Database Systems

Art Gallery Management Database

By Damian Barrous-Dume

Table of Contents

Abstract

The project will be creating a database for a fictional Art Gallery that they can use to manage all of the artwork they have in their system as well as those customers who purchase the artwork. For example, there would be an Artwork Table that would contain all of the artwork that the Gallery has owned. The Artwork table would have an Artist Table that would come before it and a Customers Table after. All of these tables would contain information (depending on which table of course) about the artist, artwork, and customer who purchased the artwork if any. The project would be a good example on how a database could improve the organization of a business such as an Art Gallery.

Objective: To create a database management system to be used by Art Galleries so that they can manage important information that pertains to them including Employees, Customers, Artwork and the Artists.

Data Types

The data types used on each of the entities attributes are as followed:

Integer (INT) - Can contain any whole number ranging from -2147483648 to 2147483647.

Decimal (FLOAT) - Can contain any decimal number precisely to 23 digits.

String (VARCHAR) - Can contain any alphanumeric string from a length of 0 - 255 characters/numbers.

String (TEXT) - Can contain any string from a length of 0 - 65535 characters.

Calendar Date (DATE) - Contains a calendar date in the format of YYYY-MM-DD.

Data Requirements

The database for the gallery will contain 5 entities:

- Gallery
- Artwork
- Artists
- Customers
- Employees

The Attributes to each Entity are as followed:

Gallery :

- Gallery_ID (Primary Key, INT)
- Gallery_Name (VARCHAR)
- Gallery_Phone (VARCHAR)
- Gallery_Street (VARCHAR)
- Gallery_Zip_Code (INT)

Artwork :

- Artwork_ID (Primary Key, INT)

- Artwork_Name (VARCHAR)
- Artwork_Genre (VARCHAR)
- Artwork_Descrip (TEXT)
- Artwork_Price (FLOAT)
- Artist_ID (Foreign Key, INT)
- Customer_ID (Foreign Key, INT)
- Gallery_ID (Foreign Key, INT)

Artist :

- Artist_ID (Primary Key, INT)
- Artist_firstName (VARCHAR)
- Artist_lastName (VARCHAR)
- Artist_Biography (TEXT)
- Artist_Phone (VARCHAR)

Customer :

- Customer_ID (Primary Key, INT)
- Cutomer_firstName (VARCHAR)
- Customer_lastName (VARCHAR)

- Customer_Phone (VARCHAR)

Employee :

- Employee_ID (Primary Key, INT)
- Employee_firstName (VARCHAR)
- Employee_lastName (VARCHAR)
- Job_Title (VARCHAR)
- Employee_StartingDate (DATE)
- Gallery_ID (Foreign Key, INT)
- Employee_Street (VARCHAR)
- Employee_Zip_Code (INT)
- Employee_Salary (INT)
- Employee_Managed_By (INT)

