CC5051NI Databases

50% Individual Coursework

Autumn 2023

Student Name: Ashim Poudel

London Met ID: 22085505

Assignment Submission Date: Jan 15, 2024

Word Count: 2709

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

1.1 Introduction of Business

Welcome to “Gadget Emporium”- premier destination for the newest gadgets and accessories. Gadget Emporium is online store that was founded by businessman and electronics enthusiast Mr. John. Their objective is to supply a wide range of electronic devices and accessories to both private individuals and commercial entities. Our goal is to provide our consumers with a flawless online purchasing experience by putting the most up-to-date technology at their fingers.

Gadget emporium is a retailer that specializes in providing a wide variety of electronic products to meet the demands of experts, regular consumers, and tech enthusiasts. Our carefully chosen assortment of guarantees that clients have access to high-end, cutting-edge technology, ranging from computers and smartphones to smart home applications and accessories.

1.1 Current Business Activities and Operations

Nepal has been stepping towards in e-commerce business to facilitate all people. The advancement of business has been enhanced by the technology we have. By using this advantage, we have developed “Gadget Emporium” for better experience in shopping specifically electronics. We provide all the electronic gadgets as well as accessories in reasonable price. The “Gadget Emporium” goal has always been to satisfy their valuable customers, providing high quality products and deleveraging in time.

1. Customers get access to modern technology due to the careful curation of each product.
2. Customers are categorized into three groups: regular, staff, and VIP. On their product purchases, each client type is eligible for a certain percentage of discount.
3. Every order is tracked by the system, which includes the products bought, their quantities, unit costs, and the total order amounts.
4. Because each product has a unique vendor, quality control and accountability are guaranteed.
5. Products include inventory information to let customers know about their availability, such as stock quantity or availability status.
6. Names, addresses, and contact data of customers are kept on file for order processing and correspondence.
7. It is simple for shoppers to browse products, place orders, and keep track of their purchases using the online store's user-friendly design.

1.2 Aims

- To create an online platform, “Gadget Emporium” to serve as marketplace for electronic devices and accessories.
- To provide a user-friendly online shopping experience, advance customer satisfaction and loyalty.
- Maintaining strong relationships with vendors and suppliers to guarantee the quality and authenticity of products.
- Facilitating secure payment processing.

1.3 Objective

- To develop a fully functional and secure E-commerce platform for “Gadget Emporium” and deploy it for public access.
- Creating a customer management system that categorizes customers and stores essential information, including address and discount rates.
- Creating order processing system that gathers all the information required when a customer checks out.
- Efficiency manages vendor relationships, ensuring that each product is associated with a single vendor and that vendors can supply a variety of products.

1.3 Business Rules

- Product Management: The system should handle the details of electronic gadgets and accessories, including product names, descriptions, categories, prices, and stock levels. Each product must be of only one category and each category can have one or many products.
- Customer Categories and Discounts: The system should be able to keep track of all its customers. Customers should be categorized as Regular (R), Staff (S), and VIP (V). Each category is entitled to a different discount rate on product purchases, such as 0%, 5% and 10% respectively. Each customer’s address must also be stored which helps in the delivery process.
- Order Processing: Customers can browse and purchase one or more electronic gadgets online. The system must record the details of each order, including the products purchased, quantities, unit price, and total order amount. An order can have multiple products and any one type of product might be included in multiple orders placed by various customers.
- Vendor Management: Maintain records of vendors or suppliers providing electronic gadgets and accessories. Each product should be associated with a single vendor. Each vendor can supply one or more products.
- Product Availability and Inventory Management: Track real-time product availability to prevent overselling and maintain accurate stock levels. A product must have inventory details like stock quantity or availability status.
- Payment Processing: The system should integrate with various payment gateways to facilitate secure and seamless transactions of each order. Payment options must be either cash on delivery, credit/debit card or e-wallet. Each order detail must have one payment option.
- An invoice must be issued once the customer checks out their order after confirmation which must include the details of order, customer and payment details (with discount).

1.4 Assumption

- Product information includes details such a product_name, Price, and product category.
- It is anticipated that the vendor_id, which serves as the primary key, uniquely identifies each vendor.
- To construct the relationship between vendors and products, it is assumed that a distinct table (Vendor_Product) exists.
- Customer discounts are based on customer categories (e.g., Regular, Staff, VIP).
- It is assumed that the Customer_Info database has foreign keys that link to the Customer and Product tables, respectively, called customer_id and product_id.
- Order_info is linked to the order and customer table.

2.0 Identification of Entities and Attributes

Entity:

An entity is an object or component of data that is to be represented in databases. It may be a tangible object, a simple fact about the business, or an actual event that takes place. An entity, which keeps data in the database, can be a location, a person, an object, an event, or a concept. Each entity is composed of certain qualities that serve as its representation. To uniquely identify each row, entities must possess properties as well as a unique key attribute.

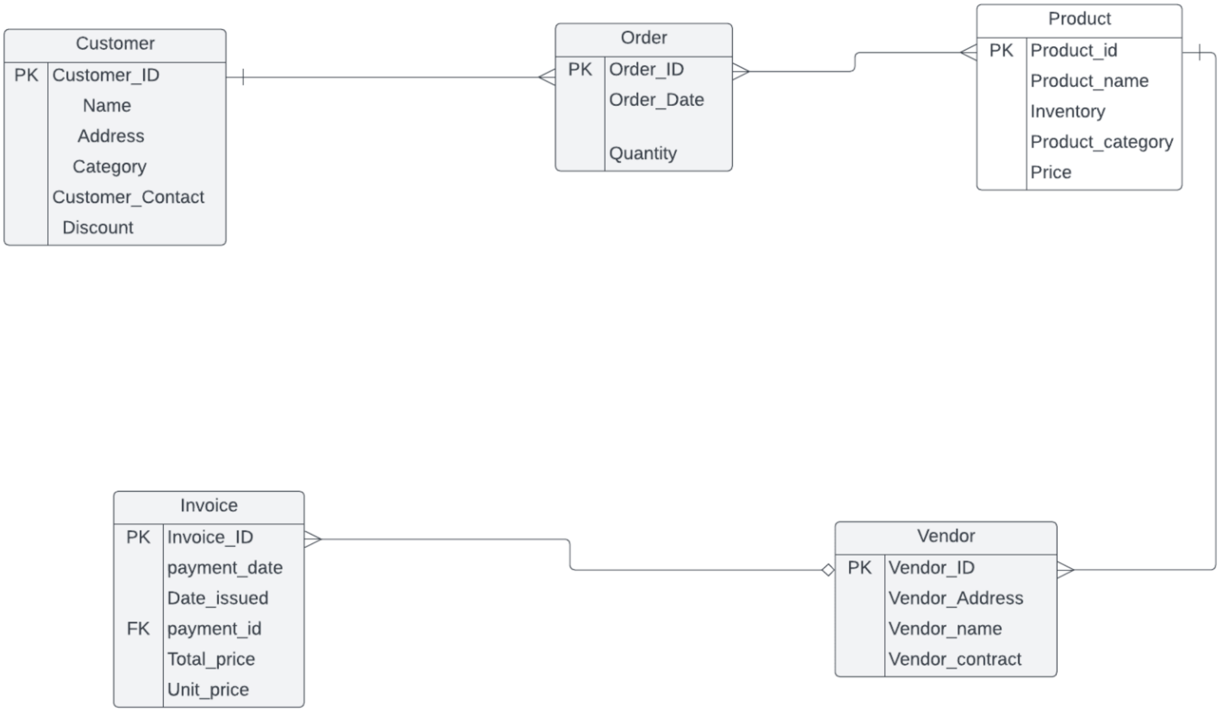
Attributes:

Attributes are properties or characteristics that describe an entity. This is the single-valued property of entity-type or relationship-type. Every attribute has unique name or a data type.

The entity and attributes used in database system are in the following table:

Entity name	Attributes
Product	(Producr_id, Product_name, Prise, Product_category, inventory)
Vendor	(Vendor_id, Vendor_name, Vendor_Address, Vendor_contact)
Order	(Order_id, Order_date, Quantity,)
Payment	(Invoice_id, date_issued, Payment_status, Unit_price, payment_date, total_amount)
Customer	(Customer_id, Customer_category, Discount, Customer_add, Customer_name)

2.1 Initial ER-Diagram



2.2 Normalization

Database Normalization is a technique of organizing the data in the database. Normalization is a database design process used for data modelling or database creation where we can of decompose and form tables in a manner to reduce data redundancy (repetition) and data dependency, so it can be added and updated efficiently. It divides larger tables to smaller tables and links them using relations. It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables. (Guru99, 2019)

If the table is not properly normalized and have data redundancy, then it will not only consume additional memory space but will also make it difficult to handle and update the database, without facing data loss. Insertion, Updating and Deletion anomalies are numerous if database is not normalized. And hence, the database should be properly normalized for creating a good database and make the database more efficient. Normalization in a DBMS (Database Management System) can be done in Oracle, Microsoft SQL Server, MySQL, PostgreSQL and any type of database. (Study tonight, 2019) (Brumm, 2019)

One method of organising the data in the database is called database normalization In order to eliminate data redundancy (repetition) and data reliance and enable efficient additions and updates, normalisation is a database design method that is used for data modelling or database building. Larger tables are divided into smaller tables and connected by relations. Data is tabulated through a multi-step procedure that eliminates duplicate data from related tables. (2019, Guru99)

The steps to obtain normalization are:

2.2.1 UNF

Gadget Emporium

(Product_id, Product_name, Price, Product category, Inventory

{Order_ID, Quantity, Order Date, invoice_id, date issued, payment status, unit price, payment_date, total_amount, customer_id, costumer category, Discount, Address, vendor_id, vendor_name, vendor_address, vendor_contact.

