LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS5001NI Databases

Assessment Weightage & Type

50% Individual Coursework

Year and Semester

2021-22 Autumn

Student Name: Kishor Shrestha

London Met ID: 20048913

College ID: np01cp4s210161

Assignment Due Date: 2021/12/28

Assignment Submission Date: 2021/12/28

Title (Where Required):

Word Count (Task B):

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

Table of Tables

1.0 Introduction

Masterpieces Limited is an online trade platform that is owned by Mr. Steve who is a local businessman as well as real estate owner. The company focuses on paintings created by a variety of artists and painters. The company's purpose is for customers to be able to rent or buy paintings from the company as well as paintings from the artist directly. Customers would be eligible to buy or rent paintings purchased by the company. Customers would be able to rent artworks from other painters and artists from the firm as well.

Mr. Steve aims to utilize his company as a mediator to contact the painter and the consumers in an indirect way. People would be allowed to buy or rent paintings depending on their tastes, including such portraits, abstractions, and landscapes. Mr. Steve's aim with this business is to help struggling painters for getting their talent appreciated, as well as to help consumers interested in the arts and painting can have such paintings at a fair price. This company's main goal is to encourage painters by introducing themselves plus their work to a larger audience.

1.1 Activities and Operations

Presently, just a few firm operations are active on a regular basis. The company is now working as an offline business in three different regions of the valley, employing a total of 30 people. The following are the active business activities of the company:

- The company rents or sells paintings made by different local and well-known painters, and also paintings owned by the company.
- The store is open 8 hours a day, from 9 a.m. to 5 p.m.
- The artworks are supplied to individuals who've already bought or rented them between two days of the transaction.
- Every branch has a manager who is in control of all of the staff in that branch.
- The company provides several reductions to customers based on current standing, including regular, loyal, privileged, and VIP.
- Holidays are available twice a week, including Saturday and Sunday, so employees can take time off either once twice a month.
- Following the transactions, the firm receives 20% of the total cost of the painting from the customer, with the remaining passing to the owner.
- If the paintings are not bought or purchased by customers within four months of their release, they are handed to the painter.

1.2 Rules and Regulation

A company should maintain a variety of rules and regulations in order to function effectively. The company may fail if the business rules really aren't correctly established. The Masterpieces Limited has some particular company requirements in order to perform properly in the marketplace. The following are Masterpieces Limited's business rules and regulations:

- One or even more artworks could be sold by a particular member of staff at the given period.
- No two employees could sell the same painting at the same time.
- It is not possible for different people to buy the identical painting at the given time.
- Artists could rent or sell many paintings at once.
- Customers are not allowed to buy and rent the same painting at the same time.
- Customers can buy one or many artworks at the given time.
- The staff member who sold the painting is now in charge of keeping track of the painting's delivery schedule.
- Employees are not allowed to spend unplanned time off.
- Customers can rent the paintings on a monthly basis.
- There is no opportunity for a refund once the customer has purchased the paintings
- Customers can hire a same painting more than once.
- Artists can rent out their pieces of art.
- Customers will receive discounts depending on their category.
- Employees should have a valid ID card engaged on the job.

.

1.3 Identification of Entities and Attributes

Identifying the items that must be in the database—the entities and attributes—based on the criteria that you have gathered. An entity is a specific sort of person, object, or thing that must be represented in the database. It could be a real item, such as a person, a car, or an employee, or it could be a mental object, such as a corporation, a job, or a project. Each entity contains characteristics that define it. These characteristics are referred to as attributes (Millington, 2021).

Entities	Attributes
staff	staff_Name(PK), phone, address, age, salary
artist	artist_id(PK), artist_Name, address, age
category	Customer_type(PK), discount,
customer	Customer_id(PK), customer_Name, phone, address, staff_id(FK), customer_type(FK)
painting	Painting_id(PK), painting_Name, painting_theme, paid_price, book_Date, rental price, artist_id(FK)
Hire	Customer_id(FK), painting_id(FK), hire_Date, return_Date, lease_id(PK)
buy	Customer_id(FK), painting_id(FK) painting_Priceso, purchase_id(PK)

Table 1: Entities and attributes table

1.4 Identification of primary key and foreign key

The main fundamental elements of database systems concept are primary and foreign keys. By uniquely identifying entity instances, primary keys ensure entity stability. By finishing a relation among two entities, foreign keys ensure referential integrity. The following stage in constructing the fundamental data model is to:

- Determine and specify every entity's primary key properties.
- Verify connections and primary keys.
- To create foreign keys, upgrade the primary keys (kalin_martin, 2021)

The primary and foreign key that are created for the project are listed in the below table.

Entity Name	Primary key	Foreign key
staff	Staff_id	-
artist	Artist_id	-
category	Customer_type	-
customer	Customer_id	Staff_id, customer_type
painting	Painting_id	Artist_id
Hire	Lease_id	Painting_id, customer_id
buy	Purchase_id	Customer_id, painting_id

Table 2: showing the primary and foreign keys

2.0 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. The above entity relationship diagram shows that it has a number of data inconsistencies and data redundancy. There are so many to many partnerships that it is important to break down the relationship by using an associative object. There is also some irregularity and duplication of the results. If these types of problems are not resolved, the problem can occur during data addition, updating and deletion. Data normalization should be done to remove data anomalies and data redundancy.

There are four entities in total: Staff, Customers, Painting, and Artist. The color Green is used to represent entities. The red color represents primary keys, whereas the color purple represents various qualities. The color yellow represents relationships.

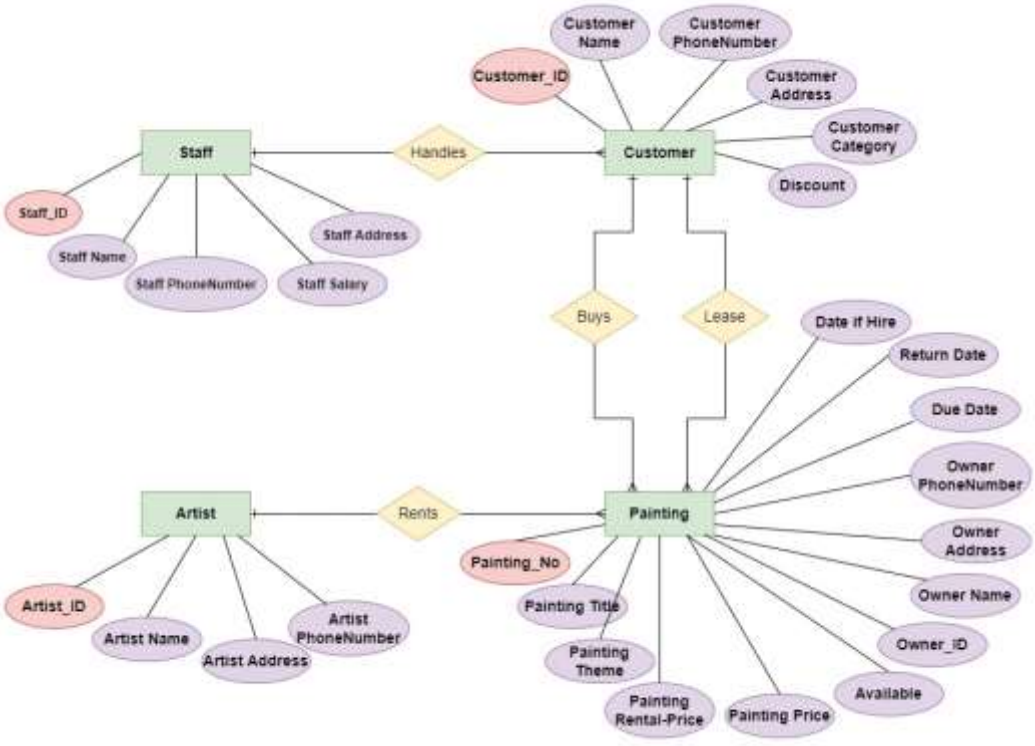

Figure 1:Inital ERD

3.0 Normalization

The procedure of restructuring information in the database such that it satisfies two minimum principles is known as normalization. There really is no data redundancy; every data is stored in a single location. Every linked data pieces are kept together in logical data connections. Normalization is critical for a variety of reasons, the most notable of which is that it enables databases to utilize as little disk space as possible, leading in improved performance. Data normalization is another name for normalization (Kant, 2021).

