Module Code & Module Title

CC5051NI Databases

Assessment Weightage & Type

50% Individual Coursework

Year and Semester

2021 Autumn

Student Name: Asal Pandey

London Met ID: 20049387

College ID: NP01CP4S210181

Assignment Due Date: 2021/12/28

Assignment Submission Date: 2021/12/24

Title (Where Required): Databases

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 marks of zero will be awarded.

Contents

Table of figures

INTRODUCTION

Mr Steve, who is a local businessman and a real estate owner owns a company named "masterpieces limited" which is an online platform focusing on renting paintings made by different artists and creators. The objective of the business is to lease the paintings to private individuals and commercial companies owned by masterpieces, as well as different painters or artists.

The business will have paintings owned by the company for rent or lease to the customers. And it will also be a platform for the painters to provide their paintings to the company, to put them on display for rental purposes by the customers. The business allows the customers to lease or buy the paintings of the painters. Steve wants to indirectly connect the painter and his clients with his business as a broker.

The Paintings will be put on a display based on their theme such as portraits, abstracts, scenery and so on.

In this business, Steve aims to help painters and artists gain recognition for their art and creativity and connect them to the customers who are willing to lease the painting paying a certain amount.

Current Business Activities and Operation

- The business provides its customers with paintings for lease or for buying created by different local and renowned painters as well as paintings owned by the business.
- The business runs for 8 hours per day starting from 9 in the morning to 5 in the evening.
- Holidays are given once a week(Saturday),
- The paintings are delivered to the people who purchase them or who lease them within 2 days of the transaction.
- All customers, staff, paintings and painters information like names, phone numbers, etc are recorded in the database
- Various discounts policies are allocated to the customers based on their sectioned categories ranging from 0% to 15%
- Each painting is entitled with its own cost as a monthly rental fee
- Individuals, as well as companies, are entitled to buy the paintings
- The paintings are returned to the painter if they are not bought or purchased by the customers within 4 months of the painting being issued.
- The returned paintings, which are returned to the painters can again be to be put on display after one month of duration
- The business takes 80% of the total price of the painting from the customer after the transaction and 20% is given to the painter.
- Each painting can only have one artist associated with its customers who can rent the painting at a certain monthly price.

Business rules

- One staff can sell one or more paintings at a time.
- a painting cannot be sold by two staff of the company
- A customer is allowed to lease or buy multiple pantings at a time
- A painting cannot be rented by two customers at the same time
- A staff can sell or rent multiple paintings at a time
- Painters can put multiple paintings for lease or sales
- Customers are allowed to lease the same painting twice.
- Painters can lease their paintings.
- Discounts are given to the customers according to their category.
- Owner/artists/ painters can have multiple paintings assigned to the company for renting.

ERD
Identification of Entities and attributes

Entity	Attribute	Data Type	Constrain
Painting	Painting_ID	VARCHAR	P.K
	Painting_Name	VARCHAR	
	Selling_Price	INTEGER	
	Lease_price	INTEGER	
	Registration_Date	DATE	
	Theme	VARCHAR	
	Painter_ID	VARCHAR	
	Painter_Name	VARCHAR	
	Painter_Contact	VARCHAR	
	Order_Date	DATE	
	Order_Type	VARCHAR	
	Return_Date	DATE	
	Order_ID	VARCHAR	
Staff	Painting_ID	VARCHAR	F.K
	Staff_ID	VARCHAR	P.K
	Staff_Name	VARCHAR	
	Staff_Address	VARCHAR	
	Staff_Contact	VARCHAR	
	Job	VARCHAR	
	Salary	VARCHAR	
Customer	Customer_ID	VARCHAR	P.K
	Painting_ID	VARCHAR	F.K
	Customer_Name	VARCHAR	
	Customer_Contact	VARCHAR	
	Customer_Address	VARCHAR	

	Category	VARCHAR	

Initial Erd

Entity-Relationship-Model uses ERdiagram to describe the structure of the database simply. ERdiagram shows the logical structure of the entire database and shows the relationships of entities between tables. ERModel is also called a database blueprint

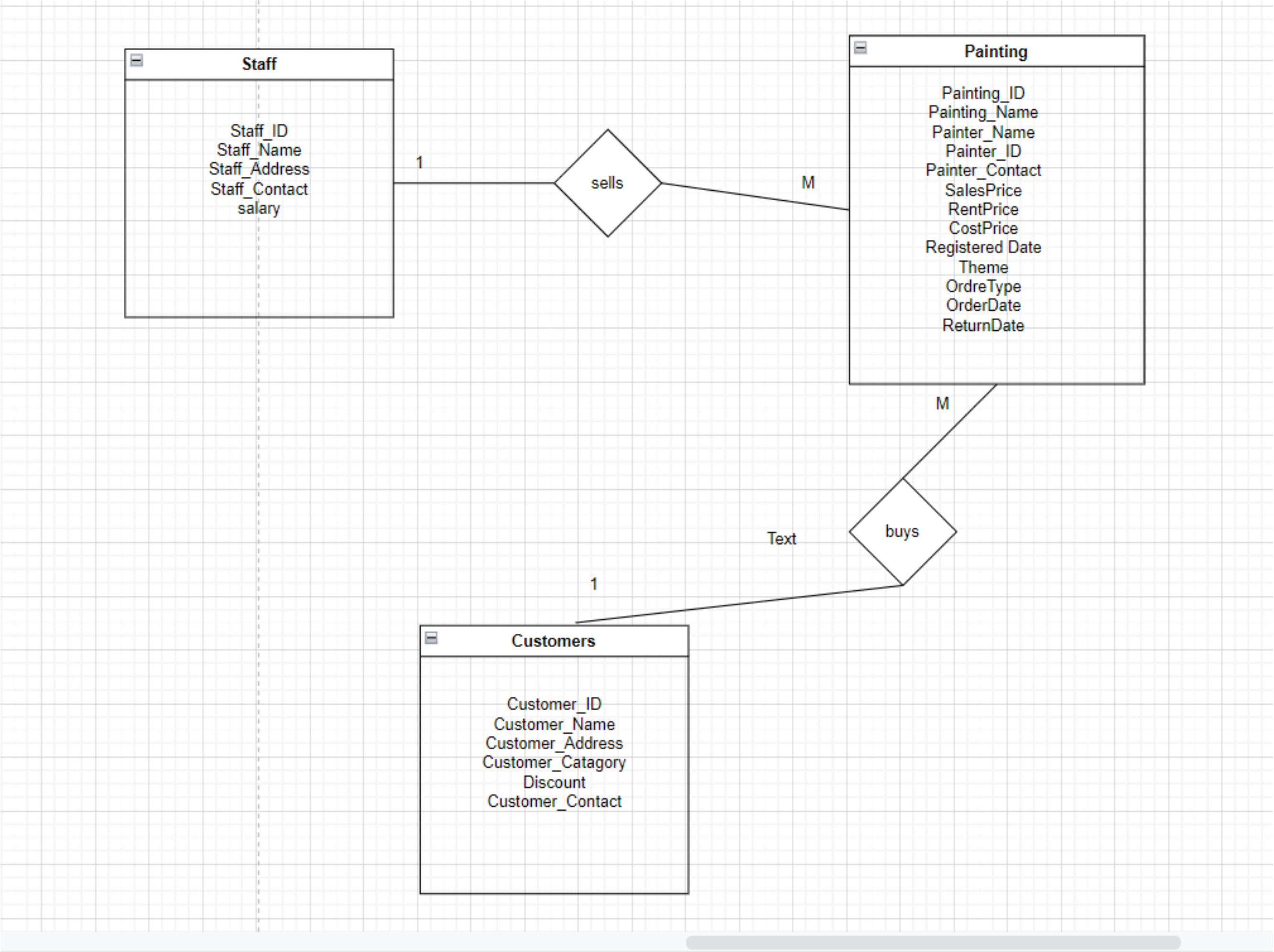


Figure 1: initial ERD

Normalization for Coursework:

Normalization

Normalization is a database design technique that reduces data redundancy and eliminates insertion, updates and deletion anomalies. Normalization rules divide large tables into smaller tables and use relationships to link them (Peterson, 2021). We require to use normalization in our database to eliminate redundant data and allow the data to be stored logically.

Un-Normalized Form (UNF)

It is the simplest database model also known as non-first normal form.it is an unordered form of the database stored in a single entity A UNF model will suffer problems like data redundancy thus it lacks the efficiency of database normalization. (GEEKS FOR GEEKS, 2020)

Showing repeating groups

Order (Order_ID (PK), Order_Date, Return_Date, {Customer_Id, Customer_Name, Customer_Address, Customer_Category, Customer_Discount, Customer_Contact}, {Staff_ID, Staff_Name, Staff_Address, Staff_Contact, Staff_Age, Salary, {Painting_ID, Painting_Name, Painter_Name, Painter_ID, Painter_Contact, SalesPrice, RentPrice, CostPrice, InvoiceDate, Theme, OrderType}})

First Normal Form (1NF)

For the 1NF the repeating group are separated into different entities from UNF, and a composite key is formed.

Order-1 (Order_ID (PK), OrderDate, ReturnDate, Customer_Id (PK), Customer_Name, Customer_Address, Customer_Category, Customer_Discount, Customer_Contact, Staff_ID, Staff_Name, Staff_Address, Staff_Contact, Staff_Age, Salary)

Painting-1 (Painting_ID (PK), Painting_Name, Painter_ID (PK), Painter_Name, Painter_ContactNo, SalesPrice, RentPrice, CostPrice, InvoiceDate, Theme, OrderType, Order_ID (FK))

Second Normal Form (2NF)

for the 1NF, the composite key is determined and in 2NF, partial dependency is checked and removed.

