LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

CC5051NI Databases

50% Individual Coursework

Autumn 2023

Student Name: Purnanand Mishra
London Met ID: 22085567
Assignment Submission Date: 2024/1/15
Word Count: 4843

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 content

Table of Contents

CHAPTER1-INTRODUCTION 6

1.1.INTRODUCTION 6

CHAPTER 2- INITIAL ERD... 7

1.1.ENTITIES AND ATTRIBUTES 7

1.2. IDENTIFICATION AND REPRESENTATION OF THE PRIMARY KEY AND FOREIGN KEY 7

IT SHOWS THE CONTACT OF VENDOR. 10

1.3.INITIAL ENTITY RELATIONSHIP DIAGRAM 11

CHAPTER 2 - NORMALIZATION 12

2.1.ASSUMPTION 12
2.2.NORMALIZATION 12
2.2.1.UNF (UN-NORMALIZED FORM) 13
2.2.2. 1NF (FIRST NORMAL FORM)...13
2.2.3. 2NF (SECOND NORMAL FORM) 14
2.2.4. 3NF (THIRD NORMAL FORM)...17
2.3. ER DIAGRAM OF NORMALIZED TABLES 22

CHAPTER 3-IMPLEMENTATION 23

3.1. TABLE CREATION 23
3.2. POPULATING DATABASE TABLE 28
3.3. FINAL TABLE 35

CHAPTER 4-INFORMATION AND TRANSACTION QUERIES 38

4.1.INFORMATION QUERIES 38
4.2. TRANSACTION QUERY 42
4.3.CREATION OF DUMPFILE 46
4.4. CREATE USER FILE 49

CHAPTER 5- CRITICAL EVALUATION 49

5.1.CRITICAL EVALUATION OF MODULE, ITS USAGE AND RELATION WITH ANOTHER SUBJECT 49
5.2. CRITICAL ASSESSMENT OF COURSEWORK.. 50

CHAPTER-6 CONCLUSION 50

CHAPTER-7 REFERENCES 50

BIBLIOGRAPHY.. 50

Figure 1:initial erd 11
Figure 2:Final erd 22
Figure 3:create customercategory 23
Figure 4:create table customer 24
Figure 5:create table ordertable.. 25
Figure 6:create vendor table 26
Figure 7:create productdetail table 26
Figure 8:create order_product table 27
Figure 9:create table customer_order_prodcut table 28
Figure 10:insert into customercatgory 29
Figure 11:insert into customer.. 30
Figure 12:insert into ordertable 32
Figure 13:insert into vendor 32

Figure 14:insert into productdetails 33
Figure 15:insert into order_product 34
Figure 16:insert into customer_order_product 35
Figure 17:select * customercategory 36
Figure 18:select * from customer 36
Figure 19:select * from ordertable 37
Figure 20:select * from vendor 37
Figure 21:select * from productdetails 37
Figure 22:select * from order_product 38
Figure 23:select * from customer_order_product 38
Figure 24:information query 1 39
Figure 25:information query 2 39
Figure 26:information query 3 40
Figure 27:information query 4 41
Figure 28:information query 5 41
Figure 29:information query 6 42
Figure 30:transaction query 1 43
Figure 31:transaction query 2 43
Figure 32:transaction query3 44
Figure 33:transaction query 4 45
Figure 34:transaction query 5 46
Figure 35:creation of dump file1 46
Figure 36:creation of spool file 48
Figure 37:create user file 49

Table 1:ordertable 8
Table 2:customer table 8
Table 3:create prodcudetails 9
Table 4:create vendor table 10

Chapter1-Introduction

1.1.introduction

This study covers the database design and implementation for the most popular delivery services provider Amazon in USA.it is one of the tech companies with quickest expansion in all over the world, and its goal is to create the best solution for people who do, not available area where he/she live and the most useful things for customer can use the amazon mobile app or website to monitor the status of their order. Real time updates on the position and anticipated arrival time are frequently included in tracking information.

Aims and objectives.

- Customer can save their time as well their money.
- The business real time track system for customer to track product.
- The business provides the fast and reliable delivery services to the customer.

Current business activities and operation and list of business rules

Amazon delivery services is an E-business that provide services to the customer product based on their demand. This is also giving many works to the people who need or want to do something because amazon is famous all over the world .it save money valuable time of customer to choose product among them. Customer use amazon g mobile app to track the product using amazon mobile we can track real time product. The delivery service is safe, fast and relivable for the customer. The most reliable for customer we can use variety method for payment customer have lots of payment. customer can use card, online pay.

Some business rule we must follow while doing coursework in this business listed below:

- Each customer can belong to a only particular customercategory.
- Each order unique orderid, orderdate, paymentoption, totalamount.
- Each prodcuct can only belong to a particular vendor.
- Each product has unique productid, productname, productcategory.
- Customer can multiply order, but each order is belonged with a customer.

  Assumption

1) Each Customer should provide the CustomerID, Firstname, Lastname, Address, Category, DiscountRate.

2)Customer category exit different customer like regular, VIP, staff.

3)Each category of customer is identified by primary key categoryid.

4)Customer place multiple order on different date and each order is recorded in order table with details such as orderid, customerid, orderdate, total amount, paymentoption.

5)Each product is identified by a unique key productid.

Chapter 2- Initial erd

1.1.Entities and attributes

To represent data in a database, two distinct concepts called entities and attributes are. utilized. As opposed to attributes, which relate to the specific properties attached to these. objects and entities refer to the precise physical things that are being kept in the database. (IBM, 2021)

1.2. Identification and representation of the primary key and foreign key

In a table, the main key is a column that unique identifies each row. To put it simply, a primary key (pk) is just unique different from all other keys. A foreign key is a column that create link between tow tables. Foreign keys are used to protect data integrity and facilitate communication between two distinct instances of entity. (Geeks for geek, 2023)

Attributes	Datatype	Constraints	Description
OrderID	INT	Primary key, unique	Store a unique id for Order.
Orderdate	DATE	NOT NULL	It represents the date of the Order.
Totalamount	INT	NOT NULL	It shows the total cost of an order.
Paymentoption	VARCHAR	NOT NULL	It describes the the method of payment used by customer when customer payment for product.

Table 1:ordertable

Attributes	Datatype	Constraints	Description
CustomerId	INT	Primary key, unique	Stores unique id of customer.
firstname	VARCHAR	NOT NULL	Store firstname of customer.
lastname	VARCHAR	NOT NULL	Store last name of customer.
Address	VARCHAR	NOT NULL	Store address of customer.
category	VARCHAR	NOT NULL	Store category of customer.
Discountrate	INT	NOT NULL	Store discount rate of customer.

Table 2:customer table

Table 3:create prodcudetails

Attributes	**Datatype**	**Constraint**	**Description**
Productid	INT	Primary key	Store unique id for product.
ProductName	VARCHAR	NOT NULL	It shows the name of product.
Product category	VARCHAR	NOT NULL	It shows the category of product.
Description	VARCHAR	NOT NULL	It shows the information of product.
Productprice	INT	NOT NULL	It shows the price of product
ProdcutQuantity	INT	NOT NULL	It shows the quantity of product in stock.
OrderQuantity	INT	NOT NULL	It shows the quantity of order product.

Attribute	**datatype**	**Constraint**	**Description**
Vendorid	INT	Primary key	Store unique id for vendor.
Vendor Name	VARCHAR	NOT NULL	it shows the name of vendor.
Vendoraddress	VARCHAR	NOT NULL	it shows the address of vendor.
Vendorcontact	INT	NOT NULL	It shows the contact of vendor.

Table 4:create vendor table

1.3.Initial entity relationship diagram

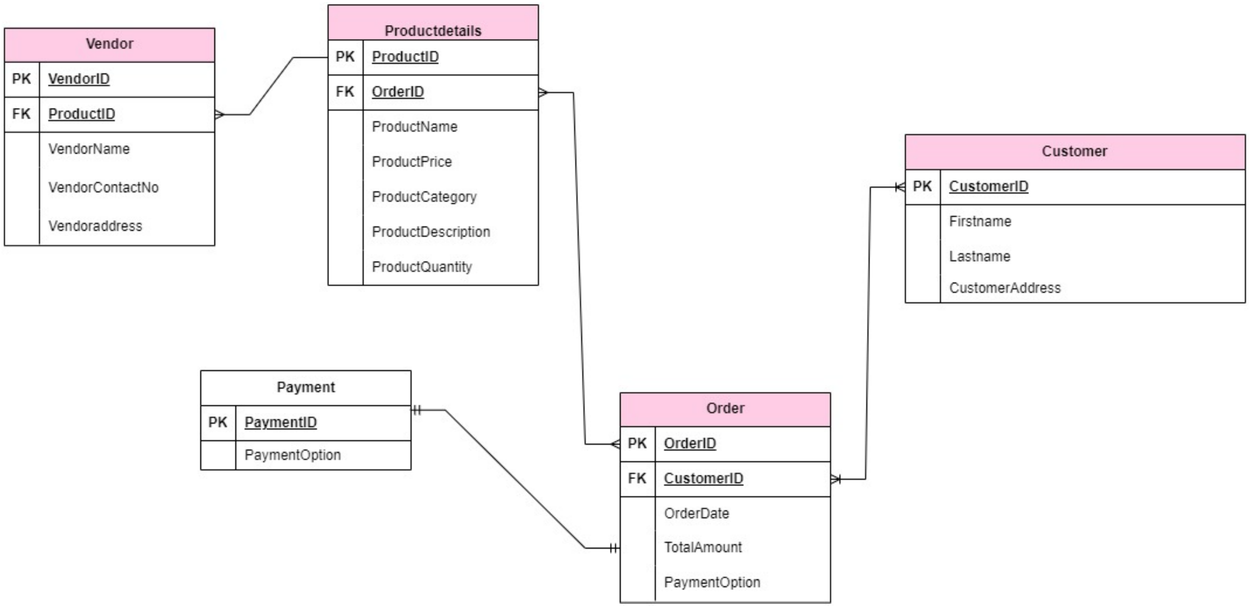


Figure 1:initial erd

Chapter 2 - Normalization

2.1.Assumption

The following assumption are used to support the ER Diagram Following Normalization is:

1) Every customer has a Customerid, Customer firstname, lastname, address.
2) Every Customer category has a Categoryid, Categoryname, Discount rate.
3) Every Order has a Orderid, orderdate, totalamount, paymentoption.
4) Every vendor has a vendorid, vendorname, vendoraddress, vendorcontact.
5) Every product has productid, ProductName, productcategory, Description, Productprice, productquantity.
6) Order link with product then orderquantity is associate with order_product table.

2.2.Normalization

The process of arranging data into tables so that queries executed on the database always return clear and intended result is known as database normalization. The foundation of relation database theory is this kind of normalization.it frequently lead to the formation of extra tables and may have the consequence of duplicating data within the database. It usually takes part in dividing an entity table into one or more tables and creating relationships between the tables. The objective is to isolate data so that addition, deletion, and modification of an attribute can be made in just one table and then propagated through the rest of the database via the defined relationship. The relation database model was first described in a paper by IBM researcher E.F.Codd in1970 ,and this is usually acknowledge as the source of the concept of database normalization this .An integral part of the relation technique was what Codd referred to as “ a normal form for database relation. (peterson, 2023)

2.2.1.UNF (Un-Normalized Form)

Scenario for UNF:

1) Each Customer should provide the CustomerID, Firstname, Lastname, Address, Category, DiscountRate.

2) A order contains with details such as orderId, customer, orderdate, total amount.

3) Each product is recorded in the product table with details like productid, ProductName, description, productcategory, product price, product quantity.

4) For each order customer can choose paymentoption. The information payment is recorded in row of order table.

5) The details of vendor who supplying each product are stored in the product table including vendroid, vendorname, vendoraddress, vendorcontact.

6) The order is placed ,the orderquantity in the product table is update.

7) Based on customer category, a discount rate is applied to the total amount in order table.

8) Customer can view her/his order history, by retrieving information or data from order and product table based on her/his customerid.

Repeating groups:

Customer (CustomerID (PK), Firstname, Lastname, Address, CategoryID, Categoryname, DiscountRate {OrderId(PK),Orderdate,Totalamount,paymentoption{ProductID(PK),Productname,Description,Productcategory,ProductPrice,Productquantity,Orderquantity,VendorID,Vendorname,Vendorcontact,Vendoraddress)}}

2.2.2. 1NF (First Normal Form)

First Normal Form (1NF), which is related to a single table in a relational database system, establishes the basic guidelines for database nomralisation. There are three fundamental steps to normalization, each building on the previous one. The initial normal form is the first of these. Each column in the table must be distinct and duplicated rows are not permitted and you cannot have identical column. (geeks for geek, 2023)

Scenario for 1NF:

1) Each Customer should provide the CustomerID, Firstname, Lastname, Address, Category, DiscountRate.

7) Customer place multiple order on different date and each order is recorded in order table with details such as orderid, customerid, orderdate, total amount, paymentoption.

8) Customer purchases various product and each product is identified by a unique key productid.

9) The details of each product name, description, category, productprice, orderquantity, product quantity, and last least vendor information is store in product detail table.

10) The product details table include foreign key customerid and orderid to link product with specific customers and orders.

Entities:

Customer-1 (CustomerID (PK), Firstname, Lastname, Address, CategoryID Categoryname, DiscountRate)

Order-1 (OrderID (PK), CustomerID (FK), Orderdate, Totalamount, paymentoptions)

ProductDetails-1 (ProductID (PK), CustomerID (FK), OrderID (FK), Productname, Description, Productcategory, price, Orderquantity, ProductQuantity, VendorID, Vendorname, VendorContact, Vendoraddress)

2.2.3. 2NF (Second Normal Form)

Scenario for 2NF:

1) Each Customer should provide the CustomerID, Firstname, Lastname, Address, Category, DiscountRate.

2)Customer place multiple order on different date and each order is recorded in order table with details such as orderid, customerid, orderdate, total amount, paymentoption.

