Database (Oracle 11g) Design and Implementation

Report by: Sayush Khadka

Table of Contents

List of Figures

1. Introduction:

1.1 About the business:

This database has been designed for Mr. Steve who is a local businessman and real estate owner. He has been looking into designing and consulting companies as he has decided to extend his business in paintings. The business he is trying to expand into is an E – business or an online business where he aims and claims to lease the paintings to private individuals and commercial companies as well as the artists and the painters. He decides to name his business as Masterpieces Limited.

Aims and objectives:

i) The system can manage customers, paintings, staff, availability of the paintings.

ii) The customers have been categorized into regular (R), loyal (L), privileged (P) and VIP (V) under customer table in customer category attribute.

iii) Customer also have a choice, which allows them to buy or lease paintings through artist or the painting theme.

iv) A customer can also lease their own painting and can lease the same painting more than once.

v) The owner of the painting is paid 20% off the allocated rental price.

vi) Paintings not leased till the deadline of 4 months are returned to the owners. Then owners, however, can resubmit their painting after a month after the painting has been returned.

vii) Each painting has only an artist.

1.2 Current business activities and operations:

i) Customer information:

The system keeps the records of the customer who were involved in the business activities of this organization. The record of the customer includes a particular No (number) for each customer. Other information about the customer is also stored which includes their name, address, phone number. The customers are also divided into 4 different categories which are regular (R), loyal (L), privileged (P) and VIP (V) and are given 0%, 5%, 10% and 15% discount respectively.

ii) Order ad staff information:

The system is also designed and operated to keep the order information. In this each customer is given an order No. These orders are given by the customer and filtered by the staffs. The order table provides information about the order date, quantity, and the total price. Order No also provides information on staff, by providing the information by what staff ID the order had been placed. And similarly, the staff ID provides details of the staff.

iii) Painting and artist information:

Painting information is also provided and operated by the system. Each painting has its unique painting ID. With the help of unique painting ID, each painting ID gives more painting information which are painting name and theme. The painting ID also provides information on the artist’s name with the help of the artist ID. Rental price, lease date and return date of the painting is also provided as information.

1.3 Business rules:

i) The system must be able to handle and manage details of customers, staffs, paintings, and the availability of the paintings.

ii) Customers are to be categorized under 4 headings, which are regular (R), loyal (L), privileged (P) and VIP (V) under customer table in customer category attribute. Then 0%, 5%, 10% and 15% discount is given to the customers respectively.

iii) Customers can order paintings based on the theme of the paintings or by the name of the artist as well.

iv) The paintings can be bought or leased.

v) Customer can lease the same painting more than once.

vi) 20% is paid to the owner off the allocated rental price.

vii) Paintings not leased till the time of 4 months are returned to the owner.

viii) Each painting is allowed to have only an artist.

ix) Owner can resubmit the painting after a month when the painting is returned.

1.4 Identification of entities and attributes:

a) Customer:

i) Customer ID
ii) Customer Name
iii) Customer Address
iv) Customer Phone no
v) Customer Email
vi) Customer Category

b) Order:

i) Order No
ii) Order Date
iii) Order Quantity
iv) Staff ID
v) Staff Name
vi) Staff Salary

c) Painting:

i) Painting ID
ii) Painting Name
iii) Painting Theme
iv) Artist ID
v) Artist Name
vi) Rental Price
vii) Lease Date
viii) Return Date
ix) Availability

2. Initial ERD

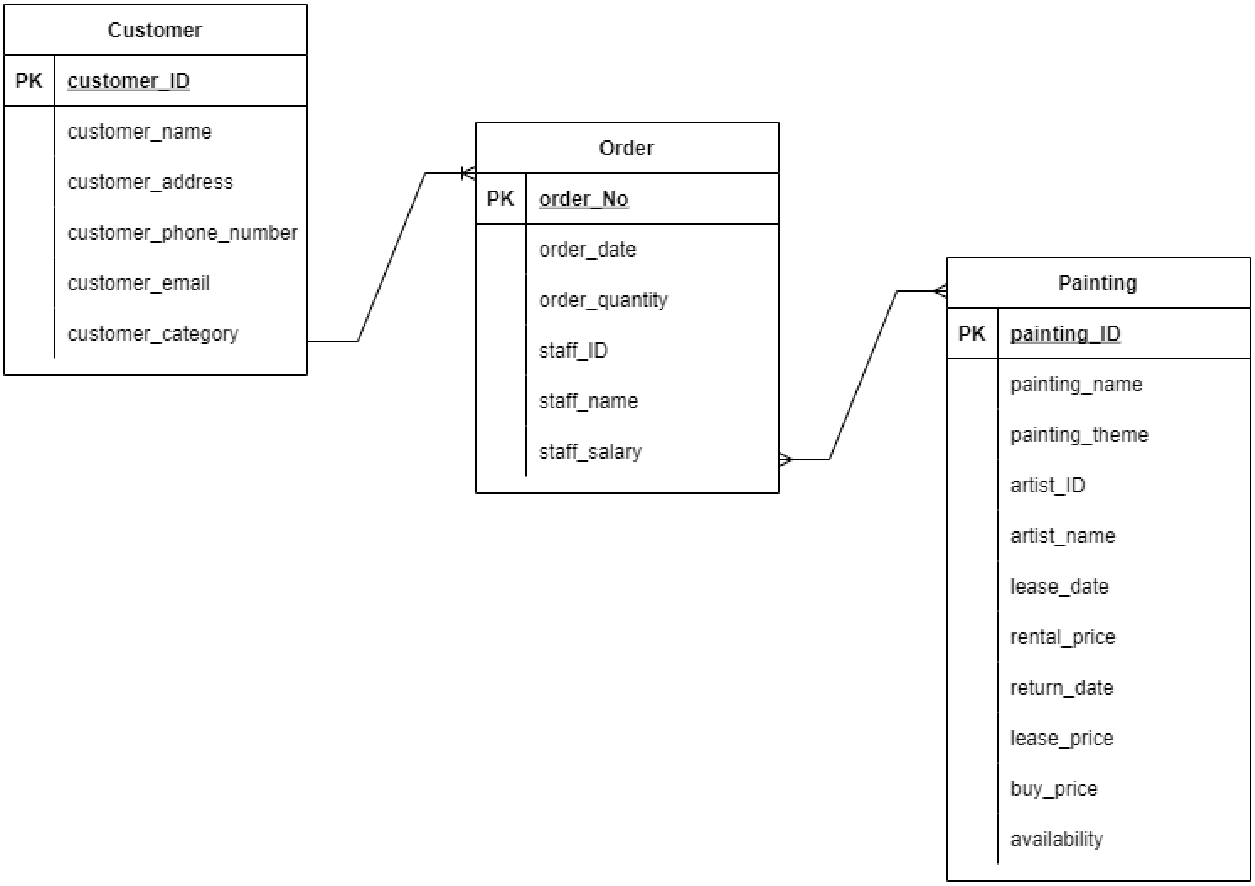


Figure 1 Initial ERD (Entity Relationship Diagram)

3. Normalization:

3.1 Scenario:

This is an Internet business called Masterpieces Limited where the company leases the paintings to private individuals and commercial companies owned by masterpieces, as well as different painters or artists.

3.2 Assumptions:

The assumptions made in this design are:

i) Customers are contacted through their phone number or email.

ii) Phone numbers and emails cannot give customer name.

iii) Discount is provided to the customer as regular (R), loyal (L), privileged (P) and VIP (V) and are given 0%, 5%, 10% and 15% discount respectively.

iv) Discount is given using the queries.

v) Staff is included in Order table in initial ERD where order_No provides staff_ID and the staff_ID provides other staff Information.

vi) Artist is included in Painting table in the initial ERD where painting_ID provides artist_ID and the artist_ID provides other artist information.

3.3 UNF

(Customer {Order, Staff {Painting}})

customer_ID, customer_name, customer_address, customer_phone_number, customer_email, customer_category, customer_discount, {order_no, order_date, order_quantity, staff_ID, staff_name, staff_salary, {painting_id, painting_name, painting_theme, artist_id, artist_name, rental_price, lease_date, return_date, lease_price, buy_price, availability}}

3.4 1NF

Customer 1 (**customer_ID**, customer_name, customer_address, customer_phone_number, customer_email, customer_category)

Order 1 (**order_no**, order_date, order_quantity, staff_ID, staff_name, staff_salary, **customer_ID**)

Painting 1 (**painting_ID**, painting_name, painting_theme, artist_id, artist_name, rental_price, lease_date, return_date, lease_price, buy_price, availability, **customer_ID**, **order_No**)

3.5 2NF

For 2NF

Customer 1 → is already in 2NF as it has no partial dependency.

order_No → order_date, order_quantity, staff_ID, staff_name, staff_salary

customer_ID →

customer_ID, order_No → forms table

painting_ID → painting_name, painting_name, painting_theme, artist_id, artist_name, rental_price, return_date, lease_price, buy_price, availability

customer_ID →

order_No →

painting_ID, order_No → lease date

customer_ID, order_No, painting_ID → forms table

Finally, in 2NF

Customer 2 (**customer_ID**, customer_name, customer_address, customer_phone_number, customer_email, customer_category)

Order 2 (**order_no**, order_date, order_quantity, staff_ID, staff_name, staff_salary)

Customer - Order (**customer_ID, order_No**)

Painting 2 (**painting_ID**, painting_name, painting_theme, artist_id, artist_name, rental_price, return_date, lease_price, buy_price, availability)

Painting – Order (lease date, **order_No, painting_ID**)

Customer – Order – Painting (**customer_ID, order_No, painting_ID**)

3.6 3NF

For 3NF

Customer 2 → is already in 3NF as it has no transitive dependency.

order_No → order_date, order_quantity, **staff_ID**

order_no gives staff_ID AND

staff_ID → staff_name, staff_ salary (There is transitive dependency)

Customer – Order → is already in 3NF as it has no transitive dependency.

painting_ID → painting_name, painting_theme, rental_price, return_date, lease_price, buy_price, availability, **artist_ID**

painting_ID gives artist_ID AND

artist_ID → artist_name

Painting – Order (lease date, order_No, painting_ID)

Customer – Order – Painting → is already in 3NF as it has no transitive dependency.

IN 3NF

Customer 3 (**customer_ID**, customer_name, customer_address, customer_phone_number, customer_email, customer_category)

Order 3 (**order_No**, order_date, order_quantity, **staff_ID**)

Staff 3 (**staff_ID**, staff_name, staff_salary)

Painting 3 (**painting_ID**, painting_name, painting_theme, rental_price, return_date, lease_price, buy_price, availability, **artist_ID)**

Painting – Order (lease_date, **order_No, painting_ID**)

Artist 3 (**artist_id**, artist_name)

Customer – Order – Painting (**customer_ID, order_No, painting_ID**)

4. Final ERD:

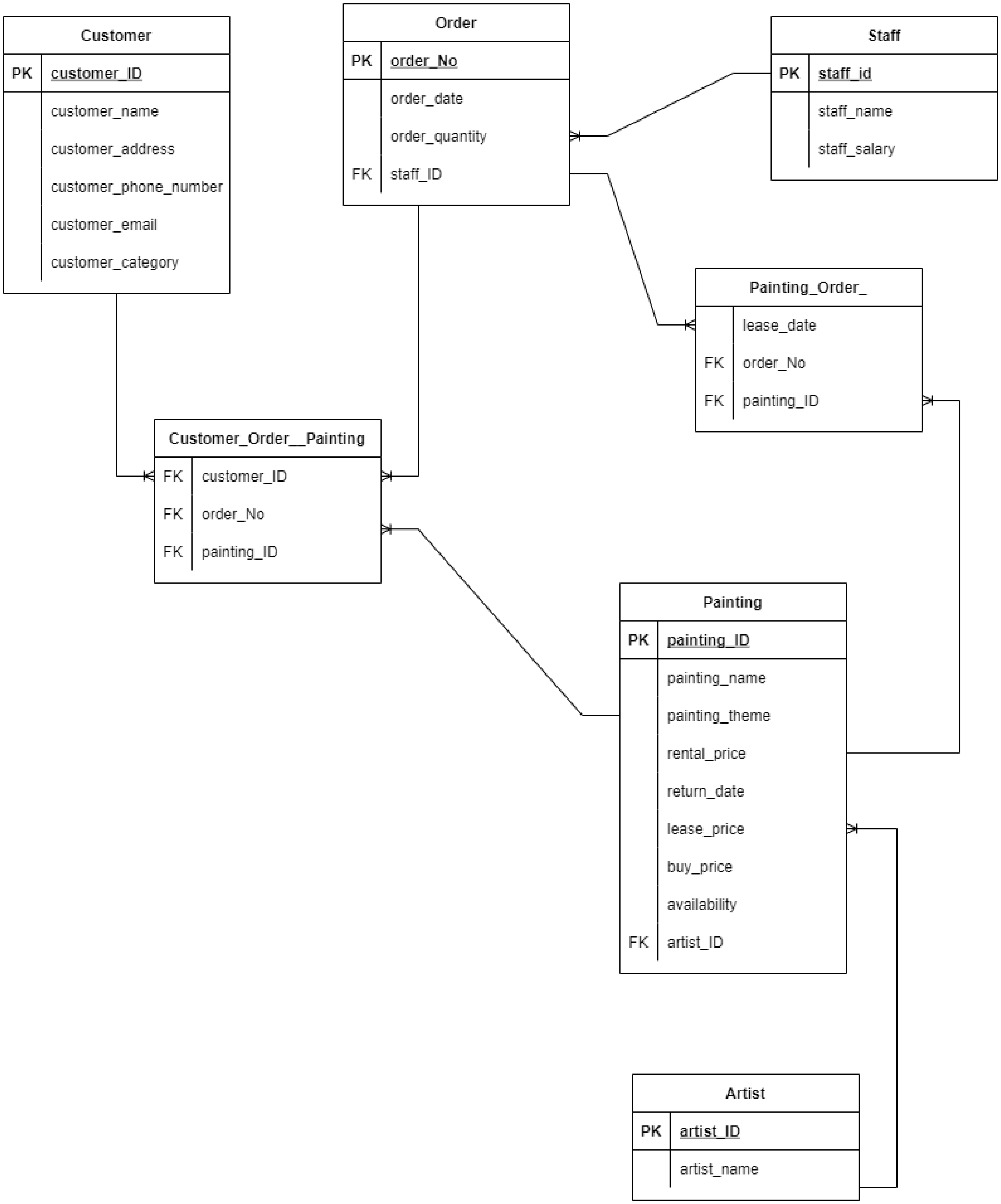
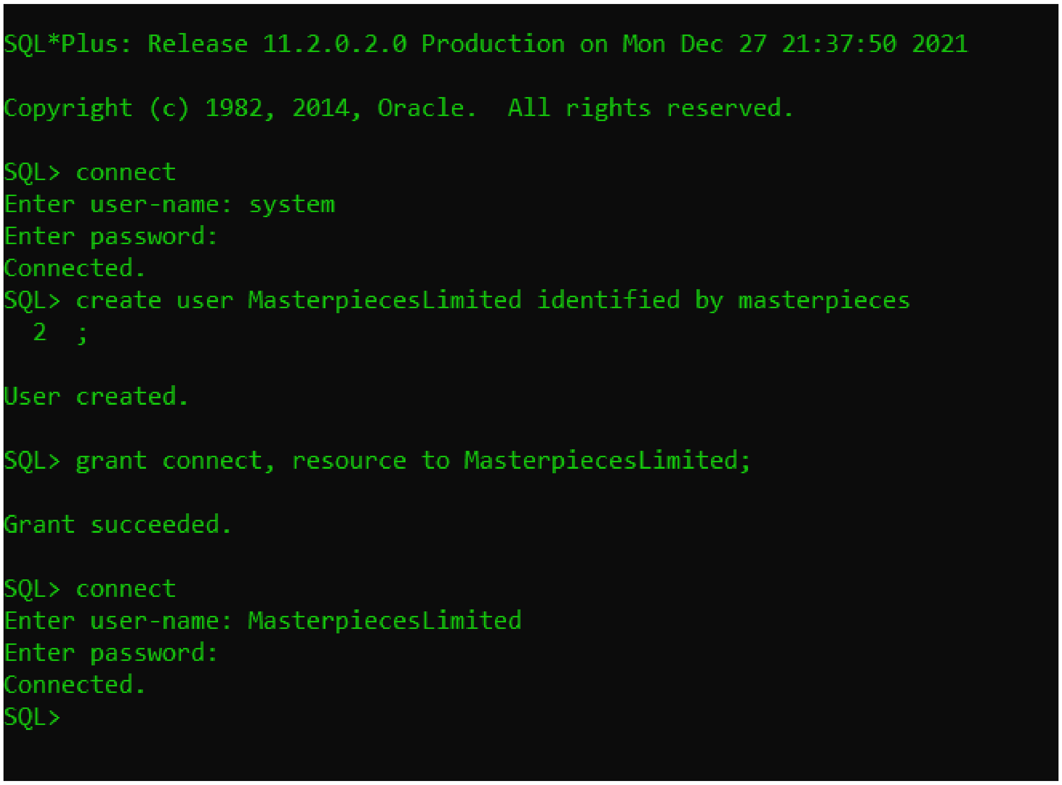


Figure 2 Final ERD (Entity Relationship Diagram)

5. Implementation:

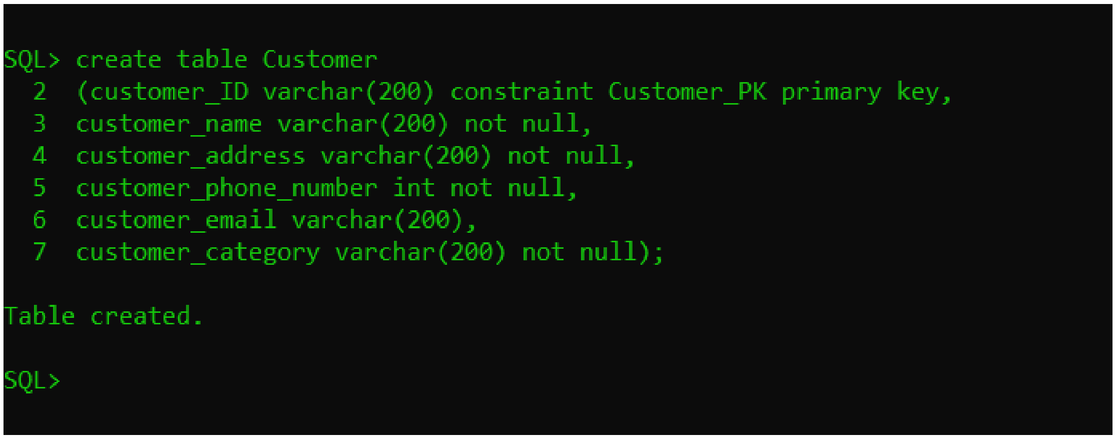
5.1 Creation and description of tables:

Creation of tables:



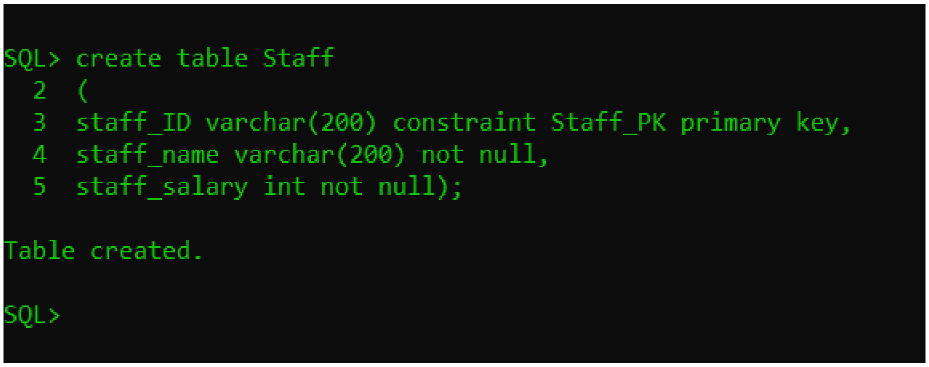
```
SQL*Plus: Release 11.2.0.2.0 Production on Mon Dec 27 21:37:50 2021

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

SQL> connect
Enter user-name: system
Enter password:
Connected.
SQL> create user MasterpiecesLimited identified by masterpieces
  2  ;

User created.

SQL> grant connect, resource to MasterpiecesLimited;

Grant succeeded.

SQL> connect
Enter user-name: MasterpiecesLimited
Enter password:
Connected.
SQL>
```

