



**slington college**  
(इस्लिङ्टन कलेज)

**Module Code & Module Title**

**CC5051NI Databases**

**Assessment Weightage & Type**

**50% Individual Coursework**

**Year and Semester**

**2021 Autumn**

**Student Name: Rabina Shrestha**

**London Met ID: 20049416**

**College ID: NP01CP4S210039**

**Assignment Due Date: 28<sup>th</sup> December, 2021.**

**Assignment Submission Date: 28<sup>th</sup> December, 2021.**

*I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.*

## Contents

|                                                              |    |
|--------------------------------------------------------------|----|
| 1. Introduction. ....                                        | 1  |
| 1.1. Introduction to the Internet Business.....              | 2  |
| 1.2. Aims and Objectives. ....                               | 3  |
| 1.3. Current Business Activities and Operations. ....        | 4  |
| 1.4. Business Rules. ....                                    | 5  |
| 1.5. Identification of Entities and Attributes. ....         | 6  |
| 2. Initial ERD. ....                                         | 13 |
| 2.1. List of Initially Created Entities and Attributes. .... | 14 |
| 2.2. Entity Relation Diagram. ....                           | 15 |
| 3. Normalization. ....                                       | 16 |
| 3.1. UNF (Unnormalized Form).....                            | 16 |
| 3.2. 1NF (First Normal Form). ....                           | 17 |
| 3.3. 2NF (Second Normal Form).....                           | 18 |
| 3.4. 3NF (Third Normal Form).....                            | 20 |
| 4. Final ERD.....                                            | 22 |
| 5. Implementation.....                                       | 24 |
| 5.1. Table Generation ....                                   | 24 |
| 5.2. Populating the Database Tables. ....                    | 29 |
| 5.3. Displaying the Table.....                               | 38 |
| 5.4. Dropping the Tables.....                                | 41 |
| 6. Database Querying. ....                                   | 43 |
| 6.1. Information Query ....                                  | 43 |

|        |                                                                                                        |    |
|--------|--------------------------------------------------------------------------------------------------------|----|
| 6.1.1. | List all customers according to category:.....                                                         | 43 |
| 6.1.2. | List paintings and their artist with monthly rental price and paid price: .....                        | 44 |
| 6.1.3. | Show total staff in Masterpieces Limited sorted by higher salary: .....                                | 45 |
| 6.1.4. | Show paintings leased before and currently by any one customer:.....                                   | 45 |
| 6.1.5. | List all paintings that have been returned to the owner: .....                                         | 46 |
| 6.2.   | Transaction Query.....                                                                                 | 47 |
| 6.2.1. | List the number of paintings available for rent according to category: .....                           | 47 |
| 6.2.2. | List the details of paintings that have not been leased within three months:<br>47                     |    |
| 6.2.3. | List the details of customers who have leased the painting more than four<br>times: 48                 |    |
| 6.2.4. | List top 5 paintings based on total collected rental amount: .....                                     | 49 |
| 6.2.5. | Show the name of the painter and their paintings sold value (in total) for the<br>current month: ..... | 49 |
| 7.     | Critical Evaluation. ....                                                                              | 50 |
| 7.1.   | Critical Evaluation of the Module.....                                                                 | 50 |
| 7.2.   | Critical Assessment of the Coursework.....                                                             | 51 |
| 8.     | Dump File Creation. ....                                                                               | 52 |
| 9.     | Bibliography .....                                                                                     | 53 |

## List of Figures

|                                                                         |    |
|-------------------------------------------------------------------------|----|
| Figure 1: The Picture Room: Sir John Soane's Museum (Soane, 1837) ..... | 2  |
| Figure 2: Initial Entities and Attributes. ....                         | 14 |
| Figure 3: Initial ERD. ....                                             | 15 |
| Figure 4: Final ERD. ....                                               | 22 |
| Figure 5: Creating the User and Granting Permissions. ....              | 24 |
| Figure 6: Creating Table: Artist. ....                                  | 25 |
| Figure 7: Creating Table: Staff. ....                                   | 26 |
| Figure 8: Creating Table: Customer. ....                                | 26 |
| Figure 9: Creating Table: Painting.....                                 | 27 |
| Figure 10: Creating Table: Orders.....                                  | 28 |
| Figure 11: Creating Table: Order_Painting.....                          | 29 |
| Figure 12: Insert into Artist Values. ....                              | 30 |
| Figure 13: Insert into Staff Values. ....                               | 31 |
| Figure 14: Insert into Customer Values. ....                            | 32 |
| Figure 15: Insert into Painting Values.....                             | 33 |
| Figure 16: Insert into Orders Values.....                               | 35 |
| Figure 17: Insert into Order_Painting Values, .....                     | 36 |
| Figure 18: Display Artist. ....                                         | 38 |
| Figure 19: Display Staff. ....                                          | 38 |

|                                                                                                         |    |
|---------------------------------------------------------------------------------------------------------|----|
| Figure 20: Display Customer.....                                                                        | 39 |
| Figure 21: Display Painting. ....                                                                       | 39 |
| Figure 22: Display Order. ....                                                                          | 40 |
| Figure 23: Display Order_Painting. ....                                                                 | 40 |
| Figure 24: Dropping Table Order_Painting.....                                                           | 41 |
| Figure 25: Dropping Table Orders.....                                                                   | 41 |
| Figure 26: Dropping Table Painting.....                                                                 | 41 |
| Figure 27: Dropping Table Customer. ....                                                                | 42 |
| Figure 28: Dropping Table Staff. ....                                                                   | 42 |
| Figure 29: Dropping Table Artist. ....                                                                  | 42 |
| Figure 30: To list all customers according to the category. ....                                        | 43 |
| Figure 31: To list paintings and artists with monthly rental and paid price.....                        | 44 |
| Figure 32: To show total staffs sorted by higher salary. ....                                           | 45 |
| Figure 33: To show paintings leased before and currently by any one customer.....                       | 45 |
| Figure 34: To list all paintings that have been returned to the owner. ....                             | 46 |
| Figure 35: To list the number of paintings available for rent according to category.....                | 47 |
| Figure 36: To list the details of paintings that have not been leased within three months.<br>.....     | 47 |
| Figure 37: To list the details of customers who have leased the paintings more than four<br>times. .... | 48 |
| Figure 38: To list the top 5 paintings based on total collected rental amount. ....                     | 49 |

Figure 39: To show the name of painter and paintings sold value for the current month.  
..... 49

Figure 40: Dump File Creation. .... 52

## List of Tables

|                                               |    |
|-----------------------------------------------|----|
| Table 1: Data Dictionary: Artist.....         | 7  |
| Table 2: Data Dictionary: Staff. ....         | 8  |
| Table 3: Data Dictionary: Customer. ....      | 9  |
| Table 4: Data Dictionary: Painting.....       | 10 |
| Table 5: Data Dictionary: Orders.....         | 11 |
| Table 6: Data Dictionary: Order_Painting..... | 12 |

## 1. Introduction.

One of the modules that the Computing Students' study in Year 2 is "Databases". Being a semester long module, it consists of two components; Coursework and Unseen Examination, each carrying 50% of the overall module. The coursework was assigned on the 4<sup>th</sup> Week. Based on the instructions given, the task was to create a database design and implement it for an organization named "Masterpieces Limited".

The given task was completed with the help of a software named Oracle SQL PLUS. The specified software permits the users to login and connect to the database. Oracle is best recognised for its database software. It is one of the most reliable and widely used database. (Oracle, 2021)

A database is a structured set of records or data, usually stored in a computer so that it can be retrieved electronically. The database accommodates information about a distinct firm. It preserves any information that could be useful in the decision-making process. The data is recorded and arranged to provide a footing for future application development. (Berrington, 2017)

A good database system is prominent to the firm so that the data is secured from damage. It should be kept in a logical manner so that other applications can utilize the data.



## 1.1. Introduction to the Internet Business.

A local businessman and real estate owner, named Mr. Steve, has considered to branch out into online painting platform. He intends to lease artworks to individuals, companies, and various artists after licensing it as an Internet Business.



*Figure 1: The Picture Room: Sir John Soane's Museum (Soane, 1837)*

Many companies are impotent to deal with new restrictions and shutdowns caused by Covid-19 and eventually altered the ways of operation to cope up with the changes. Although there may be some negative effects to this shift of online business as not all businesses are reluctant to such major changes but it is very crucial to be able to adjust so as to survive in abnormal situations. During the present crisis the world is revolving around the pandemic so, working from home / online is the bare minimum that can be done and, in a way, it is stabilizing the world economy. Hence, the best possible way to resume the inflow of cash into the business is to turn into an online organization. (Subramaniam, et al., 2021)

Since the demand for an online system has flourished, the owner is seeking for consultation and design firms and wants to be able to run it online in case, it cannot be maintained physically. For an internet business to prosper, it requires a well-structured database for which, the situation is modified to obtain an entity-relationship model, formal design techniques are conducted to acquire a database structure. A good database system is vital to any firm and as a database designer, it is an honour to assist Mr. Steve in designing and implementing a database to enable Masterpieces Limited thrive.

## 1.2. Aims and Objectives.

The aim of Masterpiece Limited is to construct an efficient, transparent, and trustworthy digital portal for artists to showcase and advertise their work to potential buyers. (Artwork Archive, 2010)

### Objectives:

- To develop Masterpieces Limited as an internet business, allowing consumers to rent or buy painting they like from an online platform.
- To create a fear-free environment for artists, customers, and the workers.
- To prevent a large crowd while still exhibiting the art.
- To increase the artist's exposure while generating additional revenue.
- To represent tens of thousands of artists through one single platform. (Artwork Archive, 2010)
- To make it easier for customers to know what is available so that they can buy or rent their choices with a single click.
- Post high-quality and accurate images of the art so that potential buyers can evaluate the art pieces. (Phillips, 2020)

### **1.3. Current Business Activities and Operations.**

Masterpiece Limited, London, which is currently attempting to adapt into the current workflow as an Internet business, had experienced phenomenal development since its foundation in 1997.

However, due to Covid-19, the museum was forced to temporarily close, putting the health and safety of artists, colleagues, possible buyers, staff, and visitors first. While waiting for the situation to improve, it experienced a significant setback. Masterpiece Limited's owner, Mr. Steve, came up with the concept of turning it into an online business in order to survive in these difficult circumstances, such as lockdown, while still trying to pay staff salaries without any income. Following consultations with the shareholders, artists, and staff, they agreed to do so and proceeded on a quest to find a good database designer in order to develop and implement a database.

As of now, having a good reputation in the industry beforehand was beneficial as there were a lot of devoted customers.

Despite the huge setback, the owner decided that, while looking for a good database designer, it would be advantageous to operate through phone calls and bookings until a database could be created. The employees are in charge of making reservations in such a way that there is no overcrowding and risks are kept to minimum.

There is a lot of redundancy to be addressed as even the records exist in a physical format. Security has been increased to ensure that only those who have made reservations for a specific day and time are permitted to enter. In order to please consumers even in this tricky time, the owner has chosen to double the discount based on the client category. And now, week-long shows are hosted as the number of people allowed are limited. This shows highlight and advertises the art of great artists for the entire week encouraging people to join at a safer side.

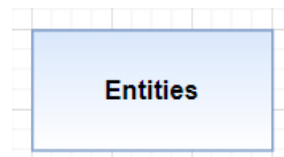
### 1.4. Business Rules.

- Any painting can be acquired or leased through the company.
- A piece of art can be leased for a fixed interval time i.e., one month.
- Only one artist should be affiliated with each piece of art.
- Clients are classified and granted discounts based on their classification i.e., If Regular, Loyal, Privileged, and VIP will get 0%, 5%, 10%, and 15% respectively.
- The same piece of art can be rented numerous times.
- A client may specify a certain artist or a certain theme.
- Artists have the option of renting their own work of art.
- The artist of each artwork charges a monthly fee for each piece. After which, the owner will receive a 20% reduction on the stated amount.
- If a piece of art is not rented throughout the four months, it is handed back to the artist. Nevertheless, an artist may resubmit a returned artwork one month after it had been returned.