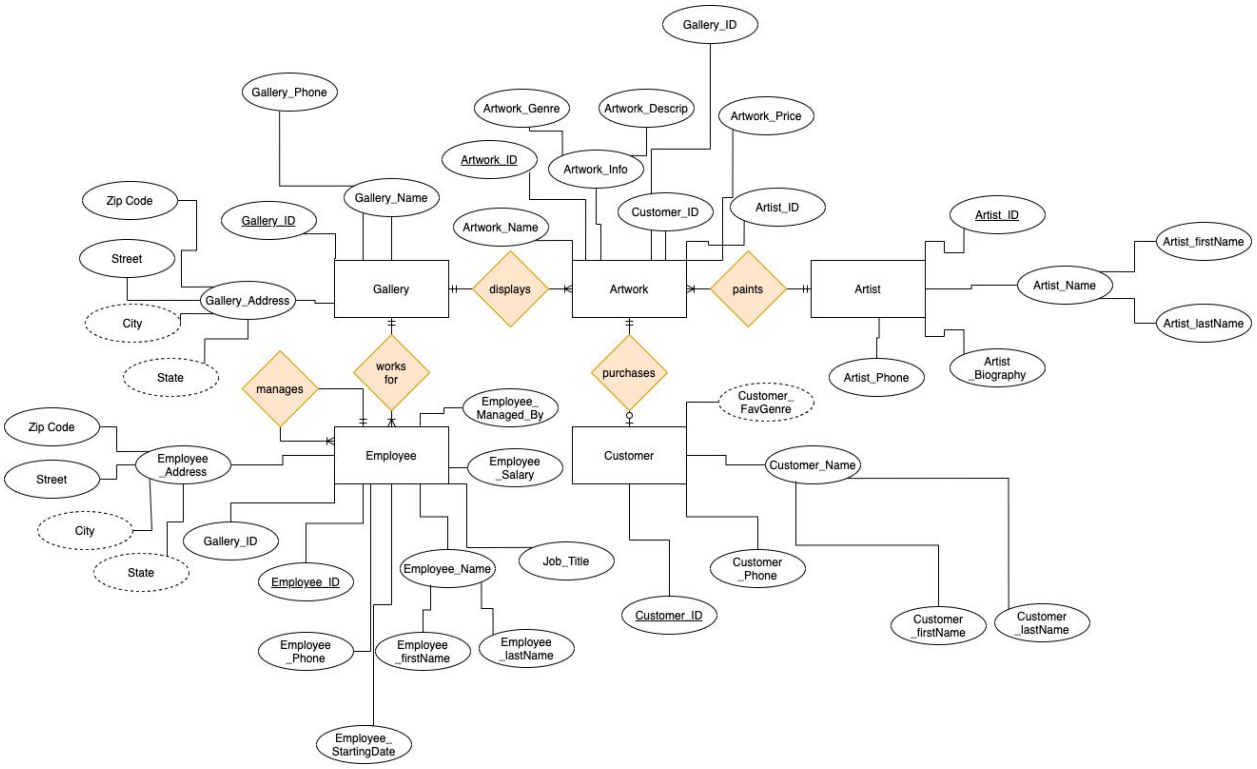
Reasoning for the Entities

My reasoning for choosing these 5 entities is because they are, they represent the most important real-life objects that would represent an Art Gallery. For example, Gallery is required because it contains all the Artwork and hires all the Employees. Artwork is required because without it there would not be an Art Gallery. Artists are required to paint the Artwork. Customers are required to purchase the Artwork and keep the Gallery in business. Each Entity has their own set of attributes.

Here are some of the Business Rules and Cardinality for the Art Gallery Database:

- One and only one Gallery displays at least one to many Artwork (1:M)
- At least one to many Employees works for one and only one Gallery (M:1)
- One and only one Artwork may have zero to one Customers (1:1)
- One and only one Artists may have at least one to many Artwork (1:M)

Entity Relation Diagram



Schema Diagram

Creating Database using MySQL

Creation & Use Of Database:

```
CREATE DATABASE ART_GALLERY_MGMT;
USE ART_GALLERY_MGMT;
```

Creation of Gallery Table:

```
CREATE TABLE Gallery (
    Gallery_ID INT(5) AUTO_INCREMENT NOT NULL,
    Gallery_Name VARCHAR(255) NOT NULL,
    Gallery_Phone VARCHAR(12) NOT NULL,
    Gallery_Street VARCHAR(95) NOT NULL,
    Gallery_Zip_Code INT(9) NOT NULL,
    PRIMARY KEY (Gallery_ID)
);
```

Creation of Artist Table:

```
CREATE TABLE Artist (
    Artist_ID INT(10) AUTO_INCREMENT NOT NULL,
    Artist_firstName VARCHAR(30) NOT NULL,
    Artist_lastName VARCHAR(30) NOT NULL,
    Artist_Biography TEXT,
    Artist_Phone VARCHAR(12),
    PRIMARY KEY (Artist_ID)
);
```

Creation of Customer Table:

```
CREATE TABLE Customer (
    Customer_ID INT(10) AUTO_INCREMENT NOT NULL,
    Customer_firstName VARCHAR(30) NOT NULL,
    Customer_lastName VARCHAR(30) NOT NULL,
    Customer_Phone VARCHAR(12) NOT NULL,
    PRIMARY KEY (Customer_ID)
);
```

Creation of Employee Table:

```
CREATE TABLE Employee (
    Employee_ID INT(10) AUTO_INCREMENT NOT NULL,
    Employee_firstName VARCHAR(30) NOT NULL,
    Employee_lastName VARCHAR(30) NOT NULL,
    Job_Title VARCHAR(30) NOT NULL,
    Employee_StartingDate DATE NOT NULL,
    Gallery_ID INT(5) NOT NULL,
    Employee_Street VARCHAR(95) NOT NULL,
    Employee_Zip_Code INT(9) NOT NULL,
    Employee_Salary INT(12) NOT NULL,
    Employee_Managed_By INT(10) NOT NULL,
    PRIMARY KEY (Employee_ID),
    FOREIGN KEY (Gallery_ID) REFERENCES
Gallery(Gallery_ID)
);
```

