**E-commerce Management System**

**Introduction:**

In the modern digital era, e-commerce has transformed the way businesses operate and how consumers shop. With the growing demand for online shopping platforms, managing data efficiently becomes crucial to ensure smooth operation, personalized customer experience, and effective decision-making. The **E-Commerce Management System** project aims to design and implement a robust database system that supports the fundamental operations of an online shopping platform, including user management, product, order processing, and payment tracking.

This project utilizes the principles of **Database Management Systems (DBMS)** to store, retrieve, and manipulate data in a structured and efficient manner. The system is designed to cater to both administrators and customers. Administrators can add, update, or delete product listings, manage categories, view customer orders, and monitor transactions. Customers can register or log in, browse products, add items to their cart, place orders, and make payments. The database ensures data integrity, security, and consistency across all operations.

The project contains all the concepts of the Database Management System. This project demonstrates the importance of relational databases in e-commerce and highlights key concepts such as normalization, ER modelling, relational schema design, and query optimization.

In conclusion, the E-Commerce Management System DBMS project provides a comprehensive solution for managing the complex data needs of an online retail platform. It serves as a practical example of how database systems are critical in real-world applications, especially in high-demand, user-driven environments like e-commerce.

**Objective:**

The primary objective of this project is to design and implement a **Database Management System (DBMS)** for an E-Commerce Management System that efficiently handles and organizes the core functionalities of an online shopping platform. The system aims to:

* Create a centralized database to store and manage information related to users, products, categories, orders, and payments.
* Ensure **data consistency, security, and integrity** across all operations.
* Enable **easy access and retrieval** of information through structured queries.
* Provide functionalities for both **administrators** (e.g., product management, order tracking) and **customers** (e.g., browsing, ordering, and payments).
* Minimize data redundancy through normalization and implement relational database concepts effectively.
* Demonstrate the use of **Entity-Relationship (ER) Modeling**, schema design, and SQL-based operations in building a real-world DBMS application.
* Offer a user-friendly and scalable solution that can support future enhancements and real-time application.

This project serves as a practical implementation of theoretical DBMS concepts in the context of e-commerce, highlighting the importance of efficient data management in modern digital platforms.

**About the Language:**

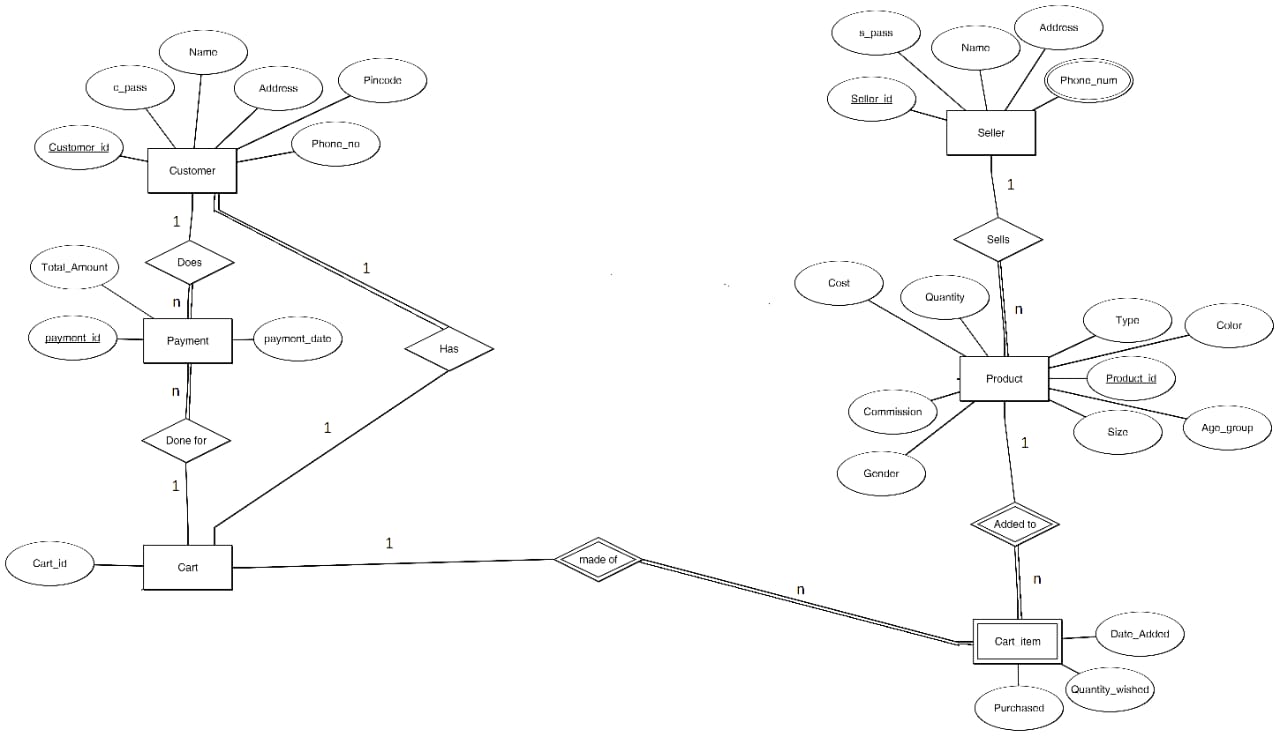
This project is developed using **Structured Query Language (SQL)**, the standard language for interacting with relational databases. SQL is widely used in Database Management Systems (DBMS) for performing various operations such as creating, updating, and retrieving data from databases.