(Product_id, Product_name, Price, Product category, Inventory {vendor id, vendor name, vendor address, vendor contact.} {customer_id, costumer category, Discount, Address

{Order ID, Quantity, Order Date {invoice id, date issued, payment status, unit price, payment date, total amount,}})

In this un-normalized form, product is the name of the relation (entity) and product_id is a unique identifier for the relation.

Now, applying first normal form to this un_normalized form.

2.2.2 First Normal Form(1NF)

First Normal Form (1NF): Your data is represented in First Normal Form once the first normalization rule has been applied to it. A table must first eliminate repetitive groups and each attribute (column) cannot have more than one value to be in 1NF. Only atomic values should be stored there. (Book for Beginners, 2012–2019)

Product-1 Product_id(pk) Product_name, Price, Product category, Inventory

Vendor-1vendor id(pk), vendor name, vendor address, vendor contact, Product_id(fk)

Customer-1 customer_id(pk), costumer category, Discount, Address, Product_id(fk)

Order-1 Order ID(pk), Quantity, Order Date, Product_id(fk), customer_id (fk)

Invoice-1 invoice id(pk), date issued, payment status, unit price, payment date, total amount, Product_id(fk) order id (fk), customer_id (fk)

2.2.3 Second Normal Form (2NF):

A table must first meet the conditions of First Normal Form and not have any partial dependencies to be in 2NF. When a table's attribute only depends on a portion of the primary key rather than the entire key, this is known as partial dependency.
(2019, Stadytonight)

Product-2 Product_id(pk) Product_name, Price, Product category, Inventory

Vendor-2vendor id(pk), vendor name, vendor address, vendor contact, Product_id(fk)

Product_id---------- nothing

vendor id-------------- vendor name, vendor address, vendor contact

vendor id, Product_id---------- nothing

Customer-2 customer_id, costumer category, Discount, Address, Product_id

Product_id ------------nothing

customer_id-----------------customer name costumer category, Discount, Address

customer_id, Product_id------------nothing

Order-2 Order ID, Quantity, Order Date, Product_id(fk), customer_id (fk)

Product_id-----------nothing

customer_id---------nothing

Order ID-------------- Quantity, Order Date

Product_id, customer_id------------------nothing

Invoice-2 invoice id, date issued, payment status, unit price, payment date, total amount, Product_id(fk) order id (fk), customer_id (fk)

Product_id--------------- unit price,

order id ------------------ nothing

customer_id -------------------- nothing

invoice id----------- date issued, payment status, unit price, total amount, payment date, payment method

order id, customer_id ----------total amount

Product_id, order id, customer_id--------------nothing

Product_id, order id--------------nothing

2.2.4 Third Normal Form (3NF)

Product-3 Product_id(pk) Product_name, Price, Product category,

Product info -3 Product id(fk) vendor id(fk) ---------------------Inventory

Vendor-3 vendor id(pk) vendor name, vendor address, vendor contact

Vendor_product-3 vendor id(fk), Product_id(fk)

Customer-3 customer_id(pk), customer name, Customer_add, Customer_contact

Discount-3 costumer_category, Discount_percentag, customer_id(fk)

Customer_info-3 customer_id(fk), Product_id(fk)

Order-3 Order_ID(pk), Quantity, Order Date

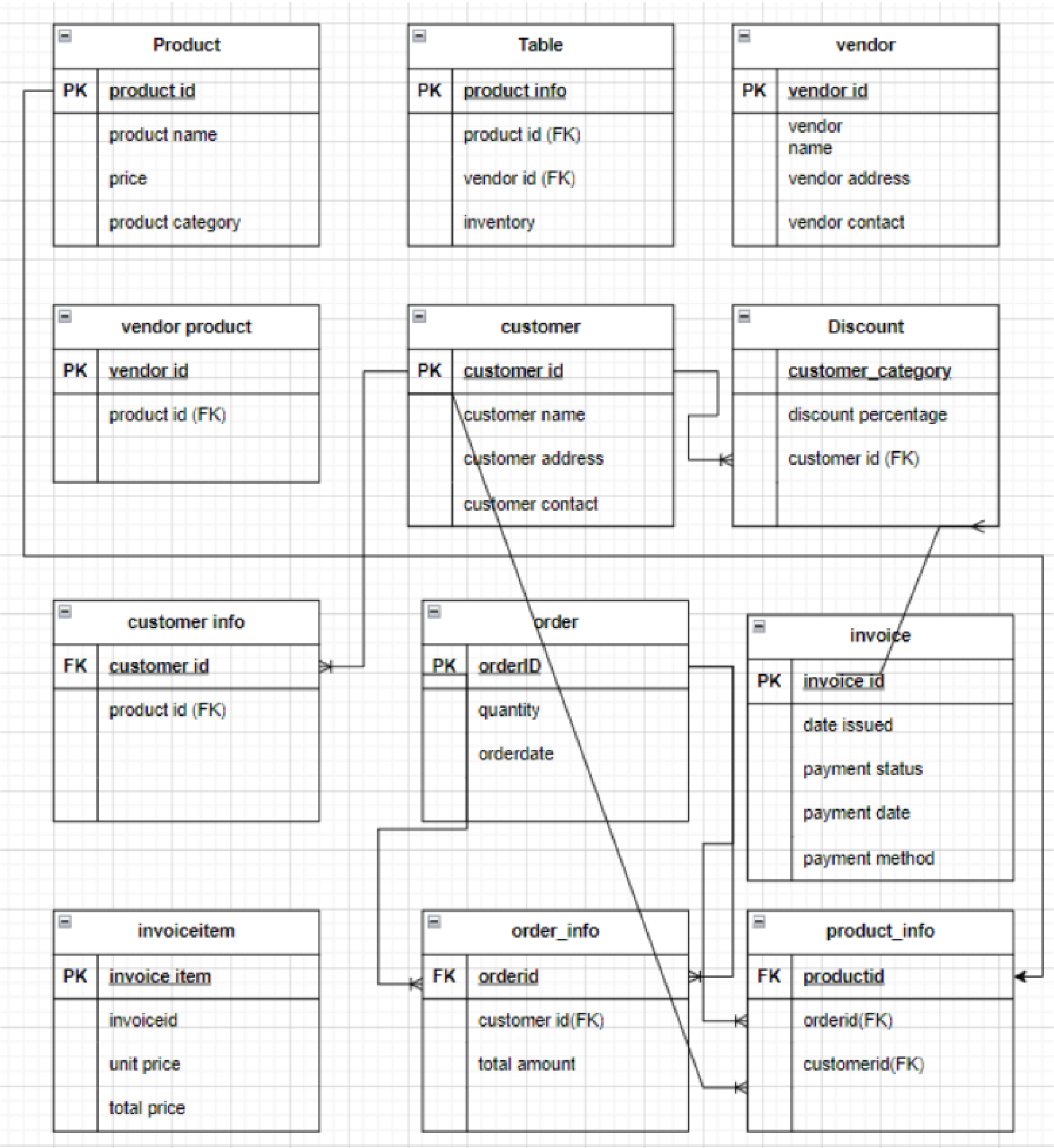
Invoice-3 invoice_id(pk) date issued, payment status, payment date, payment method