Figure 3 Create user



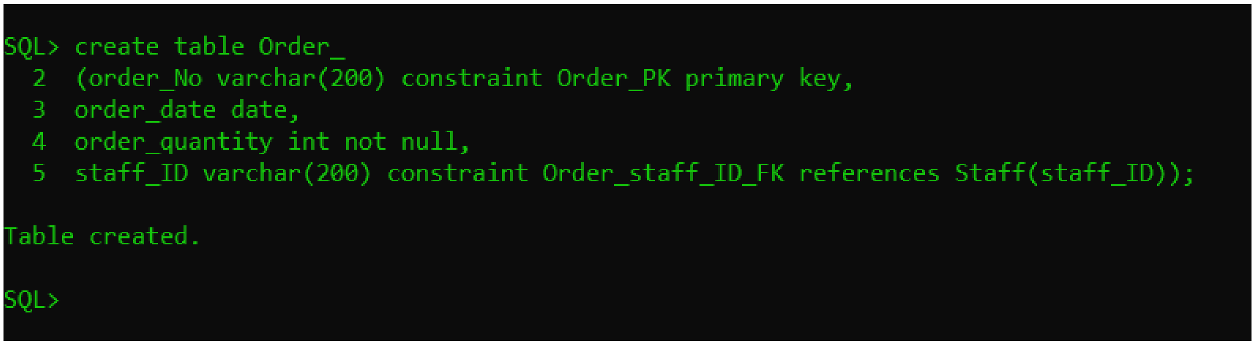
```
SQL> create table Customer
  2  (customer_ID varchar(200) constraint Customer_PK primary key,
  3  customer_name varchar(200) not null,
  4  customer_address varchar(200) not null,
  5  customer_phone_number int not null,
  6  customer_email varchar(200),
  7  customer_category varchar(200) not null);

Table created.

SQL>
```

Figure 4 Customer table



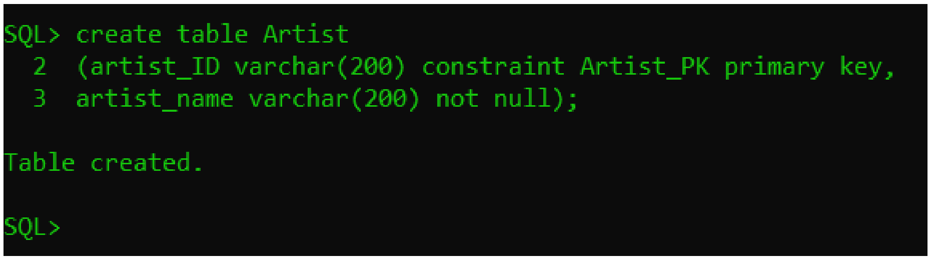
```
SQL> create table Staff
     (
     staff_ID varchar(200) constraint Staff_PK primary key,
     staff_name varchar(200) not null,
     staff_salary int not null);

Table created.

SQL>
```

Figure 5 Staff table



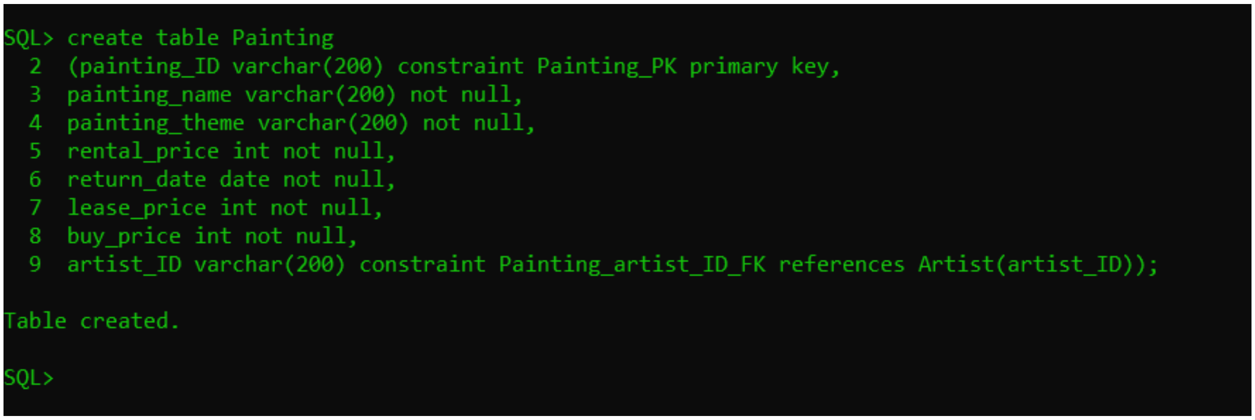
```
SQL> create table Order_
     (order_No varchar(200) constraint Order_PK primary key,
     order_date date,
     order_quantity int not null,
     staff_ID varchar(200) constraint Order_staff_ID_FK references Staff(staff_ID));

Table created.

SQL>
```

Figure 6 Order_ table



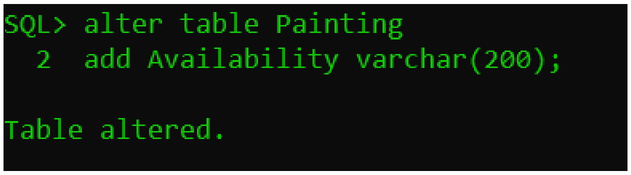
```
SQL> create table Artist
     (artist_ID varchar(200) constraint Artist_PK primary key,
     artist_name varchar(200) not null);

Table created.

SQL>
```

Figure 7 Artist table



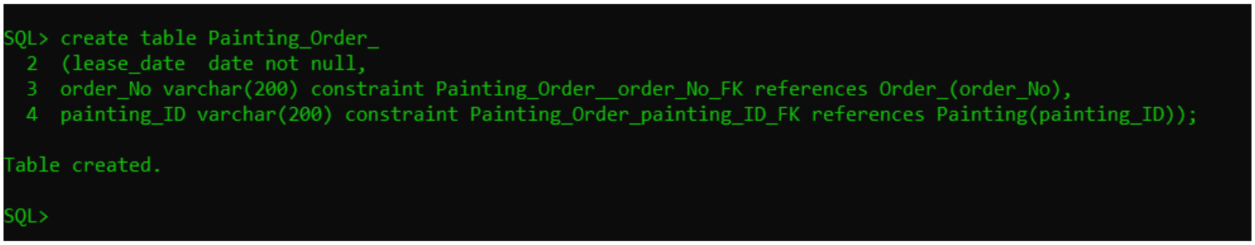
```
SQL> create table Painting
(painting_ID varchar(200) constraint Painting_PK primary key,
painting_name varchar(200) not null,
painting_theme varchar(200) not null,
rental_price int not null,
return_date date not null,
lease_price int not null,
buy_price int not null,
artist_ID varchar(200) constraint Painting_artist_ID_FK references Artist(artist_ID));

Table created.

SQL>
```

Figure 8 Painting table



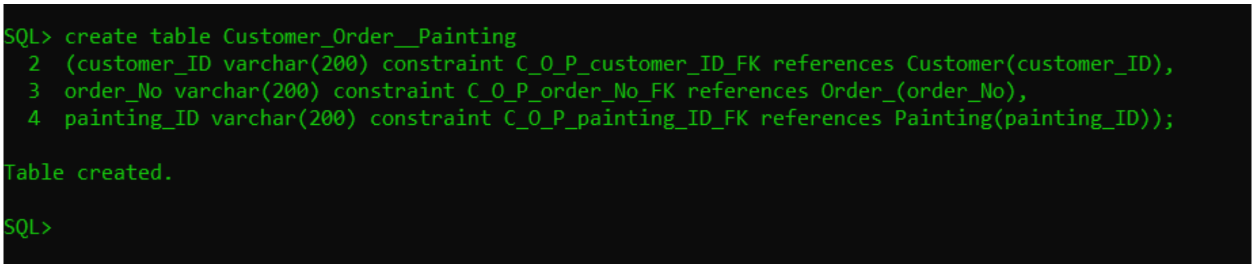
```
SQL> alter table Painting
add Availability varchar(200);

Table altered.
```

Figure 9 Adding availability column



```
SQL> create table Painting_Order_
(lease_date  date not null,
order_No varchar(200) constraint Painting_Order__order_No_FK references Order_(order_No),
painting_ID varchar(200) constraint Painting_Order_painting_ID_FK references Painting(painting_ID));

Table created.

SQL>
```

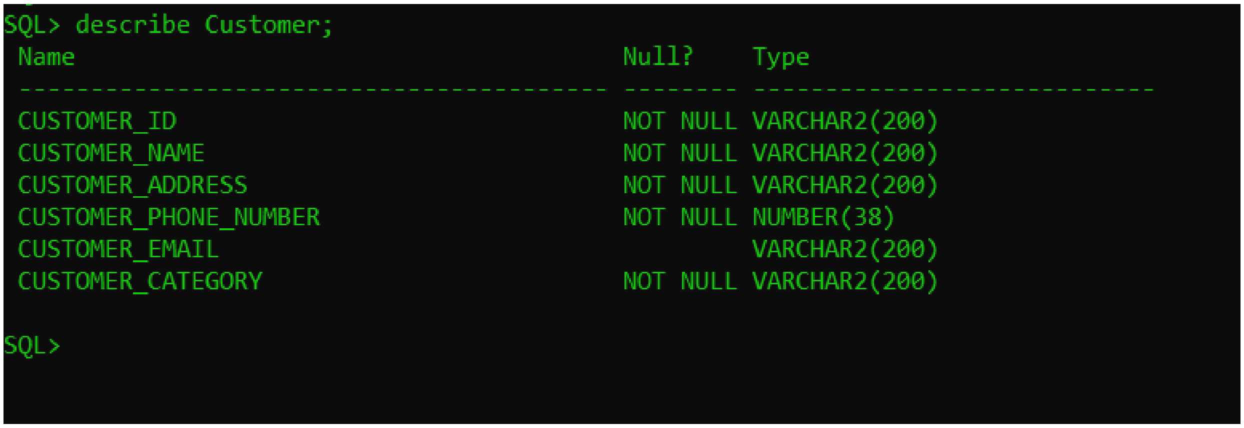
Figure 10 Painting_order_ table



```
SQL> create table Customer_Order__Painting
(customer_ID varchar(200) constraint C_O_P_customer_ID_FK references Customer(customer_ID),
order_No varchar(200) constraint C_O_P_order_No_FK references Order_(order_No),
painting_ID varchar(200) constraint C_O_P_painting_ID_FK references Painting(painting_ID));

Table created.

SQL>
```

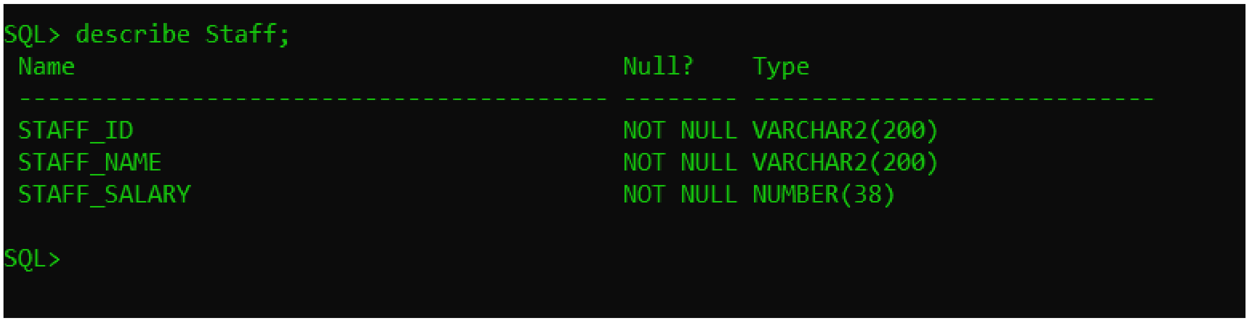
Figure 11 Customer_Order__Painting table

Description of tables:



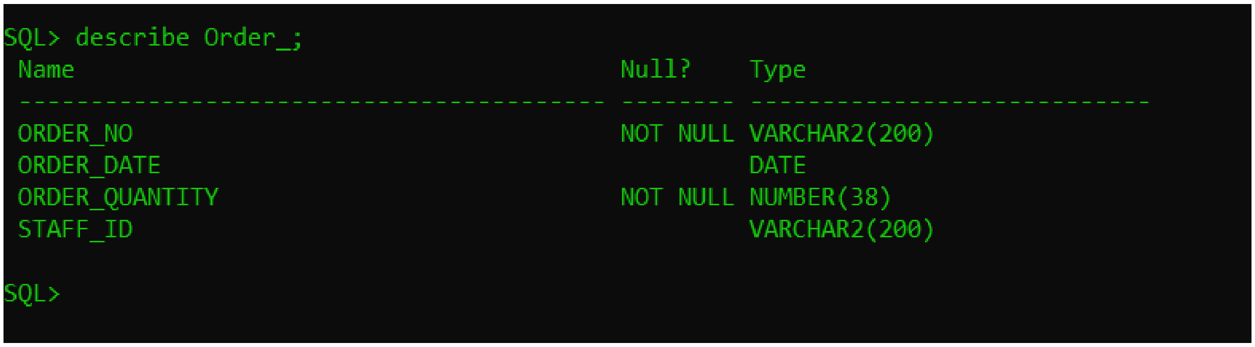
```
SQL> describe Customer;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                               NOT NULL VARCHAR2(200)
 CUSTOMER_NAME                             NOT NULL VARCHAR2(200)
 CUSTOMER_ADDRESS                          NOT NULL VARCHAR2(200)
 CUSTOMER_PHONE_NUMBER                     NOT NULL NUMBER(38)
 CUSTOMER_EMAIL                                     VARCHAR2(200)
 CUSTOMER_CATEGORY                         NOT NULL VARCHAR2(200)

SQL>
```

Figure 12 Customer table description



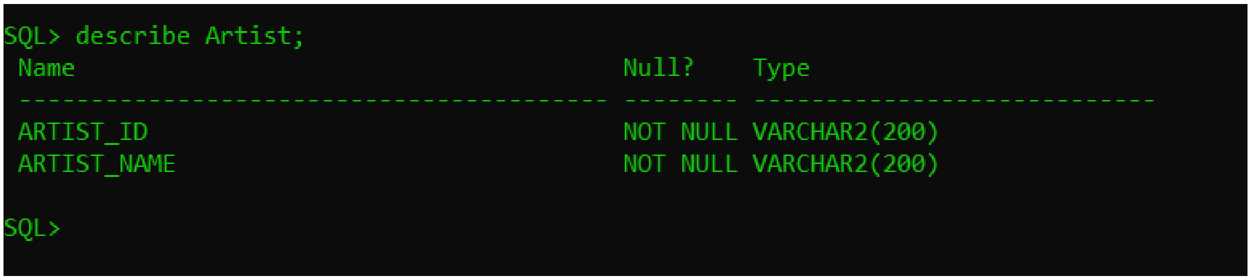
```
SQL> describe Staff;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STAFF_ID                                  NOT NULL VARCHAR2(200)
 STAFF_NAME                                NOT NULL VARCHAR2(200)
 STAFF_SALARY                              NOT NULL NUMBER(38)

SQL>
```

Figure 13 Staff table description



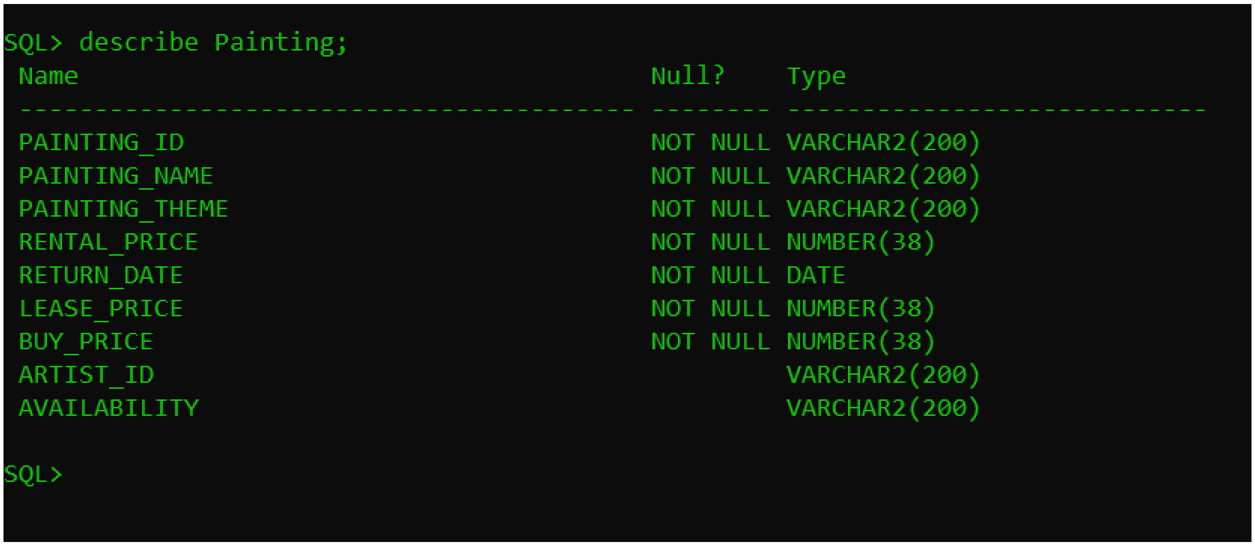
```
SQL> describe Order_;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ORDER_NO                                  NOT NULL VARCHAR2(200)
 ORDER_DATE                                         DATE
 ORDER_QUANTITY                            NOT NULL NUMBER(38)
 STAFF_ID                                           VARCHAR2(200)

SQL>
```

Figure 14 Order_ table description



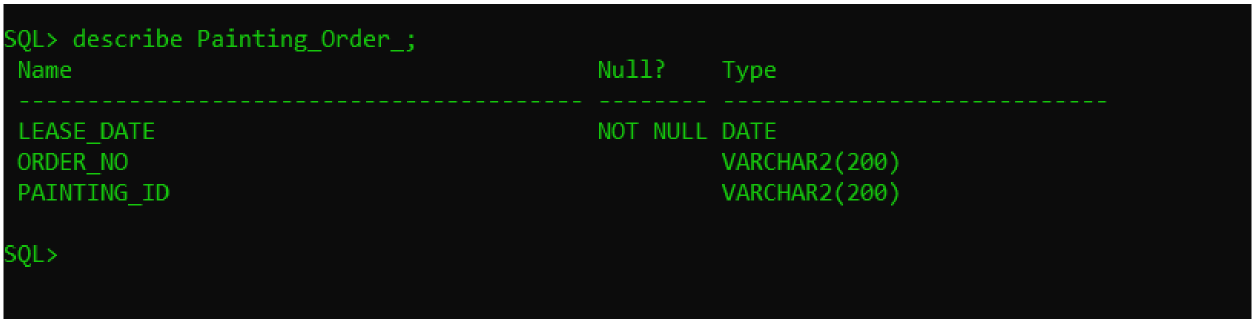
```
SQL> describe Artist;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ARTIST_ID                                 NOT NULL VARCHAR2(200)
 ARTIST_NAME                               NOT NULL VARCHAR2(200)

SQL>
```