In the context of this E-Commerce Management System, SQL plays a crucial role in:

* **Database Design:** Creating tables for users, products, orders, payments, and other entities, with appropriate data types and constraints.
* **Data Manipulation:** Inserting, updating, and deleting records using SQL commands like INSERT, UPDATE, DELETE, and SELECT.
* **Data Integrity:** Applying constraints such as PRIMARY KEY, FOREIGN KEY, UNIQUE, and NOT NULL to maintain consistency and avoid redundancy.
* **Data Retrieval:** Writing optimized queries to extract meaningful information such as order details, product availability, and customer activity.
* **Relational Operations:** Establishing and managing relationships between different entities in the system using joins and relational algebra concepts.

SQL is ideal for this project as it provides a reliable, efficient, and structured way to manage complex data operations, ensuring the e-commerce platform's backend functions smoothly and accurately.

**E-R Diagram:**



**Code:**

CREATE TABLE Cart(

order\_id VARCHAR(50) PRIMARY KEY

);

CREATE TABLE Customer(

customer\_id NUMBER PRIMARY KEY,

customer\_name VARCHAR(50) NOT NULL,

address VARCHAR(150) NOT NULL,

phone\_no NUMBER NOT NULL UNIQUE,

pincode NUMBER NOT NULL,

order\_id VARCHAR(50) NOT NULL,

FOREIGN KEY(order\_id) REFERENCES Cart(order\_id)

);

CREATE TABLE Payment(

payment\_id NUMBER PRIMARY KEY,

payment\_date DATE NOT NULL,

payment\_mode VARCHAR(50) NOT NULL,

total\_amount FLOAT NOT NULL,

customer\_id NUMBER NOT NULL,

order\_id VARCHAR(50) NOT NULL,

FOREIGN KEY(customer\_id) REFERENCES Customer(customer\_id),

FOREIGN KEY(order\_id) REFERENCES Cart(order\_id)

);

CREATE TABLE Seller(

seller\_id VARCHAR(30) PRIMARY KEY,

s\_pass varchar(10) NOT NULL,

name VARCHAR(30) NOT NULL,

address VARCHAR(150) NOT NULL

);

CREATE TABLE Seller\_phone\_details(

phone\_no NUMBER NOT NULL UNIQUE,

seller\_id VARCHAR(30) NOT NULL,

FOREIGN KEY(seller\_id) REFERENCES Seller(seller\_id)

);

CREATE TABLE Product(

product\_id varchar(30) PRIMARY KEY,

product\_name VARCHAR(50) NOT NULL,

color VARCHAR(15) DEFAULT 'N.A.',

age\_group VARCHAR(6) NOT NULL,

product\_size VARCHAR(5) DEFAULT 'N.A.',

quantity INT NOT NULL,

gender VARCHAR(15) NOT NULL,

commission FLOAT NOT NULL,

price FLOAT NOT NULL,

seller\_id VARCHAR(30) NOT NULL,

FOREIGN KEY(seller\_id) REFERENCES Seller(seller\_id)

);

CREATE TABLE order\_details(

order\_id VARCHAR(50) NOT NULL ,

product\_id VARCHAR(30) NOT NULL,

FOREIGN KEY(order\_id) REFERENCES Cart(order\_id),

FOREIGN KEY(product\_id) REFERENCES Product(product\_id)

);

INSERT ALL

INTO Cart(order\_id) VALUES (1000)

INTO Cart(order\_id) VALUES (1059)

INTO Cart(order\_id) VALUES (1094)

INTO Cart(order\_id) VALUES (1340)

SELECT \* FROM dual;

