



CC5051NI Databases
50% Individual Coursework

Autumn 2023

Student Name: Rounak Joshi

London Met ID: 22067875

Assignment Submission Date: 14 January 2024.

Word Count: 4026

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table of Contents

1. Introduction	1
1.1. Aims and Objectives	2
1.2. Current Business Activities and Operations	3
1.3. Business Rules	4
1.4. Entities and Attributes	5
2. Initial ERD.....	12
3. Normalization.....	14
3.1. Unnormalized Form (UNF)	14
3.2. First Normal Form (1NF)	14
3.3. Second Normal Form (2NF)	15
3.4. Third Normal Form (3NF)	16
4. Final Entity Relation Diagram(ERD)	18
5. Implementation	19
5.1. Creating User	19
5.2. Granting and Connecting to User.....	19
5.3. Creating Tables and Describing the Tables	19
5.4. Data Insertion.....	22
5.5. Displaying Data	27
5.6. Creating Dump File	31
5.7. Dropping Tables.....	32
6. Database Querying.....	33
6.1. Information Query	33
6.2. Transaction Query.....	35
7. Critical Evaluation	37
7.1. Critical Evaluation of the Module.....	37
7.2. Critical Assessment of Coursework.....	38
8. References	39

Table of Figures

Figure 1 Initial entity Relation Diagram.....	13
Figure 2 Final Entity Relation Diagram.....	18
Figure 3 Creating new user.	19
Figure 4 Granting and connecting to the user.	19
Figure 5 Creating and describing the table of vendor.....	19
Figure 6 Creating and describing the table of Product_Category	20
Figure 7 Creating and describing the table of Customer_Category.....	20
Figure 8 Creating and describing the table of Payment_type.....	20
Figure 9 Creating and describing the table of Product.	20
Figure 10 Creating and describing the table of Customer.	21
Figure 11 Creating and describing the table of Orders.....	21
Figure 12 Creating the table of Order_Product.	21
Figure 13 Describing the table of Order_Product.	21
Figure 14 Inserting Values in the Vendor table.	22
Figure 15 Inserting Values in the Product_Category table.	22
Figure 16 Inserting Values in the Customer_Category table.	23
Figure 17 Inserting Values in the Payment_type table.	23
Figure 18 Inserting Values in the Product table.....	23
Figure 19 Inserting Values in the Customer table.	24
Figure 20 Inserting Values in the Orders table.	24
Figure 21 Inserting Values in the Order_Product table.....	25
Figure 22 Inserting Values in the Order_Product table 2.....	26
Figure 23 Displaying the Data of the table Vendor.....	27
Figure 24 Displaying the Data of the table Product_Category.....	27
Figure 25 Displaying the Data of the table Customer_Category.	27
Figure 26 Displaying the Data of the table Payment_type.....	28
Figure 27 Displaying the Data of the table Product.	28
Figure 28 Displaying the Data of the table Customer.....	28
Figure 29 Displaying the Data of the table Orders.	29
Figure 30 Displaying the Data of the table Order_Product.....	30

Figure 31 Creating Dump File.	31
Figure 32 Dropping tables.	32
Figure 33 Information Query 1.	33
Figure 34 Information Query 2.	33
Figure 35 Information Query 3.	33
Figure 36 Information Query 4.	34
Figure 37 Information Query 5.	34
Figure 38 Transaction Query 1.....	35
Figure 39 Transaction Query 2.....	35
Figure 40 Transaction Query 3.....	36
Figure 41 Transaction Query 4.....	36
Figure 42 Transaction Query 5.....	36

Table of Tables

Table 1 The attributes of the Vendor entity.	5
Table 2 The Attributes of the Product_Category entity.....	5
Table 3 The attributes of the Product entity.....	6
Table 4 The attributes of the Customer_Category entity.....	7
Table 5 The attributes of the Customer entity.	8
Table 6 The attributes of the Payment_Type entity.....	9
Table 7 The attributes of the Orders entity.....	10
Table 8 The attributes of the Order_Prodct entity.	11
Table 9 List of all Entities and Attributes for Initial ERD.	12

1. Introduction

Gadget Emporium is an online marketplace launched by Entrepreneur and electronic enthusiast Mr. John. The online marketplace provides a platform for the private consumer and the business organizations to view purchase among the scope of various electronic goods. Online established business hopes to takeover the market of sales of electronic goods and provide its services throughout the country. The online market provides its customers with a wide scope of electronic goods at a price that benefits both the customer as well as the business. The business expects to provide the goods by purchasing the products through various vendors at a wholesale price.

The business requires a database that can record all the details of the business. The database requires to collect and store product details of the products that are being sold, Vendors who supply these products to the business, customers that have been purchasing the goods from the online market and to divide the customers into categories of regular, staff, and VIP. These categories also help the business to provide the discounts to the customers as per their category. The database also tracks the order details of all the orders. The database is source used by the business to record all the details which can be used to analyze the profits and losses faced by the business and help create future plans that benefit the business.

1.1. Aims and Objectives

The aims of Gadget Emporium are as follows:

- ➔ To make the electronic products accessible to both private consumers and business organizations.
- ➔ To create a successful online store that can attract the customers to the business.
- ➔ To make benefit for both, the business, and the vendor.
- ➔ To keep record of all customers, products, order, and vendor connected to the business.
- ➔ To operate the business systematically and efficiently.

The objectives of the business are stated below-

- ➔ Create an online marketplace for consumer to purchase electronic goods.
- ➔ Make contract with vendors for the supply of products to be sold through the online market.
- ➔ Create a database of the business to record of the customers, orders, products and vendors in the business.
- ➔ Provide a platform for private consumers as well as the business organization to purchase electronic products.

1.2. Current Business Activities and Operations

The business activities and operations done by the business are stated below:

- Separates the customers into different categories, through which the customers are provided discounts on their purchases.
- Customers can provide their address for their orders to be delivered to their address successfully.
- Products are categorized by which the customers can find the products based on the category they are finding to get various options.
- Customers can order as much as they want or need to.
- Provides the availability state of a product for the customers to see if the product they are looking for is available or not.
- The business provides customers various payment methods for them to select as per their comfortability to pay for their orders.
- Stores all the data of the orders and customers which helps the customer to track any of their orders also a single time registration is only required.
- Provides invoices of the orders paid by the customer for which can be used to see the details of the order and to return a product from the order if the product is incorrect or damaged.

1.3. Business Rules

- The customers can get discount on products from 0%. 5% and 10% as per their category.
- Customers must register their address on their details for the delivery of their order.
- The products can only be of one category, but one product category can have many products.
- Customers can have more than one order, but each order is of only one customer.
- The order can also have multiple products in it and any one type of product can be in different orders of different customers.
- A product can have only one vendor, but one vendor can supply many products.
- The product must show availability and stock quantity to prevent over selling of the product.
- There should be various payment methods for the consumers to pay for their products.
- The order can have only one payment type, but one payment type can be used for different orders.

1.4. Entities and Attributes

a. Vendor

Attribute	Description	Type	Keys
Vendor_ID	A unique ID provided to identify the vendor	VARCHAR2(6)	Primary Key
Vendor_name	The name of the vendor	VARCHAR(50)	

Table 1 The attributes of the Vendor entity.

b. Product Category

Attribute	Description	Type	Keys
Product_Category_ID	A unique ID provided to identify the category of the product	VARCHAR2(6)	Primary Key
Product_Category_name	The name of the product Category	VARCHAR(50)	

Table 2 The Attributes of the Product_Category entity.

c. Product

Attribute	Description	Type	Keys
Product_ID	A unique ID provided to identify the product	VARCHAR2(6)	Primary Key
Product_name	The name of the product	VARCHAR(50)	
Description	The description of the product	VARCHAR(50)	
Rate	The price of the product	NUMBER(10)	
Stock_Quantity	The stock quantity of the product	NUMBER(10)	
Availability	The product availability in the store	VARCHAR(50)	
Vendor_ID	A unique ID provided to identify the vendor. This attribute is used to link the product with its vendor	VARCHAR2(6)	Foreign Key
Product_Category_ID	A unique ID provided to identify the product category. This attribute is used to link the Product entity with the Product_category	VARCHAR2(6)	Foreign Key

Table 3 The attributes of the Product entity.

d. Customer Category

Attribute	Description	Type	Keys
Customer_Category_ID	A unique ID provided to identify the category of the customer.	VARCHAR2(6)	Primary Key
Customer_Category_name	Name of the customer category	VARCHAR(50)	
Discount Rate	The rate of discount to be given to the customer as per category	VARCHAR2(5)	

Table 4 The attributes of the Customer_Category entity.

