LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CC5051NI Databases

Assessment Weightage & Type

50% Individual Coursework

Year and Semester

2021 Autumn

Student Name: Aadesh Gyawali

London Met ID:20048793

College ID: NP01CP4S10260

Assignment Due Date:12/28

Assignment Submission Date:12/28

Title (Where Required):

Word Count (Where Required):

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.

Table of Contents

Table Of figures

1. Introduction

Mr. Steve is a well-known local businessman and real state owner. He have spread his business mind to all other business fields. Now he wants to move one step forward in his business fields. He has decided to expand business into paintings. Now a days online business are growing very fast in compared to others businesses. So, Mr. Steve has decided to register it as internet business under the name of "Masterpieces Limited".

It aims to lease the paintings to private individuals and commercial companies owned by masterpieces, as well as different painters or artists. To mobilize his painting business smoothly. He has been looking for consulting and designing companies. We need to help him for his services to design and implement a database to flourish his new business. There are further requirement to run his business smoothly which we will be discussing below.

The company provides the additional facility for the customers. Customers are categorized as regular, loyal, privileged and VIP and provides discount of 15%, 10%, 5% and 0%. Discount are very uniform.

1.1 Current Business Activities and Operations

- Collecting the various paintings and creativity work from public and keep for show piece.
- Hiring the consulting and designing companies.
- Masterpiece allows customer buy or lease paintings.
- Masterpiece allows customer lease the paintings and even lease their paintings.
- Its runs various exhibition program in various places frequently for the promotion of business.
- Proper record of customer who have leased the paintings.

1.2 List of Business Rules

- Customer are categorized as Regular (R), Loyal (L), privileged (P) and VIP (V) and discount are given 15%,10 %, 5% and 0% respectively.
- Customer can also lease their painting on Masterpiece.
- Each painting has a certain cost as a monthly rental price offered by the owner (painter or artist).
- The owner of leased painting is paid 20% of leased amount.
- Any painting that are not leased within four months are returned to the owner.
- After the return of painting, owner can resubmit a returned painting after one months
- Each painting can only have one artist associated with it.

1.3 Identification of Entities and Attributes

An entity is an object or component of data. An Entity can have multiple attributes. Each entity has a Primary Key and optional foreign key. In DBMS, Attributes are pre-defined values that specify and denote what kind of values are to be entered. These values maybe varchar, integer, date, etc (BeginnerBook, 2021).

Entities	**Attributes**
Order	Order_ID(PK), Order_Date, Order_Total
Staff	Staff_ID(PK),Staff_Name, Staff_Address, Salary
Customer	Customer_ID(PK), Customer_Name, Address, Category, Category_Discount,
Painting	Painting_ID(PK), Painting_Name, Theme, Availability, Artist_Name, Artist_ID,Artist_Salary, PaintingListed_Date, Rental_Price, Rental _Date,Return_Date, Line-Total_Price

2. Initial ERD

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes (LucidChart, 2021).

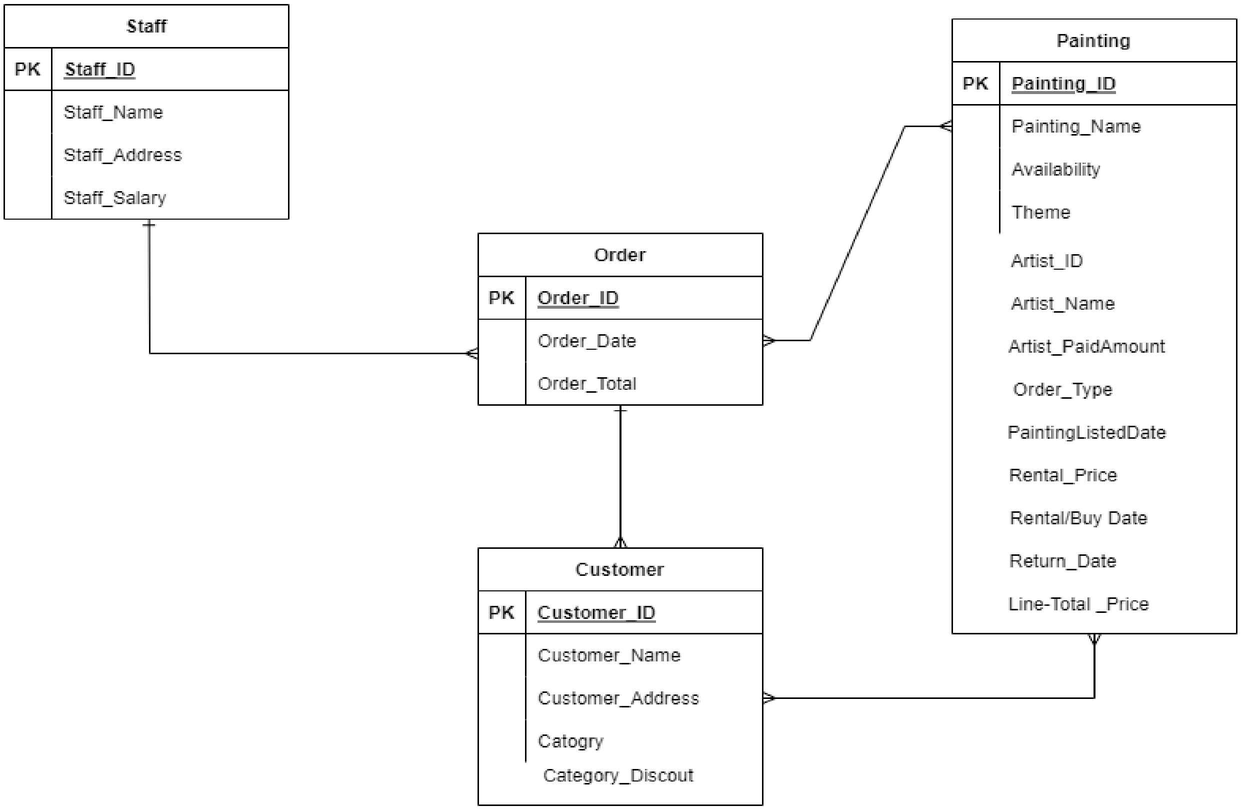


Figure 1 Initial ER Diagram

The above ER-Diagram type of relationship between our initial entities, the attributes they hold. In this ERD module the relation between customer,order,painting and staff is showed above.

3. Normalization

Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of Normalization in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically (Peterson, 2021).

3.1 UNF (un-normalized normal form)

In UNF we show all the attributes that are held in the database. We show repeating group inside curly braces. The repeating group is determined with a relation to a Primary or Main key.

Showing Repeating Group According scenario:

Order-1 (Order_ID, Order_Date,{Customer_ID, Customer_Name, Address, Category, Category_Discount, { Painting_ID, Painting_Name, Theme, Availability, Artist_Name, Artist_ID,Artist_Salary, PaintingListed_Date, Rental_Price, Rental _Date,Return_Date, Line-Total_Price} }Staff_ID,Staff_Name, Staff_Address, Salary, Order_Total)

3.2 1NF (First Normal Form)

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

A table is in 1 NF:

1. There are only Single Valued Attributes.
2. Attribute Domain does not change.
3. There is a unique name for every Attribute/Column.
4. The order in which data is stored does not matter (GeeksforGeeks, 2021).

ORDER -1= Order_ID, Order_Date,Staff_ID,Staff_Name, Staff_Address, Salary, Order_Total

Customer -1 = Customer_ID, Customer_Name, Address, Category, Category_Discount, Order_ID

Painting-1 = Painting_ID, Painting_Name, Theme, Availability, Artist Name, Artist ID, Artist_Salary,PaintingListed_Date, Rental Price, Rental Date,Return_Date, Line-Total_Price, Customer_ID, Order_ID

3.3 2NF (Second Normal Form)

In 2NF all the tables are free from Partial Dependency. All non-key attributes are fully functionally dependent on the primary key. A composite key is a combination of two or more foreign key that makes a Primary key. All partially dependent attributes are kept in the table formed by the composite keys (javatpoint, 2021).

ORDER -1= Order_ID, Order_Date,Staff_ID,Staff_Name, Staff_Address, Salary, Order_Total

Since there is only one PK, it is already in 2NF

Order-2 = Order_ID, Order_Date,Staff_ID,Staff_Name, Staff_Address, Salary, Order_Total

Customer - 1

Customer_ID -> Customer_Name, Customer_Address, Category, Category_Discount
Order_ID - >x
Order_ID,Customer_ID -> x

Customer-2= Customer_ID, Customer_Name, Customer_Address, Category, Category_Discount

Order-Customer 2= Order_ID,Customer_ID

Painting-1

Painting_ID - > Painting_Name, Theme, Availability, Artist_Name, Artist_Name, Artist_ID, Artist_Salary, PaintingListed_Date, Rental Price
Customer_ID - > x
Order_ID - > x
Painting_ID, Customer_ID -> x
Customer_ID, Order_ID ->x
Painting_ID,Order_ID - >x
Painting_ID,Customer_ID,Order_ID ->Rental Date,Return_Date, Line-Total_Price

Painting-2 = Painting_ID, Painting_Name, Theme, Availability, Artist_Name, Artist_Name, Artist_ID, Artist_Salary, PaintingListed_Date, Rental Price
Painting-Customer-Order-2-> Painting_ID,Customer_ID,Order_ID,Rental Date,Return_Date, Line-Total_Price

After the 2NF:

Order-2 = Order_ID, Order_Date,Staff ID,Staff_Name, Staff_Address, Salary, Order_Total

Customer-2= Customer_ID, Customer_Name, Customer_Address, Category, Category_Discount

Painting-2 = Painting_ID, Painting_Name, Theme, Availability, Artist_Name, Artist_ID, Artist_Salary, PaintingListed_Date, Rental Price
Painting-Customer-Order-2-> Painting_ID,Customer_ID,Order_ID,Rental Date,Return_Date, Line-Total_Price

3.4 3NF (Third Normal Form)

In 3NF all the tables are free from Transitive Property. A table is said to be 3NF (Third Normal Form) when the following conditions holds:

a) Identify any dependencies between non-key attributes within each table
b) Remove them to a new table
c) Decide on a primary key
d) This primary key becomes the foreign key in the original table.

Checking for Transitive Dependency for Order-2
OrderID- > Staff_ID - > Staff,Name, Staff_Address, Salary

Order-3 = Order_ID, Order_Date, Order_Total, Staff_ID*

Staff-3 = Staff_ID, Staff_Name, Staff_Address, Salary

Checking for Transitive Dependency for Customer-2
Customer_ID-> Category-> Category_Discount

Customer-3 = Customer_Name, Customer_Address, Category*

Category-3 = Category, Category_Discount

Checking Transitive Dependency for Painting-3

Painting_ID - > Artist_ID -> Artist_Name, Artist_Salary

Painting-3 = Painting_ID, Painting_Name, Theme, Availability, PaintingListed_Date, Rental Price

Artist-3 = Artist_ID,Artist_Name, Artist_Salary

Since there is no transitive dependency in Painting-Customer-Order-2, it is already in 3NF.

The entities after normalization are :

Final Entities:

Order-3 = Order_ID, Order_Date, Order_Total, Staff_ID*

Staff-3 = Staff_ID, Staff_Name, Staff_Address, Salary

Customer-3 =Customer_ID, Customer_Name, Customer_Address, Category*

Category-3 = Category, Category_Discount

Painting-3 = Painting_ID, Painting_Name, Theme, Availability, PaintingListed_Date, Rental Price, selling_price,Artist_ID*

Artist-3 = Artist_ID,Artist_Name, Artist_Salary

Painting-Customer-Order-3= Painting_ID,Customer_ID,Order_ID,Rental Date,Return_Date, Line-Total_Price,order_type

4. Final ERD

After we achieve the 3rd Normal form of our database, we have eliminated data redundancy and data anomalies as data insertion, data updating and data deletion. We have removed transitive and partial dependency from the tables of our database making our database increasing functionality of our database and make our data consistent. New Entities were formed after normalization and displayed and displayed in the form of ER-Diagram below.

