Выбор Сценария

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

Проектирование Базы Данных и Документация

**Идентификация сущностей и атрибутов**:

1. Клиенты(Clients)
2. Персонал (Staff)
3. Пункт меню(MenuItems)
4. Столик(Tables)
5. Бронирование клиентов(Booking management)
6. StaffTable: промежуточная сущность для связи персонала со столиком
7. Заказ(Orders): нужна для учета заказов
8. OrderItems: промежуточная сущность для связи заказа с пунктом меню

**Проектирование таблиц**:

1. **Table name: Clients**

* Description: Содержит информацию о посетителях ресторана
* Attributes:
  + - * + ClientID: INTEGER, NOT NULL, UNIQUE, PK
        + FirstName: VARCHAR(100), NOT NULL
        + LastName: VARCHAR(100), NOT NULL
        + PhoneNumber: VARCHAR(30)
* Constraints:
  + PK\_Clients: PRIMARY KEY(ClientID)
  + UQ\_ClientFullName: UNIQUE( FirstName, LastName)

1. **Table name: Staff**
   * + - Description: Содержит информацию о персонале
       - Attributes:
         * StaffID: INEGER, NOT NULL, UNIQUE, PK
         * FirstName: VARCHAR(100), NOT NULL
         * LastName: VARCHAR(100). NOT NULL
       - Constraints:
         * PK\_Staff: PRIMARY KEY(StaffID)
         * UQ\_StaffFullName: UNIQUE( FirstName, LastName)
2. **Table name: Tables**
   * Description: Содержит информацию о физических столиках
   * Attributes:
     + TableID: INEGER, NOT NULL, UNIQUE, PK
     + CountOfPeople: INEGER, NOT NULL
     + Status: VARCHAR(15), NOT NULL

Constraints:

* + - PK\_Table: PRIMARY KEY(TableID)

1. **Table name: MenuItems**
   * Description: Содержит информацию о пункте меню
   * Attributes:
     + MenuItemID: INEGER, NOT NULL, UNIQUE, PK
     + NameItem: VARCHAR(100), NOT NULL, UNIQUE
   * Constraints:
     + PK\_ MenuItem: PRIMARY KEY(MenuItemID)
     + UQ\_MenuItemName: UNIQUE(NameItem)
2. **Table name: BookingManagement**
   * Description: Таблица для учета бронирования столика
   * Attributes:
     + BookingID: INTEGER, PK, NOT NULL, UNIQUE
     + ClientID: INEGER, NOT NULL, FK( REFERENCES Clients)
     + TableID: INEGER, NOT NULL, FK( REFERENCES TAbles)
     + BookingDate: DATE, NOT NULL
     + BookingTime: TIME, NOT NULL
   * Constraints:
     + PK\_ BookingID: PRIMARY KEY(BookingID)
     + FK\_BookingManagment\_Client: FOREIGN KEY (ClientID) REFERENCES Clients(ClientID)
     + FK\_ BookingManagment \_Table: FOREIGN KEY (TableID) REFERENCES Tables(TableID)
3. **Table name: StaffTables**
   * Description: Сущность для распределения персонала по столикам.
   * Attributes:
     + StaffTableID: INTEGER, PK, NOT NULL, UNIQUE
     + StaffID: INEGER, NOT NULL, FK( REFERENCES Staff)
     + TableID: INEGER, NOT NULL, FK(REFERENCES Tables)
   * Constraints:
     + FK\_StaffTables\_Staff: FOREIGN KEY (StaffID) REFERENCES Staff(StaffID)
     + FK\_ StaffTables \_Tables: FOREIGN KEY (TableID) REFERENCES Tables(TableID)
     + PK\_StaffTable: PRIMARY KEY(StafftableID)
4. **Table name: Orders**
   * Description: Сущность хранит информацию о заказе с привязкой к столику
   * Attributes:
     + OrderID: INTEGER, PK, NOT NULL, UNIQUE
     + ClientID: INEGER,NOT NULL, FK( REFERENCES Clients)
     + TableID: INEGER, NOT NULL, FK(REFERENCES Tables)
     + OrderTime: TIMESTAMP, NOT NULL
   * Constraints
     + PK\_Order: PRIMARY KEY(OrderID)
     + FK\_Order\_Client: FOREIGN KEY (ClientID) REFERENCES Clients(ClientID)
     + FK\_Order\_Table: FOREIGN KEY (TableID) REFERENCES Tables(TableID)
5. **Table name: OrderItems**
   * Description: Таблица хранит информацию о содержании заказа
   * Attributes:
     + OrderID: INTEGER, NOT NULL, FK( REFERENCES Orders)
     + MenuItemID: INEGER, NOT NULL, FK( REFERENCES MenuItems)
     + Count: INEGER, NOT NULL
   * Constraints:
     + FK\_OrderItems\_Orders: FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)
     + FK\_OrderItems\_MenuItems: FOREIGN KEY (MenuItemID) REFERENCES MenuItems(MenuItemID)

Взаимосвязи:

* + - **Clients** и **Tables** с помощью сущности **BookingManagement** (**Многие-ко-многим**): Один клиент может забронировать несколько столиков (например, для банкета или в разное время), а один столик может быть забронирован разными клиентами в разное время.
      * + BookingManagement.ClientID является внешним ключом, ссылающимся на Clients.ClientID
        + BookingManagement.TableID является внешним ключом, ссылающимся на Tables.TableID
    - **Staff и Tables** связаны с помощью **Stafftable**: Есть два варианта развития связи.
      * Политика ресторана предусматривает, что у одного столика может быть один официант. Тогда связь будет **один-ко-многим**: один официант обслуживает несколько столиков, но за каждым столиком закреплен только один официант.
      * Политика ресторана предусматривает, что столики могут обслуживаться несколькими ресторанами. Тогда связь получается **многие-ко-многим**: один столик могут обслуживать несколько официантов, так и один официант может обслуживать несколько столиков.
        + StaffTables.StaffID является внешним ключом, ссылающимся на Staff.StaffID.
        + StaffTables.TableID является внешним ключом, ссылающимся на Tables.TableID
    - **Client** и **MenuItem** связаны с помощью дополнительных сущностей **Orders** и **OrderItem** (**многие-ко-многим**): Один клиент может заказать несколько позиций меню, а одна позиция меню может быть заказана несколькими клиентами. Но в связи между **Clients** и **Orders** связь **один-ко-многим**: один клиент может сделать множество заказов, но один заказ не может быть у нескольких клиентах одновременно.
      * Связь Orders с Clients и Tables:
        + Orders.ClientID является внешним ключом, ссылающимся на Clients.ClientID
        + Orders.TableID является внешним ключом, ссылающимся на Tables.TableID
      * Связь OrderItems с Orders и MenuItems
        + OrderItems.OrderID является внешним ключом, ссылающимся на Orders.OrderID.
        + OrderItems.MenuItemID является внешним ключом, ссылающимся на MenuItems.MenuItemID.

ER-Диаграмма