3) Customer purchases various product and each product is identified by a unique key productid.

4)The link between customer and order is recorded in customer_order table.

5) Each row in customer_order table link customerid with orderid and this shows which customer place which order.

6)The link between customer and product is recorded in customer_prodcut table.

7)Each row in customer_product table link customer id with productid and this shows the customer has buy which product.

8)The link between order and product is recorded in the order_product table.

9)Each row in order _product table is linking order id with productid, and this shows which product are part of each order along with the quantity.

10)The link between customer, order, product is recorded in customer_order_prodcut table.

11)Each row in this table link customer id, orderid, productid which shows which product a customer has ordered in each order.

Showing partial dependency:
For Customer:

1) Customer ID determines the Firstname, Lastname, Address, category, DiscountRate.

2) There is no Composite primary key in Customer, so it does not determine any attributes of Customer.
So,

Customer ID => Firstname, Lastname, Address, Category, DiscountRate.

For Order:

1)Order ID determines the Orderdate, Totalamount, paymentoption.

2)Composite primary key Customer ID, Order ID doesn't determine any of the Order attributes.

So,

OrderID=>Orderdate, Totalamount, Paymentoption

OrderID, CustomerID => Nothing

For ProdcutDetails:

1)Product ID determines Productname, Productcategory, Description, price, Stockquantity, VendorID, Vendorname, Vendorcontact, Vendor Address.

So,

Product ID => Productname, Productcategory, Description, Price, Stockquantity, Vendorname, Vendor Address, Vendor Contact.

2)Composite primary key Customer ID, ProdcutID doesn't determines any attribute of ProductDeatails Table.

So,

Customer ID, ProductID =>Nothing

3)Composite Primary key OrderID, ProductID determine the quantity for ProductDetail

So,
OrderID, ProductID => OrderQuantity.

4)Composite primary key Order ID, Product ID, CustomerID dosn, t determines any attribute of Product details.

So,
OrderID, Productid, Customer ID => Nothing

Entities:

Customer-2 (CustomerID (PK), Firstname, Lastname, Address, CategoryID, Categoryname, DiscountRate)

Order-2 (OrderID (PK), Orderdate, Totalamount, PaymentOptions)

ProductDetails-2 (ProductID (PK), Productname, Productcategory, ProdcutPrice, Description, ProductQuantity, VendorID, Vendorname, Vendorcontact, VendorAddress)

Customer_Order-2 (CustomerID (FK), OrderID (FK))

Customer_Product-2 (CustomerID (FK), ProdcutID (FK))

Order_Product-2 (OrderID (FK), ProductID (FK), OrderQuantity)

Customer_Product_Order -2 (CustomerID (FK), OrderID (FK), ProductID (FK))

2.2.4. 3NF (Third Normal Form)

Scenario for 3NF:

1) Each Customer should provide the CustomerID, Firstname, Lastname, Address, Category, DiscountRate.

2)Customer category exit different customer like regular, VIP, staff.

3)Each category of customer is identified by primary key categoryid.

3) Customer place multiple order on different date and each order is recorded in order table with details such as orderid, customerid, orderdate, total amount, paymentoption.

4) Each product is identified by a unique key productid.

5) Each product has product name, product category, description, productprice, product quantity, are stored in productdetail table.

6) The vendor has vendor id, vendorname, vedorconatact, vendoraddres.

7) Each vendor is identified by a unique key vendor id.

8) The link between customer and order is recorded in customer_order table.

9) Each row in customer_order table link customerid with orderid and this shows which customer place which order.

10) The link between customer and product is recorded in customer_prodcut table.

11) Each row in customer_product table link customer id with productid and this shows the customer has buy which product.

12) The link between order and product is recorded in the order_product table.

13) Each row in order _product table is linking order id with productid, and this shows which product are part of each order along with the quantity.

14) The link between customer, order, product is recorded in customer_order_prodcut table.

15) Each row in this table link customer id, orderid, product id which shows which product a customer has ordered in each order.

For Customer:

1) CustomerID determine the Firstname, Lastname, Address and CategoryID determine the Categoryname, DiscountRate

   So,
   CustomerID => CategoryID => DiscountRate

Resolving this:

CustomerID => CustomerCategory
CustomerCategory => Discountrate

CustomerID => Firstname, Lastname, Address,

CategoryID => Categoryname, Discountrate

Final:

Customer -3 (CustomerID (PK), CategoryID (FK), Firstname, Lastname, Address)

CustomerCategory -3 (CategoryID (PK), Categoryname, Discountrate)

For Order:

1)The table should be in 2NF, and there should be no transitive dependencies and there is no non-key attribute should depend on another non key attribute .so the table is already in 3NF.

Finally,

Order-3 (OrderID (PK), Orderdate, Totalamount, PaymentOptions)

For ProductDetails:

1)ProductID determine Productname, Productcategory, Description, Productprice, Productquantity and VendorID determines Vendorname, Vendorcontact, Vendoraddress.

So,
ProductID =>Productname, Productcategory, Description, Productprice, ProdcutQuantity, VendorID, Vendorname, Vendorcontact, Vendoraddress.

Resolving:

ProductID => Productname, Productcategory, Description, Productprice, ProductQuantity

VendorID => Vendorname, Vendoraddress, Vendorcontact

So,

Final:

Productdetails-3(ProductID (PK), VendorID (FK), Productname, Productcategory, Description, Productprice, ProductQuantity

Vendor-3 (VendorID (PK), Vendorname, Vendoraddress, Vendorcontact

For Customer_Order -3

1) The table should be in 2NF, and there should be no transitive dependencies and there is no non-key attribute should depend on another non key attribute .so the table is already in 3NF.

So finally,

Customer _Order-3 (CustomerID (FK), OrderID (FK))

For Customer_Prodcut

1) The table should be in 2NF, and there should be no transitive dependencies and there is no non-key attribute should depend on another non key attribute .so the table is already in 3NF.

So finally,

Customer_product- 3 (CustomerID (FK), ProductID (FK))

For Order_Product

1) The table should be in 2NF, and there should be no transitive dependencies and there is no non-key attribute should depend on another non key attribute .so the table is already in 3NF.

So finally,

Order_Product-3 (OrderID (FK), ProductID (FK))

For Customer_Order_Product

1) The table should be in 2NF, and there should be no transitive dependencies and there is no non-key attribute should depend on another non key attribute .so the table is already in 3NF.

So finally,

Customer_Order_Product-3 (CustomerID (FK), OrderID (FK), ProductID (FK)

Entities:

Customer -3 (CustomerID (PK), CategoryID (FK), Firstname, Lastname, Address)

CustomerCategory -3 (CategoryID (PK), Categoryname, Discountrate)

Order-3 (OrderID (PK), Orderdate, Totalamount, PaymentOptions)

Productdetails-3(ProductID (PK), VendorID (FK), Productname, Productcategory, Description, Productprice, ProductQuantity)

Vendor-3 (VendorID (PK), Vendorname, Vendoraddress, Vendorcontact)

Customer _Order-3 (CustomerID (FK), OrderID (FK)) => x

Customer_product- 3 (CustomerID (FK), ProductID (FK)) =x

Order_Product-3 (OrderID (FK), ProductID (FK))

Customer_Order_Product-3 (CustomerID (FK), OrderID (FK), ProductID (FK)

Out the 3NF tables formed customer_order_3 and customer_product-3 do not have any other attributes except foreign key, and customer_order_product-3 already the main bridge entity so,these 2 tables are delete.

Final Entities:

Customer -3 (CustomerID (PK), CategoryID (FK), Firstname, Lastname, Address)

CustomerCategory -3 (CategoryID (PK), Categoryname, Discountrate)

Order-3 (OrderID (PK), Orderdate, Totalamount, PaymentOptions)

Productdetails-3(ProductID (PK), VendorID (FK), Productname, Productcategory, Description, Productprice, ProductQuantity)

Vendor-3 (VendorID (PK), Vendorname, Vendoraddress, Vendorcontact)

Order_Product-3 (OrderID (FK), ProductID (FK))

Customer_Order_Product-3 (CustomerID (FK), OrderID (FK), ProductID (FK)

2.3. ER diagram of Normalized tables

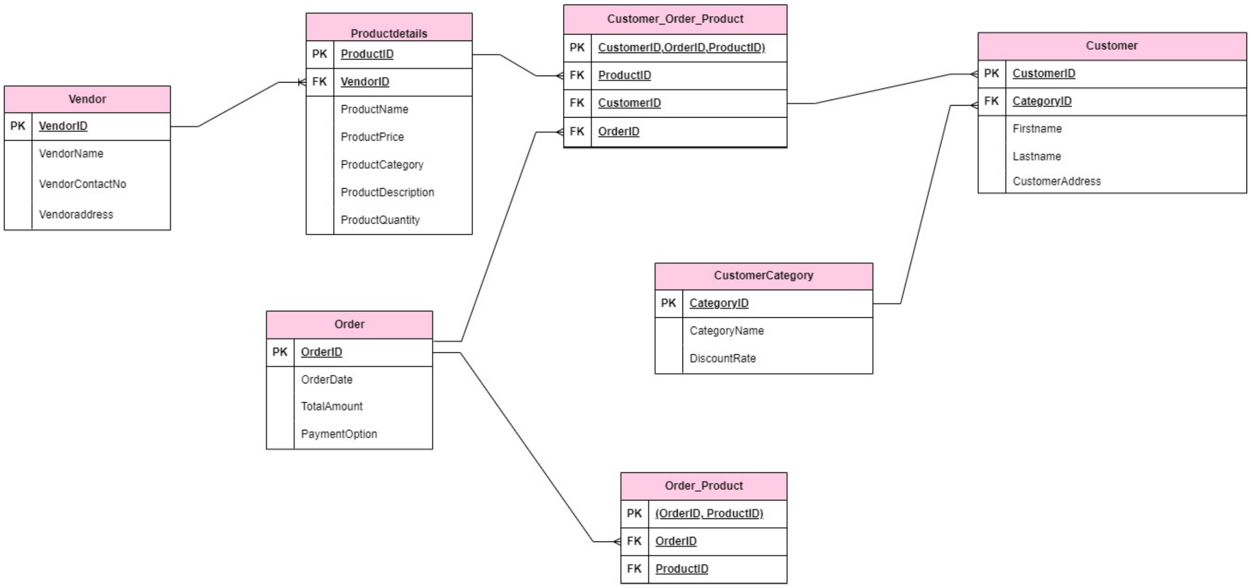


Figure 2:Final erd

Chapter 3-Implementation

3.1. Table creation

To create a table inside database “Create table” command is used which is one of the most complicated statement in sql .To create a table following things must be implemented :name of the table ,name of the field and definition of every field .The ALTER Table statement is used to add, delete ,modify , column or constraint is existing table (need reference w3 school).A primary key is a column or set of columns that uniquely identify each row in table .A foreign key is a field or collection of field in one table which refers to the primary key in another table (Software AG, 2023)

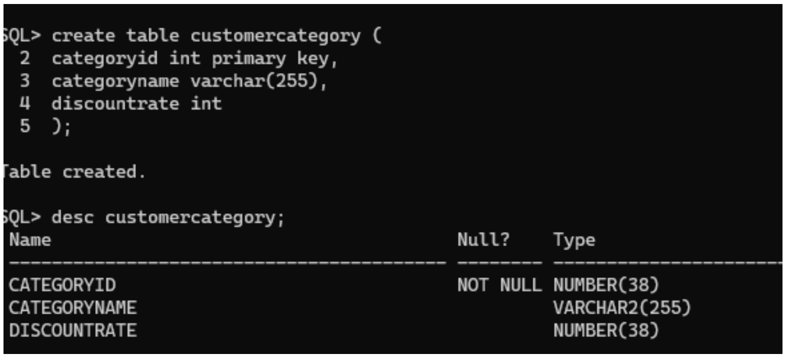
Creating table customercatogory:

Create table customercategory (

Categoryid int primary key,

Ctaegoryname varchar (255),

Discountrate int
);



```
SQL> create table customercategory (
  2  categoryid int primary key,
  3  categoryname varchar(255),
  4  discountrate int
  5  );

Table created.

SQL> desc customercategory;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 CATEGORYID                                NOT NULL NUMBER(38)
 CATEGORYNAME                                       VARCHAR2(255)
 DISCOUNTRATE                                       NUMBER(38)
```

Figure 3:create customercategory

Creating table customer:

Create table customer (

Customerid int primary key,

Firstname varchar (255),

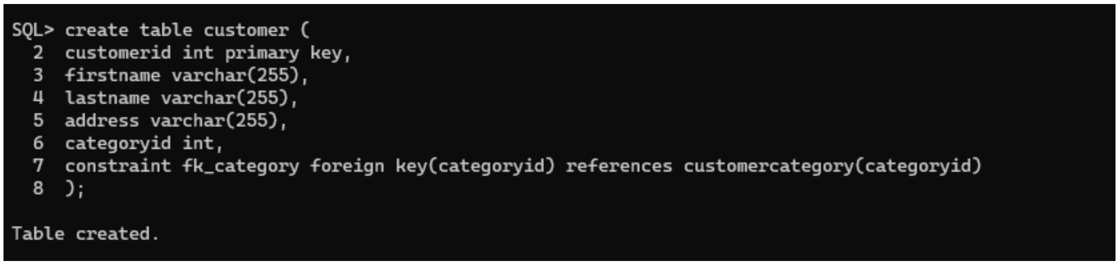
Lastname varchar (255),

Address varchar (255),

Categoryid int,

Constraint fk_category foreign key(categoryid) references

customercategory(categoryid)
);



```
SQL> create table customer (
customerid int primary key,
firstname varchar(255),
lastname varchar(255),
address varchar(255),
categoryid int,
constraint fk_category foreign key(categoryid) references customercategory(categoryid)
);

Table created.
```

Figure 4:create table customer

Creating table ordertable:

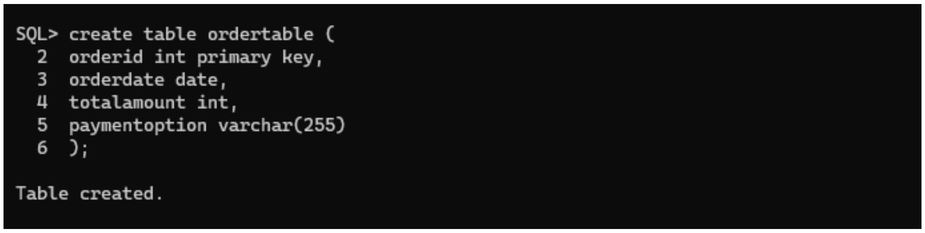
Create table ordertable (

Orderid int primary key,
Orderdate date,

Totalamount int ,

Paymentoption varchar(255)

);



```
SQL> create table ordertable (
  orderid int primary key,
  orderdate date,
  totalamount int,
  paymentoption varchar(255)
  );

Table created.
```

Figure 5:create table ordertable

Creating table vendor:

Create table vendor (

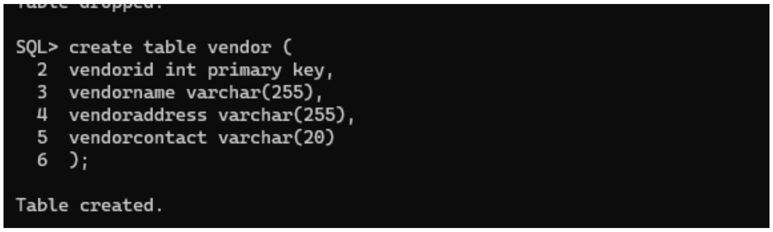
Vendorid int primary key,

Vendorname varchar(255),

Vendoraddress varchar(255),

Vendorcontact varchar(20),

);



```
Table dropped.

SQL> create table vendor (
  vendorid int primary key,
  vendorname varchar(255),
  vendoraddress varchar(255),
  vendorcontact varchar(20)
  );

Table created.
```

Figure 6:create vendor table

Create table productdetails:

Create table productdetails (

Productid int primary key,

Vendorid int,

Productname varchar(255),

Productcategory varchar(255),

Description varchar(255),

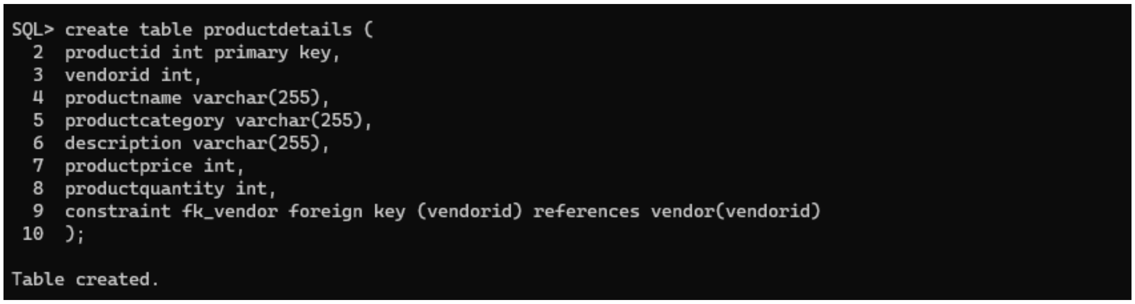
Productquantity int ,

Productprice int,

Constraint fk_vendor foreign key(vendorid) references

vendor(vendorid)

);



```
SQL> create table productdetails (
productid int primary key,
vendorid int,
productname varchar(255),
productcategory varchar(255),
description varchar(255),
productprice int,
productquantity int,
constraint fk_vendor foreign key (vendorid) references vendor(vendorid)
);

Table created.
```

Figure 7:create productdetail table

Creating table order_product:

Create table order_product (

Orderid int,

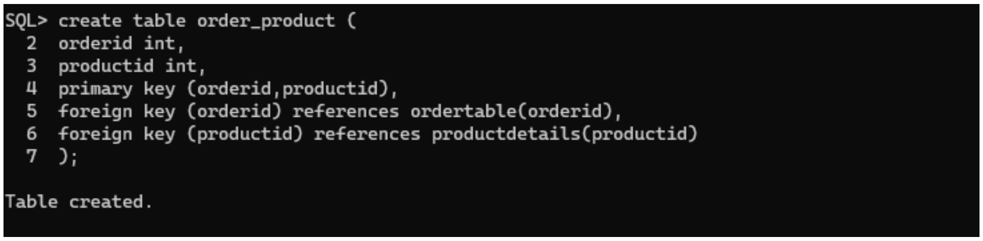
Productid int,

Primary key(orderid,productid),

Foreign key (orderid) references ordertable(orderid),

Foreign key(productid) references productdetails(productid)

);



```
SQL> create table order_product (
  2  orderid int,
  3  productid int,
  4  primary key (orderid,productid),
  5  foreign key (orderid) references ordertable(orderid),
  6  foreign key (productid) references productdetails(productid)
  7  );

Table created.
```

Figure 8:create order_product table

Creating customer_order_product table:

Create table customer_order_product (

Customerid int,

Orderid int,

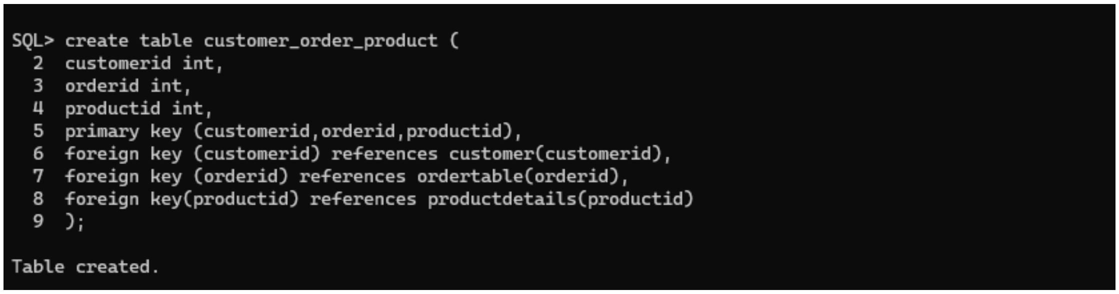
Productid int,
Primary key(customerid,orderid,productid),

Foreign key (customerid) references customer(customerid),

Foreign key (orderid) references ordertable(orderid),

Foreign key (productid) references productdetails(productid)

);



```
SQL> create table customer_order_product (
customerid int,
orderid int,
productid int,
primary key (customerid,orderid,productid),
foreign key (customerid) references customer(customerid),
foreign key (orderid) references ordertable(orderid),
foreign key(productid) references productdetails(productid)
);

Table created.
```

Figure 9:create table customer_order_prodcut table

3.2. Populating database table

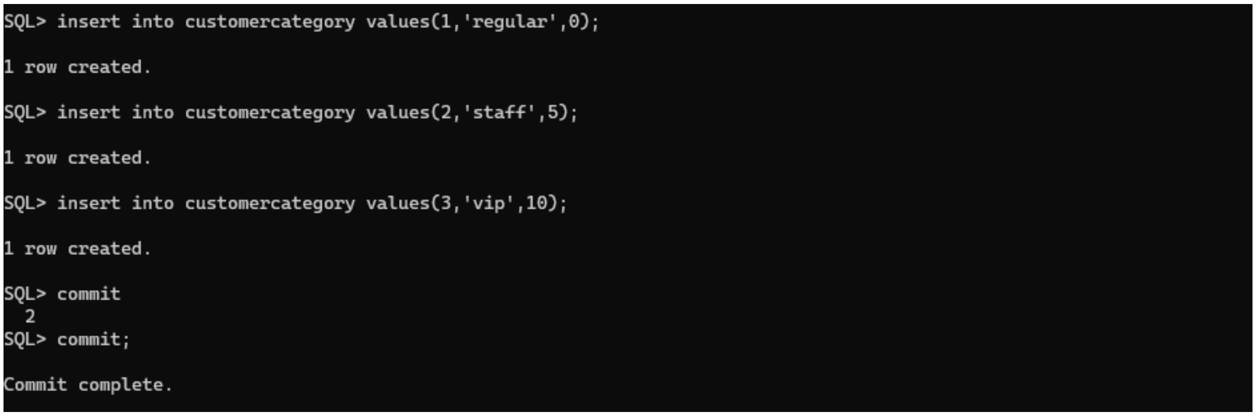
‘INSERT INTO; command is used to insert date into the tables. ‘COMMIT’ command is used to to save the inserted date permanently.by using ‘INSERT INTO’ and ‘COMMIT’ command in Oracle SQL Plus all the date will be stored and saved and secured.

Insert values in customercategory:

insert into customercategory(1,’regular’,0);

insert into customercategory(2,’staff’,5);

insert into customercategory(3,’vip’,10);



```
SQL> insert into customercategory values(1,'regular',0);

1 row created.

SQL> insert into customercategory values(2,'staff',5);

1 row created.

SQL> insert into customercategory values(3,'vip',10);

1 row created.

SQL> commit
  2
SQL> commit;

Commit complete.
```

Figure 10:insert into customercatgory

Insert values in customer:

insert into customer values(1,’sandesh’,’pokhreal’,123 biratnagar’,2);

insert into customer values(2,’anand’,’mishra’,’45 jnk’,1);

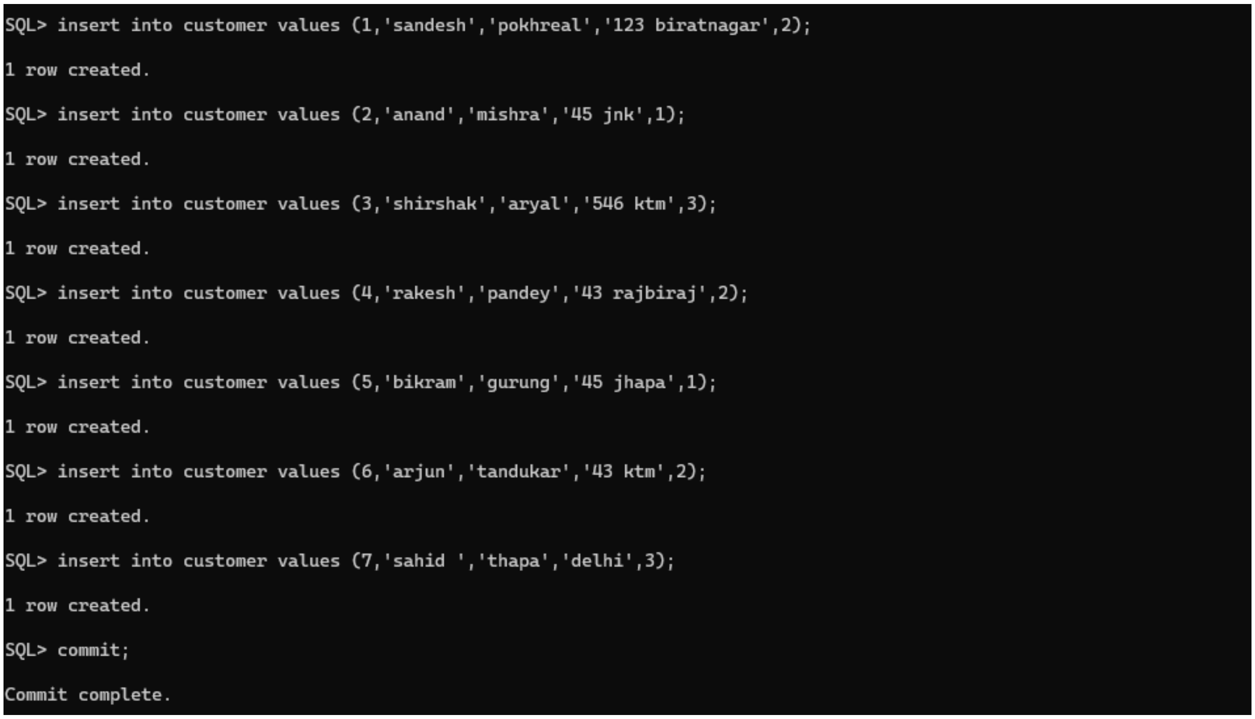
insert into customer values(3,’shirshak’,’’aryal’,’546 ktm’,3);

insert into customer values(4,’rakesh’,’pandey’,’43 rajbiraj’,2);

insert into customer values(5,’bikram’,’gurung’,’45 jhapa’,1);

insert into customer values(6,’arjun’,’tandukar’,’123 ktm’,2);

insert into customer values(7,’sahid’,’thapa’,’delhi’,3);