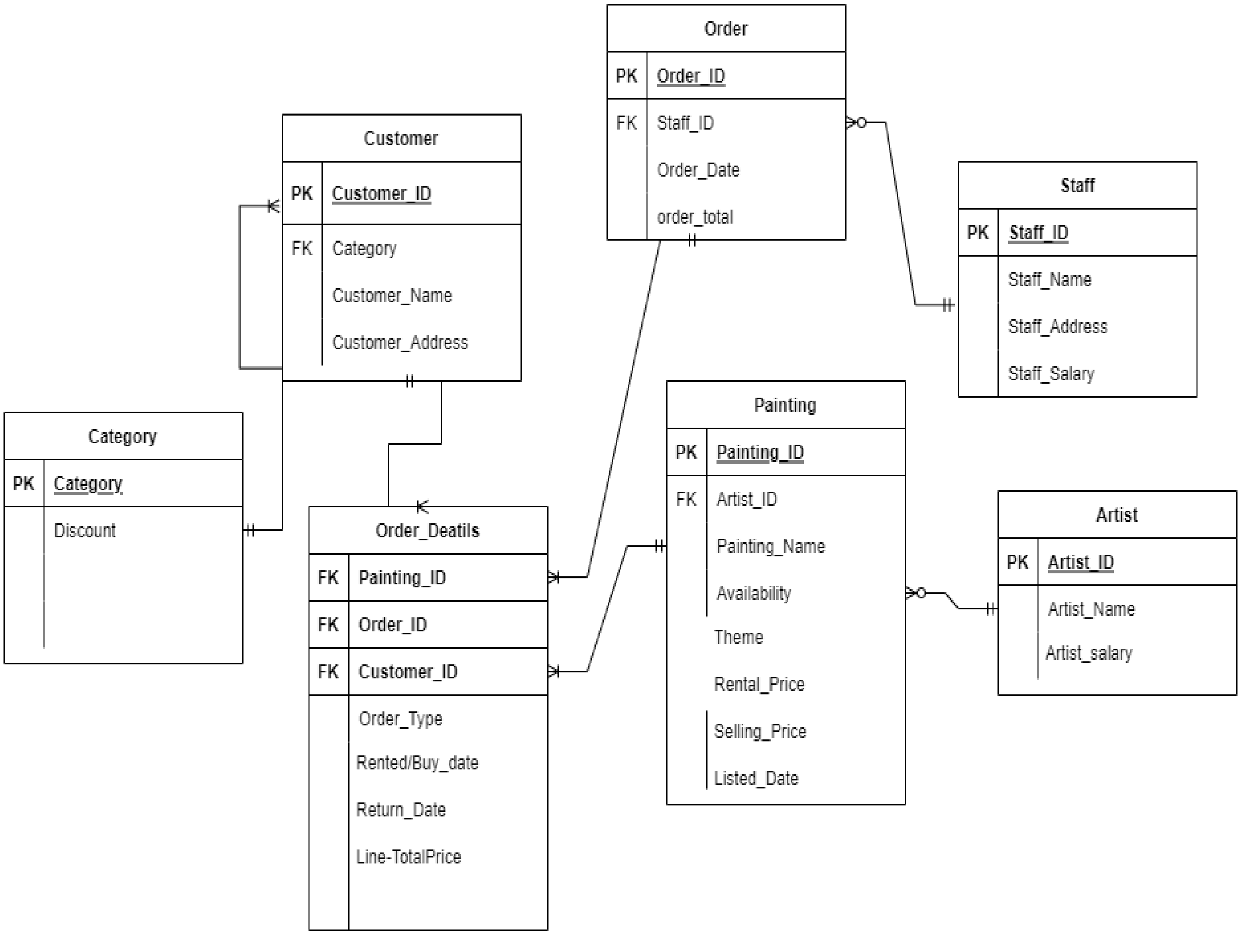


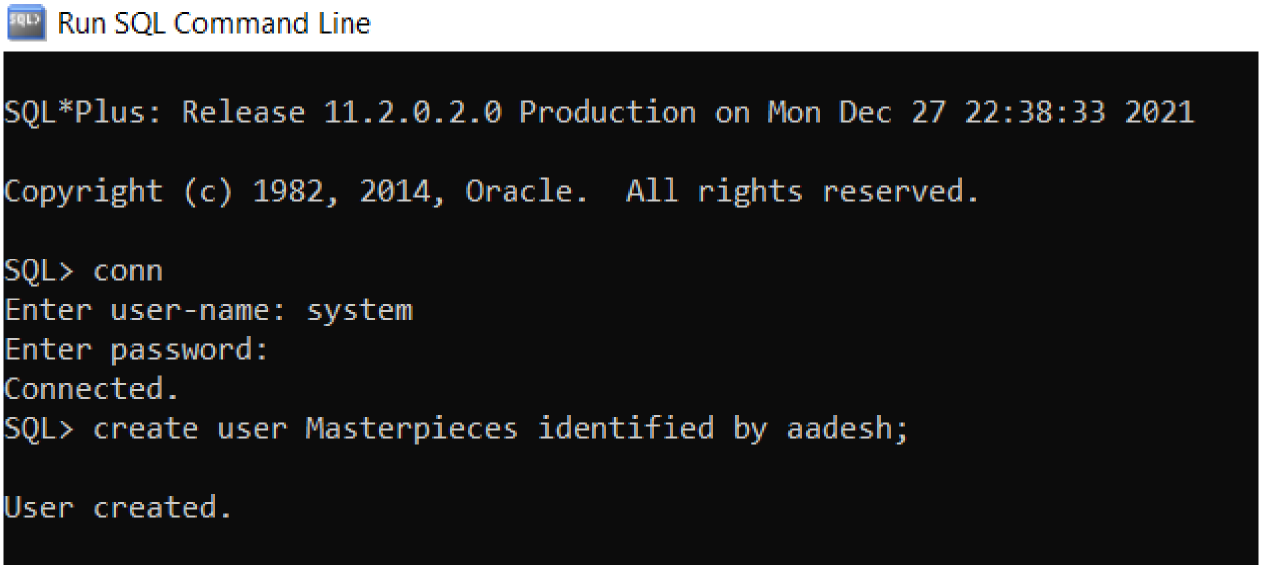
Figure 2 FINAL ERD

5. Implementation

In the initial phase we should have to create the user and grant the user with all the privileges which gives the user the right to run a particular type of SQL statement, or the rights to access the belonging object to another user, and can run a PL/SQL package and so on. Then we have connected with the user and the other queries are done.

5.1 Creating user

The above command “Create user Masterpieces identified by aadesh” creates a new user “Masterpiece” with its access password “aadesh”.



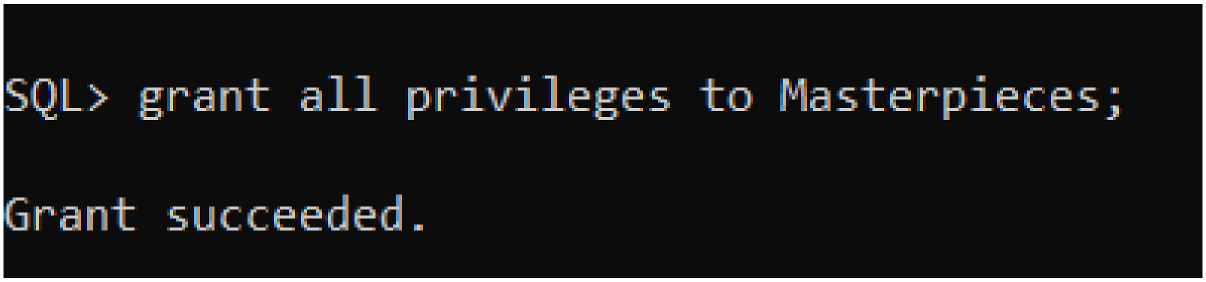
Run SQL Command Line

```
SQL*Plus: Release 11.2.0.2.0 Production on Mon Dec 27 22:38:33 2021

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

SQL> conn
Enter user-name: system
Enter password:
Connected.
SQL> create user Masterpieces identified by aadesh;

User created.
```

Figure 3 creating new user

5.2 Granting Privileges to user

The command “grant all privileges to Masterpiece” gives all the rights in oracle.

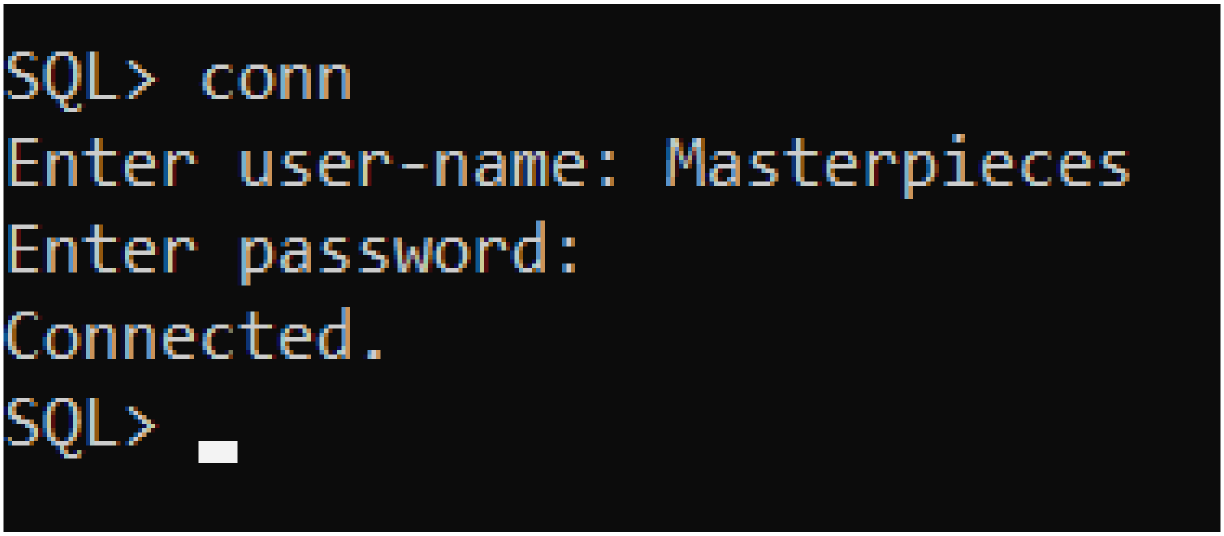


```
SQL> grant all privileges to Masterpieces;

Grant succeeded.
```

Figure 4 Granting privileges to user

5.3 Connect to user

The above command conn Masterpieces connects us to Masterpiece and after the successfully connected to user, we can run queries

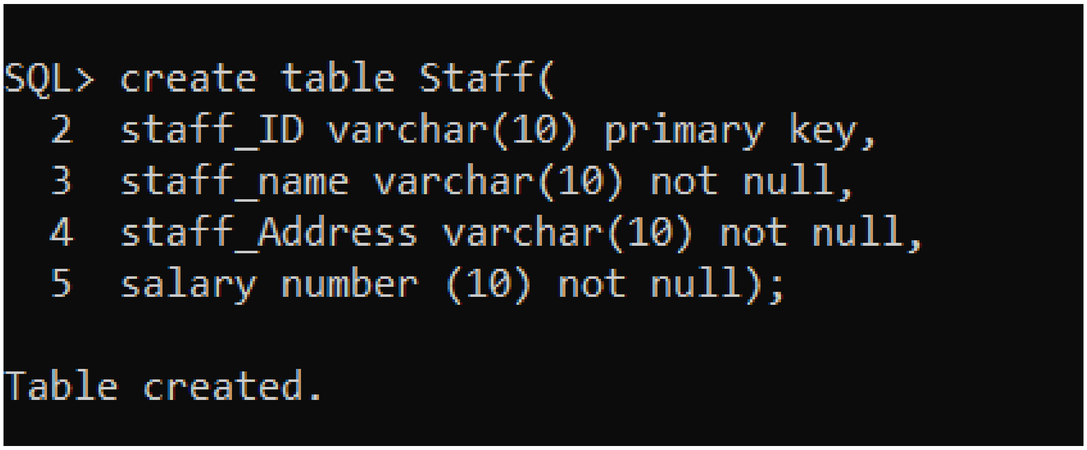


```
SQL> conn
Enter user-name: Masterpieces
Enter password:
Connected.
SQL> _
```

Figure 5 Connect to user

5.4 Creating Tables

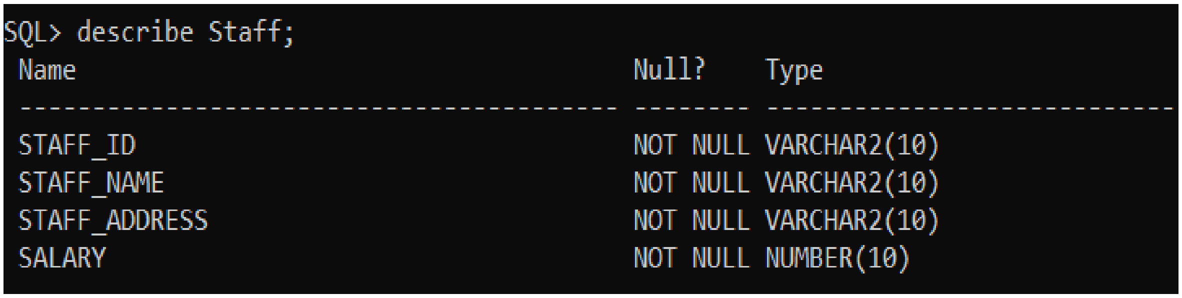
- **Staff Table**



```
SQL> create table Staff(
  2  staff_ID varchar(10) primary key,
  3  staff_name varchar(10) not null,
  4  staff_Address varchar(10) not null,
  5  salary number (10) not null);

Table created.
```

Figure 6 Creating Staff table

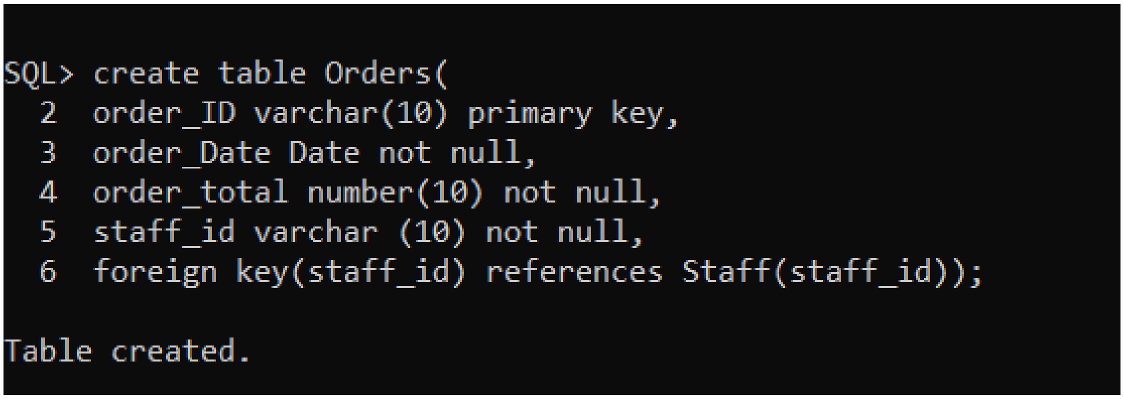
- describe staff table



```
SQL> describe Staff;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STAFF_ID                                  NOT NULL VARCHAR2(10)
 STAFF_NAME                                NOT NULL VARCHAR2(10)
 STAFF_ADDRESS                             NOT NULL VARCHAR2(10)
 SALARY                                    NOT NULL NUMBER(10)
```

Figure 7 Describe staff

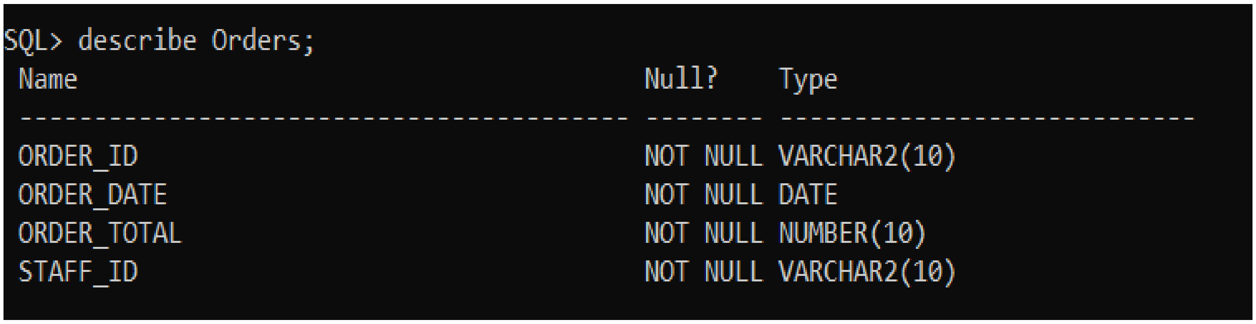
- **Order Table**



```
SQL> create table Orders(
  2  order_ID varchar(10) primary key,
  3  order_Date Date not null,
  4  order_total number(10) not null,
  5  staff_id varchar (10) not null,
  6  foreign key(staff_id) references Staff(staff_id));

Table created.
```

Figure 8 Creating order table

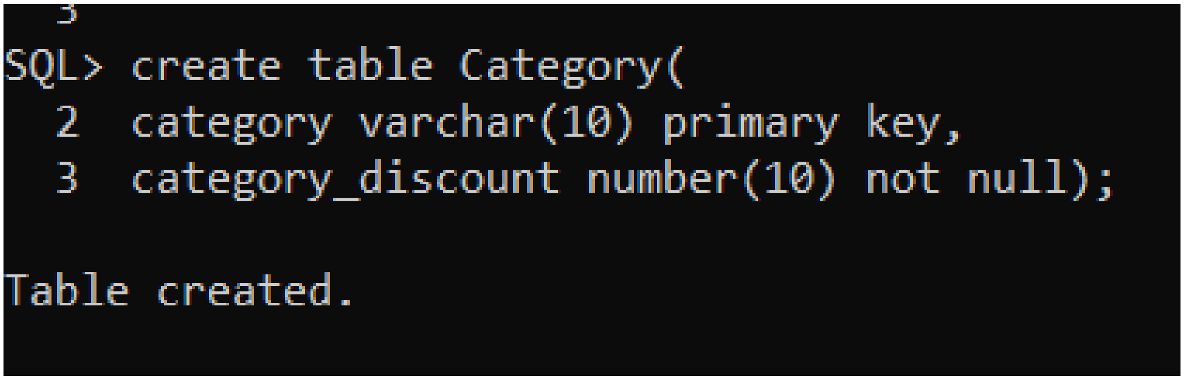
- describing order table



```
SQL> describe Orders;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ORDER_ID                                  NOT NULL VARCHAR2(10)
 ORDER_DATE                                NOT NULL DATE
 ORDER_TOTAL                               NOT NULL NUMBER(10)
 STAFF_ID                                  NOT NULL VARCHAR2(10)
```