Normalization main objectives is to minimize the redundancy of data and to remove all unnecessary data connections and dependencies from being inserted, updated and deleted.

Normalization forms are classified into three types. They are as follows:

- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)

3.1 Unnormalized form(UNF)

Un-normalised form is a process in the normalisation method that allows us to produce a structured frame that represents a piece of organizational data such as a sheet or document (e.g invoice, report, purchase order etc.). This is our first Normalisation'relation', which includes combined real data (from the form or document) plus simulated data (based on and derived from the actual form or document) (opengrass.net, 2021).

The unnormalized form of the database is given below:

Painting (Painting_id, Painting_name, Rent_price, Selling_price, Painting_status, Issue_date, Theme_id, Painting_theme, Painter_id, Painter_name, Painter_address, Painter_contact, Transaction_date, Transaction_type, {Staff_id, Staff_name, Staff_address, Staff_contact, Job_id, Job, Salary}, {Customer_id, Cust_name, Cust_address, Cust_contact, Category_id, Category})

3.2 First Normalization form(1NF)

If a relationship has a variable that cannot be broken, it is initially in a normalized shape. The multi-value as well as composite attributes, as well as their combinations, are not permitted. Every piece of information must be distinct.

1NF Scenario:

After distinguishing between the repeating and non-repeating groups, these were subdivided in to the various distinct entities. Multiple composite primary keys were discovered using this strategy.

Entities after 1NF

Painting-1 (Painting_id, Staff_id*, Cust_id* Painting_name, Rent_price, Selling_price, Painting_status, Issue_date, Theme_id, Painting_theme, Painter_id, Painter_name, Painter_address, Painter_contact, Transaction_date, Transaction_type)

Staff-1 (Staff_id, Painting_id* Staff_name, Staff_address, Staff_contact, Job_id, Job, Salary)

Customer-1 (Customer_id, Painting_id*, Cust_name, Cust_address, Cust_contact, Category_id, Category)

3.3 Second normalization forms (2NF):

The normalized form is often applied in normalizing to minimize partial dependency. It is founded on the notion of maximal functional dependency. An object with a single primary attribute is immediately in its second normal state. A relationship that is not of the second typical kind may experience upgrade anomalies.

2NF scenario:

After 1NF, partial dependencies on the entities were examined. Additionally, the organization with partial dependencies was divided into new entities in such a way in which they seem to have whole functional dependence.

Painting (Painting_id, Staff_id*, Cust_id* Painting_name, Rent_price, Selling_price, Painting_status, Issue_date, Theme_id, Painting_theme, Painter_id, Painter_name, Painter_address, Painter_contact, Transaction_date, Transaction_type)

Painting_id, Staff_id, Cust_id → Transaction_date, Transaction_type

Staff_id, Cust_id → X

Painting_id, Staff_id → X

Painting_id, Cust_id → X

Cust_id → X

Staff_id → X

Painting_id → Painting_name, Rent_price, Selling_price, Painting_status, Issue_date, Theme_id, Painting_theme, Painter_id, Painter_name, Painter_address, Painter_contact

Staff (Staff_id, Painting_id* Staff_name, Staff_address, Staff_contact, Job_id, Job, Salary)

Staff_id, Painting_id → X

Staff_id → Staff_name, Staff_address, Staff_contact, Job_id, Job, Salary

Painting_id → X

Customer (Cust_id, Painting_id*, Cust_name, Cust_address, Cust_contact, Category_id, Category)

Cust_id, Painting_id → X

Cust_id → Cust_name, Cust_address, Cust_contactNo, Category_id, Category_name

Painting_id → X

ENTITIES AFTER 2NF:

Painting-2 (Painting_id, Painting_name, Rent_price, Selling_price, Painting_status, Issue_date, Theme_id, Painting_theme, Painter_id, Painter_name, Painter_address, Painter_contact)

Staff-2 (Staff_id, Staff_name, Staff_address, Staff_contactNo, Job_id, Job_name, Salary)

Customer-2 (Cust_id, Cust_name, Cust_address, Cust_contactNo, Category_id, Category)

Transaction-2 (Painting_id*, Staff_id*, Cust_id*, Transaction_date, Transaction_type)

Since every entity contained a composite key, all of the entities had partial dependencies; the dependencies were deleted as described above. After the dependencies have been removed, the entities are further broken down. While breaking down the entities, there could have been more tables after 2NF; however, some tables were removed to avoid repeating data multiple times. For example, a table might be generated from (Staff id, Painting id), (Painting id, Cust id), and (Staff id, Cust id), but it was not produced because the Transaction table already has these fields and it is not essential to create those tables as a result.

3.4 Third Normalization forms (3NF):

The relationship is in the third normal form if the non-key characteristics have no transitive dependency after the second normal form. This procedure would be used to remove an updated anomaly produced by a bug in the update process. By only reducing transitive dependencies can a partnership be changed from 2NF to 3NF.

3NF scenario:

After 2NF, transitive dependencies on specific things were examined. The entities with transitive dependencies have been further subdivided into new entities in order to reduce the transitive reliance.

Painting (Painting_id, Painting_name, Rent_price, Selling_price, Painting_status, Issue_date, Theme_id, Painting_theme, Painter_id, Painter_name, Painter_address, Painter_contact)

Painting_id → Painting_name, Painting_name → X

Painting_id → Rent_price, Rent_price → X

Painting_id → Selling_price, Selling_price → X

Painting_id → Painting_status, Painting_status →X

Painting_id → Issue_date, Issue_date →X

Painting_id → Theme_id, Theme_id → Painting_theme

Painting_id → Painter_id, Painter_id → Painter_name, Painter_address, Painter_contact

Staff (Staff_id, Staff_name, Staff_address, Staff_contact, Job_id, Job, Salary)

Staff_id → Staff_name, Staff_name → X

Staff_id → Staff_address, Staff-address → X

Staff_id → Staff_contact, Staff_contact → X

Staff_id → Salary, Salary → X

Staff_id → Job_id, Job_id → Job

Customer-3 (Cust_id, Cust_name, Cust_address, Cust_contact, Category_id, Category)

Cust_id → Cust_name, Cust_name → X

Cust_id → Cust_address, Cust_address → X

Cust_id → Cust_contact, Cust_contact → X

Cust_id → Category_id, Category_id → Category

Transaction (Painting_id*, Staff_id*, Cust_id*, Transaction_date, Transaction_type)

Painting_id, Staff_id, Cust_id → Transaction_date, Transaction_date → X

Painting_id, Staff_id, Cust_id → Transaction_type, Transaction_type → X

ENTITIES AFTER 3NF:

Painting-3 (Painting_id, Theme_id*, Painter_id*, Painting_name, Rent_price, Selling_price, Painting_status, Issue_date)

Theme-3 (Theme_id, Theme)

Painter-3 (Painter_id, Painter_name, Painter_address, Painter_contact)

Staff-3 (Staff_id, Job_id*, Staff_name, Staff_address, Staff_contact)

Job-3 (Job_id, Job)

Customer-3 (Cust_id, Category_id*, Cust_name, Cust_address, Cust_contact)

Category-3 (Category_id, Category)

Transaction-3 (Painting_id*, Staff_id*, Cust_id*, Transaction_date, Transaction_type)

As demonstrated above, transitive dependencies were removed by determining whether any non-key attributes referenced other non-key attributes. The Painting entity was further subdivided into three entities because two non-key attributes referenced other non-key properties. There was one non-key attribute in each entity that referred to another non-key attribute in the Customer and Staff entities as well. Because the Transaction entity contained numerous foreign keys, the foreign keys have been used as composite keys, and the entity's transitive dependency was checked.

