Part 1: Выбор сценария

Для этой работы был взят сценарий: **Система бронирования в ресторане.** Эта система управляет клиентами, столиками, бронированиями, меню и предзаказанными блюдами

Part 2: Проектирование БД и Документация

**Идентификация сущностей и атрибутов:**

1. Клиенты (Customers) – хранит данные о клиентах ресторана
2. Столики (Restaurant tables) – хранит информацию о доступных столиках
3. Меню (Menu items) – хранит список блюд
4. Бронирование (Reservations) – регистрирует брони клиентов
5. Бронирование – столик (Reservation tables) – связующая таблица для реализации связи многие-ко-многим между бронированиями и столиками
6. Бронирование – блюда (Reservation menu items) – связующая таблица для реализации связи многие-ко-многим между бронированиями и блюдами

**Проектирование таблиц:**

1. Table Name: customers
   * Description: Хранит информацию о клиентах ресторана.
   * Attributes:

* *customer\_id:* INTEGER, PK, NOT NULL, UNIQUE
* *first\_name:* VARCHAR(30), NOT NULL
* *last\_name:* VARCHAR(30), NOT NULL
* *phone:* VARCHAR(20), NOT NULL, UNIQUE
* *email:* VARCHAR(50), UNIQUE
* *created\_at:* TIMESTAMP, DEFAULT now()
* Constraints:
* PK\_customers: PRIMARY KEY (customer\_id)
* UQ\_phone: UNIQUE (phone)
* UQ\_email: UNIQUE (email)

1. Table Name: restaurant\_tables

* Description: Содержит информацию о столиках.
* Attributes:
* *table\_id:* INTEGER, PK, NOT NULL, UNIQUE
* *table\_number:* INTEGER, NOT NULL, UNIQUE
* *capacity:* INTEGER, UNIQUE
* *location:* VARCHAR(30)
* *is\_active:* BOOL, DEFAULT true
* Constraints:
* PK\_restaurant\_tables: PRIMARY KEY (table\_id)
* UQ\_table\_number: UNIQUE (table\_number)
* UQ\_capacity: CHECK (capacity > 0)

1. Table Name: menu\_items

* Description: Хранит информацию о блюдах
* Attributes:
* *menu\_item\_id:* INTEGER, PK, NOT NULL, UNIQUE
* *name:* VARCHAR(100), NOT NULL, UNIQUE
* *price:* NUMERIC(8,2), NOT NULL
* *descripton:* TEXT
* *is\_available:* BOOL, DEFAULT true
* Constraints:
* PK\_MenuItems: PRIMARY KEY (menu\_item\_id)
* UQ\_menu\_item\_name: UNIQUE(Name)
* CHK\_price: CHECK (price >= 0)

1. Table Name: reservations

* Description: Хранит информацию о бронированиях клиентов.
* Attributes:
* *reservation\_id:* INTEGER, PK, NOT NULL, UNIQUE
* *customer\_id:* INTEGER, FK (REFERENCES customers.customer\_id), NOT NULL
* *reservation\_start:* TIMESTAMP, NOT NULL
* *reservation\_end:* TIMESTAMP, NOT NULL
* *guests:* INTEGER, NOT NULL
* *status:* VARCHAR(20), NOT NULL, DEFAULT ‘booked’
* *notes:* TEXT
* *created\_at:* TIMESTAMP, DEFAULT now()
* Constraints:
* PK\_reservations: PRIMARY KEY (reservation\_id)
* FK\_reservations\_customers: FOREIGN KEY (customer\_id) REFERENCES customers (customer\_id)
* CHK\_guests: CHECK (guests > 0)
* CHK\_dates: CHECK (reservation\_end > reservation\_start)
* CHK\_status: CHECK (status IN ('booked','confirmed','seated','cancelled','completed'))

1. Table Name: reservation\_tables

* Description: Реализует связь многие-ко-многим между бронированиями и столиками.
* Attributes:
* *reservation\_id:* INTEGER, FK (REFERENCES reservations.reservations\_id), NOT NULL
* *table\_id:* INTEGER, FK (REFERENCES restaurant\_tables.table\_id), NOT NULL
* *reserved\_seats:* INTEGER
* Constraints:
* PK\_reservation\_tables: PRIMARY KEY (ReservationID, TableID)
* FK\_rt\_reservations: FOREIGN KEY (reservation\_id) REFERENCES reservations(reservation\_id)
* FK\_rt\_tables: FOREIGN KEY (table\_id) REFERENCES restaurant\_tables(table\_id)
* CHK\_ reserved\_seats: CHECK (reserved\_seats > 0)

1. Table Name: reservation\_menu\_items

* Description: Реализует связь многие-ко-многим между бронированиями и пунктами в меню.
* Attributes:
* *reservation\_menu\_item\_id:* INTEGER, PK, NOT NULL, UNIQUE
* *reservation\_id:* INTEGER, FK (REFERENCES reservations.reservation\_id), NOT NULL
* *menu\_item\_id:* INTEGER, FK (REFERENCES menu\_items.menu\_item\_id), NOT NULL
* *quantity:* INTEGER, NOT NULL, DEFAULT 1
* Constraints:
* PK\_ reservation\_menu\_items: PRIMARY KEY (reservation\_menu\_item\_id)
* FK\_rmi\_reservations: FOREIGN KEY (reservation\_id) REFERENCES reservations(reservation\_id)
* FK\_rmi\_menu\_items: FOREIGN KEY (menu\_item\_id) REFERENCES menu\_items (menu\_item\_id)
* CHK\_quantity: CHECK (quantity > 0)

**Взаимосвязи:**

* **customers и reservations (один-ко-многим)**

Один клиент может иметь несколько бронирований, но каждое бронирование относится к одному клиенту.