Invoiceitem-3 invoiceitem id, Invoice id, unit price, total price

Order_info-3 order_id(fk) customer_id(fk) total amount

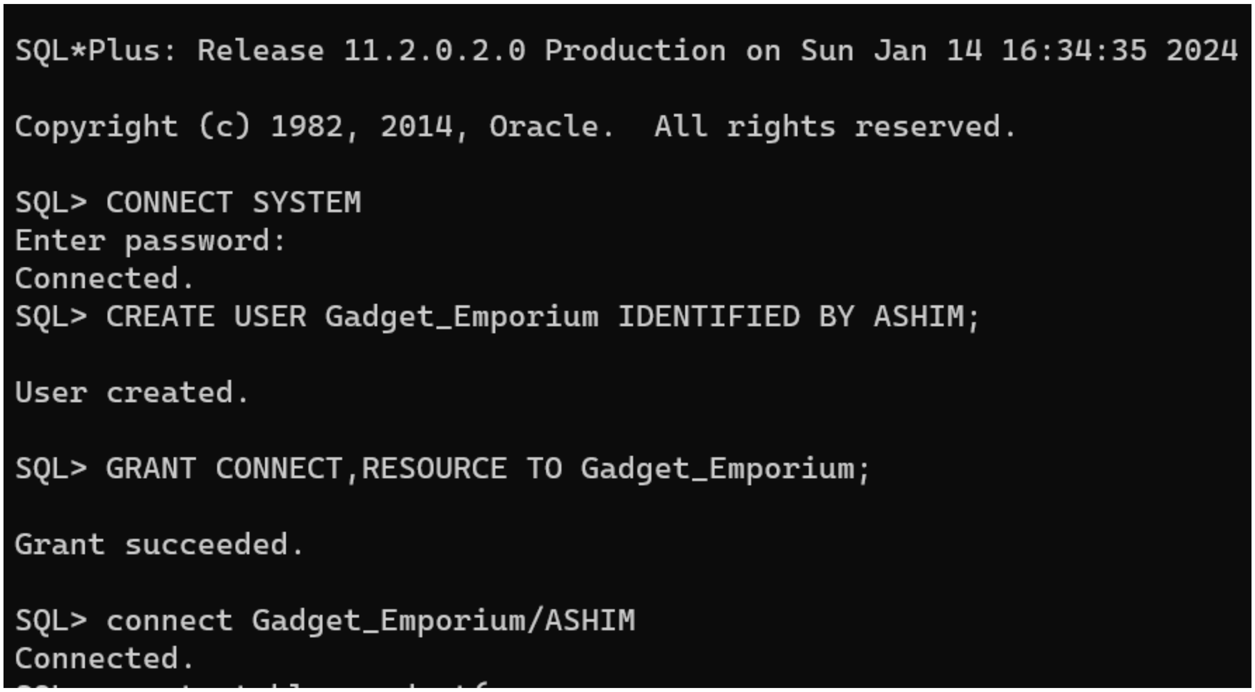
Product_info-3 Product_id(fk), order_id(fk), customer_id(fk)

2 Final ERD


Product
PK
product id
product name
price
product category
Table
PK
product info
product id (FK)
vendor id (FK)
inventory
vendor
PK
vendor id
vendor name
vendor address
vendor contact
vendor product
PK
vendor id
product id (FK)
customer
PK
customer id
customer name
customer address
customer contact
Discount
customer_category
discount percentage
customer id (FK)
customer info
FK
customer id
product id (FK)
order
PK
orderID
quantity
orderdate
invoice
PK
invoice id
date issued
payment status
payment date
payment method
invoiceitem
PK
invoice item
invoiceid
unit price
total price
order_info
FK
orderid
customer id(FK)
total amount
product_info
FK
productid
orderid(FK)
customerid(FK)

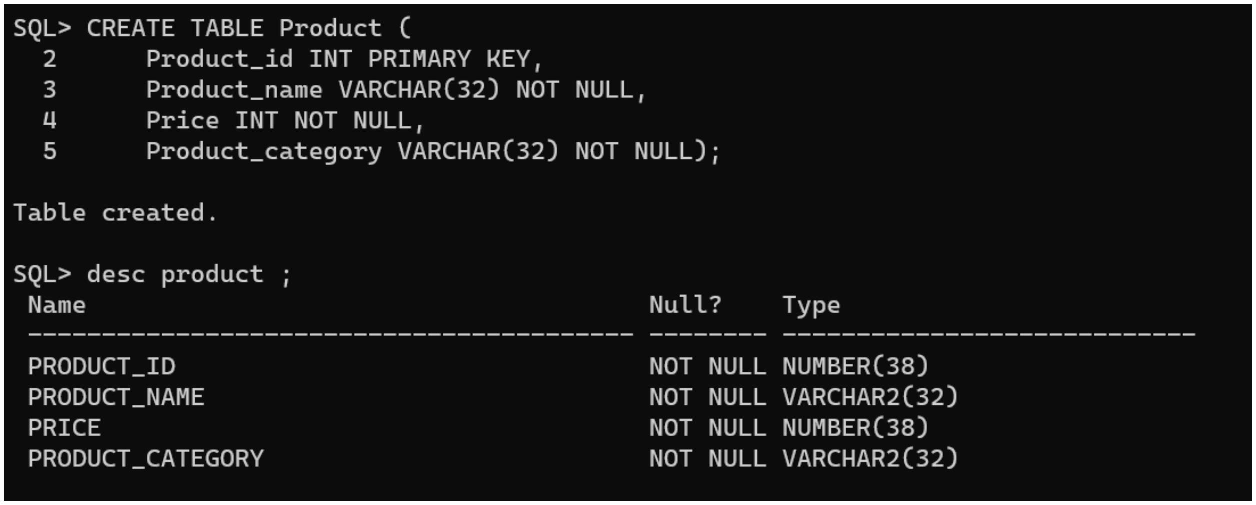
4.0 Database Implementation

4.1 User creation



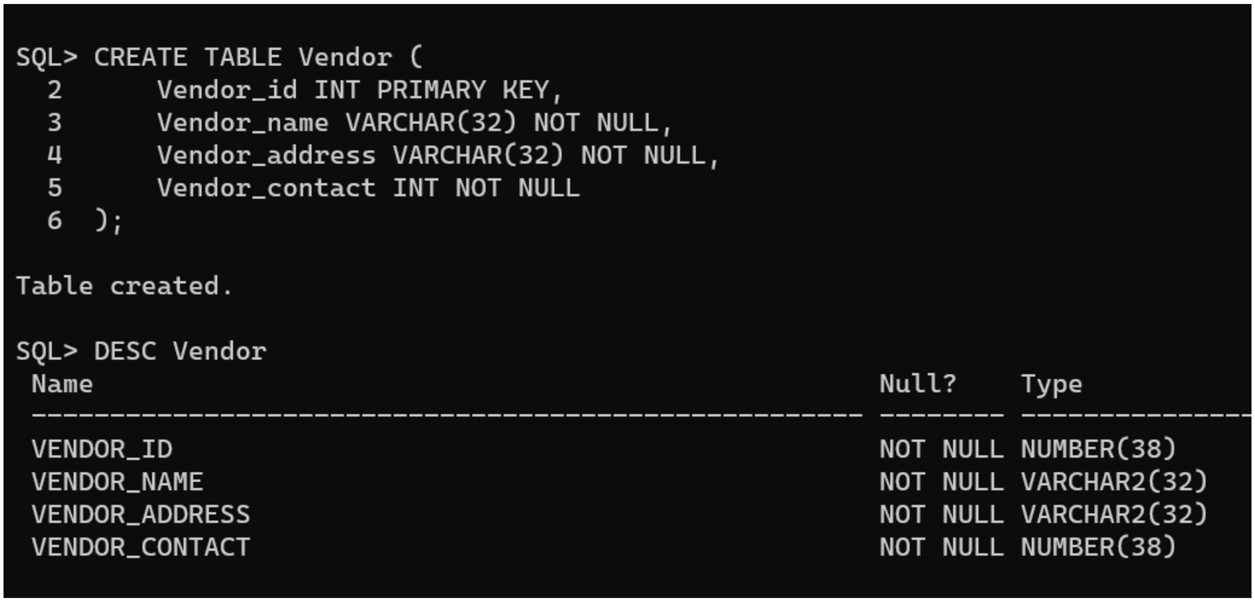
```
SQL*Plus: Release 11.2.0.2.0 Production on Sun Jan 14 16:34:35 2024

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

SQL> CONNECT SYSTEM
Enter password:
Connected.
SQL> CREATE USER Gadget_Emporium IDENTIFIED BY ASHIM;

User created.

SQL> GRANT CONNECT,RESOURCE TO Gadget_Emporium;

Grant succeeded.

SQL> connect Gadget_Emporium/ASHIM
Connected.
```

4.2 Product table creation



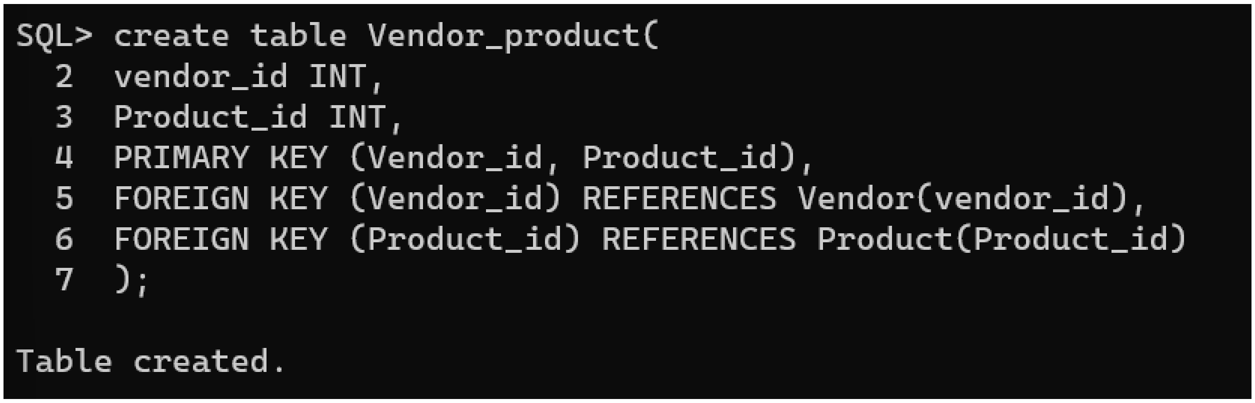
```
SQL> CREATE TABLE Product (
         Product_id INT PRIMARY KEY,
         Product_name VARCHAR(32) NOT NULL,
         Price INT NOT NULL,
         Product_category VARCHAR(32) NOT NULL);

Table created.

SQL> desc product ;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PRODUCT_ID                                NOT NULL NUMBER(38)
 PRODUCT_NAME                              NOT NULL VARCHAR2(32)
 PRICE                                     NOT NULL NUMBER(38)
 PRODUCT_CATEGORY                          NOT NULL VARCHAR2(32)
```