4.0 Final ER-Diagram after Normalization

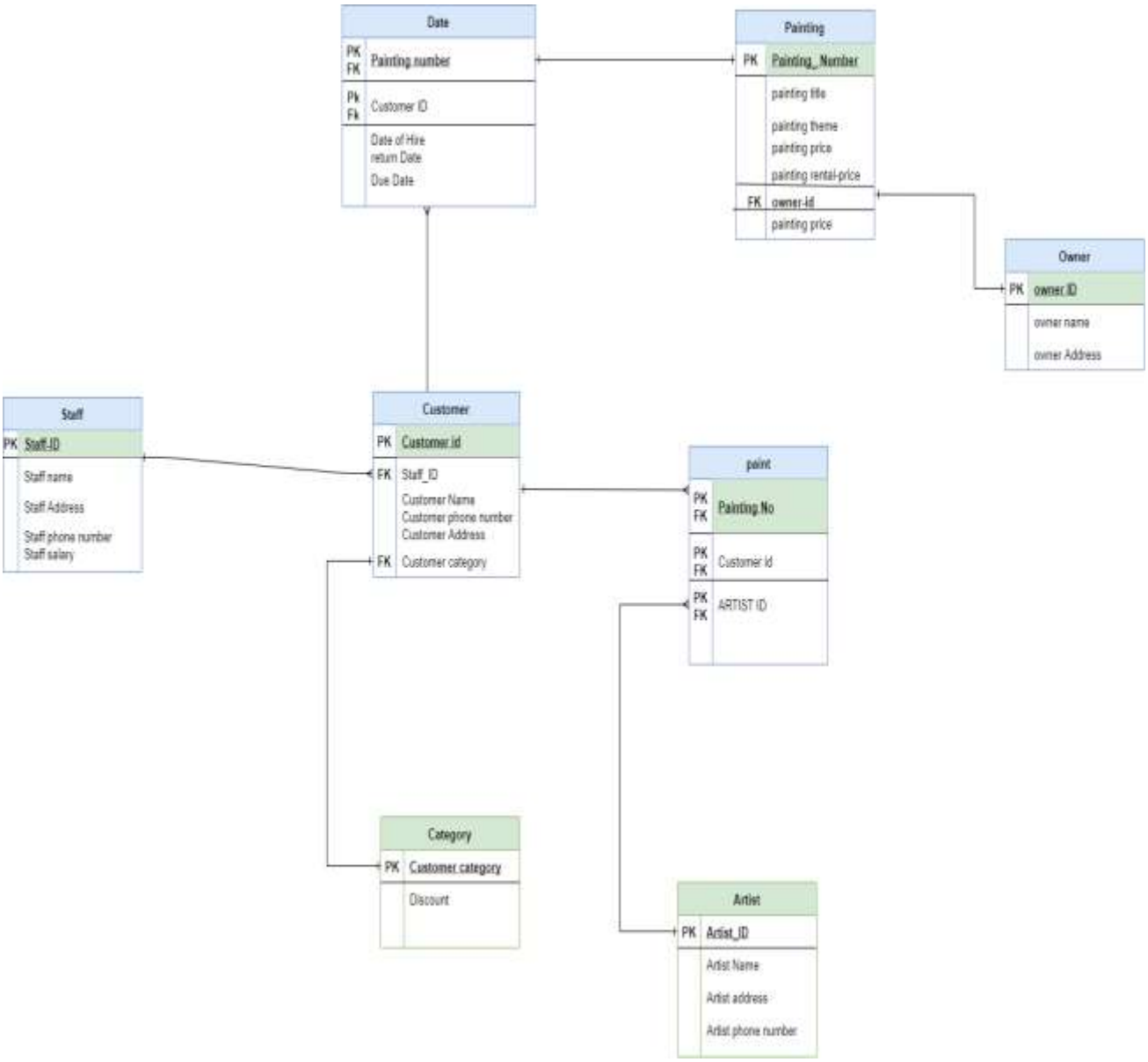

Figure 2: Final ER-Diagram after Normalization :

5.0 Implementation

5.1 Creation of table

Inside the Oracle database system, tables for every entity have been created. It is used in the Oracle 'Build TABLE statement to create a table. Each table was accessed, as well as each column was specified with the type of data for every column, in order to construct a table . The ALTER TABLE statement is being used to add, delete, or update columns in an existing table, as well as to add or remove various limits. During table creation, both primary and international keys are used. The primary key is a single field or a set of fields that contains a unique record. It must be filled. There will be no null values in any of the main key fields. There can only be one primary key in the table. A foreign key is a method of ensuring referential integrity in an Oracle database. The foreign key defines that the values in one table should also exist in another. The table that is being referenced is referred to as the parent table, whereas the table with the foreign key is referred to as the child table. In most cases, the foreign key in the child table corresponds to the main key in the parent table. To begin, we must login to the system and create a user database as well as a password for this project, as indicated in the figure below.

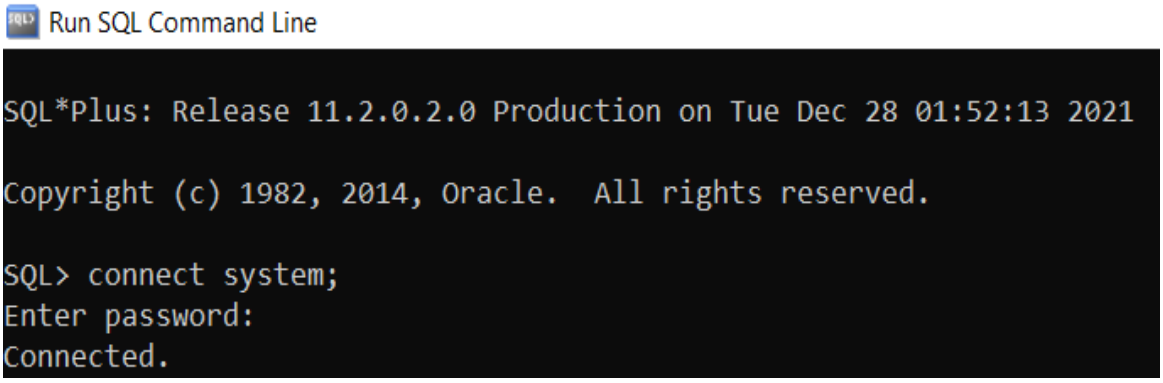

Figure 3: Connection of SQL/ CW

Creating staff Table:

```
create table staff(
        staff_id int,
        staff_Name varchar(30) not null,
        phone int,
        address varchar(30) not null,
         age int,
        salary int not null,
        CONSTRAINT staff_id_PK PRIMARY KEY(staff_id));
```

```
SQL> connect system;
Enter password:
Connected.
SQL> create table staff(staff_id int,
  2  staff_Name varchar(30) not null,
  3  phone int,
  4  address varchar(30) not null,
  5   age int,
  6  salary int not null,
  7  CONSTRAINT staff_id_PK PRIMARY KEY(staff_id));

Table created.
```

Figure 4: Creating Table staff

```
SQL> desc staff;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STAFF_ID                                  NOT NULL NUMBER(38)
 STAFF_NAME                                NOT NULL VARCHAR2(30)
 PHONE                                              NUMBER(38)
 ADDRESS                                   NOT NULL VARCHAR2(30)
 AGE                                                NUMBER(38)
 SALARY                                    NOT NULL NUMBER(38)
```

Figure 5: using describe command for table staff

In this table, staff_ID is primary key and consist of VARCHAR and NOT NULL datatype.