INSERT ALL

INTO Customer VALUES(1,'Harman Kour','Anupgarh-Sriganganagar-Rajasthan',6345953474,335701,1094)

INTO Customer VALUES(2,'Lovepreet','Vijaynagar-Rajasthan',8054390022,335701,1000)

INTO Customer VALUES(3,'Jashan','Gharuan-Punjab',7480301399,140413,1340)

INTO Customer VALUES(4,'Supriya','Kharar-Punjab',7413993054,140301,1059)

SELECT \* FROM dual;

INSERT ALL

INTO Seller VALUES('sl-594',94,'Pardeep Singh','Mohali-Punjab')

INTO Seller VALUES('sl-104',10,'Ankit','Ghurgaon-Haryana')

SELECT \* FROM dual;

INSERT ALL

INTO Seller\_phone\_details VALUES(7892345684,'sl-594')

INTO Seller\_phone\_details VALUES(9834526178,'sl-104')

SELECT \* FROM dual;

INSERT ALL

INTO Payment VALUES(67,'04-04-2025','COD',986.59,1,1094)

INTO Payment VALUES(435,'03-28-2025','UPI',345,3,1340)

SELECT \* FROM dual;

INSERT ALL

INTO Product VALUES('59649','Shampoo','white','12+','300ml',2,'Male and Female',20.0,354.56,'sl-594')

INTO Product VALUES('94584','Shirt','brown','10-12','M',1,'Female',3.0,450.00,'sl-104')

INTO Product VALUES('34748','Accesories','golden','any','Free',2,'Female',0.0,545.90,'sl-104')

INTO Product VALUES('75326','Bags','black','13+','S',4,'Male',32.0,690.96,'sl-594')

SELECT \* FROM dual;

INSERT ALL

INTO order\_details VALUES(1000,'75326')

INTO order\_details VALUES(1340,'94584')

INTO order\_details VALUES(1094,'59649')

INTO order\_details VALUES(1059,'34748')

SELECT \* FROM dual;

SELECT \* FROM Product;

