

Project Name :- Zepto (Quick Commerce) SQL based Product Dissection

Done By :- Prajwal Shridhar Biradar

GitHub Link :-

https://github.com/prajwalbiradar21/Zepto-Product-Dissection-Capstone_project-

Video link :-





★ Product Dissection for Zepto

Company Overview:

Zepto is a rapidly growing **Quick Commerce** startup in India that specializes in **ultra-fast grocery delivery**, promising deliveries within **10–20 minutes**. Founded in **2021**, during-**Post Corona Lockdown Out-break** by Two Youngsters **Audit Palicha** and **Kaivalya Vohra**, Zepto has revolutionized the grocery retail space by leveraging **dark stores**, smart logistics, and hyperlocal delivery networks.

Zepto provides a fast and convenient shopping experience for customers through its **mobile app**. Users can easily browse products, add items to their cart, place orders, and receive deliveries within 10–20 minutes.

Zepto primarily focuses on delivering:

- Groceries
- Fruits & vegetables
- Dairy & bakery items
- Snacks & beverages
- Personal and household essentials

By combining **technology-driven inventory management**, **location intelligence**, and **efficient delivery routes**, Zepto ensures high-speed order fulfillment with a seamless customer experience.

Product Dissection and Real-World Problems Solved by Zepto :

Zepto is a fast-growing quick commerce platform that delivers groceries and daily essentials in just **10 to 20 minutes Across India in urban cities**. Let's break down its product and understand how it works and what real-life problems it solves.

Zepto's Product Breakdown (Dissection)

- **Order Management:**
Helps customers place orders, check order details, and track their order status easily.
- **Product Inventory:**
Maintains a list of all products, their categories, prices, and how many are in stock.
- **Delivery System:**
Assigns delivery personnel, tracks dispatch time, delivery time, and delivery status to make sure everything is on time.
- **Payments Module:**
Supports different payment methods like UPI, cards, wallets, and Cash on Delivery (COD), and tracks each transaction's status.
- **Customer Management:**
Stores customer details like name, phone, email, city, and order history for personalized service.
- **Feedback & Ratings:**
Collects customer reviews and ratings to improve service quality and delivery experience.

Case Study: Real-World Problems and Zepto's Innovative Solutions

In today's fast-moving world, customers expect their groceries and daily essentials to be delivered instantly.

Real-World Problems Solved by Zepto

Problem 1: Slow delivery from local stores

→ Now-a-days many Kirana(local Grocery),supermarkets and e-commerce stores take longer to deliver groceries ,due lack of workers/helpers ,etc .
Zepto solves this by delivering within 10–20 minutes using nearby micro-warehouses (dark stores) and Delivery partners.

Problem 2: Product not available in local shops

→ Local retail shops have limited space and inventory, which means they often run out of popular products or may not stock certain items at all. This leads customers to visit multiple stores or switch to online platforms that may take hours or even days to deliver.

Zepto stocks products in dark stores based on customer demand in each area.

Problem 3: Difficulty in tracking inventory

→ Customers often face situations where an item appears available while placing an order but later gets canceled due to stock issues.

Zepto Uses smart technology like AI-powered software to track product availability in real-time in their warehouses.

Problem 4: Poor delivery experience in traditional systems

→ Sometimes Orders are delayed due to inefficient order processing ,Unoptimized routes , Poor coordination between delivery partners and other platforms causing missing or failed deliveries.

Zepto Assigns delivery partners quickly and uses optimized routes for fast delivery.

Problem 5: Limited payment options and failed transactions

→ Offers multiple payment modes and tracks payment status smoothly.

Zepto ensures a **smooth and hassle-free payment experience** by supporting multiple payment options like UPI,debit/credit cards,net-banking and COD.

Problem 6: No proper feedback mechanism

→Many Platforms ignore customers' feedback , or do not have feedback option due to which customers are not satisfied with their service.

Zepto Collects customer feedback and ratings after every order to improve service.

