

## Выполнение задания «db\_normalized\_2».

Выполнил: Кусков Арсений

Сценарий: система бронирования в ресторане (управление бронированиями клиентов, столиками и персоналом).

### Описание таблиц.

*Имя таблицы:* customers.

*Описание:* содержит данные о клиентах, включая имя и контактный телефон.

*Атрибуты:* customer\_id (тип SERIAL, первичный ключ PK), customer\_name (тип varchar(51), обязательное поле), contact\_phone (тип varchar(14), обязательное поле, уникальное значение).

*Ограничения:* contact\_phone должен быть уникальным для каждого клиента.

*Имя таблицы:* dining\_tables.

*Описание:* содержит данные о столах, их вместимости и текущем статусе.

*Атрибуты:* table\_id (тип SERIAL, PK), table\_number (тип varchar(20), обязательное поле, уникальное значение), capacity (тип integer, обязательное поле), status (тип table\_status, кастомный enum, обязательное поле, значение по умолчанию 'available').

*Ограничения:* capacity не может быть меньше или равно 0, status принимает значения строго из перечисления (available, occupied, reserved, maintenance).

*Имя таблицы:* staff.

*Описание:* содержит данные о персонале ресторана.

*Атрибуты:* employee\_id (тип SERIAL, PK), employee\_name (тип varchar(51), обязательное поле).

*Ограничения:* длина имени сотрудника должна быть не менее 3 символов.

*Имя таблицы:* reservation.

*Описание:* хранит данные о бронях столов клиентами.

*Атрибуты:* res\_id (тип SERIAL, PK), customer\_id (тип integer, внешний ключ FK к таблице customers), employee\_id (тип integer, FK к таблице staff), reservation\_date (тип date, обязательное поле), reservation\_time (тип time, обязательное поле).

*Ограничения:* customer\_id и employee\_id ссылаются на соответствующие таблицы, дата и время брони не могут быть пустыми.

*Имя таблицы:* reservation\_tables.

*Описание:* разрешает отношение многие-ко-многим между бронированиями и столами.

*Атрибуты:* res\_table\_id (тип SERIAL, PK), res\_id (тип integer, FK к reservation), table\_id (тип integer, FK к dining\_tables).

*Ограничения:* комбинация res\_id и table\_id уникальна (UNIQUE), что исключает дублирование привязки одного стола к одному бронированию.

### **Описание связей.**

Связь между таблицами customers и reservation: один клиент (customers) может иметь множество бронирований (reservation), но каждое бронирование принадлежит только одному клиенту (мощность один-ко-многим 1:N).

Связь между staff и reservation: один сотрудник обслуживает множество бронирований, каждое бронирование обрабатывается одним сотрудником (мощность один-ко-многим 1:N).

Связь между таблицами reservation и dining\_tables: реализована через промежуточную таблицу reservation\_tables, одно бронирование может включать несколько столов, и один стол может входить в разные бронирования в разное время (мощность многие-ко-многим M:N).