e. Customer

Attribute	Description	Type	Keys
Customer_ID	A unique ID provided to identify the customer.	VARCHAR2(6)	Primary Key
Customer_name	The name of the customer	VARCHAR(50)	
Customer_email	The email address of the customer	VARCHAR(50)	
Customer_address	The residence address of the customer.	VARCHAR(50)	
Customer_Category_ID	A unique ID provided to identify the category of the customer. This attribute is used to link the Customer_category entity with the Customer entity	VARCHAR2(6)	Foreign Key

Table 5 The attributes of the Customer entity.

f. Payment Type

Attribute	Description	Type	Keys
Payment_Type_ID	A unique ID provided to identify the type of the payment	VARCHAR2(6)	Primary Key
Payment_Type_name	The name of the method of the payment	VARCHAR(50)	

Table 6 The attributes of the Payment_Type entity.

g. Orders

Attribute	Description	Type	Keys
Order_ID	A unique ID provided to identify the order placed.	VARCHAR2(6)	Primary Key
Order_date	The date the order was placed.	DATE	
Discount	The discount received on the order placed	VARCHAR(50)	
Total_amount	The total amount of the order to be paid	NUMBER(10)	
Customer_ID	A unique ID provided to identify the customer. This attribute is used to link the Customer entity with the Orders entity	VARCHAR2(6)	Foreign Key
Payment_Type_ID	A unique ID provided to identify the type of payment. This attribute is used to link the Payment_Type entity with the Orders entity	VARCHAR2(6)	Foreign Key

Table 7 The attributes of the Orders entity.

h. Order_Product

Attribute	Description	Type	Keys
Order_ID	An Id that is given to each order. The Order_ID links the Order_Product entity with the Orders entity	VARCHAR2(6)	Primary Key, Foreign Key
Product_ID	An Id given to each product. The Product_ID links the Order_Product entity with the Product entity	VARCHAR2(6)	Primary Key, Foreign Key
Quantity	The quantity of the products in the order	NUMBER(10)	
Price	The prices of the products in the order.	NUMBER(10)	
Line_Total	The total amount gained by multiplying the unit price of the product with the quantity of the products in the order		

Table 8 The attributes of the Order_Product entity.

2. Initial ERD

An initial ERD can be stated as a simplified version and representation of the database which is created based on the provided information. An initial ERD is usually created during the beginning phase and is counted as part of the designing phase. It is created to provide a starting point for the project and to develop and start through it.

The following table shows the total list of entities and attributes represented in the initial ERD.

ENTITIES	ATTRIBUTES	KEYS
ORDERS	Order_ID	Primary Key
	Date	
	Customer_ID	Foreign Key
	Discount	
	Total_Amount	
	Payment_type_ID	
	Payment_type_name	
CUSTOMER	Customer_ID	Primary Key
	Customer_name	
	Customer_email	
	Customer_Category_ID	
	Customer_Category_name	
	Customer_address	
	Discount_Rate	
PRODUCTS	Product_ID	Primary Key
	Product_name	
	Description	
	Product_Category_ID	
	Product_Category_name	
	Vendor_ID	
	Vendor_name	
	Rate	
	Quantity	
	Price	
	Stock_Quantity	
	Availability	

Table 9 List of all Entities and Attributes for Initial ERD.

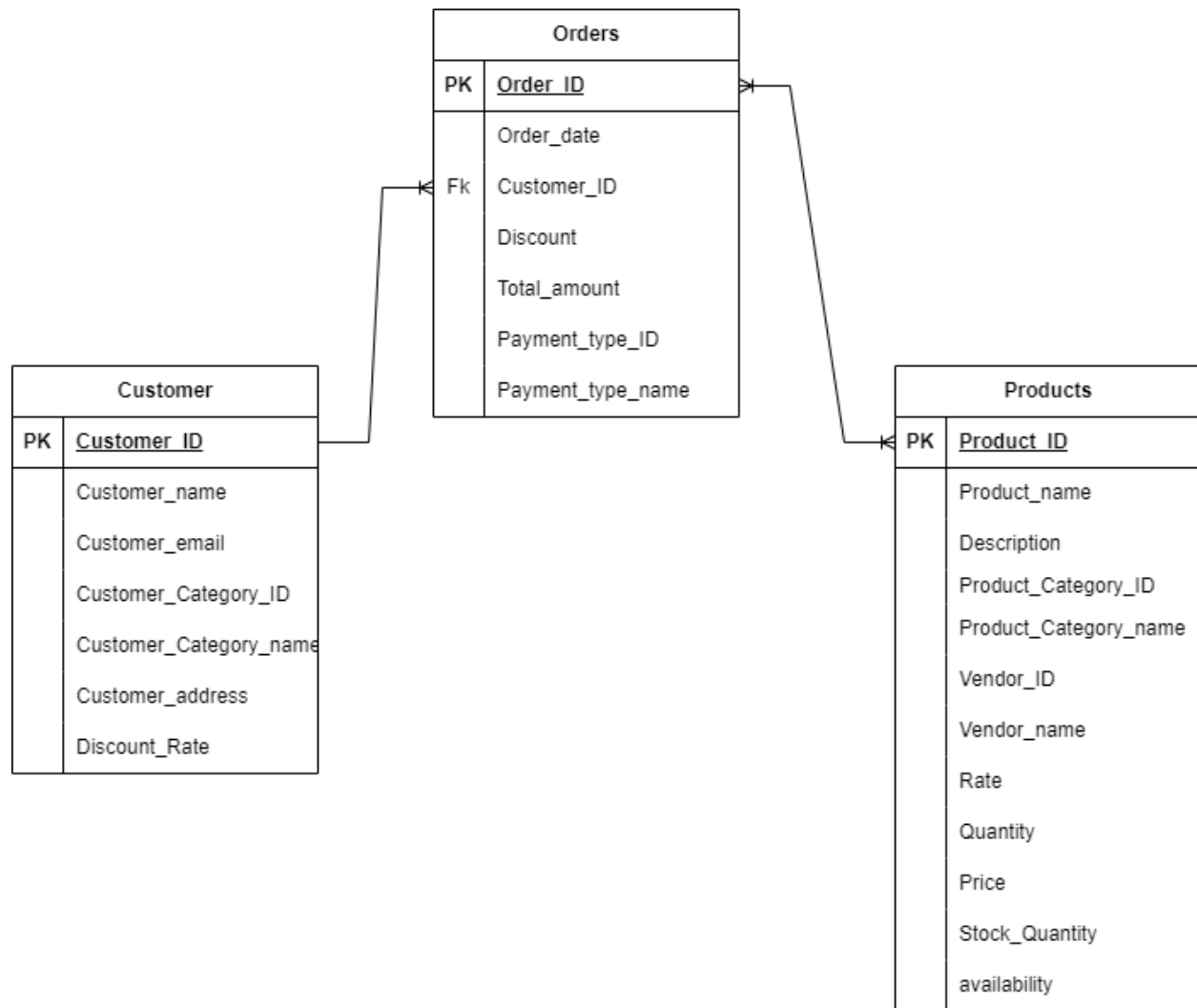


Figure 1 Initial entity Relation Diagram.

The initial ERD above gives the data flow of the business but it still consists of bottlenecks. In the above ERD the entities show many to many relationships with each other which is required to be avoided as the relationship can cause duplication of the data. To avoid these errors the normalization is done.

3. Normalization

Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of Normalization in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically. (Peterson, 2023)

3.1. Unnormalized Form (UNF)

An unnormalized relation is a relation that contains repeating values. An unnormalized relation can also contain relations nested within other relations, as well as all kinds of transitive dependencies. Sometimes unnormalized relations are signified by ONF, but an unnormalized relation is not to be confused with a denormalized relation. The unnormalized relation is any relation in its raw state, and they commonly contain repeating values, and other characteristics that are not found in denormalized relations. (Consulting, 2024)

Orders: (Order_ID, Order_date, Customer_ID, Customer_name, Customer_email, Customer_category_ID, Customer_category_name, Customer_address, Discount_rate {Product_ID, Product_name, Description, Product_category_ID, Product_category_name, Vendor_ID, Vendor_name, Rate, Quantity, Stock_Quantity, availability, Price}, Discount, Total_amount, Payment_type_ID, Payment_type_name)

3.2. First Normal Form (1NF)

If a relation contains a composite or multi-valued attribute, it violates the first normal form, or the relation is in the first normal form if it does not contain any composite or multi-valued attribute. A relation is in first normal form if every attribute in that relation is single-valued attribute. (GeeksForGeeks, 2024)

In this UNF, the repeating group can be seen as the product details which is removed from the primary table. Now two tables are created which are Orders-1 and Product-2 with the primary keys Order_ID and Product_ID. Here, the primary key Order_ID from Order-1 is a foreign key in Product-1.