Figure 15 Artist table description



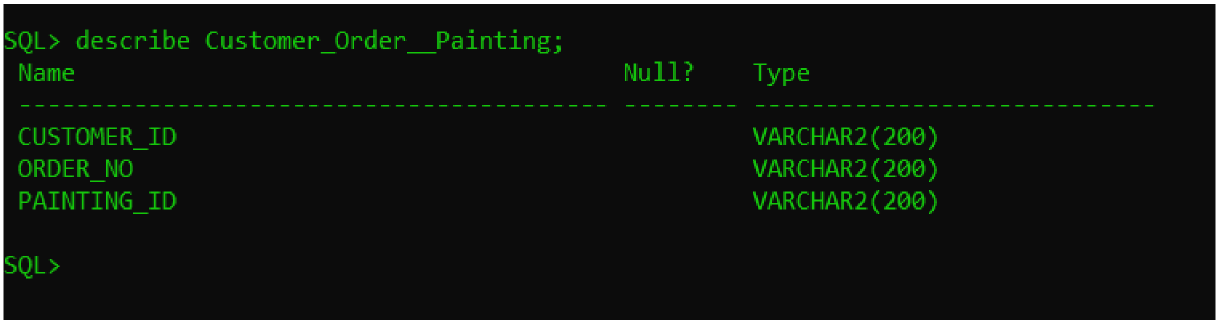
```
SQL> describe Painting;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PAINTING_ID                               NOT NULL VARCHAR2(200)
 PAINTING_NAME                             NOT NULL VARCHAR2(200)
 PAINTING_THEME                            NOT NULL VARCHAR2(200)
 RENTAL_PRICE                              NOT NULL NUMBER(38)
 RETURN_DATE                               NOT NULL DATE
 LEASE_PRICE                               NOT NULL NUMBER(38)
 BUY_PRICE                                 NOT NULL NUMBER(38)
 ARTIST_ID                                          VARCHAR2(200)
 AVAILABILITY                                       VARCHAR2(200)

SQL>
```

Figure 16 Painting table description



```
SQL> describe Painting_Order_;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 LEASE_DATE                                NOT NULL DATE
 ORDER_NO                                           VARCHAR2(200)
 PAINTING_ID                                        VARCHAR2(200)

SQL>
```

Figure 17 Painting_Order_ table description

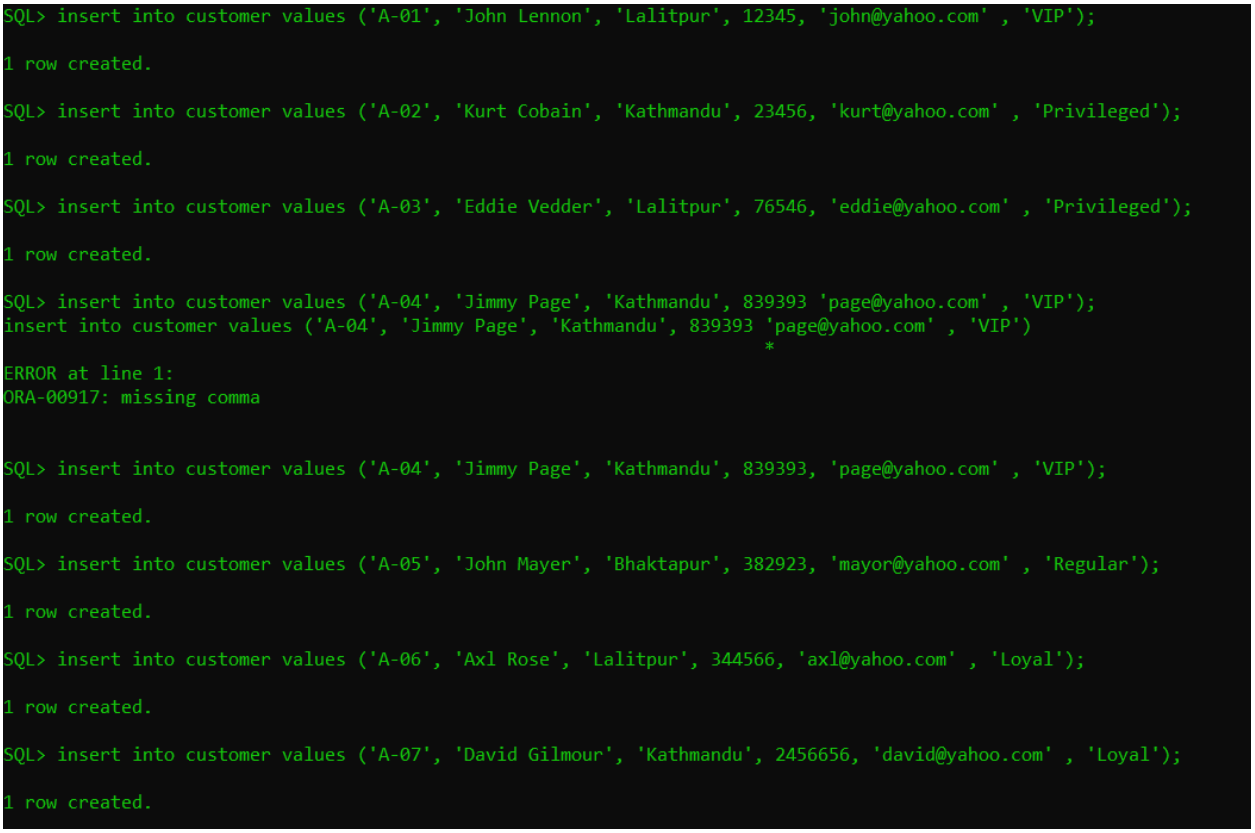


```
SQL> describe Customer_Order__Painting;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                                        VARCHAR2(200)
 ORDER_NO                                           VARCHAR2(200)
 PAINTING_ID                                        VARCHAR2(200)

SQL>
```

Figure 18 Customer_Order__Painting table description

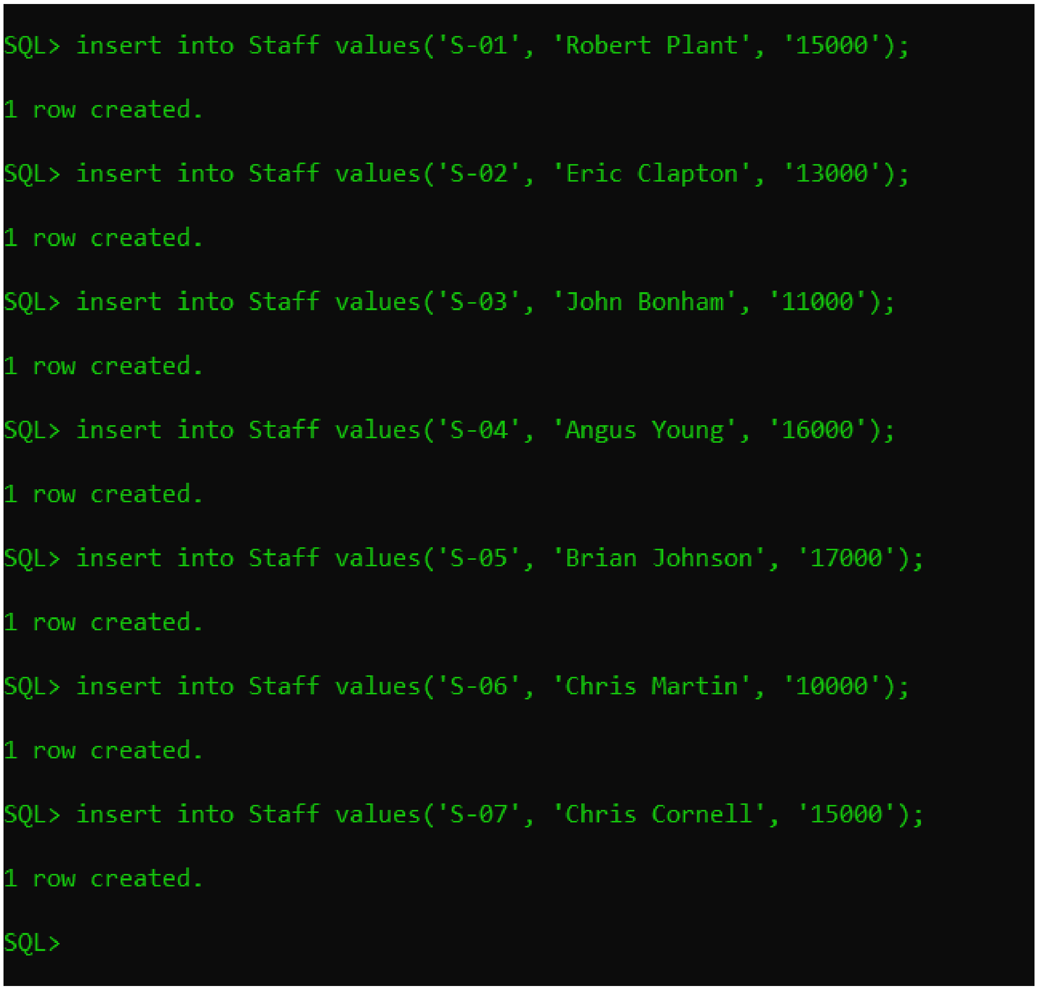
5.2 Inserting data and showing content in the table:

Inserting data:



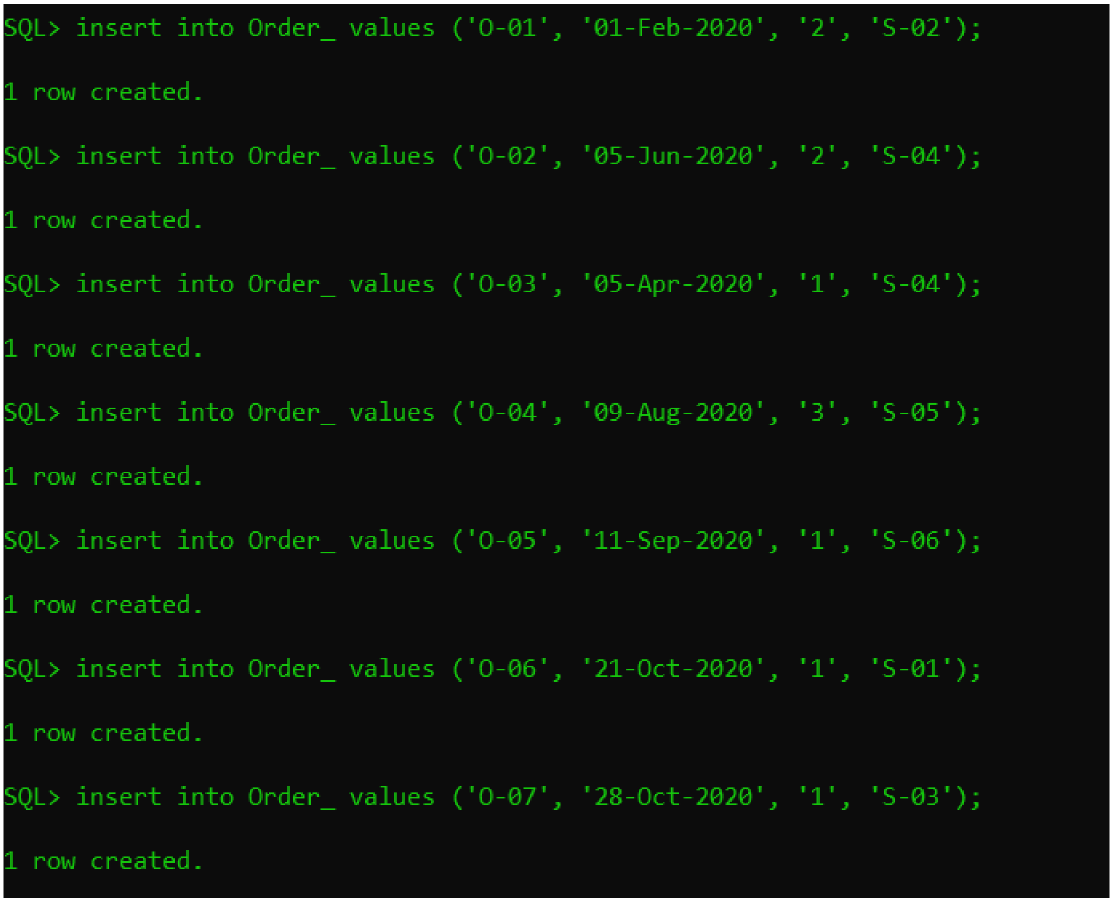
```
SQL> insert into customer values ('A-01', 'John Lennon', 'Lalitpur', 12345, 'john@yahoo.com' , 'VIP');

1 row created.

SQL> insert into customer values ('A-02', 'Kurt Cobain', 'Kathmandu', 23456, 'kurt@yahoo.com' , 'Privileged');

1 row created.

SQL> insert into customer values ('A-03', 'Eddie Vedder', 'Lalitpur', 76546, 'eddie@yahoo.com' , 'Privileged');

1 row created.

SQL> insert into customer values ('A-04', 'Jimmy Page', 'Kathmandu', 839393 'page@yahoo.com' , 'VIP');
insert into customer values ('A-04', 'Jimmy Page', 'Kathmandu', 839393 'page@yahoo.com' , 'VIP')
                                                                       *
ERROR at line 1:
ORA-00917: missing comma


SQL> insert into customer values ('A-04', 'Jimmy Page', 'Kathmandu', 839393, 'page@yahoo.com' , 'VIP');

1 row created.

SQL> insert into customer values ('A-05', 'John Mayer', 'Bhaktapur', 382923, 'mayor@yahoo.com' , 'Regular');

1 row created.

SQL> insert into customer values ('A-06', 'Axl Rose', 'Lalitpur', 344566, 'axl@yahoo.com' , 'Loyal');

1 row created.

SQL> insert into customer values ('A-07', 'David Gilmour', 'Kathmandu', 2456656, 'david@yahoo.com' , 'Loyal');

1 row created.
```

Figure 19 Inserted data in customer table



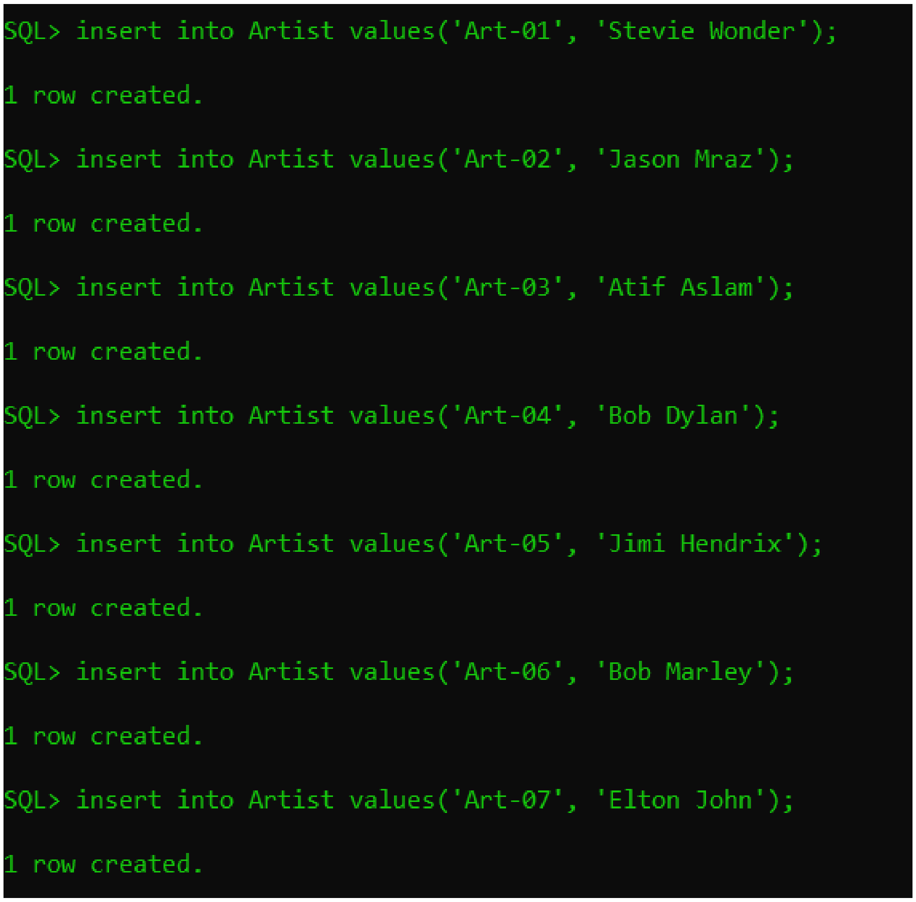
```
SQL> insert into Staff values('S-01', 'Robert Plant', '15000');

1 row created.

SQL> insert into Staff values('S-02', 'Eric Clapton', '13000');

1 row created.

SQL> insert into Staff values('S-03', 'John Bonham', '11000');

1 row created.

SQL> insert into Staff values('S-04', 'Angus Young', '16000');

1 row created.

SQL> insert into Staff values('S-05', 'Brian Johnson', '17000');

1 row created.

SQL> insert into Staff values('S-06', 'Chris Martin', '10000');

1 row created.

SQL> insert into Staff values('S-07', 'Chris Cornell', '15000');

1 row created.

SQL>
```

Figure 20 Inserted data in Staff table



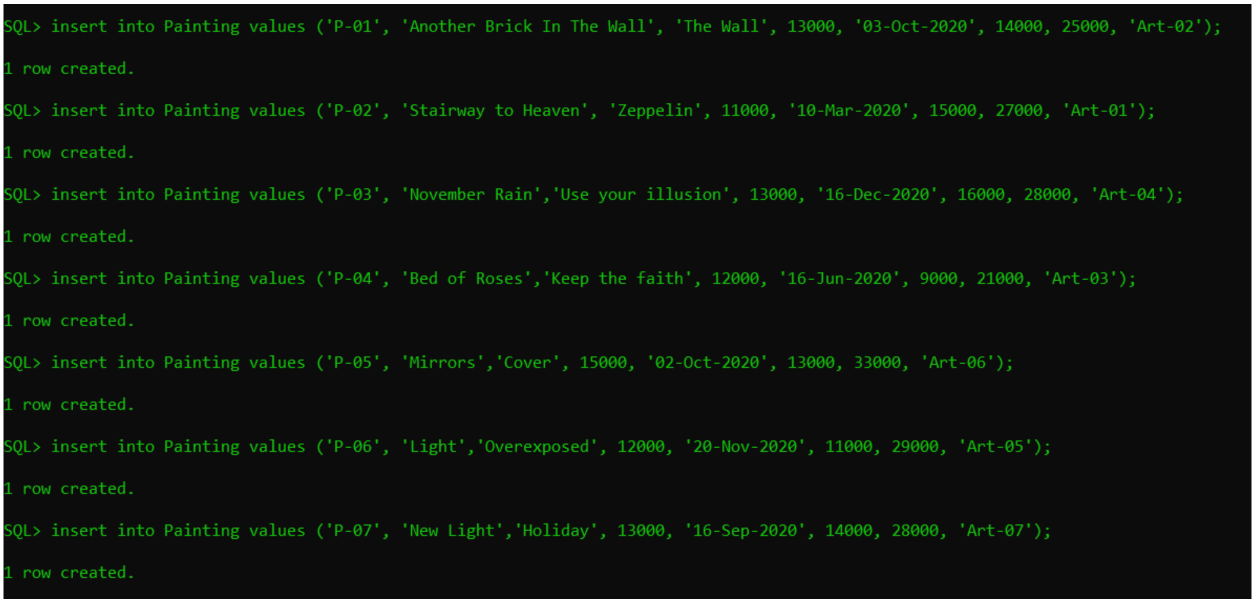
```
SQL> insert into Order_ values ('O-01', '01-Feb-2020', '2', 'S-02');

1 row created.

SQL> insert into Order_ values ('O-02', '05-Jun-2020', '2', 'S-04');

1 row created.

SQL> insert into Order_ values ('O-03', '05-Apr-2020', '1', 'S-04');

1 row created.

SQL> insert into Order_ values ('O-04', '09-Aug-2020', '3', 'S-05');

1 row created.

SQL> insert into Order_ values ('O-05', '11-Sep-2020', '1', 'S-06');

1 row created.

SQL> insert into Order_ values ('O-06', '21-Oct-2020', '1', 'S-01');

1 row created.

SQL> insert into Order_ values ('O-07', '28-Oct-2020', '1', 'S-03');

1 row created.
```