Creating artist Table:

```
create table artist(
        artist_id varchar(20),
        artist_Name varchar(30) not null,
        address varchar(30) not null,
        age int,
        CONSTRAINT artist_id_PK PRIMARY KEY(artist_id));
```

```
SQL>
SQL> create table artist(
  2  artist_id varchar(20),
  3  artist_Name varchar(30) not null,
  4  address varchar(30) not null,
  5  age int,
  6  CONSTRAINT artist_id_PK PRIMARY KEY(artist_id));

Table created.
```

Figure 6: Creating Table artist

```
SQL> desc artist;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ARTIST_ID                                 NOT NULL VARCHAR2(20)
 ARTIST_NAME                               NOT NULL VARCHAR2(30)
 ADDRESS                                   NOT NULL VARCHAR2(30)
 AGE                                                NUMBER(38)
```

Figure 7: using describe command for table artist

In this table, artist_ID is primary key and consist of VARCHAR and NOT NULL datatype.

Creating category Table:

```
create table category(
        customer_type varchar(30),
        discount int not null,
        CONSTRAINT customer_type_PK PRIMARY KEY(customer_type));
```

```
SQL> create table category(customer_type varchar(30), discount int not null,CONSTRAINT customer_type_PK
PRIMARY KEY(customer_type));

Table created.
```

Figure 8: Creating Table category

```
SQL> desc category;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_TYPE                             NOT NULL VARCHAR2(30)
 DISCOUNT                                  NOT NULL NUMBER(38)
```

Figure 9: using describe command for table category

In this table, customer_type is primary key having VARCHAR datatype.

Creating customer Table:

```
create table customer(
        customer_id int,
        customer_Name varchar(30) not null, phone int, address varchar(30),
        staff_id int not null, customer_Type varchar(30) not null,
        CONSTRAINT customer_id_PK PRIMARY KEY(customer_id),
        CONSTRAINT staff_id__FK FOREIGN KEY(staff_id) REFERENCES
        staff(staff_id),
        CONSTRAINT customer_type__FK FOREIGN KEY(customer_type)
        REFERENCES category(customer_type));
```

```
SQL> create table customer(customer_id int, customer_Name varchar(30) not null,phone int, address varcha
r(30), staff_id int not null, customer_Type varchar(30) not null,CONSTRAINT customer_id_PK PRIMARY KEY(c
ustomer_id),CONSTRAINT staff_id__FK FOREIGN KEY(staff_id) REFERENCES
  2  staff(staff_id),CONSTRAINT customer_type__FK FOREIGN KEY(customer_type) REFERENCES
  3  category(customer_type));

Table created.
```

Figure 10: Creating Table customer

```
SQL> desc customer;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                               NOT NULL NUMBER(38)
 CUSTOMER_NAME                             NOT NULL VARCHAR2(30)
 PHONE                                              NUMBER(38)
 ADDRESS                                            VARCHAR2(30)
 STAFF_ID                                  NOT NULL NUMBER(38)
 CUSTOMER_TYPE                             NOT NULL VARCHAR2(30)
```

Figure 11: using describe command for table customer

In this table, customer id is primary key and staff id as well as customer type is refered as foreign key.

Creating painting Table:

```
create table painting(
        painting_id int,
        painting_Name varchar(30) not null,
        painting_theme varchar(40) not null, available char(1),
        paid_price decimal(10,3), book_Date date not null,
        rental_price decimal(10,3),artist_id varchar(20) not null,
        CONSTRAINT painting_id_PK PRIMARY KEY(painting_id),
        CONSTRAINT artist_id__FK FOREIGN KEY(artist_id) REFERENCES
        artist(artist_id));
```

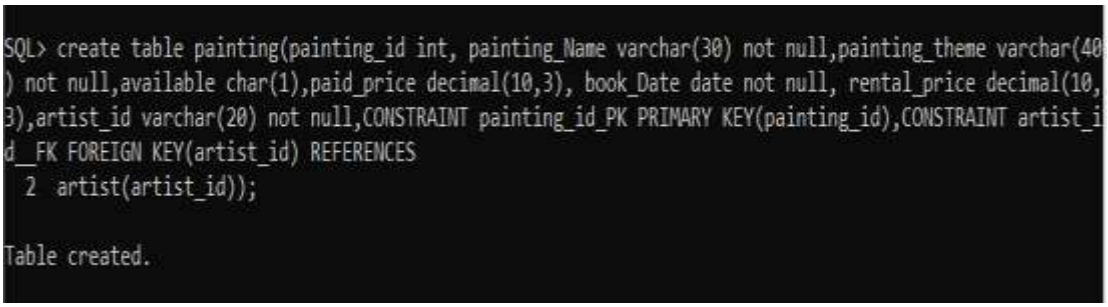

Figure 12: Creating Table painting

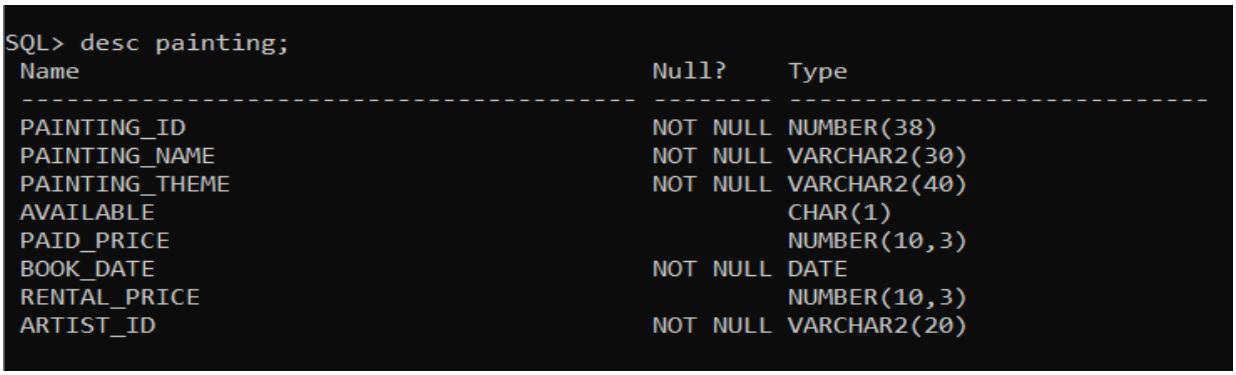

Figure 13: using describe command for table painting

In this table, painting id is primary key and artist id is refered as foreign key.