### Assumptions:

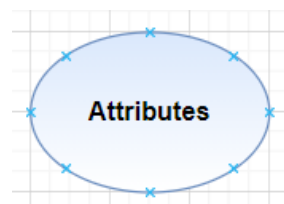
- A customer can lease or buy many items in a single order.
- A painting can only have one artist associated with it.
- One artist may create numerous paintings.
- One staff can issue many orders.
- An order can only be handled by one staff.
- One order may have multiple paintings order.

## 1.5. Identification of Entities and Attributes.



**Entities:** An entity is a thing, item, or data component that can be distinguished from other items in the actual world. It could be something real, like a person, or something abstract, like a thought.

It is the most fundamental unit of objects that can have a physical presence or form concepts or ideas. It is an easily identifiable and distinguishable animate or inanimate real-world object. It has an independent presence and is described by a set of attributes in a database whereas a rectangle represents it in an ER Diagram. (Elmasri & Navathe, 2016)



**Attributes:** A term that describes the property of an entity with which it is associated is called an attribute. The value of it is a unique instance. It is possible to classify attributes as descriptors or identifiers. Identifiers, often called keys, are used to identify a

particular instance of an entity. A descriptor is a phrase that describes a feature of an entity instance that isn't unique. An oval represents an attribute in an ER diagram. (Elmasri & Navathe, 2016)

### Keys:

**Primary Key:** This constraint written as PK ensures that no two rows will have the same value in the assigned column. It does not allow NULL values.

**Foreign Key:** This constraint written as FK identifies relationships between two or more tables by referencing columns or set of columns, in the child table that contains foreign key to the primary key columns in the parent table. Both needs to have the same data type.

**Unique Key:** This constraint written as UK consists of one or more table fields/columns that will be used to identify a record in a database table. (Sharma, 2020)

**Final Entities Data Dictionary.**

The final entities of the Masterpieces Limited are Artist, Staff, Customer, Painting, Order, and Order\_Painting making a total of six entities.

**Artist:**

One of the final entities present in Masterpieces Limited is Artist; it contains information about a person who creates a piece of art either as a career or hobby.

| <b>Entity: Artist</b> |                  |                                                                                     |            |
|-----------------------|------------------|-------------------------------------------------------------------------------------|------------|
| <b>Attributes</b>     | <b>Data Type</b> | <b>Description</b>                                                                  | <b>Key</b> |
| <b>Artist_ID</b>      | VARCHAR (5)      | Identity given to the artist consisting of unique numbers generated by the company. | PK         |
| <b>Artist_Name</b>    | VARCHAR (30)     | The name of the artist.                                                             |            |
| <b>Artist_Email</b>   | VARCHAR (50)     | Email address given to the artist.                                                  | UK         |

*Table 1: Data Dictionary: Artist.*

**Staff:**

Staff is another final entity present in Masterpieces Limited; it consists of information regarding the people who work in the organization.

| <b>Entity: Staff</b>   |                  |                                                                                    |            |
|------------------------|------------------|------------------------------------------------------------------------------------|------------|
| <b>Attributes</b>      | <b>Data Type</b> | <b>Description</b>                                                                 | <b>Key</b> |
| <b>Staff_ID</b>        | VARCHAR (5)      | Identity given to the staff consisting of unique numbers generated by the company. | PK         |
| <b>Staff_Name</b>      | VARCHAR (30)     | The name of the staff.                                                             |            |
| <b>Staff_Salary</b>    | NUMBER (10)      | Salary of the staff.                                                               |            |
| <b>Staff_ContactNo</b> | VARCHAR (10)     | Contains phone number of the staff.                                                | UK         |

*Table 2: Data Dictionary: Staff.*

**Customer:**

Customer is another final entity present in Masterpieces Limited; it contains of information regarding the people who buys or rents artworks from the business.

| <b>Entity: Customer</b>   |                  |                                                                                       |            |
|---------------------------|------------------|---------------------------------------------------------------------------------------|------------|
| <b>Attributes</b>         | <b>Data Type</b> | <b>Description</b>                                                                    | <b>Key</b> |
| <b>Customer_ID</b>        | VARCHAR (5)      | Identity given to the customer consisting of unique numbers generated by the company. | PK         |
| <b>Customer_Name</b>      | VARCHAR (30)     | The name of the customer.                                                             |            |
| <b>Customer_Address</b>   | VARCHAR (30)     | The address of the customer                                                           |            |
| <b>Customer_ContactNo</b> | VARCHAR (10)     | Contains phone number of the staff.                                                   |            |
| <b>Customer_Type</b>      | VARCHAR (15)     | Defines the type of customer; Individual or Commercial Company.                       |            |
| <b>Customer_Category</b>  | VARCHAR (10)     | The category the customer belongs to; Regular, Loyal, Privileged, VIP                 |            |

*Table 3: Data Dictionary: Customer.*



**Painting:**

Painting is another final entity present in Masterpieces Limited; it contains information about the piece of art.

| <b>Entity: Painting</b> |                  |                                                                                              |            |
|-------------------------|------------------|----------------------------------------------------------------------------------------------|------------|
| <b>Attributes</b>       | <b>Data Type</b> | <b>Description</b>                                                                           | <b>Key</b> |
| <b>Painting_ID</b>      | VARCHAR (5)      | Distinctiveness given to the painting consisting of unique numbers generated by the company. | PK         |
| <b>Artist_ID</b>        | VARCHAR (5)      | Identity given to the artist consisting of unique numbers generated by the company.          | FK         |
| <b>Painting_Name</b>    | VARCHAR (30)     | The name of the painting.                                                                    |            |
| <b>Issue_Date</b>       | DATE             | Contains the painting's issued date.                                                         |            |
| <b>Availability</b>     | VARCHAR (5)      | Says if the painting is available, sold, or returned to the owner.                           |            |
| <b>Painting_Theme</b>   | VARCHAR (10)     | Contains the theme of the painting.                                                          |            |
| <b>Monthly_Rental</b>   | NUMBER (15)      | Contains the monthly rent amount.                                                            |            |
| <b>Total_Rental</b>     | NUMBER (15)      | Contains the total monthly rental amount.                                                    |            |
| <b>Selling_Price</b>    | NUMBER (15)      | Contains the selling price.                                                                  |            |

*Table 4: Data Dictionary: Painting.*

**Orders:**

Orders is another final entity present in Masterpieces Limited; it contains information about an order.

| <b>Entity: Orders</b> |                  |                                                                                           |            |
|-----------------------|------------------|-------------------------------------------------------------------------------------------|------------|
| <b>Attributes</b>     | <b>Data Type</b> | <b>Description</b>                                                                        | <b>Key</b> |
| <b>Order_ID</b>       | VARCHAR (5)      | Distinctiveness given to the order consisting of unique numbers generated by the company. | PK         |
| <b>Customer_ID</b>    | VARCHAR (5)      | Identity given to the customer consisting of unique numbers generated by the company.     | FK         |
| <b>Staff_ID</b>       | VARCHAR (5)      | Identity given to the staff consisting of unique numbers generated by the company.        | FK         |
| <b>Order_Date</b>     | DATE             | The day customer ordered an artwork.                                                      |            |
| <b>Total_Amount</b>   | NUMBER (15)      | Contains the total price of an artwork.                                                   |            |

*Table 5: Data Dictionary: Orders.*

**Order\_Painting:**

Order\_Painting is last final entity present in Masterpieces Limited; it contains information about an order, painting and the reduced amount.

| <b>Entity: Order_Painting</b> |                  |                                                                                              |            |
|-------------------------------|------------------|----------------------------------------------------------------------------------------------|------------|
| <b>Attributes</b>             | <b>Data Type</b> | <b>Description</b>                                                                           | <b>Key</b> |
| <b>Order_ID</b>               | VARCHAR (5)      | Distinctiveness given to the order consisting of unique numbers generated by the company.    | PK, FK     |
| <b>Painting_ID</b>            | VARCHAR (5)      | Distinctiveness given to the painting consisting of unique numbers generated by the company. | PK, FK     |
| <b>Buy_Or_Lease</b>           | VARCHAR (7)      | Indicates if the customer is buying or leasing a product.                                    |            |
| <b>Return_Date</b>            | DATE             | The date when an artwork needs to get returned by.                                           |            |
| <b>Sold_Value</b>             | NUMBER (15)      | Contains the discounted amount of each painting.                                             |            |

*Table 6: Data Dictionary: Order\_Painting.*

## 2. Initial ERD.

An Entity–Relationship model (ER model) illustrates the whole logical structure of a database using a diagram known as an Entity Relationship Diagram (ER Diagram). The diagram represents a database design or blueprint that can eventually be turned into a database. It shows how entity sets are linked together. A table or a property of a table in a database that illustrates the link between tables and their attributes is referred to as an entity in a relational database management system (DBMS). (Singh, 2016)

Based on the following theory, an Initial ERD has been generated for the given situation:

- One or more orders can include a certain painting.
- An order must include a minimum of one painting.
- It is possible to order a large number of paintings at once.

A many-to-many relationship occurs when several records in one table are linked to multiple records in another table. A direct many-to-many link between two tables is not advisable in relational databases since the data cannot be stored efficiently. By combining the many-to-many connection tables with an intersection table that contains both tables' keys, one can turn them into two one-to-many relationships for faster processing. (Datanamic, 2016)

## 2.1. List of Initially Created Entities and Attributes.

| Orders               | Painting                |
|----------------------|-------------------------|
| <u>Order_ID</u> (PK) | <u>Painting_ID</u> (PK) |
| Order_Date           | Painting_Name           |
| Total_Amount         | Issue_Date              |
| Customer_ID          | Availability            |
| Customer_Name        | Painting_Theme          |
| Customer_Address     | Monthly_Rental          |
| Customer_ContactNo   | Total_Rental            |
| Customer_Type        | Selling_Price           |
| Customer_Category    | Artist_ID               |
| Staff_ID             | Artist_Name             |
| Staff_Name           | Artist_Email            |
| Staff_Salary         | Buy_Or_Lease            |
| Staff_ContactNo      | Return_Date             |
|                      | Sold_Value              |

Figure 2: Initial Entities and Attributes.

## 2.2. Entity Relation Diagram.

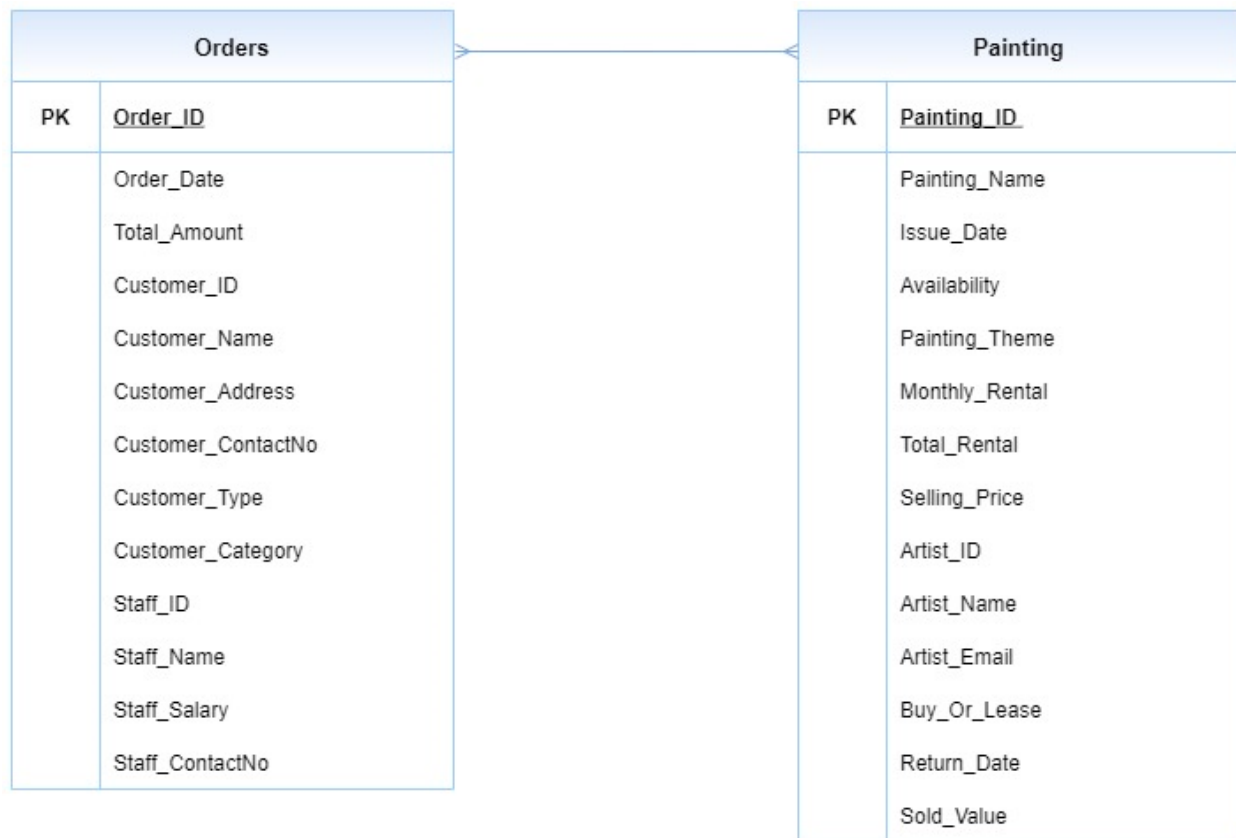


Figure 3: Initial ERD.