Figure 21 inserted data in Order_ table



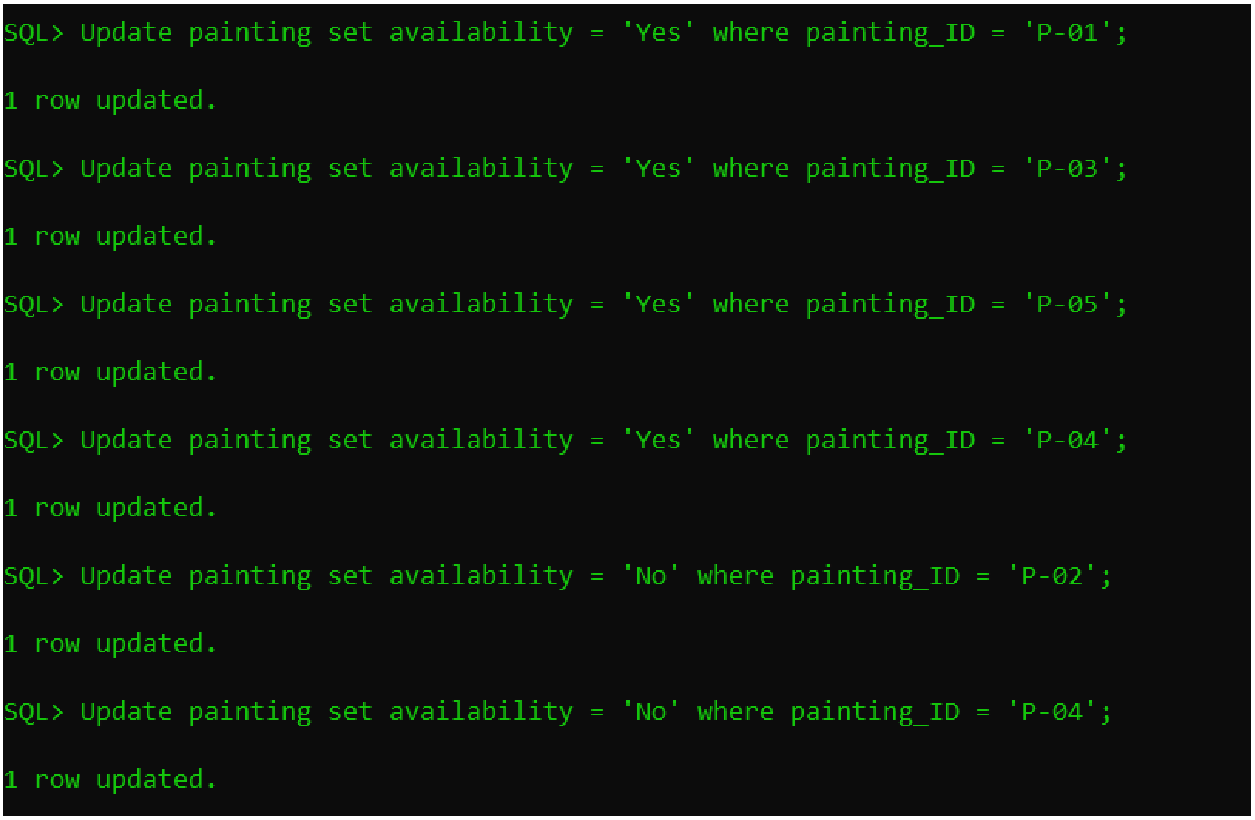
```
SQL> insert into Artist values('Art-01', 'Stevie Wonder');

1 row created.

SQL> insert into Artist values('Art-02', 'Jason Mraz');

1 row created.

SQL> insert into Artist values('Art-03', 'Atif Aslam');

1 row created.

SQL> insert into Artist values('Art-04', 'Bob Dylan');

1 row created.

SQL> insert into Artist values('Art-05', 'Jimi Hendrix');

1 row created.

SQL> insert into Artist values('Art-06', 'Bob Marley');

1 row created.

SQL> insert into Artist values('Art-07', 'Elton John');

1 row created.
```

Figure 22 Inserted data in Artist table



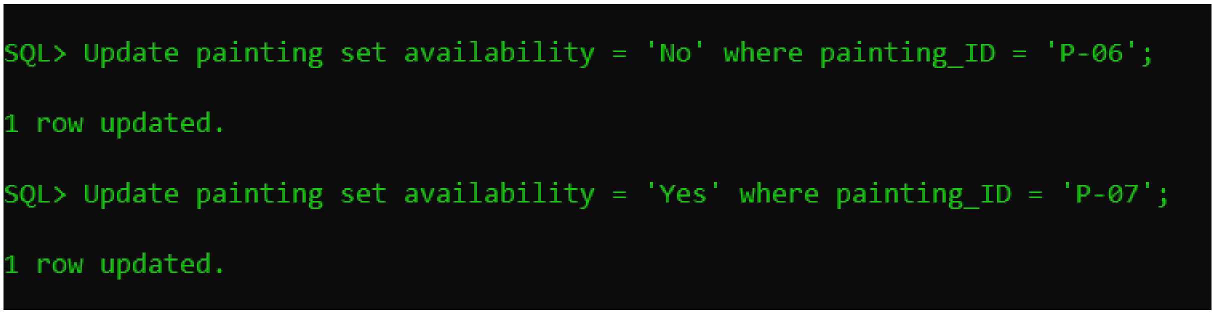
```
SQL> insert into Painting values ('P-01', 'Another Brick In The Wall', 'The Wall', 13000, '03-Oct-2020', 14000, 25000, 'Art-02');

1 row created.

SQL> insert into Painting values ('P-02', 'Stairway to Heaven', 'Zeppelin', 11000, '10-Mar-2020', 15000, 27000, 'Art-01');

1 row created.

SQL> insert into Painting values ('P-03', 'November Rain','Use your illusion', 13000, '16-Dec-2020', 16000, 28000, 'Art-04');

1 row created.

SQL> insert into Painting values ('P-04', 'Bed of Roses','Keep the faith', 12000, '16-Jun-2020', 9000, 21000, 'Art-03');

1 row created.

SQL> insert into Painting values ('P-05', 'Mirrors','Cover', 15000, '02-Oct-2020', 13000, 33000, 'Art-06');

1 row created.

SQL> insert into Painting values ('P-06', 'Light','Overexposed', 12000, '20-Nov-2020', 11000, 29000, 'Art-05');

1 row created.

SQL> insert into Painting values ('P-07', 'New Light','Holiday', 13000, '16-Sep-2020', 14000, 28000, 'Art-07');

1 row created.
```

Figure 23 Inserted data in Painting table



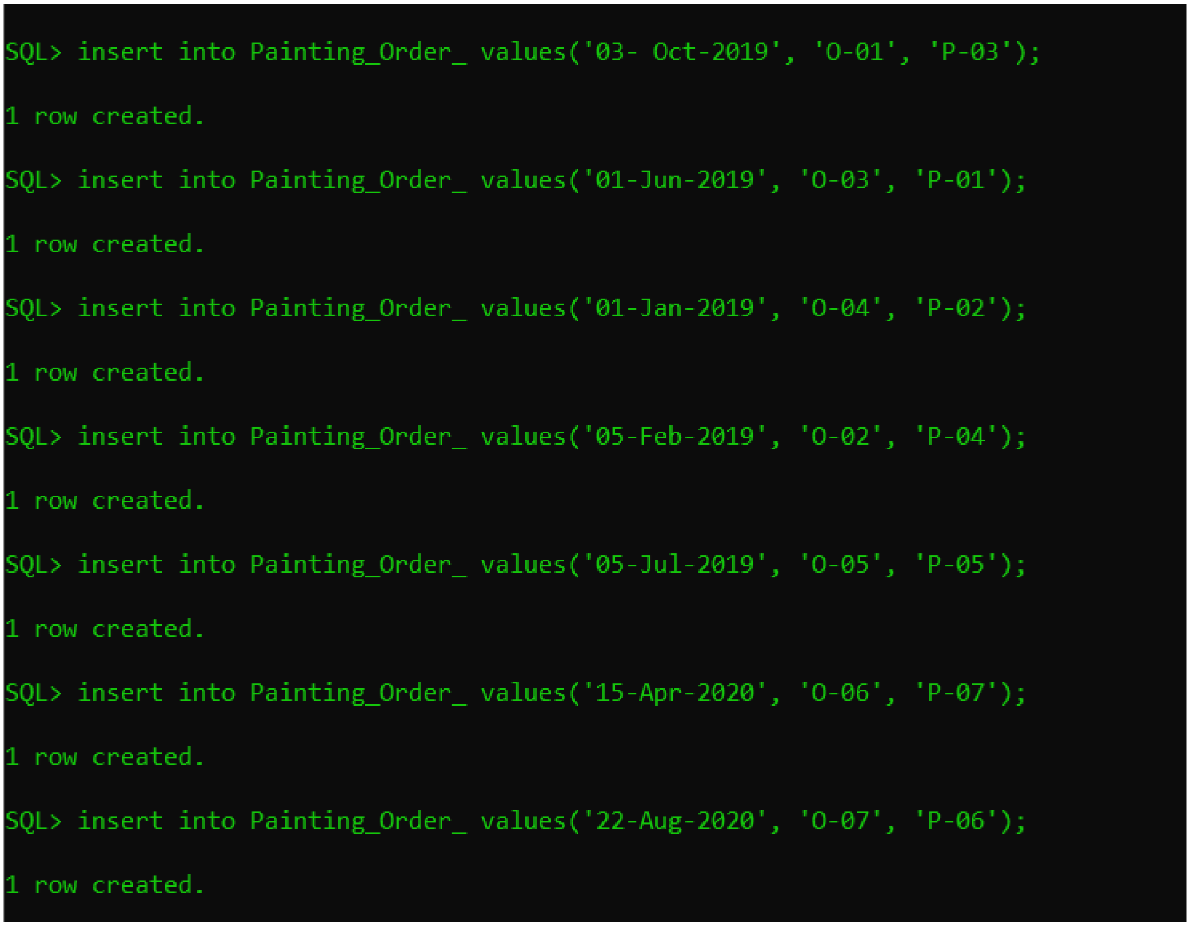
```
SQL> Update painting set availability = 'Yes' where painting_ID = 'P-01';

1 row updated.

SQL> Update painting set availability = 'Yes' where painting_ID = 'P-03';

1 row updated.

SQL> Update painting set availability = 'Yes' where painting_ID = 'P-05';

1 row updated.

SQL> Update painting set availability = 'Yes' where painting_ID = 'P-04';

1 row updated.

SQL> Update painting set availability = 'No' where painting_ID = 'P-02';

1 row updated.

SQL> Update painting set availability = 'No' where painting_ID = 'P-04';

1 row updated.
```

Figure 24 Update table (Adding values to availability)



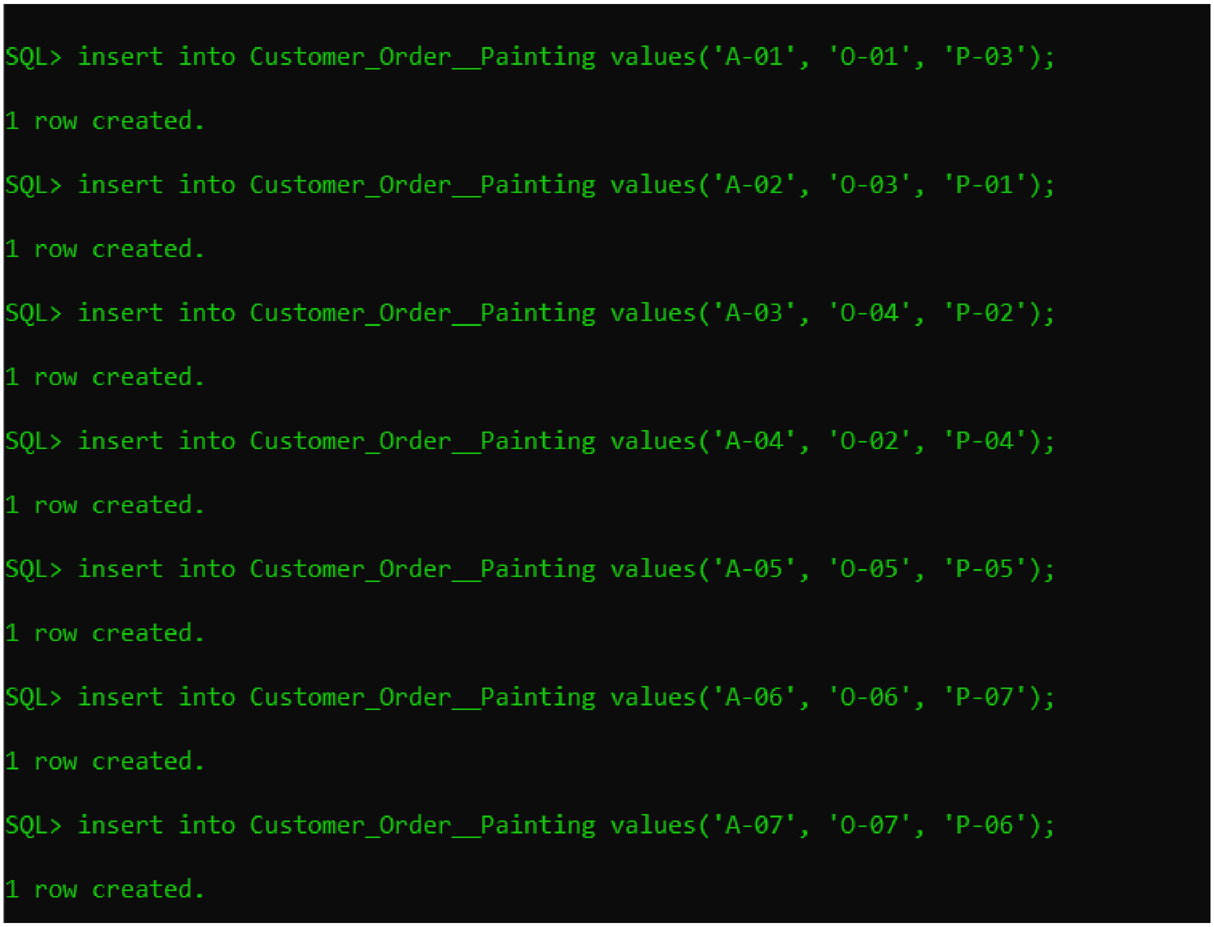
```
SQL> Update painting set availability = 'No' where painting_ID = 'P-06';

1 row updated.

SQL> Update painting set availability = 'Yes' where painting_ID = 'P-07';

1 row updated.
```

Figure 25 Update table(adding value to availability continued)



```
SQL> insert into Painting_Order_ values('03- Oct-2019', 'O-01', 'P-03');

1 row created.

SQL> insert into Painting_Order_ values('01-Jun-2019', 'O-03', 'P-01');

1 row created.

SQL> insert into Painting_Order_ values('01-Jan-2019', 'O-04', 'P-02');

1 row created.

SQL> insert into Painting_Order_ values('05-Feb-2019', 'O-02', 'P-04');

1 row created.

SQL> insert into Painting_Order_ values('05-Jul-2019', 'O-05', 'P-05');

1 row created.

SQL> insert into Painting_Order_ values('15-Apr-2020', 'O-06', 'P-07');

1 row created.

SQL> insert into Painting_Order_ values('22-Aug-2020', 'O-07', 'P-06');

1 row created.
```

*Figure 26 Inserted data in Painting_Order_*



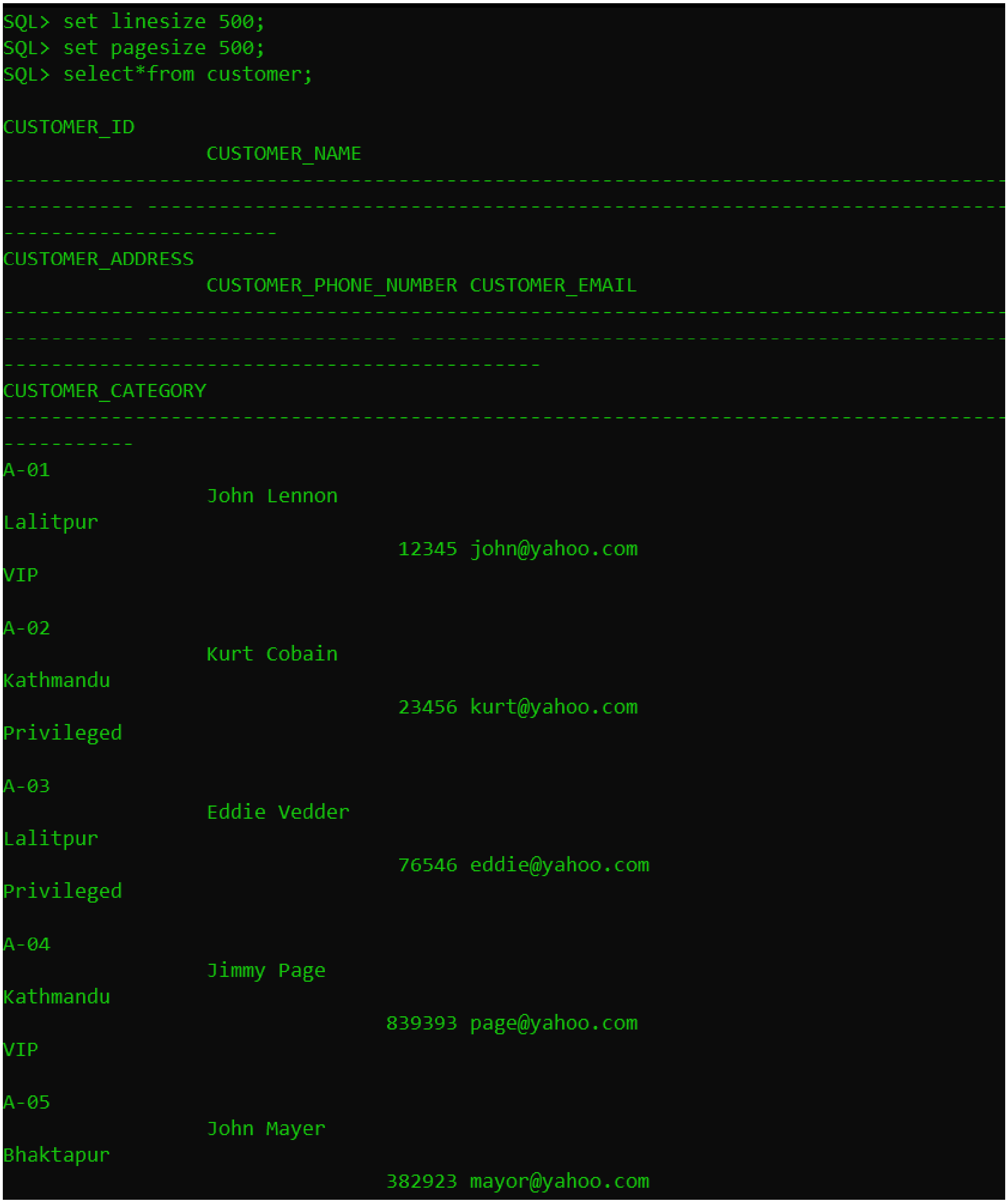
```
SQL> insert into Customer_Order__Painting values('A-01', 'O-01', 'P-03');

1 row created.

SQL> insert into Customer_Order__Painting values('A-02', 'O-03', 'P-01');

1 row created.

SQL> insert into Customer_Order__Painting values('A-03', 'O-04', 'P-02');

1 row created.

SQL> insert into Customer_Order__Painting values('A-04', 'O-02', 'P-04');

1 row created.

SQL> insert into Customer_Order__Painting values('A-05', 'O-05', 'P-05');

1 row created.

SQL> insert into Customer_Order__Painting values('A-06', 'O-06', 'P-07');

1 row created.

SQL> insert into Customer_Order__Painting values('A-07', 'O-07', 'P-06');

1 row created.
```

Figure 27 Customer_Order__Painting

Displaying data in the table:

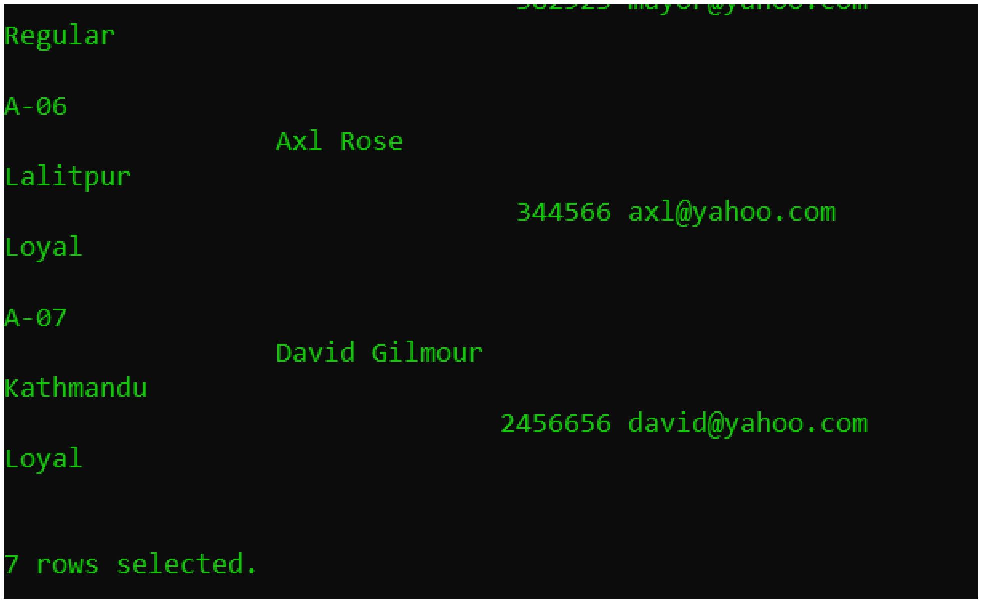
The default linesize and pagesize did not display the table data properly while using SELECT statement.