Order-2 (Order_ID, OrderDate, ReturnDate, Customer_Id, Customer_Name, Customer_Address, Customer_Category, Customer_Discount, Customer_Contact, Staff_ID, Staff_Name, Staff_Address, Staff_Contact, Customer_Age, Salary)

Painting-1 (Painting_ID, Painting_Name, Painter_ID, Painter_Name, Painter_Contact, SalesPrice, RentPrice, CostPrice, InvoiceDate, Theme, OrderType, Order_ID*)

Checking partial dependency in painting table

PaintingID →PaintingName, ArtistID, ArtistName, ArtistContactNo, SalesPrice, RentPrice, InvoiceDate, Theme, PaintingID, OrderID → CostPrice, OrderType

The tables in 2NF are as follows:

Order-2 (OrderID, OrderDate, ReturnDate, CustomerId, CustomerName, CustomerAddress, CustomerCategory, Discount, CustomerContactNo, StaffID, StaffName, StaffAddress, StaffContactNo, Salary)

Painting-2 (PaintingID, PaintingName, ArtistID, ArtistName, ArtistContactNo, SalesPrice, RentPrice, InvoiceDate, Theme)

Painting-Order-2 (PaintingID*, OrderID*, CostPrice, OrderType)

Order – 1 = (Order_ID, Order_Date, Order_Returned, Customer_ID, Customer_Name, Customer_Address, Category, Customer_Contact, Staff _ID, Staff _Name, Staff _Address, Staff _Contact, Salary)
Painting – 1 = (Painting_ID, Painting_Name, Artist_ID, Artist_Name, Selling_Price, Rent_Price, Paid_Price, Registered_Date, Theme, Order_Type, Order_ID)

Order – 2 = (Order_ID, OrderDate, ReturnDate, Customer_ID, Customer_Name, Customer_Address, Category, Category_Discount, Customer_Contact, Staff _ID, Staff _Name, Staff _Address, Staff _Contact, Salary)
Painting – 1 = (Painting_ID, Painting_Name, Artist_ID, Artist_Name, SalesPrice, RentPrice, CostPrice, Registered_Date, Theme, OrderType, Order_ID*)
Painting_ID -> Painting_Name, Painter_ID, Painter_Name, Painter_Contact, SalesPrice, RentPrice, Registered_Date, Theme, Painting_ID, Order_ID - Cost_Price, Order_Type
Order – 2 = (Order_ID, OrderDate, ReturnDate, Customer_ID, Customer_Name, Customer_Address, Customer_Category, Customer_Discount, Customer_Contact, Saff_ID, Staff_Name, Staff_Address, Staff_Contact, Salary)
Painting – 2 = (Painting_ID, Painting_Name, Artist_ID, Artist_Name, Artist_Contact, Selling_Price, Rent_Price, Registered_Date, Theme)
Painting-Order – 2 = (Painting_ID, Order_ID*, Cost_Price, Order_Type)
Order – 2 = (Order_ID, Order_Date, ReturnDate, Customer_ID, Customer_Name, Customer_Address, Customer_Category, Customer_Discount, Customer_Contact, Staff _ID, Staff _Name, Staff _Address, Staff _Contact, Salary)
Order_ID -> Customer_ID -> Customer_Name, Customer_Address, Category, Customer_Contact
Order_ID -> Staff_ID -> Staff _Name, Staff _Address, Staff _Contact, Salary
Order_ID -> Order_Date, Order_Returned,
Order – 3 = (Order_ID, Customer_ID, Staff_ID, OrderDate, ReturnDate)
Customer – 3 = (Customer_ID, Customer_Name, Customer_Address, Category,

Customer_Contact)
Employees – 3 = (Staff_ID, Staff _Name, Staff _Address, Staff _Contact, Salary)
Painting – 2 = (Painting_ID, Painting_Name, Painter_ID, Painter_Name, Painter_Contact, SalesPrice, RentPrice, Registered_Date, Theme)
Painting_ID > Painting_Name > Registered_Date > SalesPrice > RentPrice >Theme
Painting_ID >Painter_ID > Painter_Name, Painter_Contact

Order - 3 = (Order_ID, Customer_ID, Staff_ID, OrderDate, ReturnDate)
Customer – 3 = (Customer_ID, Customer_Name, Customer_Address, Customer_Category, Customers_Discount, Customer_Contact)
Employees – 3 = (Staff_ID, Staff_Name, Staff_Address, Staff_Contact, Salary)
Painting – 3 = (Painting_ID, Painting_Name, Registered_Date, SalesPrice, RentPrice, Theme, Artist_ID)
Artist – 3 = (Painter_ID, Painter_Name, Painter_Contact)
Painting-Order – 3 = (Painting_ID, Order_ID, Cost_Price, Order_Type)

Third Normal Form (3NF)

By the time 3NF is reached, most of the anomalies and redundancies are reduced. The data must be in 2NF to create 3NF.

Order-2 (Order_ID, OrderDate, ReturnDate, Customer_Id, Customer_Name, Customer_Address, Customer_Category, Customer_Discount, Customer_Contact, Staff_ID, Staff_Name, Staff_Address, Staff_Contact, Salary)

Checking transitive dependency in order table:

Order_ID → Customer_Id → Customer_Name, Customer_Address, Customer_Category, Customer_Discount, Customer_Contact Order_ID → Staff_ID → Staff_Name, Staff_Address, Staff_Contact, Salary
OrderID → OrderDate, ReturnDate,

Order-3 (OrderID, CustomerID*, StaffID*, OrderDate, ReturnDate)

Customer-3 (Customer_Id, Customer_Name, Customer_Address, Customer_Category, Customer_Discount, Customer_Contact)

Staff-3 (Staff_ID, Staff_Name, Staff_Address, Staff_ContactNo, Salary)

Again, checking transitive dependency in painting2:

Painting-2 (Painting_ID, Painting_Name, Painter_ID, Painter_Name, Painter_Address, SalesPrice, RentPrice, Registered_Date, Theme,)

PaintingID → Painting_Name → Registered_Date → SalesPrice → RentPrice→Theme → Painter_ID→Painter_Name

Since there is no transitive dependency in table Painting-Order-2 we know that these entities are in 3NF

Painting-Order-3 (Painting_ID*, Order_ID*, CostPrice, OrderType)

The entities after 3NF are

Order-3 (**OrderID, CustomerId, StaffID**, OrderDate, ReturnDate)

Customer-3 (**Customer_ID**, Customer_Name, Customer_Address, Customer_Category, Customer_Discount, Customer_Contact)

Staff-3 (**Staff_ID**, Staff_Name, Staff_Address, Staff_ContactNo, Salary)

Painting-3 (**Painting_ID**, Painting_Name, Registered_Date, SalesPrice, RentPrice, Theme, Painter_ID)

Painter-3 (**Painter_ID**, Painter_Name, Painter_Contact)

Painting-Order-3 (**Painting_ID, Order_ID**, CostPrice, OrderType)

For the last step, the UNF is changed to 3NF removing all anomalies and transitive dependencies

Final ERD

Final ERD is obtained after 3NF is completed in the normalization process. Data anomalies and redundancy in the tables and their attributes are solved in this process. The normalization process of the entire table made the database more systematic and effective.

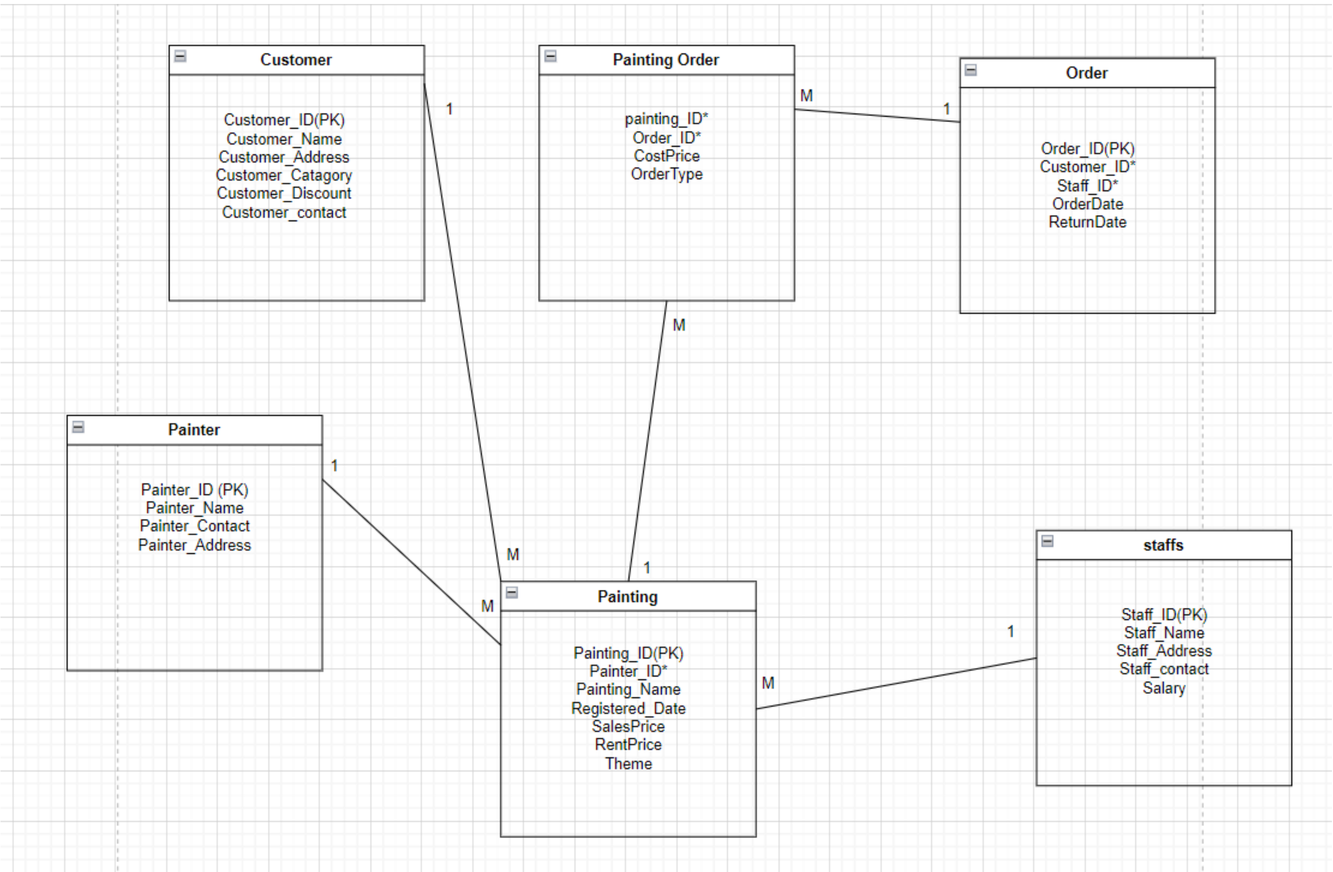
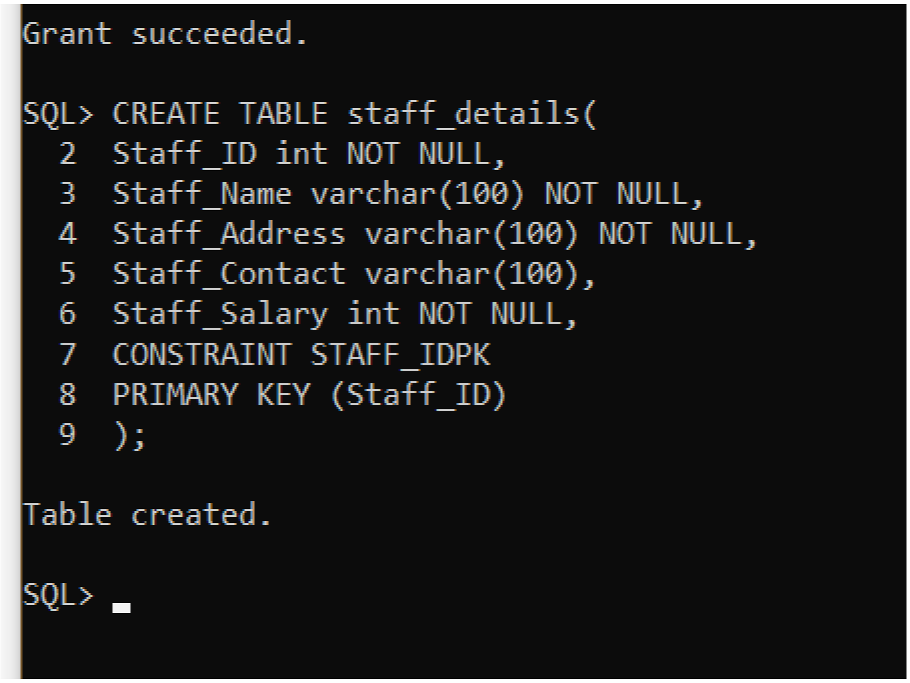