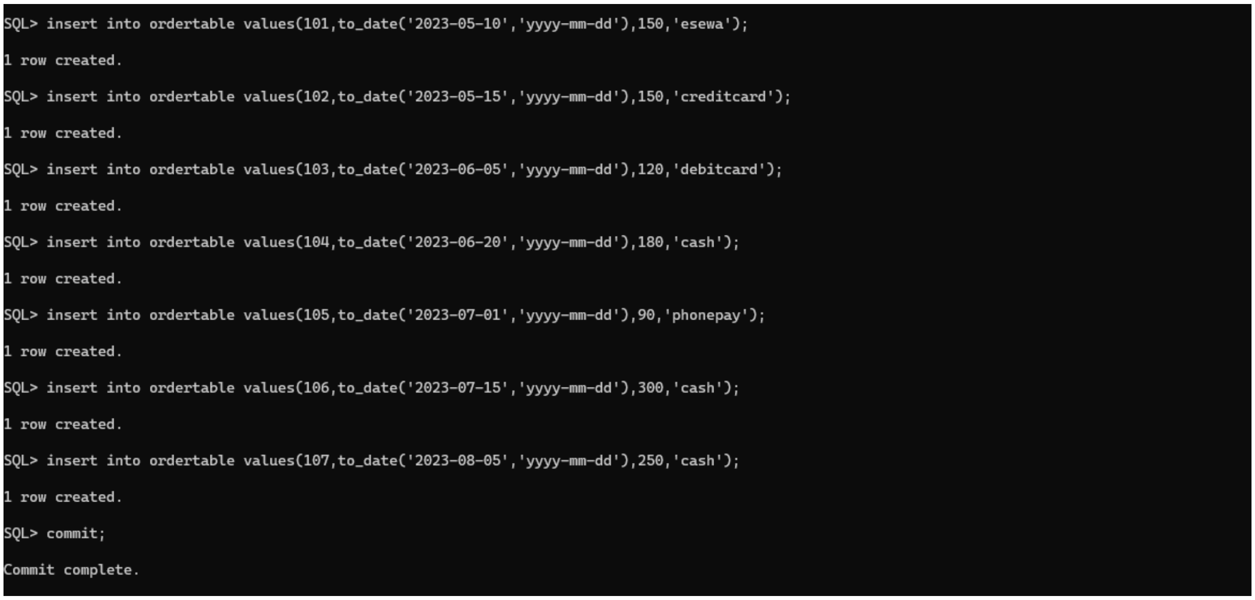


```
SQL> insert into customer values (1,'sandesh','pokhreal','123 biratnagar',2);

1 row created.

SQL> insert into customer values (2,'anand','mishra','45 jnk',1);

1 row created.

SQL> insert into customer values (3,'shirshak','aryal','546 ktm',3);

1 row created.

SQL> insert into customer values (4,'rakesh','pandey','43 rajbiraj',2);

1 row created.

SQL> insert into customer values (5,'bikram','gurung','45 jhapa',1);

1 row created.

SQL> insert into customer values (6,'arjun','tandukar','43 ktm',2);

1 row created.

SQL> insert into customer values (7,'sahid ','thapa','delhi',3);

1 row created.

SQL> commit;

Commit complete.
```

Figure 11:insert into customer

Insert values in ordertable:

```
 Insert into ordertable values(101,to_date(‘2023-05-10’,’yyyy-mm-dd’);

Insert into ordertable values(102,to_date(‘2023-05-15’,’yyyy-mm-dd’);

Insert into ordertable values(103,to_date(‘2023-06-05’,’yyyy-mm-dd’);

Insert into ordertable values(104,to_date(‘2023-06-20’,’yyyy-mm-dd’);

Insert into ordertable values(105,to_date(‘2023-07-01’,’yyyy-mm-dd’);

Insert into ordertable values(106,to_date(‘2023-07-15’,’yyyy-mm-dd’);

Insert into ordertable values(107,to_date(‘2023-08-05’,’yyyy-mm-dd’);
```



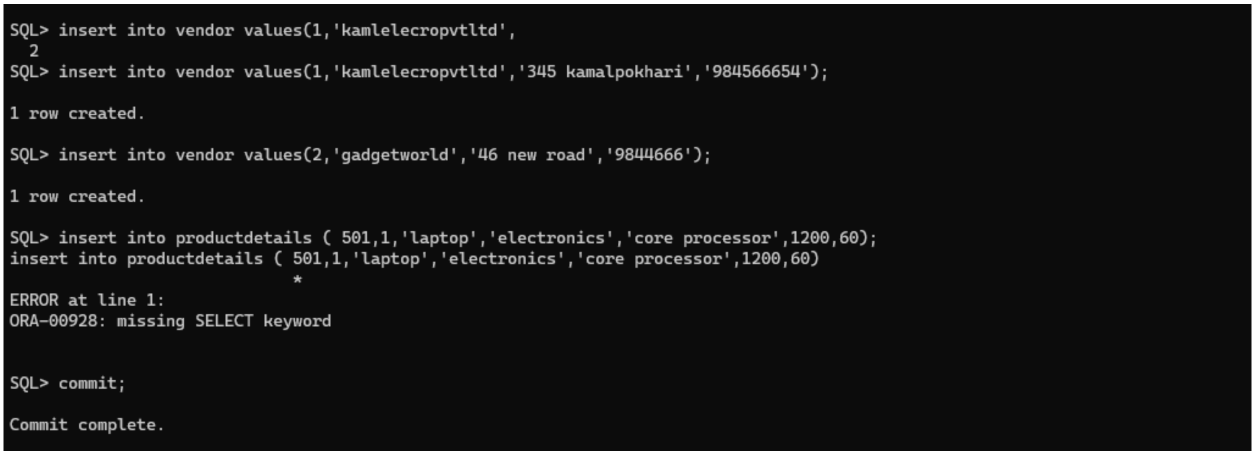
```
SQL> insert into ordertable values(101,to_date('2023-05-10','yyyy-mm-dd'),150,'esewa');

1 row created.

SQL> insert into ordertable values(102,to_date('2023-05-15','yyyy-mm-dd'),150,'creditcard');

1 row created.

SQL> insert into ordertable values(103,to_date('2023-06-05','yyyy-mm-dd'),120,'debitcard');

1 row created.

SQL> insert into ordertable values(104,to_date('2023-06-20','yyyy-mm-dd'),180,'cash');

1 row created.

SQL> insert into ordertable values(105,to_date('2023-07-01','yyyy-mm-dd'),90,'phonepay');

1 row created.

SQL> insert into ordertable values(106,to_date('2023-07-15','yyyy-mm-dd'),300,'cash');

1 row created.

SQL> insert into ordertable values(107,to_date('2023-08-05','yyyy-mm-dd'),250,'cash');

1 row created.

SQL> commit;

Commit complete.
```

Figure 12:insert into ordertable

Insert values in vendor:

insert into vendor values (1,’kamelecroppvtltd’,’345 kamalpokhari’,984566654’);

insert into vendor values (2,’gadgetworld’,’46 new road’,9844666’);



```
SQL> insert into vendor values(1,'kamlelecropvtltd',
  2
SQL> insert into vendor values(1,'kamlelecropvtltd','345 kamalpokhari','984566654');

1 row created.

SQL> insert into vendor values(2,'gadgetworld','46 new road','9844666');

1 row created.

SQL> insert into productdetails ( 501,1,'laptop','electronics','core processor',1200,60);
insert into productdetails ( 501,1,'laptop','electronics','core processor',1200,60)
                            *
ERROR at line 1:
ORA-00928: missing SELECT keyword


SQL> commit;

Commit complete.
```

Figure 13:insert into vendor

Insert values in productdetails:

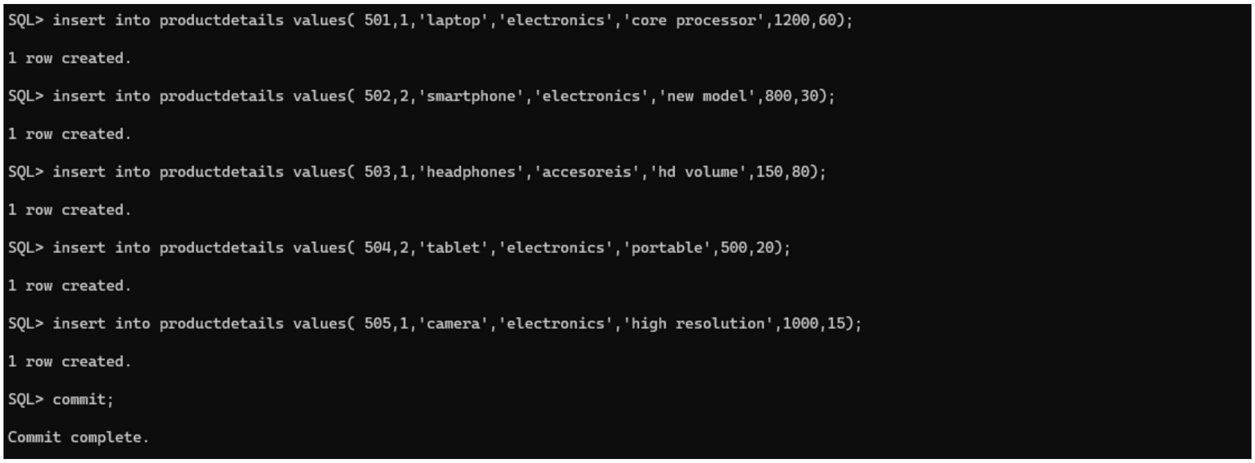
Insert into productdetails values(501,1,’laptop’,’electronics’,’core proccessor’,1200,60);

Insert into productdetails values(502,2,’smartphone’,’electronics’,’new model’,800,30);

Insert into productdetails values(503,1,’headphone’,’accesories’,’hd volume’,150,80);

Insert into productdetails values(504,2,’tablet’,’electronics’,’portable’,500,20);

Insert into productdetails values(505,1,’camera’,’electronics’,’high resolution ’,1000,15);



```
SQL> insert into productdetails values( 501,1,'laptop','electronics','core processor',1200,60);

1 row created.

SQL> insert into productdetails values( 502,2,'smartphone','electronics','new model',800,30);

1 row created.

SQL> insert into productdetails values( 503,1,'headphones','accesoreis','hd volume',150,80);

1 row created.

SQL> insert into productdetails values( 504,2,'tablet','electronics','portable',500,20);

1 row created.

SQL> insert into productdetails values( 505,1,'camera','electronics','high resolution',1000,15);

1 row created.

SQL> commit;

Commit complete.
```

Figure 14:insert into productdetails

Insert values in order_product:

Insert into order_product values(101,501);

Insert into order_product values(102,502);

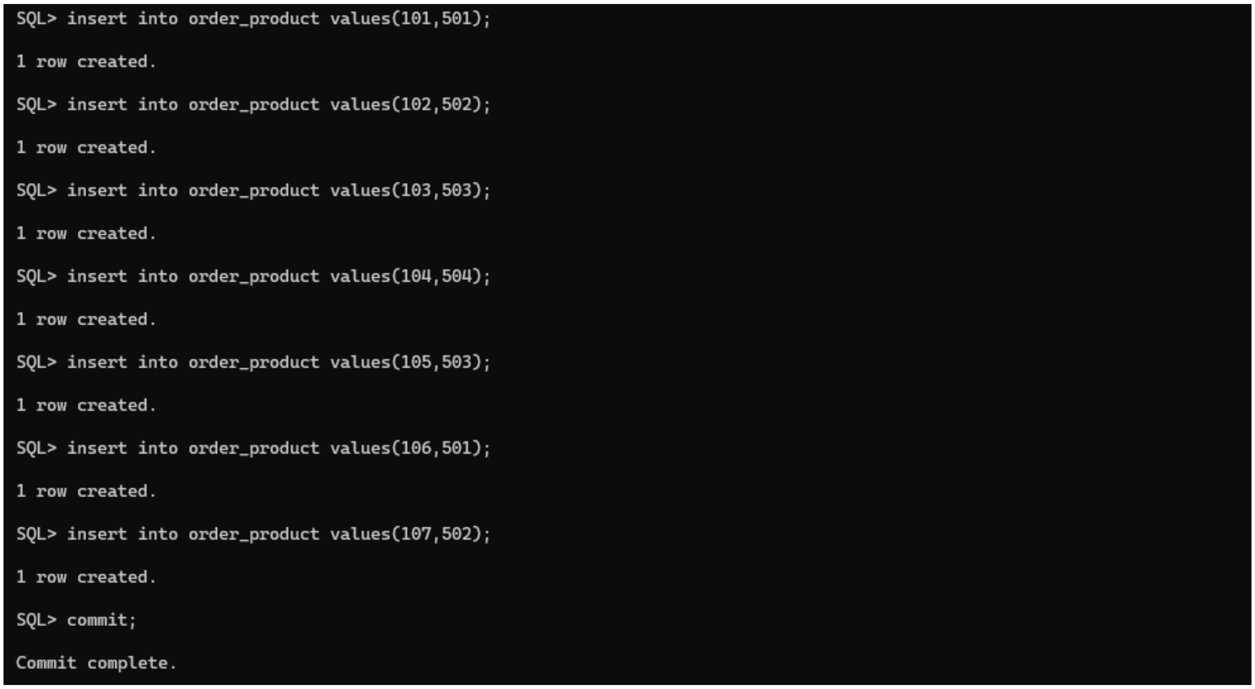
Insert into order_product values(103,503);

Insert into order_product values(104,504);

Insert into order_product values(105,503);

Insert into order_product values(106,501);

Insert into order_product values(107,502);



```
SQL> insert into order_product values(101,501);

1 row created.

SQL> insert into order_product values(102,502);

1 row created.

SQL> insert into order_product values(103,503);

1 row created.

SQL> insert into order_product values(104,504);

1 row created.

SQL> insert into order_product values(105,503);

1 row created.

SQL> insert into order_product values(106,501);

1 row created.

SQL> insert into order_product values(107,502);

1 row created.

SQL> commit;

Commit complete.
```

Figure 15:insert into order_product

Insert values in customer_order_product:

insert into customer_order_product values(1,101,501);

insert into customer_order_product values(2,102,502);