Creating Hire Table:

```
create table Hire(
        customer_id int,
        painting_id int, hire_Date date,
        return_Date date,
        CONSTRAINT customer__FK FOREIGN KEY(customer_id) REFERENCES
        customer(customer_id),
        CONSTRAINT painting__FK FOREIGN KEY(painting_id) REFERENCES
        painting(painting_id),
        CONSTRAINT lease_id_PK PRIMARY KEY(painting_id,customer_id));
```

```
SQL> create table Hire(customer_id int, painting_id int, hire_Date date, return_Date date,CONSTRAINT cus
tomer__FK FOREIGN KEY(customer_id) REFERENCES
  2  customer(customer_id),CONSTRAINT painting__FK FOREIGN KEY(painting_id) REFERENCES
  3  painting(painting_id),CONSTRAINT lease_id_PK PRIMARY KEY(painting_id,customer_id));

Table created.
```

Figure 14: Creating Table Hire

```
SQL> desc Hire;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                               NOT NULL NUMBER(38)
 PAINTING_ID                               NOT NULL NUMBER(38)
 HIRE_DATE                                          DATE
 RETURN_DATE                                        DATE
```

Figure 15: using describe command for table Hire

In this table, lease id is primary key and painting id as well as customer id is refered as foreign key.

Creating buy Table:

```
create table buy(
        customer_id int,
        painting_id int,
        painting_Priceco decimal(10,3),
        CONSTRAINT customer FOREIGN KEY(customer_id) REFERENCES
        customer(customer_id),
        CONSTRAINT painting FOREIGN KEY(painting_id) REFERENCES
        painting(painting_id),
        CONSTRAINT purchase_id_PK PRIMARY KEY(painting_id,customer_id));
```

```
SQL> create table buy(customer_id int, painting_id int, painting_Priceco decimal(10,3),CONSTRAINT custom
er FOREIGN KEY(customer_id) REFERENCES
  2  customer(customer_id),CONSTRAINT painting FOREIGN KEY(painting_id) REFERENCES
  3  painting(painting_id),CONSTRAINT purchase_id_PK PRIMARY KEY(painting_id,customer_id));

Table created.
```

Figure 16: Creating Table buy

```
SQL> desc buy;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                               NOT NULL NUMBER(38)
 PAINTING_ID                               NOT NULL NUMBER(38)
 PAINTING_PRICECO                                   NUMBER(10,3)
```

Figure 17: using describe command for table buy

In this table, purchase id is primary key and painting id as well as customer id is refered as foreign key.

5.2 Populating Data of table

The INSERT statement is used to insert information into a database. The INSERT statement is used to relate rows to a line, a view's base table, a division of a divided table or a sub-partition of a composite table, an object table, or a view's base table. To insert rows together into table, the table should be in our own scheme; otherwise, the table must have the INSERT object privilege. The COMMIT command is often used to finish our current transaction as well as making all changes made in the transaction permanent. A transaction is a collection of SQL statements that Oracle Database treats as a whole. This statement also removes every transaction savepoints and releases transaction locks (Abbott, 2021).

Inserting and displaying data in staff table:

```
insert all
  2  into staff values(1001,'Kishor Shrestha',9830378987,'Janakpur',33,13256)
  3  into staff values(1002,'Nischala Rai',9834251765,'Patan',35,54287)
  4  into staff values(1003,'Geeta Singh',9813276547,'Boudha',37,65321)
  5  into staff values(1004,'Kabir Singh',9845763210,'Kathmandu',38,76432)
  6  into staff  values(1005,'Comret thapa',9876532109,'Kathmandu',39,54329)
  7  into staff values(1006,'Virat kholi',9865123876,'Patan',41,65387)
  8  into staff values(1007,’Steve Smith’,9854732098,'Pokhara',44,98754)
  9  into staff values(1008,’Paras Khadka’,9898654098,'Boudha',46,09876)
 10  SELECT * FROM DUAL;
```

```
SQL> insert all
  2  into staff values(1001,'Kishor Shrestha',9830378987,'Janakpur',33,13256)
  3  into staff values(1002,'Nischala Rai',9834251765,'Patan',35,54287)
  4  into staff values(1003,'Geeta Singh',9813276547,'Boudha',37,65321)
  5  into staff values(1004,'Kabir Singh',9845763210,'Kathmandu',38,76432)
  6  into staff  values(1005,'Comret thapa',9876532109,'Kathmandu',39,54329)
  7  into staff values(1006,'Virat kholi',9865123876,'Patan',41,65387)
  8  into staff values(1007,'Steve Smith',9854732098,'Pokhara',44,98754)
  9  into staff values(1008,'Paras Khadka',9898654098,'Boudha',46,09876)
 10  SELECT * FROM DUAL;

8 rows created.
```

```
SQL> set linesize 150
SQL> set linesize 150
SQL> select * from staff;

  STAFF_ID STAFF_NAME                     PHONE ADDRESS                        AGE     SALARY
---------- ------------------------------ ---------- ------------------------------ ---------- ----------
      1001 Kishor Shrestha                9830378987 Janakpur                     33      13256
      1002 Nischala Rai                   9834251765 Patan                        35      54287
      1003 Geeta Singh                    9813276547 Boudha                       37      65321
      1004 Kabir Singh                    9845763210 Kathmandu                    38      76432
      1005 Comret thapa                   9876532109 Kathmandu                    39      54329
      1006 Virat kholi                    9865123876 Patan                        41      65387
      1007 Steve Smith                    9854732098 Pokhara                      44      98754
      1008 Paras Khadka                   9898654098 Boudha                       46       9876

8 rows selected.
```

Figure 18: inserting and displaying data of table staff

Inserting and displaying data in artist table:

```
insert all
  2  into artist values('K01','Irusa','Patan',10)
  3  into artist values('K02','Ashok','Patan' ,27)
  4  into artist values('K03','Aayush','Janakpur',66)
  5  into artist values('K04','Zenith','Pokhara',31)
  6  into artist values('K05','Hary','Pokhara',22)
  7  into artist values('K06','Williamson','Patan',29)
  8  into artist values('K07','Guptil','Ratnapark',25)
  9  into artist values('K08','Kane','Baneshwor',32)
 10  into artist values('K09','David','Kathmandu',33)
 11  into artist values('K10','Raj','Kathmandu',44)
 12  into artist values('K11','Karan','Palpa',83)
 13  into artist values('K12','Badal','Janakpur',40)
 14  SELECT * FROM DUAL;
```

```
SQL> insert all
  2  into artist values('K01','Irusa','Patan',10)
  3  into artist values('K02','Ashok','Patan' ,27)
  4  into artist values('K03','Aayush','Janakpur',66)
  5  into artist values('K04','Zenith','Pokhara',31)
  6  into artist values('K05','Hary','Pokhara',22)
  7  into artist values('K06','Williamson','Patan',29)
  8  into artist values('K07','Guptil','Ratnapark',25)
  9  into artist values('K08','Kane','Baneshwor',32)
 10  into artist values('K09','David','Kathmandu',33)
 11  into artist values('K10','Raj','Kathmandu',44)
 12  into artist values('K11','Karan','Palpa',83)
 13  into artist values('K12','Badal','Janakpur',40)
 14  SELECT * FROM DUAL;

12 rows created.
```

```
SQL> select * from artist;

ARTIST_ID            ARTIST_NAME                    ADDRESS                          AGE
-------------------- ------------------------------ ------------------------------ ----------
K01                  Irusa                          Patan                            10
K02                  Ashok                          Patan                            27
K03                  Aayush                         Janakpur                         66
K04                  Zenith                         Pokhara                          31
K05                  Hary                           Pokhara                          22
K06                  Williamson                     Patan                            29
K07                  Guptil                         Ratnapark                        25
K08                  Kane                           Baneshwor                        32
K09                  David                          Kathmandu                        33
K10                  Raj                            Kathmandu                        44
K11                  Karan                          Palpa                            83

ARTIST_ID            ARTIST_NAME                    ADDRESS                          AGE
-------------------- ------------------------------ ------------------------------ ----------
K12                  Badal                          Janakpur                         40

12 rows selected.
```

