

# DBMS Project Report

February 8, 2025

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Objective . . . . .	3
1.2	Scope . . . . .	3
1.3	Assumptions . . . . .	4
<b>2</b>	<b>ER Diagram</b>	<b>5</b>
<b>3</b>	<b>Information of Entities</b>	<b>7</b>
3.1	Orders . . . . .	7
3.2	Customers . . . . .	7
3.3	Products . . . . .	7
3.4	Riders . . . . .	8

# 1 Introduction

In today's fast-paced world, quick commerce (q-commerce) apps like Blinkit and Zepto have revolutionized the way consumers access everyday essentials. Unlike traditional e-commerce, which often requires longer delivery times, q-commerce focuses on ultra-fast deliveries, typically within 10 to 30 minutes. These apps utilize a network of strategically located dark stores and advanced logistics algorithms to ensure efficient order fulfillment. The growing demand for instant delivery services has made q-commerce a crucial part of modern urban life, reshaping consumer expectations and business models.

This project aims to analyze the database management system (DBMS) behind q-commerce apps, exploring how they handle vast amounts of real-time data to optimize inventory, deliveries, and user experience. By designing a database schema tailored for a q-commerce platform, the project will illustrate how structured data storage, indexing, and queries contribute to seamless operations. Additionally, it will focus on key components such as order tracking, customer data management, inventory updates, and rider allocation, all of which require efficient database management for smooth execution.

Through this project, we will develop a working model of a q-commerce DBMS, simulating how orders are processed, deliveries are assigned, and stock levels are updated dynamically. The system will demonstrate how relational databases, along with indexing and caching techniques, can enhance the performance and scalability of such applications. By studying the real-world implementation of q-commerce apps, this project will provide valuable insights into the role of database management in ensuring speed, reliability, and efficiency in the instant delivery industry.

## 1.1 Objective

The objective of this project is to design a database schema for a q-commerce platform that can efficiently manage orders, customers, products, and riders. The database should support real-time data processing, order tracking, inventory management, and rider allocation to ensure fast and reliable delivery services. By implementing a robust DBMS tailored for q-commerce, the project aims to showcase the importance of structured data storage, indexing, and query optimization in optimizing operations and enhancing user experience.

## 1.2 Scope

The scope of this project includes the following key features of a q-commerce platform:

- Order Management: Processing, tracking, and updating the status of orders in real-time.