Figure 2: final erd

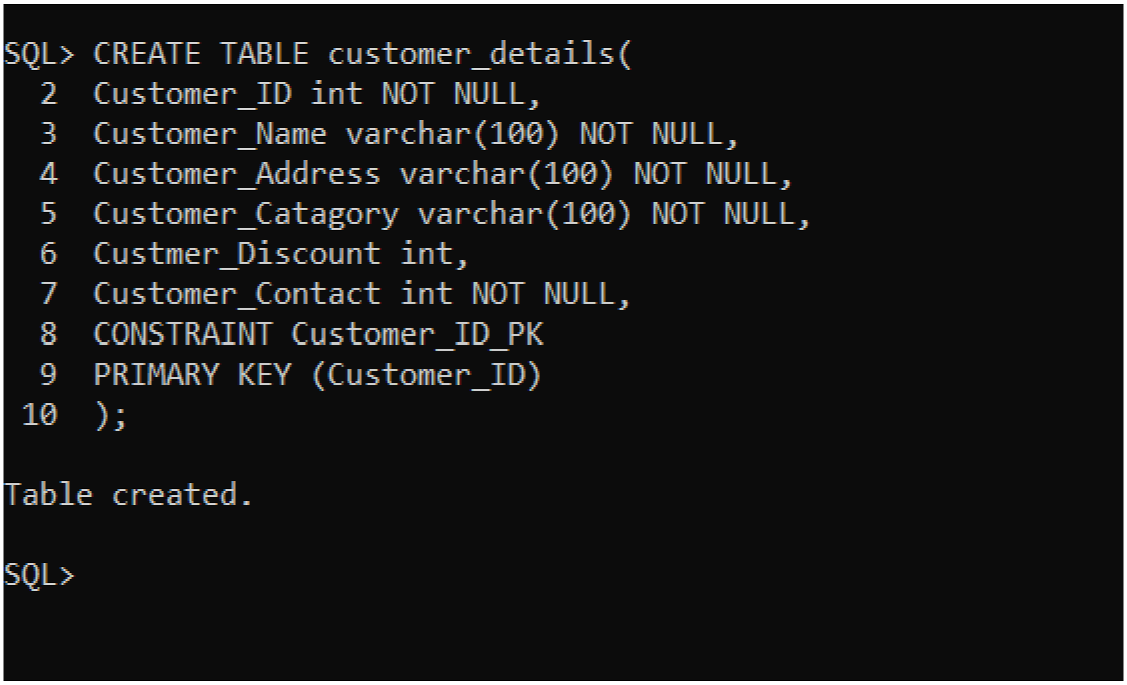
Table creations

Creating the table staff



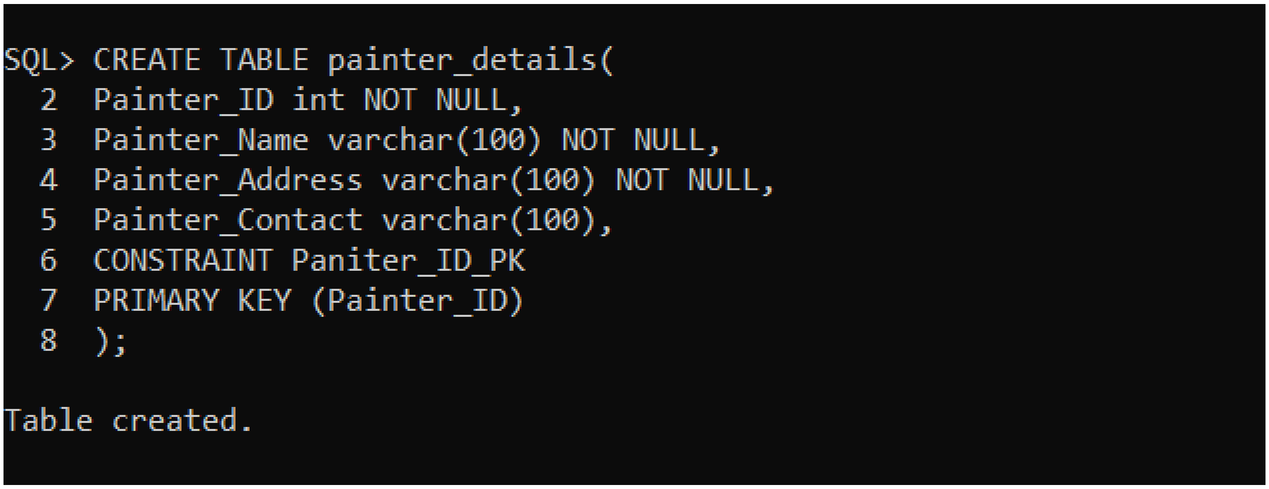
```
Grant succeeded.

SQL> CREATE TABLE staff_details(
     Staff_ID int NOT NULL,
     Staff_Name varchar(100) NOT NULL,
     Staff_Address varchar(100) NOT NULL,
     Staff_Contact varchar(100),
     Staff_Salary int NOT NULL,
     CONSTRAINT STAFF_IDPK
     PRIMARY KEY (Staff_ID)
     );

Table created.

SQL> _
```

Figure 3: table creation staff details



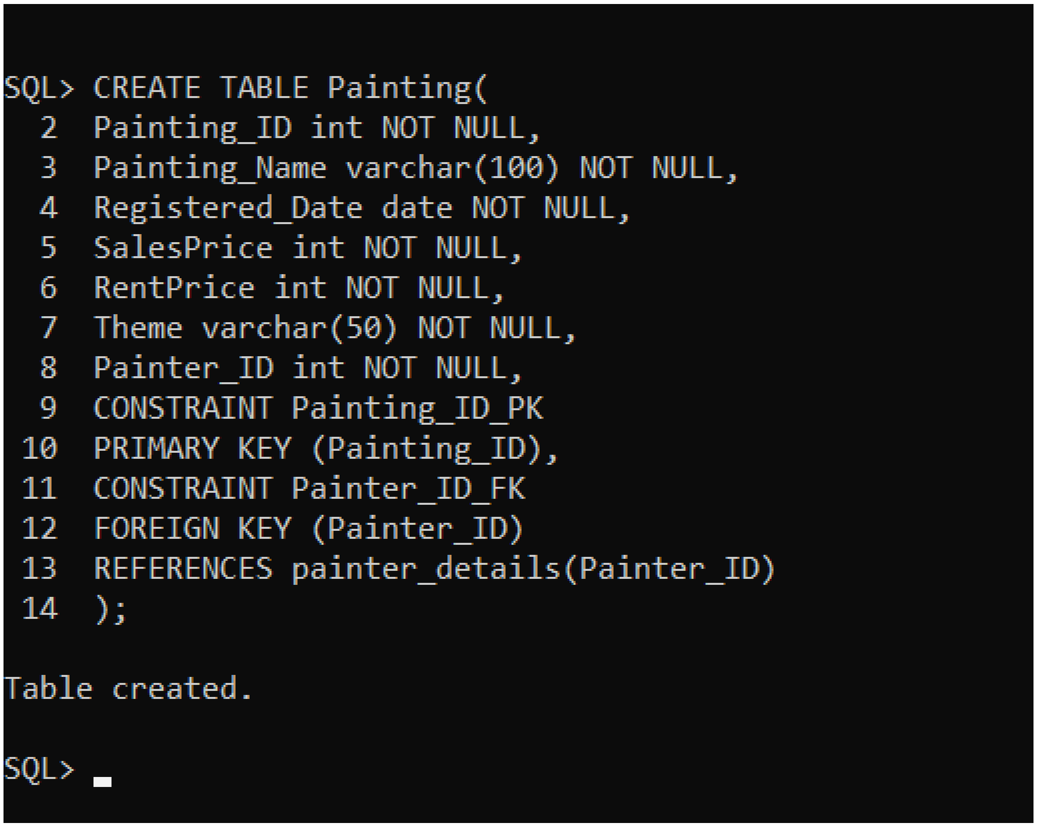
```
SQL> CREATE TABLE customer_details(
     Customer_ID int NOT NULL,
     Customer_Name varchar(100) NOT NULL,
     Customer_Address varchar(100) NOT NULL,
     Customer_Catagory varchar(100) NOT NULL,
     Custmer_Discount int,
     Customer_Contact int NOT NULL,
     CONSTRAINT Customer_ID_PK
     PRIMARY KEY (Customer_ID)
     );

Table created.

SQL>
```

Figure 4: table creation, Customers detalis



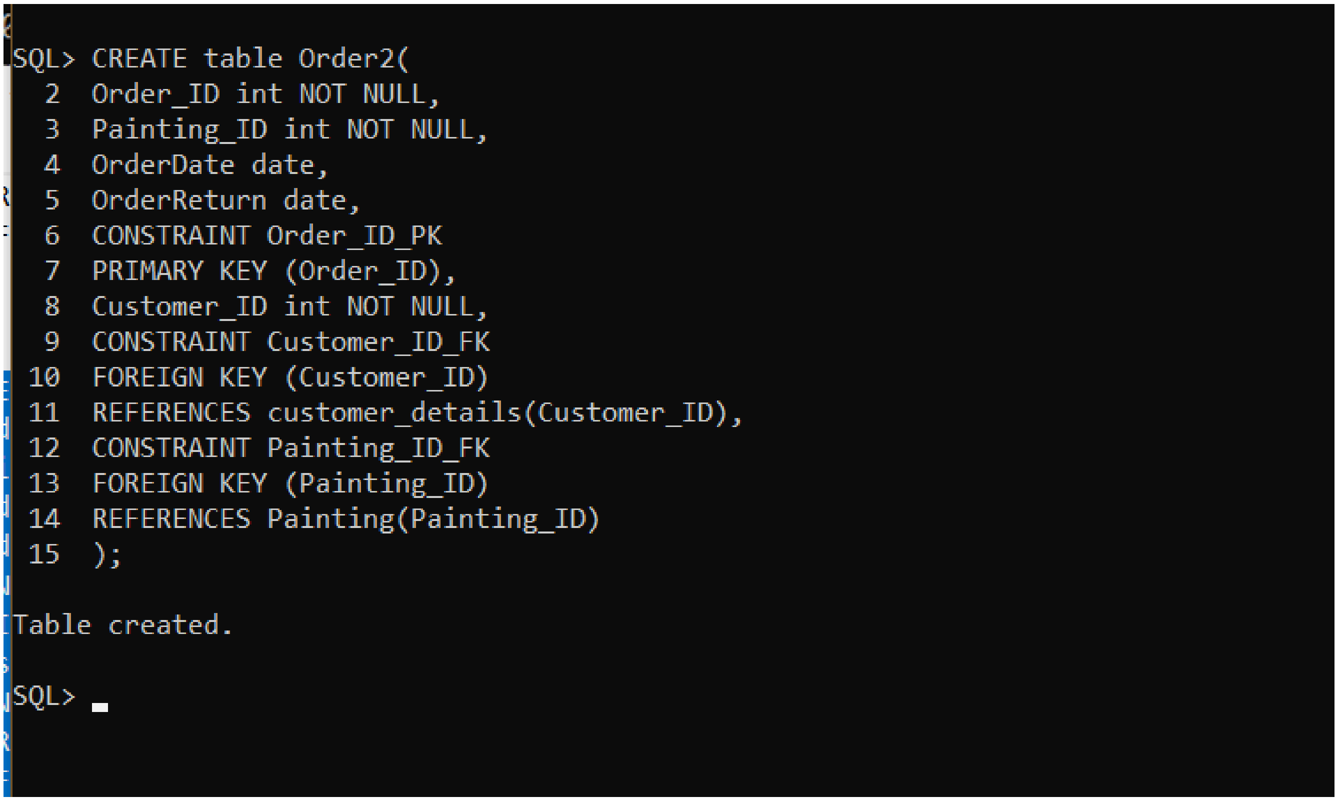
```
SQL> CREATE TABLE painter_details(
Painter_ID int NOT NULL,
Painter_Name varchar(100) NOT NULL,
Painter_Address varchar(100) NOT NULL,
Painter_Contact varchar(100),
CONSTRAINT Paniter_ID_PK
PRIMARY KEY (Painter_ID)
);

Table created.
```