Figure 9 describe order

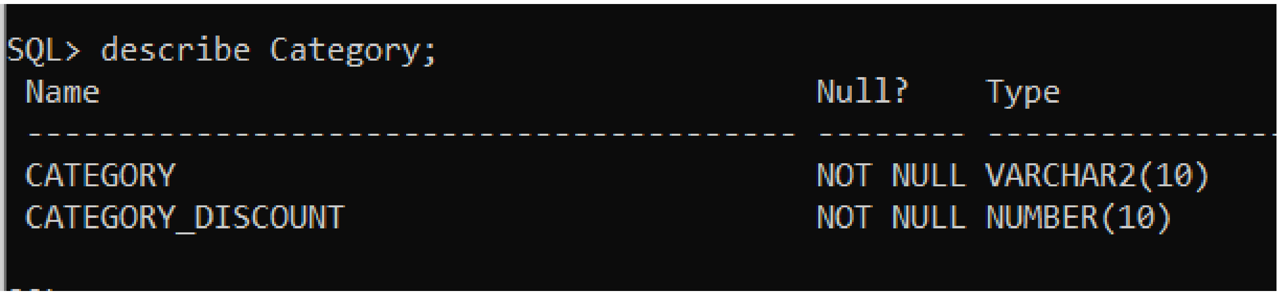
- **Category table**



```
  3
SQL> create table Category(
  2  category varchar(10) primary key,
  3  category_discount number(10) not null);

Table created.
```

Figure 10 Creating category table

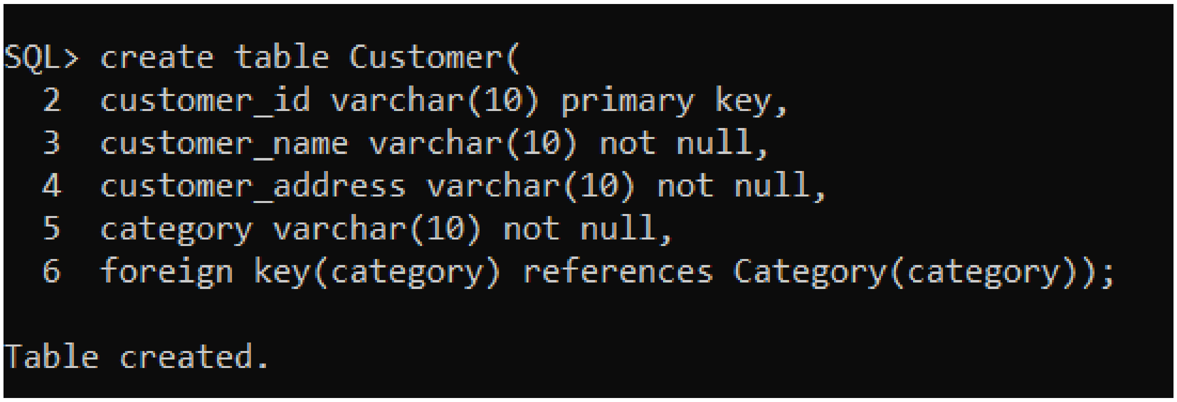
- describing category table



```
SQL> describe Category;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 CATEGORY                                  NOT NULL VARCHAR2(10)
 CATEGORY_DISCOUNT                         NOT NULL NUMBER(10)
```

Figure 11 describe category

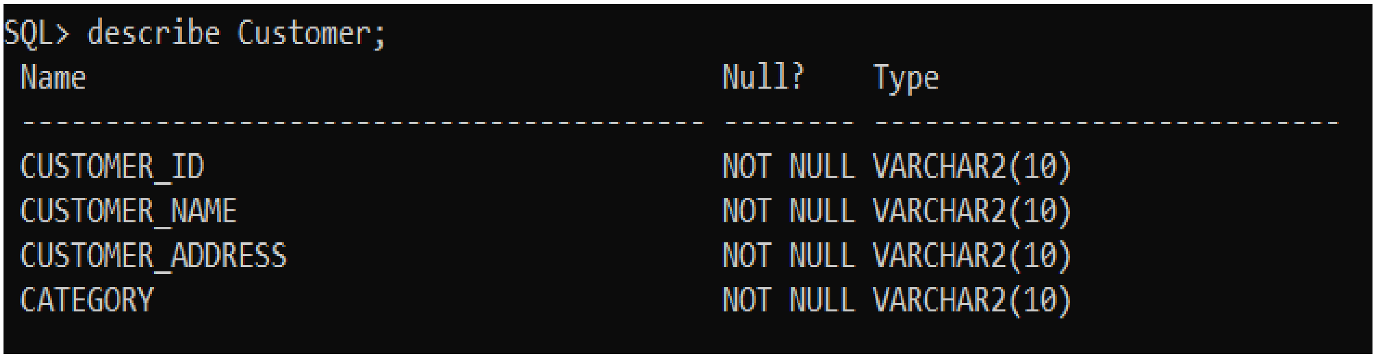
- **Customer table**



```
SQL> create table Customer(
     customer_id varchar(10) primary key,
     customer_name varchar(10) not null,
     customer_address varchar(10) not null,
     category varchar(10) not null,
     foreign key(category) references Category(category));

Table created.
```

Figure 12 Creating customer table

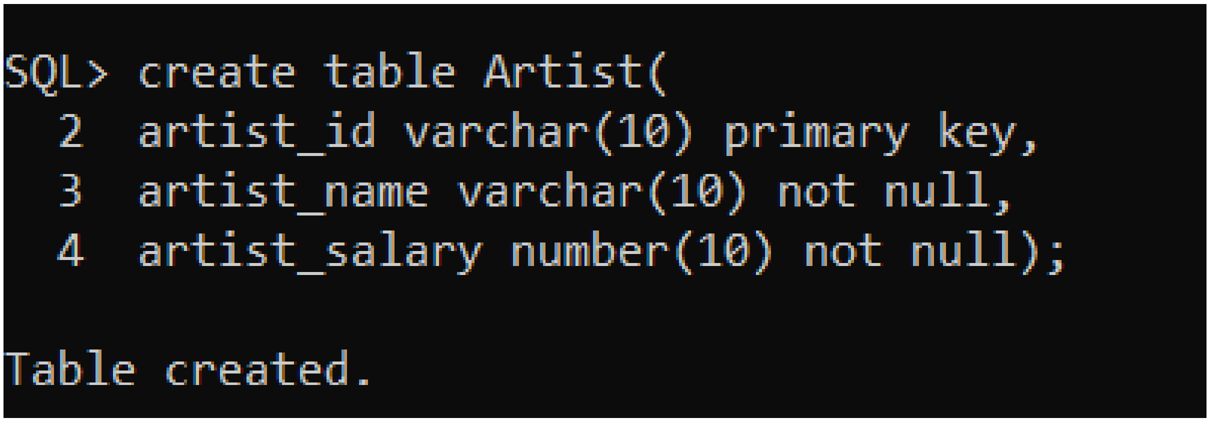
- describing customer table



```
SQL> describe Customer;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                               NOT NULL VARCHAR2(10)
 CUSTOMER_NAME                             NOT NULL VARCHAR2(10)
 CUSTOMER_ADDRESS                          NOT NULL VARCHAR2(10)
 CATEGORY                                  NOT NULL VARCHAR2(10)
```

Figure 13 describe customer

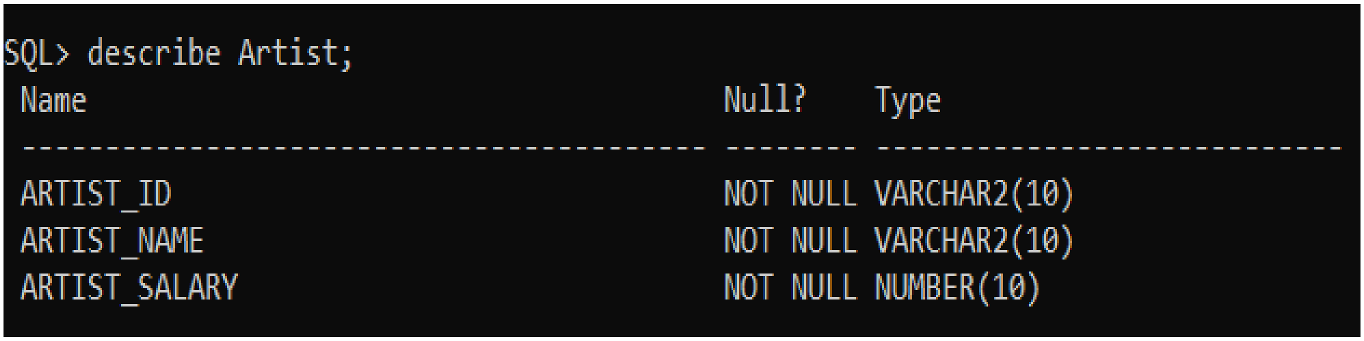
- **Artist table**



```
SQL> create table Artist(
  2  artist_id varchar(10) primary key,
  3  artist_name varchar(10) not null,
  4  artist_salary number(10) not null);

Table created.
```

Figure 14 Creating artist table

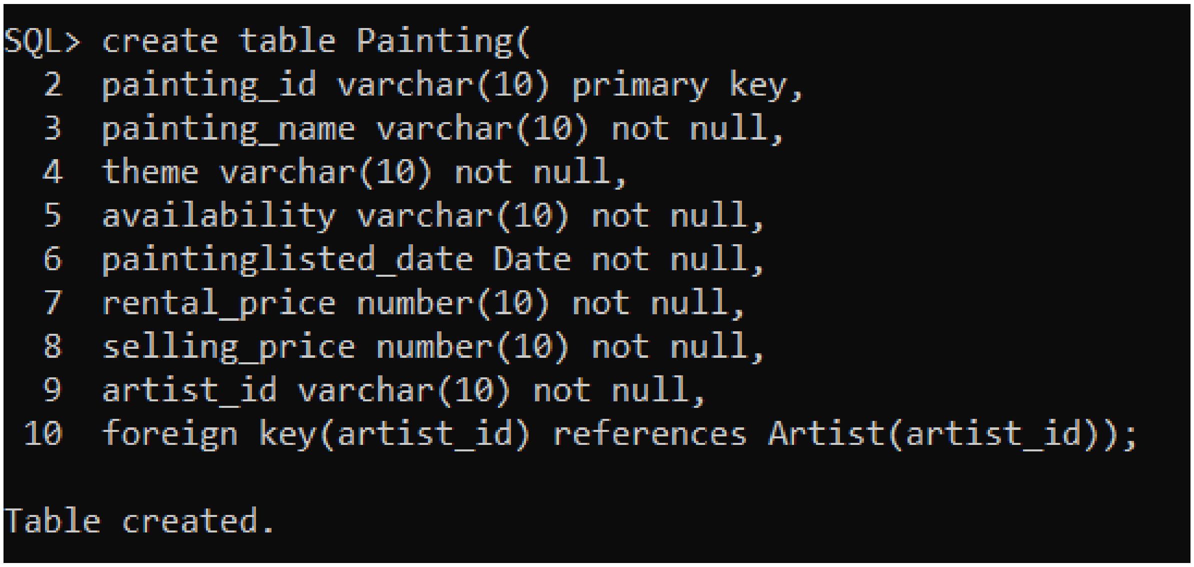
- describing artist table



```
SQL> describe Artist;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ARTIST_ID                                 NOT NULL VARCHAR2(10)
 ARTIST_NAME                               NOT NULL VARCHAR2(10)
 ARTIST_SALARY                             NOT NULL NUMBER(10)
```

Figure 15 describe artist

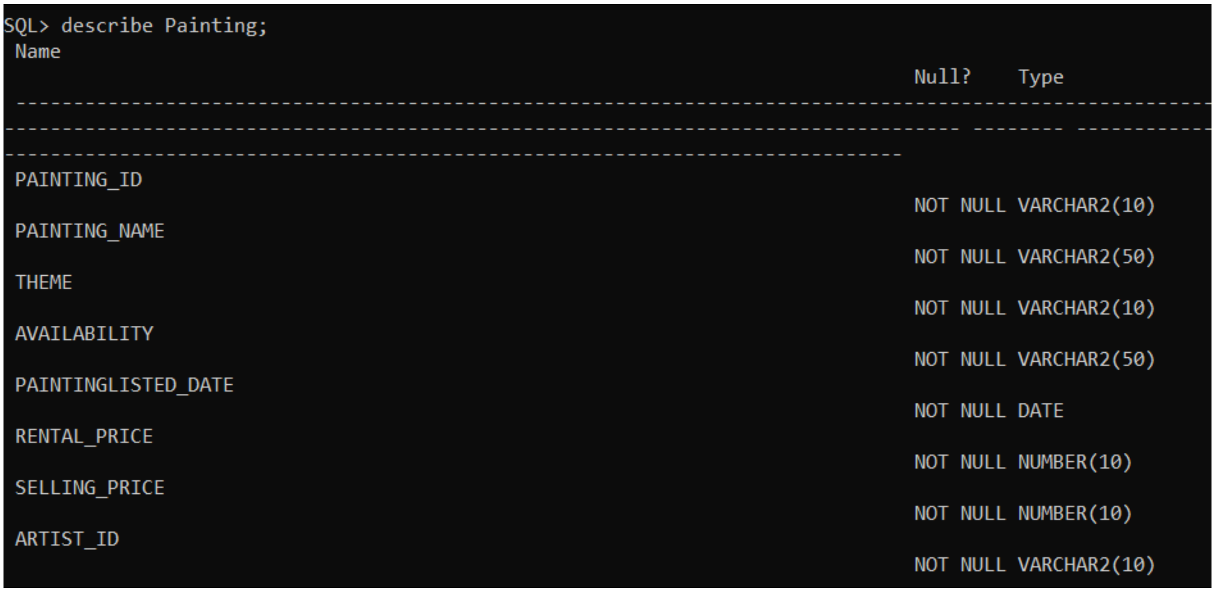
- **Panting table**



```
SQL> create table Painting(
painting_id varchar(10) primary key,
painting_name varchar(10) not null,
theme varchar(10) not null,
availability varchar(10) not null,
paintinglisted_date Date not null,
rental_price number(10) not null,
selling_price number(10) not null,
artist_id varchar(10) not null,
foreign key(artist_id) references Artist(artist_id));

Table created.
```

