1.Выбранная область: Система бронирования в ресторане

Бизнес-процесс: Клиенты делают бронирование столиков на определённую дату и время. Бронирование может быть подтверждено, отменено или не выполнено (no-show). Персонал ресторана принимает и обрабатывает бронирования. В процессе участвуют столики, клиенты, сотрудники и временные параметры.

Почему это важно:

- Ресторану нужно понимать, когда и какие столики наиболее востребованы.

- Анализ поведения клиентов помогает строить программы лояльности.

- Персонал можно оценивать по количеству обработанных бронирований.

- Можно выявлять пиковые часы и дни, чтобы оптимизировать график работы.

2. Grain — это уровень детализации, на котором фиксируются данные в таблице фактов.

В нашем случае:

Одна строка в таблице фактов = одно бронирование столика.

То есть:

Каждое бронирование — это уникальное событие, которое содержит:

- Кто бронировал (клиент)

- Какой столик был выбран

- Когда было сделано бронирование

- Сколько гостей указано

- Кто из персонала обработал бронирование

- Каков статус бронирования (подтверждено, отменено, no-show)

3.

-- dim\_customer — таблица клиентов

-- Здесь храним информацию о каждом клиенте.

-- Это позволяет анализировать поведение, частоту визитов, лояльность и предпочтения.

CREATE TABLE dim\_customer (

customer\_id SERIAL PRIMARY KEY, -- Уникальный ID клиента

full\_name VARCHAR(100), -- Полное имя клиента

email VARCHAR(100), -- Email для связи и аналитики

phone VARCHAR(20), -- Телефон для подтверждений

birth\_date DATE -- Дата рождения — можно использовать для персональных акций

);

-- dim\_table — таблица столиков

-- Описывает физические столики в ресторане.

-- Важно для анализа загрузки, популярности и оптимизации рассадки.

CREATE TABLE dim\_table (

table\_id SERIAL PRIMARY KEY, -- Уникальный ID столика

table\_number INT, -- Номер столика (виден в зале)

capacity INT, -- Максимальное количество гостей

location VARCHAR(50) -- Расположение: "зал", "терраса", "VIP"

);

-- dim\_time — таблица времени

-- Время — ключевой фактор для анализа.

-- Позволяет агрегировать данные по дням недели, месяцам, годам.

CREATE TABLE dim\_time (

time\_id SERIAL PRIMARY KEY, -- Уникальный ID времени

reservation\_date DATE, -- Дата бронирования

reservation\_time TIME, -- Время бронирования

weekday VARCHAR(10), -- День недели (например, "Понедельник")

month VARCHAR(10), -- Название месяца (например, "Март")

year INT -- Год — для анализа по годам

);

-- dim\_staff — таблица сотрудников

-- Храним информацию о сотрудниках, которые обрабатывают бронирования.

-- Это позволяет оценивать их эффективность и нагрузку.

CREATE TABLE dim\_staff (

staff\_id SERIAL PRIMARY KEY, -- Уникальный ID сотрудника

full\_name VARCHAR(100), -- Имя сотрудника

role VARCHAR(50) -- Должность: "менеджер", "официант", "администратор"

);

4.

-- fact\_reservation — таблица фактов

-- Сердце хранилища: здесь фиксируются все бронирования.

-- Каждая строка — это одно событие бронирования.

CREATE TABLE fact\_reservation (

reservation\_id SERIAL PRIMARY KEY, -- Уникальный ID бронирования

-- Внешние ключи — связи с измерениями

customer\_id INT REFERENCES dim\_customer(customer\_id), -- Кто бронировал

table\_id INT REFERENCES dim\_table(table\_id), -- Какой столик

time\_id INT REFERENCES dim\_time(time\_id), -- Когда

staff\_id INT REFERENCES dim\_staff(staff\_id), -- Кто обработал

guest\_count INT, -- Количество гостей, указанных в бронировании

-- Статус бронирования

-- Возможные значения: 'confirmed', 'cancelled', 'no-show'

status VARCHAR(20),

-- Время создания бронирования

reservation\_created\_at TIMESTAMP

);

5. Выбор: Star Schema (звёздная схема)

Почему:

- Простая структура

- Быстрая навигация между таблицами

- Удобна для аналитических запросов

Центр — таблица фактов fact\_reservation, к ней напрямую подключены измерения:

- dim\_customer

- dim\_table

- dim\_time

- dim\_staff

Это позволяет быстро строить отчёты, не углубляясь в вложенные связи.

6.

-- 1. Сколько бронирований было в каждом месяце

-- Цель: понять сезонность, когда ресторан наиболее загружен

SELECT dt.month, COUNT(\*) AS total\_reservations

FROM fact\_reservation fr

JOIN dim\_time dt ON fr.time\_id = dt.time\_id

GROUP BY dt.month

ORDER BY total\_reservations DESC;

-- 2. Самые активные клиенты

-- Цель: выявить VIP-клиентов, предложить им бонусы

SELECT dc.full\_name, COUNT(\*) AS bookings

FROM fact\_reservation fr

JOIN dim\_customer dc ON fr.customer\_id = dc.customer\_id

GROUP BY dc.full\_name

ORDER BY bookings DESC

LIMIT 5;

-- 3. Загрузка столиков

-- Цель: понять, какие столики наиболее популярны

SELECT dt.table\_number, COUNT(\*) AS usage\_count

FROM fact\_reservation fr

JOIN dim\_table dt ON fr.table\_id = dt.table\_id

GROUP BY dt.table\_number

ORDER BY usage\_count DESC;

-- . Процент отменённых бронирований

-- Цель: оценить стабильность бронирований и поведение клиентов

SELECT

ROUND(100.0 \* SUM(CASE WHEN status = 'cancelled' THEN 1 ELSE 0 END) / COUNT(\*), 2) AS cancellation\_rate

FROM fact\_reservation;

-- 5. Среднее количество гостей по дням недели

-- Цель: понять, когда приходят большие компании, а когда — пары

SELECT dt.weekday, AVG(fr.guest\_count) AS avg\_guests

FROM fact\_reservation fr

JOIN dim\_time dt ON fr.time\_id = dt.time\_id

GROUP BY dt.weekday

ORDER BY avg\_guests DESC;