Figure 5: table creation , painter_details



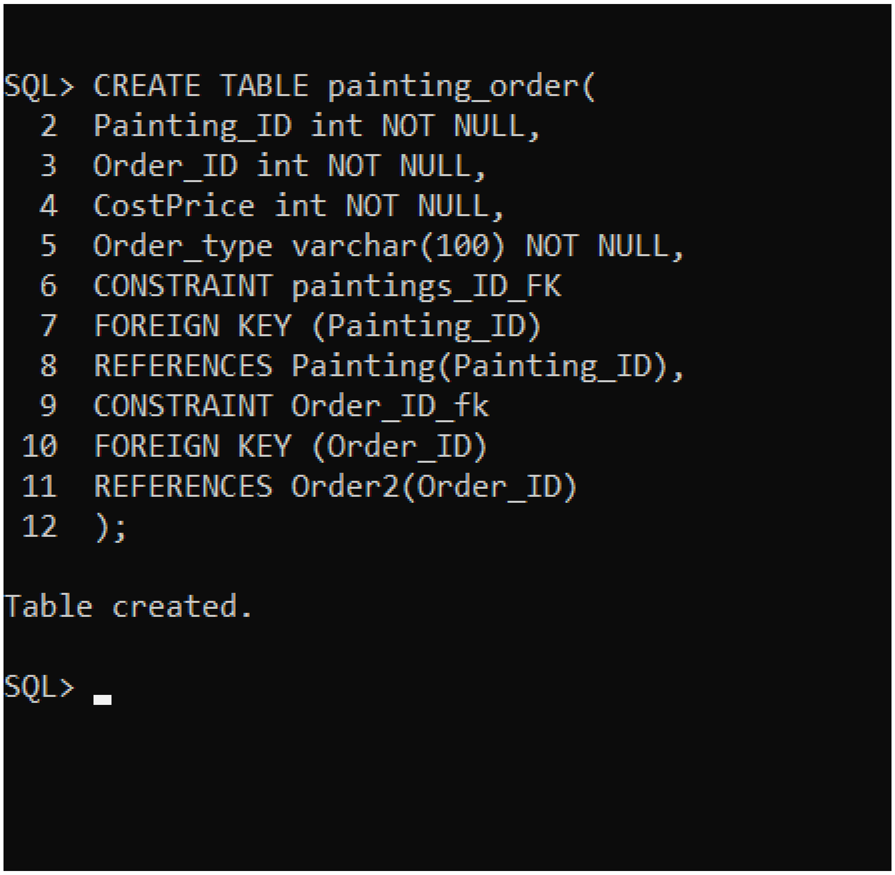
```
SQL> CREATE TABLE Painting(
Painting_ID int NOT NULL,
Painting_Name varchar(100) NOT NULL,
Registered_Date date NOT NULL,
SalesPrice int NOT NULL,
RentPrice int NOT NULL,
Theme varchar(50) NOT NULL,
Painter_ID int NOT NULL,
CONSTRAINT Painting_ID_PK
PRIMARY KEY (Painting_ID),
CONSTRAINT Painter_ID_FK
FOREIGN KEY (Painter_ID)
REFERENCES painter_details(Painter_ID)
);

Table created.

SQL>
```

Figure 6:TABLE CREATION , PAINTINGS



```
SQL> CREATE table Order2(
  2  Order_ID int NOT NULL,
  3  Painting_ID int NOT NULL,
  4  OrderDate date,
  5  OrderReturn date,
  6  CONSTRAINT Order_ID_PK
  7  PRIMARY KEY (Order_ID),
  8  Customer_ID int NOT NULL,
  9  CONSTRAINT Customer_ID_FK
 10  FOREIGN KEY (Customer_ID)
 11  REFERENCES customer_details(Customer_ID),
 12  CONSTRAINT Painting_ID_FK
 13  FOREIGN KEY (Painting_ID)
 14  REFERENCES Painting(Painting_ID)
 15  );

Table created.

SQL>
```

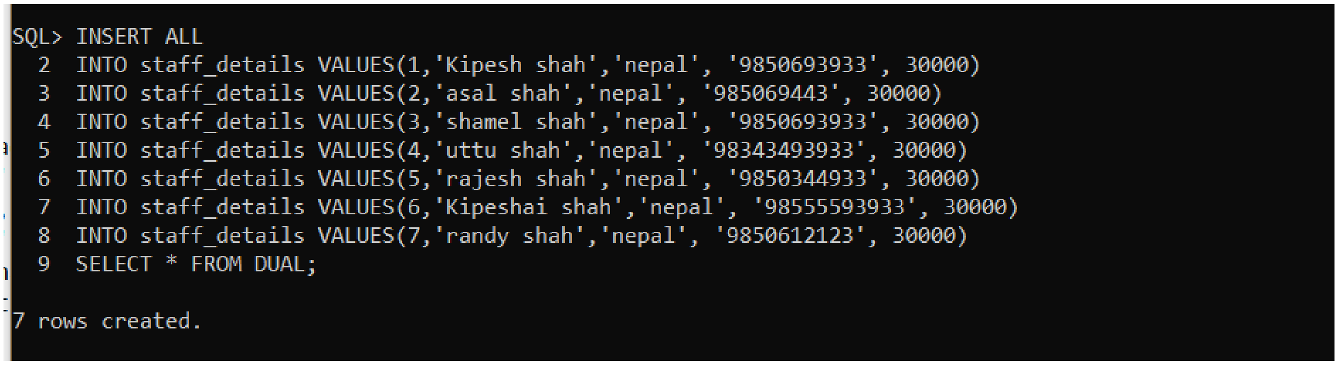
Figure 7: table creation, order table



```
SQL> CREATE TABLE painting_order(
     Painting_ID int NOT NULL,
     Order_ID int NOT NULL,
     CostPrice int NOT NULL,
     Order_type varchar(100) NOT NULL,
     CONSTRAINT paintings_ID_FK
     FOREIGN KEY (Painting_ID)
     REFERENCES Painting(Painting_ID),
     CONSTRAINT Order_ID_fk
     FOREIGN KEY (Order_ID)
     REFERENCES Order2(Order_ID)
     );

Table created.

SQL>
```

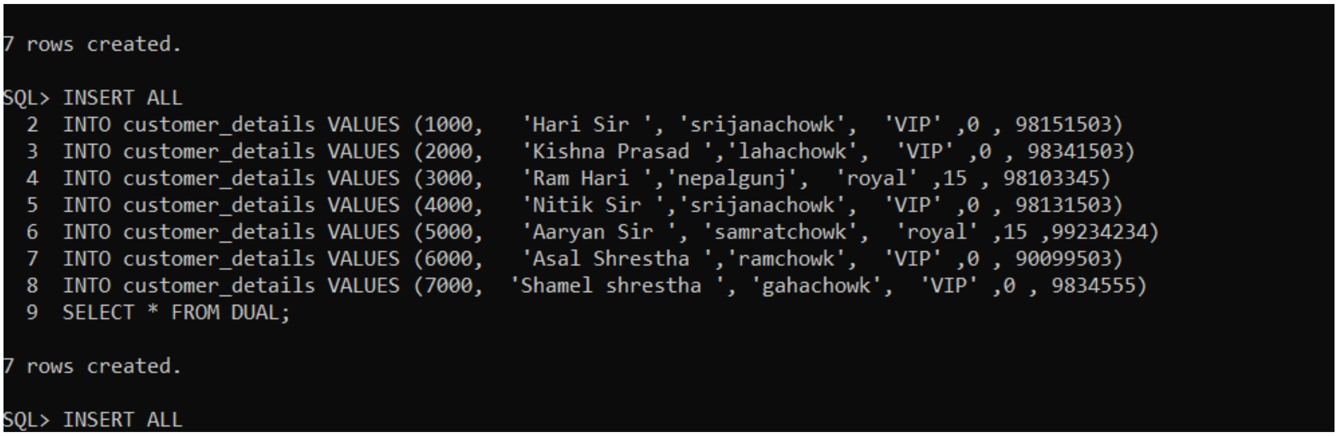
Figure 8: table creation, painting order