4.3 Vendor table creation



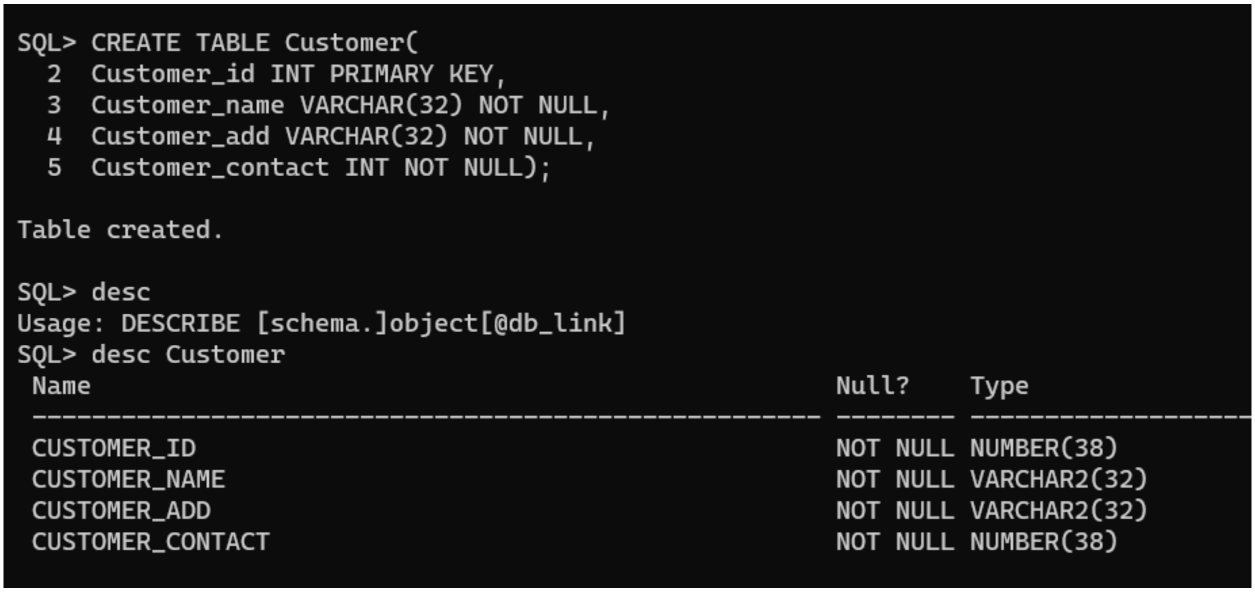
```
SQL> CREATE TABLE Vendor (
        Vendor_id INT PRIMARY KEY,
        Vendor_name VARCHAR(32) NOT NULL,
        Vendor_address VARCHAR(32) NOT NULL,
        Vendor_contact INT NOT NULL
    );

Table created.

SQL> DESC Vendor
 Name                                                      Null?    Type
 --------------------------------------------------------- -------- -------------
 VENDOR_ID                                                 NOT NULL NUMBER(38)
 VENDOR_NAME                                               NOT NULL VARCHAR2(32)
 VENDOR_ADDRESS                                            NOT NULL VARCHAR2(32)
 VENDOR_CONTACT                                            NOT NULL NUMBER(38)
```

4.4 Vendor_product table creation



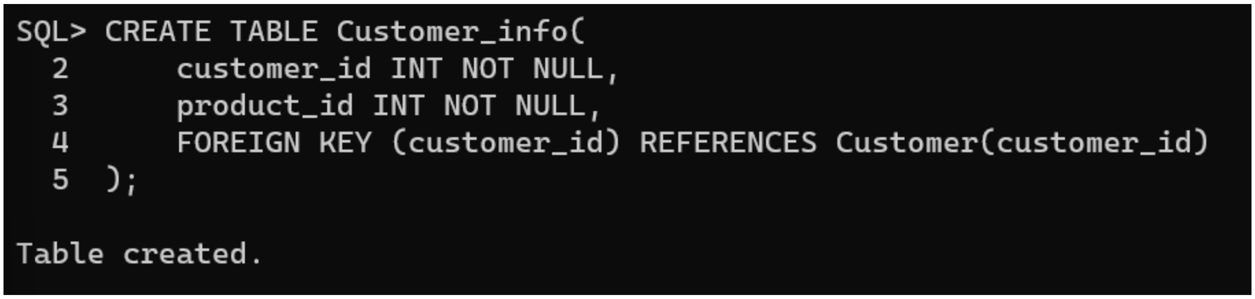
```
SQL> create table Vendor_product(
     vendor_id INT,
     Product_id INT,
     PRIMARY KEY (Vendor_id, Product_id),
     FOREIGN KEY (Vendor_id) REFERENCES Vendor(vendor_id),
     FOREIGN KEY (Product_id) REFERENCES Product(Product_id)
     );

Table created.
```

4.5 customer table creation



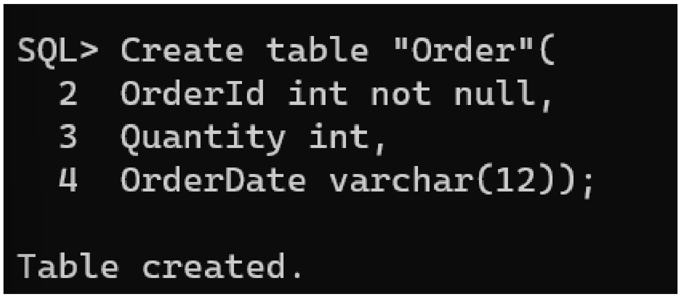
```
SQL> CREATE TABLE Customer(
     Customer_id INT PRIMARY KEY,
     Customer_name VARCHAR(32) NOT NULL,
     Customer_add VARCHAR(32) NOT NULL,
     Customer_contact INT NOT NULL);

Table created.

SQL> desc
Usage: DESCRIBE [schema.]object[@db_link]
SQL> desc Customer
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- --------------------
 CUSTOMER_ID                                           NOT NULL NUMBER(38)
 CUSTOMER_NAME                                         NOT NULL VARCHAR2(32)
 CUSTOMER_ADD                                          NOT NULL VARCHAR2(32)
 CUSTOMER_CONTACT                                      NOT NULL NUMBER(38)
```

4.6 Customer_info table creation



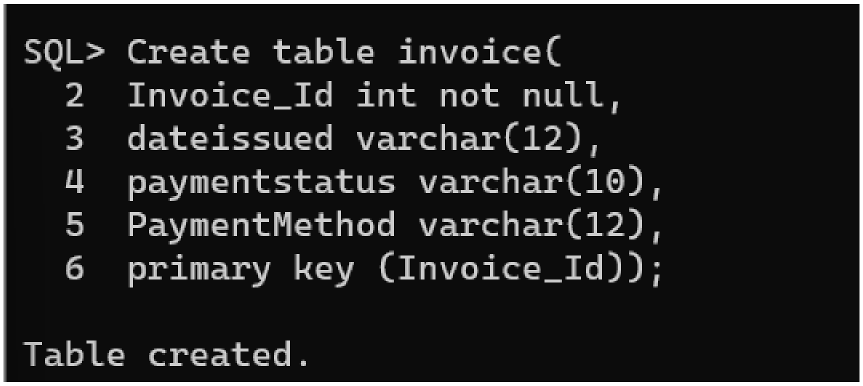
```
SQL> CREATE TABLE Customer_info(
         customer_id INT NOT NULL,
         product_id INT NOT NULL,
         FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
     );

Table created.
```