Linesize set to 500 and Pagesize set to 500 also did not work. Other numerical values were also given to the linesize and the pagesize but it did not work.



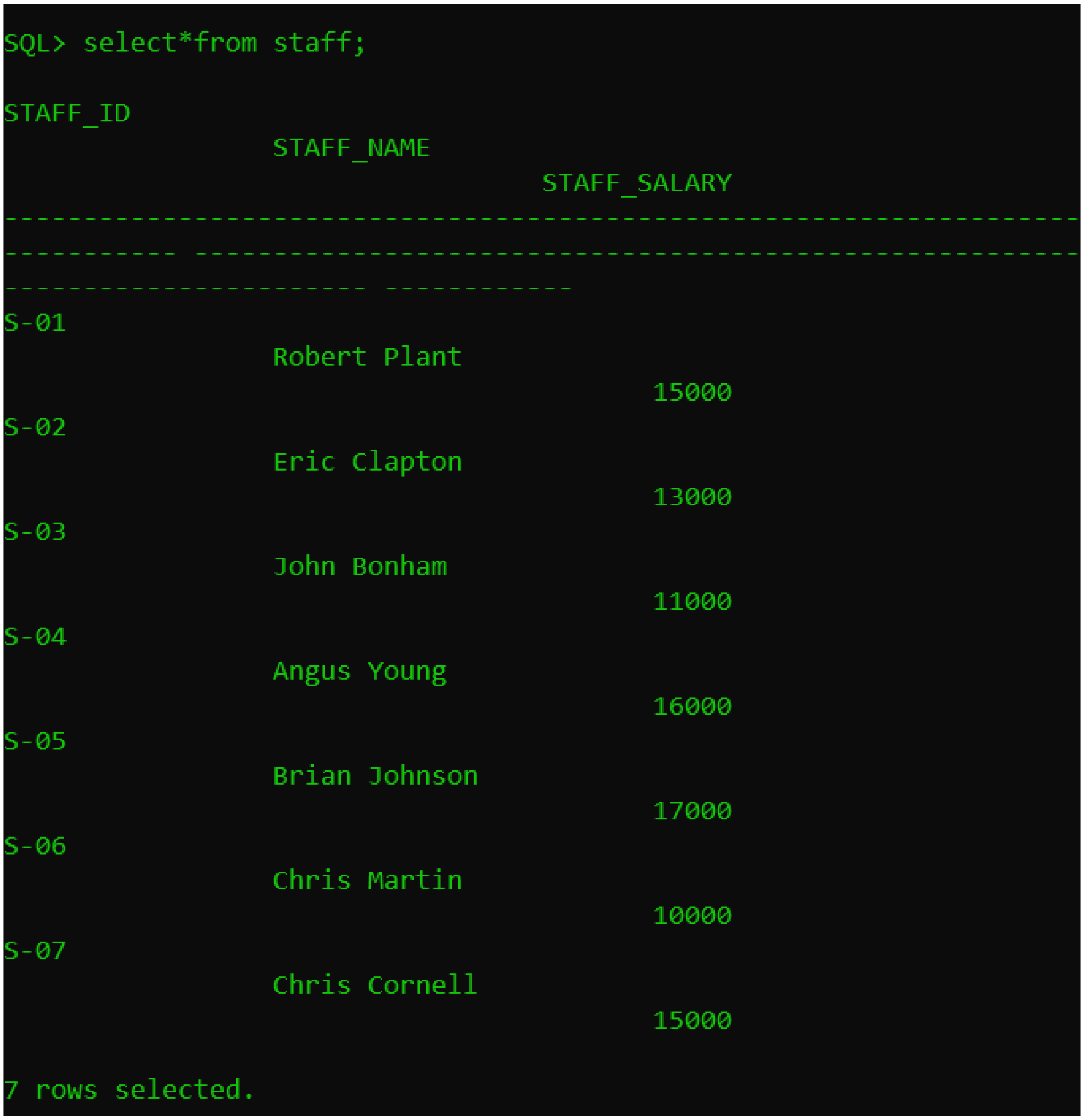
```
SQL> set linesize 500;
SQL> set pagesize 500;
SQL> select*from customer;

CUSTOMER_ID
                 CUSTOMER_NAME
------------------------------------------------------------------------------------
----------- ------------------------------------------------------------------------
-----------------------
CUSTOMER_ADDRESS
                 CUSTOMER_PHONE_NUMBER CUSTOMER_EMAIL
------------------------------------------------------------------------------------
----------- -------------------- --------------------------------------------------
---------------------------------------------
CUSTOMER_CATEGORY
------------------------------------------------------------------------------------
-----------
A-01
                 John Lennon
Lalitpur
                                 12345 john@yahoo.com
VIP

A-02
                 Kurt Cobain
Kathmandu
                                 23456 kurt@yahoo.com
Privileged

A-03
                 Eddie Vedder
Lalitpur
                                 76546 eddie@yahoo.com
Privileged

A-04
                 Jimmy Page
Kathmandu
                                839393 page@yahoo.com
VIP

A-05
                 John Mayer
Bhaktapur
                                382923 mayor@yahoo.com
```

Figure 28 Data in Customer table



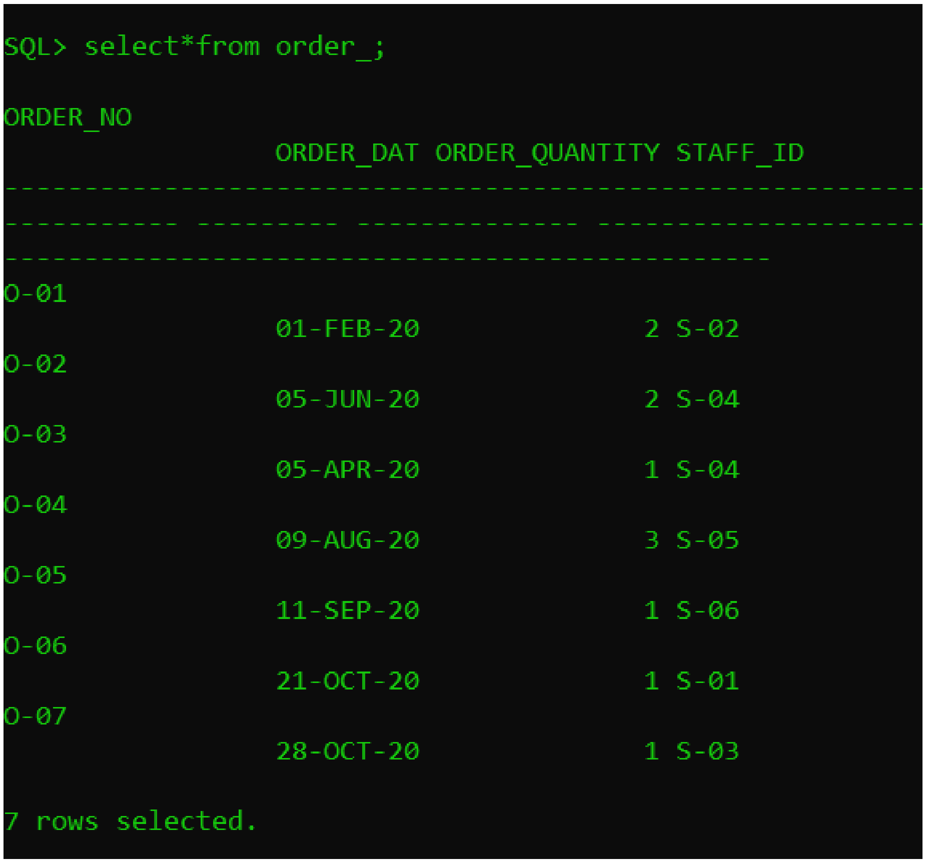
```
                                382923 mayor@yahoo.com
Regular

A-06
                 Axl Rose
Lalitpur
                                344566 axl@yahoo.com
Loyal

A-07
                 David Gilmour
Kathmandu
                               2456656 david@yahoo.com
Loyal


7 rows selected.
```

Figure 29 Data in Customer table (continued)



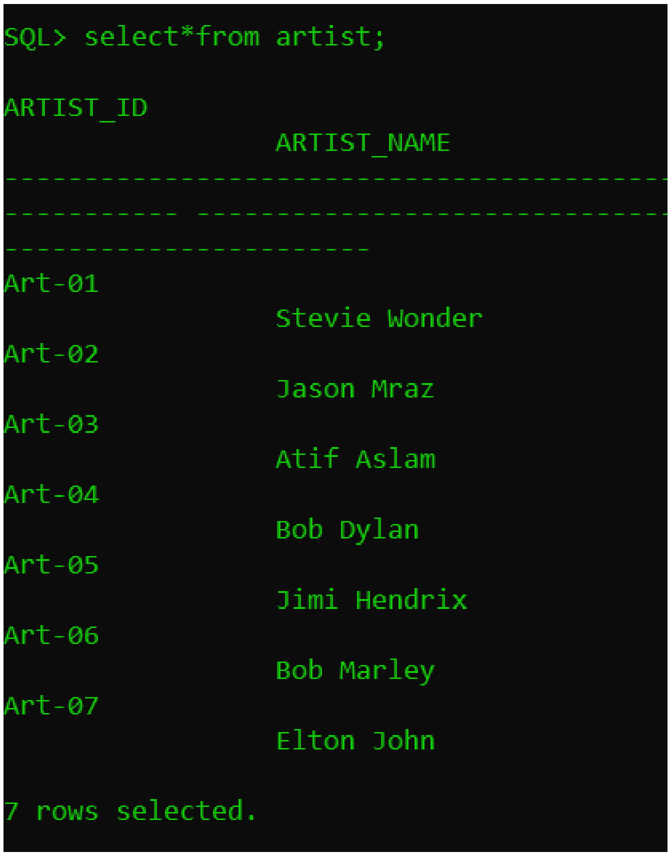
```
SQL> select*from staff;

STAFF_ID
                 STAFF_NAME
                                  STAFF_SALARY
--------------------------------------------------------------------
----------- --------------------------------------------------------
----------------------- ------------
S-01
                 Robert Plant
                                         15000
S-02
                 Eric Clapton
                                         13000
S-03
                 John Bonham
                                         11000
S-04
                 Angus Young
                                         16000
S-05
                 Brian Johnson
                                         17000
S-06
                 Chris Martin
                                         10000
S-07
                 Chris Cornell
                                         15000

7 rows selected.
```

Figure 30 Data in Staff table



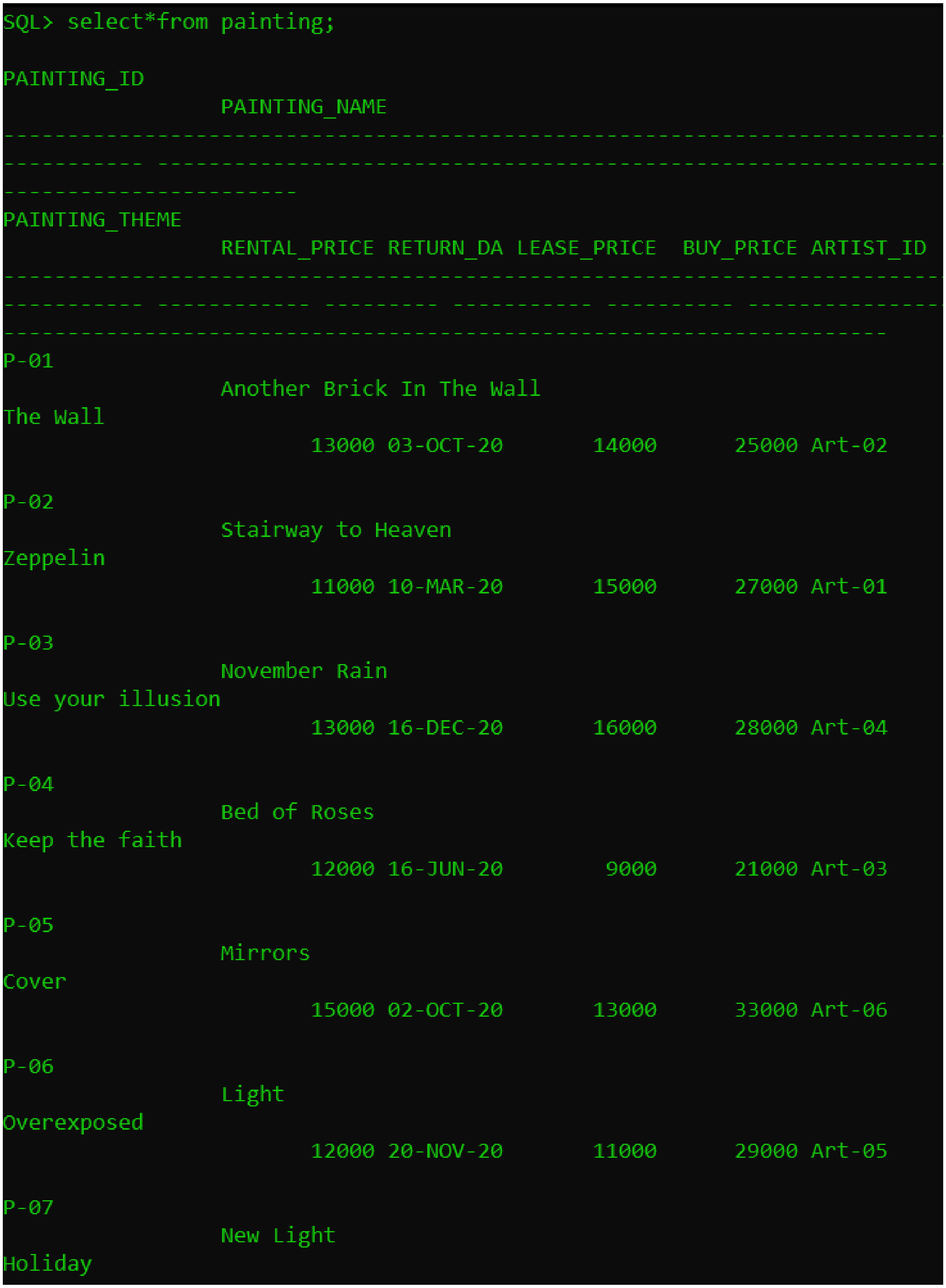
```
SQL> select*from order_;

ORDER_NO
                 ORDER_DAT ORDER_QUANTITY STAFF_ID
---------------------------------------------------------
----------- --------- -------------- --------------------
------------------------------------------------
O-01
                 01-FEB-20              2 S-02
O-02
                 05-JUN-20              2 S-04
O-03
                 05-APR-20              1 S-04
O-04
                 09-AUG-20              3 S-05
O-05
                 11-SEP-20              1 S-06
O-06
                 21-OCT-20              1 S-01
O-07
                 28-OCT-20              1 S-03

7 rows selected.
```

Figure 31 Data in Order_ table



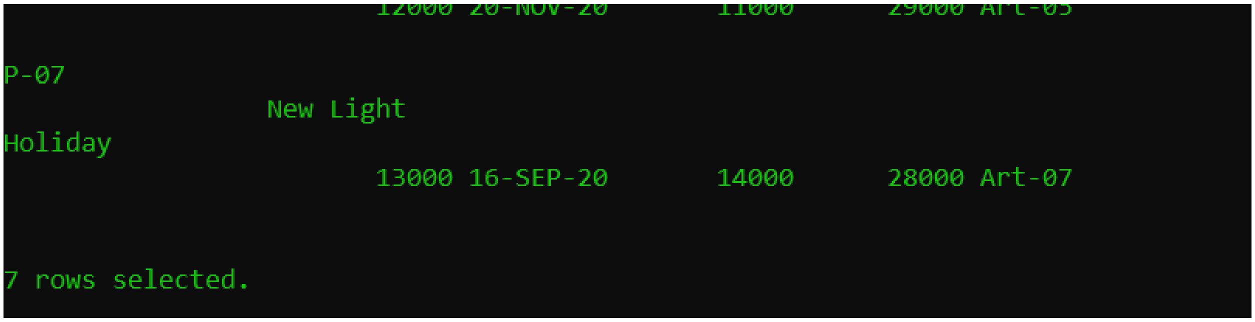
```
SQL> select*from artist;

ARTIST_ID
                 ARTIST_NAME
------------------------------------------
----------- ------------------------------
-----------------------
Art-01
                 Stevie Wonder
Art-02
                 Jason Mraz
Art-03
                 Atif Aslam
Art-04
                 Bob Dylan
Art-05
                 Jimi Hendrix
Art-06
                 Bob Marley
Art-07
                 Elton John

7 rows selected.
```