Populating data in the tables



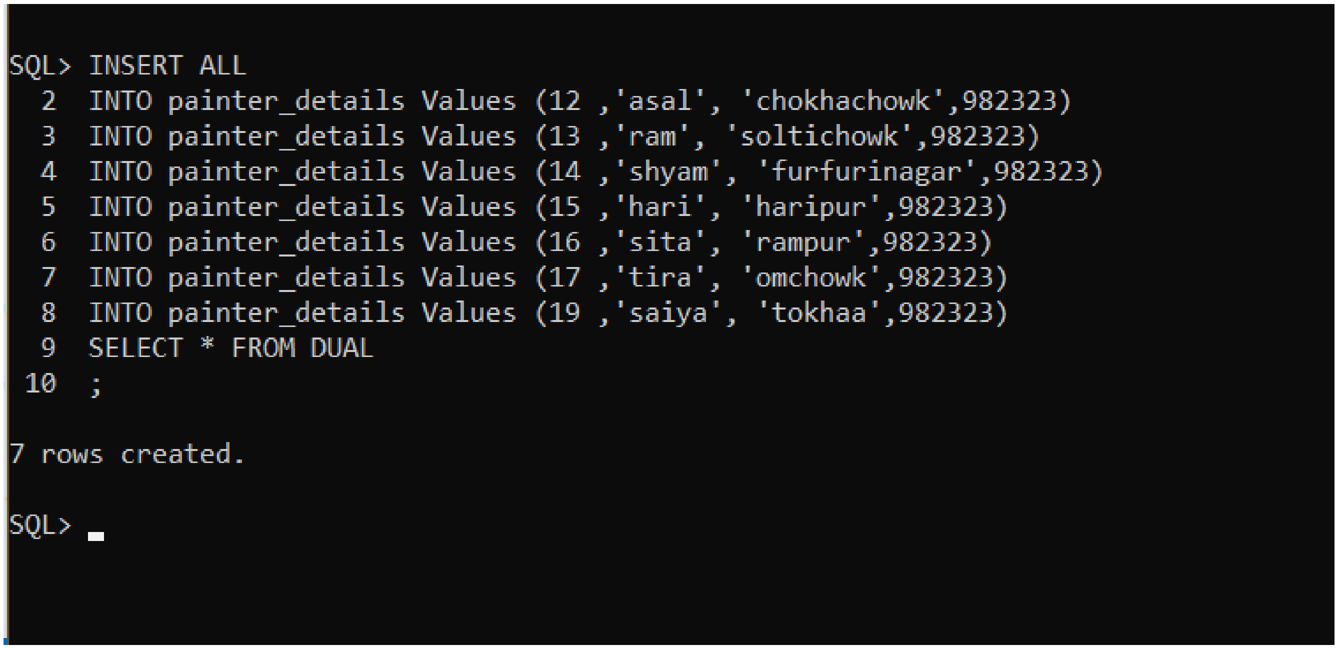
```
SQL> INSERT ALL
     INTO staff_details VALUES(1,'Kipesh shah','nepal', '9850693933', 30000)
     INTO staff_details VALUES(2,'asal shah','nepal', '985069443', 30000)
     INTO staff_details VALUES(3,'shamel shah','nepal', '9850693933', 30000)
     INTO staff_details VALUES(4,'uttu shah','nepal', '98343493933', 30000)
     INTO staff_details VALUES(5,'rajesh shah','nepal', '9850344933', 30000)
     INTO staff_details VALUES(6,'Kipeshai shah','nepal', '98555593933', 30000)
     INTO staff_details VALUES(7,'randy shah','nepal', '9850612123', 30000)
     SELECT * FROM DUAL;

7 rows created.
```

Figure 9: table valur insertion



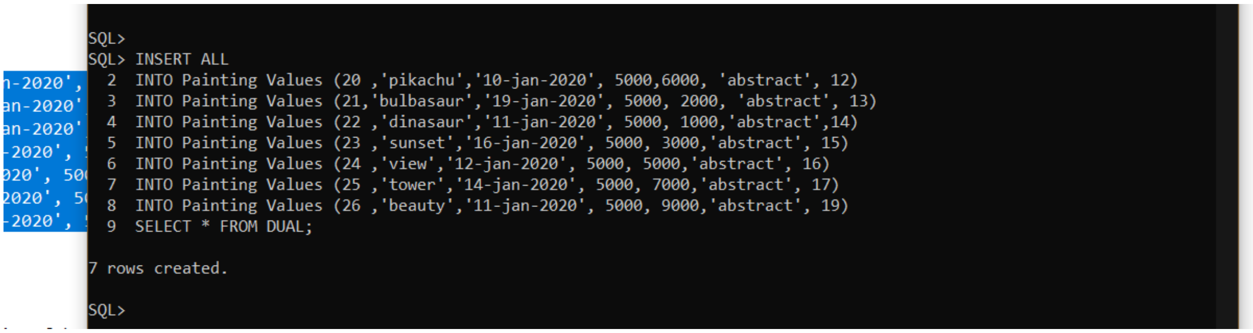
```
7 rows created.

SQL> INSERT ALL
  2  INTO customer_details VALUES (1000,   'Hari Sir ', 'srijanachowk',  'VIP' ,0 , 98151503)
  3  INTO customer_details VALUES (2000,   'Kishna Prasad ','lahachowk',  'VIP' ,0 , 98341503)
  4  INTO customer_details VALUES (3000,   'Ram Hari ','nepalgunj',  'royal' ,15 , 98103345)
  5  INTO customer_details VALUES (4000,   'Nitik Sir ','srijanachowk',  'VIP' ,0 , 98131503)
  6  INTO customer_details VALUES (5000,   'Aaryan Sir ', 'samratchowk',  'royal' ,15 ,99234234)
  7  INTO customer_details VALUES (6000,   'Asal Shrestha ','ramchowk',  'VIP' ,0 , 90099503)
  8  INTO customer_details VALUES (7000,  'Shamel shrestha ', 'gahachowk',  'VIP' ,0 , 9834555)
  9  SELECT * FROM DUAL;

7 rows created.

SQL> INSERT ALL
```

Figure 10: table valur insertion



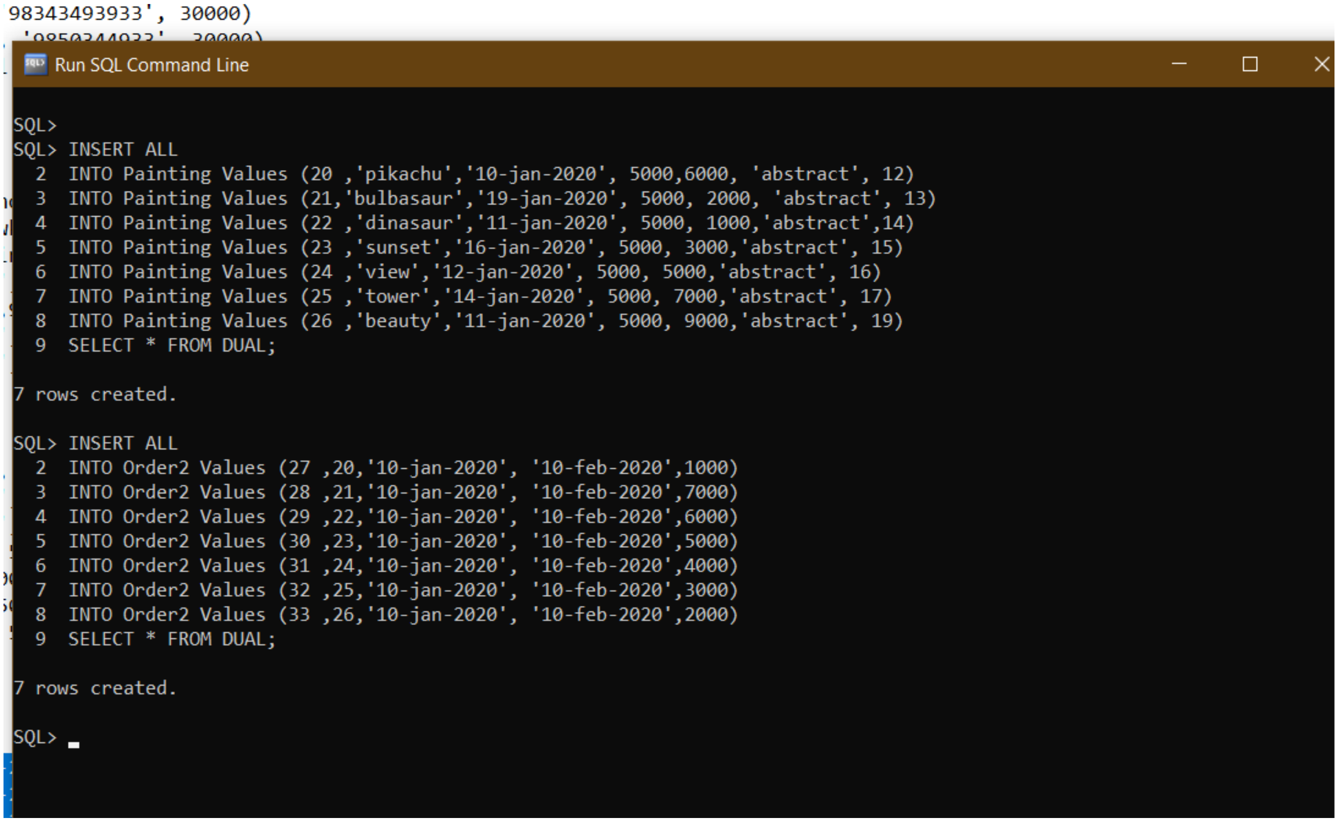
```
SQL> INSERT ALL
  2  INTO painter_details Values (12 ,'asal', 'chokhachowk',982323)
  3  INTO painter_details Values (13 ,'ram', 'soltichowk',982323)
  4  INTO painter_details Values (14 ,'shyam', 'furfurinagar',982323)
  5  INTO painter_details Values (15 ,'hari', 'haripur',982323)
  6  INTO painter_details Values (16 ,'sita', 'rampur',982323)
  7  INTO painter_details Values (17 ,'tira', 'omchowk',982323)
  8  INTO painter_details Values (19 ,'saiya', 'tokhaa',982323)
  9  SELECT * FROM DUAL
 10  ;

7 rows created.

SQL> _
```

Figure 11: table valur insertion



```
SQL>
SQL> INSERT ALL
  2  INTO Painting Values (20 ,'pikachu','10-jan-2020', 5000,6000, 'abstract', 12)
  3  INTO Painting Values (21,'bulbasaur','19-jan-2020', 5000, 2000, 'abstract', 13)
  4  INTO Painting Values (22 ,'dinasaur','11-jan-2020', 5000, 1000,'abstract',14)
  5  INTO Painting Values (23 ,'sunset','16-jan-2020', 5000, 3000,'abstract', 15)
  6  INTO Painting Values (24 ,'view','12-jan-2020', 5000, 5000,'abstract', 16)
  7  INTO Painting Values (25 ,'tower','14-jan-2020', 5000, 7000,'abstract', 17)
  8  INTO Painting Values (26 ,'beauty','11-jan-2020', 5000, 9000,'abstract', 19)
  9  SELECT * FROM DUAL;

7 rows created.

SQL>
```