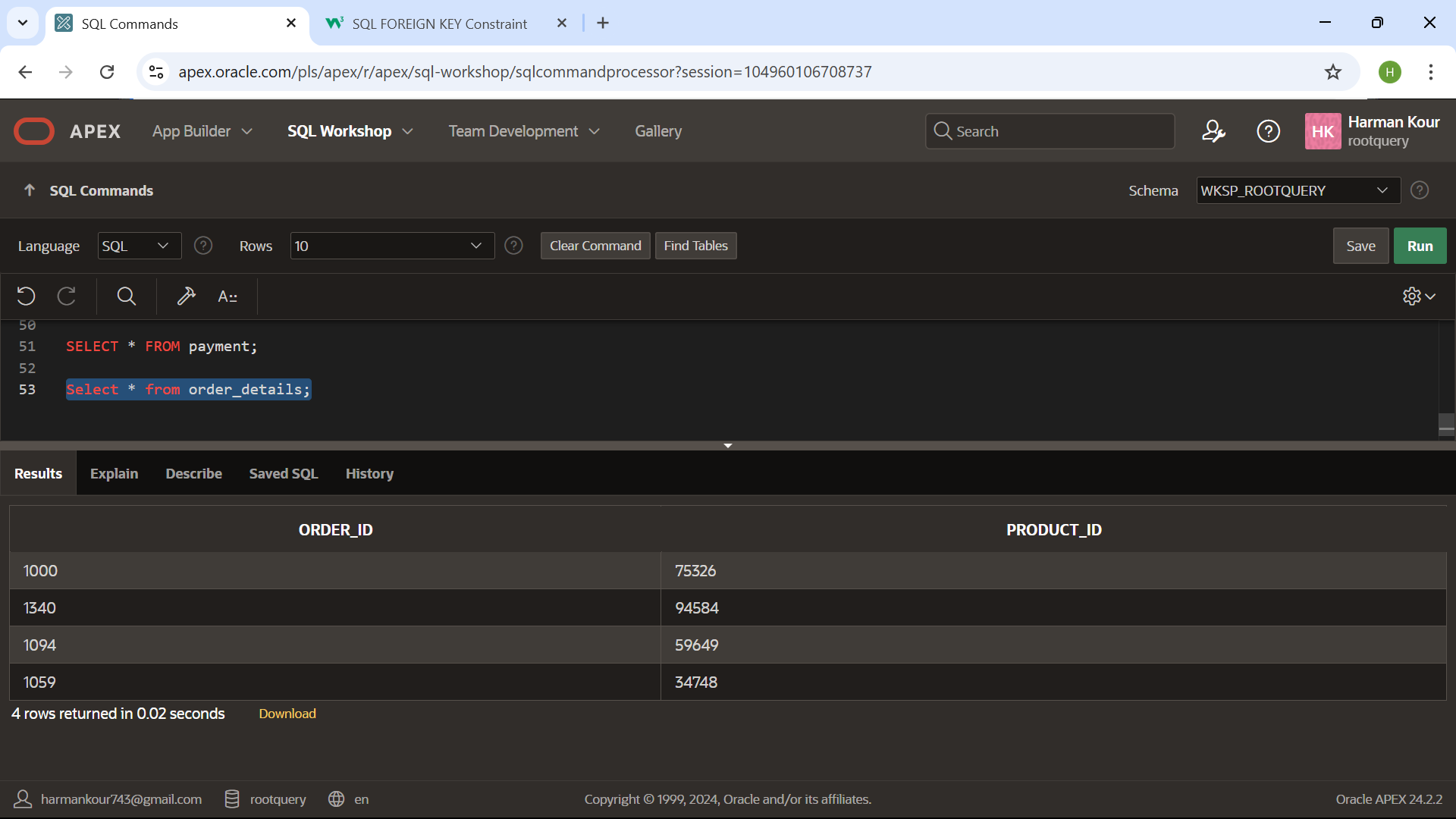
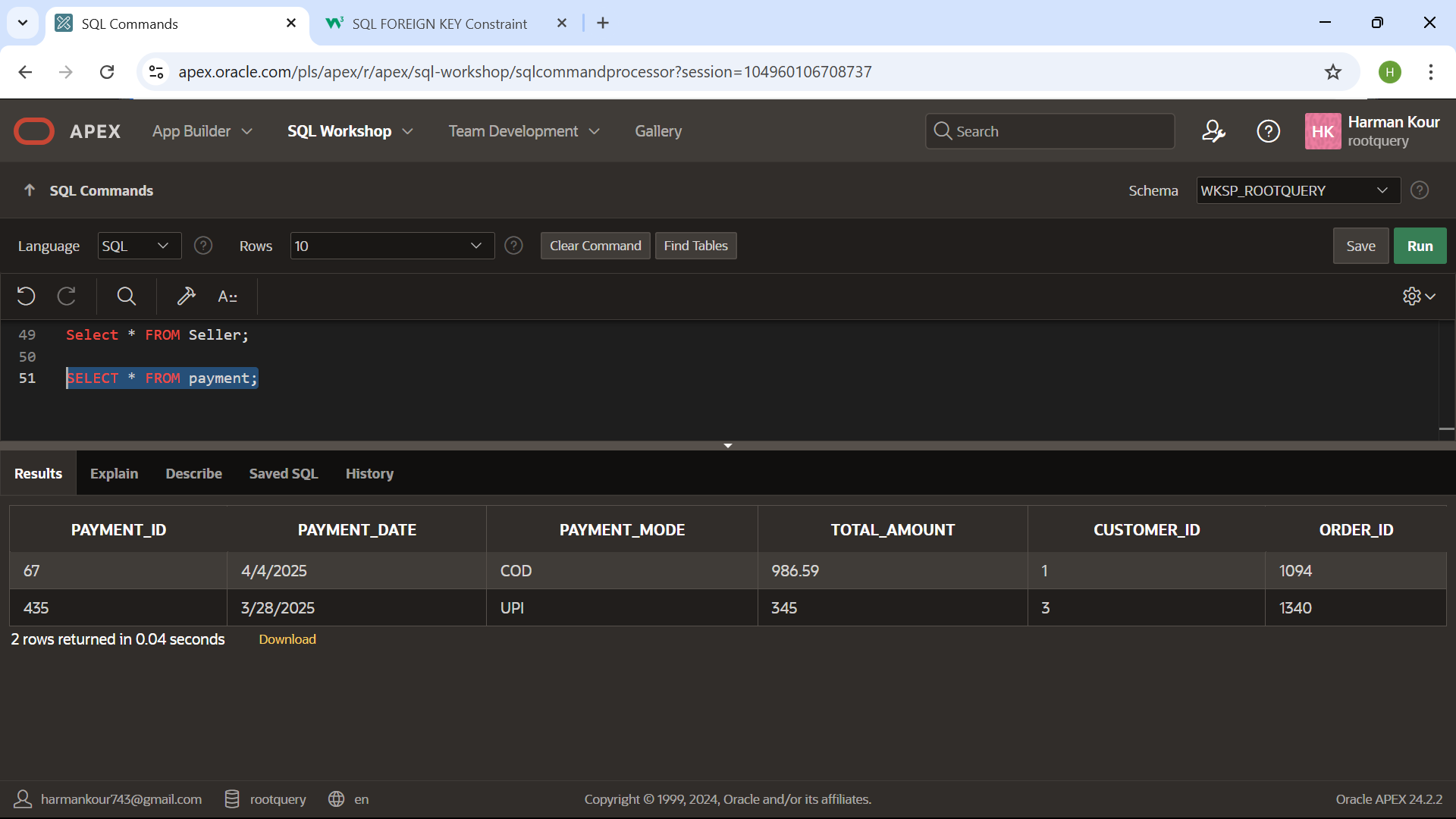