Figure 32 Data in Artist table



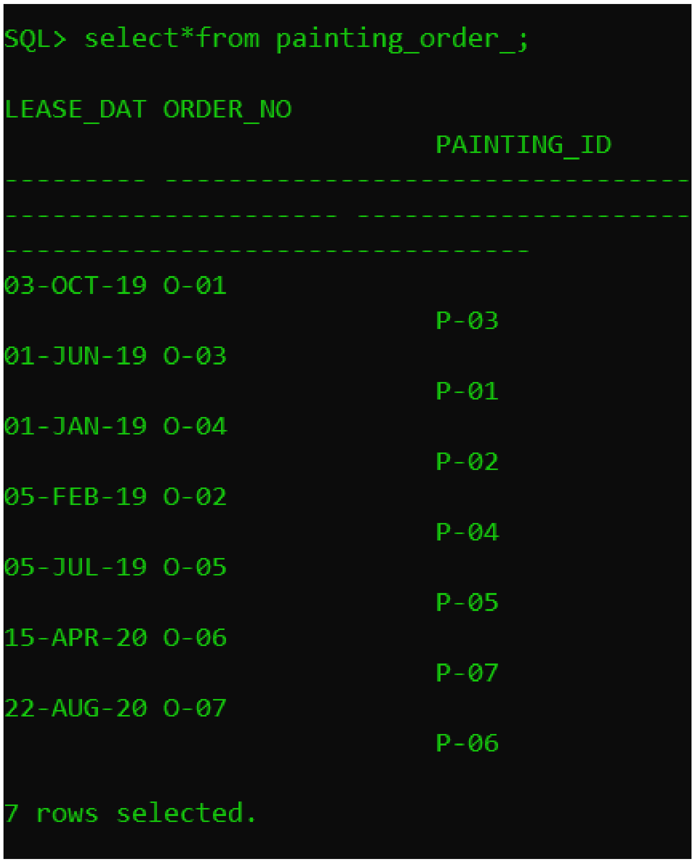
```
SQL> select*from painting;

PAINTING_ID
                 PAINTING_NAME
-------------------------------------------------------------------------
----------- -------------------------------------------------------------
-----------------------
PAINTING_THEME
                 RENTAL_PRICE RETURN_DA LEASE_PRICE  BUY_PRICE ARTIST_ID
-------------------------------------------------------------------------
----------- ------------ --------- ----------- ---------- ---------------
---------------------------------------------------------------------
P-01
                 Another Brick In The Wall
The Wall
                        13000 03-OCT-20       14000      25000 Art-02

P-02
                 Stairway to Heaven
Zeppelin
                        11000 10-MAR-20       15000      27000 Art-01

P-03
                 November Rain
Use your illusion
                        13000 16-DEC-20       16000      28000 Art-04

P-04
                 Bed of Roses
Keep the faith
                        12000 16-JUN-20        9000      21000 Art-03

P-05
                 Mirrors
Cover
                        15000 02-OCT-20       13000      33000 Art-06

P-06
                 Light
Overexposed
                        12000 20-NOV-20       11000      29000 Art-05

P-07
                 New Light
Holiday
```

Figure 33 Data in Painting table



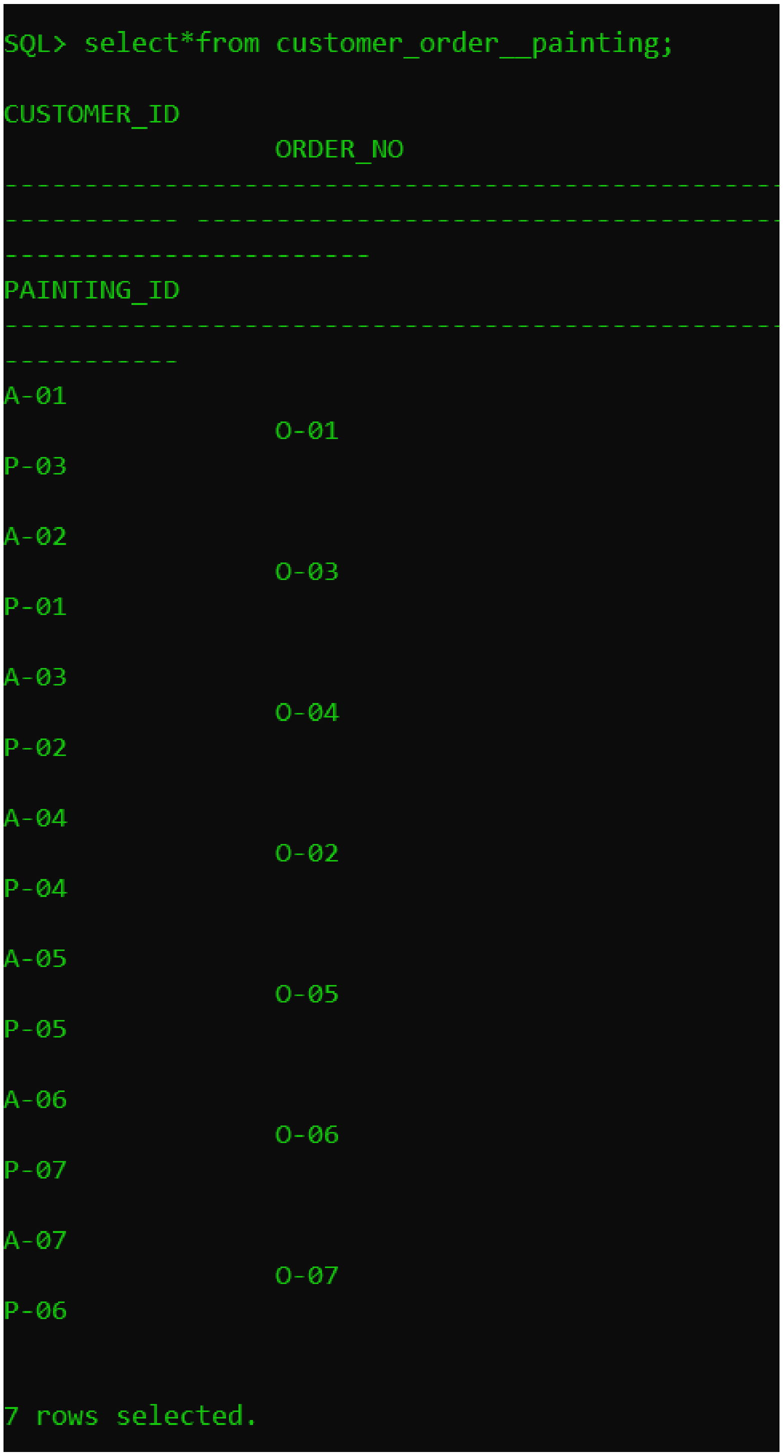
```
                        12000 20-NOV-20         11000       29000 Art-05

P-07
                 New Light
Holiday
                        13000 16-SEP-20         14000       28000 Art-07


7 rows selected.
```

Figure 34 Data in Painting table (Continued)



```
SQL> select*from painting_order_;

LEASE_DAT ORDER_NO
                           PAINTING_ID
--------- ---------------------------------
--------------------- ---------------------
---------------------------------
03-OCT-19 O-01
                           P-03
01-JUN-19 O-03
                           P-01
01-JAN-19 O-04
                           P-02
05-FEB-19 O-02
                           P-04
05-JUL-19 O-05
                           P-05
15-APR-20 O-06
                           P-07
22-AUG-20 O-07
                           P-06

7 rows selected.
```

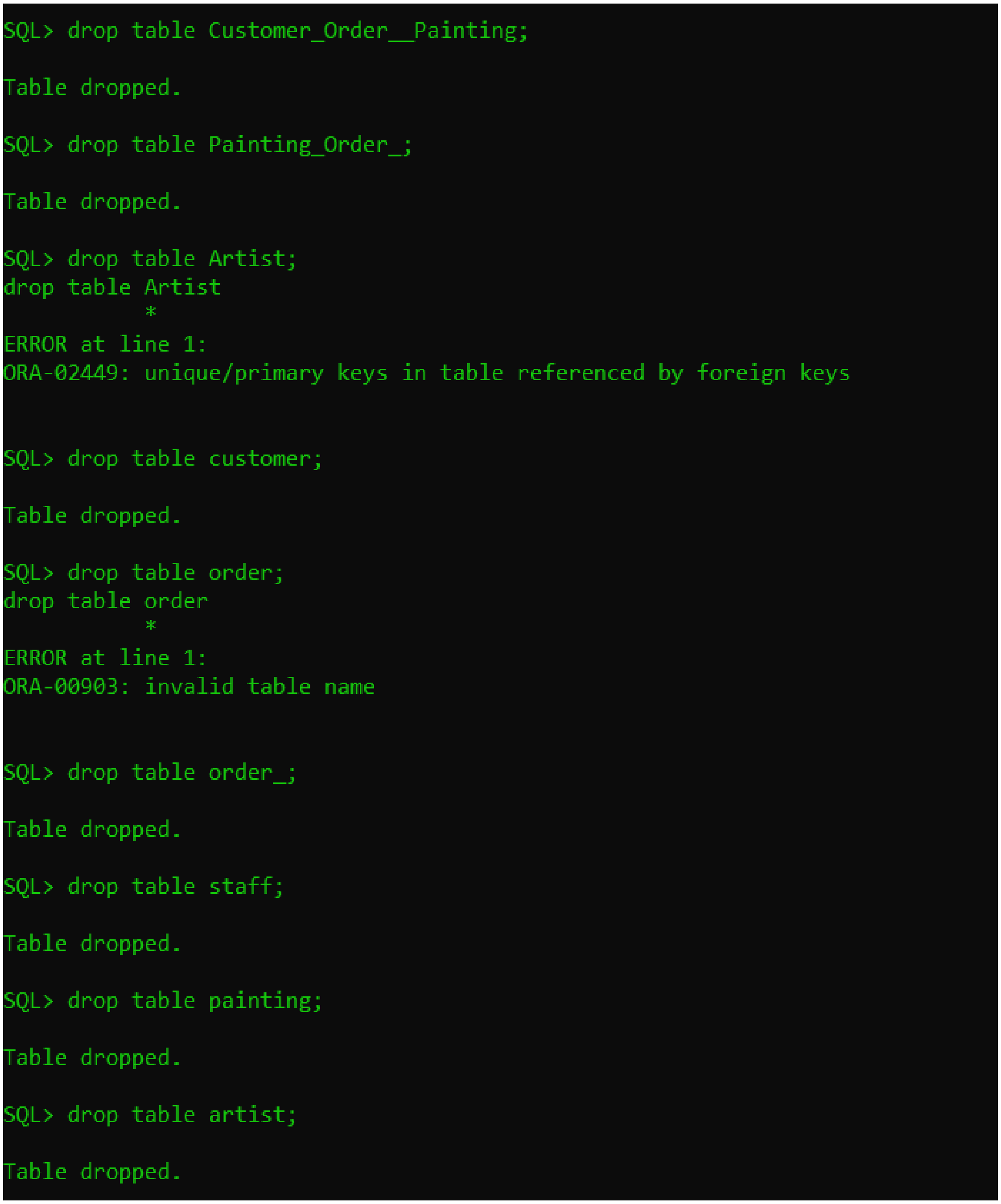
Figure 35 Data in Painting_Order_ table



```
SQL> select*from customer_order__painting;

CUSTOMER_ID
                 ORDER_NO
-------------------------------------------------
----------- ------------------------------------
-----------------------
PAINTING_ID
-------------------------------------------------
-----------
A-01
                 O-01
P-03

A-02
                 O-03
P-01

A-03
                 O-04
P-02

A-04
                 O-02
P-04

A-05
                 O-05
P-05

A-06
                 O-06
P-07

A-07
                 O-07
P-06


7 rows selected.
```

Figure 36 Data in Customer_Order__Painting table

5.3 Drop tables:



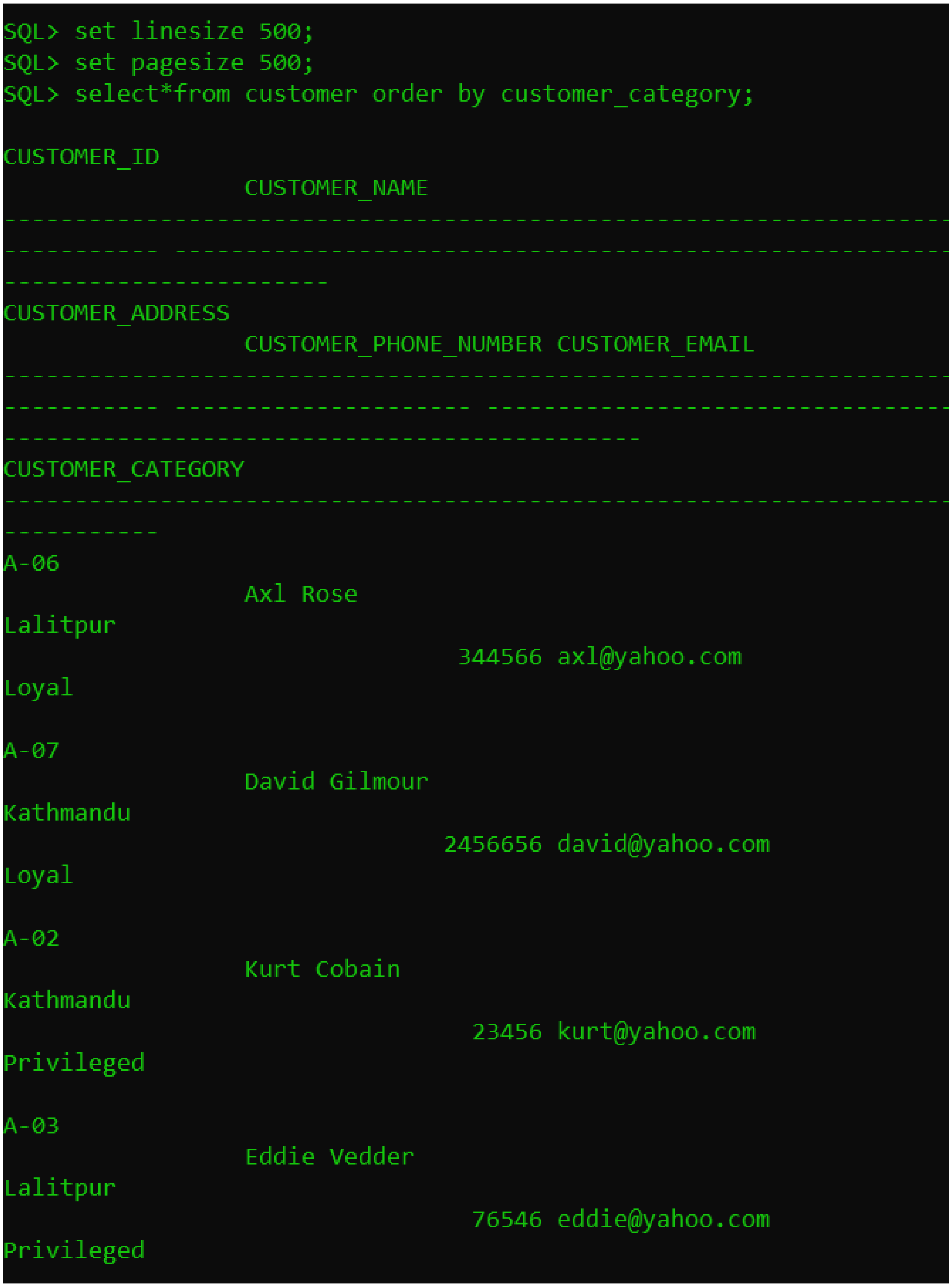
```
SQL> drop table Customer_Order__Painting;

Table dropped.

SQL> drop table Painting_Order_;

Table dropped.

SQL> drop table Artist;
drop table Artist
           *
ERROR at line 1:
ORA-02449: unique/primary keys in table referenced by foreign keys


SQL> drop table customer;

Table dropped.

SQL> drop table order;
drop table order
           *
ERROR at line 1:
ORA-00903: invalid table name


SQL> drop table order_;

Table dropped.

SQL> drop table staff;

Table dropped.

SQL> drop table painting;

Table dropped.

SQL> drop table artist;

Table dropped.
```

Figure 37 Dropping table in order

6. Database Querying:

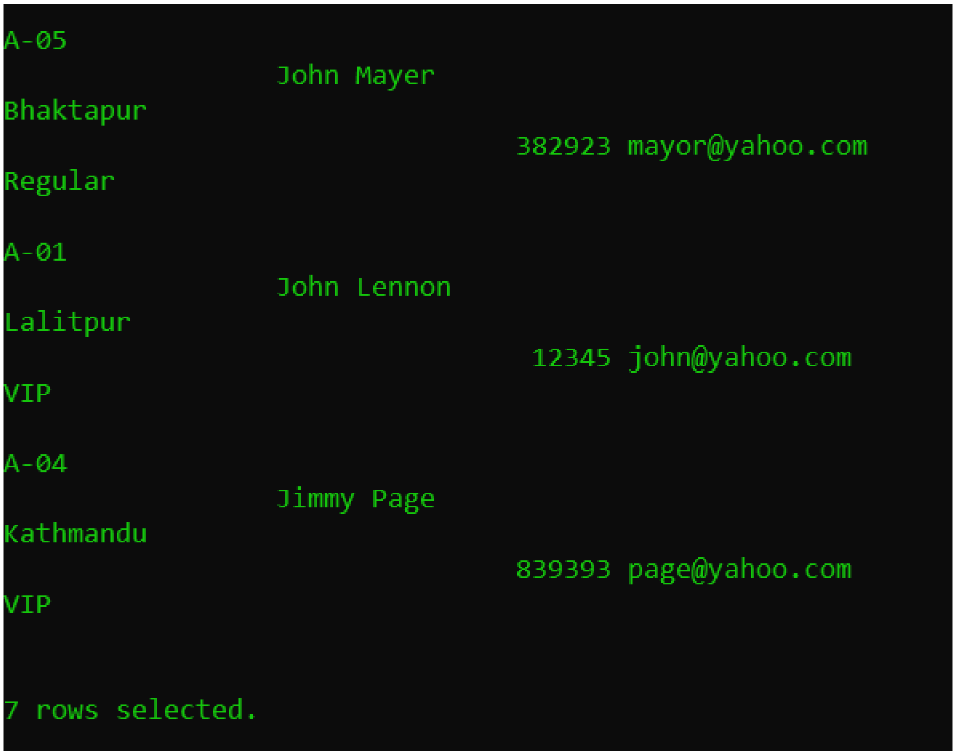
A) Information Query

1) List all customers according to category



```
SQL> set linesize 500;
SQL> set pagesize 500;
SQL> select*from customer order by customer_category;

CUSTOMER_ID
                 CUSTOMER_NAME
-------------------------------------------------------------------
----------- -------------------------------------------------------
-----------------------
CUSTOMER_ADDRESS
                 CUSTOMER_PHONE_NUMBER CUSTOMER_EMAIL
-------------------------------------------------------------------
----------- -------------------- ----------------------------------
---------------------------------------------
CUSTOMER_CATEGORY
-------------------------------------------------------------------
-----------
A-06
                 Axl Rose
Lalitpur
                                300344566 axl@yahoo.com
Loyal

A-07
                 David Gilmour
Kathmandu
                               2456656 david@yahoo.com
Loyal

A-02
                 Kurt Cobain
Kathmandu
                                 23456 kurt@yahoo.com
Privileged

A-03
                 Eddie Vedder
Lalitpur
                                 76546 eddie@yahoo.com
Privileged
```