Figure 12: table valur insertion

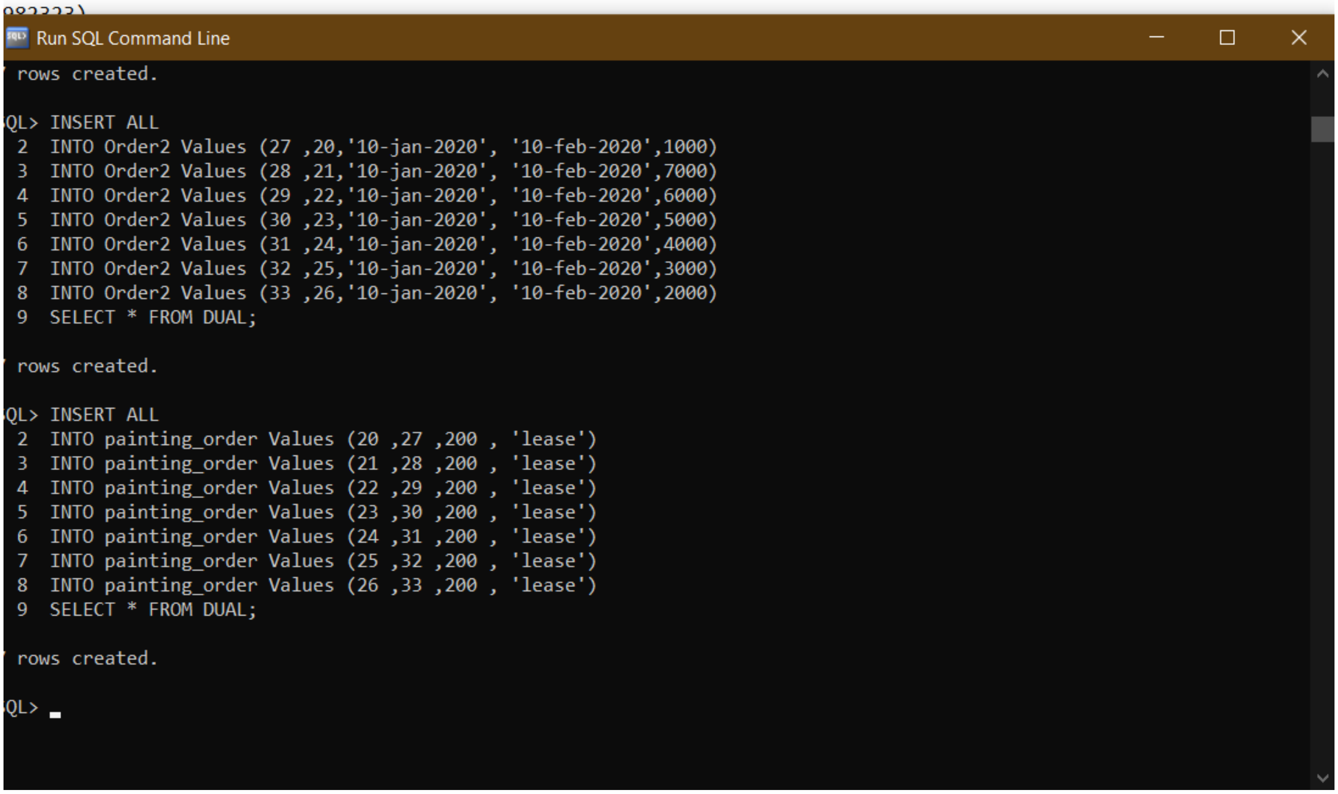


```
98343493933', 30000)
```

```
SQL>
SQL> INSERT ALL
  2  INTO Painting Values (20 ,'pikachu','10-jan-2020', 5000,6000, 'abstract', 12)
  3  INTO Painting Values (21,'bulbasaur','19-jan-2020', 5000, 2000, 'abstract', 13)
  4  INTO Painting Values (22 ,'dinasaur','11-jan-2020', 5000, 1000,'abstract',14)
  5  INTO Painting Values (23 ,'sunset','16-jan-2020', 5000, 3000,'abstract', 15)
  6  INTO Painting Values (24 ,'view','12-jan-2020', 5000, 5000,'abstract', 16)
  7  INTO Painting Values (25 ,'tower','14-jan-2020', 5000, 7000,'abstract', 17)
  8  INTO Painting Values (26 ,'beauty','11-jan-2020', 5000, 9000,'abstract', 19)
  9  SELECT * FROM DUAL;

7 rows created.

SQL> INSERT ALL
  2  INTO Order2 Values (27 ,20,'10-jan-2020', '10-feb-2020',1000)
  3  INTO Order2 Values (28 ,21,'10-jan-2020', '10-feb-2020',7000)
  4  INTO Order2 Values (29 ,22,'10-jan-2020', '10-feb-2020',6000)
  5  INTO Order2 Values (30 ,23,'10-jan-2020', '10-feb-2020',5000)
  6  INTO Order2 Values (31 ,24,'10-jan-2020', '10-feb-2020',4000)
  7  INTO Order2 Values (32 ,25,'10-jan-2020', '10-feb-2020',3000)
  8  INTO Order2 Values (33 ,26,'10-jan-2020', '10-feb-2020',2000)
  9  SELECT * FROM DUAL;

7 rows created.

SQL> _
```

Figure 13: table valur insertion

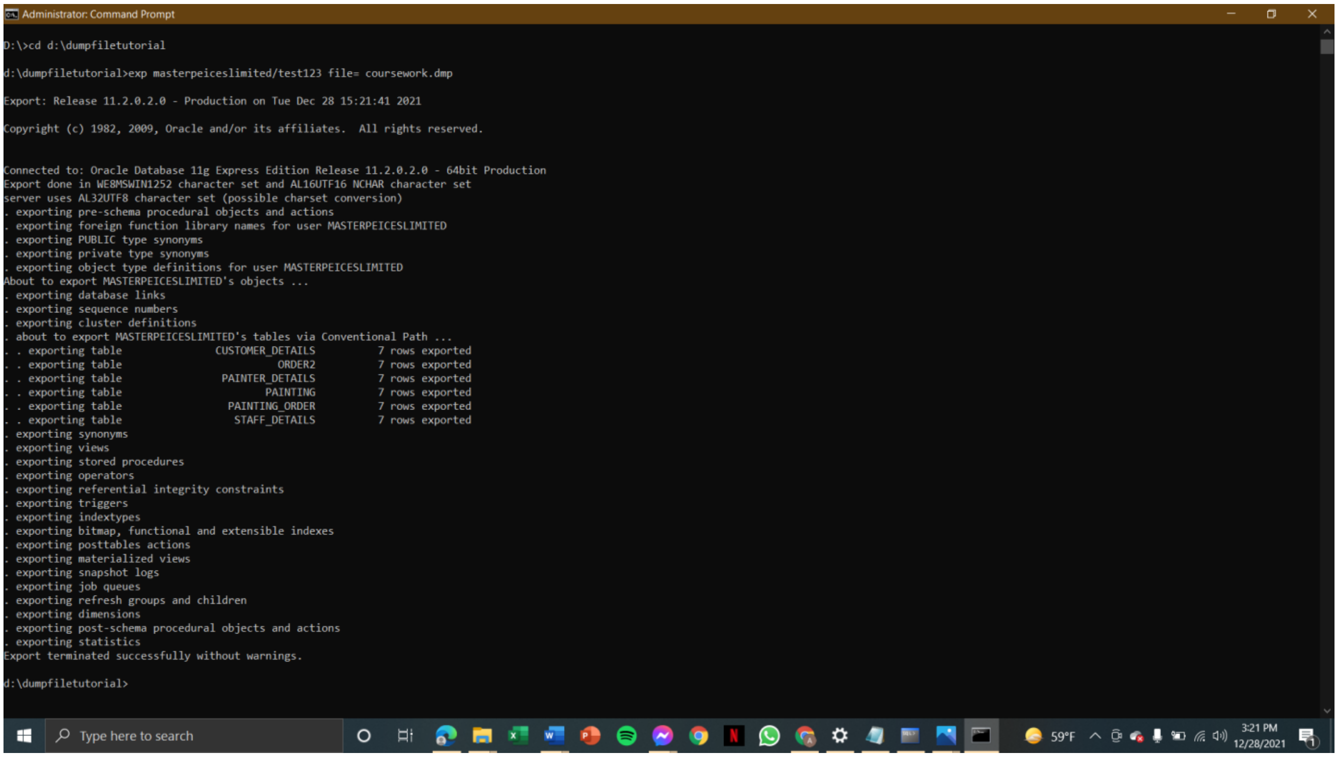
```
agar' ,982323)
982323)
```



```
 rows created.

QL> INSERT ALL
  INTO Order2 Values (27 ,20,'10-jan-2020', '10-feb-2020',1000)
  INTO Order2 Values (28 ,21,'10-jan-2020', '10-feb-2020',7000)
  INTO Order2 Values (29 ,22,'10-jan-2020', '10-feb-2020',6000)
  INTO Order2 Values (30 ,23,'10-jan-2020', '10-feb-2020',5000)
  INTO Order2 Values (31 ,24,'10-jan-2020', '10-feb-2020',4000)
  INTO Order2 Values (32 ,25,'10-jan-2020', '10-feb-2020',3000)
  INTO Order2 Values (33 ,26,'10-jan-2020', '10-feb-2020',2000)
  SELECT * FROM DUAL;

 rows created.

QL> INSERT ALL
  INTO painting_order Values (20 ,27 ,200 , 'lease')
  INTO painting_order Values (21 ,28 ,200 , 'lease')
  INTO painting_order Values (22 ,29 ,200 , 'lease')
  INTO painting_order Values (23 ,30 ,200 , 'lease')
  INTO painting_order Values (24 ,31 ,200 , 'lease')
  INTO painting_order Values (25 ,32 ,200 , 'lease')
  INTO painting_order Values (26 ,33 ,200 , 'lease')
  SELECT * FROM DUAL;

 rows created.

QL> _
```