Conclusion:

Zepto has changed the grocery shopping experience by solving everyday problems through **technology and customer thinking**. With a strong focus on speed and efficiency, it has changed the way people get their daily use essentials. By solving real problems with technology, Zepto ensures a smooth and hassle-free shopping experience.

Top Features of Zepto(Quick Commerce Platform) :

1. **10-Minute Delivery Promise:** Zepto's most standout feature is - it promises ultra_fast delivery (usually takes 10-20 max 30 minutes) using nearby micro-warehouses (known as dark stores). That helps customers/users to get items and essentials instantly.
2. **Hyperlocal Micro-Warehouse Network:** Zepto operates small warehouses in different local areas Mainly in Metro-cities. These warehouses are stocked based on local demand, making deliveries faster and more efficient. This helps reduce delivery distance and ensures quick restocking of popular products.

3. **Real-Time Order Tracking:** Customers can track their orders live on the application on their mobile, from order placement to delivery at their doorstep. This increases transparency and customer satisfaction.
4. **Product Categorization and Search:** The Zepto app has a user-friendly interface with smart product categorization (e.g., dairy, snacks, fruits, beverages, etc.) and powerful search filters. It helps customers/users quickly find what they need without scrolling too much.
5. **Flexible Payment Options:** Zepto supports various payment methods like UPI, Credit/Debit Cards, Wallets, and Cash on Delivery (COD). This ensures convenience for customers with different preferences.

Schema Description:

The database schema for Zepto involves multiple entities that represent different aspects of the platform. These entities include Customers, Products, Orders, Order_Items , Delivery_Partners, Feedback , and more. Each entity has specific attributes that describe its properties and relationships with other entities.

1. Customers Table/Entity:

Customers are the core users of Zepto. The Customers entity contains information about each Customer :

- **Customer_id - SERIAL - (Primary Key)** : A unique identifier for each Customer.
- **Name - VARCHAR(50)- NOT NULL -** : Name of the customer.
- **Email - VARCHAR(100) - UNIQUE, NOT NULL**: Email address of the customer .
- **Phone_number - VARCHAR(15)-UNIQUE CHECK(LENGTH(phone_number) >=10)** : The Customer's Contact number.
- **Address - TEXT -**: Full address of the customer.
- **City - VARCHAR(50) NOT NULL -** : City of the customer.
- **Registration_Date - DATE -NOT NULL**: The date when the user joined Instagram.

2. Products Table/Entity:

Products entity Stores information about each product, including **name, category, price, and stock quantity**. :

- **Product_id - SERIAL- (Primary Key)**: A unique identifier for each product .
- **Product_name - VARCHAR(100) NOT NULL -** : Name of the product.
- **Category - VARCHAR(100) NOT NULL -** : Product categories (e.g., Dairy, Snacks) .
- **Price - DECIMAL(10,2) -** :Price per unit.
- **Stock_quantity - INT -** : Stock available in the inventory.

3.Orders Table/Entity:

In Order Entity tracks each order placed by customer , including **order date, total amount, payment method, and delivery status.**

- **Order_id - serial - (Primary Key)**: A unique identifier for each order .
- **Customer_id - INT - (Foreign Key referencing customers Entity)**: order placed by specific customer .
- **Order_date - DATETIME -** :Date and time of order placement.
- **Total_amount - DECIMAL(10,2) -** : Total bill amount for the order .
- **Payment_method - VARCHAR(50) -NOT NULL** : Payment mode used (UPI, COD, etc.) .
- **Delivery_status - VARCHAR(20) DEFAULT 'PROCESSING' -** : Status (Processing ,Delivered, Cancelled, Pending).
- **Delivery_time - TIME -** : Time taken for delivery
- **Partner_id - INT - (Foreign Key referencing Delivery_Partners Entity)**:Which order was delivered by which delivery agent.--(Optional)

4.Order_Items Table/Entity:

It Represents the **many-to-many relationship** between **orders and products**.