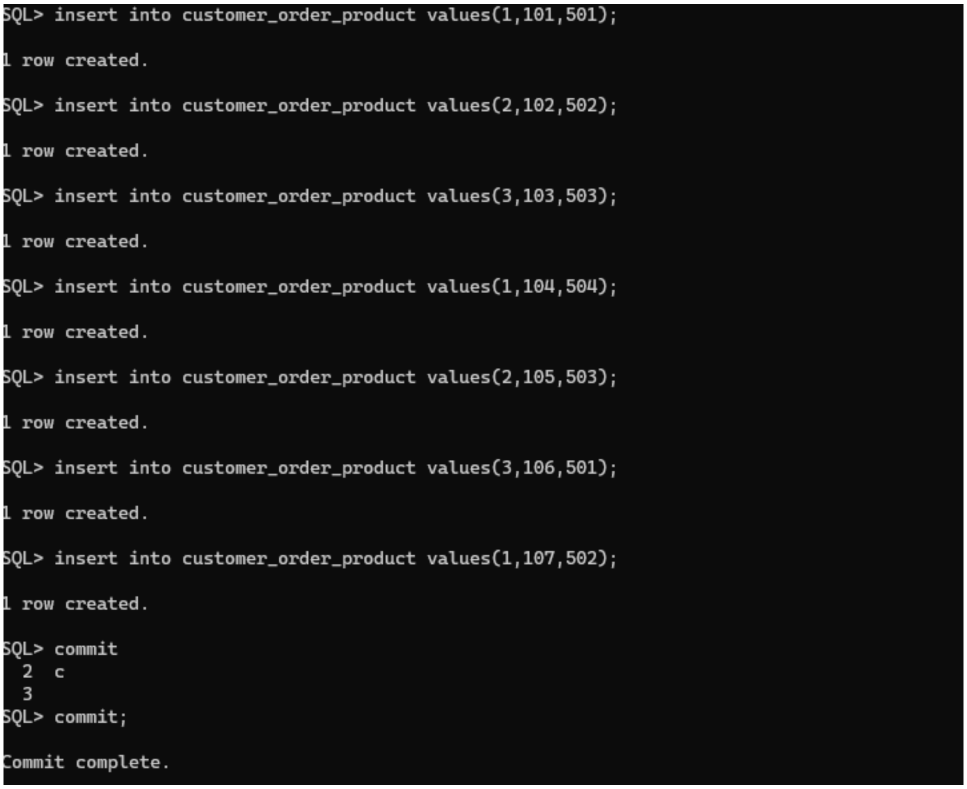
insert into customer_order_product values(3,103,503);

insert into customer_order_product values(1,104,504);

insert into customer_order_product values(2,105,503);

insert into customer_order_product values(3,106,501);

insert into customer_order_product values(1,107,502);



```
SQL> insert into customer_order_product values(1,101,501);

1 row created.

SQL> insert into customer_order_product values(2,102,502);

1 row created.

SQL> insert into customer_order_product values(3,103,503);

1 row created.

SQL> insert into customer_order_product values(1,104,504);

1 row created.

SQL> insert into customer_order_product values(2,105,503);

1 row created.

SQL> insert into customer_order_product values(3,106,501);

1 row created.

SQL> insert into customer_order_product values(1,107,502);

1 row created.

SQL> commit
  2  c
  3
SQL> commit;

Commit complete.
```

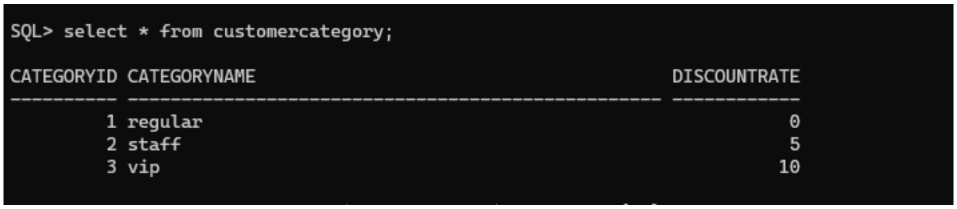
Figure 16:insert into customer_order_product

3.3. Final table

“select” command is implemented to provide the inserted date in orace sql.

Customercategory Table:

Select * from customercategory;

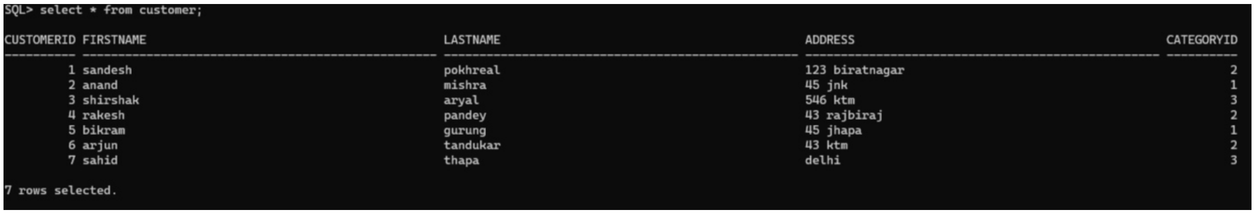


```
SQL> select * from customercategory;

CATEGORYID CATEGORYNAME                                       DISCOUNTRATE
---------- -------------------------------------------------- ------------
         1 regular                                                       0
         2 staff                                                         5
         3 vip                                                          10
```

*Figure 17:select * customercategory*

Customer Table:

Select * from customer:

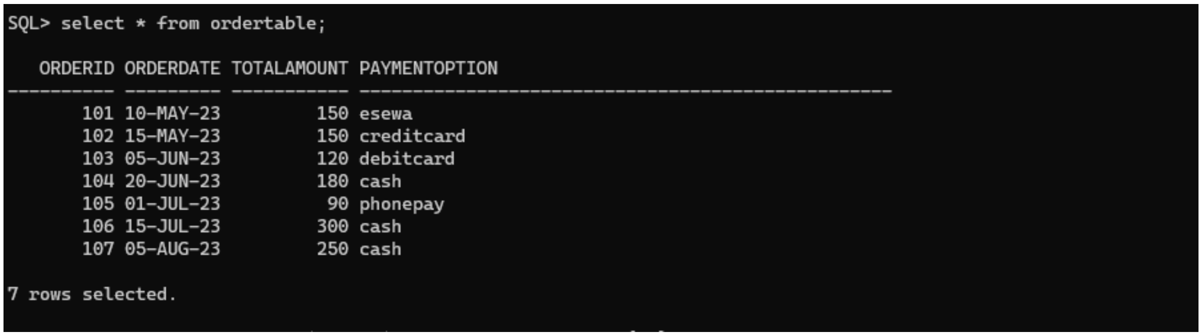


```
SQL> select * from customer;

CUSTOMERID FIRSTNAME                                          LASTNAME                                           ADDRESS                                            CATEGORYID
---------- -------------------------------------------------- -------------------------------------------------- -------------------------------------------------- ----------
         1 sandesh                                            pokhreal                                           123 biratnagar                                              2
         2 anand                                              mishra                                             45 jnk                                                      1
         3 shirshak                                           aryal                                              546 ktm                                                     3
         4 rakesh                                             pandey                                             43 rajbiraj                                                 2
         5 bikram                                             gurung                                             45 jhapa                                                    1
         6 arjun                                              tandukar                                           43 ktm                                                      2
         7 sahid                                              thapa                                              delhi                                                       3

7 rows selected.
```

*Figure 18:select * from customer*

Ordertable Table:

Select * from ordertable;

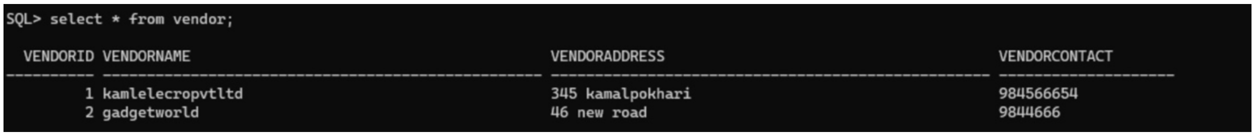


```
SQL> select * from ordertable;

   ORDERID ORDERDATE TOTALAMOUNT PAYMENTOPTION
---------- --------- ----------- --------------------------------------------------
       101 10-MAY-23         150 esewa
       102 15-MAY-23         150 creditcard
       103 05-JUN-23         120 debitcard
       104 20-JUN-23         180 cash
       105 01-JUL-23          90 phonepay
       106 15-JUL-23         300 cash
       107 05-AUG-23         250 cash

7 rows selected.
```

*Figure 19:select * from ordertable*

Vendor Table:

Select * from vendor;

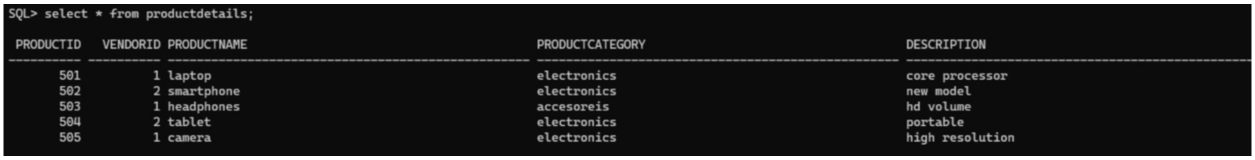


```
SQL> select * from vendor;

  VENDORID VENDORNAME                                         VENDORADDRESS                                      VENDORCONTACT
---------- -------------------------------------------------- -------------------------------------------------- --------------------
         1 kamlelecropvtltd                                   345 kamalpokhari                                   984566654
         2 gadgetworld                                        46 new road                                        9844666
```

*Figure 20:select * from vendor*

Productdetails table:

Select * from productdetails;

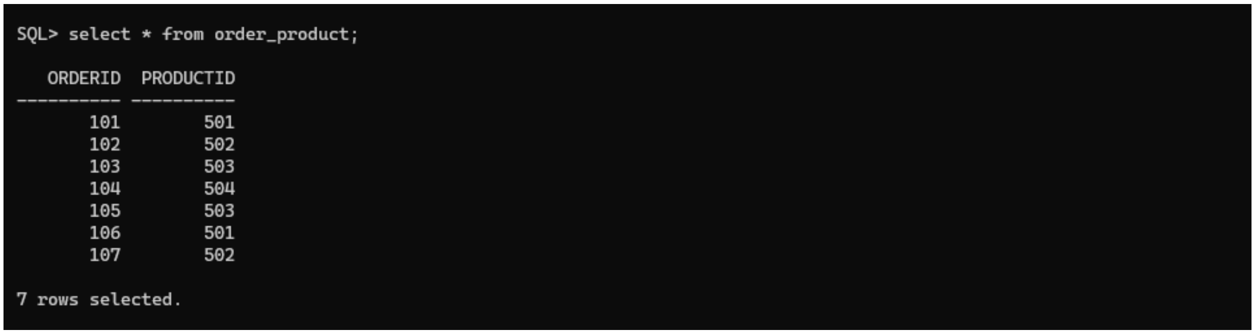


```
SQL> select * from productdetails;

 PRODUCTID   VENDORID PRODUCTNAME                                        PRODUCTCATEGORY                                    DESCRIPTION
---------- ---------- -------------------------------------------------- -------------------------------------------------- --------------------------------------------------
       501          1 laptop                                             electronics                                        core processor
       502          2 smartphone                                         electronics                                        new model
       503          1 headphones                                         accesoreis                                         hd volume
       504          2 tablet                                             electronics                                        portable
       505          1 camera                                             electronics                                        high resolution
```

*Figure 21:select * from productdetails*

Order_product table:

Select * from order_product;

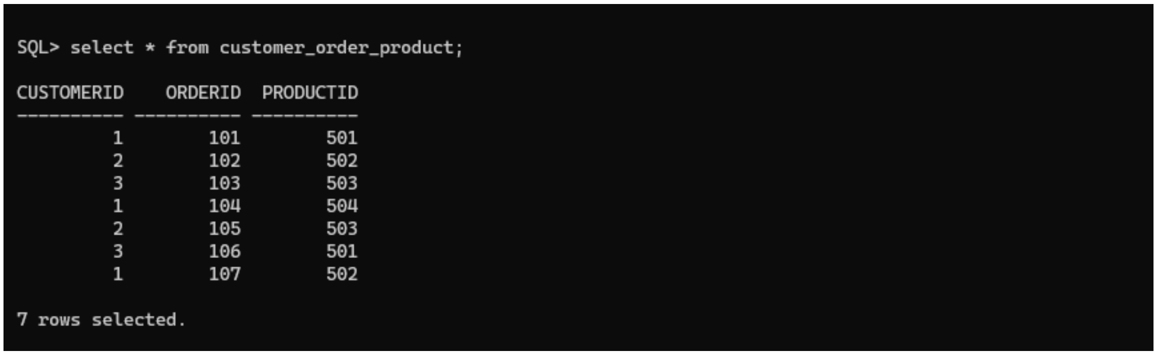


```
SQL> select * from order_product;

   ORDERID  PRODUCTID
---------- ----------
       101        501
       102        502
       103        503
       104        504
       105        503
       106        501
       107        502

7 rows selected.
```

*Figure 22:select * from order_product*

Customer_order_product Table:

Select * from customer_order_product;



```
SQL> select * from customer_order_product;

CUSTOMERID    ORDERID  PRODUCTID
---------- ---------- ----------
         1        101        501
         2        102        502
         3        103        503
         1        104        504
         2        105        503
         3        106        501
         1        107        502

7 rows selected.
```

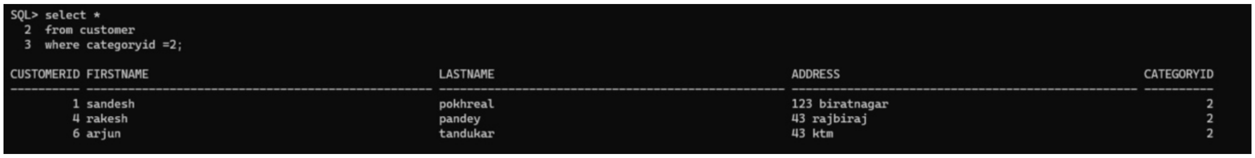
*Figure 23:select * from customer_order_product*

Chapter 4-Information and Transaction Queries

4.1.Information Queries

1.) List all the customer that are also staff of the company.