As a rough sketch, an initial ERD has been made. In the first diagrammatical form, the firm's rough data flow has been illustrated. Two entities are created; Orders and Painting. There are many inconsistencies in the ERD i.e., Many-to-many relationships, data redundancy, partial dependency, and transitive dependency. Although a direct many-to-many link between two tables is not encouraged in relational databases since the data cannot be stored efficiently it can be used in the initial erd as it is basically a rough sketch whereas the ERD after normalization depicts the database design for Masterpieces Limited.

### 3. Normalization.

The process of structuring data in a database is known as normalization also known as data normalization. It involves developing tables and developing relationships between them based on rules aimed at securing data while also making the database more adaptive by eliminating redundancy and inconsistent dependencies. Normalization, accomplishes this by breaking large tables into smaller ones and using relationships to link them. (Qui, et al., 2021)

#### 3.1. UNF (Unnormalized Form).

Un-normalized relationships have a lot of flaws and have not been subjected to any normalization rules.

To achieve normalization, the very first step is to:

- Make a list of all the attributes one can think of while reading the scenario and try not repeating the traits.
- Identify the main Primary Key (Unique field). A primary key should be distinguishable so it is presented by an underline.
- It is necessary to indent the repeated groupings. Use curly brackets {} to do so.
- Give the relation a name. (Lawsons, 2012)

---

#### UNF:

**Orders** (Order\_ID, Order\_Date, Total\_Amount, Customer\_ID, Customer\_Name, Customer\_Address, Customer\_ContactNo, Customer\_Type, Customer\_Category, Staff\_ID, Staff\_Name, Staff\_Salary, Staff\_ContactNo, {Painting\_ID, Painting\_Name, Issue\_Date, Availability, Painting\_Theme, Monthly\_Rental, Total\_Rental, Selling\_Price, Artist\_ID, Artist\_Name, Artist\_Email, Buy\_Or\_Lease, Return\_Date, Sold\_Value})

---

### 3.2. 1NF (First Normal Form).

A table is considered to be in First Normal Form if it has no data repetition and no multi-valued attributes. The 1NF is one of the simplest steps in the Normalization procedure.

In order to convert the UNF to 1NF; the following steps are to be followed:

- Confiscate the repeating groups into a fresh entity.
- Determine a new primary key for the entity and underline it.
- Insert a foreign key between the two tables to establish a relationship amongst them.

A foreign key should be represented by an underline and asterisk \*. (Lawsons, 2012)

---

#### 1NF:

**Orders\_1** (Order\_ID, Order\_Date, Total\_Amount, Customer\_ID, Customer\_Name, Customer\_Address, Customer\_ContactNo, Customer\_Type, Customer\_Category, Staff\_ID, Staff\_Name, Staff\_Salary, Staff\_ContactNo)

**Painting\_1** (Painting\_ID, Painting\_Name, Issue\_Date, Availability, Painting\_Theme, Monthly\_Rental, Total\_Rental, Selling\_Price, Artist\_ID, Artist\_Name, Artist\_Email, Buy\_Or\_Lease, Return\_Date, Sold\_Value, Order\_ID\*)

---



### 3.3. 2NF (Second Normal Form).

It is considered to be in second normal form if a table is in the first normal form and there is no partial reliance, and every non-key attribute is completely dependent on the primary key.

To change 1NF to 2NF; the following steps are to be trailed:

- Examine tables that have a composite key.
- Evaluate each non-key attribute to discover if it relies on a segment or the entire key.
- Any partial keys and dependents should be moved to a new table. (Lawsons, 2012)

#### Checking for Partial Dependency:

##### Orders\_2 = Orders\_1

When examining for partial dependency, it is discovered that the order table only has one primary key, however two primary keys are necessary to create a composite key so no partial dependency is found. As a result, Order\_1 is already in 2NF.

#### However, in Painting\_1 partial dependency is found:

Painting\_ID → Painting\_Name, Issue\_Date, Availability, Painting\_Theme,  
Monthly\_Rental, Total\_Rental, Selling\_Price, Artist\_ID, Artist\_Name, Artist\_Email

Orders\_ID → ✕

Order\_ID, Painting\_ID → Buy\_Or\_Lease, Return\_Date, Sold\_Value

**Three tables are now formed which is listed down below:**

---

**2NF:**

**Orders\_2** (Order\_ID, Order\_Date, Total\_Amount, Customer\_ID, Customer\_Name, Customer\_Address, Customer\_ContactNo, Customer\_Type, Customer\_Category, Staff\_ID, Staff\_Name, Staff\_Salary, Staff\_ContactNo)

**Painting\_2** (Painting\_ID, Issue\_Date, Availability, Painting\_Theme, Monthly\_Rental, Total\_Rental, Selling\_Price, Artist\_ID, Artist\_Name, Artist\_Email)

**Order\_Painting\_2** (Buy\_Or\_Lease, Return\_Date, Sold\_Value, Painting\_ID\*, Order\_ID\*)

---

### 3.4. 3NF (Third Normal Form).

In 3NF, transitive dependencies should be eliminated. Records that are not dependent on the main key but are dependent on any non-key should be separated into multiple tables.

To convert 2NF to 3NF; the following steps should be followed:

- Within each table, look out for any non-key attribute dependencies.
- It should be moved on to a different table.
- Primary key should be decided.
- This main key will become the foreign key in the original table.

#### Checking for Transitive Dependency:

##### Orders\_2

Order\_ID → Customer\_ID → Customer\_Name, Customer\_Address  
Customer\_ContactNo, Customer\_Type, Customer\_Category

Order\_ID → Staff\_ID → Staff\_Name, Staff\_Salary, Staff\_ContactNo

##### Painting\_2

Painting\_ID → Artist\_ID → Artist\_Name, Artist\_Email

##### Order\_Painting\_3 = Order\_Painting\_2

Order\_Painting\_2's composite primary key is the only thing that connects all of the values in the database. A non-value keys has no bearing on the value of another non-key. As a result, there are no transitive dependencies in this table.

**3NF formed six tables listed down below:**


---

**Orders\_3** (Order\_ID, Order\_Date, Total\_Amount, Customer\_ID\*, Staff\_ID\*)

**Customer\_3** (Customer\_ID, Customer\_Name, Customer\_Address, Customer\_ContactNo, Customer\_Type, Customer\_Category)

**Staff\_3** (Staff\_ID, Staff\_Name, Staff\_Salary, Staff\_ContactNo)

**Painting\_3** (Painting\_ID, Painting\_Name, Issue\_Date, Availability, Painting\_Theme, Monthly\_Rental, Total\_Rental, Selling\_Price, Artist\_ID\*)

**Artist\_3** (Artist\_ID, Artist\_Name, Artist\_Email)

**Order\_Painting\_3** (Buy\_Or\_Lease, Return\_Date, Sold\_Value, Painting\_ID\*, Order\_ID\*)

---

**The final tables are listed down below:**


---

**Orders** (Order\_ID, Order\_Date, Total\_Amount, Customer\_ID\*, Staff\_ID\*)

**Customer** (Customer\_ID, Customer\_Name, Customer\_Address, Customer\_ContactNo, Customer\_Type, Customer\_Category)

**Staff** (Staff\_ID, Staff\_Name, Staff\_Salary, Staff\_ContactNo)

**Painting** (Painting\_ID, Painting\_Name, Issue\_Date, Availability, Painting\_Theme, Monthly\_Rental, Total\_Rental, Selling\_Price, Artist\_ID\*)

**Artist** (Artist\_ID, Artist\_Name, Artist\_Email)

**Order\_Painting** (Buy\_Or\_Lease, Return\_Date, Sold\_Value, Painting\_ID\*, Order\_ID\*)

---

## 4. Final ERD



Figure 4: Final ERD.

In a final diagrammatical form, the firm's data flow has been illustrated. The inconsistencies in the original ERD have been removed. Many-to-many relationships,

data redundancy, partial dependency, and transitive dependency are all eradicated. A direct many-to-many link between two tables is not encouraged in relational databases since the data cannot be stored efficiently. The completed ERD depicts the database design for Masterpieces Limited.

With the help of normalization, a total of six entities have been created. The terms utilized include:

- Artist
- Staff
- Customer
- Painting
- Order
- Order Painting.

Following the creation of a database design, the company can now go forth to the implementation.

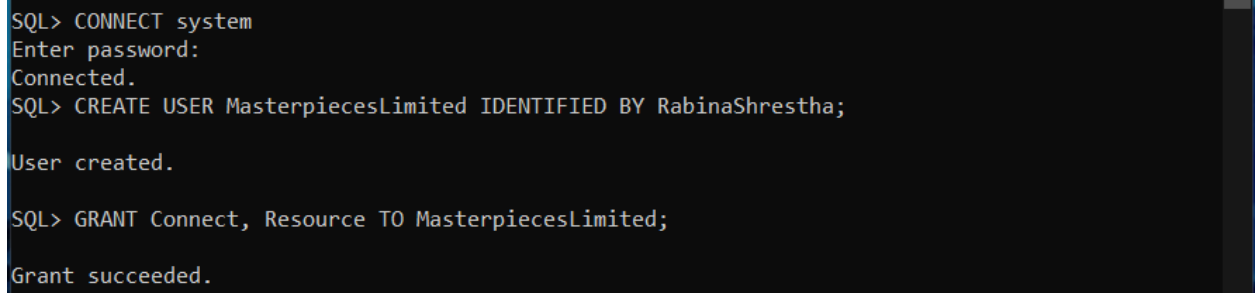
## 5. Implementation.

Database implementation is the process of installing database software, setting and modifying it, operating and testing the database, and finally integrating it with applications. (Enterprise Integration, 2020)

### 5.1. Table Generation

Let us now generate the six required tables for Masterpieces Limited. For a table to be generated one must create a user and grant privileges.

#### 1. Creating the User and Granting Permissions:

A screenshot of a terminal window with a dark background and light-colored text. It shows a sequence of SQL commands and their outputs. The commands are: 'CONNECT system', 'CREATE USER MasterpiecesLimited IDENTIFIED BY RabinaShrestha;', and 'GRANT Connect, Resource TO MasterpiecesLimited;'. The outputs are: 'Enter password:', 'Connected.', 'User created.', and 'Grant succeeded.'.

```
SQL> CONNECT system
Enter password:
Connected.
SQL> CREATE USER MasterpiecesLimited IDENTIFIED BY RabinaShrestha;

User created.

SQL> GRANT Connect, Resource TO MasterpiecesLimited;

Grant succeeded.
```

*Figure 5: Creating the User and Granting Permissions.*

In order, to create a user one must connect / login to the system.