*reservations.customer\_id это FK на customers.customer\_id*

* **reservations и restaurant\_tables (многие-ко-многим)**

Одно бронирование может занимать несколько столиков, и один столик может быть забронирован в разные дни, часы.

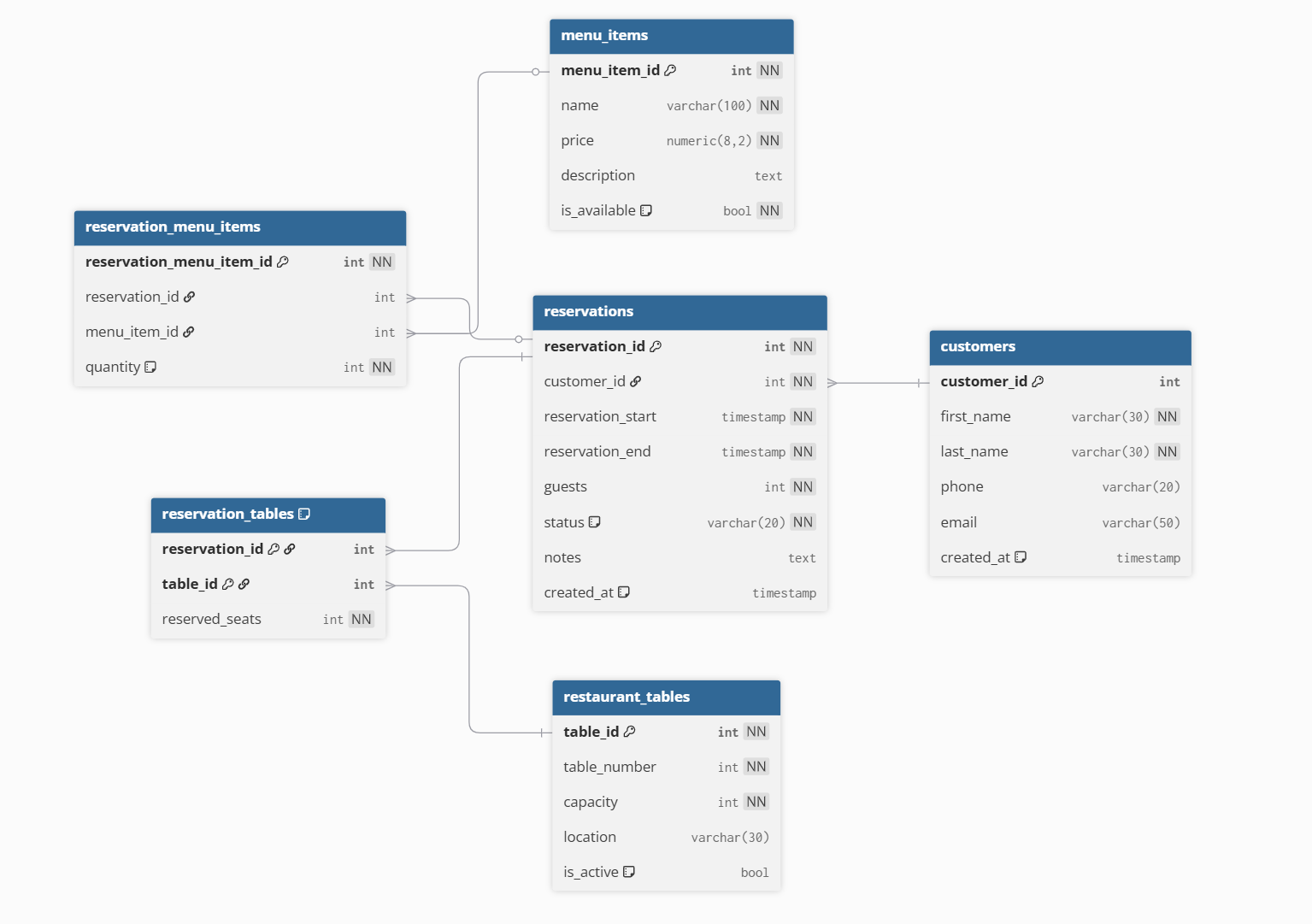
*Реализовано через таблицу reservation\_tables*

* **reservations и menu\_items (многие-ко-многим)**

В одном бронировании можно заказать несколько блюд, а блюдо может встречаться в разных бронированиях.

*Реализовано через таблицу reservation\_menu\_items*

Part 3: ER-Диаграмма

****