Select *
From customer
Where categoryid =2;



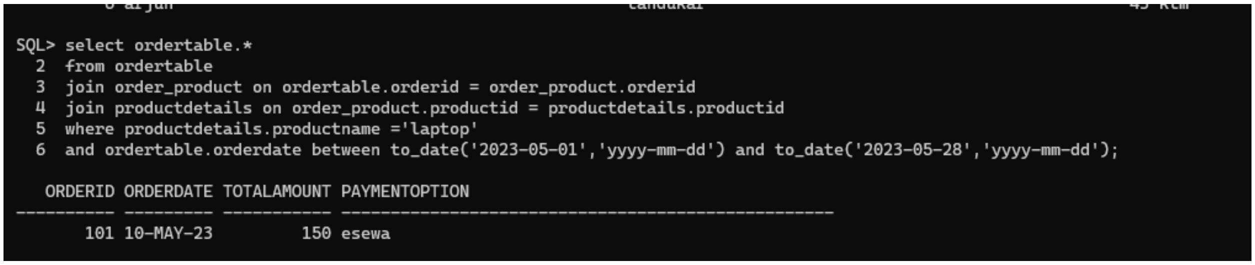
```
SQL> select *
  2  from customer
  3  where categoryid =2;

CUSTOMERID FIRSTNAME                                          LASTNAME                                           ADDRESS                                            CATEGORYID
---------- -------------------------------------------------- -------------------------------------------------- -------------------------------------------------- ----------
         1 sandesh                                            pokhreal                                           123 biratnagar                                              2
         4 rakesh                                             pandey                                             43 rajbiraj                                                 2
         6 arjun                                              tandukar                                           43 ktm                                                      2
```

Figure 24:information query 1

In this query select all column from the customer table where value in the column category id is equal to 2.

2.)List all the orders made for any particular product between the dates 01-05-2023 till 28-05-2923.

Select ordertable. *
From ordertable
Join order_product on ordertable. Orderid = order_product. Orderid
Join productdetails on order_product. Productid = productdetails. Productid
Where productdetails. ProductName =’laptop’
and ordertable. Orderdate between to_date(‘2023-05-01’,’yyyy-mm-dd’) and to_date(‘2023-05-28’,’yyyy-mm-dd’);



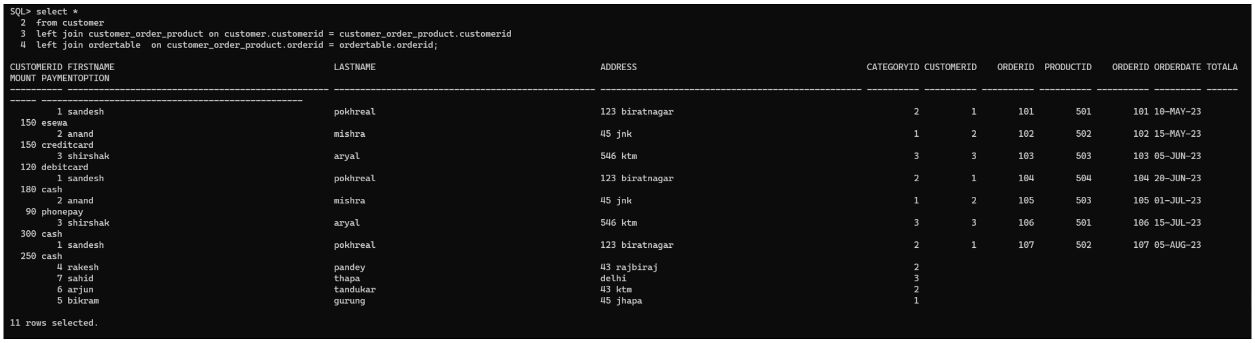
```
SQL> select ordertable.*
  2  from ordertable
  3  join order_product on ordertable.orderid = order_product.orderid
  4  join productdetails on order_product.productid = productdetails.productid
  5  where productdetails.productname ='laptop'
  6  and ordertable.orderdate between to_date('2023-05-01','yyyy-mm-dd') and to_date('2023-05-28','yyyy-mm-dd');

   ORDERID ORDERDATE TOTALAMOUNT PAYMENTOPTION
---------- --------- ----------- --------------------------------------------------
       101 10-MAY-23         150 esewa
```

Figure 25:information query 2

In this query select all column from ordertable for order the product name laptop where order between the specific date 2023-05-01 and 2023-05028.

3.) List all the customer with their orderdetail and also the customer who have not ordered any product yet.

Select *
From customer
Left join customer_order_product on customer.customerid = customer_order_product.customerid
Left join ordertable on customer_order_product.orderid = ordertable.orderid;

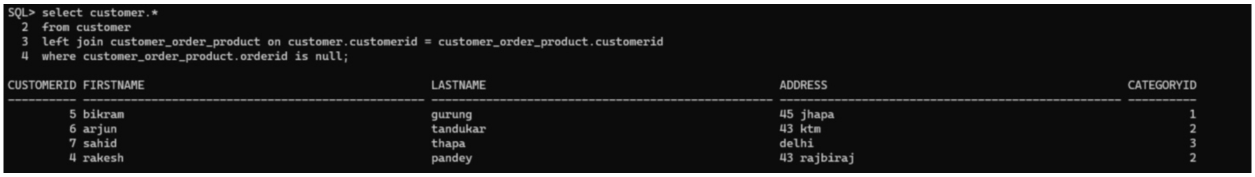


```
SQL> select *
  2  from customer
  3  left join customer_order_product on customer.customerid = customer_order_product.customerid
  4  left join ordertable  on customer_order_product.orderid = ordertable.orderid;

CUSTOMERID FIRSTNAME                                          LASTNAME                                           ADDRESS                                            CATEGORYID CUSTOMERID    ORDERID  PRODUCTID    ORDERID ORDERDATE TOTALA
MOUNT PAYMENTOPTION
---------- -------------------------------------------------- -------------------------------------------------- -------------------------------------------------- ---------- ---------- ---------- ---------- ---------- --------- ------
----- --------------------------------------------------
         1 sandesh                                            pokhreal                                           123 biratnagar                                              2          1        101        501        101 10-MAY-23
  150 esewa
         2 anand                                              mishra                                             45 jnk                                                      1          2        102        502        102 15-MAY-23
  150 creditcard
         3 shirshak                                           aryal                                              546 ktm                                                     3          3        103        503        103 05-JUN-23
  120 debitcard
         1 sandesh                                            pokhreal                                           123 biratnagar                                              2          1        104        504        104 20-JUN-23
  180 cash
         2 anand                                              mishra                                             45 jnk                                                      1          2        105        503        105 01-JUL-23
   90 phonepay
         3 shirshak                                           aryal                                              546 ktm                                                     3          3        106        501        106 15-JUL-23
  300 cash
         1 sandesh                                            pokhreal                                           123 biratnagar                                              2          1        107        502        107 05-AUG-23
  250 cash
         4 rakesh                                             pandey                                             43 rajbiraj                                                 2
         7 sahid                                              thapa                                              delhi                                                       3
         6 arjun                                              tandukar                                           43 ktm                                                      2
         5 bikram                                             gurung                                             45 jhapa                                                    1

11 rows selected.
```

Figure 26:information query 3

In this query select all column from bridge or combined table customer_order_product and ordertable also and left join is used to take all record from the customer table and match the record with order table and return all row from left table ,if there is no matching with right table the value are not select from right table .

Select customer. *
From customer
Left join customer_order_product on customer. customerid = customer_order_product. customerid
Where customer_order_product. orderid is null;



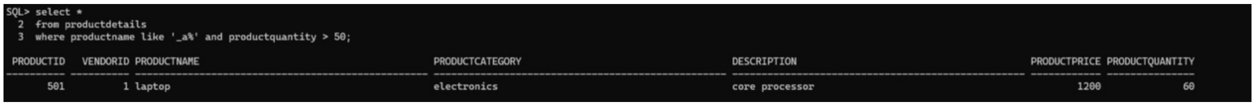
```
SQL> select customer.*
  2  from customer
  3  left join customer_order_product on customer.customerid = customer_order_product.customerid
  4  where customer_order_product.orderid is null;

CUSTOMERID FIRSTNAME                                          LASTNAME                                           ADDRESS                                            CATEGORYID
---------- -------------------------------------------------- -------------------------------------------------- -------------------------------------------------- ----------
         5 bikram                                             gurung                                             45 jhapa                                                    1
         6 arjun                                              tandukar                                           43 ktm                                                      2
         7 sahid                                              thapa                                              delhi                                                       3
         4 rakesh                                             pandey                                             43 rajbiraj                                                 2
```

Figure 27:information query 4

This query select all row data from customer first then for each customer find matching row in bridge table or combined table of customer_order_product by using customerid of customer table and then using where clause to filter the data of customer who belong only to customer does not belong to bridge or corresponding table customer_order_product table.

4.) List all the product details that have the second letter ‘a’ in their productname and have stock quantity more than 50.

Select *
From productdetails
Where productname like ‘_a%’ and productqunatity > 50;



```
SQL> select *
  2  from productdetails
  3  where productname like '_a%' and productquantity > 50;

 PRODUCTID   VENDORID PRODUCTNAME                                        PRODUCTCATEGORY                                    DESCRIPTION                                        PRODUCTPRICE PRODUCTQUANTITY
---------- ---------- -------------------------------------------------- -------------------------------------------------- -------------------------------------------------- ------------ ---------------
       501          1 laptop                                             electronics                                        core processor                                             1200              60
```

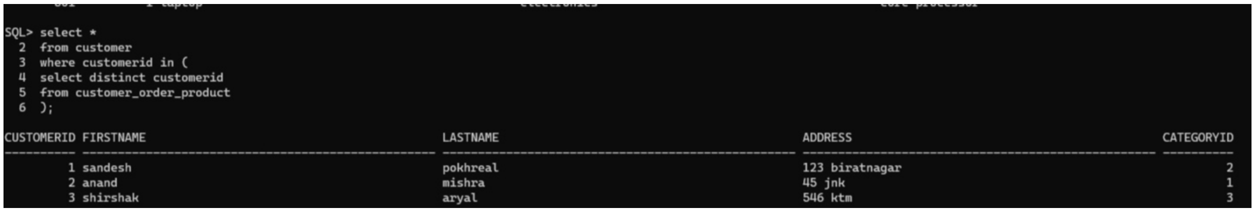
Figure 28:information query 5

The use of this query retrieves all data from the productdetails where the ProductName second letter start with ‘a’ and the product quantity of product the second letter ‘a’ is greater than 50.

5.) Find out the customer who has ordered recently.

Select *
From customer
Where customer in (
Select distinct customerid.

```
From customer_order_product
);
```



```
SQL> select *
  2  from customer
  3  where customerid in (
  4  select distinct customerid
  5  from customer_order_product
  6  );

CUSTOMERID FIRSTNAME                                          LASTNAME                                           ADDRESS                                            CATEGORYID
---------- -------------------------------------------------- -------------------------------------------------- -------------------------------------------------- ----------
         1 sandesh                                            pokhreal                                           123 biratnagar                                              2
         2 anand                                              mishra                                             45 jnk                                                      1
         3 shirshak                                           aryal                                              546 ktm                                                     3
```

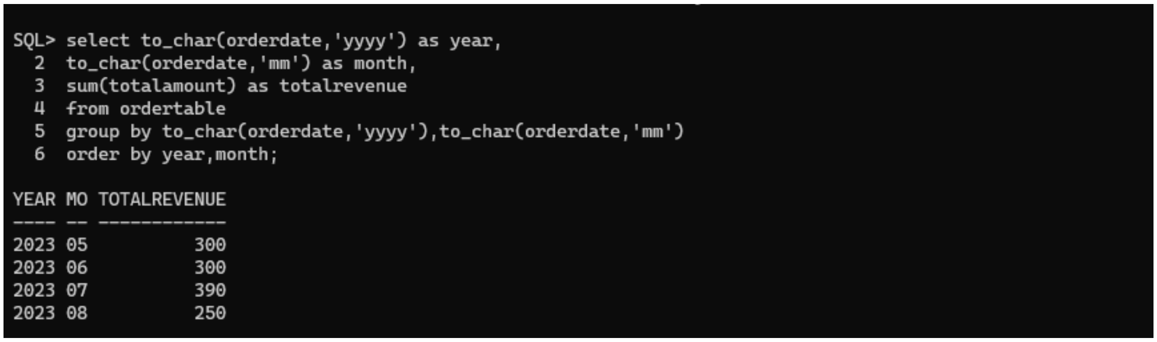
Figure 29:information query 6

In this query select all data from customer table it can find the result of inner query and then in query customerid select distinct customer id values from customer_order_prodcut table.

4.2. Transaction query

1.) show the total revenue of the company for each month.

```
Select to_char(orderdate,’yyyy’) as year,
to_char (orderdate,’mm’) as month,
sum(totalamount) as totalrevenue
from ordertable
group by to_char(orderdate,’yyyy’),to_char(orderdate,’mm’)
order by year,month;
```



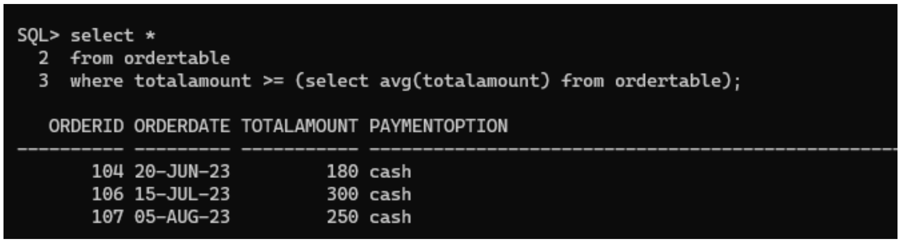
```
SQL> select to_char(orderdate,'yyyy') as year,
     to_char(orderdate,'mm') as month,
     sum(totalamount) as totalrevenue
     from ordertable
     group by to_char(orderdate,'yyyy'),to_char(orderdate,'mm')
     order by year,month;

YEAR MO TOTALREVENUE
---- -- ------------
2023 05          300
2023 06          300
2023 07          390
2023 08          250
```

Figure 30:transaction query 1

In this query retrieved date from order table and this query show the the total revenue for each month by using group by clause it group all order on the basis of the year and month of the order data in ordertable.