The basic command to create a user is

```
CREATE USER <username in this case MasterpiecesLimited> IDENTIFIED BY
<password: RabinaShrestha>;
```

After creating a user, permission should be granted to them.

```
GRANT <privilege i.e., connect, resource> TO <user: MasterpiecesLimited>;
```

As seen in the picture above a user MastepieceLimited has been created which is identified by the password RabinaShrestha and it has been granted privileges.

## 2. Creating Tables:

The query to create a table:

```
CREATE TABLE <tablename> (column data_type, column1 data_type, CONSTRAINT  
constraint_name <column_constraint> (constraint_element));
```

### Creating table Artist.

```
SQL> CONNECT MasterpiecesLimited
Enter password:
Connected.
SQL> SET LINESIZE 75
SQL> SET PAGESIZE 20
SQL> CREATE TABLE Artist (Artist_ID VARCHAR2(5), Artist_Name VARCHAR2(30), Artist_Email VARCHAR  
2(40), CONSTRAINT Artist_pk PRIMARY KEY(Artist_ID), CONSTRAINT Artist_uk UNIQUE(Artist_Email));

Table created.

SQL> DESC Artist
Name                                Null?    Type
-----
ARTIST_ID                          NOT NULL VARCHAR2(5)
ARTIST_NAME                        VARCHAR2(30)
ARTIST_EMAIL                       VARCHAR2(40)
```

Figure 6: Creating Table: Artist.

Linesize controls the maximum number of characters that can be printed on a single physical line.

Pagesize controls the number of written lines that will fit on a single output page.

Desc command is used to illustrate the structure of a table. It shows the column name, nullability and the data type.

**Query:** CREATE TABLE Artist (Artist\_ID VARCHAR2(5), Artist\_Name VARCHAR2(30), Artist\_Email VARCHAR2(40), CONSTRAINT Artist\_pk PRIMARY KEY(Artist\_ID), CONSTRAINT Artist\_uk UNIQUE(Artist\_Email));



### Creating table Staff.

```
SQL> CREATE TABLE Staff (Staff_ID VARCHAR2(5), Staff_Name VARCHAR2(30), Staff_Salary NUMBER(10), Staff_ContactNo VARCHAR2(10), CONSTRAINT Staff_pk PRIMARY KEY(Staff_ID), CONSTRAINT Staff_uk UNIQUE(Staff_ContactNo));
```

Table created.

```
SQL> DESC Staff
```

| Name            | Null?    | Type         |
|-----------------|----------|--------------|
| STAFF_ID        | NOT NULL | VARCHAR2(5)  |
| STAFF_NAME      |          | VARCHAR2(30) |
| STAFF_SALARY    |          | NUMBER(10)   |
| STAFF_CONTACTNO |          | VARCHAR2(10) |

Figure 7: Creating Table: Staff.

**Query:** CREATE TABLE Staff (Staff\_ID VARCHAR2(5), Staff\_Name VARCHAR2(30), Staff\_Salary NUMBER (10), Staff\_ContactNo VARCHAR2(10), CONSTRAINT Staff\_pk PRIMARY KEY(Staff\_ID), CONSTRAINT Staff\_uk UNIQUE(Staff\_ContactNo));

### Creating table Customer.

```
SQL> CREATE TABLE Customer (Customer_ID VARCHAR2(5), Customer_Name VARCHAR2(30), Customer_Address VARCHAR2(30), Customer_ContactNo VARCHAR2(10), Customer_Type VARCHAR2(15), Customer_Category VARCHAR2(10), CONSTRAINT Customer_pk PRIMARY KEY(Customer_ID));
```

Table created.

```
SQL> DESC Customer
```

| Name               | Null?    | Type         |
|--------------------|----------|--------------|
| CUSTOMER_ID        | NOT NULL | VARCHAR2(5)  |
| CUSTOMER_NAME      |          | VARCHAR2(30) |
| CUSTOMER_ADDRESS   |          | VARCHAR2(30) |
| CUSTOMER_CONTACTNO |          | VARCHAR2(10) |
| CUSTOMER_TYPE      |          | VARCHAR2(15) |
| CUSTOMER_CATEGORY  |          | VARCHAR2(10) |

Figure 8: Creating Table: Customer.

**Query:** CREATE TABLE Customer (Customer\_ID VARCHAR2(5), Customer\_Name VARCHAR2(30), Customer\_Address VARCHAR2(30), Customer\_ContactNo VARCHAR2(10), Customer\_Type VARCHAR2(15), Customer\_Category VARCHAR2(10), CONSTRAINT Customer\_pk PRIMARY KEY(Customer\_ID));

## Creating table Painting.

```
SQL> CREATE TABLE Painting (Painting_ID VARCHAR2(5), Artist_ID VARCHAR2(5), Painting_Name
VARCHAR2(30), Issue_Date DATE, Availability VARCHAR2(10), Painting_Theme VARCHAR2(10), M
onthly_Rental NUMBER(15) NOT NULL, Total_Rental NUMBER(15), Selling_Price NUMBER(15) NOT
NULL, CONSTRAINT Painting_pk PRIMARY KEY(Painting_ID), FOREIGN KEY (Artist_ID) REFERENCES
Artist(Artist_ID));

Table created.

SQL> DESC Painting;
Name                                     Null?      Type
-----
PAINTING_ID                             NOT NULL   VARCHAR2(5)
ARTIST_ID                               VARCHAR2(5)
PAINTING_NAME                           VARCHAR2(30)
ISSUE_DATE                             DATE
AVAILABILITY                           VARCHAR2(10)
PAINTING_THEME                         VARCHAR2(10)
MONTHLY_RENTAL                         NOT NULL   NUMBER(15)
TOTAL_RENTAL                           NUMBER(15)
SELLING_PRICE                          NOT NULL   NUMBER(15)
```

Figure 9: Creating Table: Painting.

**Query:** CREATE TABLE Painting (Painting\_ID VARCHAR2(5), Artist\_ID VARCHAR2(5), Painting\_Name VARCHAR2(30), Issue\_Date DATE, Availability VARCHAR2(10), Painting\_Theme VARCHAR2(10), Monthly\_Rental NUMBER (15) NOT NULL, Total\_Rental NUMBER (15), Selling\_Price NUMBER (15) NOT NULL, CONSTRAINT Painting\_pk PRIMARY KEY(Painting\_ID), FOREIGN KEY (Artist\_ID) REFERENCES Artist (Artist\_ID));

**Creating table Orders.**

```
SQL> CREATE TABLE Orders (Order_ID VARCHAR2(5), Customer_ID VARCHAR2(5), Staff_ID VARCHAR2(5), Order_Date DATE, Total_Amount NUMBER(15), CONSTRAINT Order_pk PRIMARY KEY (Order_ID), FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID), FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID));
```

Table created.

```
SQL> DESC Orders
```

| Name         | Null?    | Type        |
|--------------|----------|-------------|
| ORDER_ID     | NOT NULL | VARCHAR2(5) |
| CUSTOMER_ID  |          | VARCHAR2(5) |
| STAFF_ID     |          | VARCHAR2(5) |
| ORDER_DATE   |          | DATE        |
| TOTAL_AMOUNT |          | NUMBER(15)  |

*Figure 10: Creating Table: Orders.*

**Query:** CREATE TABLE Orders (Order\_ID VARCHAR2(5), Customer\_ID VARCHAR2(5), Staff\_ID VARCHAR2(5), Order\_Date DATE, Total\_Amount NUMBER(15), CONSTRAINT Order\_pk PRIMARY KEY (Order\_ID), FOREIGN KEY (Customer\_ID) REFERENCES Customer (Customer\_ID), FOREIGN KEY (Staff\_ID) REFERENCES Staff (Staff\_ID));

### Creating table Order\_Painting.

```
SQL> CREATE TABLE Order_Painting (Order_ID VARCHAR2(5), Painting_ID VARCHAR2(5), Buy_Or_Lease
VARCHAR(7), Return_Date DATE, Sold_Value NUMBER(15), CONSTRAINT Painting_Order_pk PRIMARY KEY
(Painting_ID, Order_ID), FOREIGN KEY (Painting_ID) REFERENCES Painting(Painting_ID), FOREIGN K
EY (Order_ID) REFERENCES Orders(Order_ID));

Table created.

SQL> DESC Order_Painting;
```

| Name         | Null?    | Type        |
|--------------|----------|-------------|
| ORDER_ID     | NOT NULL | VARCHAR2(5) |
| PAINTING_ID  | NOT NULL | VARCHAR2(5) |
| BUY_OR_LEASE |          | VARCHAR2(7) |
| RETURN_DATE  |          | DATE        |
| SOLD_VALUE   |          | NUMBER(15)  |

Figure 11: Creating Table: Order\_Painting.

**Query:** CREATE TABLE Order\_Painting (Order\_ID VARCHAR2(5), Painting\_ID VARCHAR2(5), Buy\_Or\_Lease VARCHAR (7), Return\_Date DATE, Sold\_Value NUMBER (15), CONSTRAINT Painting\_Order\_pk PRIMARY KEY (Painting\_ID, Order\_ID), FOREIGN KEY (Painting\_ID) REFERENCES Painting (Painting\_ID), FOREIGN KEY (Order\_ID) REFERENCES Orders (Order\_ID));

## 5.2. Populating the Database Tables.

The basic query for inserting values is:

```
INSERT INTO tablename (columnvalues, columnvalues2);
```

It can also be inserted into some columns by the following query

```
INSERT INTO tablename (columnnames) (columnvalues);
```

**Inserting values in the table Artist.**

```

SQL> INSERT ALL
  2 INTO Artist VALUES ('A101', 'Anugraha Ghale', 'anugraha.ghale.artist.mltd@gmail.com')
  3 INTO Artist VALUES ('A102', 'Mihir Shrestha', 'mihir.shrestha.artist.mltd@gmail.com')
  4 INTO Artist VALUES ('A103', 'Padu Pradhan', 'padu.pradhan.artist.mltd@gmail.com')
  5 INTO Artist VALUES ('A104', 'Rimjhim Agrawal', 'rimjhim.agrawal.artist.mltd@gmail.com')
  6 INTO Artist VALUES ('A105', 'Roshish Shrestha', 'roshish.shrestha.artist.mltd@gmail.com')
  7 INTO Artist VALUES ('A106', 'Yuthika Gauchan', 'yuthika.gauchan.artist.mltd@gmail.com')
  8 INTO Artist VALUES ('A107', 'Srija Pradhan', 'srija.pradhan.artist.mltd@gmail.com')
  9 INTO Artist VALUES ('A108', 'Aastha Sthapit', 'aastha.sthapit.artist.mltd@gmail.com')
 10 INTO Artist VALUES ('A109', 'Biji Sharma', 'biji.sharma.artist.mltd@gmail.com')
 11 INTO Artist VALUES ('A110', 'Dawa Karki', 'dawa.karki.artist.mltd@gmail.com')
 12 SELECT * FROM DUAL;

10 rows created.