Figure 16 creating painting table

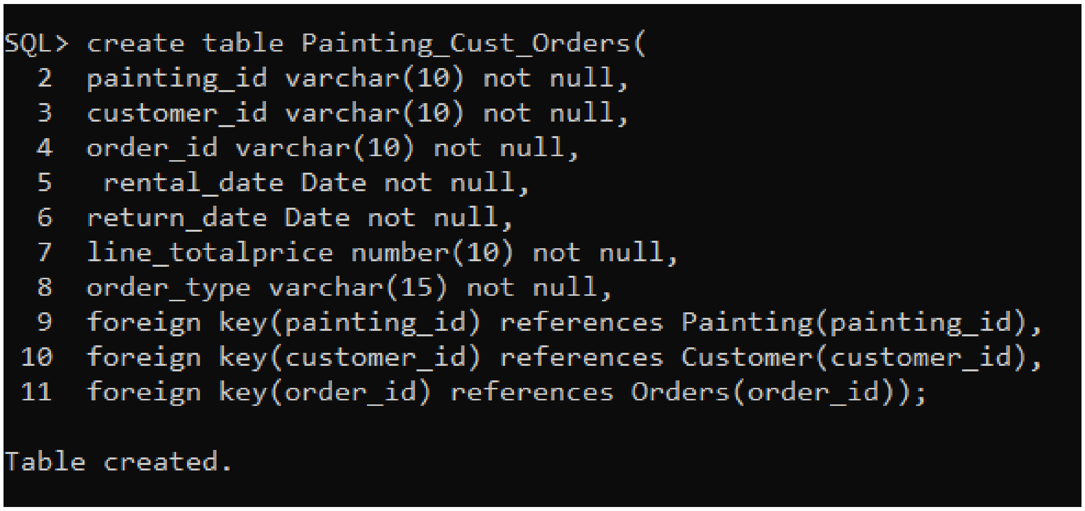
- describing painting table



```
SQL> describe Painting;
 Name
                                                                                  Null?    Type
 -----------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------ -------- ----------
-------------------------------------------------------------------------------
 PAINTING_ID
                                                                                  NOT NULL VARCHAR2(10)
 PAINTING_NAME
                                                                                  NOT NULL VARCHAR2(50)
 THEME
                                                                                  NOT NULL VARCHAR2(10)
 AVAILABILITY
                                                                                  NOT NULL VARCHAR2(50)
 PAINTINGLISTED_DATE
                                                                                  NOT NULL DATE
 RENTAL_PRICE
                                                                                  NOT NULL NUMBER(10)
 SELLING_PRICE
                                                                                  NOT NULL NUMBER(10)
 ARTIST_ID
                                                                                  NOT NULL VARCHAR2(10)
```

Figure 17 describe painting

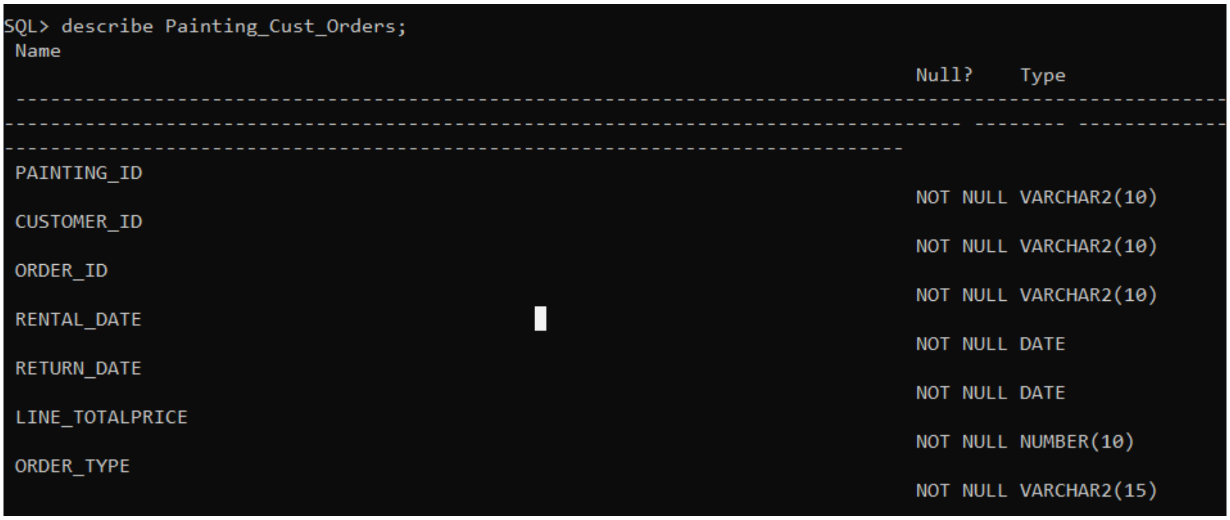
- **Painting_Cust_Orders tables**



```
SQL> create table Painting_Cust_Orders(
painting_id varchar(10) not null,
customer_id varchar(10) not null,
order_id varchar(10) not null,
 rental_date Date not null,
return_date Date not null,
line_totalprice number(10) not null,
order_type varchar(15) not null,
foreign key(painting_id) references Painting(painting_id),
foreign key(customer_id) references Customer(customer_id),
foreign key(order_id) references Orders(order_id));

Table created.
```

Figure 18 creating painting_cust_orders table

- describing Painting_Cust_Orders

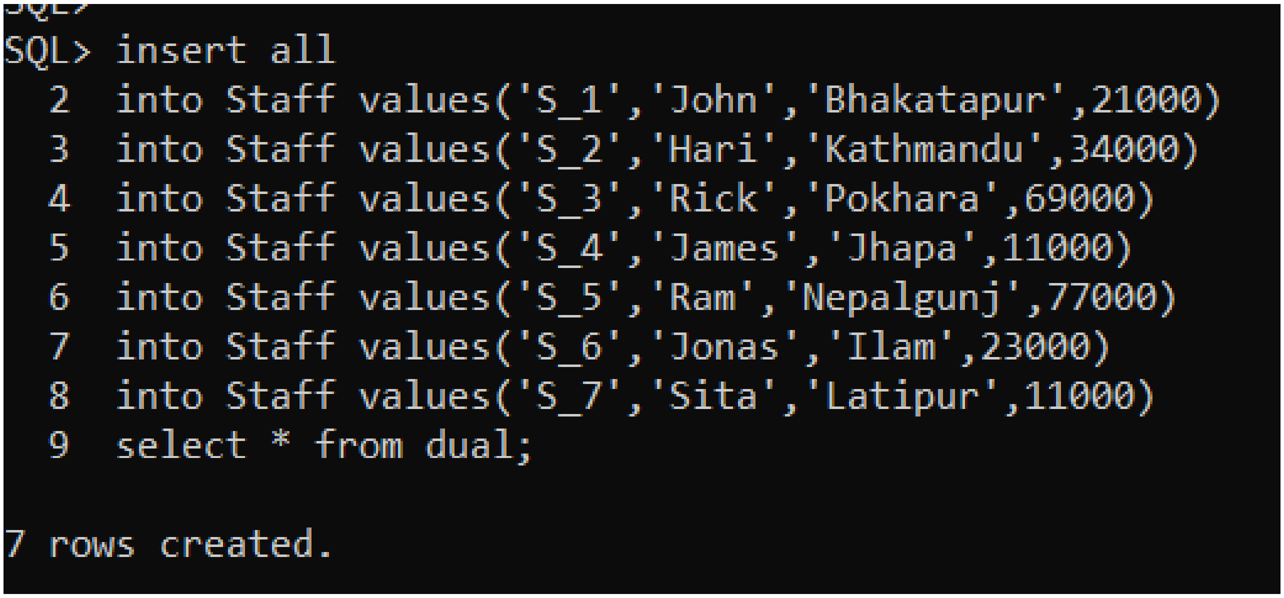


```
SQL> describe Painting_Cust_Orders;
 Name
                                                                              Null?    Type
 ------------------------------------------------------------------------------------------------------
 -------------------------------------------------------------------------------- -------- ------------
-----------------------------------------------------------------------------
 PAINTING_ID
                                                                              NOT NULL VARCHAR2(10)
 CUSTOMER_ID
                                                                              NOT NULL VARCHAR2(10)
 ORDER_ID
                                                                              NOT NULL VARCHAR2(10)
 RENTAL_DATE
                                                                              NOT NULL DATE
 RETURN_DATE
                                                                              NOT NULL DATE
 LINE_TOTALPRICE
                                                                              NOT NULL NUMBER(10)
 ORDER_TYPE
                                                                              NOT NULL VARCHAR2(15)
```

Figure 19 describe Painting_Cust_Orders

5.5 Inserting Queries

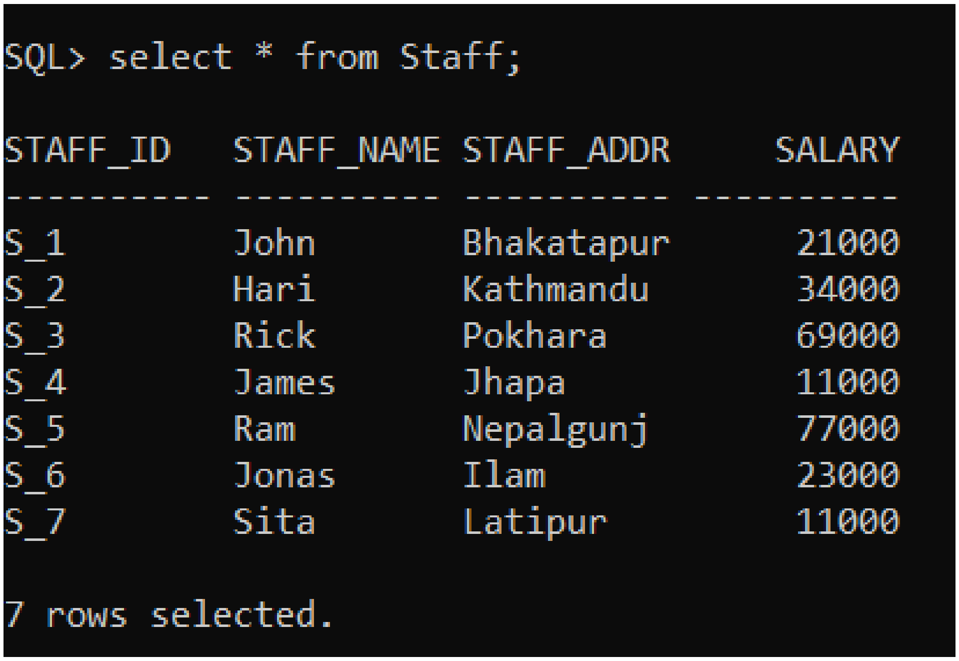
- Inserting data in **Staff table**



```
SQL>
SQL> insert all
  into Staff values('S_1','John','Bhakatapur',21000)
  into Staff values('S_2','Hari','Kathmandu',34000)
  into Staff values('S_3','Rick','Pokhara',69000)
  into Staff values('S_4','James','Jhapa',11000)
  into Staff values('S_5','Ram','Nepalgunj',77000)
  into Staff values('S_6','Jonas','Ilam',23000)
  into Staff values('S_7','Sita','Latipur',11000)
  select * from dual;

7 rows created.
```

Figure 20 Inserting values in staff table

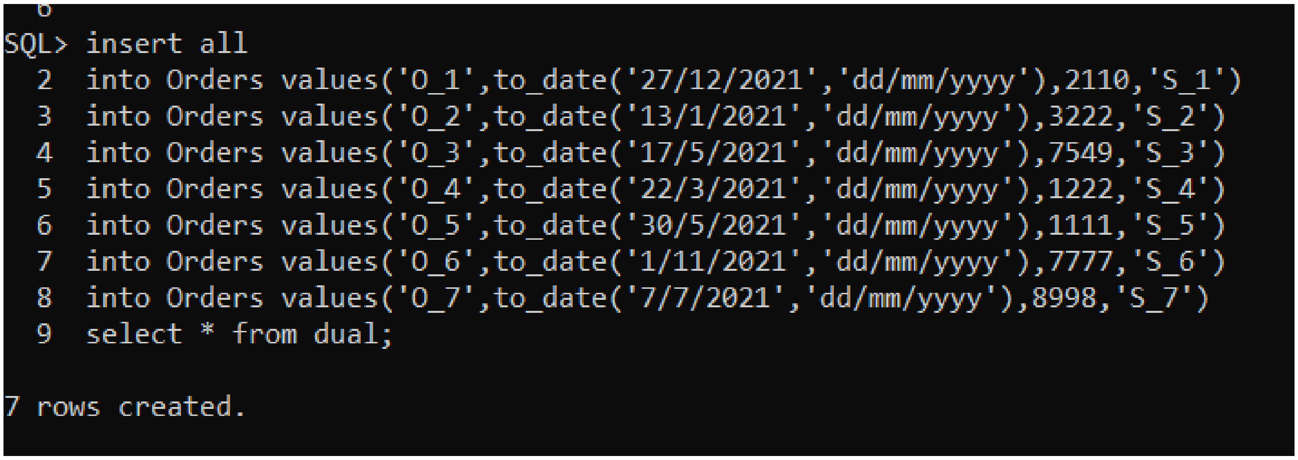
Showing Staff table detail



```
SQL> select * from Staff;

STAFF_ID   STAFF_NAME STAFF_ADDR     SALARY
---------- ---------- ---------- ----------
S_1        John       Bhakatapur      21000
S_2        Hari       Kathmandu       34000
S_3        Rick       Pokhara         69000
S_4        James      Jhapa           11000
S_5        Ram        Nepalgunj       77000
S_6        Jonas      Ilam            23000
S_7        Sita       Latipur         11000

7 rows selected.
```