Figure 14: table value insertion

Creating dump file

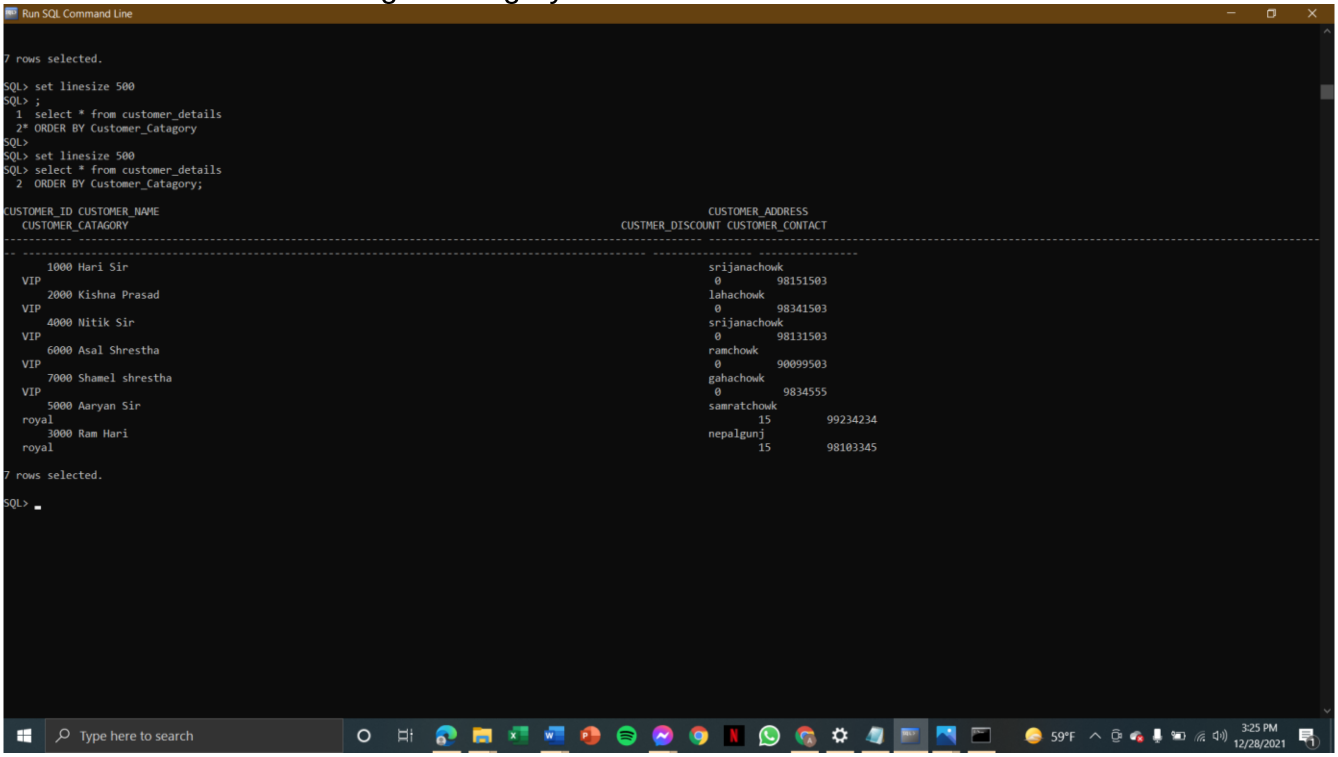


```
D:\>cd d:\dumpfiletutorial

d:\dumpfiletutorial>exp masterpeiceslimited/test123 file= coursework.dmp

Export: Release 11.2.0.2.0 - Production on Tue Dec 28 15:21:41 2021

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 MASTERPEICESLIMITED
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user MASTERPEICESLIMITED
About to export MASTERPEICESLIMITED's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export MASTERPEICESLIMITED's tables via Conventional Path ...
. . exporting table               CUSTOMER_DETAILS          7 rows exported
. . exporting table                         ORDER2          7 rows exported
. . exporting table                PAINTER_DETAILS          7 rows exported
. . exporting table                       PAINTING          7 rows exported
. . exporting table                 PAINTING_ORDER          7 rows exported
. . exporting table                  STAFF_DETAILS          7 rows exported
. 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 without warnings.

d:\dumpfiletutorial>
```

Figure 15:dump file creation

Queries

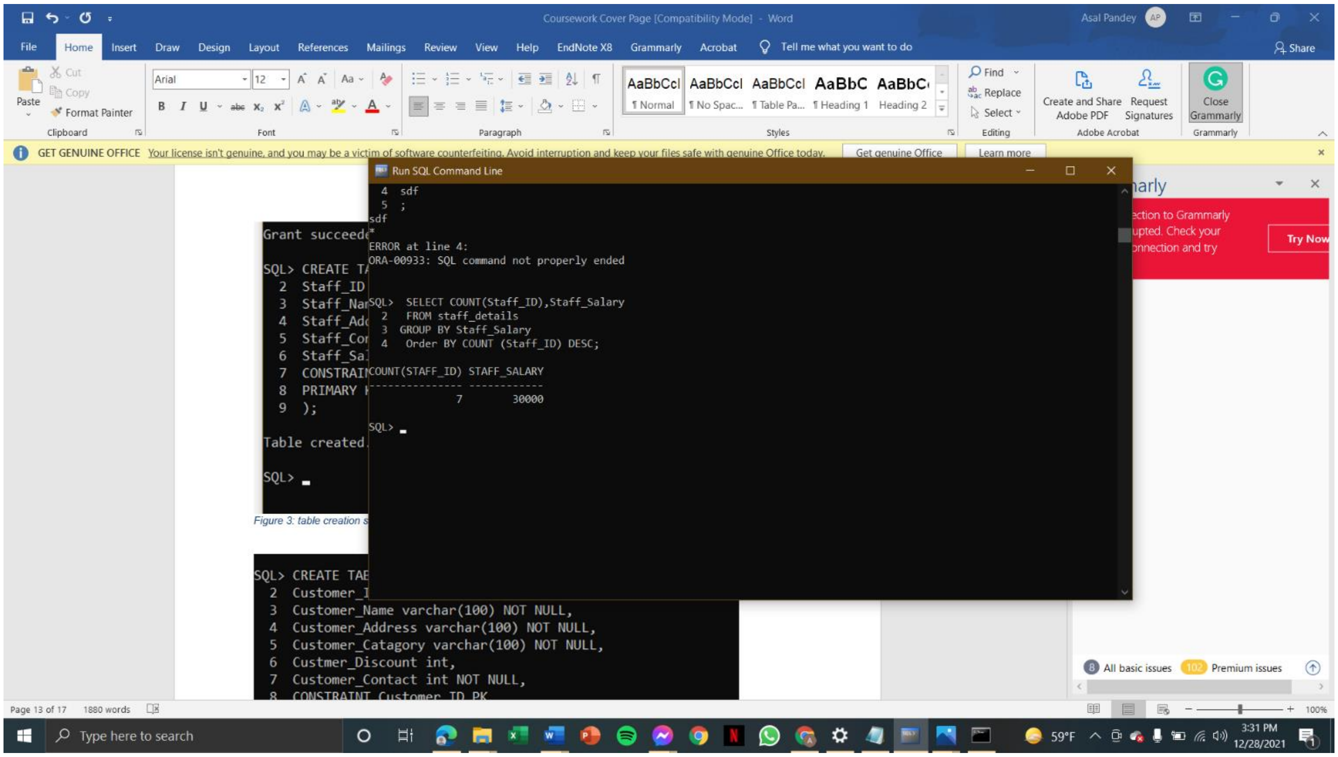
List all customers according to category



```
7 rows selected.

SQL> set linesize 500
SQL> ;
  1  select * from customer_details
  2* ORDER BY Customer_Catagory
SQL>
SQL> set linesize 500
SQL> select * from customer_details
  2  ORDER BY Customer_Catagory;

CUSTOMER_ID CUSTOMER_NAME                                                                                         CUSTOMER_ADDRESS
   CUSTOMER_CATAGORY                                                                             CUSTMER_DISCOUNT CUSTOMER_CONTACT
---------- ------------------------------------------------------------------------------------------------------ --------------------------------------------------------------------------------------------------
-- ---------------------------------------------------------------------------------------------------- --------------- ---------------
      1000 Hari Sir                                                                                               srijanachowk
   VIP                                                                                                             0         98151503
      2000 Kishna Prasad                                                                                          lahachowk
   VIP                                                                                                             0         98341503
      4000 Nitik Sir                                                                                              srijanachowk
   VIP                                                                                                             0         98131503
      6000 Asal Shrestha                                                                                          ramchowk
   VIP                                                                                                             0         90099503
      7000 Shamel shrestha                                                                                        gahachowk
   VIP                                                                                                             0          9834555
      5000 Aaryan Sir                                                                                             samratchowk
   royal                                                                                                                  15         99234234
      3000 Ram Hari                                                                                               nepalgunj
   royal                                                                                                                  15         98103345

7 rows selected.

SQL> _
```

Figure 16:List all customers according to category

Show total staff in Masterpieces Limited sorted by higher salary.

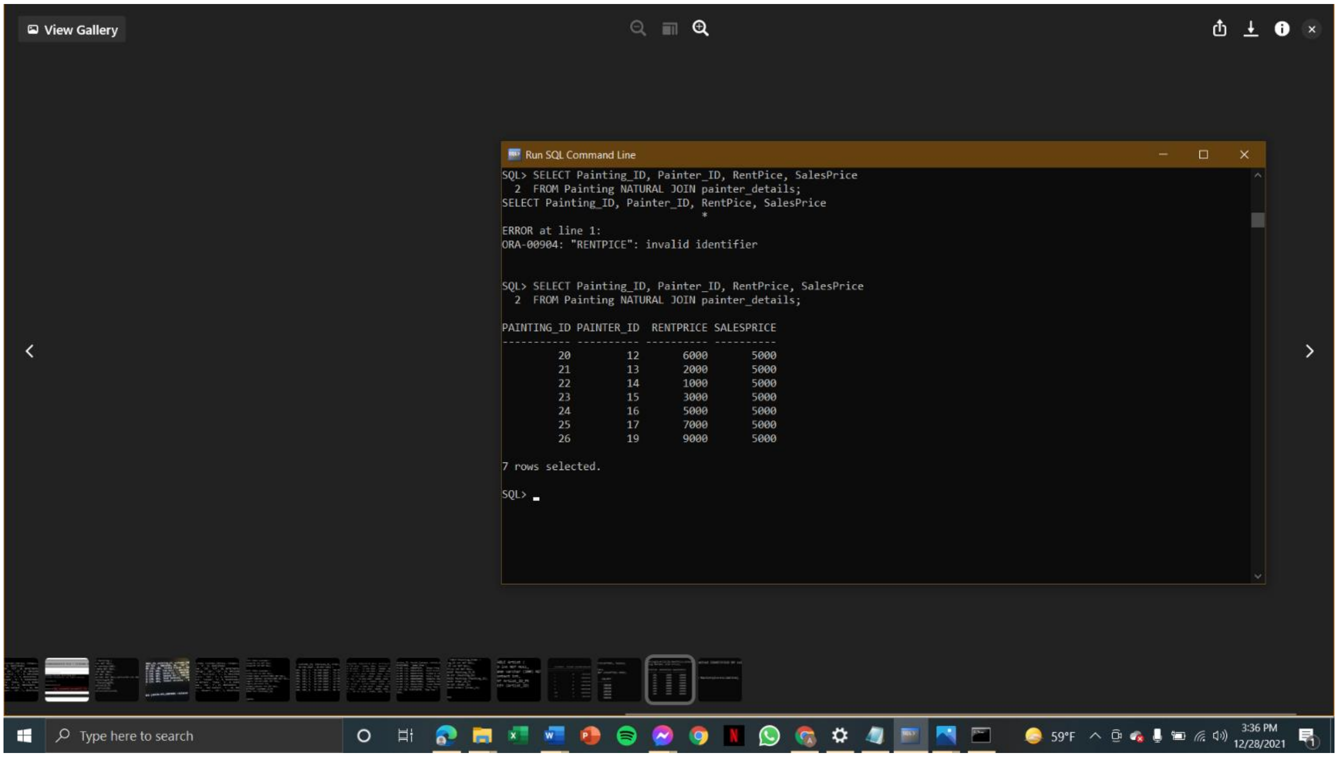


```
  4  sdf
  5  ;
sdf
*
ERROR at line 4:
ORA-00933: SQL command not properly ended


SQL>  SELECT COUNT(Staff_ID),Staff_Salary
  2   FROM staff_details
  3  GROUP BY Staff_Salary
  4   Order BY COUNT (Staff_ID) DESC;

COUNT(STAFF_ID) STAFF_SALARY
--------------- ------------
              7        30000

SQL> _
```

```
  3  Customer_Name varchar(100) NOT NULL,
  4  Customer_Address varchar(100) NOT NULL,
  5  Customer_Catagory varchar(100) NOT NULL,
  6  Custmer_Discount int,
  7  Customer_Contact int NOT NULL,
  8  CONSTRAINT Customer_ID_PK
```

Figure 17; Show total staff in Masterpieces Limited sorted by higher salary.