```

*Figure 12: Insert into Artist Values.***Query: INSERT ALL**

INTO Artist VALUES ('A101', 'Anugraha Ghale', 'anugraha.ghale.artist.mltd@gmail.com')

INTO Artist VALUES ('A102', 'Mihir Shrestha', 'mihir.shrestha.artist.mltd@gmail.com')

INTO Artist VALUES ('A103', 'Padu Pradhan', 'padu.pradhan.artist.mltd@gmail.com')

INTO Artist VALUES ('A104', 'Rimjhim Agrawal', 'rimjhim.agrawal.artist.mltd@gmail.com')

INTO Artist VALUES ('A105', 'Roshish Shrestha', 'roshish.shrestha.artist.mltd@gmail.com')

INTO Artist VALUES ('A106', 'Yuthika Gauchan', 'yuthika.gauchan.artist.mltd@gmail.com')

INTO Artist VALUES ('A107', 'Srija Pradhan', 'srija.pradhan.artist.mltd@gmail.com')

INTO Artist VALUES ('A108', 'Aastha Sthapit', 'aastha.sthapit.artist.mltd@gmail.com')

INTO Artist VALUES ('A109', 'Biji Sharma', 'biji.sharma.artist.mltd@gmail.com')

INTO Artist VALUES ('A110', 'Dawa Karki', 'dawa.karki.artist.mltd@gmail.com')

SELECT \* FROM DUAL;

**Inserting values in the table Staff.**

```
SQL> INSERT ALL
  2 INTO Staff VALUES ('S101', 'Kenny Stokes', 45000, 7456541579)
  3 INTO Staff VALUES ('S102', 'Nicolas Cameron', 13500, 7456541680)
  4 INTO Staff VALUES ('S103', 'Clarence Vega', 17800, 7456541781)
  5 INTO Staff VALUES ('S104', 'Royal Serrano', 20000, 7456541882)
  6 INTO Staff VALUES ('S105', 'Julius Avila', 34000, 7456541983)
  7 INTO Staff VALUES ('S106', 'Valentin Black', 50000, 7456542084)
  8 INTO Staff VALUES ('S107', 'Flynn Rider', 57000, 7456542185)
  9 SELECT * FROM DUAL;

7 rows created.
```

*Figure 13: Insert into Staff Values.*

**Query: INSERT ALL**

```
INTO Staff VALUES ('S101', 'Kenny Stokes', 45000, 7456541579)

INTO Staff VALUES ('S102', 'Nicolas Cameron', 13500, 7456541680)

INTO Staff VALUES ('S103', 'Clarence Vega', 17800, 7456541781)

INTO Staff VALUES ('S104', 'Royal Serrano', 20000, 7456541882)

INTO Staff VALUES ('S105', 'Julius Avila', 34000, 7456541983)

INTO Staff VALUES ('S106', 'Valentin Black', 50000, 7456542084)

INTO Staff VALUES ('S107', 'Flynn Rider', 57000, 7456542185)

SELECT * FROM DUAL;
```

**Inserting values in the table Customer.**

```
SQL> INSERT ALL
  2 INTO Customer VALUES ('C101', 'Merle Liu', 'Green Lane', 7911875087, 'Individual', 'Loyal')
  3 INTO Customer VALUES ('C102', 'Akatsuki Ltd', 'Kings Road', 7789870698, 'Commercial', 'VIP')
  4 INTO Customer VALUES ('C103', 'Rubin Evans', 'Victoria Road', 7700118726, 'Individual', 'Regular')
  5 INTO Customer VALUES ('C104', 'EdRile Ltd', 'North Street', 7911846809, 'Commercial', 'Regular')
  6 INTO Customer VALUES ('C105', 'Elvis Ltd', 'Queensway', 7502237612, 'Commercial', 'Privileged')
  7 INTO Customer VALUES ('C106', 'Sidney Nelson', 'Richmond Road', 7700099098, 'Individual', 'Loyal')
  8 INTO Customer VALUES ('C107', 'Winter Avila', 'Grange Road', 7700072059, 'Individual', 'Privileged')
  9 INTO Customer VALUES ('C108', 'Shelly Scott', 'Manor Road', 7456542296, 'Individual', 'VIP')
 10 SELECT * FROM DUAL;

8 rows created.
```

*Figure 14: Insert into Customer Values.*

**Query: INSERT ALL**

INTO Customer VALUES ('C101', 'Merle Liu', 'Green Lane', 7911875087, 'Individual', 'Loyal')

INTO Customer VALUES ('C102', 'Akatsuki Ltd', 'Kings Road', 7789870698, 'Commercial', 'VIP')

INTO Customer VALUES ('C103', 'Rubin Evans', 'Victoria Road', 7700118726, 'Individual', 'Regular')

INTO Customer VALUES ('C104', 'EdRile Ltd', 'North Street', 7911846809, 'Commercial', 'Regular')

INTO Customer VALUES ('C105', 'Elvis Ltd', 'Queensway', 7502237612, 'Commercial', 'Privileged')

INTO Customer VALUES ('C106', 'Sidney Nelson', 'Richmond Road', 7700099098, 'Individual', 'Loyal')

INTO Customer VALUES ('C107', 'Winter Avila', 'Grange Road', 7700072059, 'Individual', 'Privileged')

INTO Customer VALUES ('C108', 'Shelly Scott', 'Manor Road', 7456542296, 'Individual', 'VIP')

SELECT \* FROM DUAL;

## Inserting values in the table Painting.

```

SQL> INSERT ALL
2 INTO Painting VALUES ('P101','A101', 'Autumn', '27-May-21', 'Leased Out', 'Landscape', 1200,2688, 15500)
3 INTO Painting VALUES ('P102','A105', 'Signs of Childhood', '30-May-21', 'Sold', 'Person', 900, 1224, 11000)
4 INTO Painting VALUES ('P103','A104', 'Judgment of Weakness', '31-May-21', 'Leased Out', 'Person', 800, 1152, 9600)
5 INTO Painting VALUES ('P104','A108', 'Muddled Beggar', '02-Jun-21', 'Returned', 'Animal', 1300, 884, 15600)
6 INTO Painting VALUES ('P105','A110', 'Luxury of Fire', '12-Jul-21', 'Sold', 'Landscape', 780, 6365, 9360)
7 INTO Painting (Painting_ID, Artist_ID, Painting_Name, Issue_Date, Availability, Painting_Theme, Monthly_Rental, Selling_Price
) VALUES ('P106','A102', 'Lifetime Partner', '03-Aug-21', 'Returned', 'Animal', 1000, 12000)
8 INTO Painting VALUES ('P107','A109', 'Nobody', '22-Aug-21', 'Available', 'Person', 950, 760, 11400)
9 INTO Painting (Painting_ID, Artist_ID, Painting_Name, Issue_Date, Availability, Painting_Theme, Monthly_Rental, Selling_Price
) VALUES ('P108','A103', 'Serene Wave', '25-Aug-21', 'Returned', 'Landscape', 1000, 12350)
10 INTO Painting (Painting_ID, Artist_ID, Painting_Name, Issue_Date, Availability, Painting_Theme, Monthly_Rental, Selling_Price
) VALUES ('P109','A102', 'Instrument of Death', '05-Sep-21', 'Sold', 'Landscape', 1150, 13800)
11 INTO Painting VALUES ('P110','A103', 'Tainted Throne', '06-Sep-21', 'Leased Out', 'Person', 600, 408, 7200)
12 INTO Painting VALUES ('P111','A107', 'Frivolous Animal', '18-Sep-21', 'Leased Out', 'Animal', 750, 600, 9000)
13 INTO Painting VALUES ('P112','A106', 'Familiar Night', '19-Oct-21', 'Sold', 'Landscape', 1000, 680, 11000)
14 INTO Painting (Painting_ID, Artist_ID, Painting_Name, Issue_Date, Availability, Painting_Theme, Monthly_Rental, Selling_Price
) VALUES ('P113','A106', 'Monstrous Dictatorship', '11-Nov-21', 'Sold', 'Person', 860, 10500)
15 INTO Painting VALUES ('P114','A105', 'Mask of Flame', '13-Nov-21', 'Available', 'Person', 980, 705, 11700)
16 INTO Painting (Painting_ID, Artist_ID, Painting_Name, Issue_Date, Availability, Painting_Theme, Monthly_Rental, Selling_Price
) VALUES ('P115','A102', 'Luxury', '16-Dec-21', 'Available', 'Landscape', 500, 6000)
17 SELECT * FROM DUAL;

15 rows created.

```

Figure 15: Insert into Painting Values.

### Query: INSERT ALL

INTO Painting VALUES ('P101','A101', 'Autumn', '27-May-21', 'Leased Out', 'Landscape',  
1200, 2688, 15500)

INTO Painting VALUES ('P102','A105', 'Signs of Childhood', '30-May-21', 'Sold', 'Person',  
900, 1224, 11000)

INTO Painting VALUES ('P103','A104', 'Judgment of Weakness', '31-May-21', 'Leased  
Out', 'Person', 800, 1152, 9600)

INTO Painting VALUES ('P104','A108', 'Muddled Beggar', '02-Jun-21', 'Returned',  
'Animal', 1300, 884, 15600)

INTO Painting VALUES ('P105','A110', 'Luxury of Fire', '12-Jul-21', 'Sold', 'Landscape',  
780, 6365, 9360)

INTO Painting (Painting\_ID, Artist\_ID, Painting\_Name, Issue\_Date, Availability,  
Painting\_Theme, Monthly\_Rental, Selling\_Price) VALUES ('P106','A102', 'Lifetime  
Partner', '03-Aug-21', 'Returned', 'Animal', 1000, 12000)



```
INTO Painting VALUES ('P107','A109', 'Nobody', '22-Aug-21', 'Available', 'Person', 950, 760, 11400)
```

```
INTO Painting (Painting_ID, Artist_ID, Painting_Name, Issue_Date, Availability, Painting_Theme, Monthly_Rental, Selling_Price) VALUES ('P108','A103', 'Serene Wave', '25-Aug-21', 'Returned', 'Landscape', 1000, 12350)
```

```
INTO Painting (Painting_ID, Artist_ID, Painting_Name, Issue_Date, Availability, Painting_Theme, Monthly_Rental, Selling_Price) VALUES ('P109','A102', 'Instrument of Death', '05-Sep-21', 'Sold', 'Landscape', 1150, 13800)
```

```
INTO Painting VALUES ('P110','A103', 'Tainted Throne', '06-Sep-21', 'Leased Out', 'Person', 600, 408, 7200)
```

```
INTO Painting VALUES ('P111','A107', 'Frivolous Animal', '18-Sep-21', 'Leased Out', 'Animal', 750, 600, 9000)
```

```
INTO Painting VALUES ('P112','A106', 'Familiar Night', '19-Oct-21', 'Sold', 'Landscape', 1000, 680, 11000)
```

```
INTO Painting (Painting_ID, Artist_ID, Painting_Name, Issue_Date, Availability, Painting_Theme, Monthly_Rental, Selling_Price) VALUES ('P113','A106', 'Monstrous Dictatorship', '11-Nov-21', 'Sold', 'Person', 860, 10500)
```

```
INTO Painting VALUES ('P114','A105', 'Mask of Flame', '13-Nov-21', 'Available', 'Person', 980, 705, 11700)
```

```
INTO Painting (Painting_ID, Artist_ID, Painting_Name, Issue_Date, Availability, Painting_Theme, Monthly_Rental, Selling_Price) VALUES ('P115','A102', 'Luxury', '16-Dec-21', 'Available', 'Landscape', 500, 6000)
```

```
SELECT * FROM DUAL;
```

**Inserting values in the table Orders.**

```
SQL> INSERT ALL
  2 INTO Orders VALUES ('O101', 'C108', 'S107', '07-Jun-21', 1870)
  3 INTO Orders VALUES ('O102', 'C101', 'S101', '27-Jul-21', 1140)
  4 INTO Orders VALUES ('O103', 'C107', 'S107', '29-Jul-21', 720)
  5 INTO Orders VALUES ('O104', 'C103', 'S105', '15-Sep-21', 950)
  6 INTO Orders VALUES ('O105', 'C108', 'S104', '21-Sep-21', 8721)
  7 INTO Orders VALUES ('O106', 'C103', 'S102', '09-Oct-21', 750)
  8 INTO Orders VALUES ('O107', 'C104', 'S106', '18-Oct-21', 12200)
  9 INTO Orders VALUES ('O108', 'C102', 'S103', '01-Nov-21', 850)
 10 INTO Orders VALUES ('O109', 'C105', 'S101', '05-Nov-21', 882)
 11 INTO Orders VALUES ('O110', 'C108', 'S107', '08-Dec-21', 1530)
 12 INTO Orders VALUES ('O111', 'C105', 'S104', '14-Dec-21', 13140)
 13 INTO Orders VALUES ('O112', 'C106', 'S101', '20-Dec-21', 20425)
 14 SELECT * FROM DUAL;

12 rows created.
```

