEY(customerId) REFERENCES Customers(id)
        );

        CREATE TABLE IF NOT EXISTS Materials (
            id INTEGER NOT NULL PRIMARY KEY,
            name TEXT NOT NULL,
            description TEXT NOT NULL,
            price REAL NOT NULL,
            stock REAL NOT NULL
        );
        """

        cur.executescript(sql)
```

Заполнение таблиц:

Для добавления записей в базу данных реализованы следующие функции:

```
def add_customer(customer: Customer) → None:
    with get_connection() as conn:
        cur = conn.cursor()
        sql = """
            INSERT INTO Customers
            (name, surname, address, phone)
            VALUES (?, ?, ?, ?)
        """
        cur.execute( sql, (customer.name, customer.surname,
customer.address, customer.phone))

def add_material(material: Material) → None:
    with get_connection() as conn:
        cur = conn.cursor()
        sql = """
            INSERT INTO Materials
            (name, description, price, stock)
            VALUES(?, ?, ?, ?)
        """
        cur.execute(sql, (material.name, material.description,
material.price, material.stock))

def add_book_order(order: BookOrder) → None:
    with get_connection() as conn:
        cur = conn.cursor()
        sql = """
            INSERT INTO BookOrders
            (customerId, datetime, cost, numberOfTurns, size,
materialId)
            VALUES (?, ?, ?, ?, ?, ?)
        """
        cur.execute(sql, (order.customer_id, order.datetime, order.cost,
order.number_of_turns, order.size, order.materialId))

def add_painting_order(order: PaintingOrder) → None:
    with get_connection() as conn:
        cur = conn.cursor()
        sql = """
            INSERT INTO PaintingOrders
            (customerId, datetime, cost, width, height)
            VALUES (?, ?, ?, ?, ?)
        """
        cur.execute(sql, (order.customer_id, order.datetime, order.cost,
order.width, order.height))
```

Далее скрипт, который наполняет таблицу данными, используя вышеприведённые функции:

```
customers = [  
    Customer("Виталий", "Петров", "ул. Пушкина, д. 22", "88005553535"),  
    Customer("Андрей", "Штефанов", "ул. Привокзальная, д. 12",  
"89811213123"),  
    Customer("Родион", "Базаров", "пр. Меншевииков, д. 143",  
"89814744747"),  
    Customer("Светлана", "Владимирова", "ул. Крутая, д. 121",  
"88001239098"),  
    Customer("Мёске", "Кобарёв", "ул. Японская, д. 211", "88005758495"),  
    Customer("Аристарх", "Попов", "пр. Энергетиков, д. 32",  
"88005767384"),  
]  
  
materials = [  
    Material("Крокодил", "Зелёная крокодиля кожа", 120, 24.6),  
    Material("Кожа", "Обычная коричневая кожа", 60, 36.1),  
    Material("Картон", "Печатный картон средней плотности", 23, 67),  
    Material("Пластик", "Эксперементальный прочный пластик", 100, 3.4),  
]  
  
book_orders = [  
    BookOrder(3, 3500, 12, "Большой", 1, "2024-01-06 12:40:21"),  
    BookOrder(1, 1400, 4, "Маленький", 2, "2024-01-08 16:12:55"),  
    BookOrder(4, 8000, 120, "Большой", 2, "2024-01-09 13:00:43"),  
    BookOrder(6, 6700, 18, "Средний", 4, "2024-01-09 13:43:26"),  
]  
  
painting_orders = [  
    PaintingOrder(4, 1500, 240, 127, "2024-01-09 13:00:43"),  
    PaintingOrder(2, 1800, 220, 140, "2024-01-10 13:12:31"),  
    PaintingOrder(5, 1000, 160, 100, "2024-01-10 14:40:61"),  
]  
  
for customer in customers:  
    db.add_customer(customer)  
  
for material in materials:  
    db.add_material(material)  
  
for order in book_orders:  
    db.add_book_order(order)  
  
for order in painting_orders:  
    db.add_painting_order(order)
```

Изменения в таблицах:

Пример функции для обновления количества материала на складе:

```
def update_material_stock(name: str, new_stock: float) → None:
    with get_connection() as conn:
        cur = conn.cursor()
        sql = "UPDATE Materials SET stock = ? WHERE name = ?"
        cur.execute(sql, [new_stock, name])
```

Пример функции для удаления информации о клиенте:

```
def remove_customer_by_id(id: int) → None:
    with get_connection() as conn:
        cur = conn.cursor()
        sql = "DELETE FROM Customers WHERE id = ?"
        cur.execute(sql, [id])
```

Результаты заполнения:

1	SELECT * From Customers				
	id	name	surname	address	phone
1	1	Виталий	Петров	ул. Пушкина, д. 22	88005553535
2	2	Андрей	Штефанов	ул. Привокзальная, д. 12	89811213123
3	3	Родион	Базаров	пр. Меньшевилов, д. 143	89814744747
4	4	Светлана	Владимирова	ул. Крутая, д. 121	88001239098
5	5	Мёске	Кобарёв	ул. Японская, д. 211	88005758495
6	6	Аристарх	Попов	пр. Энергетиков, д. 32	88005767384

1	SELECT * From BookOrders						
	id	customerId	datetime	cost	numberOfTurns	size	materialId
1	1	3	2024-01-06 12:40:21	3500.0	12	Большой	1
2	2	1	2024-01-08 16:12:55	1400.0	4	Маленький	2
3	3	4	2024-01-09 13:00:43	8000.0	120	Большой	2
4	4	6	2024-01-09 13:43:26	6700.0	18	Средний	4

1	SELECT * From PaintingOrders						
	id	customerId	datetime	cost	width	height	
1	1	4	2024-01-09 13:00:43	1500.0	240	127	
2	2	2	2024-01-10 13:12:31	1800.0	220	140	
3	3	5	2024-01-10 14:40:61	1000.0	160	100	
4	4	5	2024-03-04 00:38:57.838052	2000.0	160	120	

1	SELECT * From Materials				
	id	name	description	price	stock
1	1	Крокодил	Зелёная крокодиля кожа	120.0	24.6
2	2	Кожа	Обычная коричневая кожа	60.0	36.1
3	3	Картон	Печатный картон средней плотности	23.0	67.0
4	4	Пластик	Экспериментальный прочный пластик	100.0	3.4

Оценочный лист результатов ЛР №1

Ф.И.О. студента: Шефнер Альберт

Группа: ИВБ-211

№ п/п	Материалы необходимые для оценки знаний, умений и навыков	Показатель оценивания	Критерии Оценивания	Шкала оценивания	
1	Лабораторная работа №	Наличие заготовки	Присутствует	1	
			Отсутствует	0	
		Правильность ответов на вопрос	Получены правильные ответы на вопросы	4	
			Получены частично правильные ответы	2,5	
			Получены неправильные ответы	0	
		Соответствие методике выполнения	Соответствует	2,5	
			Не соответствует	1,5	
		Срок выполнения работы	Работа выполнена в срок	2,5	
			Работа выполнена с опозданием на 1 неделю	1,5	
			Работа выполнена с опозданием на 2 недели и более	1,0	
Итого максимальное количество баллов за лабораторную работу №			10		

Хомоненко А. Д.

«__»_____2024 г.