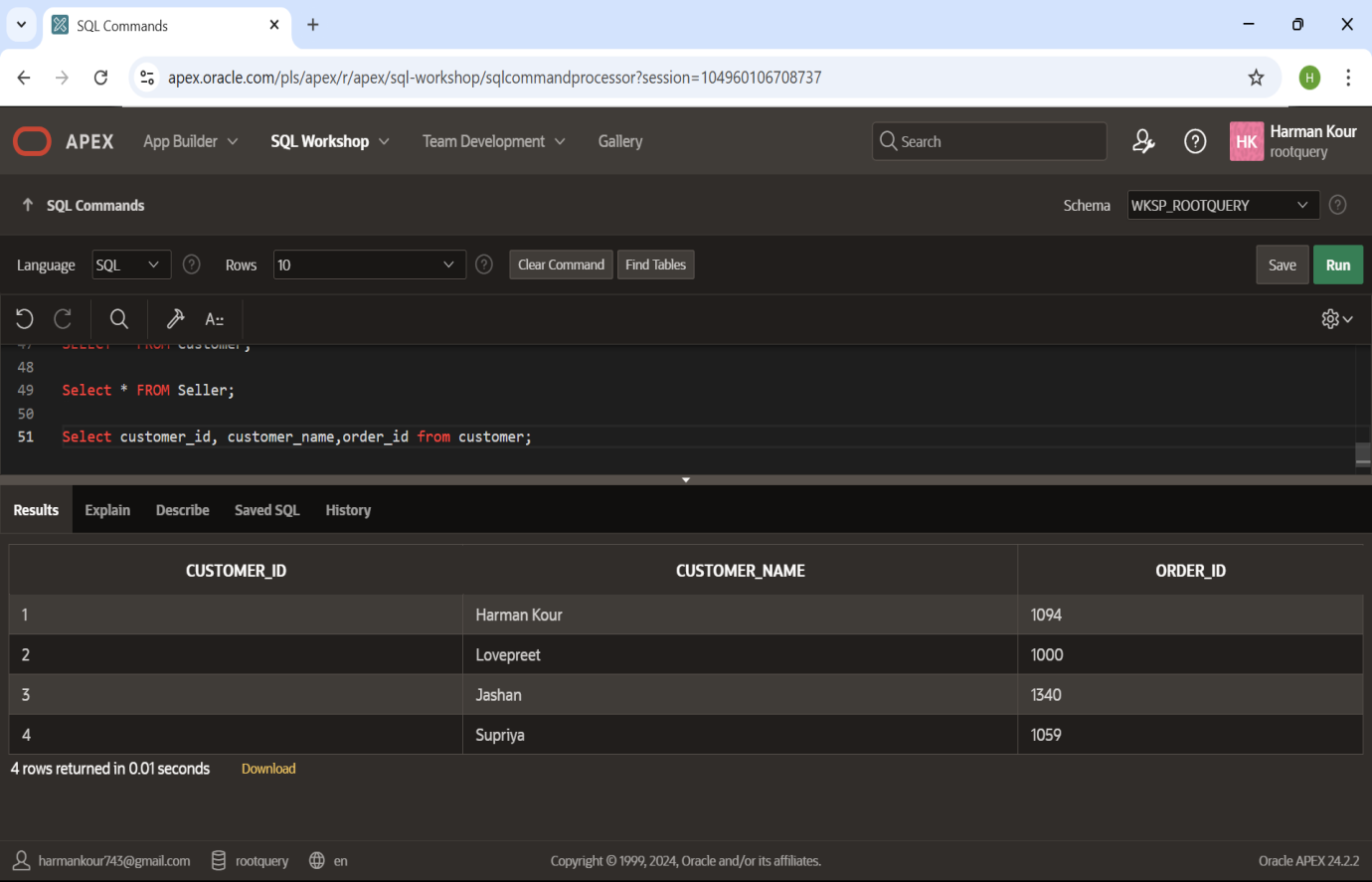