*Figure 16: Insert into Orders Values.*

**Query: INSERT ALL**

```
INSERT INTO Orders VALUES ('O101', 'C108', 'S107', '07-Jun-21', 1870)
INSERT INTO Orders VALUES ('O102', 'C101', 'S101', '27-Jul-21', 1140)
INSERT INTO Orders VALUES ('O103', 'C107', 'S107', '29-Jul-21', 720)
INSERT INTO Orders VALUES ('O104', 'C103', 'S105', '15-Sep-21', 950)
INSERT INTO Orders VALUES ('O105', 'C108', 'S104', '21-Sep-21', 8721)
INSERT INTO Orders VALUES ('O106', 'C103', 'S102', '09-Oct-21', 750)
INSERT INTO Orders VALUES ('O107', 'C104', 'S106', '18-Oct-21', 12200)
INSERT INTO Orders VALUES ('O108', 'C102', 'S103', '01-Nov-21', 850)
INSERT INTO Orders VALUES ('O109', 'C105', 'S101', '05-Nov-21', 882)
INSERT INTO Orders VALUES ('O110', 'C108', 'S107', '08-Dec-21', 1530)
INSERT INTO Orders VALUES ('O111', 'C105', 'S104', '14-Dec-21', 13140)
INSERT INTO Orders VALUES ('O112', 'C106', 'S101', '20-Dec-21', 20425)
SELECT * FROM DUAL;
```

### Inserting values in the table Order\_Painting.

```

SQL> INSERT ALL
 2 INTO Order_Painting VALUES ('O101', 'P102', 'Lease', '07-Jul-21', 765)
 3 INTO Order_Painting VALUES ('O101', 'P104', 'Lease', '07-Jul-21', 1105)
 4 INTO Order_Painting VALUES ('O102', 'P101', 'Lease', '27-Aug-21', 1140)
 5 INTO Order_Painting VALUES ('O103', 'P103', 'Lease', '29-Aug-21', 720)
 6 INTO Order_Painting VALUES ('O104', 'P107', 'Lease', '15-Oct-21', 950)
 7 INTO Order_Painting (Order_ID, Painting_ID, Buy_Or_Lease, Sold_Value) VALUES ('O105', 'P105', 'Buy', 7956)
 8 INTO Order_Painting VALUES ('O105', 'P102', 'Lease', '21-Oct-21', 765)
 9 INTO Order_Painting VALUES ('O106', 'P111', 'Lease', '09-Nov-21', 750)
10 INTO Order_Painting (Order_ID, Painting_ID, Buy_Or_Lease, Sold_Value) VALUES ('O107', 'P102', 'Buy', 11000)
11 INTO Order_Painting VALUES ('O107', 'P101', 'Lease', '19-Nov-21', 1200)
12 INTO Order_Painting VALUES ('O108', 'P112', 'Lease', '01-Dec-21', 850)
13 INTO Order_Painting VALUES ('O109', 'P114', 'Lease', '05-Dec-21', 882)
14 INTO Order_Painting VALUES ('O110', 'P101', 'Lease', '08-Jan-21', 1020)
15 INTO Order_Painting VALUES ('O110', 'P110', 'Lease', '08-Jan-21', 510)
16 INTO Order_Painting (Order_ID, Painting_ID, Buy_Or_Lease, Sold_Value) VALUES ('O111', 'P109', 'Buy', 12420)
17 INTO Order_Painting VALUES ('O111', 'P103', 'Lease', '14-Jan-21', 720)
18 INTO Order_Painting (Order_ID, Painting_ID, Buy_Or_Lease, Sold_Value) VALUES ('O112', 'P112', 'Buy', 10450)
19 INTO Order_Painting (Order_ID, Painting_ID, Buy_Or_Lease, Sold_Value) VALUES ('O112', 'P113', 'Buy', 9975)
20 SELECT * FROM DUAL;

18 rows created.

```

Figure 17: Insert into Order\_Painting Values,

#### Query:INSERT ALL

INTO Order\_Painting VALUES ('O101', 'P102', 'Lease', '07-Jul-21', 765)

INTO Order\_Painting VALUES ('O101', 'P104', 'Lease', '07-Jul-21', 1105)

INTO Order\_Painting VALUES ('O102', 'P101', 'Lease', '27-Aug-21', 1140)

INTO Order\_Painting VALUES ('O103', 'P103', 'Lease', '29-Aug-21', 720)

INTO Order\_Painting VALUES ('O104', 'P107', 'Lease', '15-Oct-21', 950)

INTO Order\_Painting (Order\_ID, Painting\_ID, Buy\_Or\_Lease, Sold\_Value) VALUES ('O105', 'P105', 'Buy', 7956)

INTO Order\_Painting VALUES ('O105', 'P102', 'Lease', '21-Oct-21', 765)

INTO Order\_Painting VALUES ('O106', 'P111', 'Lease', '09-Nov-21', 750)

INTO Order\_Painting (Order\_ID, Painting\_ID, Buy\_Or\_Lease, Sold\_Value) VALUES ('O107', 'P102', 'Buy', 11000)

```
INTO Order_Painting VALUES ('O107', 'P101', 'Lease', '19-Nov-21', 1200)
```

```
INTO Order_Painting VALUES ('O108', 'P112', 'Lease', '01-Dec-21', 850)
```

```
INTO Order_Painting VALUES ('O109', 'P114', 'Lease', '05-Dec-21', 882)
```

```
INTO Order_Painting VALUES ('O110', 'P101', 'Lease', '08-Jan-21', 1020)
```

```
INTO Order_Painting VALUES ('O110', 'P110', 'Lease', '08-Jan-21', 510)
```

```
INTO Order_Painting (Order_ID, Painting_ID, Buy_Or_Lease, Sold_Value) VALUES  
('O111', 'P109', 'Buy', 12420)
```

```
INTO Order_Painting VALUES ('O111', 'P103', 'Lease', '14-Jan-21', 720)
```

```
INTO Order_Painting (Order_ID, Painting_ID, Buy_Or_Lease, Sold_Value) VALUES  
('O112', 'P112', 'Buy', 10450)
```

```
INTO Order_Painting (Order_ID, Painting_ID, Buy_Or_Lease, Sold_Value) VALUES  
('O112', 'P113', 'Buy', 9975)
```

```
SELECT * FROM DUAL;
```

### 5.3. Displaying the Table.

The syntax to display a table is:

SELECT columnname or all \* from tablename;

#### Displaying Artist.

```
SQL> SELECT * FROM Artist;
```

| ARTIS | ARTIST_NAME      | ARTIST_EMAIL                           |
|-------|------------------|----------------------------------------|
| A101  | Anugraha Ghale   | anugraha.ghale.artist.mltd@gmail.com   |
| A102  | Mihir Shrestha   | mihir.shrestha.artist.mltd@gmail.com   |
| A103  | Padu Pradhan     | padu.pradhan.artist.mltd@gmail.com     |
| A104  | Rimjhim Agrawal  | rimjhim.agrawal.artist.mltd@gmail.com  |
| A105  | Roshish Shrestha | roshish.shrestha.artist.mltd@gmail.com |
| A106  | Yuthika Gauchan  | yuthika.gauchan.artist.mltd@gmail.com  |
| A107  | Srija Pradhan    | srija.pradhan.artist.mltd@gmail.com    |
| A108  | Aastha Sthapit   | aastha.sthapit.artist.mltd@gmail.com   |
| A109  | Biji Sharma      | biji.sharma.artist.mltd@gmail.com      |
| A110  | Dawa Karki       | dawa.karki.artist.mltd@gmail.com       |

10 rows selected.

Figure 18: Display Artist.

**Query:** SELECT \* FROM Artist;

#### Displaying Staff.

```
SQL> SELECT * FROM Staff;
```

| STAFF | STAFF_NAME      | STAFF_SALARY | STAFF_CONT |
|-------|-----------------|--------------|------------|
| S101  | Kenny Stokes    | 45000        | 7456541579 |
| S102  | Nicolas Cameron | 13500        | 7456541680 |
| S103  | Clarence Vega   | 17800        | 7456541781 |
| S104  | Royal Serrano   | 20000        | 7456541882 |
| S105  | Julius Avila    | 34000        | 7456541983 |
| S106  | Valentin Black  | 50000        | 7456542084 |
| S107  | Flynn Rider     | 57000        | 7456542185 |

7 rows selected.

Figure 19: Display Staff.

**Query:** SELECT \* FROM Staff

## Displaying Customer.

```
SQL> SELECT * FROM Customer;
```

| CUSTO | CUSTOMER_NAME | CUSTOMER_ADDRESS | CUSTOMER_C | CUSTOMER_TYPE | CUSTOMER_C |
|-------|---------------|------------------|------------|---------------|------------|
| C101  | Merle Liu     | Green Lane       | 7911875087 | Individual    | Loyal      |
| C102  | Akatsuki Ltd  | Kings Road       | 7789870698 | Commercial    | VIP        |
| C103  | Rubin Evans   | Victoria Road    | 7700118726 | Individual    | Regular    |
| C104  | EdRile Ltd    | North Street     | 7911846809 | Commercial    | Regular    |
| C105  | Elvis Ltd     | Queensway        | 7502237612 | Commercial    | Privileged |
| C106  | Sidney Nelson | Richmond Road    | 7700099098 | Individual    | Loyal      |
| C107  | Winter Avila  | Grange Road      | 7700072059 | Individual    | Privileged |
| C108  | Shelly Scott  | Manor Road       | 7456542296 | Individual    | VIP        |

8 rows selected.

Figure 20: Display Customer.

Query: SELECT \* FROM Customer;

## Displaying Painting.

```
SQL> SELECT * FROM Painting;
```

| PAINT | ARTIS | PAINTING_NAME          | ISSUE_DAT | AVAILABILI | PAINTING_T | MONTHLY_RENTAL | TOTAL_RENTAL | SELLING_PRICE |
|-------|-------|------------------------|-----------|------------|------------|----------------|--------------|---------------|
| P101  | A101  | Autumn                 | 27-MAY-21 | Leased Out | Landscape  | 1200           | 2688         | 15500         |
| P102  | A105  | Signs of Childhood     | 30-MAY-21 | Sold       | Person     | 900            | 1224         | 11000         |
| P103  | A104  | Judgment of Weakness   | 31-MAY-21 | Leased Out | Person     | 800            | 1152         | 9600          |
| P104  | A108  | Muddled Beggar         | 02-JUN-21 | Returned   | Animal     | 1300           | 884          | 15600         |
| P105  | A110  | Luxury of Fire         | 12-JUL-21 | Sold       | Landscape  | 780            | 6365         | 9360          |
| P106  | A102  | Lifetime Partner       | 03-AUG-21 | Returned   | Animal     | 1000           |              | 12000         |
| P107  | A109  | Nobody                 | 22-AUG-21 | Available  | Person     | 950            | 760          | 11400         |
| P108  | A103  | Serene Wave            | 25-AUG-21 | Returned   | Landscape  | 1000           |              | 12350         |
| P109  | A102  | Instrument of Death    | 05-SEP-21 | Sold       | Landscape  | 1150           |              | 13800         |
| P110  | A103  | Tainted Throne         | 06-SEP-21 | Leased Out | Person     | 600            | 408          | 7200          |
| P111  | A107  | Frivolous Animal       | 18-SEP-21 | Leased Out | Animal     | 750            | 600          | 9000          |
| P112  | A106  | Familiar Night         | 19-OCT-21 | Sold       | Landscape  | 1000           | 680          | 11000         |
| P113  | A106  | Monstrous Dictatorship | 11-NOV-21 | Sold       | Person     | 860            |              | 10500         |
| P114  | A105  | Mask of Flame          | 13-NOV-21 | Available  | Person     | 980            | 705          | 11700         |
| P115  | A102  | Luxury                 | 16-DEC-21 | Available  | Landscape  | 500            |              | 6000          |

15 rows selected.

Figure 21: Display Painting.