Figure 19: inserting and displaying data of table artist

Inserting and displaying data in category table:

```
insert all
  2  into category values('R',15)
  3  into category values('L',10)
  4  into category values('p',5)
  5  into category values('VIP',0)
  6  SELECT * FROM DUAL;
```

```
SQL> insert all
  2  into category values('R',15)
  3  into category values('L',10)
  4  into category values('p',5)
  5  into category values('VIP',0)
  6  SELECT * FROM DUAL;

4 rows created.
```

```
SQL> select * from category;

CUSTOMER_TYPE                    DISCOUNT
------------------------------ ----------
R                                      15
L                                      10
p                                       5
VIP                                     0
```

Figure 20: inserting and displaying data of table category

Inserting and displaying data in customer table:

```
insert all
  2  into customer values (1,'Boult',9865321098,'Ilam',1001,'L')
  3  into customer  values (2,'Babar',9875098765,'Rara',1002,'VIP')
  4  into customer  values (3,'Fakar',9823476542,'khotang',1008,'L')
  5  into customer  values (4,'Azham',9835276187,'bhojpur',1002,'R')
  6  into customer  values (5,'Shanwaj',9824376890,'Pokhara',1006,'R')
  7  into customer  values (6,'Amir',9845387623,'Mustang',1003,'p')
  8  into customer  values (7,'Sami',9834276541,'Jumla',1005,'p')
  9  into customer  values (8,'Jadega',9856498709,'Humla',1003,'L')
 10  SELECT * FROM DUAL;
```

```
SQL>
SQL> insert all
  2  into customer values (1,'Boult',9865321098,'Ilam',1001,'L')
  3  into customer  values (2,'Babar',9875098765,'Rara',1002,'VIP')
  4  into customer  values (3,'Fakar',9823476542,'khotang',1008,'L')
  5  into customer  values (4,'Azham',9835276187,'bhojpur',1002,'R')
  6  into customer  values (5,'Shanwaj',9824376890,'Pokhara',1006,'R')
  7  into customer  values (6,'Amir',9845387623,'Mustang',1003,'p')
  8  into customer  values (7,'Sami',9834276541,'Jumla',1005,'p')
  9  into customer  values (8,'Jadega',9856498709,'Humla',1003,'L')
 10  SELECT * FROM DUAL;

8 rows created.
```

```
SQL> select * from customer;

CUSTOMER_ID CUSTOMER_NAME                 PHONE ADDRESS                  STAFF_ID CUSTOMER_TYPE
----------- ----------------------------- --------- ---------------------------- ---------- ------------------------------
          1 Boult                         9865321098 Ilam                   1001 L
          2 Babar                         9875098765 Rara                   1002 VIP
          3 Fakar                         9823476542 khotang                1008 L
          4 Azham                         9835276187 bhojpur                1002 R
          5 Shanwaj                       9824376890 Pokhara                1006 R
          6 Amir                          9845387623 Mustang                1003 p
          7 Sami                          9834276541 Jumla                  1005 p
          8 Jadega                        9856498709 Humla                  1003 L

8 rows selected.
```

Figure 21: inserting and displaying data of table customer

Inserting and displaying data in painting table:

insert all

2 into painting values(101,'Sachin','Person','y',9995.33,To_Date('2021-12-01', 'YYYY-MM-DD'),2234.33,'K01')

3 into painting values(102,'Monkey','animal','y',7756.33,To_Date('2021-12-02', 'YYYY-MM-DD'),3322.43,'K02')

4 into painting values(103,'Donkey','animal', 'n',6590.44, To_Date('2021-12-05', 'YYYY-MM-DD'),4532.11,'K03')

5 into painting values(104,'Road','landscape','y',9970.01,To_Date('2021-12-14', 'YYYY-MM-DD'),4423.10,'K04')

6 into painting values(105,'Hills','landscape','y',7685.66,To_Date('2021-12-12', 'YYYY-MM-DD'),1276.44,'K05')

7 into painting values(106,'Zebra','animal','y',8675.77,To_Date('2021-12-11', 'YYYY-MM-DD'),2548.44,'K06')

8 into painting values(107,'Snake','animal','n',8799.66,To_Date('2021-12-19', 'YYYY-MM-DD'),4325.22,'K07')

9 into painting values(108,'Thor','Person','y',6788.77,To_Date('2021-12-04', 'YYYY-MM-DD'),4327.44,'K08')

10 into painting values(109,'Garden','landscape','y',9706.77,To_Date('2021-12-29', 'YYYY-MM-DD'),2315.33,'K09')

11 into painting values(110,'Rabit','animal','y',7866.77,To_Date('2021-12-28', 'YYYY-MM-DD'),3421.33,'K10')

12 into painting values(111,'Gayle','Person','y',6786.77,To_Date('2021-12-27', 'YYYY-MM-DD'),3422.33,'K11')

13 into painting values(112,'Butterfly','animal','y',7786.77,To_Date('2021-12-30', 'YYYY-MM-DD'),3421.33,'K12')

SELECT * FROM DUAL;

```
SQL> insert all
  2  into painting  values(101,'Sachin','Person','y',9995.33,To_Date('2021-12-01', 'YYYY-MM-DD'),2234.33,'K01')
  3  into painting  values(102,'Monkey','animal','y',7756.33,To_Date('2021-12-02', 'YYYY-MM-DD'),3322.43,'K02')
  4  into painting  values(103,'Donkey','animal', 'n',6590.44, To_Date('2021-12-05', 'YYYY-MM-DD'),4532.11,'K03')
  5  into painting  values(104,'Road','landscape','y',9970.01,To_Date('2021-12-14', 'YYYY-MM-DD'),4423.10,'K04')
  6  into painting  values(105,'Hills','landscape','y',7685.66,To_Date('2021-12-12', 'YYYY-MM-DD'),1276.44,'K05')
  7  into painting  values(106,'Zebra','animal','y',8675.77,To_Date('2021-12-11', 'YYYY-MM-DD'),2548.44,'K06')
  8  into painting  values(107,'Snake','animal','n',8799.66,To_Date('2021-12-19', 'YYYY-MM-DD'),4325.22,'K07')
  9  into painting  values(108,'Thor','Person','y',6788.77,To_Date('2021-12-04', 'YYYY-MM-DD'),4327.44,'K08')
 10  into painting  values(109,'Garden','landscape','y',9706.77,To_Date('2021-12-29', 'YYYY-MM-DD'),2315.33,'K09')
 11  into painting  values(110,'Rabit','animal','y',7866.77,To_Date('2021-12-28', 'YYYY-MM-DD'),3421.33,'K10')
 12  into painting  values(111,'Gayle','Person','y',6786.77,To_Date('2021-12-27', 'YYYY-MM-DD'),3422.33,'K11')
 13  into painting  values(112,'Butterfly','animal','y',7786.77,To_Date('2021-12-30', 'YYYY-MM-DD'),3421.33,'K12')
 14  SELECT * FROM DUAL;

12 rows created.
```

```
SQL> select * from painting;

PAINTING_ID PAINTING_NAME              PAINTING_THEME                         A PAID_PRICE BOOK_DATE RENTAL_PRICE ARTIST_ID
----------- ------------------------- -------------------------------------- - ---------- --------- ------------ ----------
        101 Sachin                    Person                                 y    9995.33 01-DEC-21      2234.33 K01
        102 Monkey                    animal                                 y    7756.33 02-DEC-21      3322.43 K02
        103 Donkey                    animal                                 n    6590.44 05-DEC-21      4532.11 K03
        104 Road                      landscape                              y    9970.01 14-DEC-21       4423.1 K04
        105 Hills                     landscape                              y    7685.66 12-DEC-21      1276.44 K05
        106 Zebra                     animal                                 y    8675.77 11-DEC-21      2548.44 K06
        107 Snake                     animal                                 n    8799.66 19-DEC-21      4325.22 K07
        108 Thor                      Person                                 y    6788.77 04-DEC-21      4327.44 K08
        109 Garden                    landscape                              y    9706.77 29-DEC-21      2315.33 K09
        110 Rabit                     animal                                 y    7866.77 28-DEC-21      3421.33 K10
        111 Gayle                     Person                                 y    6786.77 27-DEC-21      3422.33 K11

PAINTING_ID PAINTING_NAME              PAINTING_THEME                         A PAID_PRICE BOOK_DATE RENTAL_PRICE ARTIST_ID
----------- ------------------------- -------------------------------------- - ---------- --------- ------------ ----------
        112 Butterfly                 animal                                 y    7786.77 30-DEC-21      3421.33 K12

12 rows selected.
```