Figure 21 Staff table detail

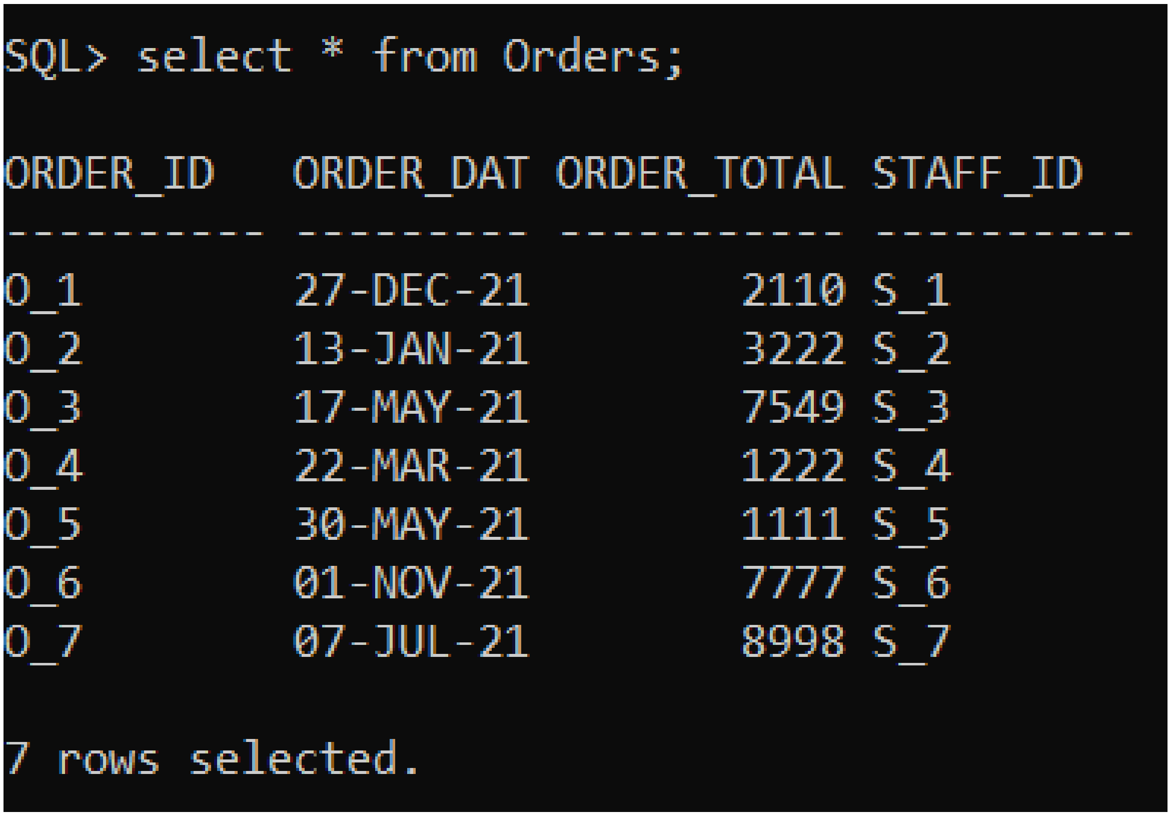
- Inserting data in **Orders table**



```
SQL> insert all
  2  into Orders values('O_1',to_date('27/12/2021','dd/mm/yyyy'),2110,'S_1')
  3  into Orders values('O_2',to_date('13/1/2021','dd/mm/yyyy'),3222,'S_2')
  4  into Orders values('O_3',to_date('17/5/2021','dd/mm/yyyy'),7549,'S_3')
  5  into Orders values('O_4',to_date('22/3/2021','dd/mm/yyyy'),1222,'S_4')
  6  into Orders values('O_5',to_date('30/5/2021','dd/mm/yyyy'),1111,'S_5')
  7  into Orders values('O_6',to_date('1/11/2021','dd/mm/yyyy'),7777,'S_6')
  8  into Orders values('O_7',to_date('7/7/2021','dd/mm/yyyy'),8998,'S_7')
  9  select * from dual;

7 rows created.
```

Figure 22 Inserting values in orders table

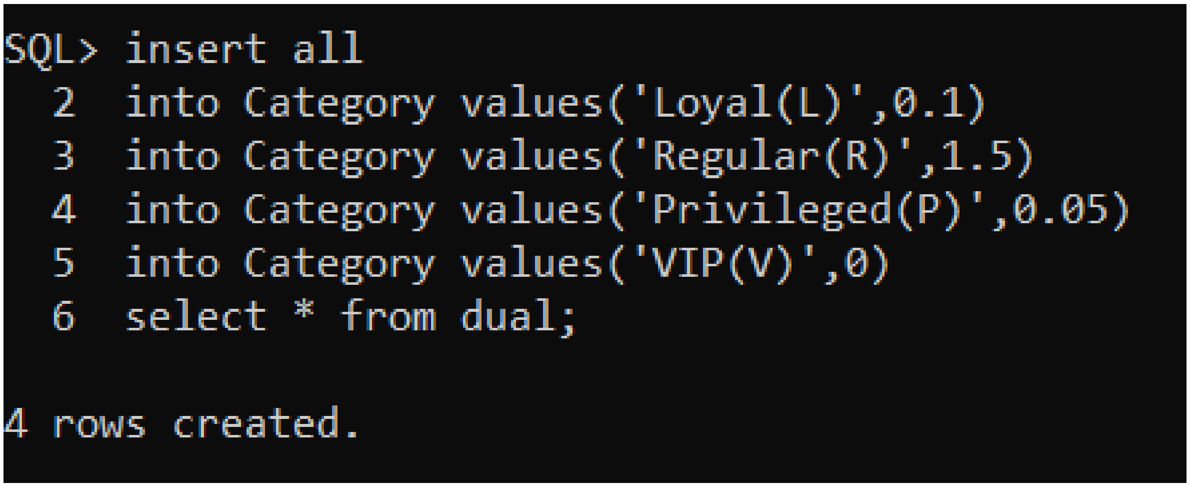
Showing Orders table detail



```
SQL> select * from Orders;

ORDER_ID   ORDER_DAT ORDER_TOTAL STAFF_ID
---------- --------- ----------- ----------
O_1        27-DEC-21        2110 S_1
O_2        13-JAN-21        3222 S_2
O_3        17-MAY-21        7549 S_3
O_4        22-MAR-21        1222 S_4
O_5        30-MAY-21        1111 S_5
O_6        01-NOV-21        7777 S_6
O_7        07-JUL-21        8998 S_7

7 rows selected.
```

Figure 23 Orders table detail

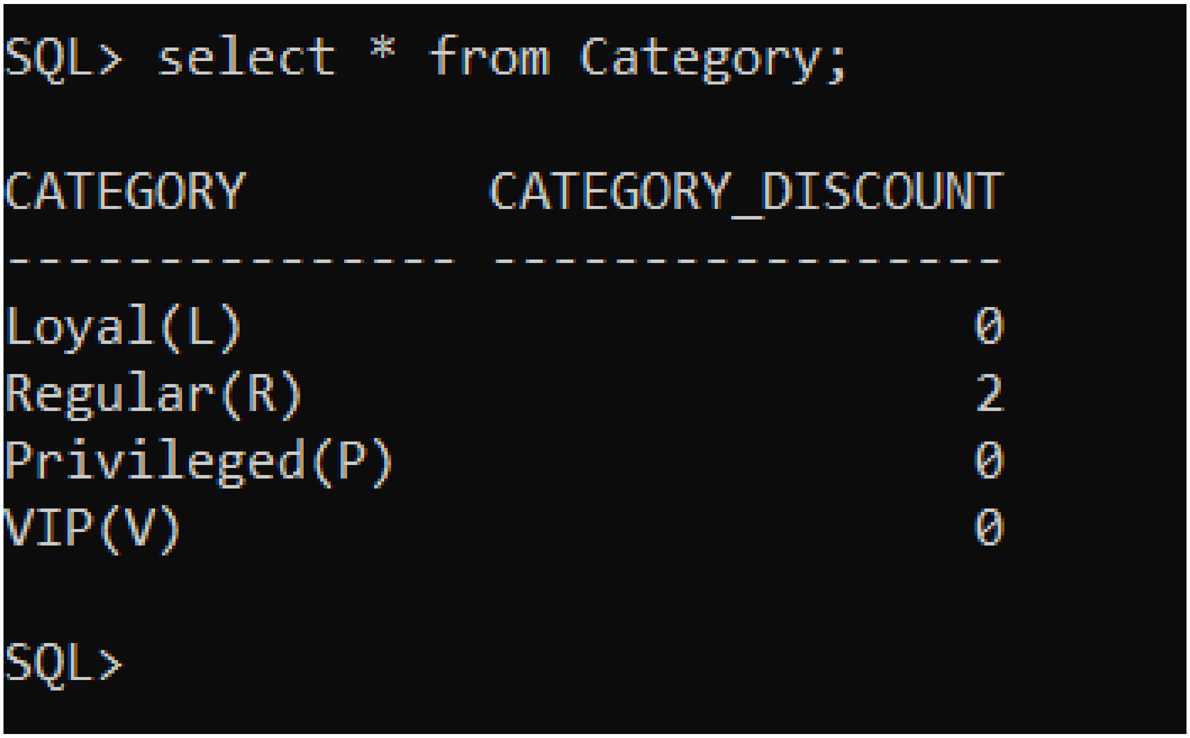
- Inserting data in **Category table**



```
SQL> insert all
  2  into Category values('Loyal(L)',0.1)
  3  into Category values('Regular(R)',1.5)
  4  into Category values('Privileged(P)',0.05)
  5  into Category values('VIP(V)',0)
  6  select * from dual;

4 rows created.
```

Figure 24 Inserting values in category table

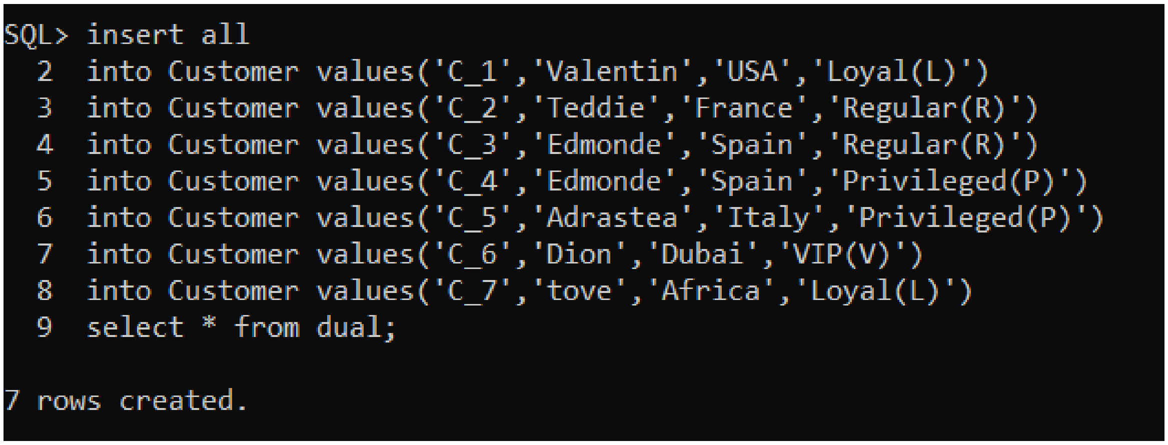
Showing Category table detail



```
SQL> select * from Category;

CATEGORY          CATEGORY_DISCOUNT
---------------- -----------------
Loyal(L)                         0
Regular(R)                       2
Privileged(P)                    0
VIP(V)                           0

SQL>
```

Figure 25 category table detail

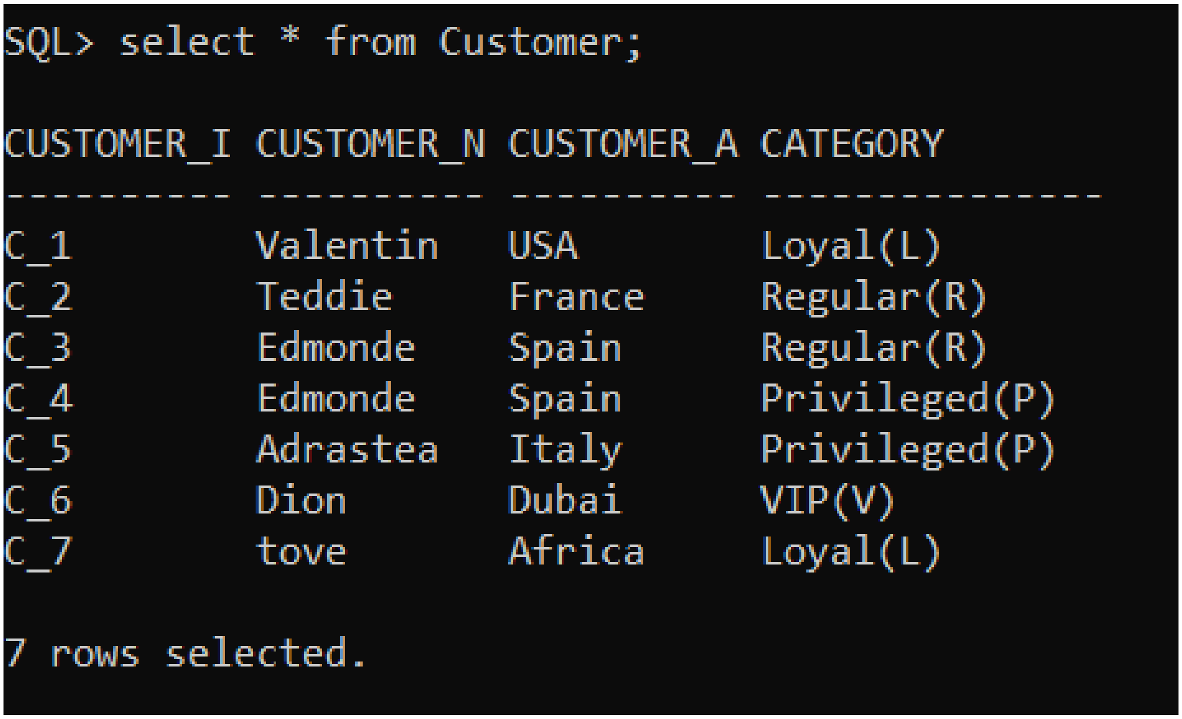
- Inserting data in **Customer table**



```
SQL> insert all
     into Customer values('C_1','Valentin','USA','Loyal(L)')
     into Customer values('C_2','Teddie','France','Regular(R)')
     into Customer values('C_3','Edmonde','Spain','Regular(R)')
     into Customer values('C_4','Edmonde','Spain','Privileged(P)')
     into Customer values('C_5','Adrastea','Italy','Privileged(P)')
     into Customer values('C_6','Dion','Dubai','VIP(V)')
     into Customer values('C_7','tove','Africa','Loyal(L)')
     select * from dual;

7 rows created.
```