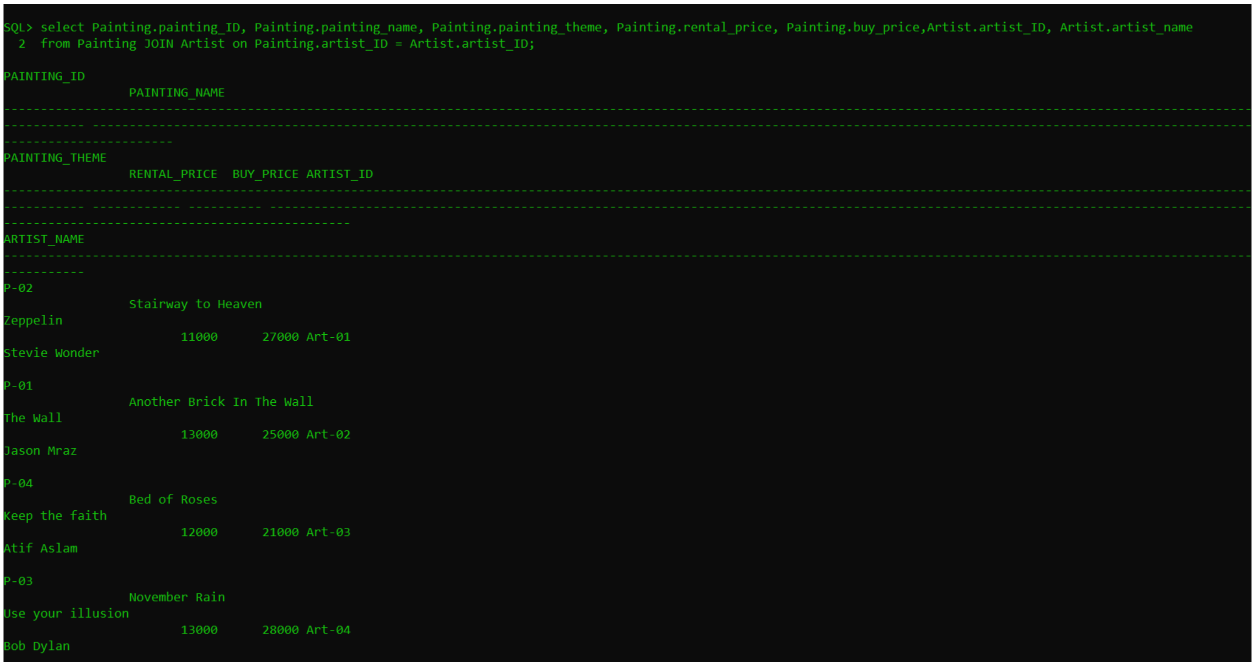
Figure 38 List all customers according to category



```
A-05
                 John Mayer
Bhaktapur
                                382923 mayor@yahoo.com
Regular

A-01
                 John Lennon
Lalitpur
                                 12345 john@yahoo.com
VIP

A-04
                 Jimmy Page
Kathmandu
                                839393 page@yahoo.com
VIP


7 rows selected.
```

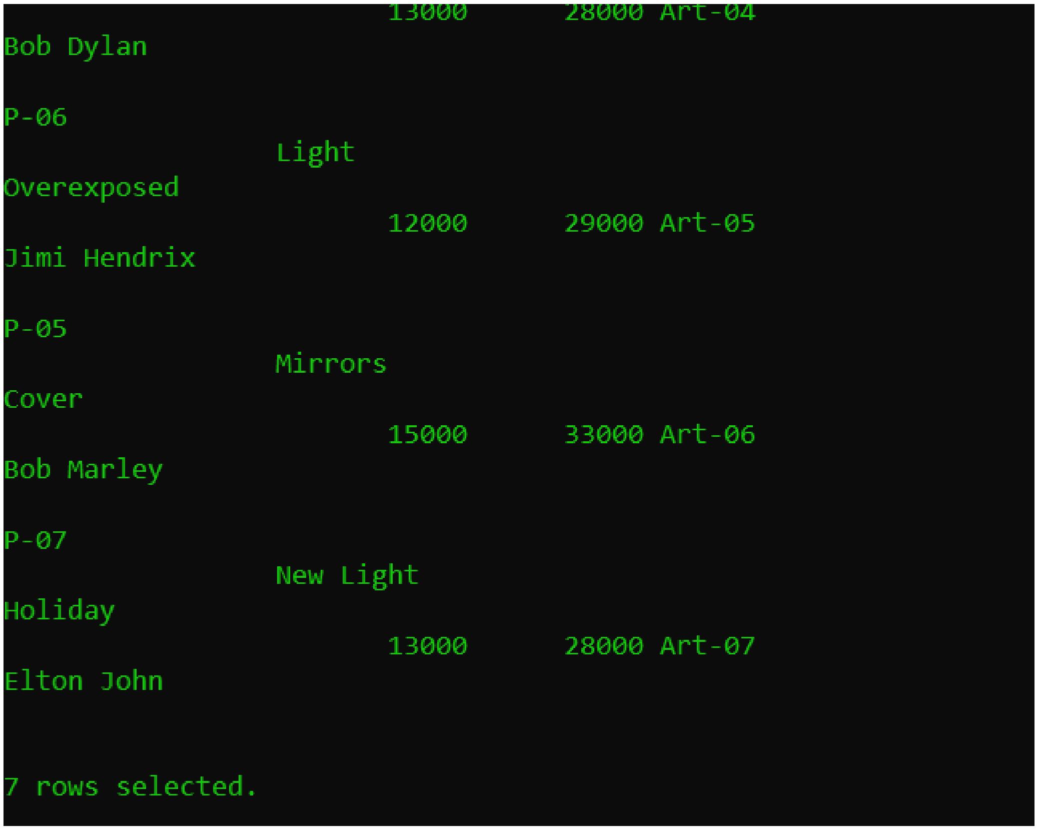
Figure 39 List all customers according to category(continued)

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



```
SQL> select Painting.painting_ID, Painting.painting_name, Painting.painting_theme, Painting.rental_price, Painting.buy_price,Artist.artist_ID, Artist.artist_name
  2  from Painting JOIN Artist on Painting.artist_ID = Artist.artist_ID;

PAINTING_ID
                 PAINTING_NAME
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
----------- ----------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------
PAINTING_THEME
                 RENTAL_PRICE  BUY_PRICE ARTIST_ID
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
----------- ------------ ---------- -----------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------
ARTIST_NAME
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------
P-02
                 Stairway to Heaven
Zeppelin
                        11000      27000 Art-01
Stevie Wonder

P-01
                 Another Brick In The Wall
The Wall
                        13000      25000 Art-02
Jason Mraz

P-04
                 Bed of Roses
Keep the faith
                        12000      21000 Art-03
Atif Aslam

P-03
                 November Rain
Use your illusion
                        13000      28000 Art-04
Bob Dylan
```

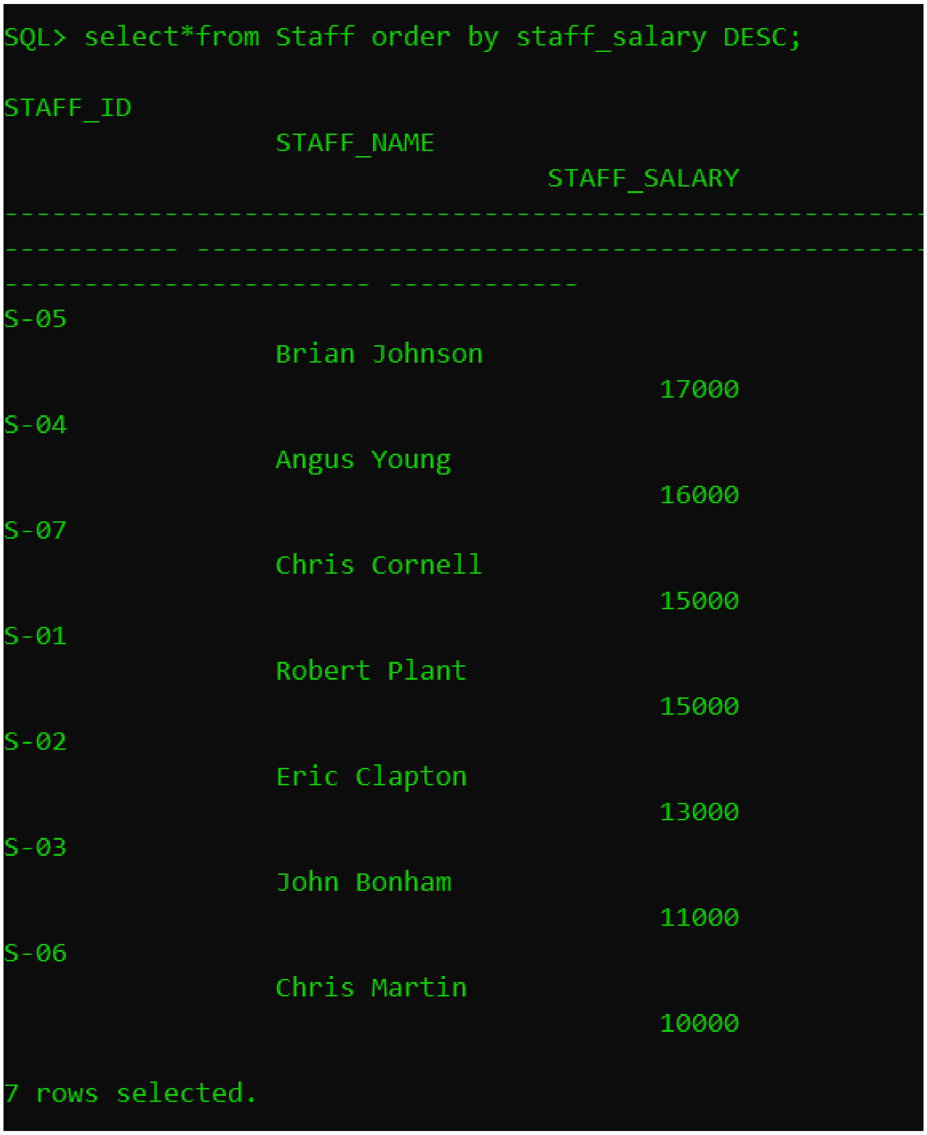
Figure 40 List paintings and their artist with monthly rental price and paid price.



```
                        13000      28000 Art-04
Bob Dylan

P-06
                 Light
Overexposed
                        12000      29000 Art-05
Jimi Hendrix

P-05
                 Mirrors
Cover
                        15000      33000 Art-06
Bob Marley

P-07
                 New Light
Holiday
                        13000      28000 Art-07
Elton John


7 rows selected.
```

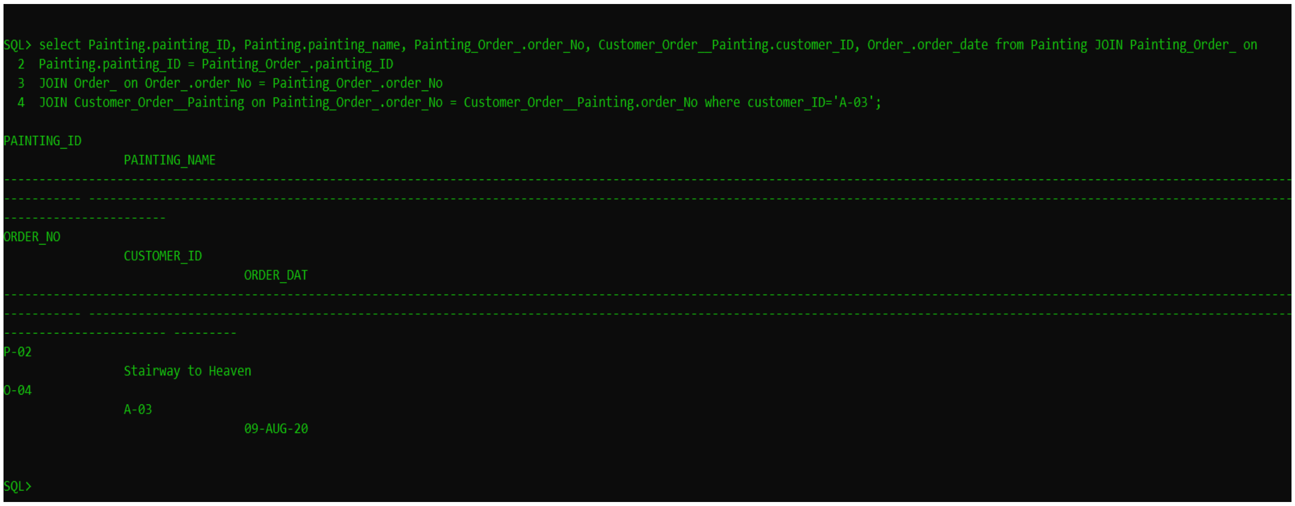
Figure 41 List paintings and their artist with monthly rental price and paid price. (continued)

3) Show total staff in Masterpieces Limited sorted by higher salary



```
SQL> select*from Staff order by staff_salary DESC;

STAFF_ID
                 STAFF_NAME
                                  STAFF_SALARY
----------------------------------------------------------
----------- ----------------------------------------------
----------------------- ------------
S-05
                 Brian Johnson
                                         17000
S-04
                 Angus Young
                                         16000
S-07
                 Chris Cornell
                                         15000
S-01
                 Robert Plant
                                         15000
S-02
                 Eric Clapton
                                         13000
S-03
                 John Bonham
                                         11000
S-06
                 Chris Martin
                                         10000

7 rows selected.
```

4) Show paintings leased before and currently by any one customer

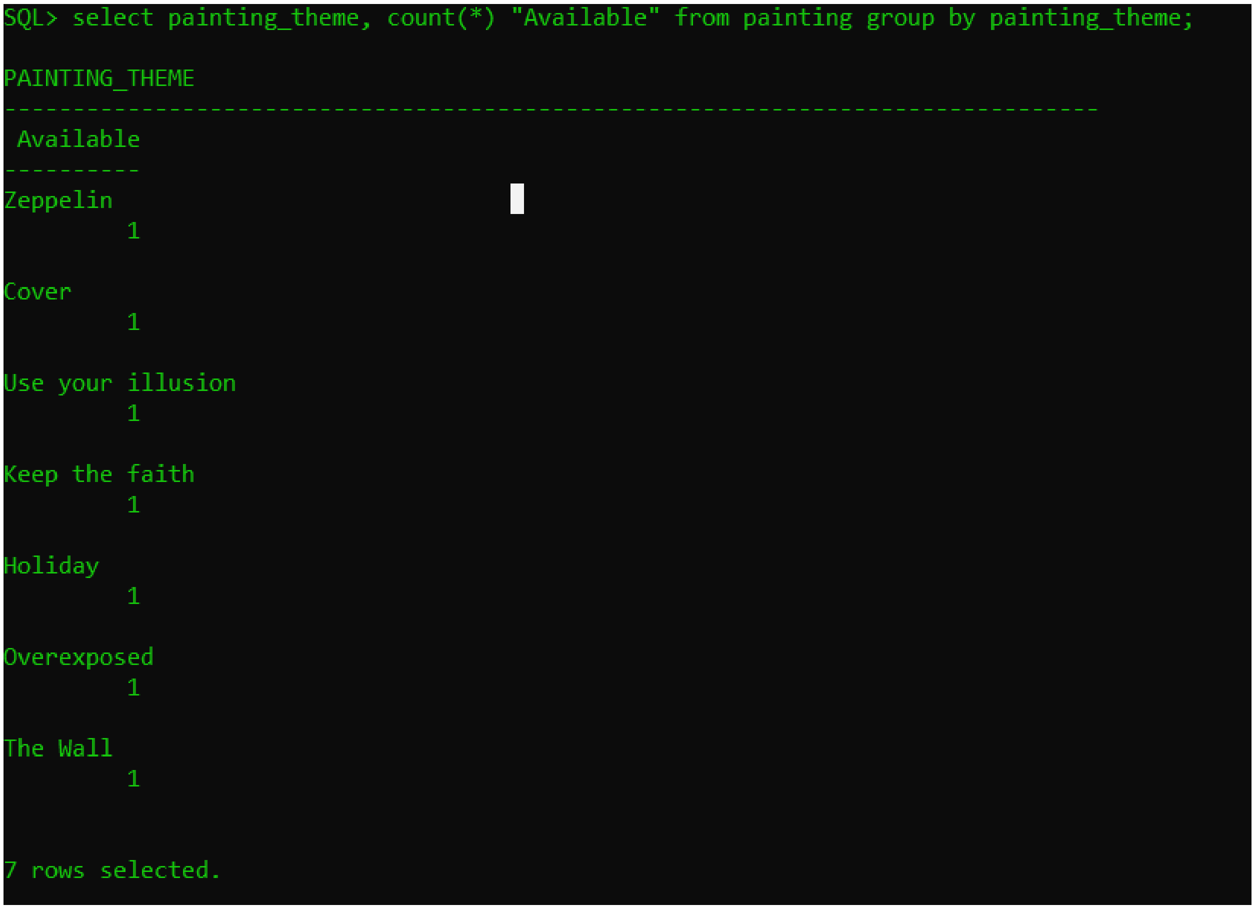


```
SQL> select Painting.painting_ID, Painting.painting_name, Painting_Order_.order_No, Customer_Order__Painting.customer_ID, Order_.order_date from Painting JOIN Painting_Order_ on
  2  Painting.painting_ID = Painting_Order_.painting_ID
  3  JOIN Order_ on Order_.order_No = Painting_Order_.order_No
  4  JOIN Customer_Order__Painting on Painting_Order_.order_No = Customer_Order__Painting.order_No where customer_ID='A-03';

PAINTING_ID
                 PAINTING_NAME
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
----------- ----------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------
ORDER_NO
                 CUSTOMER_ID
                                  ORDER_DAT
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
----------- ----------------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------- ---------
P-02
                 Stairway to Heaven
O-04
                 A-03
                                  09-AUG-20


SQL>
```

Figure 42 Show paintings leased before and currently by any one customer.

B) Transaction query:

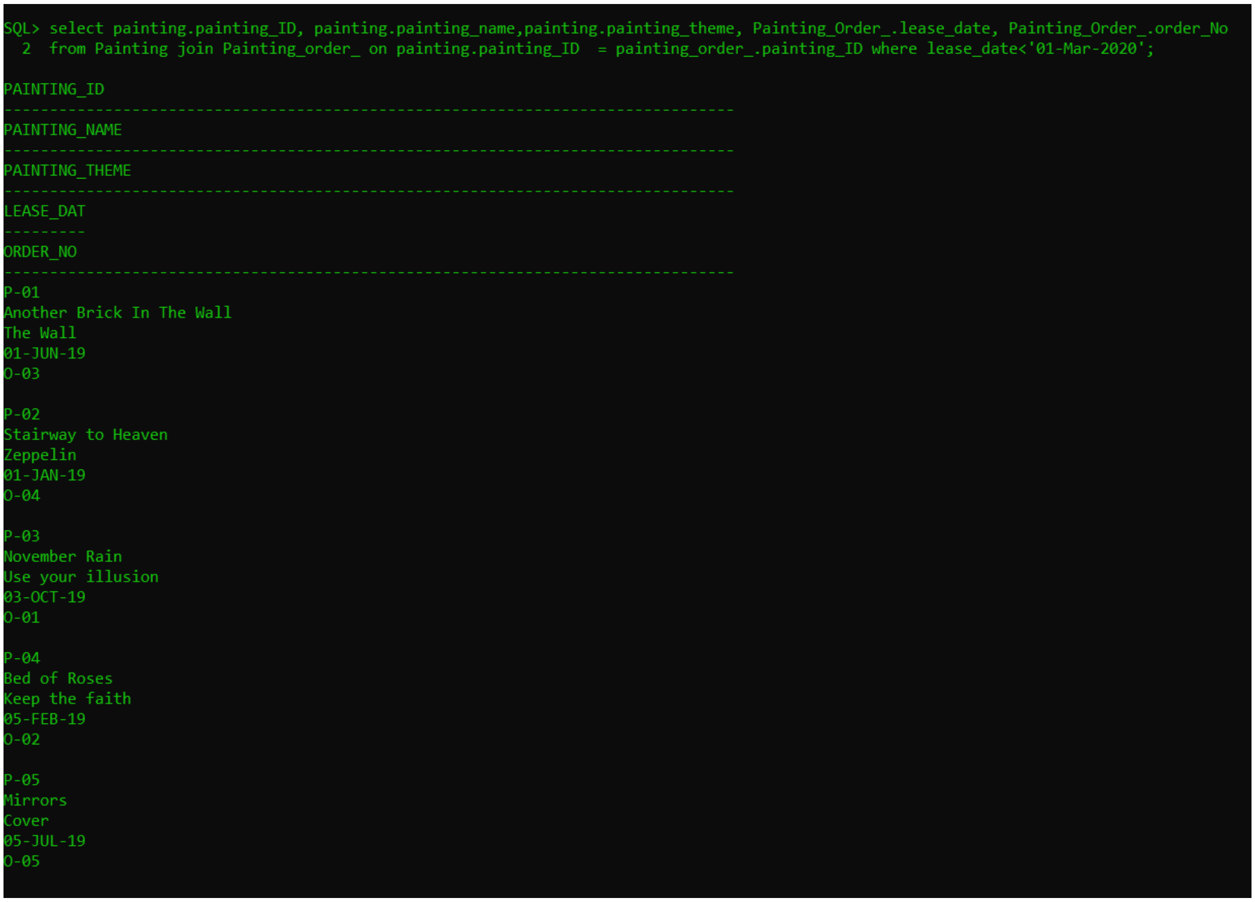
1) List the number of paintings available for rent according to category



```
SQL> select painting_theme, count(*) "Available" from painting group by painting_theme;

PAINTING_THEME
--------------------------------------------------------------------------------
 Available
----------
Zeppelin
         1

Cover
         1

Use your illusion
         1

Keep the faith
         1

Holiday
         1

Overexposed
         1

The Wall
         1


7 rows selected.
```

Figure 43 List the number of paintings available for rent according to category.