Representing the data in First Normal form:

Orders-1: (Order_ID, Order_date, Customer_ID, Customer_name, Customer_email, Customer_category_ID, Customer_category_name, Customer_address, Discount, Total_amount, Payment_type_ID, Payment_type_name)

Product-1: (Product_ID, Product_name, Description, Product_category_ID, Product_category_name, Vendor_ID, Vendor_name, Rate, Quantity, Stock_quantity, availability, Price, Order_ID*)

3.3. Second Normal Form (2NF)

A relation is in second normal form (2NF) if and only if it is in 1NF and no non-prime attributes are functionally dependent on a subset of the candidate key(s). In other words, any column that's not a key column is dependent on the whole information in the candidate key. (Kozubek, 2020)

Checking for partial dependencies in the Orders-1 and Product-1:**Orders-1:**

Partial dependency only occurs if there is composite primary key. As there is only one primary key in the Orders-1, the entity cannot have partial dependency. So, the Orders-1 is already in Second Normal Form.

Product-1:

Product_ID → Product_name, Description, Product_category_ID, Product_category_name, Vendor_ID, Vendor_name, Rate, Quantity, Price, Stock_quantity, availability

Order_ID → X

Product_ID, Order_ID → quantity, price, line total

Representing data in Second Normal Form:

Orders-2: (Order_ID, Order_date, Customer_ID, Customer_name, Customer_email, Customer_category_ID, Customer_category_name, Customer_address, Discount, Total_amount, Payment_type_ID, Payment_ty

e_name)

Product-2: (Product_ID, Product_name, Description, Product_category_ID, Product_category_name, Vendor_ID, Vendor_name, Rate, Stock_quantity, availability)

Order_Product-2: (Order_ID*, Product_ID*, Quantity, Price, Line_Total)

3.4. Third Normal Form (3NF)

A relation is in third normal form if there is no transitive dependency for non-prime attributes. Also, in a 3NF-compliant table, no non-primary key attribute has transitively dependent relationships to the primary key. (Becker, 2020)

Checking for transitive dependencies in Orders-2, Product-2 and Order_Product-2:

Orders-2:

Order_ID-> Payment_type_id, Payment_type_name

Order_ID-> Customer_ID, Customer_name, Customer_email, Customer_address

Order_ID-> Customer_category_ID, Customer_category_name

Product-2:

Product_ID-> Vendor_ID, Vendor_name

Product_ID-> Product_category_ID, Product_category_name

Order_Product-2:

The data in the Order_Product-2 table are only dependent on the composite primary key. As a non-key does not determine the value of another non-key. So, there is no transitive dependency in this table.

Representing the data in the third normal form:

Orders-3: (Order_ID, Order_date, Discount, Total_amount, Customer_ID*, Payment_type_ID*)

Payment_Type-3: (Payment_type_ID, Payment_type_name)

Customer-3: (Customer_ID, Customer_name, Customer_email, Customer_address, Customer_category_ID*)

Customer_Category-3: (Customer_category_ID, Customer_category_name, Discount_rate)

Product-3: (Product_ID, Product_name, Description, Rate, Stock_quantity, availability, Vendor_ID*, Product_category_ID*)

Vendor-3: (Vendor_ID, Vendor_name)

Product_Category-3: (Product_category_ID, Product_category_name)

Order_Product-3: (Order_ID*, Product_ID*, Quantity, Price, Line_Total)

4. Final Entity Relation Diagram(ERD)

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education, and research. (Lucidchart, 2024)

After the normalization process, total 8 entities were created. They are Orders, Product, Customer, Payment_Type, Customer_Category, Product_Category, Vendor, and Order_Product. The Orders entity display the order details. Product entity displays the product details. Customer entity displays the different customer details. Payment_Type entity displays the payment method details. Customer_Category displays the category a customer is listed as in the business. The Product_Category displays the category of product. The Vendor display the details of the vendor.

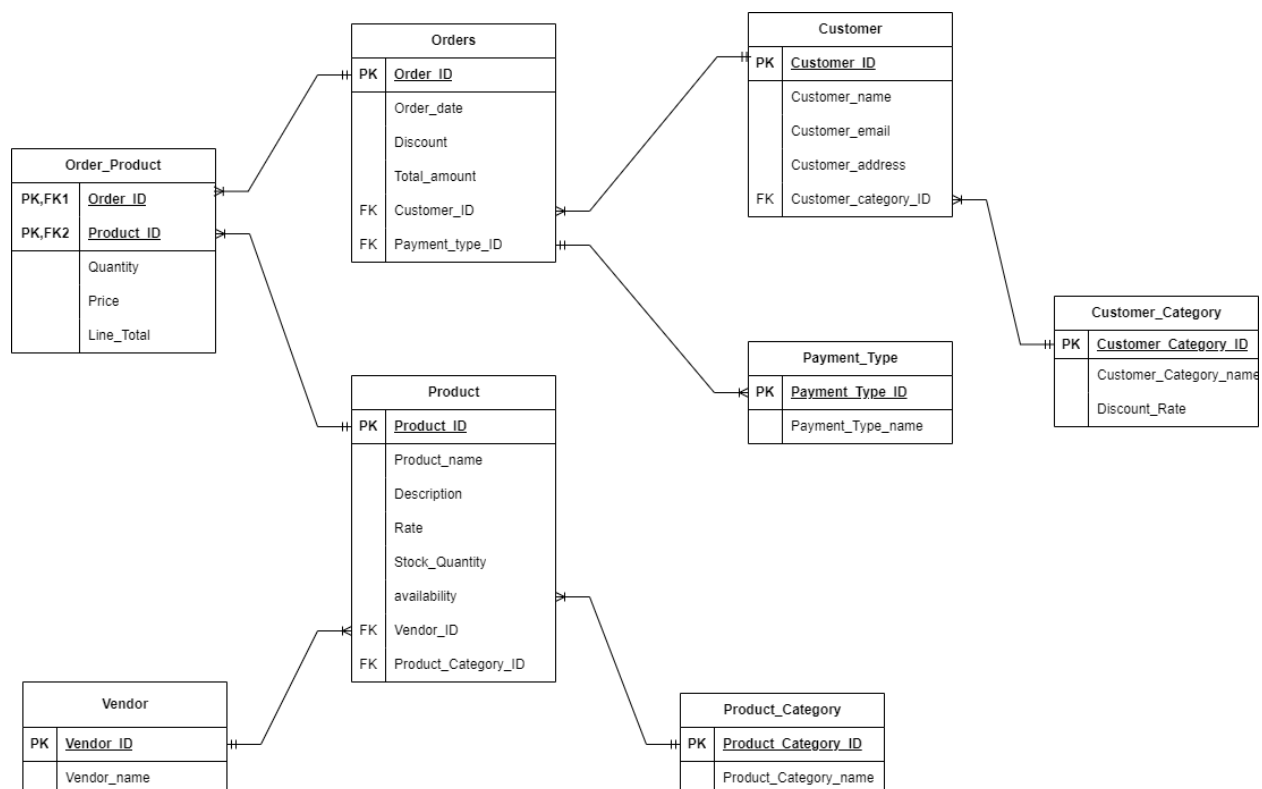


Figure 2 Final Entity Relation Diagram.

5. Implementation

5.1. Creating User

```
SQL> connect
Enter user-name: system
Enter password:
Connected.
SQL> CREATE USER GadgetEmporium IDENTIFIED BY Gadgetemporium;

User created.
```

Figure 3 Creating new user.

5.2. Granting and Connecting to User

```
SQL> GRANT CONNECT, RESOURCE TO GadgetEmporium;

Grant succeeded.

SQL> CONNECT GadgetEmporium;
Enter password:
Connected.
SQL>
```

Figure 4 Granting and connecting to the user.

5.3. Creating Tables and Describing the Tables

a. Vendor

```
SQL> CREATE TABLE Vendor (Vendor_ID VARCHAR2(6), Vendor_name VARCHAR2(50), CONSTRAINT vendor_pk PRIMARY KEY(Vendor_ID));

Table created.

SQL> DESC Vendor;
```

Name	Null?	Type
VENDOR_ID	NOT NULL	VARCHAR2(6)
VENDOR_NAME		VARCHAR2(50)

Figure 5 Creating and describing the table of vendor.

b. Product_Category

```
SQL> CREATE TABLE Product_Category ( Product_Category_ID VARCHAR2(6), Product_Category_name VARCHAR2(50), CONSTRAINT product_category_pk PRIMARY KEY(Product_Category_ID));
```

Table created.

```
SQL> DESC Product_Category;
```

Name	Null?	Type
PRODUCT_CATEGORY_ID	NOT NULL	VARCHAR2(6)
PRODUCT_CATEGORY_NAME		VARCHAR2(50)

Figure 6 Creating and describing the table of Product_Category .

c. Customer_Category

```
SQL> CREATE TABLE Customer_Category (Customer_Category_ID VARCHAR2(6), Customer_Category_name VARCHAR(50), Discount_rate VARCHAR2(5), CONSTRAINT Customer_Category_pk PRIMARY KEY(Customer_Category_ID));
```