Figure 26 inserting values in customer table

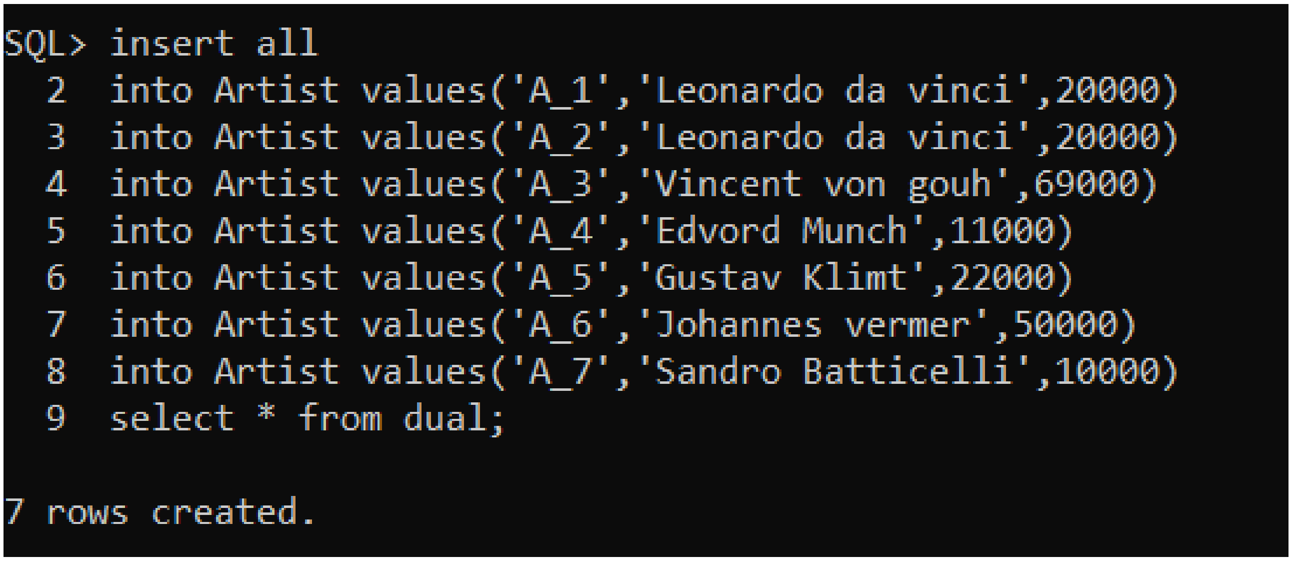
Showing Customer table detail



```
SQL> select * from Customer;

CUSTOMER_I CUSTOMER_N CUSTOMER_A CATEGORY
---------- ---------- ---------- ---------------
C_1        Valentin   USA        Loyal(L)
C_2        Teddie     France     Regular(R)
C_3        Edmonde    Spain      Regular(R)
C_4        Edmonde    Spain      Privileged(P)
C_5        Adrastea   Italy      Privileged(P)
C_6        Dion       Dubai      VIP(V)
C_7        tove       Africa     Loyal(L)

7 rows selected.
```

Figure 27 customer table detail

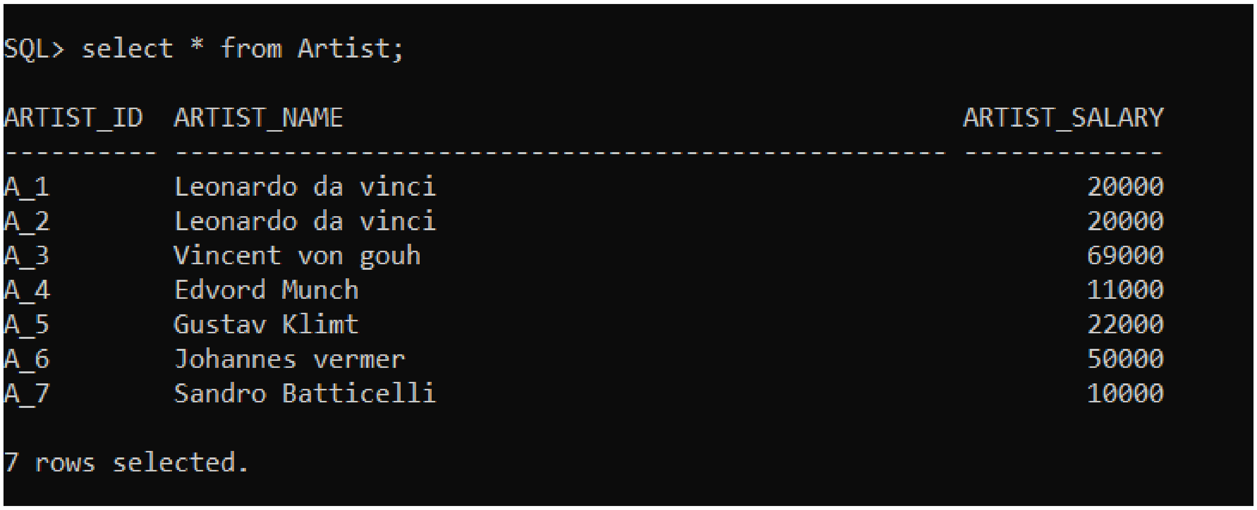
- Inserting data in **Artist table**



```
SQL> insert all
  2  into Artist values('A_1','Leonardo da vinci',20000)
  3  into Artist values('A_2','Leonardo da vinci',20000)
  4  into Artist values('A_3','Vincent von gouh',69000)
  5  into Artist values('A_4','Edvord Munch',11000)
  6  into Artist values('A_5','Gustav Klimt',22000)
  7  into Artist values('A_6','Johannes vermer',50000)
  8  into Artist values('A_7','Sandro Batticelli',10000)
  9  select * from dual;

7 rows created.
```

Figure 28 Inserting values in Customer table

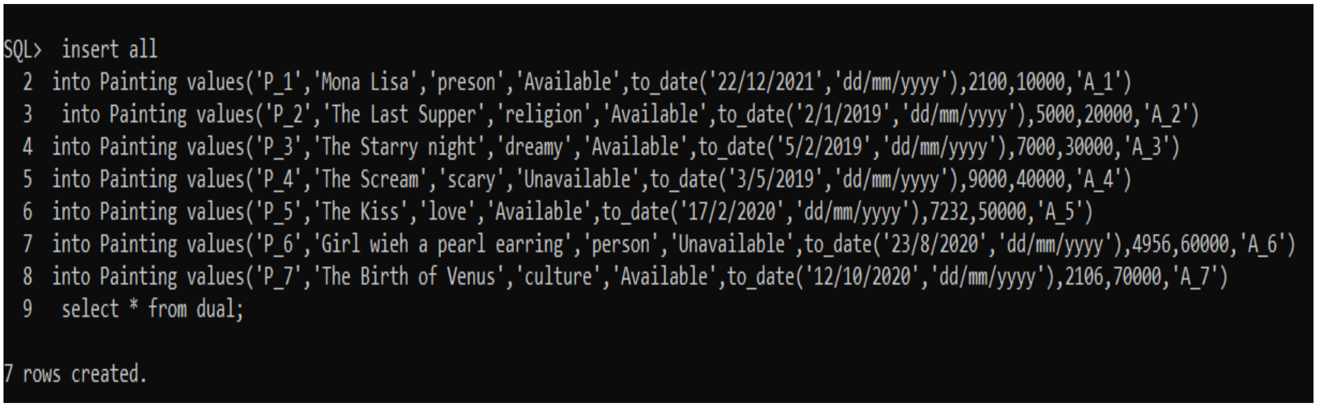
Showing Artist table detail



```
SQL> select * from Artist;

ARTIST_ID  ARTIST_NAME                                          ARTIST_SALARY
---------- -------------------------------------------------- -------------
A_1        Leonardo da vinci                                          20000
A_2        Leonardo da vinci                                          20000
A_3        Vincent von gouh                                           69000
A_4        Edvord Munch                                               11000
A_5        Gustav Klimt                                               22000
A_6        Johannes vermer                                            50000
A_7        Sandro Batticelli                                          10000

7 rows selected.
```

Figure 29 Artist table details

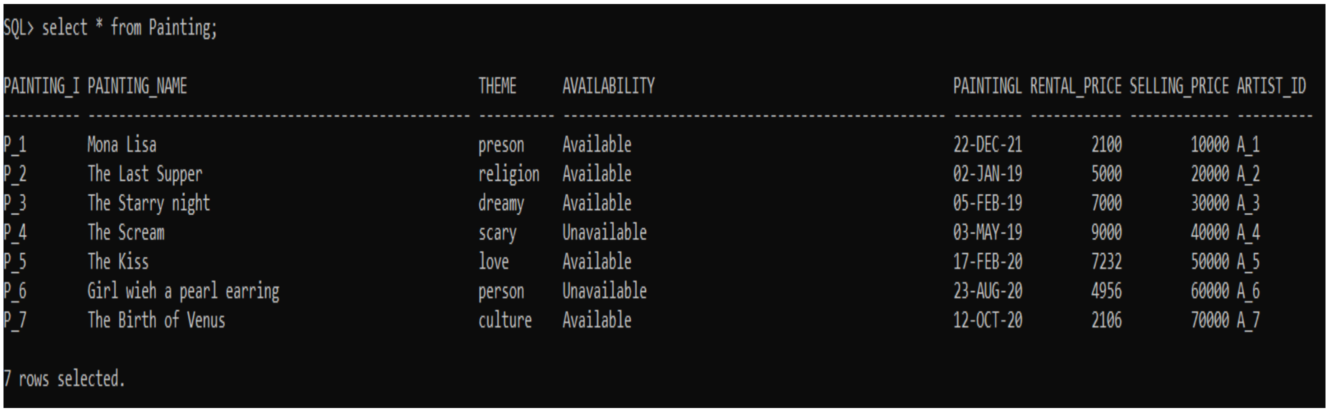
- Inserting data in **painting table**



```
SQL>  insert all
 into Painting values('P_1','Mona Lisa','preson','Available',to_date('22/12/2021','dd/mm/yyyy'),2100,10000,'A_1')
  into Painting values('P_2','The Last Supper','religion','Available',to_date('2/1/2019','dd/mm/yyyy'),5000,20000,'A_2')
 into Painting values('P_3','The Starry night','dreamy','Available',to_date('5/2/2019','dd/mm/yyyy'),7000,30000,'A_3')
 into Painting values('P_4','The Scream','scary','Unavailable',to_date('3/5/2019','dd/mm/yyyy'),9000,40000,'A_4')
 into Painting values('P_5','The Kiss','love','Available',to_date('17/2/2020','dd/mm/yyyy'),7232,50000,'A_5')
 into Painting values('P_6','Girl wieh a pearl earring','person','Unavailable',to_date('23/8/2020','dd/mm/yyyy'),4956,60000,'A_6')
 into Painting values('P_7','The Birth of Venus','culture','Available',to_date('12/10/2020','dd/mm/yyyy'),2106,70000,'A_7')
  select * from dual;

7 rows created.
```

Figure 30 inserting values in painting table

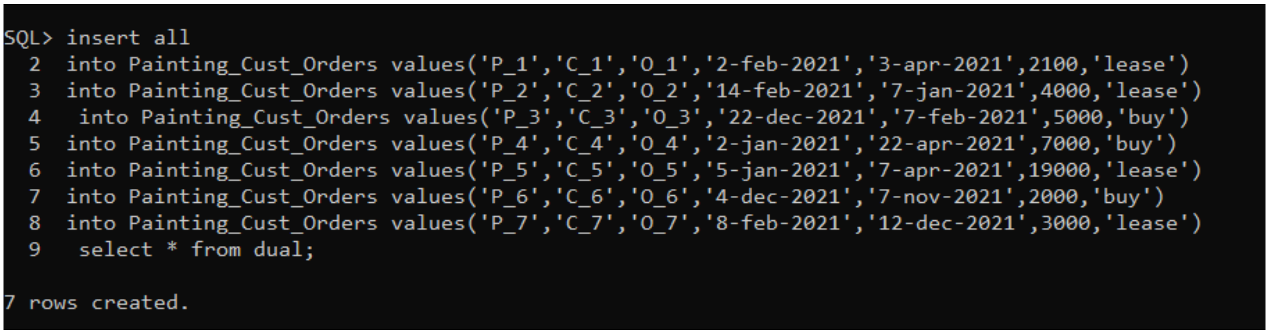
Showing Artist table detail



```
SQL> select * from Painting;

PAINTING_I PAINTING_NAME                                  THEME     AVAILABILITY                                        PAINTINGL RENTAL_PRICE SELLING_PRICE ARTIST_ID
---------- ---------------------------------------------- --------- --------------------------------------------------- --------- ------------ ------------- ---------
P_1        Mona Lisa                                      preson    Available                                           22-DEC-21         2100         10000 A_1
P_2        The Last Supper                                religion  Available                                           02-JAN-19         5000         20000 A_2
P_3        The Starry night                               dreamy    Available                                           05-FEB-19         7000         30000 A_3
P_4        The Scream                                     scary     Unavailable                                         03-MAY-19         9000         40000 A_4
P_5        The Kiss                                       love      Available                                           17-FEB-20         7232         50000 A_5
P_6        Girl wieh a pearl earring                      person    Unavailable                                         23-AUG-20         4956         60000 A_6
P_7        The Birth of Venus                             culture   Available                                           12-OCT-20         2106         70000 A_7

7 rows selected.
```

Figure 31 painting table detail

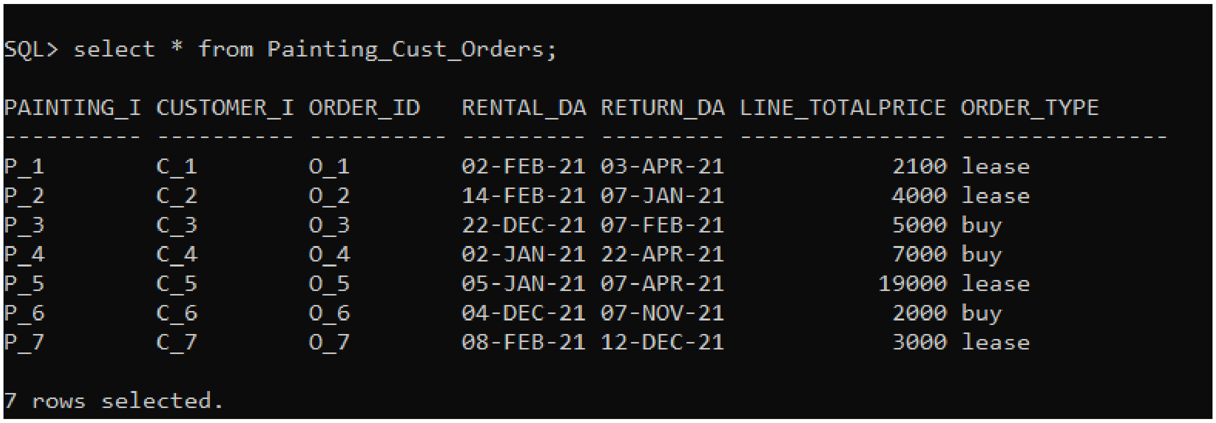
- Inserting data in **Painting_Cust_Orders table**



```
SQL> insert all
into Painting_Cust_Orders values('P_1','C_1','O_1','2-feb-2021','3-apr-2021',2100,'lease')
into Painting_Cust_Orders values('P_2','C_2','O_2','14-feb-2021','7-jan-2021',4000,'lease')
 into Painting_Cust_Orders values('P_3','C_3','O_3','22-dec-2021','7-feb-2021',5000,'buy')
into Painting_Cust_Orders values('P_4','C_4','O_4','2-jan-2021','22-apr-2021',7000,'buy')
into Painting_Cust_Orders values('P_5','C_5','O_5','5-jan-2021','7-apr-2021',19000,'lease')
into Painting_Cust_Orders values('P_6','C_6','O_6','4-dec-2021','7-nov-2021',2000,'buy')
into Painting_Cust_Orders values('P_7','C_7','O_7','8-feb-2021','12-dec-2021',3000,'lease')
 select * from dual;

7 rows created.
```