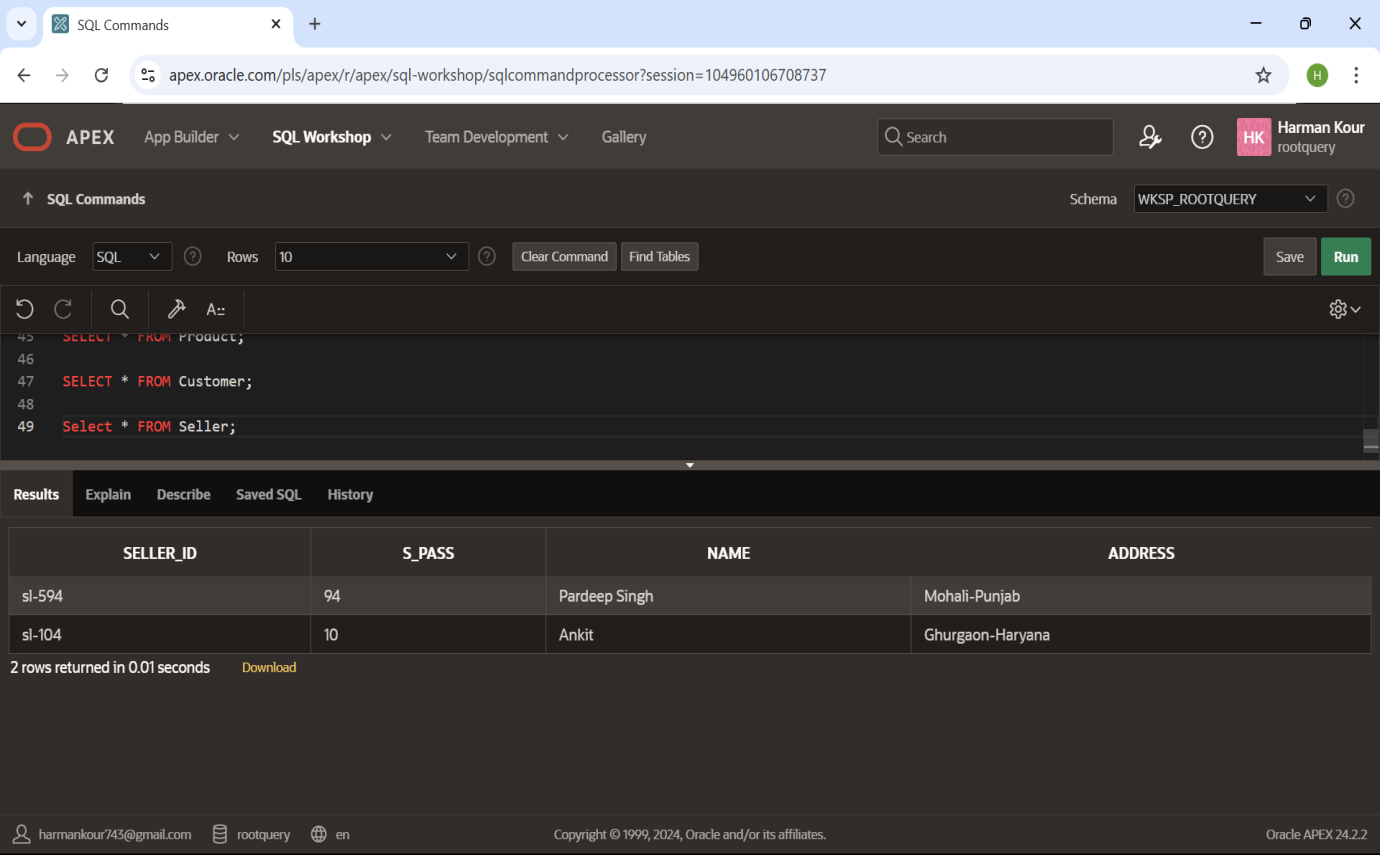
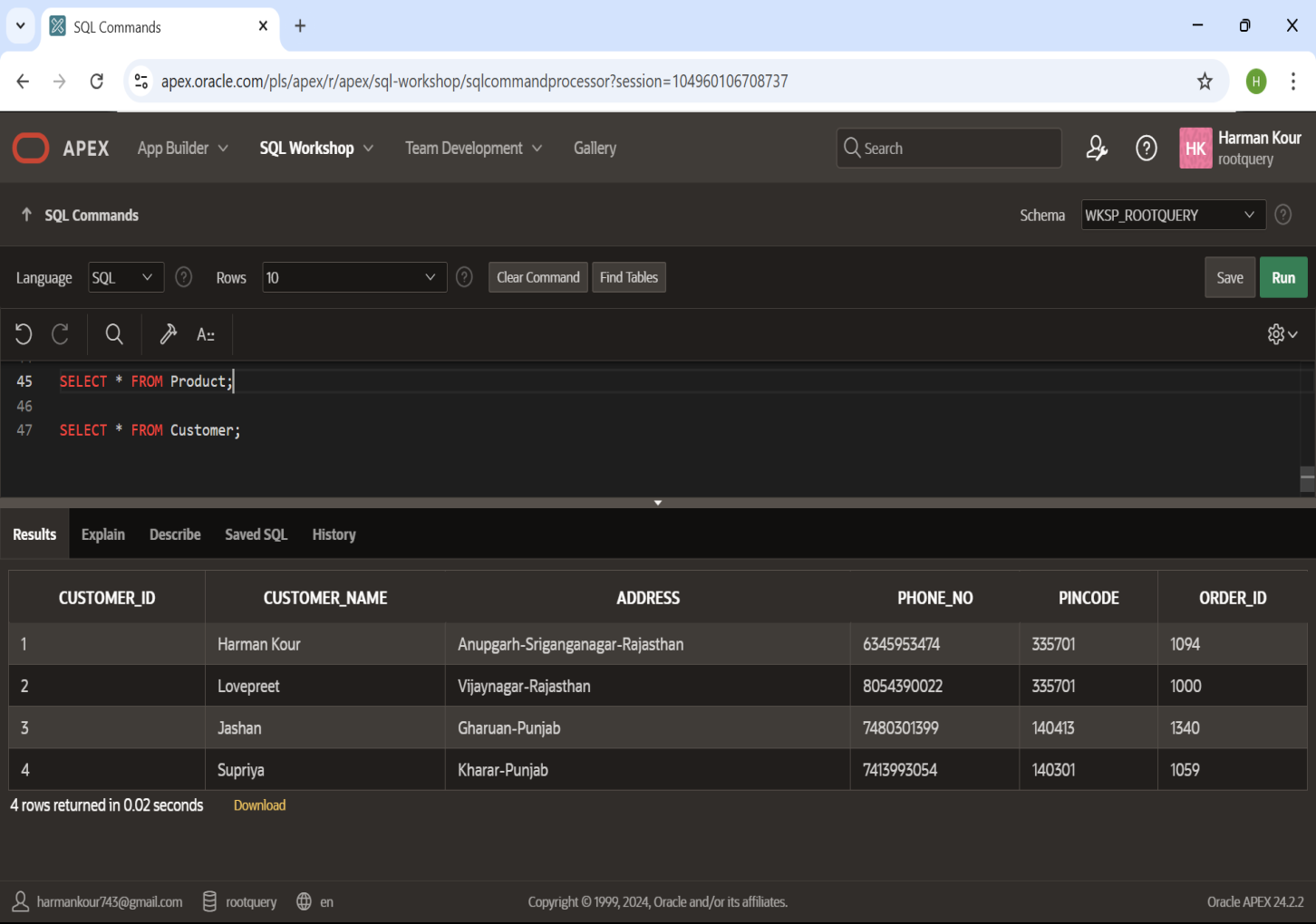