Table created.

```
SQL> DESC Customer_Category;
```

Name	Null?	Type
CUSTOMER_CATEGORY_ID	NOT NULL	VARCHAR2(6)
CUSTOMER_CATEGORY_NAME		VARCHAR2(50)
DISCOUNT_RATE		VARCHAR2(5)

Figure 7 Creating and describing the table of Customer_Category.

d. Payment_type

```
SQL> CREATE TABLE Payment_type (Payment_type_ID VARCHAR2(6), Payment_type_name VARCHAR2(50), CONSTRAINT payment_type_pk PRIMARY KEY(Payment_type_ID));
```

Table created.

```
SQL> DESC Payment_type;
```

Name	Null?	Type
PAYMENT_TYPE_ID	NOT NULL	VARCHAR2(6)
PAYMENT_TYPE_NAME		VARCHAR2(50)

SQL>

Figure 8 Creating and describing the table of Payment_type.

e. Product

```
SQL> CREATE TABLE Product ( Product_ID VARCHAR2(6), Product_name VARCHAR2(50), Description VARCHAR2(50), Rate NUMBER(10) NOT NULL, Stock_quantity NUMBER(10) NOT NULL, Availability VARCHAR2(50), Vendor_ID VARCHAR2(6), Product_Category_ID VARCHAR2(6), CONSTRAINT Product_pk PRIMARY KEY(Product_ID), FOREIGN KEY(Vendor_ID) REFERENCES Vendor(Vendor_ID), FOREIGN KEY(Product_Category_ID) REFERENCES Product_Category(Product_Category_ID));
```

Table created.

```
SQL> DESC Product;
```

Name	Null?	Type
PRODUCT_ID	NOT NULL	VARCHAR2(6)
PRODUCT_NAME		VARCHAR2(50)
DESCRIPTION		VARCHAR2(50)
RATE	NOT NULL	NUMBER(10)
STOCK_QUANTITY	NOT NULL	NUMBER(10)
AVAILABILITY		VARCHAR2(50)
VENDOR_ID		VARCHAR2(6)
PRODUCT_CATEGORY_ID		VARCHAR2(6)

Figure 9 Creating and describing the table of Product.

f. Customer

```
SQL> CREATE TABLE Customer (Customer_ID VARCHAR2(6), Customer_name VARCHAR2(50), Customer_email VARCHAR2(50), Customer_address VARCHAR2(50), Customer_Category_ID VARCHAR2(6), CONSTRAINT Customer_pk PRIMARY KEY(Customer_ID), FOREIGN KEY(Customer_Category_ID) REFERENCES Customer_Category(Customer_Category_ID));
```

Table created.

```
SQL> DESC Customer;
```

Name	Null?	Type
CUSTOMER_ID	NOT NULL	VARCHAR2(6)
CUSTOMER_NAME		VARCHAR2(50)
CUSTOMER_EMAIL		VARCHAR2(50)
CUSTOMER_ADDRESS		VARCHAR2(50)
CUSTOMER_CATEGORY_ID		VARCHAR2(6)

SQL>

Figure 10 Creating and describing the table of Customer.

g. Orders

```
SQL> CREATE TABLE Orders (Order_ID VARCHAR2(6), Order_Date DATE, Discount VARCHAR2(50), Total_amount NUMBER(10) NOT NULL, Customer_ID VARCHAR2(6), Payment_type_ID VARCHAR2(6), CONSTRAINT Order_pk PRIMARY KEY(Order_ID), FOREIGN KEY(Customer_ID) REFERENCES Customer(Customer_ID), FOREIGN KEY(Payment_type_ID) REFERENCES Payment_type(Payment_type_ID));
```

Table created.

```
SQL> DESC Orders;
```

Name	Null?	Type
ORDER_ID	NOT NULL	VARCHAR2(6)
ORDER_DATE		DATE
DISCOUNT		VARCHAR2(50)
TOTAL_AMOUNT	NOT NULL	NUMBER(10)
CUSTOMER_ID		VARCHAR2(6)
PAYMENT_TYPE_ID		VARCHAR2(6)

Figure 11 Creating and describing the table of Orders.

h. Order_Product

```
SQL> CREATE TABLE Order_Product (
2   Order_ID VARCHAR2(6),
3   Product_ID VARCHAR2(6),
4   Quantity NUMBER(10),
5   Price NUMBER(10),
6   Line_Total NUMBER(20, 2) GENERATED ALWAYS AS (Quantity * Price) VIRTUAL,
7   CONSTRAINT Order_Product_pk PRIMARY KEY (Order_ID, Product_ID),
8   FOREIGN KEY (Order_ID) REFERENCES Orders(Order_ID),
9   FOREIGN KEY (Product_ID) REFERENCES Product(Product_ID)
10 );
```

Table created.

Figure 12 Creating the table of Order_Product.

```
SQL> DESC Order_Product;
```

Name	Null?	Type
ORDER_ID	NOT NULL	VARCHAR2(6)
PRODUCT_ID	NOT NULL	VARCHAR2(6)
QUANTITY		NUMBER(10)
PRICE		NUMBER(10)
LINE_TOTAL		NUMBER(20, 2)

Figure 13 Describing the table of Order_Product.

5.4. Data Insertion

a. Vendor

```
SQL> INSERT ALL
  2 INTO Vendor VALUES ('VE01', 'Abhinav Baral')
  3 INTO Vendor VALUES ('VE02', 'Rishab Khadka')
  4 INTO Vendor VALUES ('VE03', 'Ayush Das')
  5 INTO Vendor VALUES ('VE04', 'Manoj Paudel')
  6 INTO Vendor VALUES ('VE05', 'Suman Shrestha')
  7 INTO Vendor VALUES ('VE06', 'Hari Pradhan')
  8 INTO Vendor VALUES ('VE07', 'Binod Dhakal')
  9 INTO Vendor VALUES ('VE08', 'Suraj Kharel')
 10 SELECT * FROM DUAL;
```

Figure 14 Inserting Values in the Vendor table.

b. Product_Category

```
SQL> INSERT ALL
  2 INTO Product_Category VALUES ('PG01', 'Laptops')
  3 INTO Product_Category VALUES ('PG02', 'Mobile Phones')
  4 INTO Product_Category VALUES ('PG03', 'Television')
  5 INTO Product_Category VALUES ('PG04', 'Camera')
  6 INTO Product_Category VALUES ('PG05', 'PC')
  7 INTO Product_Category VALUES ('PG06', 'Chargers and Wires')
  8 INTO Product_Category VALUES ('PG07', 'Bluetooth Speakers')
  9 INTO Product_Category VALUES ('PG08', 'Keyboard')
 10 INTO Product_Category VALUES ('PG09', 'Mouse')
 11 INTO Product_Category VALUES ('PG10', 'Headphones')
 12 SELECT * FROM DUAL;

10 rows created.
```

Figure 15 Inserting Values in the Product_Category table.

c. Customer_Category

```
SQL> INSERT ALL
  2 INTO Customer_Category VALUES ('R', 'Regular','0%')
  3 INTO Customer_Category VALUES ('S', 'Staff','5%')
  4 INTO Customer_Category VALUES ('V', 'VIP','10%')
  5 SELECT * FROM DUAL;
```

Figure 16 Inserting Values in the Customer_Category table.

d. Payment_type

```
SQL> INSERT ALL
  2 INTO Payment_type VALUES ('PAT01', 'Cash on Delivery')
  3 INTO Payment_type VALUES ('PAT02', 'Credit/Debit Card')
  4 INTO Payment_type VALUES ('PAT03', 'E-Wallet')
  5 SELECT * FROM DUAL;

3 rows created.
```

Figure 17 Inserting Values in the Payment_type table.