Figure 32 inserting values Painting_Cust_Orders table

showing Painting_Cust_Orders table detail

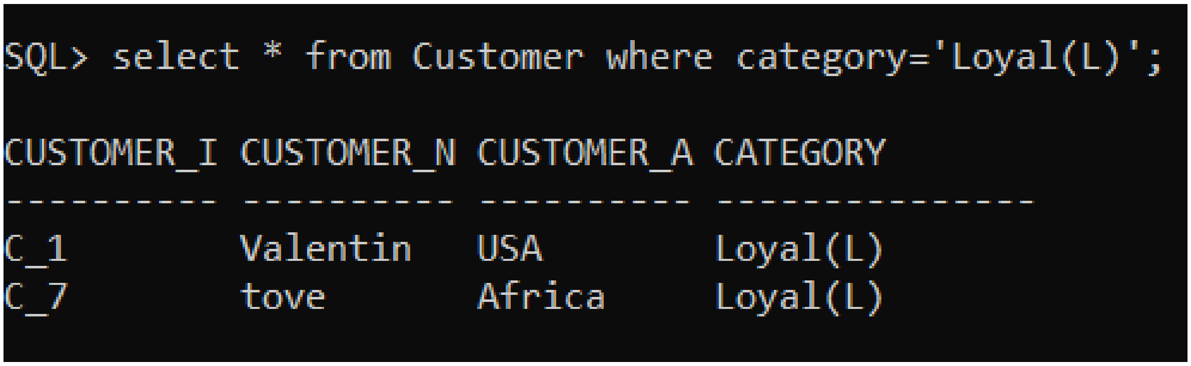


```
SQL> select * from Painting_Cust_Orders;

PAINTING_I CUSTOMER_I ORDER_ID   RENTAL_DA RETURN_DA LINE_TOTALPRICE ORDER_TYPE
---------- ---------- ---------- --------- --------- --------------- ---------------
P_1        C_1        O_1        02-FEB-21 03-APR-21            2100 lease
P_2        C_2        O_2        14-FEB-21 07-JAN-21            4000 lease
P_3        C_3        O_3        22-DEC-21 07-FEB-21            5000 buy
P_4        C_4        O_4        02-JAN-21 22-APR-21            7000 buy
P_5        C_5        O_5        05-JAN-21 07-APR-21           19000 lease
P_6        C_6        O_6        04-DEC-21 07-NOV-21            2000 buy
P_7        C_7        O_7        08-FEB-21 12-DEC-21            3000 lease

7 rows selected.
```

Figure 33 Painting_Cust_Orders table detail

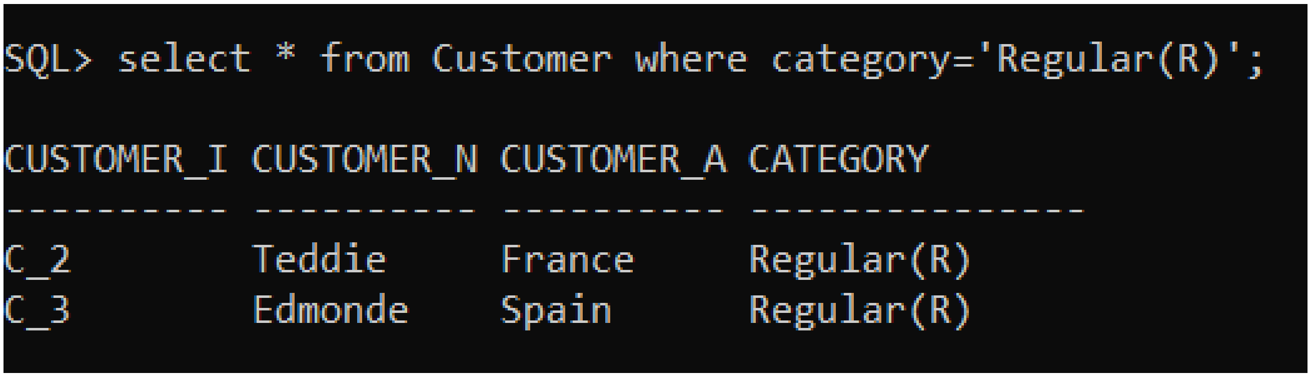
6. Information query

6.1 List all customers according to category



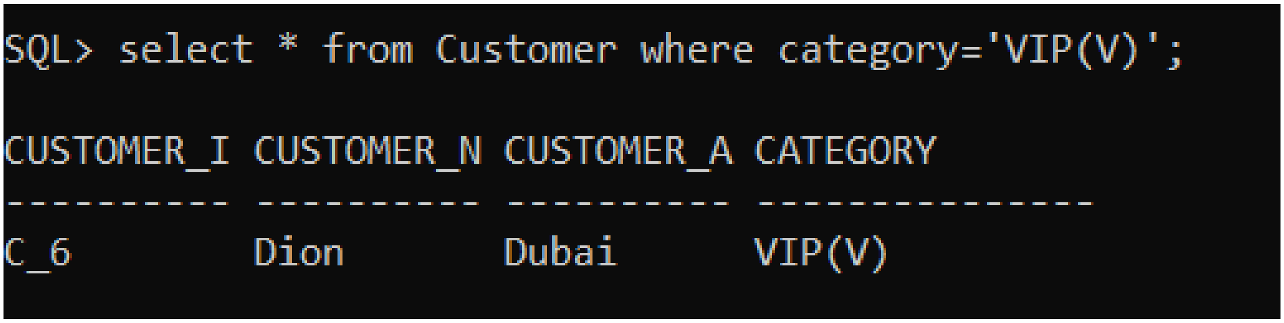
```
SQL> select * from Customer where category='Loyal(L)';

CUSTOMER_I CUSTOMER_N CUSTOMER_A CATEGORY
---------- ---------- ---------- ---------------
C_1        Valentin   USA        Loyal(L)
C_7        tove       Africa     Loyal(L)
```

Figure 34 loyal customer category



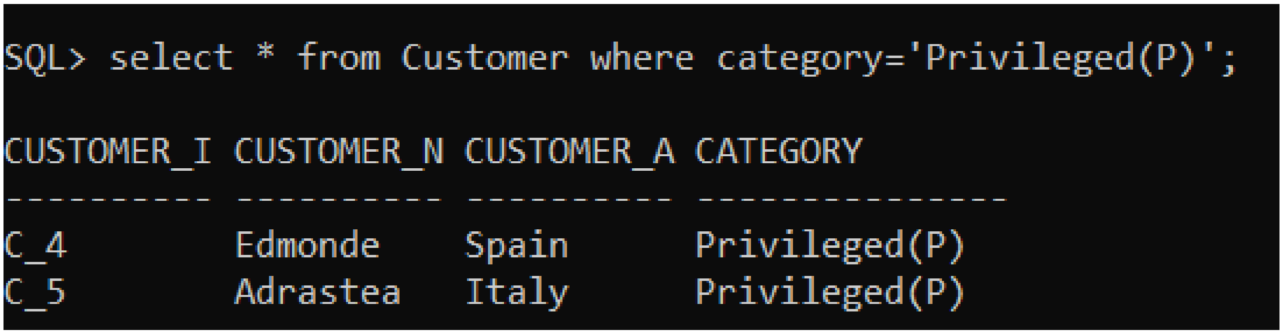
```
SQL> select * from Customer where category='Regular(R)';

CUSTOMER_I CUSTOMER_N CUSTOMER_A CATEGORY
---------- ---------- ---------- ---------------
C_2        Teddie     France     Regular(R)
C_3        Edmonde    Spain      Regular(R)
```

Figure 35 regular customer category



```
SQL> select * from Customer where category='VIP(V)';

CUSTOMER_I CUSTOMER_N CUSTOMER_A CATEGORY
---------- ---------- ---------- ---------------
C_6        Dion       Dubai      VIP(V)
```

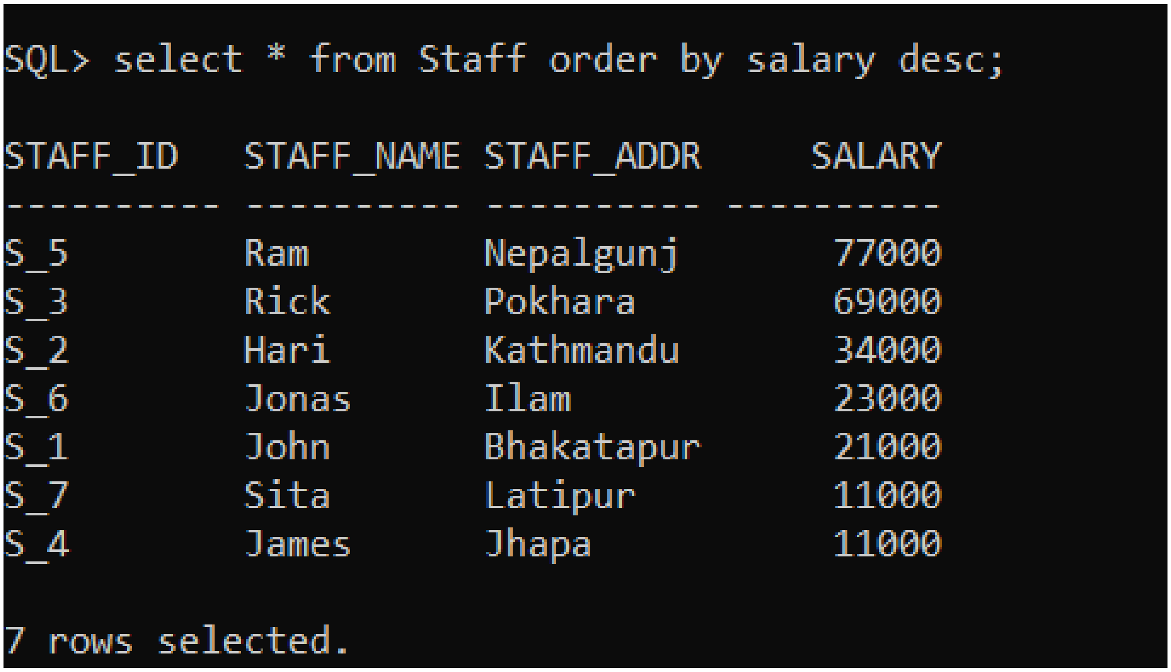
Figure 36 vip customer category



```
SQL> select * from Customer where category='Privileged(P)';

CUSTOMER_I CUSTOMER_N CUSTOMER_A CATEGORY
---------- ---------- ---------- ---------------
C_4        Edmonde    Spain      Privileged(P)
C_5        Adrastea   Italy      Privileged(P)
```

Figure 37 privileged customer category

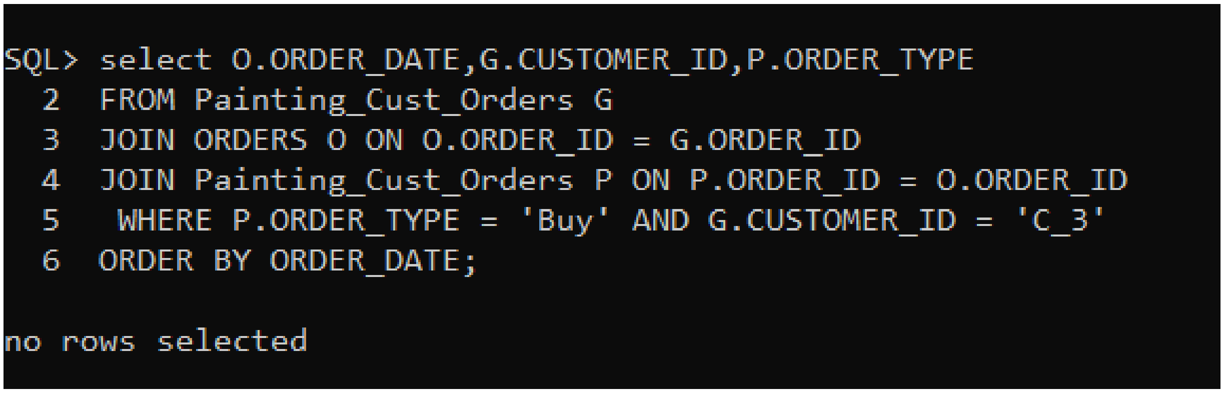
6.2 Show total staff in Masterpieces Limited sorted by higher salary.



```
SQL> select * from Staff order by salary desc;

STAFF_ID   STAFF_NAME STAFF_ADDR     SALARY
---------- ---------- ---------- ----------
S_5        Ram        Nepalgunj       77000
S_3        Rick       Pokhara         69000
S_2        Hari       Kathmandu       34000
S_6        Jonas      Ilam            23000
S_1        John       Bhakatapur      21000
S_7        Sita       Latipur         11000
S_4        James      Jhapa           11000

7 rows selected.
```

Figure 38 total staff in Masterpieces Limited sorted by higher salary

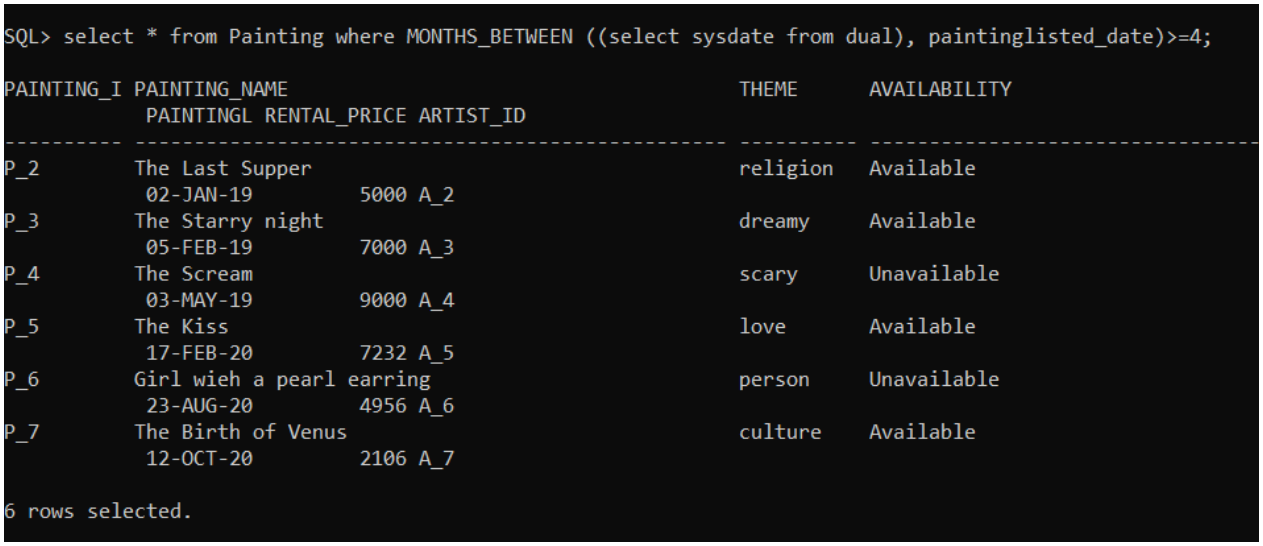
6.3 Show paintings leased before and currently by any one customer.



```
SQL> select O.ORDER_DATE,G.CUSTOMER_ID,P.ORDER_TYPE
     FROM Painting_Cust_Orders G
     JOIN ORDERS O ON O.ORDER_ID = G.ORDER_ID
     JOIN Painting_Cust_Orders P ON P.ORDER_ID = O.ORDER_ID
      WHERE P.ORDER_TYPE = 'Buy' AND G.CUSTOMER_ID = 'C_3'
     ORDER BY ORDER_DATE;

no rows selected
```