4.7 Order table creation



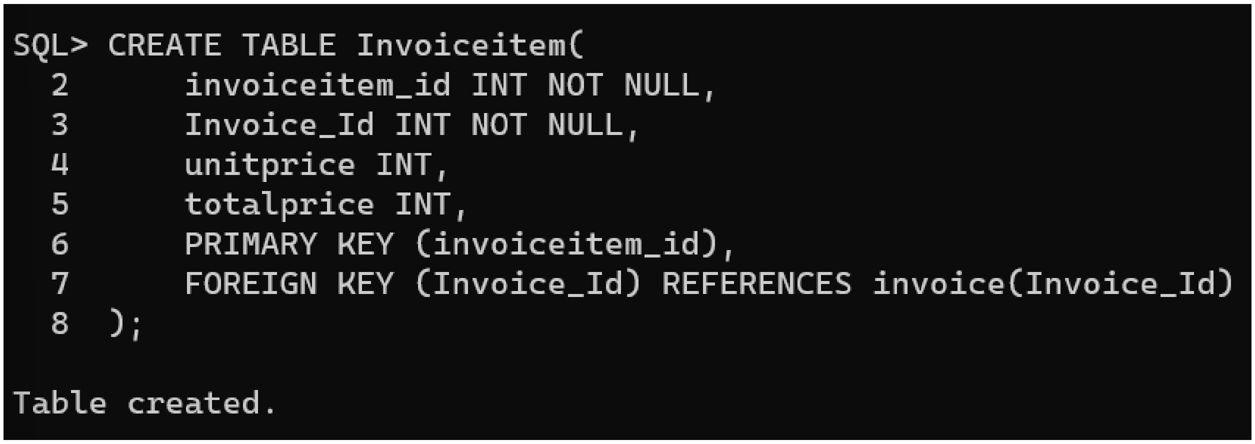
```
SQL> Create table "Order"(
     OrderId int not null,
     Quantity int,
     OrderDate varchar(12));

Table created.
```

4.8 Invoice table creation



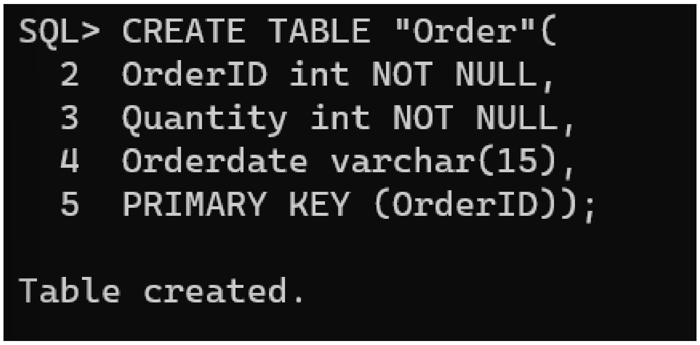
```
SQL> Create table invoice(
  2  Invoice_Id int not null,
  3  dateissued varchar(12),
  4  paymentstatus varchar(10),
  5  PaymentMethod varchar(12),
  6  primary key (Invoice_Id));

Table created.
```

4.9 Invoiceitem table creation



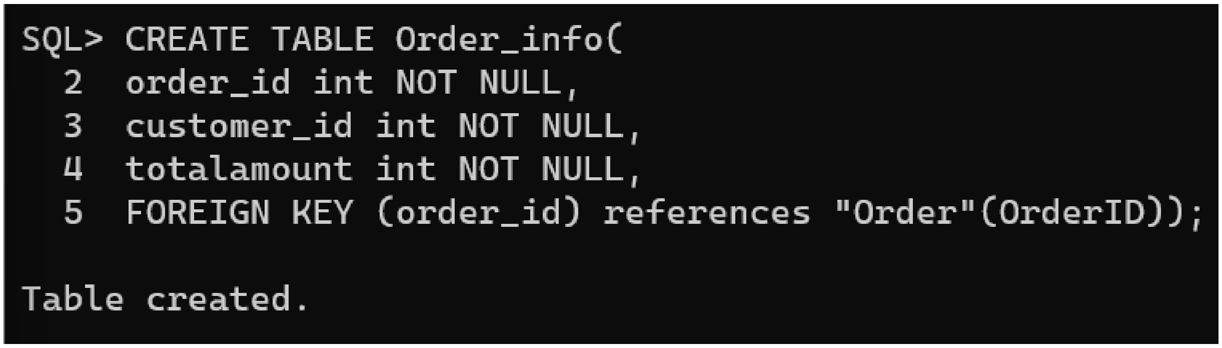
```
SQL> CREATE TABLE Invoiceitem(
  2      invoiceitem_id INT NOT NULL,
  3      Invoice_Id INT NOT NULL,
  4      unitprice INT,
  5      totalprice INT,
  6      PRIMARY KEY (invoiceitem_id),
  7      FOREIGN KEY (Invoice_Id) REFERENCES invoice(Invoice_Id)
  8  );

Table created.
```

4.01 “Order” table creation



```
SQL> CREATE TABLE "Order"(
  2  OrderID int NOT NULL,
  3  Quantity int NOT NULL,
  4  Orderdate varchar(15),
  5  PRIMARY KEY (OrderID));

Table created.
```

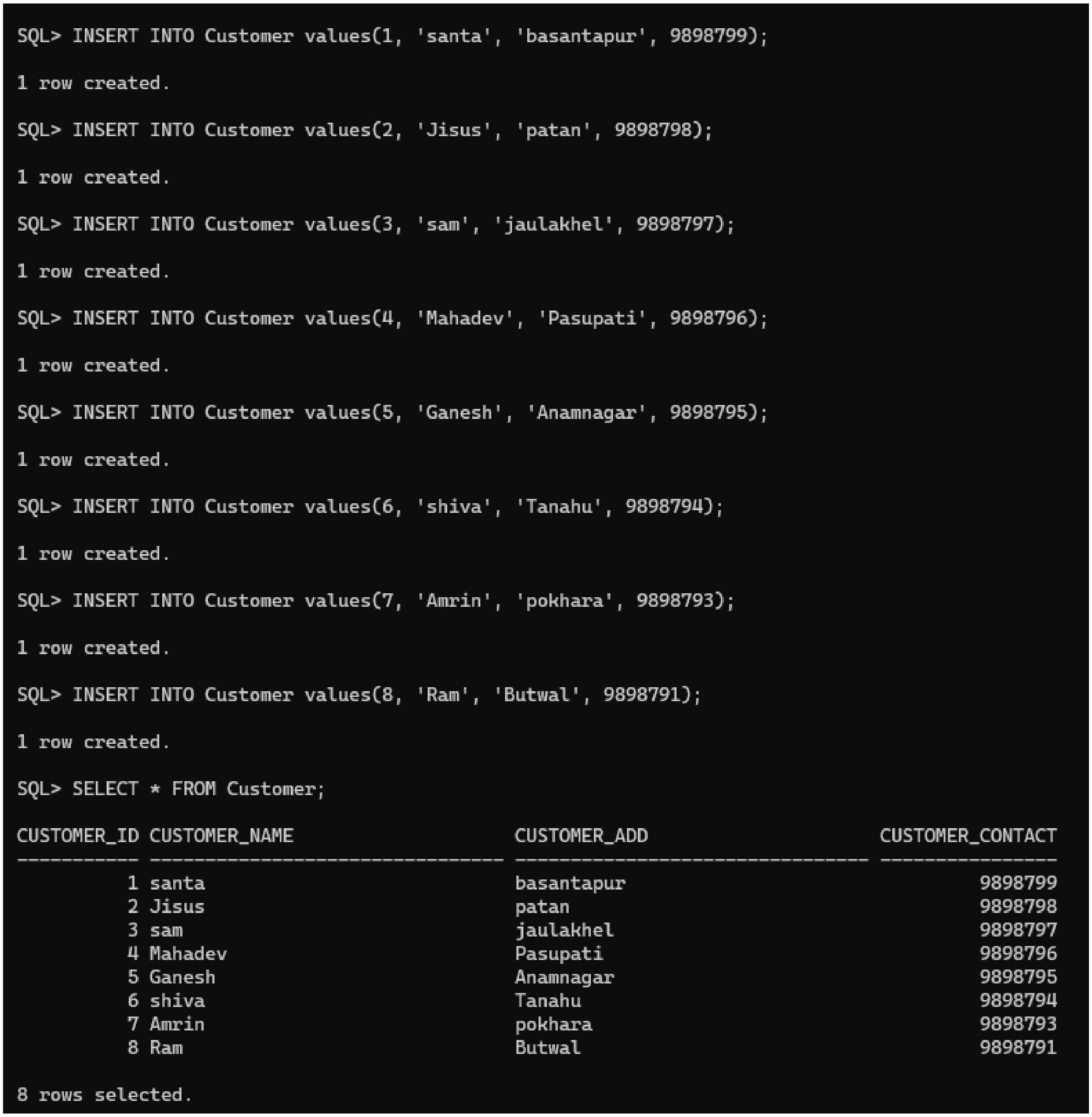
4.02 Order_info table creation



```
SQL> CREATE TABLE Order_info(
     order_id int NOT NULL,
     customer_id int NOT NULL,
     totalamount int NOT NULL,
     FOREIGN KEY (order_id) references "Order"(OrderID));

Table created.
```

