Detailed Project Documentation

1. System Overview

This project showcases an AWS RDS MySQL setup and a Node.js backend API, implementing a normalized schema for a simplified e-commerce scenario with analytics capabilities.

2. Installation & Prerequisites

Operating System: Ubuntu 22.04 LTS or equivalent

Install MySQL Client:

```
sudo apt install mysql-client
```

Backend Setup:

```
cd api
npm install
```

AWS Setup:

- Create a MySQL RDS instance
- Open port 3306 in security group
- Get endpoint and credentials

3. Database Schema

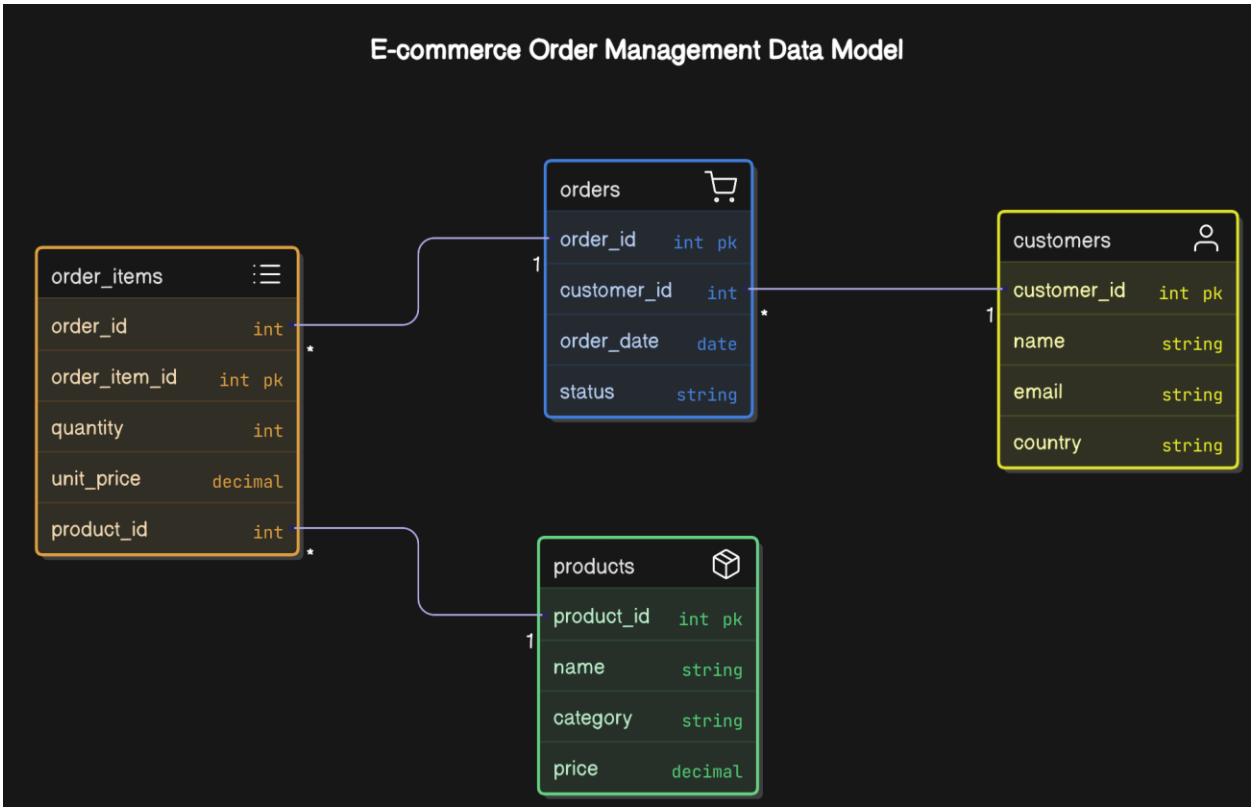
Tables:

- customers
- orders
- order_items
- products

Relationships:

- customers → orders (1:N)
- orders → order_items (1:N)
- products → order_items (1:N)

Visual representation in screenshots/ERD.png



4. SQL Scripts

- setup_db.sql: Schema definitions
- insert_data.sql: Sample dataset
- queries.sql: Key analytical SQL queries

5. Backend API

Endpoint	Method	Description
/top-customers	GET	Rank customers by total spend
/monthly-sales	GET	Monthly sales of shipped/delivered orders

/products-never-ordered	GET	Products with no orders
/avg-order-value	GET	Average order value by country
/frequent-buyers	GET	Customers with more than one order

6. Example Usage

Example API Call:

curl http://localhost:3000/top-customers

7. Style Guide & Contribution

Follow naming consistency (snake_case for SQL, camelCase for JS).

Document endpoints in OpenAPI spec (suggested).

PRs should include schema/logic updates and documentation.