e. Product

```
SQL> INSERT ALL
  2 INTO Product VALUES ('PR01', 'Acer N5', 'This laptop was released in 2023 and has intel i9',80000, 50, 'Available', 'VE01', 'PG01')
  3 INTO Product VALUES ('PR02', 'Dell Inspiron', 'This laptop was released in 2021 and has intel i7',45000, 30, 'Available', 'VE01', 'PG01')
  4 INTO Product VALUES ('PR03', 'Lenovo Legion 5 Pro', 'This laptop was released in 2023 and has ryzen 7',120000, 0, 'Sold Out', 'VE01', 'PG01')
  5 INTO Product VALUES ('PR04', 'Realme 10', 'This Phone was released in 2023 with 128GB storage',40000, 65, 'Available', 'VE02', 'PG02')
  6 INTO Product VALUES ('PR05', 'Iphone 15', 'This Phone released in 2024 has 128 GB storage',180000, 12, 'Limited Quantity', 'VE02', 'PG02')
  7 INTO Product VALUES ('PR06', 'Samsung A26', 'This Phone released in 2024 has 128 GB storage',95000, 65, 'Available', 'VE02', 'PG02')
  8 INTO Product VALUES ('PR07', 'Barton PC', 'This is pre built pc with ryzen 7 and storage 1TB',12000, 02, 'Available', 'VE01', 'PG05')
  9 INTO Product VALUES ('PR08', 'Samsung 65', 'The samsung 65 is and 65 inch Oled releaded 2023',80000, 60, 'Available', 'VE03', 'PG03')
  10 INTO Product VALUES ('PR09', 'Samsung 50', 'The Samsung 50 is 50 inch tv releaded 2022',70000, 77, 'Available', 'VE03', 'PG03')
  11 INTO Product VALUES ('PR10', 'Canon m5', 'The canon m5 is a camera that has AI retouch',160000, 51, 'Available', 'VE06', 'PG04')
  12 INTO Product VALUES ('PR11', 'Nikon Jett 4', 'nikon jett 4 is camera built for video',140000, 24, 'Available', 'VE06', 'PG04')
  13 INTO Product VALUES ('PR12', 'Beats Booster', 'The beats booster has Anc',2000, 120, 'Available', 'VE05', 'PG10')
  14 INTO Product VALUES ('PR13', 'JBL Bluebeats', 'It is a compact bluetooth speaker',5000, 22, 'Available', 'VE08', 'PG07')
  15 INTO Product VALUES ('PR14', 'Macro Charger', 'It is a type c which support fast charge',200, 0, 'Sold Out', 'VE04', 'PG06')
  16 INTO Product VALUES ('PR15', 'Dell Hello', 'Dell hello charger is a chargers for laptops',540, 74, 'available', 'VE04', 'PG06')
  17 INTO Product VALUES ('PR16', 'Dell keyboard', 'dell keyboard is a membrane keyboard for office',1000, 240, 'available', 'VE07', 'PG08')
  18 INTO Product VALUES ('PR17', 'Fantech K502', 'fantech k502 is a mechanical keyboard',4500, 45, 'available', 'VE07', 'PG08')
  19 INTO Product VALUES ('PR18', 'Lenovo digi', 'lenovo digi is a normal optical mouse for office',400, 220, 'available', 'VE07', 'PG09')
  20 INTO Product VALUES ('PR19', 'Redragon mirage 122', 'Redragon mirage 122 is a gaming mouse', 1200, 31, 'available', 'VE07', 'PG09')
  21 SELECT * FROM DUAL;

19 rows created.
```

Figure 18 Inserting Values in the Product table.

f. Customer

```

SQL> INSERT ALL
  2 INTO Customer VALUES ('CUS01','Prakash Subedi', 'prakash1@gmail.com', 'kumaripati,lalitpur', 'S')
  3 INTO Customer VALUES ('CUS02','Raman Chaudhary', 'raman32@gmail.com', 'Jhapa', 'R')
  4 INTO Customer VALUES ('CUS03','Shretha Technologies', 'shresthatech@gmail.com', 'Thamel,kathmandu', 'V')
  5 INTO Customer VALUES ('CUS04','Nishesh bhattarai', 'nishesh@gmail.com', 'kamalPokhari Kathmandu', 'R')
  6 INTO Customer VALUES ('CUS05','Sayunk Dhakal', 'Sayu123@gmail.com', 'Sinamangal, Kathmandu', 'R')
  7 INTO Customer VALUES ('CUS06','VT Tech Solutions', 'VTtechSolution@gmail.com', 'Sanobharyang', 'V')
  8 INTO Customer VALUES ('CUS07','Binaya Kharel', 'khrelbhai@gmail.com', 'patan, mangalbazar', 'S')
  9 INTO Customer VALUES ('CUS08','Pritesh Lamichhane', 'pritesha@gmail.com', 'Swoyambhu', 'S')
 10 INTO Customer VALUES ('CUS09','Ram bhandari', 'ram@gmail.com', 'Bhaktapur', 'R')
 11 INTO Customer VALUES ('CUS10','Abhinav Adhikari', 'ram@gmail.com', 'Jawalakhe, pulchowk', 'R')
 12 SELECT * FROM DUAL;

10 rows created.

```

Figure 19 Inserting Values in the Customer table.

g. Orders

```

SQL> INSERT ALL
  2 INTO Orders VALUES ('ORD01','01-JAN-23', '10%', '72000', 'CUS03', 'PAT01')
  3 INTO Orders VALUES ('ORD02','24-JAN-23', '0%', 4500, 'CUS10', 'PAT03')
  4 INTO Orders VALUES ('ORD03','10-FEB-23', '0%', 160000, 'CUS09', 'PAT01')
  5 INTO Orders VALUES ('ORD04','26-FEB-23', '5%', 80750, 'CUS07', 'PAT02')
  6 INTO Orders VALUES ('ORD05','20-MAR-23', '0%', 180000, 'CUS02', 'PAT02')
  7 INTO Orders VALUES ('ORD06','07-APR-23', '0%', 85200, 'CUS04', 'PAT03')
  8 INTO Orders VALUES ('ORD07','18-APR-23', '5%', 42750, 'CUS01', 'PAT01')
  9 INTO Orders VALUES ('ORD08','03-MAY-23', '10%', 50400, 'CUS03', 'PAT02')
 10 INTO Orders VALUES ('ORD09','16-MAY-23', '10%', 360000, 'CUS06', 'PAT02')
 11 INTO Orders VALUES ('ORD10','28-MAY-23', '0%', 80000, 'CUS10', 'PAT01')
 12 INTO Orders VALUES ('ORD11','11-JUN-23', '0%', 70000, 'CUS09', 'PAT02')
 13 INTO Orders VALUES ('ORD12','22-JUL-23', '0%', 200, 'CUS02', 'PAT01')
 14 INTO Orders VALUES ('ORD13','01-AUG-23', '5%', 1900, 'CUS07', 'PAT01')
 15 INTO Orders VALUES ('ORD14','09-AUG-23', '5%', 513, 'CUS07', 'PAT01')
 16 INTO Orders VALUES ('ORD15','18-AUG-23', '0%', 140000, 'CUS04', 'PAT02')
 17 INTO Orders VALUES ('ORD16','19-OCT-23', '10%', 630000, 'CUS06', 'PAT02')
 18 INTO Orders VALUES ('ORD17','20-DEC-23', '0%', 40000, 'CUS05', 'PAT03')
 19 INTO Orders VALUES ('ORD18','29-DEC-23', '0%', 5000, 'CUS05', 'PAT03')
 20 SELECT * FROM DUAL;

18 rows created.

```

Figure 20 Inserting Values in the Orders table.

h. Order_Product

```
SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD01', 'PR01', 1, 80000);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD02', 'PR17', 1, 4500);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD03', 'PR10', 1, 160000);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD04', 'PR08', 1, 80000);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD04', 'PR13', 1, 5000);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD05', 'PR05', 1, 180000);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD06', 'PR01', 1, 80000);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD06', 'PR13', 1, 5000);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD06', 'PR14', 1, 200);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD07', 'PR02', 1, 45000);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD08', 'PR16', 40, 1000);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD08', 'PR18', 40, 400);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD09', 'PR08', 5, 80000);
```

Figure 21 Inserting Values in the Order_Product table.

```
1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD10', 'PR08', 1, 80000);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD11', 'PR09', 1, 70000);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD12', 'PR14', 1, 200);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD13', 'PR12', 1, 2000);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD14', 'PR15', 1, 540);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD15', 'PR11', 1, 140000);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD16', 'PR09', 10, 70000);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD17', 'PR04', 1, 40000);

1 row created.

SQL> INSERT INTO Order_Product (Order_ID, Product_ID, Quantity, Price)
  2 VALUES ('ORD18', 'PR13', 1, 5000);

1 row created.
```

Figure 22 Inserting Values in the Order_Product table 2.

5.5. Displaying Data

a. Vendor

```
SQL> SELECT * FROM Vendor;

VENDOR_ID          VENDOR_NAME
-----
VE01               Abhinav Baral
VE02               Rishab Khadka
VE03               Ayush Das
VE04               Manoj Paudel
VE05               Suman Shrestha
VE06               Hari Pradhan
VE07               Binod Dhakal
VE08               Suraj Kharel

8 rows selected.
```

Figure 23 Displaying the Data of the table Vendor.

b. Product_Category

```
SQL> SELECT * FROM Product_Category;

PRODUCT_CATEGORY_ID  PRODUCT_CATEGORY_NAME
-----
PG01                 Laptops
PG02                 Mobile Phones
PG03                 Television
PG04                 Camera
PG05                 PC
PG06                 Chargers and Wires
PG07                 Bluetooth Speakers
PG08                 Keyboard
PG09                 Mouse
PG10                 Headphones

10 rows selected.
```

Figure 24 Displaying the Data of the table Product_Category.

c. Customer_Category

```
CUSTOMER_CATEGORY_ID  CUSTOMER_CATEGORY_NAME  DISCOUNT_RATE
-----
R                     Regular                 0%
S                     Staff                  5%
V                     VIP                    10%

3 rows selected.
```

Figure 25 Displaying the Data of the table Customer_Category.

d. Payment_type

```
SQL> SELECT * FROM Payment_type;
```

PAYMENT_TYPE_ID	PAYMENT_TYPE_NAME
PAT01	Cash on Delivery
PAT02	Credit/Debit Card
PAT03	E-Wallet

3 rows selected.