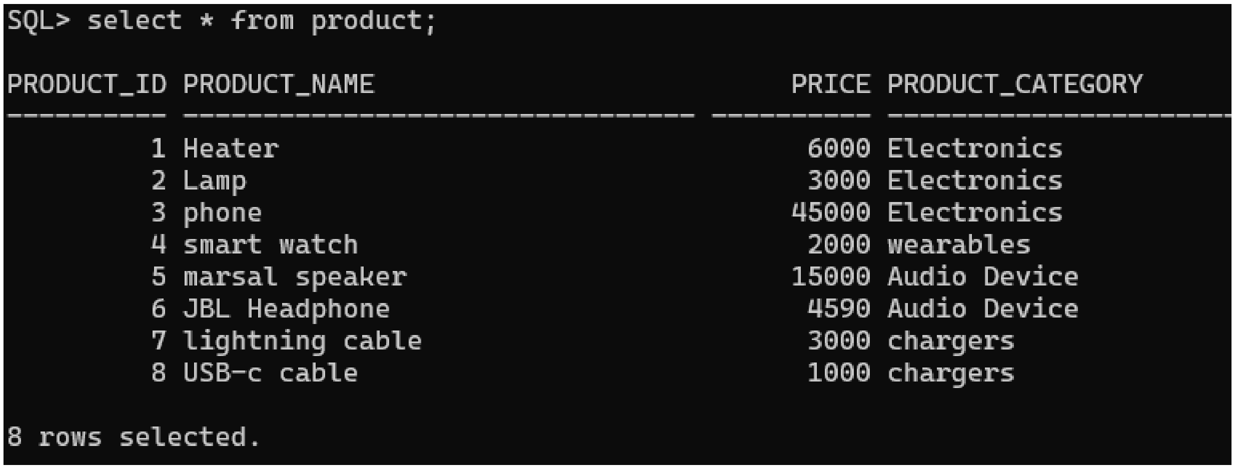
4.1.1 Data Insertion

i. Customer data insertion



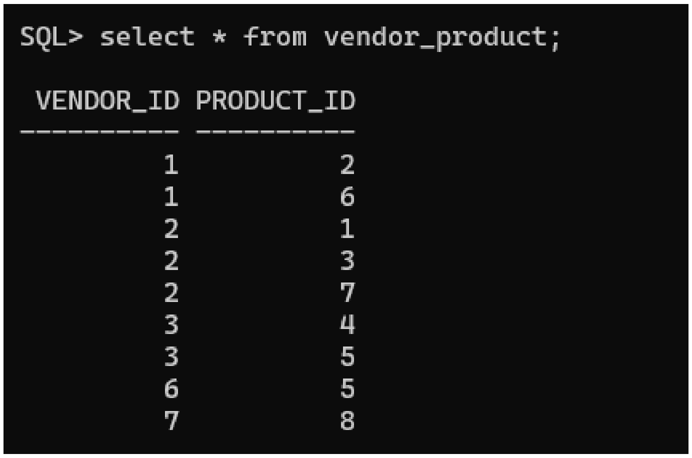
```
SQL> INSERT INTO Customer values(1, 'santa', 'basantapur', 9898799);

1 row created.

SQL> INSERT INTO Customer values(2, 'Jisus', 'patan', 9898798);

1 row created.

SQL> INSERT INTO Customer values(3, 'sam', 'jaulakhel', 9898797);

1 row created.

SQL> INSERT INTO Customer values(4, 'Mahadev', 'Pasupati', 9898796);

1 row created.

SQL> INSERT INTO Customer values(5, 'Ganesh', 'Anamnagar', 9898795);

1 row created.

SQL> INSERT INTO Customer values(6, 'shiva', 'Tanahu', 9898794);

1 row created.

SQL> INSERT INTO Customer values(7, 'Amrin', 'pokhara', 9898793);

1 row created.

SQL> INSERT INTO Customer values(8, 'Ram', 'Butwal', 9898791);

1 row created.

SQL> SELECT * FROM Customer;

CUSTOMER_ID CUSTOMER_NAME                      CUSTOMER_ADD                       CUSTOMER_CONTACT
----------- ---------------------------------- ---------------------------------- ----------------
          1 santa                              basantapur                                  9898799
          2 Jisus                              patan                                       9898798
          3 sam                                jaulakhel                                   9898797
          4 Mahadev                            Pasupati                                    9898796
          5 Ganesh                             Anamnagar                                   9898795
          6 shiva                              Tanahu                                      9898794
          7 Amrin                              pokhara                                     9898793
          8 Ram                                Butwal                                      9898791

8 rows selected.
```

ii. Product data insert



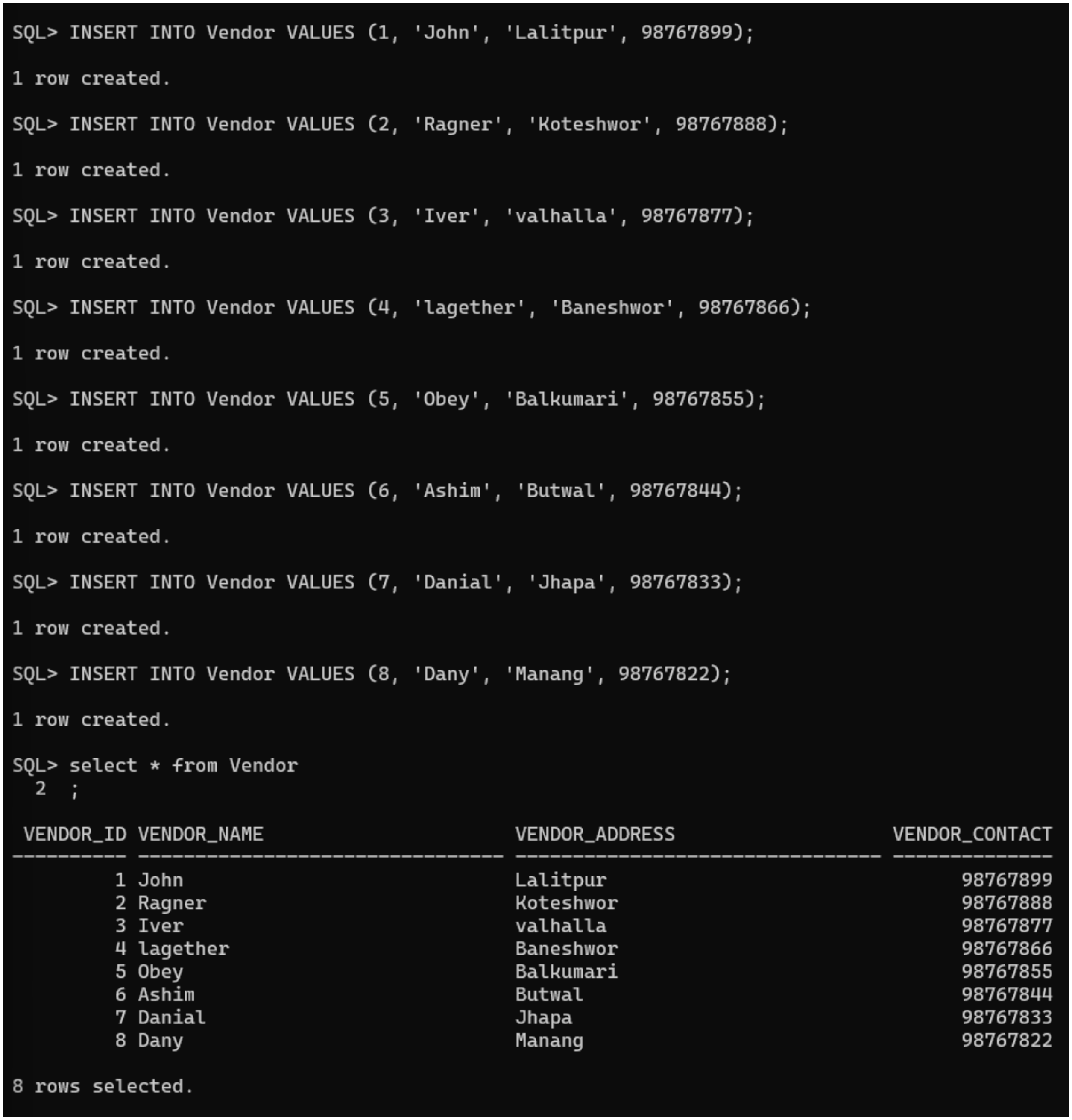
```
SQL> select * from product;

PRODUCT_ID PRODUCT_NAME                                  PRICE PRODUCT_CATEGORY
---------- -------------------------------- ---------- ----------------------
         1 Heater                                         6000 Electronics
         2 Lamp                                           3000 Electronics
         3 phone                                         45000 Electronics
         4 smart watch                                    2000 wearables
         5 marsal speaker                                15000 Audio Device
         6 JBL Headphone                                  4590 Audio Device
         7 lightning cable                                3000 chargers
         8 USB-c cable                                    1000 chargers

8 rows selected.
```