Query: SELECT \* FROM Painting;

**Displaying Order.**

```
SQL> SELECT * FROM Orders;
```

| ORDER | CUSTO | STAFF | ORDER_DAT | TOTAL_AMOUNT |
|-------|-------|-------|-----------|--------------|
| 0101  | C108  | S107  | 07-JUN-21 | 1870         |
| 0102  | C101  | S101  | 27-JUL-21 | 1140         |
| 0103  | C107  | S107  | 29-JUL-21 | 720          |
| 0104  | C103  | S105  | 15-SEP-21 | 950          |
| 0105  | C108  | S104  | 21-SEP-21 | 8721         |
| 0106  | C103  | S102  | 09-OCT-21 | 750          |
| 0107  | C104  | S106  | 18-OCT-21 | 12200        |
| 0108  | C102  | S103  | 01-NOV-21 | 850          |
| 0109  | C105  | S101  | 05-NOV-21 | 882          |
| 0110  | C108  | S107  | 08-DEC-21 | 1530         |
| 0111  | C105  | S104  | 14-DEC-21 | 13140        |
| 0112  | C106  | S101  | 20-DEC-21 | 20425        |

*Figure 22: Display Order.***Query:** SELECT \* FROM Orders.**Displaying Order\_Painting.**

```
SQL> SELECT * FROM Order_Painting;
```

| ORDER | PAINT | BUY_OR_ | RETURN_DA | SOLD_VALUE |
|-------|-------|---------|-----------|------------|
| 0101  | P102  | Lease   | 07-JUL-21 | 765        |
| 0101  | P104  | Lease   | 07-JUL-21 | 1105       |
| 0102  | P101  | Lease   | 27-AUG-21 | 1140       |
| 0103  | P103  | Lease   | 29-AUG-21 | 720        |
| 0104  | P107  | Lease   | 15-OCT-21 | 950        |
| 0105  | P105  | Buy     |           | 7956       |
| 0105  | P102  | Lease   | 21-OCT-21 | 765        |
| 0106  | P111  | Lease   | 09-NOV-21 | 750        |
| 0107  | P102  | Buy     |           | 11000      |
| 0107  | P101  | Lease   | 19-NOV-21 | 1200       |
| 0108  | P112  | Lease   | 01-DEC-21 | 850        |
| 0109  | P114  | Lease   | 05-DEC-21 | 882        |
| 0110  | P101  | Lease   | 08-JAN-21 | 1020       |
| 0110  | P110  | Lease   | 08-JAN-21 | 510        |
| 0111  | P109  | Buy     |           | 12420      |
| 0111  | P103  | Lease   | 14-JAN-21 | 720        |
| 0112  | P112  | Buy     |           | 10450      |
| 0112  | P113  | Buy     |           | 9975       |

18 rows selected.


*Figure 23: Display Order\_Painting.***Query:** SELECT \* FROM Order\_Painting.

## 5.4. Dropping the Tables.

The query for dropping a table is simply: DROP TABLE tablename;

However, it shows an error when the parent table is dropped first. The dropping of tables must begin with the child table. If the child-table relation is confusing one can start dropping tables from the last created table.

### Dropping Table Order\_Painting.


A screenshot of a terminal window with a dark background. The text 'SQL> DROP TABLE Order\_Painting;' is on the first line, and 'Table dropped.' is on the second line.

```
SQL> DROP TABLE Order_Painting;  
Table dropped.
```

*Figure 24: Dropping Table Order\_Painting.*

**Query:** DROP TABLE Order\_Painting;

### Dropping Table Orders.


A screenshot of a terminal window with a dark background. The text 'SQL> DROP TABLE Orders;' is on the first line, and 'Table dropped.' is on the second line.

```
SQL> DROP TABLE Orders;  
Table dropped.
```

*Figure 25: Dropping Table Orders.*

**Query:** DROP TABLE Orders;

### Dropping Table Painting.


A screenshot of a terminal window with a dark background. The text 'SQL> DROP TABLE Painting;' is on the first line, and 'Table dropped.' is on the second line.

```
SQL> DROP TABLE Painting;  
Table dropped.
```

*Figure 26: Dropping Table Painting.*

**Query:** DROP TABLE Painting;




**Dropping Table Customer.**A screenshot of a SQL command prompt window with a black background. The text 'SQL> DROP TABLE CUSTOMER;' is on the first line, and 'Table dropped.' is on the second line.

```
SQL> DROP TABLE CUSTOMER;  
Table dropped.
```

*Figure 27: Dropping Table Customer.*


**Query:** DROP TABLE Customer;

**Dropping Table Staff.**A screenshot of a SQL command prompt window with a black background. The text 'SQL> DROP TABLE Staff;' is on the first line, and 'Table dropped.' is on the second line.

```
SQL> DROP TABLE Staff;  
Table dropped.
```

*Figure 28: Dropping Table Staff.*

**Query:** DROP TABLE Staff;

**Dropping Table Artist.**A screenshot of a SQL command prompt window with a black background. The text 'SQL> DROP TABLE Artist;' is on the first line, and 'Table dropped.' is on the second line.

```
SQL> DROP TABLE Artist;  
Table dropped.
```

*Figure 29: Dropping Table Artist.*

**Query:** DROP TABLE Artist;

## 6. Database Querying.

A request for information from a database is referred to as a "database query." Data retrieval is the most typical request, although queries can also be used to alter data. One or more tables, as well as supplementary queries, could provide the data. (Gibbs, 2021)

### 6.1. Information Query

The following information queries are spooled in order to create a direct output of any query to a server-side flat file. (Burleson, 2015)

#### 6.1.1. List all customers according to category:

```
SQL> spool D:\DatabaseCW\InformationQuery\Query1.sql
SQL> SELECT Customer_Name, Customer_Category FROM CUSTOMER ORDER BY Customer_Category;

CUSTOMER_NAME          CUSTOMER_C
-----
Merle Liu               Loyal
Sidney Nelson           Loyal
Elvis Ltd               Privileged
Winter Avila            Privileged
Rubin Evans             Regular
EdRile Ltd              Regular
Akatsuki Ltd            VIP
Shelly Scott            VIP

8 rows selected.

SQL> spool off
```

*Figure 30: To list all customers according to the category.*

**Query:** SELECT Customer\_Name, Customer\_Category FROM CUSTOMER ORDER BY Customer\_Category;

The above query displays the name and customer category of a customer.

### 6.1.2. List paintings and their artist with monthly rental price and paid price:

```
SQL> spool D:\DatabaseCW\InformationQuery\Query2.sql
SQL> SELECT p.Painting_ID, a.Artist_Name, p.Monthly_Rental, p.Total_Rental FROM Painting
p JOIN Artist a ON a.Artist_ID= p.Artist_ID;
```

| PAINT | ARTIST_NAME      | MONTHLY_RENTAL | TOTAL_RENTAL |
|-------|------------------|----------------|--------------|
| P101  | Anugraha Ghale   | 1200           | 2688         |
| P102  | Roshish Shrestha | 900            | 1224         |
| P103  | Rimjhim Agrawal  | 800            | 1152         |
| P104  | Aastha Sthapit   | 1300           | 884          |
| P105  | Dawa Karki       | 780            | 6365         |
| P106  | Mihir Shrestha   | 1000           |              |
| P107  | Biji Sharma      | 950            | 760          |
| P108  | Padu Pradhan     | 1000           |              |
| P109  | Mihir Shrestha   | 1150           |              |
| P110  | Padu Pradhan     | 600            | 408          |
| P111  | Srija Pradhan    | 750            | 600          |
| P112  | Yuthika Gauchan  | 1000           | 680          |
| P113  | Yuthika Gauchan  | 860            |              |
| P114  | Roshish Shrestha | 980            | 705          |
| P115  | Mihir Shrestha   | 500            |              |

```
15 rows selected.
SQL> spool off
```

Figure 31: To list paintings and artists with monthly rental and paid price.

**Query:** SELECT p.Painting\_ID, a.Artist\_Name, p.Monthly\_Rental, p.Total\_Rental FROM Painting p JOIN Artist a ON a.Artist\_ID= p.Artist\_ID;

The above query displays the painting id and artist name along with their monthly rental and total rental.

### 6.1.3. Show total staff in Masterpieces Limited sorted by higher salary:

```
SQL> spool D:\DatabaseCW\InformationQuery\Query3.sql
SQL> SELECT * FROM Staff ORDER BY Staff_Salary DESC;

STAFF STAFF_NAME                STAFF_SALARY STAFF_CONT
-----
S107 Flynn Rider                57000 7456542185
S106 Valentin Black             50000 7456542084
S101 Kenny Stokes               45000 7456541579
S105 Julius Avila               34000 7456541983
S104 Royal Serrano              20000 7456541882
S103 Clarence Vega              17800 7456541781
S102 Nicolas Cameron            13500 7456541680

7 rows selected.

SQL> spool off
```

Figure 32: To show total staffs sorted by higher salary.

**Query:** SELECT \* FROM Staff ORDER BY Staff\_Salary DESC;

The above query displays the staff details according to the highest staff salary.

### 6.1.4. Show paintings leased before and currently by any one customer:

```
SQL> spool D:\DatabaseCW\InformationQuery\Query4.sql
SQL> SELECT c.Customer_Name, op.Painting_ID, p.Painting_Name, o.Order_Date, op.Return_Date
FROM Customer c JOIN Orders o ON c.Customer_ID=o.Customer_ID JOIN Order_Painting op ON op
.Order_ID=o.Order_ID JOIN Painting p ON p.Painting_ID=op.Painting_ID WHERE op.Buy_Or_Lease
='Lease' AND o.Customer_ID='C108';

CUSTOMER_NAME      PAINT PAINTING_NAME      ORDER_DAT RETURN_DA
-----
Shelly Scott       P101  Autumn              08-DEC-21 08-JAN-21
Shelly Scott       P102  Signs of Childhood  21-SEP-21 21-OCT-21
Shelly Scott       P102  Signs of Childhood  07-JUN-21 07-JUL-21
Shelly Scott       P104  Muddled Beggar      07-JUN-21 07-JUL-21
Shelly Scott       P110  Tainted Throne      08-DEC-21 08-JAN-21

SQL> spool off
```

Figure 33: To show paintings leased before and currently by any one customer.

**Query:** SELECT c.Customer\_Name, op.Painting\_ID, p.Painting\_Name, o.Order\_Date, op.Return\_Date FROM Customer c JOIN Orders o ON c.Customer\_ID=o.Customer\_ID

```
JOIN Order_Painting op ON op.Order_ID=o.Order_ID JOIN Painting p ON
p.Painting_ID=op.Painting_ID WHERE op.Buy_Or_Lease='Lease' AND
o.Customer_ID='C108';
```

The above query displays the leased before and currently along with the customer and painting details of one specific customer.

#### 6.1.5. List all paintings that have been returned to the owner:

```
SQL> spool D:\DatabaseCW\InformationQuery\Query5.sql
SQL> SELECT Painting_ID, Artist_ID, Painting_Name, Painting_Theme, Issue_Date FROM
Painting WHERE Availability='Returned';

PAINT  ARTIS  PAINTING_NAME                PAINTING_T  ISSUE_DAT
-----
P104   A108   Muddled Beggar                Animal      02-JUN-21
P106   A102   Lifetime Partner              Animal      03-AUG-21
P108   A103   Serene Wave                   Landscape   25-AUG-21

SQL> spool off
```

Figure 34: To list all paintings that have been returned to the owner.

**Query:** SELECT Painting\_ID, Artist\_ID, Painting\_Name, Painting\_Theme, Issue\_Date  
FROM Painting WHERE Availability='Returned';

The above query displays the list of all paintings that have been returned to the owner.

## 6.2. Transaction Query

The following transaction queries are spooled in order to create a direct output of any query to a server-side flat file. (Burleson, 2015)

### 6.2.1. List the number of paintings available for rent according to category:

```
SQL> spool D:\DatabaseCW\TransactionQuery\Query1.sql
SQL> SELECT Painting_Theme, COUNT(*) FROM Painting WHERE Availability='Available'
GROUP BY Painting_Theme;

PAINTING_T    COUNT(*)
-----
Landscape      1
Person         2

SQL> spool off
```

Figure 35: To list the number of paintings available for rent according to category.