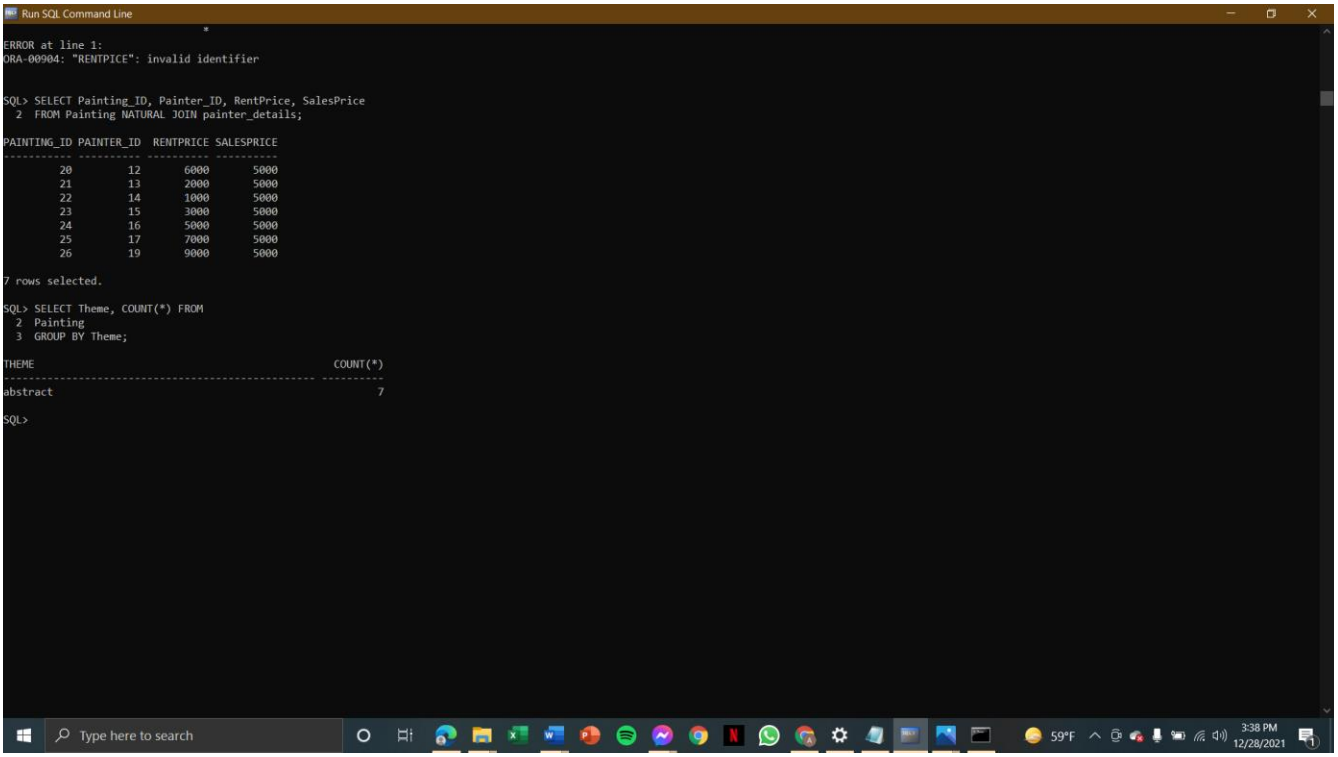
List paintings and their artist with monthly rental price and paid price.



```
SQL> SELECT Painting_ID, Painter_ID, RentPice, SalesPrice
  2  FROM Painting NATURAL JOIN painter_details;
SELECT Painting_ID, Painter_ID, RentPice, SalesPrice
                                *
ERROR at line 1:
ORA-00904: "RENTPICE": invalid identifier


SQL> SELECT Painting_ID, Painter_ID, RentPrice, SalesPrice
  2  FROM Painting NATURAL JOIN painter_details;

PAINTING_ID PAINTER_ID  RENTPRICE SALESPRICE
----------- ---------- ---------- ----------
         20         12       6000       5000
         21         13       2000       5000
         22         14       1000       5000
         23         15       3000       5000
         24         16       5000       5000
         25         17       7000       5000
         26         19       9000       5000

7 rows selected.

SQL> _
```

Figure 18; List paintings and their artist with monthly rental price and paid price.

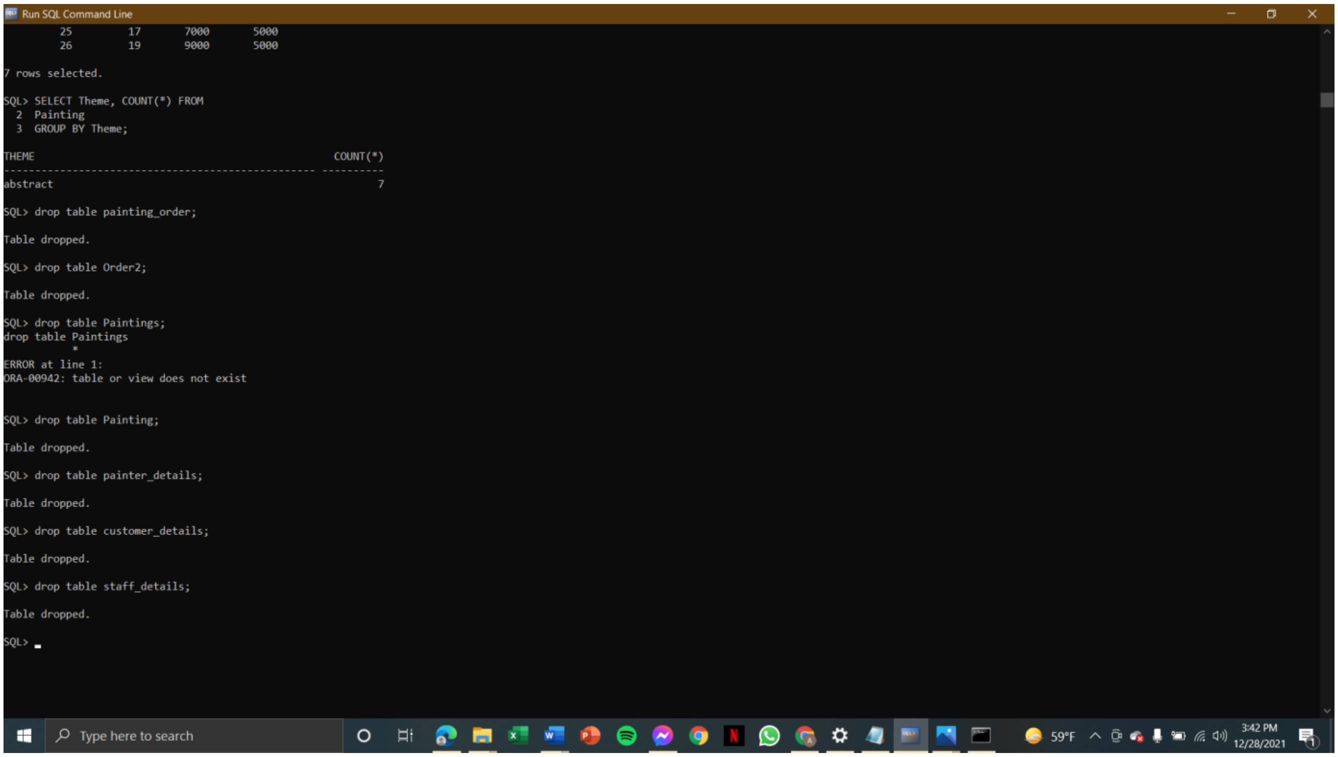
List the number of paintings available for rent according to category



```
                                                  *
ERROR at line 1:
ORA-00904: "RENTPICE": invalid identifier


SQL> SELECT Painting_ID, Painter_ID, RentPrice, SalesPrice
  2  FROM Painting NATURAL JOIN painter_details;

PAINTING_ID PAINTER_ID  RENTPRICE SALESPRICE
----------- ---------- ---------- ----------
         20         12       6000       5000
         21         13       2000       5000
         22         14       1000       5000
         23         15       3000       5000
         24         16       5000       5000
         25         17       7000       5000
         26         19       9000       5000

7 rows selected.

SQL> SELECT Theme, COUNT(*) FROM
  2  Painting
  3  GROUP BY Theme;

THEME                                                COUNT(*)
-------------------------------------------------- ----------
abstract                                                    7

SQL>
```

Figure 19: List the number of paintings available for rent according to category

Dropping tables



```
        25          17       7000       5000
        26          19       9000       5000

7 rows selected.

SQL> SELECT Theme, COUNT(*) FROM
  2  Painting
  3  GROUP BY Theme;

THEME                                                      COUNT(*)
-------------------------------------------------- ----------
abstract                                                          7

SQL> drop table painting_order;

Table dropped.

SQL> drop table Order2;

Table dropped.

SQL> drop table Paintings;
drop table Paintings
           *
ERROR at line 1:
ORA-00942: table or view does not exist


SQL> drop table Painting;

Table dropped.

SQL> drop table painter_details;

Table dropped.

SQL> drop table customer_details;

Table dropped.

SQL> drop table staff_details;

Table dropped.

SQL> _
```

Figure 20: dropping tables

Critical Evaluation

This project was about creating a database for a company named “masterpieces limited” which is an online platform focusing on renting paintings made by different artists and creators. The objective of the business was to lease the paintings to private individuals and commercial companies owned by masterpieces, as well as different painters or artists for commercial purposes. Tons of research was done and various tasks were performed during this project like creating ERDs, normalization of the tables, creation the database. The database was created where different pieces of information about staff, customers, painters, orders, paintings and painters were stored. As data and information are the key factors for any organization to run the organization effectively, Various research was done to complete this project in time.

Many new things were learned during the project regarding ERD, normalization, SQL commands and many more. Normalization was a new topic for me this semester for this coursework and it played a vital role in the completion of this coursework. The creation of the initial ERD contained various problems like data redundancy, data anomalies because of it normalization was carried out and final ERD was obtained. After that creation of tables and insertion of values in the tables were done. The research on the UPDATE, ALTER, DROP query was done and performed in this coursework.

New knowledge was obtained about normalization, maintaining relationships among the table, use of primary key and foreign key etc and they are very useful. Various queries were cleared while researching this project. lecturer and tutors also helped in carrying out this project by being there and supervising everything at every point. Many errors and mistakes and tons of queries were solved with the help of researchers, teachers and the internet

References

GEEKS FOR GEEKS, 2020. *TYPES OF NORMAL FORM IN DBMS.* [Online]
Available at: https://www.geeksforgeeks.org/types-of-normal-forms-in-dbms/
[Accessed 27 12 2021].

Peterson, R., 2021. *Normalisation in DBMS.* [Online]
Available at: https://www.guru99.com/database-normalization.html
[Accessed 28 12 2021].