iii. Vendor_product



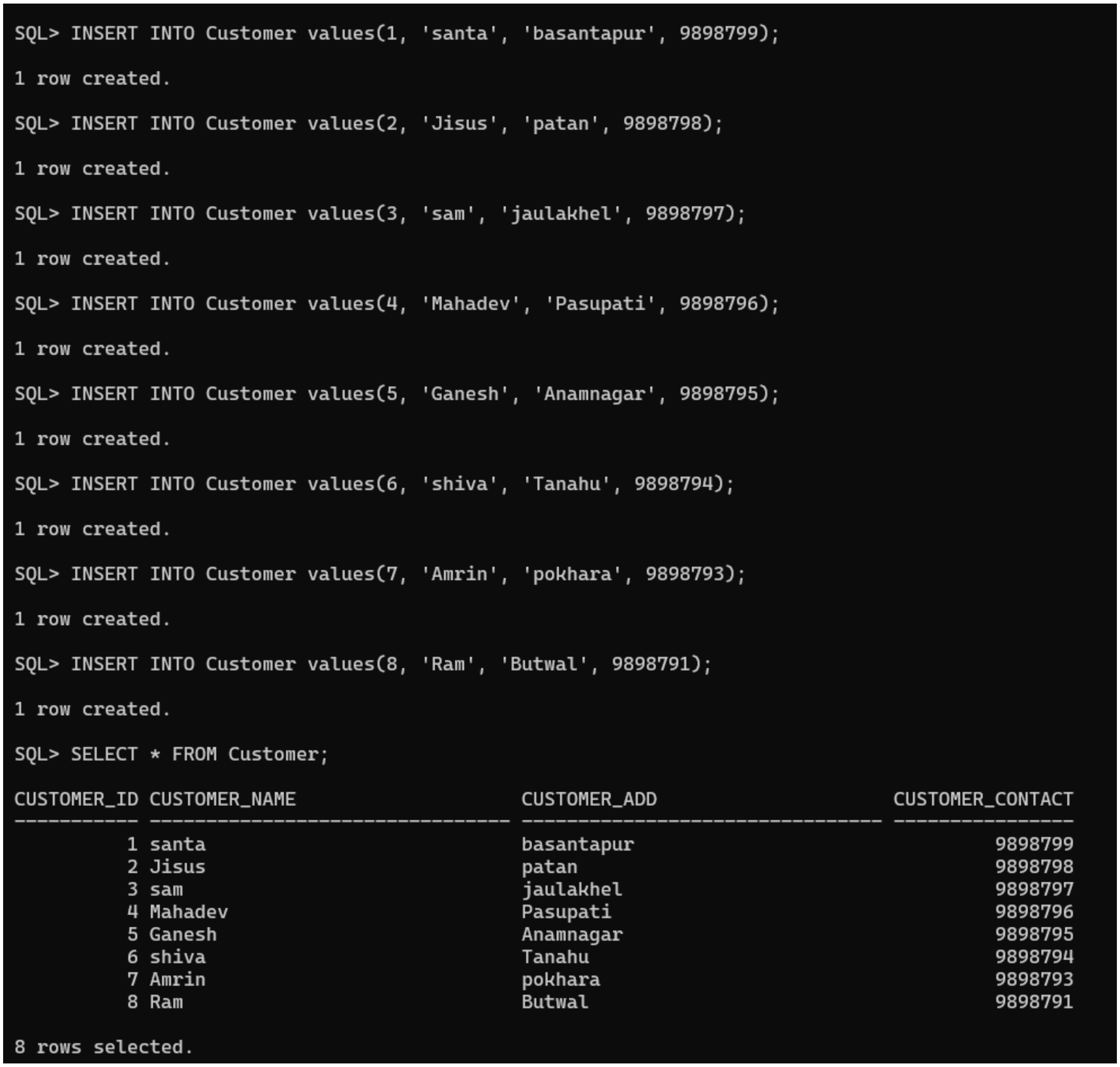
```
SQL> select * from vendor_product;

 VENDOR_ID PRODUCT_ID
---------- ----------
         1          2
         1          6
         2          1
         2          3
         2          7
         3          4
         3          5
         6          5
         7          8
```

iv. Vendor data insert



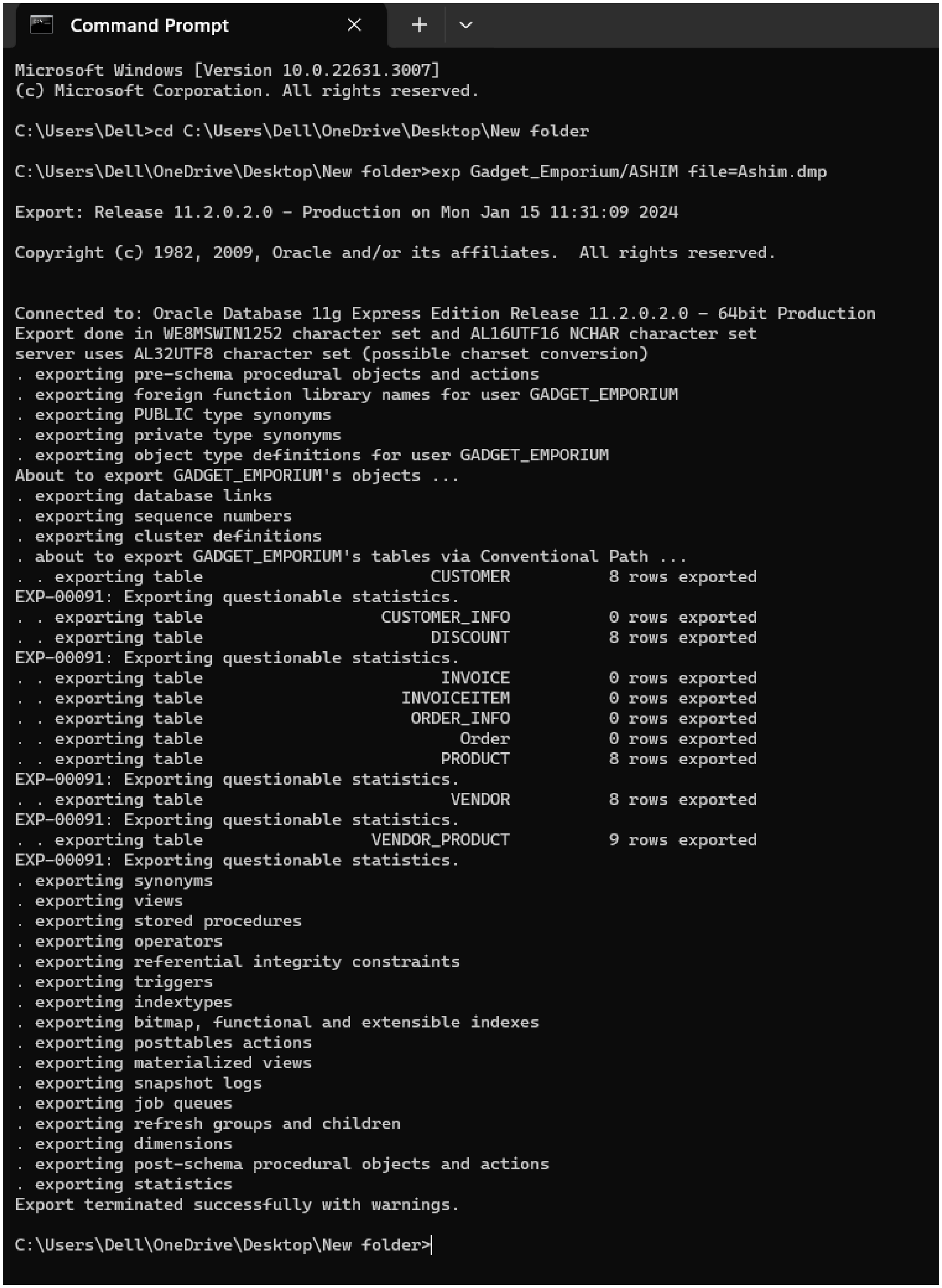
```
SQL> INSERT INTO Vendor VALUES (1, 'John', 'Lalitpur', 98767899);

1 row created.

SQL> INSERT INTO Vendor VALUES (2, 'Ragner', 'Koteshwor', 98767888);

1 row created.

SQL> INSERT INTO Vendor VALUES (3, 'Iver', 'valhalla', 98767877);

1 row created.

SQL> INSERT INTO Vendor VALUES (4, 'lagether', 'Baneshwor', 98767866);

1 row created.

SQL> INSERT INTO Vendor VALUES (5, 'Obey', 'Balkumari', 98767855);

1 row created.

SQL> INSERT INTO Vendor VALUES (6, 'Ashim', 'Butwal', 98767844);

1 row created.

SQL> INSERT INTO Vendor VALUES (7, 'Danial', 'Jhapa', 98767833);

1 row created.

SQL> INSERT INTO Vendor VALUES (8, 'Dany', 'Manang', 98767822);

1 row created.

SQL> select * from Vendor
  2  ;

 VENDOR_ID VENDOR_NAME                      VENDOR_ADDRESS                   VENDOR_CONTACT
---------- -------------------------------- -------------------------------- --------------
         1 John                             Lalitpur                               98767899
         2 Ragner                           Koteshwor                              98767888
         3 Iver                             valhalla                               98767877
         4 lagether                         Baneshwor                              98767866
         5 Obey                             Balkumari                              98767855
         6 Ashim                            Butwal                                 98767844
         7 Danial                           Jhapa                                  98767833
         8 Dany                             Manang                                 98767822

8 rows selected.
```