Figure 26 Displaying the Data of the table Payment_type.

e. Product

```
SQL> SELECT * FROM Product;
```

PRODUCT_ID	PRODUCT_NAME	DESCRIPTION	RATE	STOCK_QUANTITY	AVAILABILITY	VENDOR_ID	PRODUCT_CATEGORY_ID
PR01	Acer N5	This laptop was released in 2023 and has intel i9	80000	50	Available	VE01	PG01
PR02	Dell Inspiron	This laptop was released in 2021 and has intel i7	45000	30	Available	VE01	PG01
PR03	Lenovo Legion 5 Pro	This laptop was released in 2023 and has ryzen 7	120000	0	Sold Out	VE01	PG01
PR04	Realme 10	This Phone was released in 2023 with 128GB storage	40000	65	Available	VE02	PG02
PR05	Iphone 15	This Phone released in 2024 has 128 GB storage	180000	12	Limited Quantity	VE02	PG02
PR06	Samsung A26	This Phone released in 2024 has 128 GB storage	95000	65	Available	VE02	PG02
PR07	Barton PC	This is pre built pc with ryzen 7 and storage 1TB	12000	2	Available	VE01	PG05
PR08	Samsung 65	The samsung 65 is and 65 inch Oled releaded 2023	80000	60	Available	VE03	PG03
PR09	Samsung 59	The Samsung 59 is 59 inch tv releaded 2022	70000	77	Available	VE03	PG03
PR10	Canon m5	The canon m5 is a camera that has AI retouch	160000	51	Available	VE06	PG04
PR11	Nikon Jett 4	nikon jett 4 is camera built for video	140000	24	Available	VE06	PG04
PR12	Beats Booster	The beats booster has Anc	2000	120	Available	VE05	PG10
PR13	JBL Bluebeats	It is a compact bluetooth speaker	5000	22	Available	VE08	PG07
PR14	Macro Charger	It is a type c which support fast charge	200	0	Sold Out	VE04	PG06
PR15	Dell Hello	Dell hello charger is a chargers for Laptops	540	74	available	VE04	PG06
PR16	Dell keyboard	dell keyboard is a membrane keyboard for office	1000	200	available	VE07	PG08
PR17	Fantech K502	fantech K502 is a mechanical keyboard	4500	45	available	VE07	PG08
PR18	lenovo digi	lenovo digi is a normal optical mouse for office	400	220	available	VE07	PG09
PR19	Redragon mirage 122	Redragon mirage 122 is a gaming mouse	1200	31	available	VE07	PG09

19 rows selected.

Figure 27 Displaying the Data of the table Product.

f. Customer

```
SQL> SELECT * FROM Customer;
```

CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_EMAIL	CUSTOMER_ADDRESS	CUSTOMER CATEGORY_ID
CUS01	Prakash Subedi	prakash1@gmail.com	kumaripati,lalitpur	S
CUS02	Raman Chaudhary	raman32@gmail.com	Jhapa	R
CUS03	Shretha Technologies	shresthatech@gmail.com	Thamel,kathmandu	V
CUS04	Nishesh bhattarai	nishesh@gmail.com	kamalPokhari Kathmandu	R
CUS05	Sayunk Dhakal	Sayul23@gmail.com	Sinamangal, Kathmandu	R
CUS06	VT Tech Solutions	VTtechSolution@gmail.com	Sanobharyang	V
CUS07	Binaya Kharel	khrelbhai@gmail.com	patan, mangalbazar	S
CUS08	Pritesh Lamichhane	pritesha@gmail.com	Swoyambhu	S
CUS09	Ram bhandari	ram@gmail.com	Bhaktapur	R
CUS10	Abhinav Adhikari	ram@gmail.com	Jawalakhel, pulchowk	R

10 rows selected.

Figure 28 Displaying the Data of the table Customer.

g. Orders

```
SQL> SELECT * FROM Orders;
```

ORDER_ID	ORDER_DATE	DISCOUNT	TOTAL_AMOUNT	CUSTOMER_ID	PAYMENT_TYPE_ID
ORD01	01-JAN-23	10%	72000	CUS03	PAT01
ORD02	24-JAN-23	0%	4500	CUS10	PAT03
ORD03	10-FEB-23	0%	160000	CUS09	PAT01
ORD04	26-FEB-23	5%	80750	CUS07	PAT02
ORD05	20-MAR-23	0%	180000	CUS02	PAT02
ORD06	07-APR-23	0%	85200	CUS04	PAT03
ORD07	18-APR-23	5%	42750	CUS01	PAT01
ORD08	03-MAY-23	10%	50400	CUS03	PAT02
ORD09	16-MAY-23	10%	360000	CUS06	PAT02
ORD10	28-MAY-23	0%	80000	CUS10	PAT01
ORD11	11-JUN-23	0%	70000	CUS09	PAT02
ORD12	22-JUL-23	0%	200	CUS02	PAT01
ORD13	01-AUG-23	5%	1900	CUS07	PAT01
ORD14	09-AUG-23	5%	513	CUS07	PAT01
ORD15	18-AUG-23	0%	140000	CUS04	PAT02
ORD16	19-OCT-23	10%	630000	CUS06	PAT02
ORD17	20-DEC-23	0%	40000	CUS05	PAT03
ORD18	29-DEC-23	0%	5000	CUS05	PAT03

18 rows selected.

Figure 29 Displaying the Data of the table Orders.

h. Order_Product

```
SQL> SELECT * FROM Order_Product;
```

ORDER_ID	PRODUCT_ID	QUANTITY	PRICE	LINE_TOTAL
ORD01	PR01	1	80000	80000
ORD02	PR17	1	4500	4500
ORD03	PR10	1	160000	160000
ORD04	PR08	1	80000	80000
ORD04	PR13	1	5000	5000
ORD05	PR05	1	180000	180000
ORD06	PR01	1	80000	80000
ORD06	PR13	1	5000	5000
ORD06	PR14	1	200	200
ORD07	PR02	1	45000	45000
ORD08	PR16	40	1000	40000
ORD08	PR18	40	400	16000
ORD09	PR08	5	80000	400000
ORD10	PR08	1	80000	80000
ORD11	PR09	1	70000	70000
ORD12	PR14	1	200	200
ORD13	PR12	1	2000	2000
ORD14	PR15	1	540	540
ORD15	PR11	1	140000	140000
ORD16	PR09	10	70000	700000
ORD17	PR04	1	40000	40000
ORD18	PR13	1	5000	5000

22 rows selected.

Figure 30 Displaying the Data of the table Order_Product

5.6. Creating Dump File

```

C:\Users\LEGION> D:
D:\> cd Database Coursework\Dump file
D:\Database Coursework\Dump file> exp GadgetEmporium/Gadgetemporium file = Gadgetemporium.dmp
Export: Release 11.2.0.2.0 - Production on Sun Jan 14 14:48:56 2024
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
Export done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set
server uses AL32UTF8 character set (possible charset conversion)
. exporting pre-schema procedural objects and actions
. exporting foreign function library names for user GADGETEMPORIUM
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user GADGETEMPORIUM
About to export GADGETEMPORIUM's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export GADGETEMPORIUM's tables via Conventional Path ...
. . exporting table CUSTOMER 10 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table CUSTOMER_CATEGORY 3 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table ORDERS 18 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table ORDER_PRODUCT
EXP-00107: Feature (VIRTUAL COLUMN) of column LINE_TOTAL in table GADGETEMPORIUM.ORDER_PRODUCT is not supported. The table will not be exported.
. . exporting table PAYMENT_TYPE 3 rows exported
EXP-00091: Exporting questionable statistics.

```

Figure 31 Creating Dump File.

5.7. Dropping Tables

```
SQL> DROP TABLE Order_Product;

Table dropped.

SQL> DROP TABLE Order;
DROP TABLE Order
      *
ERROR at line 1:
ORA-00903: invalid table name

SQL> DROP TABLE Orders;

Table dropped.

SQL> DROP TABLE Customer;

Table dropped.

SQL> DROP TABLE Product;

Table dropped.

SQL> DROP TABLE Payment_type;

Table dropped.

SQL> DROP TABLE Customer_Category;

Table dropped.

SQL> DROP TABLE Product_Category;

Table dropped.

SQL> DROP TABLE Vendor;

Table dropped.

SQL> |
```

Figure 32 Dropping tables.