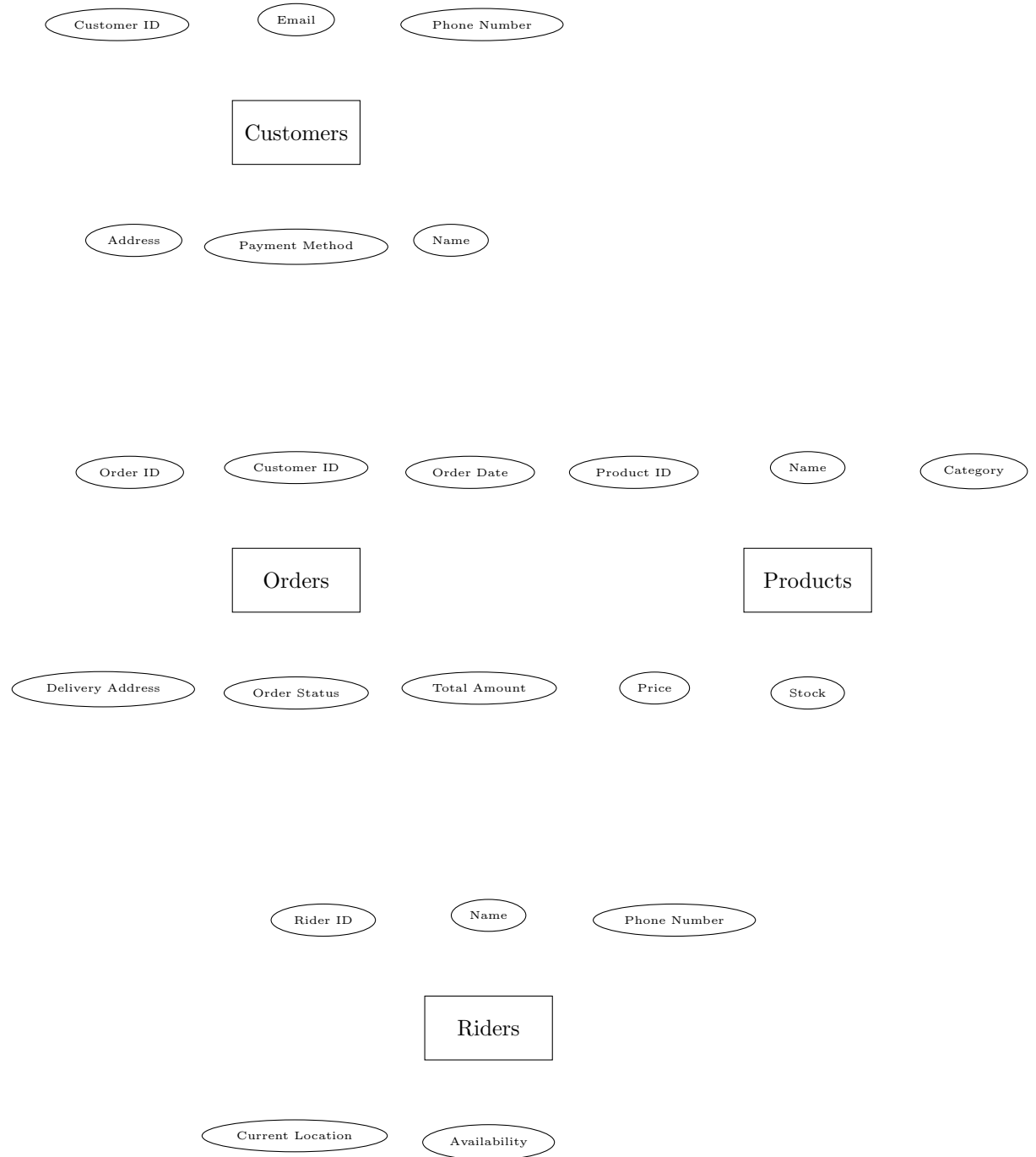
- Customer Data Management: Storing and managing customer information, preferences, and order history.
- Product Inventory: Tracking stock levels, prices, and categories of products available for sale.
- Rider Allocation: Assigning delivery tasks to riders based on availability, location, and order details.

### 1.3 Assumptions

The following assumptions are made for the purpose of this project:

- The q-commerce platform operates in a single city or region with a limited number of dark stores and delivery zones.
- The platform focuses on delivering essential items such as groceries, household goods, and personal care products.
- Customers can place orders through a mobile app or website, with options for express delivery within 30 minutes.
- Riders are independent contractors who use their own vehicles for deliveries and are paid per order completed.

## 2 ER Diagram





## 3 Information of Entities

### 3.1 Orders

- **Order ID:** Unique identifier for each order.
- **Customer ID:** Unique identifier for each customer.
- **Order Date:** Date and time when the order was placed.
- **Delivery Address:** Address where the order is to be delivered.
- **Order Status:** Current status of the order (e.g., pending, confirmed, dispatched, delivered).
- **Total Amount:** Total cost of the order.
- **Payment Status:** Payment status of the order (e.g., pending, paid, refunded).

### 3.2 Customers

- **Customer ID:** Unique identifier for each customer.
- **Name:** Name of the customer.
- **Email:** Email address of the customer.
- **Phone Number:** Contact number of the customer.
- **Address:** Address of the customer.
- **Payment Method:** Preferred payment method of the customer (e.g., credit card, debit card, UPI).

### 3.3 Products

- **Product ID:** Unique identifier for each product.
- **Name:** Name of the product.
- **Category:** Category to which the product belongs (e.g., groceries, household items, personal care).
- **Price:** Price of the product.
- **Stock:** Current stock level of the product.

### 3.4 Riders

- **Rider ID:** Unique identifier for each rider.
- **Name:** Name of the rider.
- **Phone Number:** Contact number of the rider.
- **Current Location:** Current location of the rider.
- **Availability:** Availability status of the rider (e.g., available, busy).