v. customer data insert

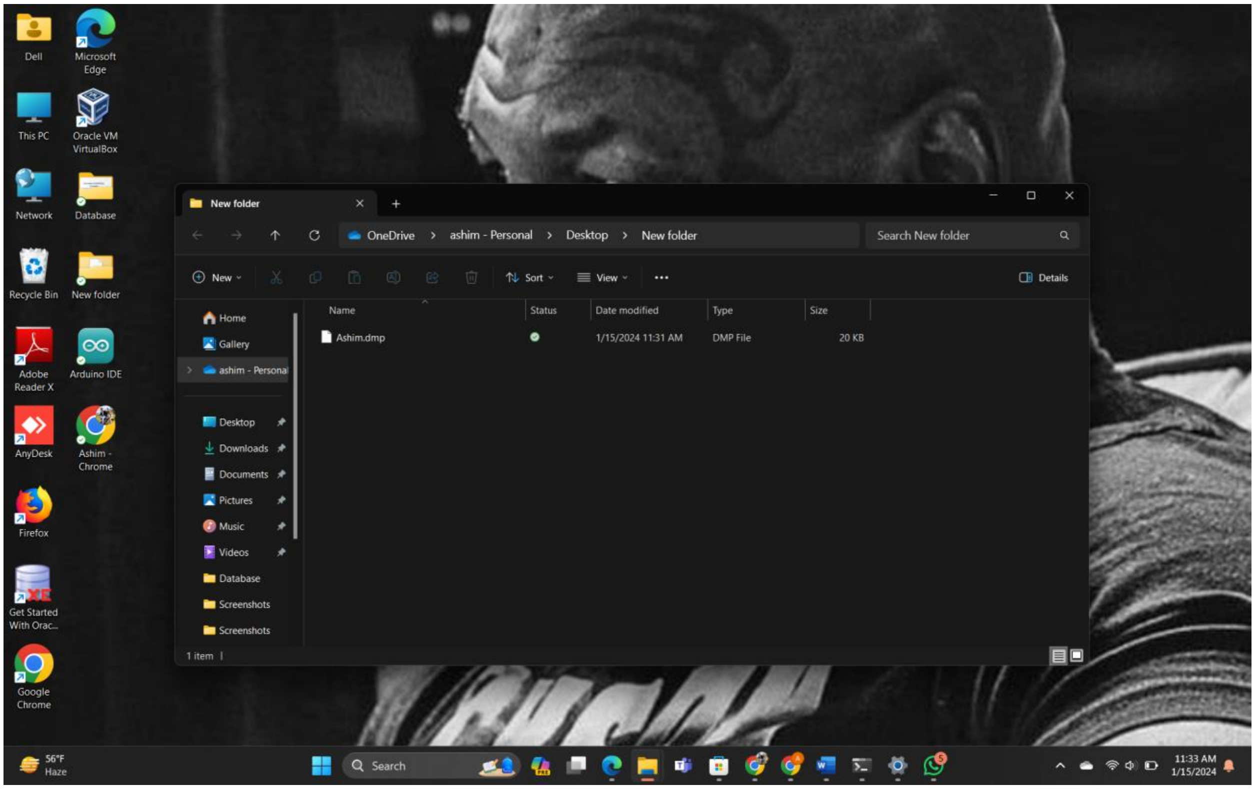


```
SQL> INSERT INTO Customer values(1, 'santa', 'basantapur', 9898799);

1 row created.

SQL> INSERT INTO Customer values(2, 'Jisus', 'patan', 9898798);

1 row created.

SQL> INSERT INTO Customer values(3, 'sam', 'jaulakhel', 9898797);

1 row created.

SQL> INSERT INTO Customer values(4, 'Mahadev', 'Pasupati', 9898796);

1 row created.

SQL> INSERT INTO Customer values(5, 'Ganesh', 'Anamnagar', 9898795);

1 row created.

SQL> INSERT INTO Customer values(6, 'shiva', 'Tanahu', 9898794);

1 row created.

SQL> INSERT INTO Customer values(7, 'Amrin', 'pokhara', 9898793);

1 row created.

SQL> INSERT INTO Customer values(8, 'Ram', 'Butwal', 9898791);

1 row created.

SQL> SELECT * FROM Customer;

CUSTOMER_ID CUSTOMER_NAME                    CUSTOMER_ADD                     CUSTOMER_CONTACT
----------- -------------------------------- -------------------------------- ----------------
          1 santa                            basantapur                               9898799
          2 Jisus                            patan                                    9898798
          3 sam                              jaulakhel                                9898797
          4 Mahadev                          Pasupati                                 9898796
          5 Ganesh                           Anamnagar                                9898795
          6 shiva                            Tanahu                                   9898794
          7 Amrin                            pokhara                                  9898793
          8 Ram                              Butwal                                   9898791

8 rows selected.
```

5.0 Dump file



```
Microsoft Windows [Version 10.0.22631.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dell>cd C:\Users\Dell\OneDrive\Desktop\New folder

C:\Users\Dell\OneDrive\Desktop\New folder>exp Gadget_Emporium/ASHIM file=Ashim.dmp

Export: Release 11.2.0.2.0 - Production on Mon Jan 15 11:31:09 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 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 table                       CUSTOMER          8 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                  CUSTOMER_INFO          0 rows exported
. . exporting table                       DISCOUNT          8 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                        INVOICE          0 rows exported
. . exporting table                    INVOICEITEM          0 rows exported
. . exporting table                     ORDER_INFO          0 rows exported
. . exporting table                          Order          0 rows exported
. . exporting table                        PRODUCT          8 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                         VENDOR          8 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                 VENDOR_PRODUCT          9 rows exported
EXP-00091: Exporting questionable statistics.
. 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
Export terminated successfully with warnings.

C:\Users\Dell\OneDrive\Desktop\New folder>
```



OneDrive > ashim - Personal > Desktop > New folder

Name	Status	Date modified	Type	Size
Ashim.dmp	●	1/15/2024 11:31 AM	DMP File	20 KB

6.0 Critical Evaluation

Additional talk on the learning process

I've never completed any assignment related to the database module. The creation of a database for data storage was the main goal of this course. I had trouble breaking down the case scenario. I had to go to a few websites to learn more about the different firms. Determining the entities and their characteristics was the most challenging part of this course. I worked with the instructors on my subject to construct entities and attributes after finishing my studies.

I found this course to be difficult. I couldn't utilize databases, even though I knew about them. I was having trouble understanding the scenario; thus, I found this assignment to be quite challenging and informative. My search for an existing database for this purpose and design was somewhat difficult. Characterizing the problem, creating entities and characteristics, adding data, normalizing the data, and executing SQL queries were some of the challenging tasks. I never believed that I would be able to accomplish the assignment in the allowed time.

My buddies and the module leaders help me get through the challenges. I went over all the information I could find online, and it was incredibly beneficial for me to do my homework. My participation in the project and my ongoing study sessions enabled me to complete all my assignments for this semester on time. This program has provided me with a wealth of information. I also learned how to create and edit database entries and columns using SQL queries. Furthermore, this effort has improved my understanding of normalization and its significance.

7.0 Critical Assessment

The proposed Gadget Emporium database system demonstrates significant capabilities in essential areas such as normalized data formats and real-time inventory management, providing a solid foundation for efficient electronic goods administration. However, there are many shortcomings, chief among which being the disdain for user identity and security. Due to the possible risks this error poses, it emphasizes the need for robust security measures. Furthermore, a comprehensive data backup plan and unambiguous order tracking systems could negatively impact customer satisfaction and raise the risk of data loss. To build a reliable, user-focused, and safe e-commerce platform—all essential for Gadget Emporium to thrive in the competitive online market—these problems must be resolved.

8 Conclusion

In conclusion, to support Mr. John's "Gadget Emporium" e-commerce business, a complete database system that can effectively manage customer information, product inventory, order processing, and tracking must be built. The database needs to be scalable to handle a wide variety of items and an expanding client base. This will guarantee that the company can accommodate the expected rise in data flow as it grows.

Robust security measures to safeguard sensitive data, an easy-to-use interface, and customization possibilities to customize the database to the unique requirements of an online marketplace selling electronics are all necessary for the system to securely store client data. The database should be set up with several tables that record all pertinent product information, such as categories, brands, and inventory levels, to facilitate the administration of product inventories. It should make interacting with the order processing system easy and protect data integrity by using linkages and limits that are well-defined.

A centralized platform that provides rapid access to order-related data for several business divisions is necessary for the order processing and tracking system. An advanced interface with accounting and invoicing software is required to handle order-related duties efficiently. Assuring continuous improvement and monitoring key performance metrics also need clever analytics. The links between orders, items, and consumers need to be properly considered in the database. One-to-many linkages between consumers and their orders and many-to-many ties between items and orders will be established using a relational database management system (RDBMS). In this procedure, a junction table for order items may also be utilized. This makes it possible to trace a customer's purchases of goods indirectly.

By focusing on these essential components of order processing, inventory management, and customer data management along with relational dynamics in database design, Mr. John's "Gadget Emporium" will be well-positioned with a strong database system that can withstand the demands of an online retail environment that facilitates transactions for both individual customers and corporate entities.