Figure 39 leach before and current by any one customer

6.5 List all paintings that have been returned to the owner

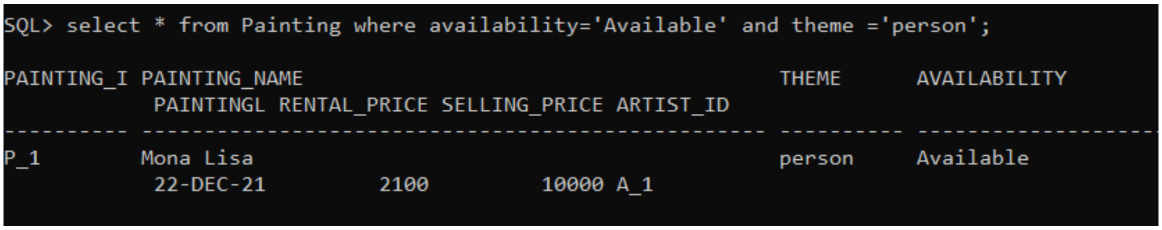


```
SQL> select * from Painting where MONTHS_BETWEEN ((select sysdate from dual), paintinglisted_date)>=4;

PAINTING_I PAINTING_NAME                                      THEME      AVAILABILITY
            PAINTINGL RENTAL_PRICE ARTIST_ID
---------- -------------------------------------------------- ---------- -----------------------------------
P_2        The Last Supper                                    religion   Available
            02-JAN-19         5000 A_2
P_3        The Starry night                                   dreamy     Available
            05-FEB-19         7000 A_3
P_4        The Scream                                         scary      Unavailable
            03-MAY-19         9000 A_4
P_5        The Kiss                                           love       Available
            17-FEB-20         7232 A_5
P_6        Girl wieh a pearl earring                          person     Unavailable
            23-AUG-20         4956 A_6
P_7        The Birth of Venus                                 culture    Available
            12-OCT-20         2106 A_7

6 rows selected.
```

Figure 40 All paintings that have been returned to the owner

7. Transaction Querying

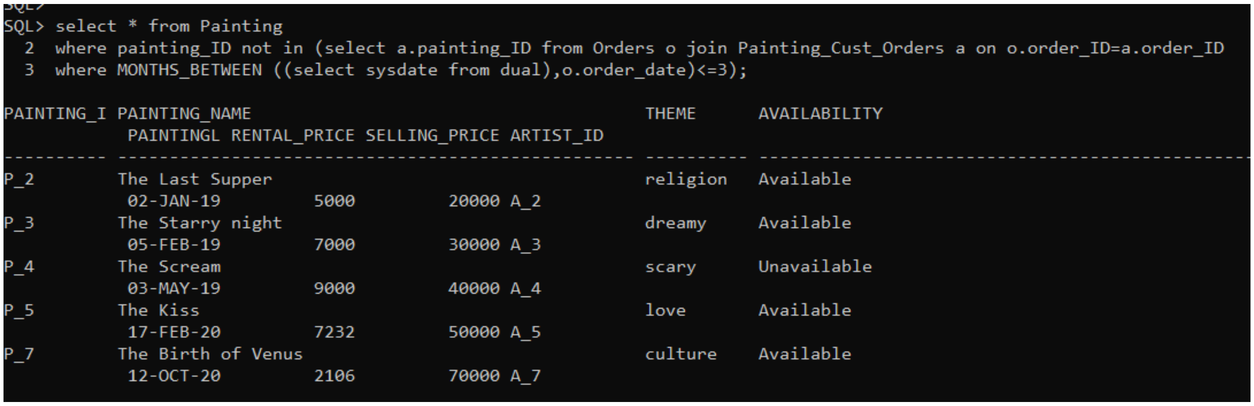
7.1 List the number of paintings available for rent according to category.



```
SQL> select * from Painting where availability='Available' and theme ='person';

PAINTING_I PAINTING_NAME                                      THEME      AVAILABILITY
           PAINTINGL RENTAL_PRICE SELLING_PRICE ARTIST_ID
---------- -------------------------------------------------- ---------- --------------------
P_1        Mona Lisa                                          person     Available
           22-DEC-21         2100         10000 A_1
```

Figure 41 available painting

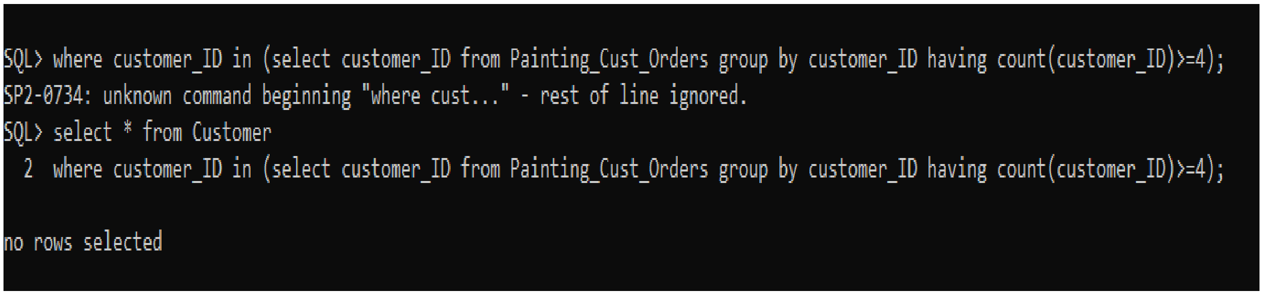
7.2 List the details of paintings that have not been leased within three months



```
SQL>
SQL> select * from Painting
  2  where painting_ID not in (select a.painting_ID from Orders o join Painting_Cust_Orders a on o.order_ID=a.order_ID
  3  where MONTHS_BETWEEN ((select sysdate from dual),o.order_date)<=3);

PAINTING_I PAINTING_NAME                                      THEME      AVAILABILITY
           PAINTINGL RENTAL_PRICE SELLING_PRICE ARTIST_ID
---------- -------------------------------------------------- ---------- --------------------------------------------
P_2        The Last Supper                                    religion   Available
           02-JAN-19         5000         20000 A_2
P_3        The Starry night                                   dreamy     Available
           05-FEB-19         7000         30000 A_3
P_4        The Scream                                         scary      Unavailable
           03-MAY-19         9000         40000 A_4
P_5        The Kiss                                           love       Available
           17-FEB-20         7232         50000 A_5
P_7        The Birth of Venus                                 culture    Available
           12-OCT-20         2106         70000 A_7
```

Figure 42 leased within three month

7.3 List the details of customers who have leased the painting more than four times.



```
SQL> where customer_ID in (select customer_ID from Painting_Cust_Orders group by customer_ID having count(customer_ID)>=4);
SP2-0734: unknown command beginning "where cust..." - rest of line ignored.
SQL> select * from Customer
  2  where customer_ID in (select customer_ID from Painting_Cust_Orders group by customer_ID having count(customer_ID)>=4);

no rows selected
```

Figure 43 leased painting more than 4 times

8. Critical Evaluation

8.1 Critical Evaluation of module

Through this module we have learned a lot about different components of database data storage strategies. Prior to normalization, the database for the masterpiece painting was successfully finished, including the proper introduction, goals and objectives, business activities, business rules, initial data dictionary, and first ER diagram. This has provided us with a sufficient understanding of data dictionaries, ER diagrams, queries, and other topics. The final data dictionary and ER diagram, as well as screenshots for creating a user, providing connecting to the user, creating a table, entering data, and running various queries. Database is important in real world life so it is used in our modules to teach us from which we can easily do the work in real world life. Now in upcoming time it can help us in different fields. Many regards to our module teacher who had helped in this database module. And helped us in the time of need for our module and coursework.

8.2 Critical Assessment of coursework

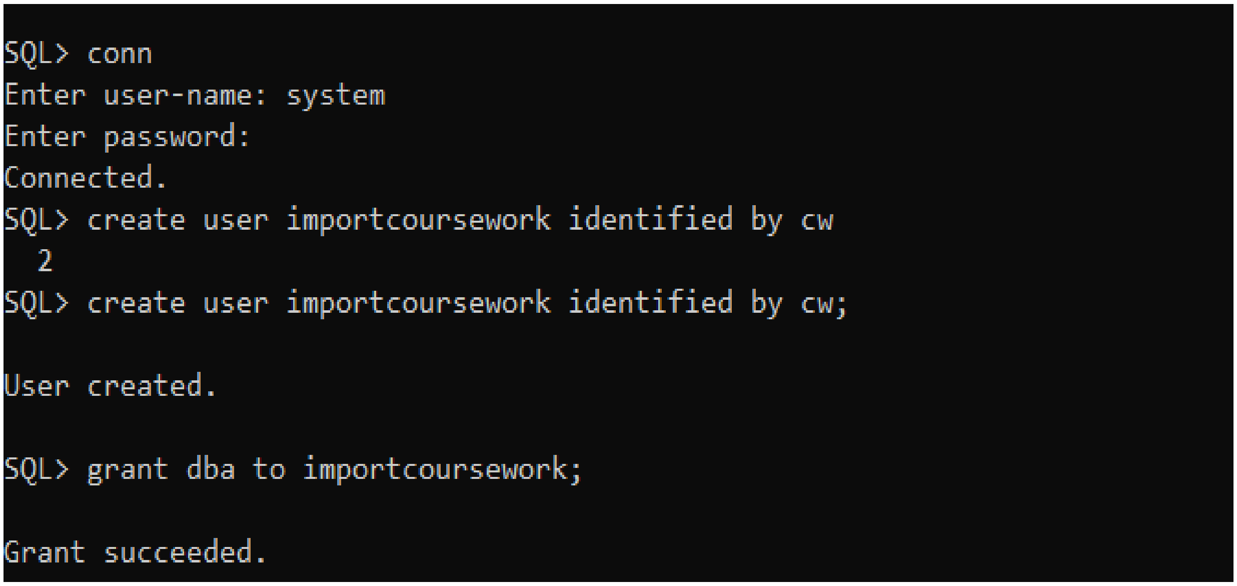
In this coursework a case scenario of a real-life internet business which contains to create a database for Mr. Steve's newly created painting firm, "Masterpiece Limited." Information for developing a database which handles the details about Customer, painter, Artist, Order and Staff details. With some specifications in the case study. I had to design a Initial ERD, need to implement normalization, Design the Final ERD, Create User and Table in Oracle, Populate Data in the Tables Created and Perform various queries. After completing research and receiving guidance from the module leader, I was able to learn about normalization.

During the normalization process, I added one new entity (Painting_Cust_Order) to the data to better meet my requirements and design. Every UNF attribute is stored in a single table. 1NF splits the repeating groups and creates a new entity. In 2NF, partial dependencies are separated, but in 3NF, transitive dependencies are split. I learned about SQL queries for creating users, granting access/privileges to users, connecting to users, building tables, adding data, displaying data, and a variety of other SQL questions throughout my time at work. Following normalization, I learned how it aids in the reduction of data redundancy and complexity in databases and forms. s. A Final ER Diagram was made after Normalization.

After Normalization table creation was started in oracle. A new User ‘Masterpieces’ was made and identified by aadesh and privilege was granted to the user then connecting to the system. Tables were created by using the ‘CREATE TABLE’ command and given Constraints and Data Types. Primary and Foreign Key were also allocated. Tables were Described by using the ‘describe(desc) ’ command. After the table creation, relevant data was inputted into each table. The Values were seen by using the ‘SELECT * FROM Table_Name’ command. After Completion of the populating data into the database. 7 total new table were formed.

By this way, the database of Masterpieces was formed for painting business.since normalization was applied, Tables were created, Data was Populated and Queries were executed all went smoothly. This Coursework has helped understand the topics like Normalization. A database model with actual data which is functional. With the help of real life scenario I have understood the importance of Normalization and how it increases the functionality of the database. I have understood the basics of Designing, Creating, Organizing and maintaining the database of any type of company.

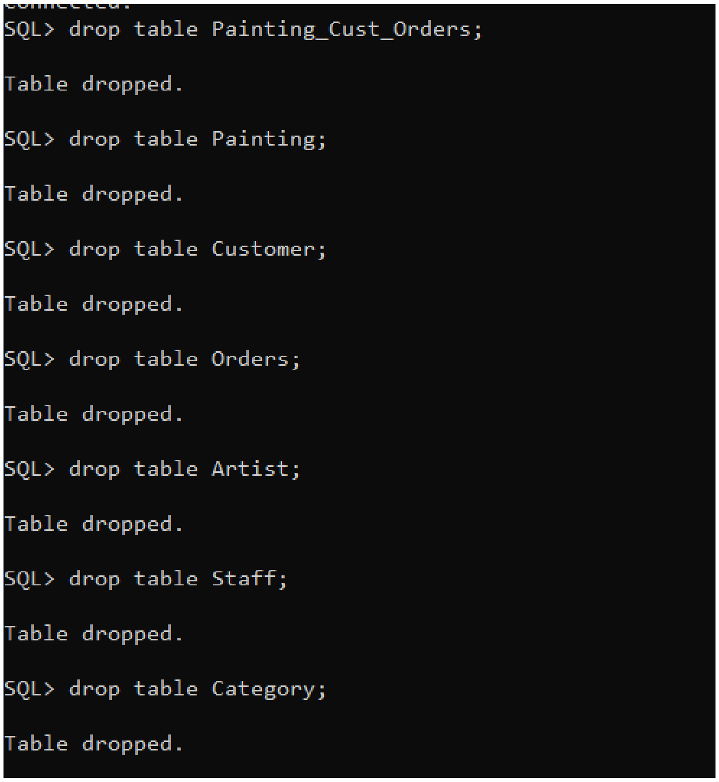
9. Database Dump file creation



```
SQL> conn
Enter user-name: system
Enter password:
Connected.
SQL> create user importcoursework identified by cw
  2
SQL> create user importcoursework identified by cw;

User created.

SQL> grant dba to importcoursework;

Grant succeeded.
```

Figure 44 dump file created

10. Drop the table



```
Connected.
SQL> drop table Painting_Cust_Orders;

Table dropped.

SQL> drop table Painting;

Table dropped.

SQL> drop table Customer;

Table dropped.

SQL> drop table Orders;

Table dropped.

SQL> drop table Artist;

Table dropped.

SQL> drop table Staff;

Table dropped.

SQL> drop table Category;

Table dropped.
```

Figure 45 table Drop

References

BeginnerBook, 2021. [Online]

Available at: https://beginnersbook.com/2015/04/e-r-model-in-dbms/

[Accessed 26 12 2021].

GeeksforGeeks, 2021. *GeeksforGeeks.* [Online]

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

[Accessed 27 12 2021].

javatpoint, 2021. *Java T point.* [Online]

Available at: https://www.javatpoint.com/dbms-second-normal-form

[Accessed 27 12 2021].

LucidChart, 2021. *Lucid.* [Online]

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

[Accessed 28 12 2021].

Peterson, R., 2021. *Guru99.* [Online]

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

[Accessed 27 12 2021].