- **Order_item_id - Serial- (Primary Key)**: A unique identifier for each loder_item.
- **Order_id - INT - (Foreign Key referencing orders Entity)**: Linked to Orders table.
- **Product_id - INT -(Foreign Key referencing products Entity)**: Linked to Products table.
- **Quantity - INT -** : Quantity of the product ordered
- **Item_price - DECIMAL(10,2) -**:Price of the item at time of purchase

5.Delivery_Partners Table/Entity:

Stores details of delivery personnel/agent, including **name, phone number, and assigned delivery zone..**

- **Partner_id - INT - (Primary Key)**: A unique identifier for each Delivery partner.
- **Name - VARCHAR(50) -**: Name of the delivery agent .
- **Phone_number-VARCHAR(15)-UNIQUE**
CHECK(LENGTH(agent_phone_number)>=10) : Contact number of Delivery agent.
- **Zone_assigned - VARCHAR(50) -**: Area/zone assigned for delivery to delivery agent

6. Payments Table/Entity:

It Records transactions related to each order, including **payment method, status, and at what time the transaction is done.**

- **Payment_id - SERIAL- (Primary Key)**: A unique identifier for each Payment.
- **Order_id - INT - (Foreign Key referencing Orders Entity)**: Linked to Orders table.
- **Payment_method - Varchar(50)-**: Payment method done by each user like UPI,cards ,COD ,Etc.

- **Payment_status**- Varchar(50) - : Status of each transaction Done.
- **Transaction_time** - TIMESTAMP-: Date and time of each transaction done.

7. Feedback Table/Entity:

Captures customer feedback for each order, including **ratings and reviews/comments**.

- **Feedback_id** - SERIAL- (**Primary Key**): A unique identifier for each feedback.
- **Customer_id** - INT - (**Foreign Key referencing Customers Entity**): Linked to Customers table.
- **Order_id** - INT -(**Foreign Key referencing orders Entity**): Linked to orders table.
- **Rating** - INT - CHECK (rating >=1 AND rating <= 5) : Ratings (1 TO 5).
- **Comments** - TEXT :: Customers Feedback .
- **Relationships are:**

1. **Customers Entity → Orders Entity** → One customer can place multiple orders.
 → Customer_id in **Orders** table is a **foreign key** referencing the Customers table.

2. **Orders Entity → Order_Items Entity** → Each order can have multiple items.
 → order_id in **Order_Items** table is a **foreign key** referencing **Orders** table .

3. **Products Entity → Order_Items entity** → Each product can appear in **multiple order items** (across different orders).
 → product_id in **Order_Items** table is a **foreign key** referencing **Products** table.

4. **Orders entity→ Feedback entity** → Each order can receive **one feedback entry**, or in some cases, customers may give **multiple feedback per order** (depending on design).
 → order_id in **Feedback** table is a **foreign key** referencing **Orders** table.

5. **Customers entity → Feedback entity** → One customer can give **multiple feedback entries** (for each order).
 → customer_id in **Feedback** table is a **foreign key** referencing the Customers table.

