**Part 1: Выбор Сценария**

Для данной работы выбран сценарий: **Система бронирования в ресторане.** Эта система будет управлять бронированиями клиентов, столиками, персоналом и пунктами меню.

**Part 2: Проектирование Базы Данных и Документация** **Идентификация Сущностей и Атрибутов:**

1. Клиенты (Clients).

2. Столы (Tables).

3. Меню (Menu).

4. Персонал (Staff).

5. Бронирования (Bookings).

6. Заказы (Orders).

7. Детали заказа (Order\_details).

8. Бронирование столов (Booking\_tables).

**Проектирование Таблиц:**

**1.**  **Table Name: Clients**

○ **Description:** Хранит информацию о клиентах ресторана.

○ **Attributes:**

■ client\_id: INTEGER, PK, NOT NULL, UNIQUE

■ name: VARCHAR(50), NOT NULL

■ surname: VARCHAR(50), NOT NULL

■ phone: VARCHAR(20), NOT NULL

■ email: VARCHAR(50), UNIQUE

○ **Constraints:**

■ PK\_Clients: PRIMARY KEY (client\_id)

■ UQ\_ClientsFullName: UNIQUE (name, surname)

**2.**  **Table Name: Tables**

○ **Description:** Хранит информацию о столах ресторана.

○ **Attributes:**

■ table\_id: INTEGER, PK, NOT NULL, UNIQUE

■ number\_of\_table: INTEGER, NOT NULL, UNIQUE

■ number\_of\_clients: INTEGER, NOT NULL

■ location: VARCHAR(10), NOT NULL

■ status: VARCHAR(10), UNIQUE

○ **Constraints:**

■ PK\_Tables: PRIMARY KEY (table\_id)

**3.**  **Table Name: Menu**

○ **Description:** Хранит информацию о позициях меню.

○ **Attributes:**

■ menu\_id: INTEGER, PK, NOT NULL, UNIQUE

■ set\_name: VARCHAR(50), NOT NULL

■ title: VARCHAR(255), NOT NULL

■ price: NUMERIC, NOT NULL, CHECK (price >= 0)

■ availability: BOOLEAN

○ **Constraints:**

■ PK\_Menu: PRIMARY KEY (menu\_id)

**4.**  **Table Name: Staff**

○ **Description:** Хранит информацию о сотрудниках ресторана.

○ **Attributes:**

■ staff\_id: INTEGER, PK, NOT NULL, UNIQUE

■ name: VARCHAR(50), NOT NULL

■ surname: VARCHAR(50), NOT NULL

■ job\_title: VARCHAR(10), NOT NULL

○ **Constraints:**

■ PK\_Clients: PRIMARY KEY (staff\_id)

■ UQ\_ClientsFullName: UNIQUE (name, surname)

**5.**  **Table Name: Bookings**

○ **Description:** Хранит информацию о бронированиях.

○ **Attributes:**

■ booking\_id: INTEGER, PK, NOT NULL, UNIQUE

■ client\_id: INTEGER, FK

■ date\_time: TIMESTAMP

■ num\_of\_guests: VARCHAR(10), NOT NULL

○ **Constraints:**

■ PK\_Bookings: PRIMARY KEY (booking\_id)

■ FK\_Bookings\_Clients: FOREIGN KEY (client\_id) REFERENCES Clients(client\_id)

**6.**  **Table Name: Orders**

○ **Description:** Хранит информацию о заказах.

○ **Attributes:**

■ order\_id: INTEGER, PK, NOT NULL, UNIQUE

■ booking\_id: INTEGER, FK

■ staff\_id: INTEGER, FK

■ order\_time: TIMESTAMP

■ status: VARCHAR(10), NOT NULL

○ **Constraints:**

■ PK\_Orders: PRIMARY KEY (order\_id)

■ FK\_Orders\_Bookings: FOREIGN KEY (booking\_id) REFERENCES Bookings(booking\_id)

■ FK\_Orders\_Staff: FOREIGN KEY (staff\_id) REFERENCES Staff(staff\_id)

**7.**  **Table Name: Order\_details**

○ **Description:** Таблица для реализации связи многие-ко-многим между таблицами с информацией о заказах и заказанных позициях из меню.

○ **Attributes:**

■ order\_id: INTEGER, FK

■ menu\_id: INTEGER, FK

■ quantity: INTEGER, NOT NULL

○ **Constraints:**

■ FK\_Order­s\_Menu: FOREIGN KEY (order\_id) REFERENCES Orders(order\_id)

■ FK\_Orders\_Menu: FOREIGN KEY (menu\_id) REFERENCES Menu(menu\_id)

**8.**  **Table Name: Booking\_tables**

○ **Description:** Таблица для реализации связи многие-ко-многим между таблицами с информацией о бронировании и информацией о столиках ресторана.

○ **Attributes:**

■ table\_id: INTEGER, FK

■ booking\_id: INTEGER, FK

○ **Constraints:**

■ FK\_Bookings\_Tables: FOREIGN KEY (table\_id) REFERENCES Tables(table\_id)

■ FK\_Bookings\_Tables: FOREIGN KEY (booking\_id) REFERENCES Bookings(booking\_id)

**Взаимосвязи:**

● Clients и Bookings (Один-ко-Многим): один клиент может сделать множество бронирований, но каждое бронирование принадлежит только одному клиенту.

○ Bookings.client\_id является внешним ключом, ссылающимся на Clients.client\_id.

● Tables и Bookings (Многие-ко-Многим): одно бронирование может быть назначено на несколько столов (в ресторане отдыхает большая компания), один стол может быть забронирован неоднократно в разное время. Реализуется при помощи таблицы Booking\_tables.

○ Booking\_tables.table\_id является внешним ключом, ссылающимся на Tables.table\_id.

○ Booking\_tables.booking\_id является внешним ключом, ссылающимся на Bookings.booking\_id.

● Staff и Orders (Один-ко-Многим): один сотрудник (например, официант) может принять множество заказов, но каждый заказ принимается только одним сотрудником.

○ Orders.staff\_id является внешним ключом, ссылающимся на Staff.staff\_id.

● Bookings и Orders (Один-ко-Многим): одно бронирование может включить в себя множество заказов (несколько блюд), но каждый заказ, связанный с бронированием, относится только к одному бронированию.

○ Orders.booking\_id является внешним ключом, ссылающимся на Bookings.booking\_id.

● Orders и Menu (Многие-ко-Многим): один заказ может включать множество пунктов меню, один пункт меню может быть заказан во многих заказах. Реализуется при помощи таблица Orders\_details.

○ Orders\_details.menu\_id является внешним ключом, ссылающимся на Menu.menu\_id.

○ Orders\_details.order\_id является внешним ключом, ссылающимся на Orders.order\_id.

**Part 3: ER-Диаграмма**