**Query:** SELECT Painting\_Theme, COUNT (\*) FROM Painting WHERE Availability='Available' GROUP BY Painting\_Theme;

The above query displays the numbers of painting available for the customer to rent according to the painting theme.

### 6.2.2. List the details of paintings that have not been leased within three months:

```
SQL> spool D:\DatabaseCW\TransactionQuery\Query2.sql
SQL> SELECT * FROM Painting p JOIN Order_Painting op ON p.Painting_ID=op.Painting_ID WHERE sysdate - Return_Date < 90;

PAINT ARTIS PAINTING_NAME      ISSUE_DAT AVAILABILI PAINTING_T MONTHLY_RENTAL TOTAL_RENTAL SELLING_PRICE ORDER PAINT BUY_OR RETURN_DA SOLD_VALUE
-----
P101 A101 Autumn                27-MAY-21 Leased Out Landscape      1200      2688      15500 0107 P101 Lease 19-NOV-21 1200
P102 A105 Signs of Childhood    30-MAY-21 Sold Person           900      1224      11000 0105 P102 Lease 21-OCT-21 765
P107 A109 Nobody                22-AUG-21 Available Person          950      760      11400 0104 P107 Lease 15-OCT-21 950
P111 A107 Frivolous Animal      18-SEP-21 Leased Out Animal        750      600      9000 0106 P111 Lease 09-NOV-21 750
P112 A106 Familiar Night       19-OCT-21 Sold Landscape         1000      600      11000 0108 P112 Lease 01-DEC-21 850
P114 A105 Mask of Flame         13-NOV-21 Available Person          980      705      11700 0109 P114 Lease 05-DEC-21 882

6 rows selected.

SQL> spool off
```

Figure 36: To list the details of paintings that have not been leased within three months.

**Query:** SELECT \* FROM Painting p JOIN Order\_Painting op ON p.Painting\_ID=op.Painting\_ID WHERE sysdate - Return\_Date < 90;

The above query displays the details of painting that has not been leased within the range of three months.

### 6.2.3. List the details of customers who have leased the painting more than four times:

```
SQL> spool D:\DatabaseCW\TransactionQuery\Query3.sql
SQL> SELECT * FROM customer c WHERE c.Customer_ID IN (SELECT c.Customer_ID FROM Customer c JOIN Orders o ON
c.Customer_ID=o.Customer_ID JOIN Order_Painting op ON o.Order_ID=op.Order_ID WHERE op.Buy_Or_Lease='Lease' G
ROUP BY c.Customer_Id HAVING COUNT(op.Order_ID) > 4);
```

| CUSTO | CUSTOMER_NAME | CUSTOMER_ADDRESS | CUSTOMER_C | CUSTOMER_TYPE | CUSTOMER_C |
|-------|---------------|------------------|------------|---------------|------------|
| C108  | Shelly Scott  | Manor Road       | 7456542296 | Individual    | VIP        |

```
SQL> spool off
```

Figure 37: To list the details of customers who have leased the paintings more than four times.

**Query:** SELECT \* FROM customer c WHERE c.Customer\_ID IN (SELECT c.Customer\_ID FROM Customer c JOIN Orders o ON c.Customer\_ID=o.Customer\_ID JOIN Order\_Painting op ON o.Order\_ID=op.Order\_ID WHERE op.Buy\_Or\_Lease='Lease' GROUP BY c.Customer\_Id HAVING COUNT(op.Order\_ID) > 4);

The above query displays the details of customer who has leased paintings more than four times.

### 6.2.4. List top 5 paintings based on total collected rental amount:

```
SQL> spool D:\DatabaseCW\TransactionQuery\Query4.sql
SQL> SELECT * FROM (SELECT * FROM Painting ORDER BY Total_Rental DESC) WHERE ROWNUM <= 5 AND Total_Rental > 0;

PAINT ARTIS PAINTING_NAME      ISSUE_DAT AVAILABILI PAINTING_T MONTHLY_RENTAL TOTAL_RENTAL SELLING_PRICE
-----
P105  A110  Luxury of Fire                12-JUL-21 Sold      Landscape      780          6365          9360
P101  A101  Autumn                      27-MAY-21 Leased Out Landscape     1200          2688          15500
P102  A105  Signs of Childhood           30-MAY-21 Sold      Person          900          1224          11000
P103  A104  Judgment of Weakness         31-MAY-21 Leased Out Person      800          1152           9600
P104  A108  Muddled Beggar               02-JUN-21 Returned  Animal         1300           884          15600

SQL> spool off
```

Figure 38: To list the top 5 paintings based on total collected rental amount.

**Query:** SELECT \* FROM (SELECT \* FROM Painting ORDER BY Total\_Rental DESC) WHERE ROWNUM <= 5 AND Total\_Rental > 0;

The above query displays the top five paintings details according to the total collected rent amount.

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

```
SQL> spool D:\DatabaseCW\TransactionQuery\Query5.sql
SQL> SELECT op.Painting_ID, a.Artist_Name, op.Sold_Value FROM Order_Painting op JOIN
Painting p ON op.Painting_ID=p.Painting_ID JOIN Artist a ON p.Artist_ID=a.Artist_ID
JOIN Orders o ON op.Order_ID=o.Order_ID WHERE op.Buy_Or_Lease='Buy' AND (sysdate -
o.Order_Date) < 30;

PAINT ARTIST_NAME      SOLD_VALUE
-----
P109  Mihir Shrestha        12420
P113  Yuthika Gauchan        9975
P112  Yuthika Gauchan       10450

SQL> spool off
```

Figure 39: To show the name of painter and paintings sold value for the current month.

**Query:** SELECT op.Painting\_ID, a.Artist\_Name, op.Sold\_Value FROM Order\_Painting op JOIN Painting p ON op.Painting\_ID=p.Painting\_ID JOIN Artist a ON p.Artist\_ID=a.Artist\_ID JOIN Orders o ON op.Order\_ID=o.Order\_ID WHERE op.Buy\_Or\_Lease='Buy' AND (sysdate - o.Order\_Date) < 30;

The above query displays the name of an artist along with the painting id and its sold value in the current month.



## **7. Critical Evaluation.**

The critical evaluation topic is further divided into two, critically evaluating the module and critically evaluating the coursework.

### **7.1. Critical Evaluation of the Module.**

"Databases" is one of the modules that Computing students learn in Year 2. It is a semester-long curriculum with two components: Coursework and Unseen Examination, each accounting for 50% of the overall module. With the near end of the semester and this course, we are now able to comprehend and apply techniques to database analysis, design, and its development. The issues that arose while completing this project gave us a real-life scenario of how to overcome it with designing and implementing database systems.

Databases are used in practically every industry. Whether we realize it or not, databases have a significant impact on our daily lives; from banking to online movies and a variety of other services, databases play an important role. Any company needs a good database system. It has the power to assist in the management of a company, and on a smaller scale, they may assist us in improving our lifestyle. Many people feel it is only for complicated activities when, in fact, it can benefit us on a regular basis.

Overall, it is seen that this module has proved to be very useful. This course not only provided an overview of the theoretical aspects of database system design and implementation but also a practical overview of it.

## 7.2. Critical Assessment of the Coursework.

We the students had to create a database model of a Masterpieces Limited for an Internet business as part of the coursework. The coursework's scenario was challenging and intricate. The scenario initially caused a lot of uncertainty but with regular consultations with the tutor helped to clear up these misunderstandings.

Entities and their properties were separated after a thorough understanding of the scenario. The normalization was then carried out depending on the assumptions stated. During the normalization process, there were numerous misunderstandings and blunders. With the help of the tutors and the lecture and tutorial slides that were provided to us, we were able to clear up these misunderstandings. The tutors assisted in the correction of the inaccuracies. The final ER diagram was then created using the 3NF result obtained during the normalization phase.

Data insertion into the table was another puzzling task because there were various considerations to be made. For the queries to be accurate, the entered data had to be error-free. To insert the data, a sample list of data was first prepared, and then the data was solely inserted into the SQL.

Creating a database design and implementing it was an interesting project yet proved to be very challenging but with much effort, I was able to solve this problem and I am sure this experience will make my future attempt much smoother.

## 8. Dump File Creation.

For backup and recovery, data dumps (.dmp files) are created. Instead of using them as-is, it is re-imported into the Oracle DBMS.

```
C:\Users\Rabina>exp MasterpiecesLimited/RabinaShrestha file=20049416 Rabina Shrestha.dmp

Export: Release 11.2.0.2.0 - Production on Tue Dec 28 02:23:25 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          10 rows exported
. . exporting table          CUSTOMER         8 rows exported
. . exporting table          ORDERS           12 rows exported
. . exporting table          ORDER_PAINTING    18 rows exported
. . exporting table          PAINTING          15 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 40: Dump File Creation.

## 9. Bibliography

Artwork Archive, 2010. *ARTWORK ARCHIVE*. [Online]  
Available at: <https://www.artworkarchive.com/blog/is-selling-art-online-for-you>  
[Accessed 20 December 2021].

Berrington, J., 2017. *ScienceDirect*. [Online]  
Available at: <https://www.sciencedirect.com/science/article/abs/pii/S1472029916302181>  
[Accessed 20 December 2021].

Burleson, D., 2015. *Burleson Consulting*. [Online]  
Available at: [http://www.dba-oracle.com/t\\_sqlplus\\_spool.htm](http://www.dba-oracle.com/t_sqlplus_spool.htm)  
[Accessed 21 December 2021].

Datanamic, 2016. *Datanamic*. [Online]  
Available at: <https://www.datanamic.com/data-modeling/many-to-many-relationships/>  
[Accessed 24 December 2021].

Elmasri, R. & Navathe, S. B., 2016. *Fundamentals of Database Systems*. 7th ed. Texas: Pearson.

Enterprise Integration, 2020. *Enterprise Integration*. [Online]  
Available at: <https://entint.com/it-support-services/infrastructure-services/database-implementation/#:~:text=working%20with%20EI%3F-.Q%3A%20What%20is%20database%20implementation%3F,process%20involves%20training%20the%20users.>  
[Accessed 21 December 2021].

Gibbs, M., 2021. *Study.com*. [Online]  
Available at: <https://study.com/academy/lesson/database-query-definition-tools.html>  
[Accessed 21 December 2021].

Lawsons, R., 2012. *Ryanhmfrc*. [Online]  
Available at: <https://ryanhmfrc.wordpress.com/2012/09/13/normalisation-unf-1nf-2nf-3nf/>  
[Accessed 22 December 2021].

Oracle, 2021. *ORACLE Help Center*. [Online]  
Available at: [https://docs.oracle.com/cd/B19306\\_01/server.102/b14357/qstart.htm](https://docs.oracle.com/cd/B19306_01/server.102/b14357/qstart.htm)  
[Accessed 20 December 2021].

Phillips, R., 2020. *Renee Phillips The Artrepreneur Coach*. [Online]  
Available at: <https://renee-phillips.com/10-of-the-best-sales-tips-for-artists/#comments>  
[Accessed 20 December 2021].

Qui, M., Chen, L., Woitasen, D. & Xu, S., 2021. *Microsoft*. [Online]  
Available at: <https://docs.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description>  
[Accessed 22 December 2021].

Sharma, S., 2020. *tutorialspoint*. [Online]  
Available at: <https://www.tutorialspoint.com/primary-key-vs-unique-key>  
[Accessed 20 December 2021].

Singh, C., 2016. *BeginnersBook*. [Online]  
Available at: <https://beginnersbook.com/2015/04/e-r-model-in-dbms/>  
[Accessed 24 December 2021].

Soane, J., 1837. *The Picture Room*. [Art] (Sir John Soane's Museum London).

Subramaniam, R. et al., 2021. Sustainability. *Positive and Negative Impacts of COVID-19 in Digital Transformation*, 13(16), p. 18.