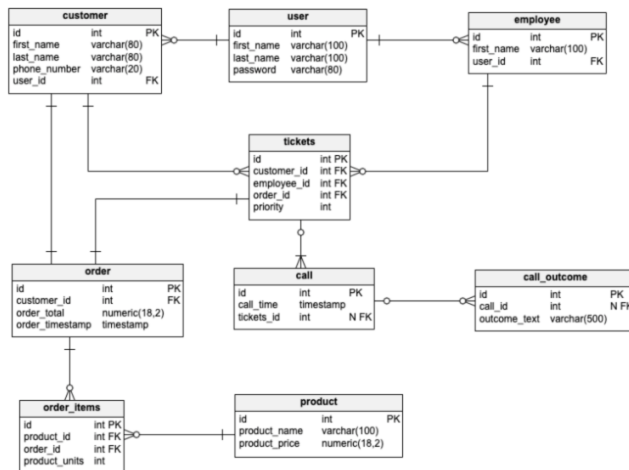


ETL. Семинар №3

1. Викторина
2. Блок 1

Задание 1

Ниже представлена схема базы данных приложения в котором клиент может размещать заказы и открывать тикеты для службы поддержки по своим заказам. Посмотрите схему базы данных, опишите структуру и дату которая хранится в этих таблицах.



15 минут

Решение:

- В таблице `customer` содержится основная информация о покупателях магазина.
- Таблица `user` содержит информацию для входа как для покупателей, так и для сотрудников магазина.
- В таблице `employee` содержится информация о сотрудниках магазина.
- В таблице `tickets` содержится информация о тикетах, отправленных клиентами по размещенным ими заказам.
- В таблице `call` содержится информация о звонках между клиентами и представителями службы поддержки по поводу тикетов.
- В таблице `call_outcome` содержится информация о результатах вызовов.
- В таблице `order` содержится основная информация о заказах, сделанных клиентами.

- Таблица `order_items` содержит информацию «многие ко многим», связывающую продукты, купленные клиентами в заказах. Она также хранит количество каждого заказанного продукта.
- В таблице `product` содержится информация о продуктах, которые могут заказать покупатели.

Задание 2

Предположим что у нас появилось две новых бизнес-задачи.

Во-первых, каждому представителю службы поддержки нужен быстрый доступ к некоторым показателям клиента. Для новой функциональности в приложении требуется информационная панель о клиенте с общим объемом продаж, количеством отправленных билетов и количеством обращений в службу поддержки клиентов.

Во-вторых, представителям службы поддержки нужна панель управления тикетами со списком тикетов и их соответствующими данными. Это нужно им для выбора тикета по адресу в зависимости от приоритета.

Создайте таблицу `customer_statistics` которая будет содержать информацию о покупательских привычках каждого клиента.

Сделайте схему таблицы используя app.diagrams.net и поделитесь картинкой в чате



Решение:

customer_statistics		
id	int	PK
customer_id	int	FK
orders_placed	int	
orders_total_amount	numeric(18,2)	
total_tickets_raised	int	
total_calls	int	

Каждый раз, когда через приложение вносятся изменения, новые данные сохраняются в соответствующих нормализованных таблицах и обновляется таблица `customer_statistics`. Например, когда клиент размещает новый заказ, `orders_total_amount` увеличивается на общую сумму нового заказа для этого клиента.

Задание 3

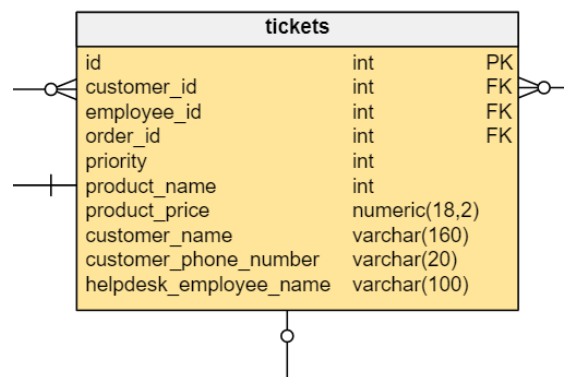
Измените таблицу tickets таким образом чтобы она соответствовала требованиям из предыдущего задания.

Представителям службы поддержки нужна панель управления тикетами со списком тикетов и их соответствующими данными. Это нужно им для выбора тикета по адресу в зависимости от приоритета.

Сделайте схему таблицы используя app.diagrams.net и поделитесь картинкой в чате

15 минут

Решение:



Панель управления тикетами для представителей службы поддержки. Мы денормируем данные в таблице ticket и добавляем дополнительные столбцы, содержащие нужную нам информацию. Затем приложение считывает всю сводную информацию из одной таблицы, не объединяя данные о клиентах, тикетах, звонках, заказах, заказах и продуктах. Это правда, что теперь у нас есть дублирующаяся информация о названиях продуктов, ценах, именах клиентов, номерах телефонов и т. д. Но время выполнения нашего запроса теперь намного короче, что и является целью.

Задание 4

Представитель службы поддержки заметил, что сумма, уплаченная за продукт, не соответствует цене, указанной в дашборде. Это произошло потому что с момента оплаты клиентом товара цена изменилась.

Подумайте что нужно изменить в исходной схеме (не создавая новых таблиц) чтобы решить описанную выше проблему.

Нарисуйте схему измененных таблиц используя app.diagrams.net и поделитесь картинкой в чате.

15 минут

Решение:

Для этого нам приходится дублировать `product_price` в таблице `order_items`, чтобы она сохранялась для каждого заказа. Теперь таблица `order_items` имеет следующую схему.