Figure 22: inserting and displaying data of table painting

Inserting and displaying data in Hire table:

```
insert all
2 into  hire values (1,109,To_Date('2021-11-11', 'YYYY-MM-DD'),To_Date('2021-12-11', 'YYYY-MM-DD'))
3 into  hire values (4,105,To_Date('2021-11-12', 'YYYY-MM-DD'),To_Date('2021-12-12', 'YYYY-MM-DD'))
4 into  hire values (2,111,To_Date('2021-11-13', 'YYYY-MM-DD'),To_Date('2021-12-13', 'YYYY-MM-DD'))
5 into  hire values (6,110,To_Date('2021-11-21', 'YYYY-MM-DD'),To_Date('2021-12-21', 'YYYY-MM-DD'))
6 into  hire values (3,107,To_Date('2021-11-20', 'YYYY-MM-DD'),To_Date('2021-12-20', 'YYYY-MM-DD'))
7 into  hire values (8,109,To_Date('2021-11-23', 'YYYY-MM-DD'),To_Date('2021-12-23', 'YYYY-MM-DD'))
8 into  hire values (5,102,To_Date('2021-11-22', 'YYYY-MM-DD'),To_Date('2021-12-22', 'YYYY-MM-DD'))
9 into  hire values (4,112,To_Date('2021-11-01', 'YYYY-MM-DD'),To_Date('2022-12-01', 'YYYY-MM-DD'))
SELECT * FROM DUAL;
```

```
SQL> insert all
  2  into  hire values (1,109,To_Date('2021-11-11', 'YYYY-MM-DD'),To_Date('2021-12-11', 'YYYY-MM-DD'))
  3  into  hire values (4,105,To_Date('2021-11-12', 'YYYY-MM-DD'),To_Date('2021-12-12', 'YYYY-MM-DD'))
  4  into  hire values (2,111,To_Date('2021-11-13', 'YYYY-MM-DD'),To_Date('2021-12-13', 'YYYY-MM-DD'))
  5  into  hire values (6,110,To_Date('2021-11-21', 'YYYY-MM-DD'),To_Date('2021-12-21', 'YYYY-MM-DD'))
  6  into  hire values (3,107,To_Date('2021-11-20', 'YYYY-MM-DD'),To_Date('2021-12-20', 'YYYY-MM-DD'))
  7  into  hire values (8,109,To_Date('2021-11-23', 'YYYY-MM-DD'),To_Date('2021-12-23', 'YYYY-MM-DD'))
  8  into  hire values (5,102,To_Date('2021-11-22', 'YYYY-MM-DD'),To_Date('2021-12-22', 'YYYY-MM-DD'))
  9  into  hire values (4,112,To_Date('2021-11-01', 'YYYY-MM-DD'),To_Date('2022-12-01', 'YYYY-MM-DD'))
 10  SELECT * FROM DUAL;

8 rows created.
```

```
SQL> select * from Hire;

CUSTOMER_ID PAINTING_ID HIRE_DATE RETURN_DA
----------- ---------- --------- ---------
          1        109 11-NOV-21 11-DEC-21
          4        105 12-NOV-21 12-DEC-21
          2        111 13-NOV-21 13-DEC-21
          6        110 21-NOV-21 21-DEC-21
          3        107 20-NOV-21 20-DEC-21
          8        109 23-NOV-21 23-DEC-21
          5        102 22-NOV-21 22-DEC-21
          4        112 01-NOV-21 01-DEC-22

8 rows selected.
```

Figure 23: inserting and displaying data of table Hire

Inserting and displaying data in buy table:

insert all

2 into buy values (1,105,4532.11)

3 into buy values (3,112,3251.11)

4 into buy values (2,104,2451.11)

5 into buy values (4,111,3645.11)

6 into buy values (8,105,3427.11)

7 into buy values (5,107,4352.11)

8 into buy values (2,102,2436.11)

9 into buy values (6,108,4532.11)

10 into buy values (2,105,2435.11)

11 into buy values (7,109,2324.11)

SELECT * FROM DUAL;

```
SQL> insert all
  2  into buy values (1,105,4532.11)
  3  into buy values (3,112,3251.11)
  4  into buy values (2,104,2451.11)
  5  into buy values (4,111,3645.11)
  6  into buy values (8,105,3427.11)
  7  into buy values (5,107,4352.11)
  8  into buy values (2,102,2436.11)
  9  into buy values (6,108,4532.11)
 10  into buy values (2,105,2435.11)
 11  into buy values (7,109,2324.11)
 12  SELECT * FROM DUAL;

10 rows created.
```

```
SQL> select * from buy;

CUSTOMER_ID PAINTING_ID PAINTING_PRICECO
----------- ----------- ----------------
          1         105          4532.11
          3         112          3251.11
          2         104          2451.11
          4         111          3645.11
          8         105          3427.11
          5         107          4352.11
          2         102          2436.11
          6         108          4532.11
          2         105          2435.11
          7         109          2324.11

10 rows selected.
```

Figure 24: inserting and displaying data of table buy

6.0 Database Queries

6.1 Information query

1. List all customers according to category

SELECT * FROM Customer ORDER BY customer_type;

```
SQL> SELECT * FROM customer
  2   ORDER BY CUSTOMER_TYPE;

CUSTOMER_ID CUSTOMER_NAME                    PHONE ADDRESS                        STAFF_ID CUSTOMER_TYPE
----------- ------------------------------ ---------- ------------------------------ ---------- ------------------------------
          1 Boult                          9865321098 Ilam                             1001 L
          8 Jadega                         9856498709 Humla                            1003 L
          3 Fakar                          9823476542 khotang                          1008 L
          4 Azham                          9835276187 bhojpur                          1002 R
          5 Shanwaj                        9824376890 Pokhara                          1006 R
          2 Babar                          9875098765 Rara                             1002 VIP
          6 Amir                           9845387623 Mustang                          1003 p
          7 Sami                           9834276541 Jumla                            1005 p

8 rows selected.
```

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

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

```
SQL> SELECT Staff_Name, Salary from Staff ORDER By Salary DESC;

STAFF_NAME                     SALARY
------------------------------ ----------
Steve Smith                     98754
Kabir Singh                     76432
Virat kholi                     65387
Geeta Singh                     65321
Comret thapa                    54329
Nischala Rai                    54287
Kishor Shrestha                 13256
Paras Khadka                     9876

8 rows selected.
```