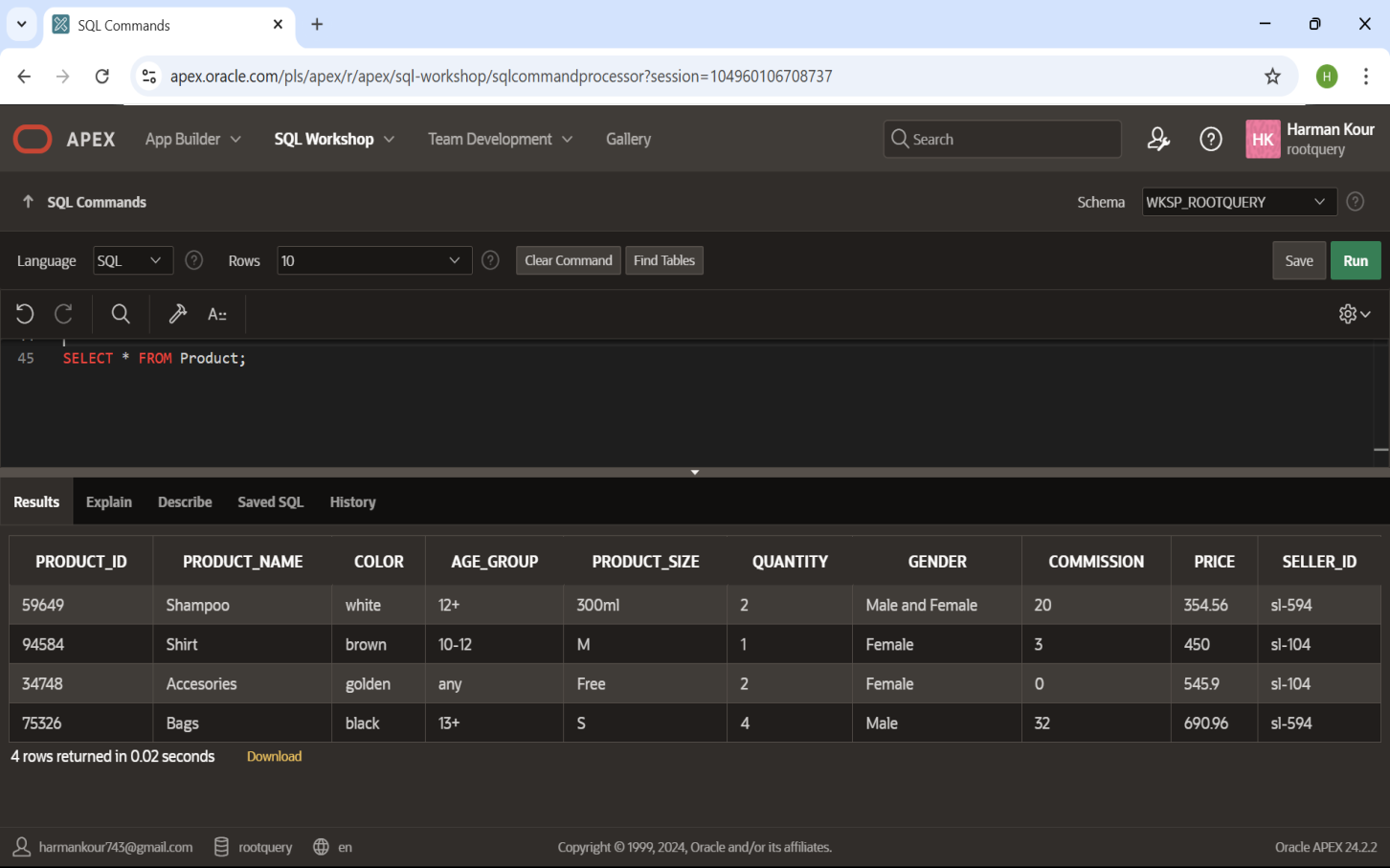
SELECT \* FROM Customer;