6. Database Querying

6.1. Information Query

1. List all customers that are also a staff of the company.

```
SQL> spool 'D:\Database Coursework\Information_Query_1.sql'
SQL> SELECT * FROM Customer WHERE Customer_Category_ID = 'S';

no rows selected

SQL> SELECT * FROM Customer WHERE Customer_Category_ID = 'S';
```

CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_EMAIL	CUSTOMER_ADDRESS	CUSTOMER_CATEGORY_ID
CUS01	Prakash Subedi	prakash1@gmail.com	kumaripati,lalitpur	S
CUS07	Binaya Kharel	khrelbhai@gmail.com	patan, mangalbazar	S
CUS08	Pritesh Lamichhane	pritesha@gmail.com	Swoyambhu	S

```
3 rows selected.

SQL> spool off;
```

Figure 33 Information Query 1.

2. List all orders that are made for any particular product between the dates 01-05-2023 till 28-05-2023.

```
SQL> spool 'D:\Database Coursework\Information_Query_2.sql'
SQL> SELECT o.ORDER_ID, o.ORDER_DATE, o.DISCOUNT, o.TOTAL_AMOUNT, o.CUSTOMER_ID, o.PAYMENT_TYPE_ID
2 FROM ORDERS o
3 JOIN ORDER_PRODUCT op ON o.ORDER_ID = op.ORDER_ID
4 WHERE o.ORDER_DATE BETWEEN TO_DATE('01-05-2023', 'DD-MM-YYYY') AND TO_DATE('28-05-2023', 'DD-MM-YYYY')
5 AND op.PRODUCT_ID = 'PR08';
```

ORDER_ID	ORDER_DATE	DISCOUNT	TOTAL_AMOUNT	CUSTOMER_ID	PAYMENT_TYPE_ID
ORD09	16-MAY-23	10%	360000	CUS06	PAT02
ORD10	28-MAY-23	0%	80000	CUS10	PAT01

```
2 rows selected.

SQL> spool off;
```

Figure 34 Information Query 2.

3. List all the customers with their order details and those customers who have not ordered any products yet.

```
SQL> spool 'D:\Database Coursework\Information_Query_3.sql'
SQL> SELECT c.CUSTOMER_ID, c.CUSTOMER_NAME, c.CUSTOMER_EMAIL, c.CUSTOMER_ADDRESS, c.CUSTOMER_CATEGORY_ID,
2 o.ORDER_ID, o.ORDER_DATE, o.DISCOUNT, o.TOTAL_AMOUNT, o.PAYMENT_TYPE_ID
3 FROM CUSTOMER c
4 LEFT JOIN ORDERS o ON c.CUSTOMER_ID = o.CUSTOMER_ID
5 ORDER BY c.CUSTOMER_ID, o.ORDER_DATE;
```

CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_EMAIL	CUSTOMER_ADDRESS	CUSTOMER_CATEGORY_ID	ORDER_ID	ORDER_DATE	DISCOUNT	TOTAL_AMOUNT	PAYMENT_TYPE_ID
CUS01	Prakash Subedi	prakash1@gmail.com	kumaripati,lalitpur	S	ORD07	18-APR-23	5%	42750	PAT01
CUS02	Raman Chaudhary	raman32@gmail.com	Jhapa	R	ORD05	20-MAR-23	0%	180000	PAT02
CUS02	Raman Chaudhary	raman32@gmail.com	Jhapa	R	ORD12	22-JUL-23	0%	260	PAT01
CUS03	Shretha Technologies	shresthatech@gmail.com	Thamel,kathmandu	V	ORD01	01-JAN-23	10%	72000	PAT01
CUS03	Shretha Technologies	shresthatech@gmail.com	Thamel,kathmandu	V	ORD08	03-MAY-23	10%	50400	PAT02
CUS04	Nishesh bhattarai	nishesh@gmail.com	kanalPokhari Kathmandu	R	ORD06	07-APR-23	0%	85200	PAT03
CUS04	Nishesh bhattarai	nishesh@gmail.com	kanalPokhari Kathmandu	R	ORD15	18-AUG-23	0%	140000	PAT02
CUS05	Sayunk Dhakal	Sayul23@gmail.com	Sinanangal, Kathmandu	R	ORD17	20-DEC-23	0%	40000	PAT03
CUS05	Sayunk Dhakal	Sayul23@gmail.com	Sinanangal, Kathmandu	R	ORD18	20-DEC-23	0%	5000	PAT03
CUS06	VT Tech Solutions	VTtechSolution@gmail.com	Sanobharyang	V	ORD09	16-MAY-23	10%	360000	PAT02
CUS06	VT Tech Solutions	VTtechSolution@gmail.com	Sanobharyang	V	ORD16	19-OCT-23	10%	630000	PAT02
CUS07	Binaya Kharel	khrelbhai@gmail.com	patan, mangalbazar	S	ORD04	26-FEB-23	5%	80750	PAT02
CUS07	Binaya Kharel	khrelbhai@gmail.com	patan, mangalbazar	S	ORD13	01-AUG-23	5%	1900	PAT01
CUS07	Binaya Kharel	khrelbhai@gmail.com	patan, mangalbazar	S	ORD14	09-AUG-23	5%	513	PAT01
CUS08	Pritesh Lamichhane	pritesha@gmail.com	Swoyambhu	S					
CUS09	Ram bhandari	ram@gmail.com	Bhaktapur	R	ORD03	10-FEB-23	0%	160000	PAT01
CUS09	Ram bhandari	ram@gmail.com	Bhaktapur	R	ORD11	11-JUN-23	0%	70000	PAT02
CUS10	Abhinav Adhikari	ram@gmail.com	Jawalakhel, pulchowk	R	ORD02	24-JAN-23	0%	4500	PAT03
CUS10	Abhinav Adhikari	ram@gmail.com	Jawalakhel, pulchowk	R	ORD10	28-MAY-23	0%	80000	PAT01

```
19 rows selected.

SQL> spool off;
```

Figure 35 Information Query 3.

4. List all product details that have the second letter 'a' in their product name and have a stock quantity more than 50.

```
SQL> spool 'D:\Database Coursework\Information_Query_4.sql'
SQL> SELECT *
  2 FROM Product
  3 WHERE SUBSTR(PRODUCT_NAME, 2, 1) = 'a' AND STOCK_QUANTITY > 50;

PRODUCT_ID  PRODUCT_NAME  DESCRIPTION  RATE  STOCK_QUANTITY  AVAILABILITY  VENDOR_ID  PRODUCT_CATEGORY_ID
-----
PR06        Samsung A26   This Phone released in 2024 has 128 GB storage  95000      65 Available    VE02        PG02
PR08        Samsung 65   The samsung 65 is and 65 inch Oled releaded 2023  80000      60 Available    VE03        PG03
PR09        Samsung 50   The Samsung 50 is 50 inch tv releaded 2022    70000      77 Available    VE03        PG03
PR10        Canon m5     The canon m5 is a camera that has AI retouch    160000     51 Available    VE06        PG04

4 rows selected.

SQL> spool off;
```

Figure 36 Information Query 4.

5. Find out the customer who has ordered recently.

```
SQL> spool 'D:\Database Coursework\Information_Query_5.sql'
SQL> SELECT C.*
  2 FROM Customer C
  3 JOIN (
  4     SELECT CUSTOMER_ID, MAX(ORDER_DATE) AS RECENT_ORDER_DATE
  5     FROM Orders
  6     GROUP BY CUSTOMER_ID
  7 ) O ON C.CUSTOMER_ID = O.CUSTOMER_ID
  8 ORDER BY RECENT_ORDER_DATE DESC;

CUSTOMER_ID  CUSTOMER_NAME  CUSTOMER_EMAIL  CUSTOMER_ADDRESS  CUSTOMER CATEGORY_ID
-----
CUS05        Sayunk Dhakal  Sayu123@gmail.com  Sinamangal, Kathmandu  R
CUS06        VT Tech Solutions  VTtechSolution@gmail.com  Sanobharyang  V
CUS04        Nishesh bhattarai  nishesh@gmail.com  kamalPokhari Kathmandu  R
CUS07        Binaya Kharel  khrelbhai@gmail.com  patan, mangalbazar  S
CUS02        Raman Chaudhary  raman32@gmail.com  Jhapa  R
CUS09        Ram bhandari  ram@gmail.com  Bhaktapur  R
CUS10        Abhinav Adhikari  ram@gmail.com  Jawalakhel, pulchowk  R
CUS03        Shretha Technologies  shresthatech@gmail.com  Thamel, Kathmandu  V
CUS01        Prakash Subedi  prakash1@gmail.com  kumaripati, Lalitpur  S

9 rows selected.

SQL> spool off;
```

Figure 37 Information Query 5.