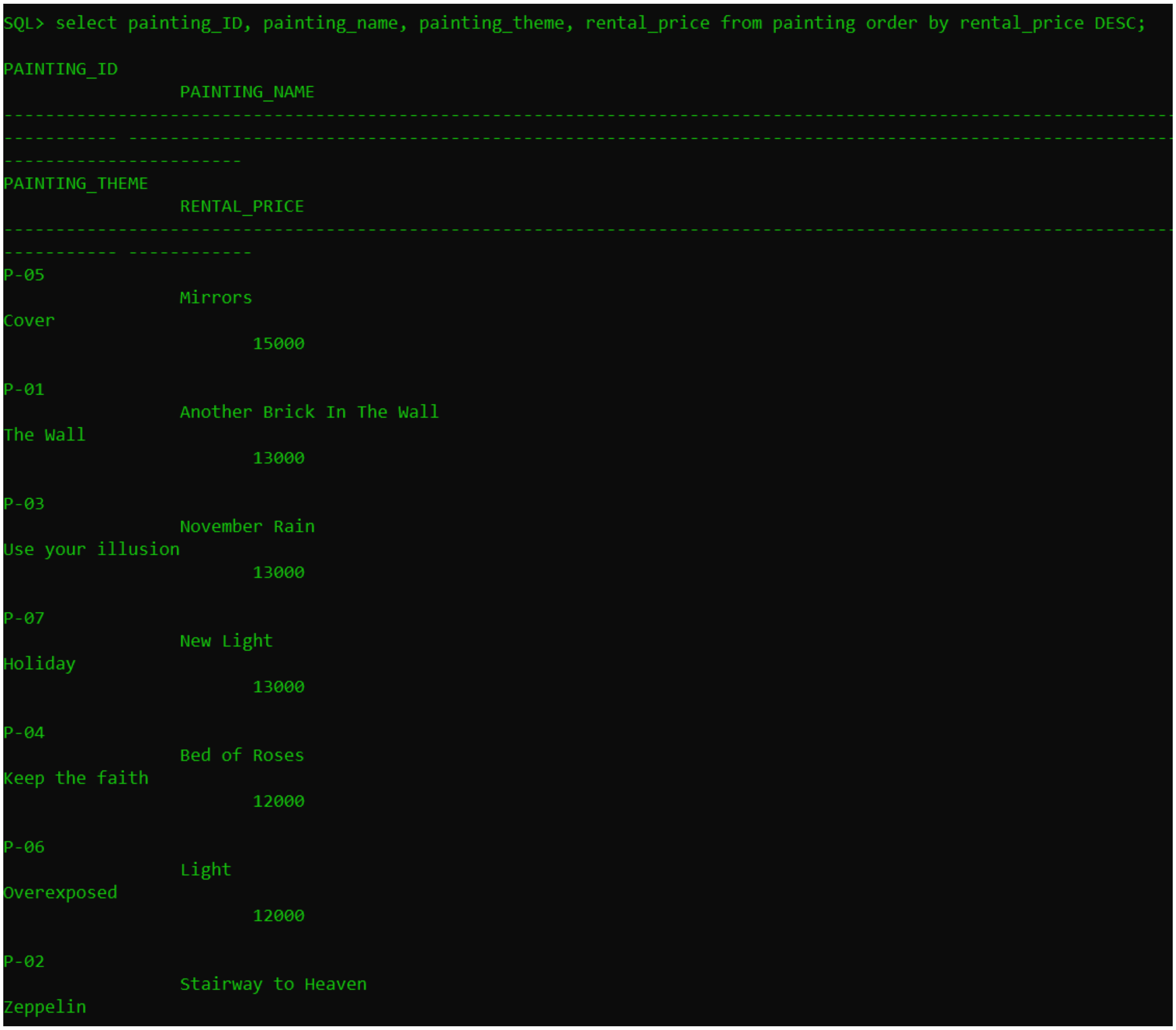
2) List the details of paintings that have not been leased within three months



```
SQL> select painting.painting_ID, painting.painting_name,painting.painting_theme, Painting_Order_.lease_date, Painting_Order_.order_No
  2  from Painting join Painting_order_ on painting.painting_ID  = painting_order_.painting_ID where lease_date<'01-Mar-2020';

PAINTING_ID
--------------------------------------------------------------------------------
PAINTING_NAME
--------------------------------------------------------------------------------
PAINTING_THEME
--------------------------------------------------------------------------------
LEASE_DAT
---------
ORDER_NO
--------------------------------------------------------------------------------
P-01
Another Brick In The Wall
The Wall
01-JUN-19
O-03

P-02
Stairway to Heaven
Zeppelin
01-JAN-19
O-04

P-03
November Rain
Use your illusion
03-OCT-19
O-01

P-04
Bed of Roses
Keep the faith
05-FEB-19
O-02

P-05
Mirrors
Cover
05-JUL-19
O-05
```

Figure 44 List the details of paintings that have not been leased within three months

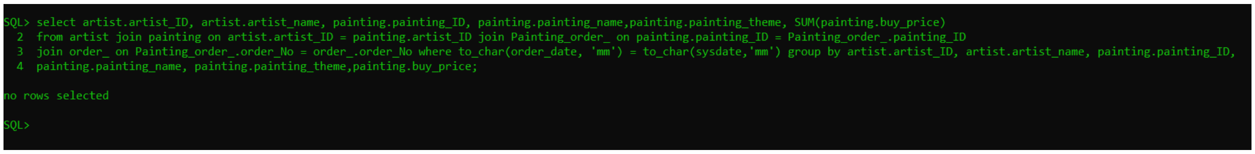
3) List top 5 paintings based on total collected rental amount



```
SQL> select painting_ID, painting_name, painting_theme, rental_price from painting order by rental_price DESC;

PAINTING_ID
                 PAINTING_NAME
----------------------------------------------------------------------------------------------------------------
----------- ----------------------------------------------------------------------------------------------------
-----------------------
PAINTING_THEME
                 RENTAL_PRICE
----------------------------------------------------------------------------------------------------------------
----------- ------------
P-05
                 Mirrors
Cover
                        15000

P-01
                 Another Brick In The Wall
The Wall
                        13000

P-03
                 November Rain
Use your illusion
                        13000

P-07
                 New Light
Holiday
                        13000

P-04
                 Bed of Roses
Keep the faith
                        12000

P-06
                 Light
Overexposed
                        12000

P-02
                 Stairway to Heaven
Zeppelin
```

Figure 45 List top 5 paintings based on total collected rental amount.

4) Show the name of the painter and their paintings sold value (in total) for the current month.



```
SQL> select artist.artist_ID, artist.artist_name, painting.painting_ID, painting.painting_name,painting.painting_theme, SUM(painting.buy_price)
     from artist join painting on artist.artist_ID = painting.artist_ID join Painting_order_ on painting.painting_ID = Painting_order_.painting_ID
     join order_ on Painting_order_.order_No = order_.order_No where to_char(order_date, 'mm') = to_char(sysdate,'mm') group by artist.artist_ID, artist.artist_name, painting.painting_ID,
     painting.painting_name, painting.painting_theme,painting.buy_price;

no rows selected

SQL>
```

Figure 46 Show the name of the painter and their paintings sold value (in total) for the current month

7. Critical Evaluation:

7.1 Critical Evaluation of Module:

The module was a great topic on how we can and should be managing a database of a company. The database module assisted in making a database for a company named Masterpieces Limited which is an Internet based company. With the help of this module, we now can design a database for a company. Database wasn't completely a new topic as we had some experiences of making ERD and database in the first year itself. But what really was useful and new was the proper way of designing a database for a company.

The normalization process was new and quite difficult to implement it in real life scenario. This process avoided data redundancy in the database and helped us get the correct and the most accurate form of data. The module itself is a very vast topic as database comes along with other modules and other sector of Information Technology as well. In software engineering it is very important to know about the ERD (Entity Relationship Diagram). And if the ERD is not correct then it might cause a huge loss of time, money and resources while developing a software for a company. In programming, database is used to store the data and extract those data where necessary. For example, while displaying records of the employees, a GUI might be used. But that GUI extracts information from the database of that company and displays it via the GUI. Similarly, different queries performed and learned helped us to extract the information easily and efficiently.

Moreover, database is safe and easy to store huge amount of data. The proper learning of this module was made possible with the help of our module lecturers and tutors by helping us with our difficulties and wherever possible.

7.2 Critical Assessment of Course Work:

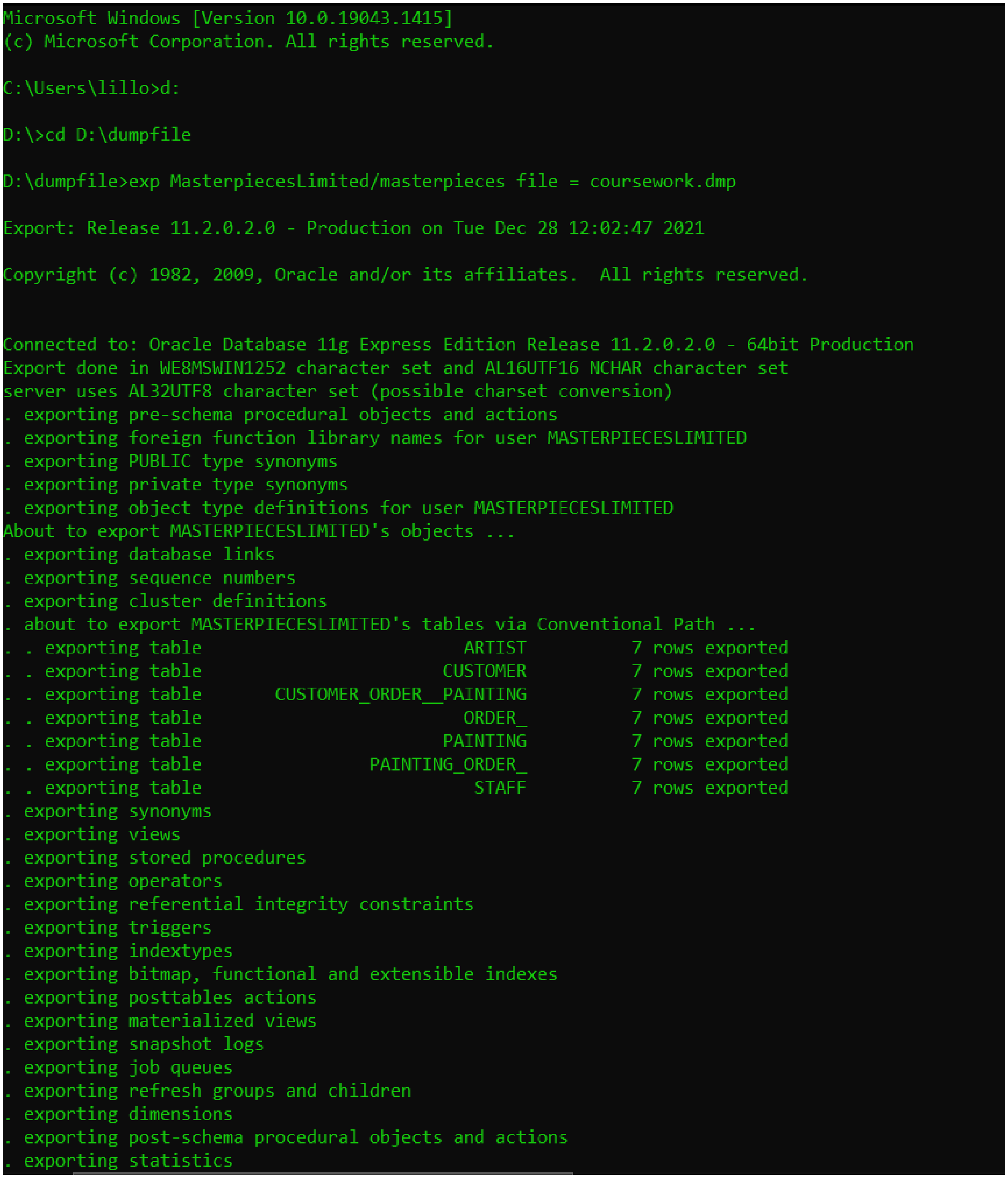
The coursework was designing a database for a company named Masterpieces Limited. The company is an Internet based business which provides paintings on lease or even lets the customers to buy them. The company had some certain business rules to be followed. And the database designed was also abided by some assumptions made while designing the database.

Firstly, an initial ERD (Entity Relationship Diagram) was made according to the given scenario. This initial ERD had its entities, and those entities had their attributes. But the problem was with data redundancy. To remove this problem The initial ERD went through normalization process. Where repeating groups and repeating data were separated, and similarly partial dependencies and transitive dependencies were also sorted out reducing the data redundancy. A final ERD was designed after the 3rd normal form(3NF). A total of 7 different tables with their respective primary keys and foreign keys were formed. Then with the help of these available resources and information, tables were created based on the entities formed after the 3rd normal form was completed. Different values were given to the attributes of the table accordingly. After the completion of inserting data into tables, different queries were performed. The database designed can manage the details of customers, staffs, orders, paintings, and availability of the paintings. The system is also capable of allowing the customers to buy or take the paintings on lease. Also, the customers are divided into different categories and when a customer asks for discount, it can be given using queries. The system is also capable of providing the name of the artist and making it is easier and convenient for the customer to choose their painting with the theme of the painting or with the name of the painting itself. Overall, the assessment was a great assist to gain knowledge on proper implementation of database in real life scenario.

8. Structure and Formatting:

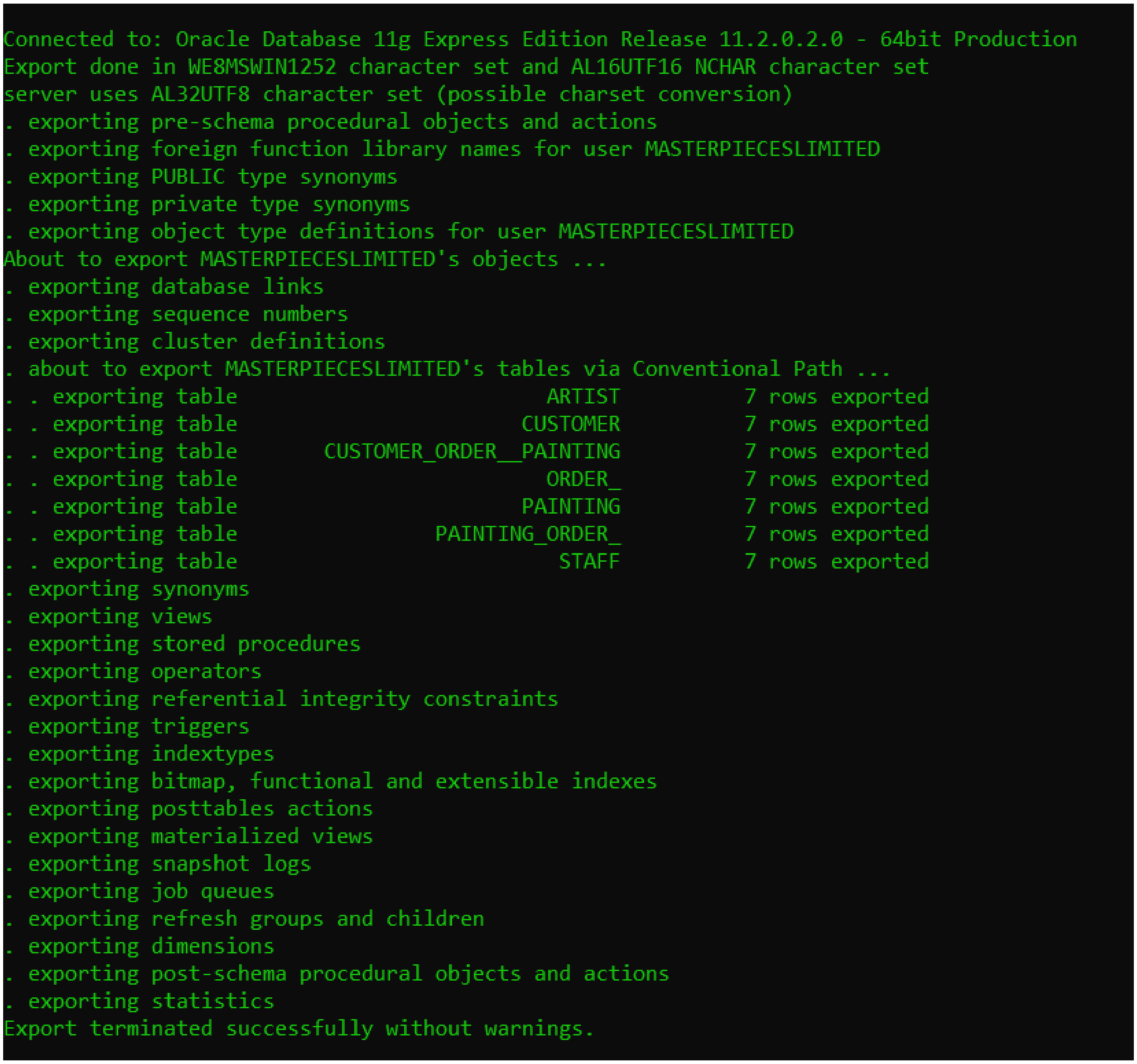
The dump file has been attached to the coursework file.

9. Database dump file creation:



```
Microsoft Windows [Version 10.0.19043.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lillo>d:

D:\>cd D:\dumpfile

D:\dumpfile>exp MasterpiecesLimited/masterpieces file = coursework.dmp

Export: Release 11.2.0.2.0 - Production on Tue Dec 28 12:02:47 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 MASTERPIECESLIMITED
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user MASTERPIECESLIMITED
About to export MASTERPIECESLIMITED's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export MASTERPIECESLIMITED's tables via Conventional Path ...
. . exporting table                         ARTIST          7 rows exported
. . exporting table                       CUSTOMER          7 rows exported
. . exporting table       CUSTOMER_ORDER__PAINTING          7 rows exported
. . exporting table                         ORDER_          7 rows exported
. . exporting table                       PAINTING          7 rows exported
. . exporting table                PAINTING_ORDER_          7 rows exported
. . exporting table                          STAFF          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
```

Figure 47 Database dump file creation



```
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 MASTERPIECESLIMITED
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user MASTERPIECESLIMITED
About to export MASTERPIECESLIMITED's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export MASTERPIECESLIMITED's tables via Conventional Path ...
. . exporting table                         ARTIST          7 rows exported
. . exporting table                       CUSTOMER          7 rows exported
. . exporting table        CUSTOMER_ORDER__PAINTING          7 rows exported
. . exporting table                         ORDER_          7 rows exported
. . exporting table                       PAINTING          7 rows exported
. . exporting table                PAINTING_ORDER_          7 rows exported
. . exporting table                          STAFF          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.
```

Figure 48 Database dumping file creation complete without warnings