SELECT \* FROM Seller;

SELECT \* FROM payment;

SELECT \* FROM order\_details;

**Output:**

****

**Conclusion:**

The **E-Commerce Management System** project serves as a comprehensive example of how Database Management Systems (DBMS) are essential in managing, organizing, and retrieving large volumes of data in a real-world application. With the rising importance of online shopping platforms, an efficient and reliable backend database is critical to ensure seamless user experience and effective business operations.

In this project, we designed and implemented a relational database using **Structured Query Language (SQL)** to manage various aspects of an e-commerce platform, including user accounts, product listings, orders, payments, and inventory. Each module was carefully designed with proper normalization to reduce data redundancy and maintain consistency across the database. Key constraints such as primary keys, foreign keys, and unique identifiers were used to ensure **referential integrity** and logical relationships between tables.

We also created complex SQL queries for retrieving data, handling transactions, and generating meaningful reports, such as sales summaries and order histories. These queries demonstrated our understanding of **joins, subqueries, aggregate functions**, and **set operations**, which are critical in handling multi-table operations in real-time applications.

Through this project, we have deepened our understanding of important DBMS concepts such as:

* **Entity-Relationship (ER) modelling** for representing real-world entities and their relationships.
* **Relational schema design** for implementing the ER model in a structured form.
* **Normalization** for organizing the data efficiently.
* **Transaction management** to ensure data reliability and accuracy.
* **Security principles** by controlling access to sensitive data using SQL privileges.

Overall, this project illustrates the foundational role that DBMS plays in building scalable and maintainable systems for modern digital services like e-commerce. It not only fulfilled its intended purpose of simulating the backend of an online store but also provided a hands-on opportunity to apply theoretical knowledge in a practical, meaningful way. This experience will be valuable for future projects involving data-driven systems, business applications, or web-based platforms.