6.2. Transaction Query

1. Show the total revenue of the company for each month.

```
SQL> spool 'D:\Database Coursework\Transaction_Query_1.sql'
SQL> SELECT
  2     TO_CHAR(ORDER_DATE, 'MM-YYYY') AS MONTH_YEAR,
  3     SUM(TOTAL_AMOUNT) AS TOTAL_REVENUE
  4 FROM
  5     Orders
  6 GROUP BY
  7     TO_CHAR(ORDER_DATE, 'MM-YYYY')
  8 ORDER BY
  9     TO_CHAR(ORDER_DATE, 'MM-YYYY');

MONTH_Y TOTAL_REVENUE
-----
01-2023      76500
02-2023     240750
03-2023     180000
04-2023     127950
05-2023     490400
06-2023      70000
07-2023         200
08-2023     142413
10-2023     630000
12-2023      45000

10 rows selected.

SQL> spool off;
SP2-0042: unknown command "spool off" - rest of line ignored.
SQL> spool off;
SQL> |
```

Figure 38 Transaction Query 1.

2. Find those orders that are equal or higher than the average order total value.

```
SQL> spool 'D:\Database Coursework\Transaction_Query_2.sql'
SQL> SELECT * FROM Orders WHERE TOTAL_AMOUNT >= (SELECT AVG(TOTAL_AMOUNT) FROM Orders);

ORDER_ID      ORDER_DATE      DISCOUNT      TOTAL_AMOUNT  CUSTOMER_ID  PAYMENT_TYPE_ID
-----
ORD03         10-FEB-23       0%             160000      CUS09        PAT01
ORD05         20-MAR-23       0%             180000      CUS02        PAT02
ORD09         16-MAY-23       10%            360000      CUS06        PAT02
ORD15         18-AUG-23       0%             140000      CUS04        PAT02
ORD16         19-OCT-23       10%            630000      CUS06        PAT02

5 rows selected.

SQL> spool off;
```

Figure 39 Transaction Query 2.

- List the details of vendors who have supplied more than 3 products to the company.

```
SQL> spool 'D:\Database Coursework\Transaction_Query_3.sql'
SQL> SELECT V.VENDOR_ID, V.VENDOR_NAME, COUNT(P.PRODUCT_ID) AS PRODUCTS_SUPPLIED
2 FROM VENDOR V
3 JOIN PRODUCT P ON V.VENDOR_ID = P.VENDOR_ID
4 GROUP BY V.VENDOR_ID, V.VENDOR_NAME
5 HAVING COUNT(P.PRODUCT_ID) > 3;
```

VENDOR_ID	VENDOR_NAME	PRODUCTS_SUPPLIED
VE07	Binod Dhakal	4
VE01	Abhinav Baral	4

2 rows selected.

Figure 40 Transaction Query 3.

- Show the top 3 product details that have been ordered the most.

```
SQL> spool 'D:\Database Coursework\Transaction_Query_4.sql'
SQL> SELECT *
2 FROM (
3 SELECT P.PRODUCT_ID, P.PRODUCT_NAME, P.DESRIPTION, P.RATE, P.STOCK_QUANTITY, P.AVAILABILITY, P.VENDOR_ID, P.PRODUCT_CATEGORY_ID,
4 COUNT(O.ORDER_ID) AS ORDER_COUNT
5 FROM PRODUCT P
6 JOIN ORDER_PRODUCT OP ON P.PRODUCT_ID = OP.PRODUCT_ID
7 JOIN ORDERS O ON OP.ORDER_ID = O.ORDER_ID
8 GROUP BY P.PRODUCT_ID, P.PRODUCT_NAME, P.DESRIPTION, P.RATE, P.STOCK_QUANTITY, P.AVAILABILITY, P.VENDOR_ID, P.PRODUCT_CATEGORY_ID
9 ORDER BY ORDER_COUNT DESC
10 )
11 WHERE ROWNUM <= 3;
```

PRODUCT_ID	PRODUCT_NAME	DESCRIPTION	RATE	STOCK_QUANTITY	AVAILABILITY	VENDOR_ID	PRODUCT_CATEGORY_ID	ORDER_COUNT
PR13	JBL Bluebeats	It is a compact bluetooth speaker	5000	22	Available	VE08	PG07	3
PR08	Samsung 65	The samsung 65 is and 65 inch Oled releaded 2023	80000	60	Available	VE03	PG03	3
PR01	Acer N5	This laptop was released in 2023 and has intel I9	80000	50	Available	VE01	PG01	2

3 rows selected.

SQL> SPOOL OFF;

Figure 41 Transaction Query 4.

- Find out the customer who has ordered the most in August with his/her total spending on that month.

```
SQL> spool 'D:\Database Coursework\Transaction_Query_5.sql'
SQL> SELECT CUSTOMER_ID, CUSTOMER_NAME, TOTAL_SPENDING
2 FROM (
3 SELECT C.CUSTOMER_ID, C.CUSTOMER_NAME, SUM(O.TOTAL_AMOUNT) AS TOTAL_SPENDING,
4 ROW_NUMBER() OVER (ORDER BY SUM(O.TOTAL_AMOUNT) DESC) AS RN
5 FROM CUSTOMER C
6 JOIN ORDERS O ON C.CUSTOMER_ID = O.CUSTOMER_ID
7 WHERE TO_CHAR(O.ORDER_DATE, 'MM') = '08' -- Filter orders in August
8 GROUP BY C.CUSTOMER_ID, C.CUSTOMER_NAME
9 )
10 WHERE RN = 1;
```

CUSTOMER_ID	CUSTOMER_NAME	TOTAL_SPENDING
CUS04	Nishesh bhattarai	140000

1 row selected.

SQL> spool off;

Figure 42 Transaction Query 5.

7. Critical Evaluation

7.1. Critical Evaluation of the Module

The module Database {CC505NI} is a module taught and conducted in the second year of the London Metropolitan University's BSc(Hons) Computing faculty. The module is conducted through lecture, tutorial and workshop every week. The module contains a module leader who leads the module resources and topics, lecturers who conduct lecture classes for us to understand about the module and database background, and tutors who conduct tutorial classed as well as lab classes who guide us through the topics taught in the lecture and for us to put in real life scenarios.

The module provides us a wide range of knowledge about the database. It allows us to gain knowledge about the database system, how the system works, how to handle the database, create a database and more to follow. The module not only provides knowledge but also allows us to apply the knowledge gained in the module to put to real life scenarios. The module has helped us to learn about database how it is created, functioned, managed, and executed in real world. The module has also helped us to learn more about many other topics along with the database system such as normalization of the data that is provided, create the entities in the database, their attributes, and the relation how an entity connects to another entity. Through this we can gain knowledge of how to normalize a data into simpler form using normalization, create an ERD of a database. Combining the knowledge gained from this module it allows us to prepare a successful working system.

7.2. Critical Assessment of Coursework

The Database coursework we are assigned with has asked us to create a database system for Mr. John for his established online marketplace for electronic goods which is named Gadget Emporium. In the given scenario through the given details, an initial ER diagram is prepared. Through the information provided by the ER diagram, normalization of the data is performed, where data is converted from unnormalized form to 1NF, then to 2NF and finally to 3NF which is the final form of the data. After the normalization process is done, a final ER diagram is prepared that displays the full flow of the database. Through the final ER diagram prepared, the tables of the entity and its attributes are entered in the SQL command line. After the tables are prepared data are inserted and tables are fully created. After the tables have been created, various queries are performed to display various information required by the user in the database through the SQL command line.

The coursework certainly tests our knowledge on the module but also provides valuable knowledge that we can use in the future projects. Various information can be gained that are helpful to understand the database module throughout this coursework. The coursework helps to understand a scenario and convert the scenario into data and the data to be converted into normalized simple form.

The coursework has brought many confusions and barriers throughout the process but also through these it has helped gain valuable knowledge which can be put on use for the future and created a confusing yet an interesting task to complete.

8. References

Becker, R., 2020. *Techopedia*. [Online]

Available at: <https://www.techopedia.com/definition/22561/third-normal-form-3nf>

[Accessed 10 1 2024].

Consulting, B., 2024. *Burleson Remote DBA*. [Online]

Available at: https://www.remote-dba.net/tp_Oracle_physical_design_unnormalized.htm

[Accessed 10 1 2024].

GeeksForGeeks, 2024. *GeeksForGeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/first-normal-form-1nf/>

[Accessed 10 1 2024].

Kozubek, A., 2020. *Vertabelo*. [Online]

Available at: <https://vertabelo.com/blog/normalization-1nf-2nf-3nf/>

[Accessed 10 1 2024].

Lucidchart, 2024. *Lucidchart*. [Online]

Available at: <https://www.lucidchart.com/pages/er-diagrams>

[Accessed 10 1 2024].

Peterson, R., 2023. *GURU99*. [Online]

Available at: <https://www.guru99.com/database-normalization.html>

[Accessed 10 1 2024].