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

Select *
from ordertable
where totalamount >= (select avg(totalamount) from ordertable);



```
SQL> select *
     from ordertable
     where totalamount >= (select avg(totalamount) from ordertable);

   ORDERID ORDERDATE TOTALAMOUNT PAYMENTOPTION
---------- --------- ----------- --------------------------------------------------
       104 20-JUN-23         180 cash
       106 15-JUL-23         300 cash
       107 05-AUG-23         250 cash
```

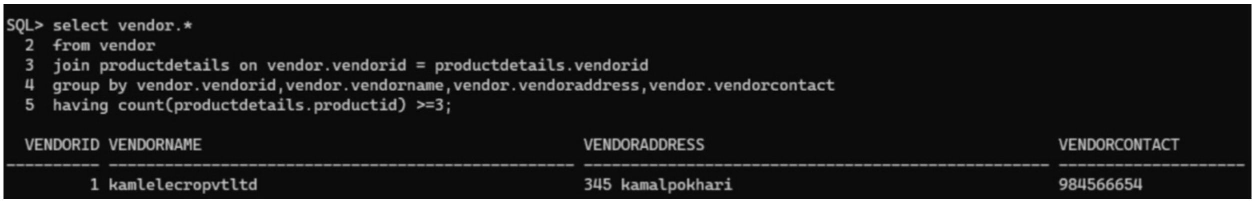
Figure 31:transaction query 2

In this query select all column from ordertable where the totalamount is equal to or higher than the average order total value .

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

Select vendor. *
From vendor
Join productdetails on vendor. vendorid = productdetails. vendorid

```
Group by vendor. vendorid, vendor. vendorname,vendor.
vendoraddress,vendor.vendorcontact
having count (productdetails. productid) >=3;
```



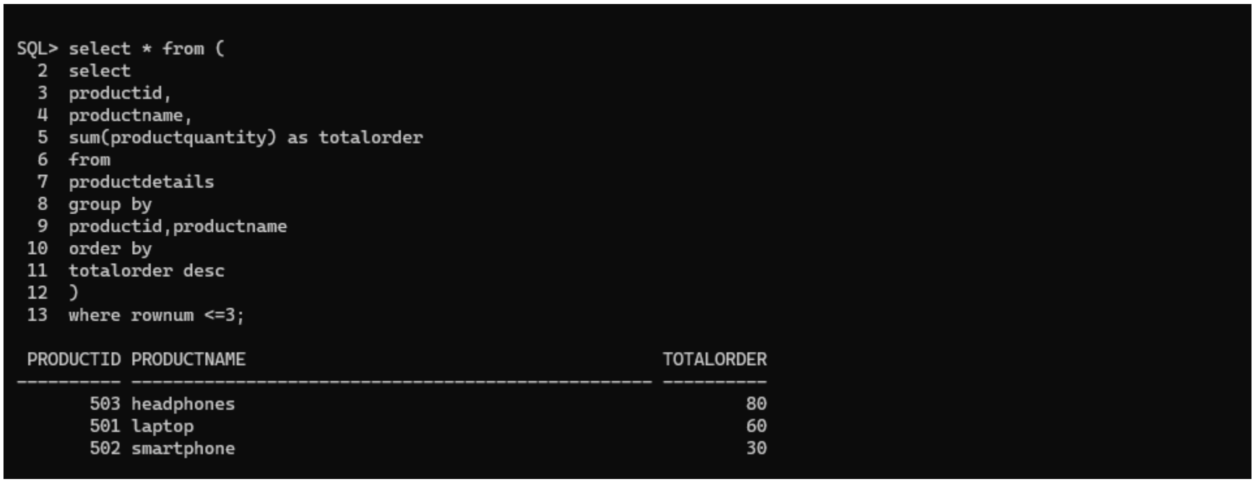
```
SQL> select vendor.*
  2  from vendor
  3  join productdetails on vendor.vendorid = productdetails.vendorid
  4  group by vendor.vendorid,vendor.vendorname,vendor.vendoraddress,vendor.vendorcontact
  5  having count(productdetails.productid) >=3;

  VENDORID VENDORNAME                                         VENDORADDRESS                                      VENDORCONTACT
---------- -------------------------------------------------- -------------------------------------------------- --------------------
         1 kamlelecropvtltd                                   345 kamalpokhari                                   984566654
```

Figure 32:transaction query3

In this query select all column from vendor who have supplied three or more than three products to the company.

4.) show the top 3 product details that have been ordered the most.

```
Select * from (
Select
Productid,
Productname,
Sum(productquantity) as totalorder
From
Productdetails
Group by
Productid,productname
Order by
totalorder desc
)
Where rownum <=3;
```



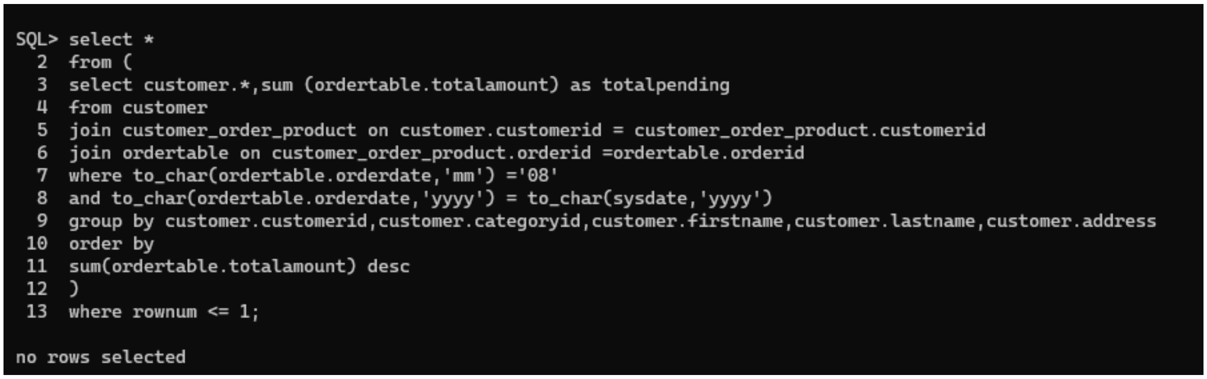
```
SQL> select * from (
select
productid,
productname,
sum(productquantity) as totalorder
from
productdetails
group by
productid,productname
order by
totalorder desc
)
where rownum <=3;

 PRODUCTID PRODUCTNAME                                        TOTALORDER
---------- -------------------------------------------------- ----------
       503 headphones                                                 80
       501 laptop                                                     60
       502 smartphone                                                 30
```

Figure 33:transaction query 4

In this query select the top 3 product and the three product is highest orderqunatity from productdetails table.

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

```
Select *
from (
select customer.*,sum(ordertable.totalamount) as totalspending
from customer
join customer_order_prodcut on customer.customerid =
customer_order_product.customerid
join ordertable on customer_order_product.orderid =
ordertable.orderid
where to_char(ordertable.orderdate,’yyyy’) =
to_char(sysdate,’yyyy’)
group by
customer.customerid,customer.categoryid,customer.firstname,cus
tomer.lastname,customer.address
order by
sum(ordertable.totalamount) desc
)
Where rownum <=1;
```

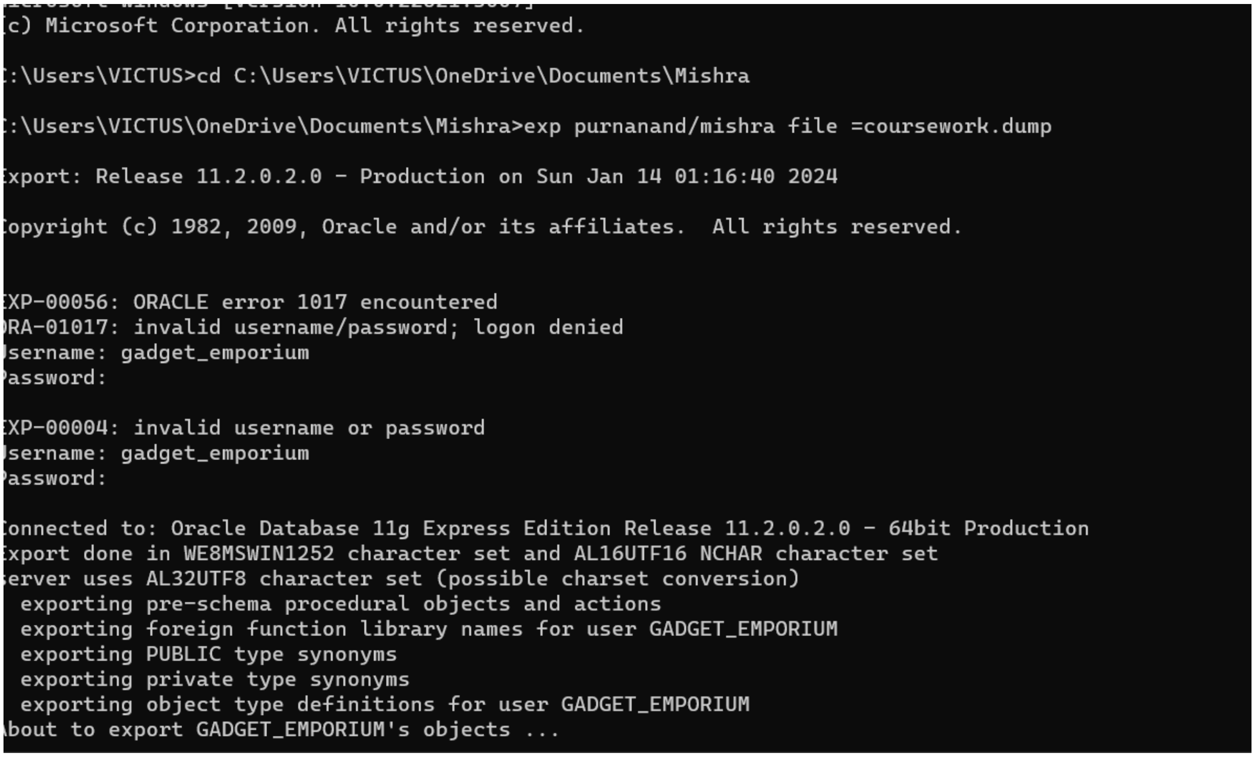


```
SQL> select *
from (
select customer.*,sum (ordertable.totalamount) as totalpending
from customer
join customer_order_product on customer.customerid = customer_order_product.customerid
join ordertable on customer_order_product.orderid =ordertable.orderid
where to_char(ordertable.orderdate,'mm') ='08'
and to_char(ordertable.orderdate,'yyyy') = to_char(sysdate,'yyyy')
group by customer.customerid,customer.categoryid,customer.firstname,customer.lastname,customer.address
order by
sum(ordertable.totalamount) desc
)
where rownum <= 1;

no rows selected
```

Figure 34:transaction query 5

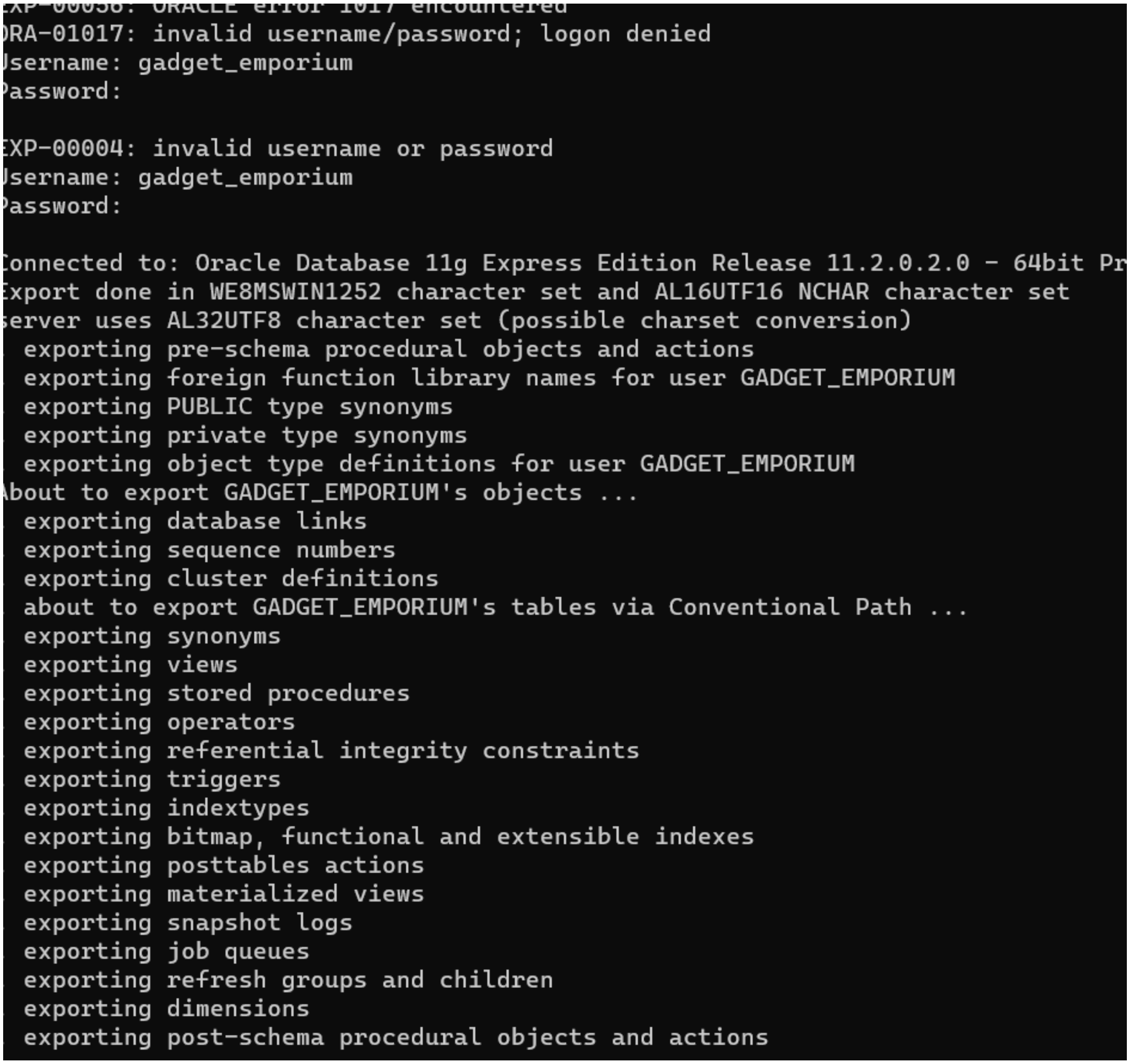
In this query to find the customer who has most ordered in august but there is no customer who ordered most in august .