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

5. List all paintings that have been returned to the owner.

```
SQL> SELECT * FROM PAINTING where available = 'y';

PAINTING_ID PAINTING_NAME                  PAINTING_THEME                           A PAID_PRICE BOOK_DATE RENTAL_PRICE ARTIST_ID
----------- ------------------------------ ---------------------------------------- - ---------- --------- ------------ --------------------
        101 Sachin                         Person                                   y    9995.33 01-DEC-21      2234.33 K01
        102 Monkey                         animal                                   y    7756.33 02-DEC-21      3322.43 K02
        104 Road                           landscape                                y    9970.01 14-DEC-21       4423.1 K04
        105 Hills                          landscape                                y    7685.66 12-DEC-21      1276.44 K05
        106 Zebra                          animal                                   y    8675.77 11-DEC-21      2548.44 K06
        108 Thor                           Person                                   y    6788.77 04-DEC-21      4327.44 K08
        109 Garden                         landscape                                y    9706.77 29-DEC-21      2315.33 K09
        110 Rabit                          animal                                   y    7866.77 28-DEC-21      3421.33 K10
        111 Gayle                          Person                                   y    6786.77 27-DEC-21      3422.33 K11
        112 Butterfly                      animal                                   y    7786.77 30-DEC-21      3421.33 K12

10 rows selected.
```

6.2 Transaction query

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

```
SQL> SELECT * from painting
  2    WHERE available = 'y'
  3     ORDER BY painting_theme;

PAINTING_ID PAINTING_NAME                  PAINTING_THEME                           A PAID_PRICE BOOK_DATE RENTAL_PRICE ARTIST_ID
----------- ------------------------------ ---------------------------------------- - ---------- --------- ------------ --------------------
        101 Sachin                         Person                                   y    9995.33 01-DEC-21      2234.33 K01
        108 Thor                           Person                                   y    6788.77 04-DEC-21      4327.44 K08
        111 Gayle                          Person                                   y    6786.77 27-DEC-21      3422.33 K11
        112 Butterfly                      animal                                   y    7786.77 30-DEC-21      3421.33 K12
        102 Monkey                         animal                                   y    7756.33 02-DEC-21      3322.43 K02
        106 Zebra                          animal                                   y    8675.77 11-DEC-21      2548.44 K06
        110 Rabit                          animal                                   y    7866.77 28-DEC-21      3421.33 K10
        104 Road                           landscape                                y    9970.01 14-DEC-21       4423.1 K04
        109 Garden                         landscape                                y    9706.77 29-DEC-21      2315.33 K09
        105 Hills                          landscape                                y    7685.66 12-DEC-21      1276.44 K05

10 rows selected.
```

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

```
SQL> SELECT * FROM painting WHERE MONTHS_BETWEEN(sysdate,book_date)<3 AND available = 'y';

PAINTING_ID PAINTING_NAME                    PAINTING_THEME                                A PAID_PRICE BOOK_DATE RENTAL_PRICE ARTIST_ID
----------- ------------------------------   --------------------------------------------- - ---------- --------- ------------ ----------
        101 Sachin                           Person                                        y    9995.33 01-DEC-21      2234.33 K01
        102 Monkey                           animal                                        y    7756.33 02-DEC-21      3322.43 K02
        104 Road                             landscape                                     y    9970.01 14-DEC-21       4423.1 K04
        105 Hills                            landscape                                     y    7685.66 12-DEC-21      1276.44 K05
        106 Zebra                            animal                                        y    8675.77 11-DEC-21      2548.44 K06
        108 Thor                             Person                                        y    6788.77 04-DEC-21      4327.44 K08
        109 Garden                           landscape                                     y    9706.77 29-DEC-21      2315.33 K09
        110 Rabit                            animal                                        y    7866.77 28-DEC-21      3421.33 K10
        111 Gayle                            Person                                        y    6786.77 27-DEC-21      3422.33 K11
        112 Butterfly                        animal                                        y    7786.77 30-DEC-21      3421.33 K12

10 rows selected.
```

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

4. List top 5 paintings based on total collected rental amount.

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

❖ Drop Table

```
SQL> drop table buy;

Table dropped.

SQL> drop table Hire;

Table dropped.

SQL> drop table painting;;
drop table painting;
                  *
ERROR at line 1:
ORA-00911: invalid character


SQL> drop table painting;

Table dropped.

SQL> drop table customer;

Table dropped.

SQL> drop table category;

Table dropped.

SQL> drop table artist;

Table dropped.

SQL> drop table staff;

Table dropped.

SQL> _
```

7.0 Conclusion

I've discovered that now the period of time I focus on Database coursework is improving at a considerable pace day after day, which is excellent for a learner. However, there is still a considerable distance to go until the course's work is completed. I may have grown tired of a variety of activities and SQL syntax studies at times. Even though the SQL syntax is challenging to understand at first, I'm working my way through it with more Oracle practices and usage. Because Oracle has one of our current course's massive SQL syntax, the module teachers are assisting me greatly while doing assignments, and I have also been capable of going deeper into study courses such as analyzing case scenarios, trying to define entities and attributes, assigning relationships when creating ER diagrams, generating 3NF result sets, Data Insertion, as well as creating SQL statements for the provided queries. Now, after a great deal of experience, I'm more hesitant to SQL Syntax.

This coursework has assisted us in improving our performance for a better outcome of our work. Similarly, this coursework was difficult, although I completed it on time. The initial stage in developing a database is to describe the entities and attributes which are formed as a result of understanding the scenario. Following that, the connection between the properties was examined, as well as an ER diagram was created. I created tthe ER diagram was used to normalize the database's unnormalized shape. Afterwards when, I tweaked the UNF to 1NF, 2NF, and 3NF, in that order. The project revolves entirely around the database management system procedure and is completed (DBMS). Oracle is used to complete the project. 11g database software (run sql command line). In this project, data values are saved in accordance with which is provided in the question In this project, we create a first ERD based on the data requirements.

While Covering the College Record System, as well as displaying all of the entities, respective attributes, and its relationships. Following the initial ERD, we establish our own structure and proceed with the normalization procedure to avoid receiving any sorts of redundant (repetitive) data.

I'd want to thanks my professors for assisting me even during this challenging era and for devoting their time to my benefit. This course provided a lot of information regarding data management using the Oracle database system. To accomplish this job, I had to watch the lecture and tutorial videos several times. This course improved my DBMS skills and experience.

The purpose of this course was to create a database management system. The first stage was to identify the entities and attributes, and then a first Entity Relationship diagram was created by analyzing the relationships between the attributes. I was unfamiliar with data administration through Oracle, however this coursework assisted me in learning how to use Oracle. I'm feeling a little more confidence now that I've successfully been capable of managing data for a large corporation using DBMS, as well as I believe I'll be capable of handling more projects like this in the future.

8.0 References

Abbott, K. (2021, 12 20). *oracle.com*. Retrieved from oracle: https://docs.oracle.com/cd/B19306_01/appdev.102/b14261/commit_statement.htm

kalin_martin. (2021, 12 20). *condor.depaul.edu*. Retrieved from condor.depaul: https://condor.depaul.edu/gandrus/240IT/accesspages/primary-foreign-keys.htm

Kant, K. (2021, 12 25). *techopedia.com*. Retrieved from techopedia: https://www.techopedia.com/definition/1221/normalization

Millington, D. (2021, 12 20). *embarcadero.com*. Retrieved from embarcadero: https://docwiki.embarcadero.com/InterBase/2020/en/Identifying_Entities_and_Attributes

opengrass.net. (2021, 12 20). Retrieved from opengrass: http://rdbms.opengrass.net/2_Database%20Design/2.2_Normalisation/r/UNF%20Create%20Model%20Data.pdf