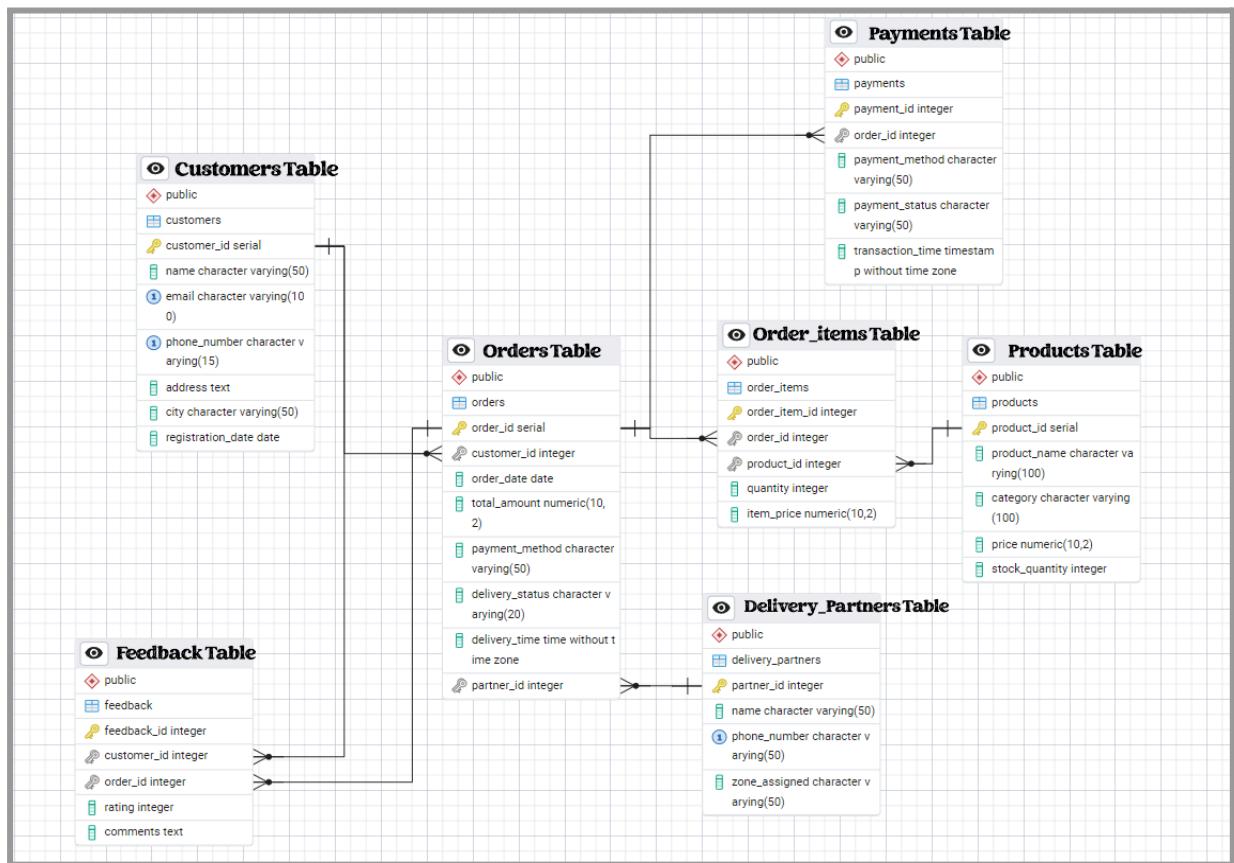
6. **Payments entity → Orders entity** → Linking Of payments table with each Order done .
 → order_id in **Payments** table is a **foreign key** referencing **Orders** table.

7.Delivery_Partners entity → Orders entity → One delivery partner can deliver many orders.

→ We can add a `partner_id` in the Orders table to link each order with the assigned delivery agent.

ER Diagram:

Let's construct an ER diagram that vividly portrays the relationships and attributes of the entities within the Zepto schema. This ER diagram will serve as a visual representation, shedding light on the pivotal components of Zepto's data model. By employing this diagram, you'll gain a clearer grasp of the intricate interactions and connections that define the platform's dynamics.



Conclusion

In this case study, we delved into the design of Zepto's schema and Entity-Relationship diagram. By designing a real-world database schema, we have explored how Zepto efficiently manages its core operations such as order processing, customer management, product tracking, delivery logistics, and customer feedback.

Through structured tables and relational models, we were able to understand the flow of data between different departments like customers, delivery partners, orders, and products. This not only helped in simulating real-world business operations but also provided hands-on learning in SQL, data modeling, and database design.

This project helped to understand how businesses alike Zepto use data to solve real-life problems such as fast delivery ,customer satisfaction and managing inventories .