4.3.Creation of dumpfile

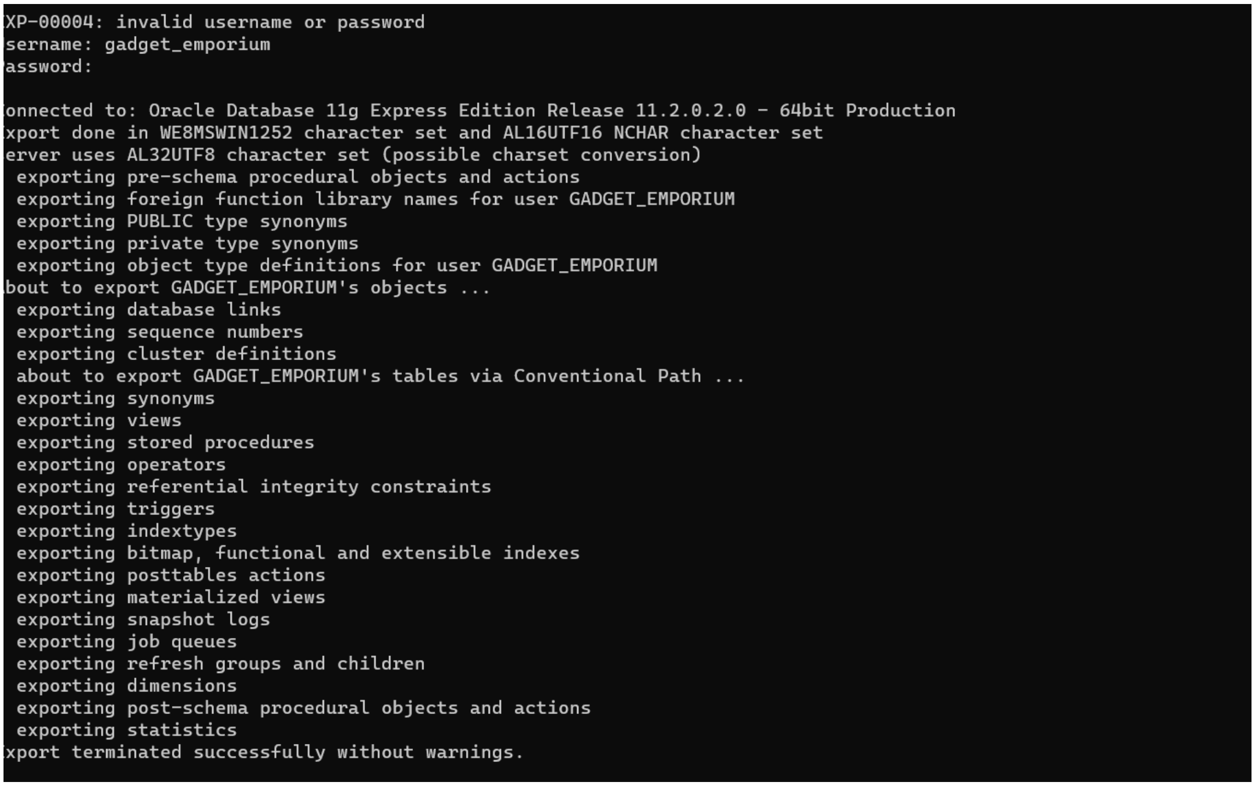


```
(c) Microsoft Corporation. All rights reserved.

:\Users\VICTUS>cd C:\Users\VICTUS\OneDrive\Documents\Mishra

:\Users\VICTUS\OneDrive\Documents\Mishra>exp purnanand/mishra file =coursework.dump

xport: Release 11.2.0.2.0 - Production on Sun Jan 14 01:16:40 2024

opyright (c) 1982, 2009, Oracle and/or its affiliates.  All rights reserved.


XP-00056: ORACLE error 1017 encountered
RA-01017: invalid username/password; logon denied
sername: gadget_emporium
assword:

XP-00004: invalid username or password
sername: gadget_emporium
assword:

onnected to: Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
xport done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set
erver uses AL32UTF8 character set (possible charset conversion)
 exporting pre-schema procedural objects and actions
 exporting foreign function library names for user GADGET_EMPORIUM
 exporting PUBLIC type synonyms
 exporting private type synonyms
 exporting object type definitions for user GADGET_EMPORIUM
bout to export GADGET_EMPORIUM's objects ...
```

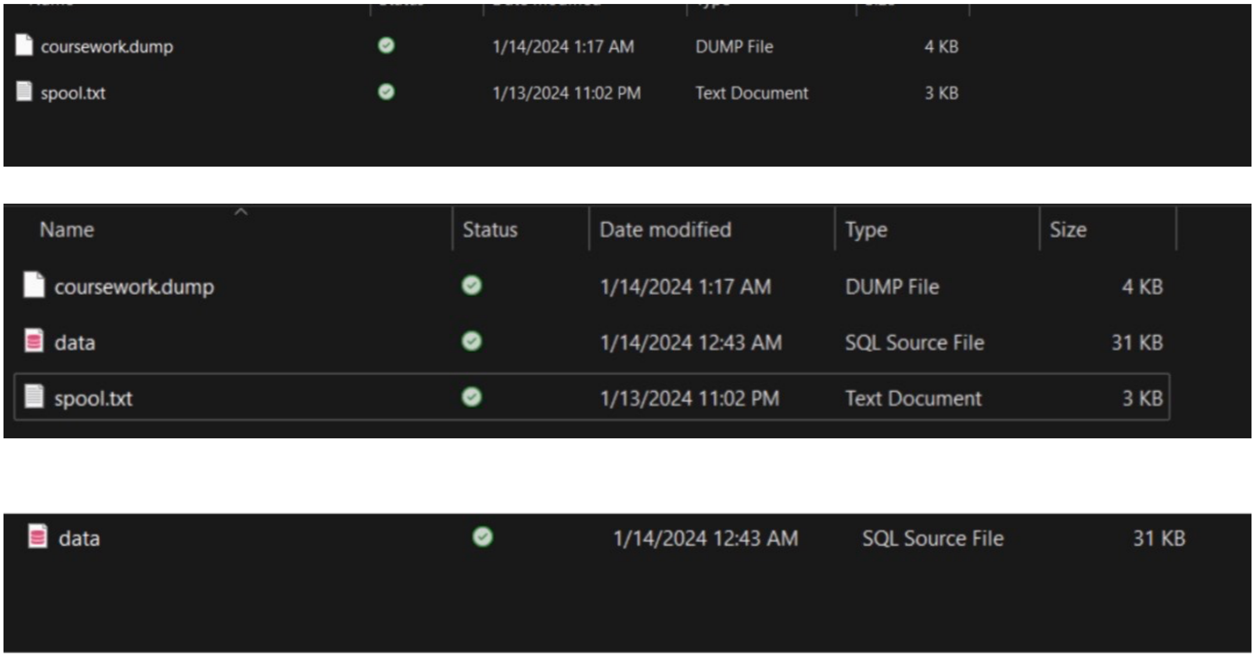
Figure 35:creation of dump file1



```
EXP-00056: ORACLE error 1017 encountered
ORA-01017: invalid username/password; logon denied
Username: gadget_emporium
Password:

EXP-00004: invalid username or password
Username: gadget_emporium
Password:

Connected to: Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Pr
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 GADGET_EMPORIUM
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user GADGET_EMPORIUM
About to export GADGET_EMPORIUM's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export GADGET_EMPORIUM's tables via Conventional Path ...
. exporting synonyms
. exporting views
. exporting stored procedures
. exporting operators
. exporting referential integrity constraints
. exporting triggers
. exporting indextypes
. exporting bitmap, functional and extensible indexes
. exporting posttables actions
. exporting materialized views
. exporting snapshot logs
. exporting job queues
. exporting refresh groups and children
. exporting dimensions
. exporting post-schema procedural objects and actions
```



```
XP-00004: invalid username or password
sername: gadget_emporium
assword:

onnected to: Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
xport done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set
erver uses AL32UTF8 character set (possible charset conversion)
 exporting pre-schema procedural objects and actions
 exporting foreign function library names for user GADGET_EMPORIUM
 exporting PUBLIC type synonyms
 exporting private type synonyms
 exporting object type definitions for user GADGET_EMPORIUM
bout to export GADGET_EMPORIUM's objects ...
 exporting database links
 exporting sequence numbers
 exporting cluster definitions
 about to export GADGET_EMPORIUM's tables via Conventional Path ...
 exporting synonyms
 exporting views
 exporting stored procedures
 exporting operators
 exporting referential integrity constraints
 exporting triggers
 exporting indextypes
 exporting bitmap, functional and extensible indexes
 exporting posttables actions
 exporting materialized views
 exporting snapshot logs
 exporting job queues
 exporting refresh groups and children
 exporting dimensions
 exporting post-schema procedural objects and actions
 exporting statistics
xport terminated successfully without warnings.
```



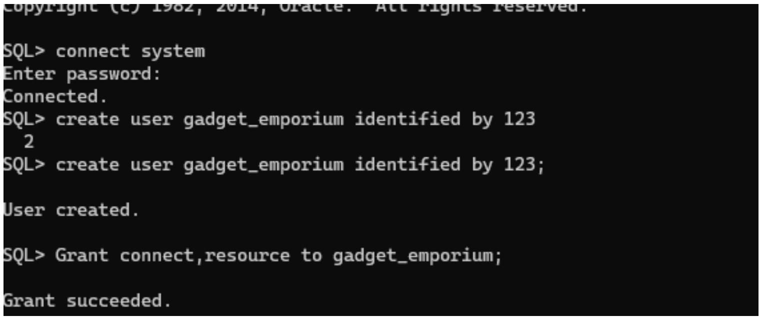
coursework.dump	✓	1/14/2024 1:17 AM	DUMP File	4 KB
spool.txt	✓	1/13/2024 11:02 PM	Text Document	3 KB

Name	Status	Date modified	Type	Size
coursework.dump	✓	1/14/2024 1:17 AM	DUMP File	4 KB
data	✓	1/14/2024 12:43 AM	SQL Source File	31 KB
spool.txt	✓	1/13/2024 11:02 PM	Text Document	3 KB

data	✓	1/14/2024 12:43 AM	SQL Source File	31 KB

Figure 36:creation of spool file

4.4. create user file



```
Copyright (c) 1982, 2014, Oracle.  All rights reserved.

SQL> connect system
Enter password:
Connected.
SQL> create user gadget_emporium identified by 123
  2
SQL> create user gadget_emporium identified by 123;

User created.

SQL> Grant connect,resource to gadget_emporium;

Grant succeeded.
```

Figure 37:create user file

Chapter 5- Critical Evaluation

5.1.Critical Evaluation of module, its usage and relation with another subject

Any kind of data can be found in the database. A method of accessing, modifying, and updating it should be available once it has been stored. we have a tool for this called. a query that can be used to complete all of these tasks .SQL,DMX(Data mining language), and other query and other language only a few examples .Each of these has a unique set of function, and each of these language has a separate set of function structure .The goal of this project was to build a database for whatever e-commerce company we choose ,whether it be local or worldwide one like amazon or daraz or another .from among these e-commerce company ,I have been chosen ‘amazon’ one of all over famous and whose main office in USA .Giving customers delivery services based on their needs was the primary goal of this company.in addition to offering courier and equipment and gods services .

Many other tasks, including establishing an ER diagram, normalizing the tables, and building databases, were carried out during this project. To store various type of product, product details, productname, vendor, vendor address, vendor contact, customer, customercategory, a database was established. Since data and information are essential for any firm to operate profitably. To finish this job on time, several types of research were conducted.

5.2. Critical Assessment of coursework.

While working on the project, I learn lot of new things like or lot of new information about ER diagram, normalization, sql command, and another subject while working on the project. This semester course work introduced me to the concept of normalization, which was very useful concept or topic in database. A number of problem arise with initial ER diagram ,such as data redundancy and anomalies ,the final erd was obtained when it was simulated .we must carefully finish the normalization we must go through the various stage in order to normalize .The Un-normalized form(UNF),first normal form(1NF),and third normal form(3NF) are those steps .After , the tables were made and values were added to the

Chapter-6 Conclusion

Using oracle sql plus ,gadget emporium may now manage every product record from the database ,according to the coursework implemented in this report .After lots of research ,consultation with the instructor ,Mrs.yunisha bajracharya,lot of hard effort ,the report was completed .gaining lots of knowledge about business rules ,erd ,normalization, data implementation ,database populating, information queries and transaction query ,.after the obtaining the concepts of the scenario from the coursework, the report was completed

Chapter-7 references

Bibliography

IBM. (2021, 8 1). From IBM: https://www.ibm.com/docs/en/imdm/12.0?topic=concepts-key-entity-attribute-entity-type

Geeks for geek. (2023, may 1). From Geeks for geek: https://www.geeksforgeeks.org/difference-between-primary-key-and-unique-key/

peterson, R. (2023, 12 26). *Guru 99*. From Guru 99: https://www.guru99.com/database-normalization.html

geeks for geek. (2023, nov 7). From geeks for geek: https://www.geeksforgeeks.org/first-normal-form-1nf/

Software AG. (2023, 2 15). From Software Ag : https://documentation.softwareag.com/webmethods/compendiums/v10-7/C_Design_and_Implement_BPMs/index.html#page/design_implement_bpm_compendium/D352039.html