Creation of Artwork Table:

```
CREATE TABLE Artwork (
    Artwork_ID INT(5) AUTO_INCREMENT,
    Artwork_Name VARCHAR(255) NOT NULL, Artwork_Genre
VARCHAR(20) NOT NULL,
    Artwork_Descrip TEXT NOT NULL,
    Artwork_Price Float(25) NOT NULL,
    Artist_ID INT(10) NOT NULL,
    Customer_ID INT(10),
    Gallery_ID INT(5),
    PRIMARY KEY (Artwork_ID),
    FOREIGN KEY (Artist_ID) REFERENCES
Artist(Artist_ID),
    FOREIGN KEY (Customer_ID) REFERENCES
Customer(Customer_ID),
    FOREIGN KEY (Gallery_ID) REFERENCES
Gallery(Gallery_ID)
);
```

Test Queries

1. **List all of the Galleries in the Database**

   Command:

   ```
   Select * FROM Gallery;
   ```

```
mysql> SELECT * FROM Gallery;
+------------+--------------------+--------------+--------------------+------------------+
| Gallery_ID | Gallery_Name       | Gallery_Phone | Gallery_Street     | Gallery_Zip_Code |
+------------+--------------------+--------------+--------------------+------------------+
|          1 | Alter Salt Gallery | 9783235421    | 12 Vangough St.    |            34230 |
|          2 | EGO LLC.           | 6173234421    | 12 Binge St.       |            89220 |
|          3 | Killer I           | 8192347865    | 75 Queen Ave.      |            90380 |
|          4 | Banquet Site       | 9012348953    | 43 Bingo St.       |            90380 |
|          5 | Bounce Art Central | 9782346748    | 73 Queen Ave.      |            90380 |
|          6 | Puncture Row       | 9782347865    | 12 Row St.         |            90385 |
|          7 | Starving Art       | 3192347335    | 1220 15th NW Ave.  |            90002 |
|          8 | GO OUTSIDE ART     | 3948710840    | 121 17th NE Ave.   |            90380 |
|          9 | THIS is Art        | 9782128213    | 23 Artist Ln.      |            90332 |
+------------+--------------------+--------------+--------------------+------------------+
9 rows in set (0.00 sec)
```

2. **List all the Artists with their Artwork**

   Command:

   ```
   SELECT Artist_firstName,Artwork_Name FROM Artwork
   LEFT JOIN Artist ON Artwork.Artist_ID =
   Artist.Artist_ID;
   ```

```
mysql> SELECT Artist.Artist_firstName,Artist.Artist_lastName,Artwork.Artwork_Name FROM
Artwork LEFT JOIN Artist ON Artwork.Artist_ID = Artist.Artist_ID;
+-----------------+----------------+--------------+
| Artist_firstName | Artist_lastName | Artwork_Name |
+-----------------+----------------+--------------+
| Aleem           | Olsson         | Untitled     |
| Aleem           | Olsson         | Upended      |
| Johanna         | Thompson       | HERO         |
| Michael         | Fountain       | Fingers      |
| Dorthy          | Smith          | Evian        |
| Michael         | Fountain       | IRON         |
| Julia           | Kurz           | Ubuntu       |
| Paul            | Baatz          | Ebay         |
| Nikos           | Kanarelis      | Judy         |
| Nikos           | Kanarelis      | Paper        |
| Nikos           | Kanarelis      | Chanting     |
| Nikos           | Kanarelis      | LOS          |
+-----------------+----------------+--------------+
12 rows in set (0.00 sec)
```

3. **List all the Customers with their Artwork Purchases**

   Command:

   SELECT Customer.Customer_firstName,Customer.Customer_lastName, Artwork.Artwork_Name,Artwork.Artwork_Price FROM Customer INNER JOIN Artwork ON Customer.Customer_ID = Artwork.Customer_ID ;

```
mysql> SELECT Customer.Customer_firstName,Customer.Customer_lastName,Artwork.Artwork
_Name,Artwork.Artwork_Price FROM Customer INNER JOIN Artwork ON Customer.Customer_ID
 = Artwork.Customer_ID ;
+--------------------+-------------------+--------------+---------------+
| Customer_firstName | Customer_lastName | Artwork_Name | Artwork_Price |
+--------------------+-------------------+--------------+---------------+
| Simona             | Medina            | Upended      |           899 |
| Leoni              | Kumar             | HERO         |         20.99 |
| Leoni              | Kumar             | Fingers      |         89.99 |
| Fionnuala          | Camacho           | Judy         |          1000 |
| Fionnuala          | Camacho           | Paper        |        999.99 |
| Fionnuala          | Camacho           | Chanting     |        999.99 |
+--------------------+-------------------+--------------+---------------+
6 rows in set (0.00 sec)
```

4. **List all Employees by their Last Name**

   Command:

   SELECT Employee_lastName FROM Employee;

```
mysql> SELECT Employee_lastName FROM Employee;
+-------------------+
| Employee_lastName |
+-------------------+
| Yang              |
| Mendoza           |
| Brandt            |
| Brant             |
| Fields            |
| Rigby             |
| Gosh              |
+-------------------+
7 rows in set (0.00 sec)
```

5. **List all Galleries Names with their Artwork Names**

Command:

```
SELECT Gallery.Gallery_Name, Artwork.Artwork_Name FROM
Gallery INNER JOIN Artwork ON Gallery.Gallery_ID =
Artwork.Gallery_ID;
```

```
mysql> SELECT Gallery.Gallery_Name, Artwork.Artwork_Name FROM Gallery INNER JOIN Artwork
 ON Gallery.Gallery_ID = Artwork.Gallery_ID;
+--------------------+--------------+
| Gallery_Name       | Artwork_Name |
+--------------------+--------------+
| Alter Salt Gallery | Untitled     |
| Alter Salt Gallery | Upended      |
| Alter Salt Gallery | HERO         |
| Alter Salt Gallery | Fingers      |
| Alter Salt Gallery | Evian        |
| Alter Salt Gallery | IRON         |
| Alter Salt Gallery | Ubuntu       |
| Alter Salt Gallery | Ebay         |
| EGO LLC.           | Judy         |
| EGO LLC.           | Paper        |
| EGO LLC.           | Chanting     |
| Alter Salt Gallery | LOS          |
+--------------------+--------------+
12 rows in set (0.00 sec)
```

6. **List all Artwork with the name of the Artists**

Command:

```
SELECT Artwork.Artwork_Name, Artist.Artist_firstName,
Artist.Artist_lastName FROM Artwork INNER JOIN Artist
ON Artist.Artist_ID = Artwork.Artist_ID;
```

```
mysql> SELECT Artwork.Artwork_Name, Artist.Artist_firstName, Artist.Artist_lastName FROM
 Artwork INNER JOIN Artist ON Artist.Artist_ID = Artwork.Artist_ID;
+--------------+-----------------+----------------+
| Artwork_Name | Artist_firstName | Artist_lastName |
+--------------+-----------------+----------------+
| Untitled     | Aleem           | Olsson         |
| Upended      | Aleem           | Olsson         |
| HERO         | Johanna         | Thompson       |
| Fingers      | Michael         | Fountain       |
| Evian        | Dorthy          | Smith          |
| IRON         | Michael         | Fountain       |
| Ubuntu       | Julia           | Kurz           |
| Ebay         | Paul            | Baatz          |
| Judy         | Nikos           | Kanarelis      |
| Paper        | Nikos           | Kanarelis      |
| Chanting     | Nikos           | Kanarelis      |
| LOS          | Nikos           | Kanarelis      |
+--------------+-----------------+----------------+
12 rows in set (0.00 sec)
```

7. **Group the Art Pieces by Genre** (Extra Query)
   Command:

```
SELECT COUNT(Artwork_ID) AS Number_Of_Art_Pieces,
Artwork_Genre FROM Artwork GROUP BY Artwork_Genre;
```

```
mysql> SELECT COUNT(Artwork_ID) AS Number_Of_Art_Pieces, Artwork_Genre FROM Artwork
GROUP BY Artwork_Genre;
+----------------------+---------------+
| Number_Of_Art_Pieces | Artwork_Genre |
+----------------------+---------------+
|                    7 | Painting      |
|                    5 | Sculpture     |
+----------------------+---------------+
2 rows in set (0.00 sec)
```

8. **Show Preferred Genre of Art made by All of the Artists Together** (Extra Query)
   Command:

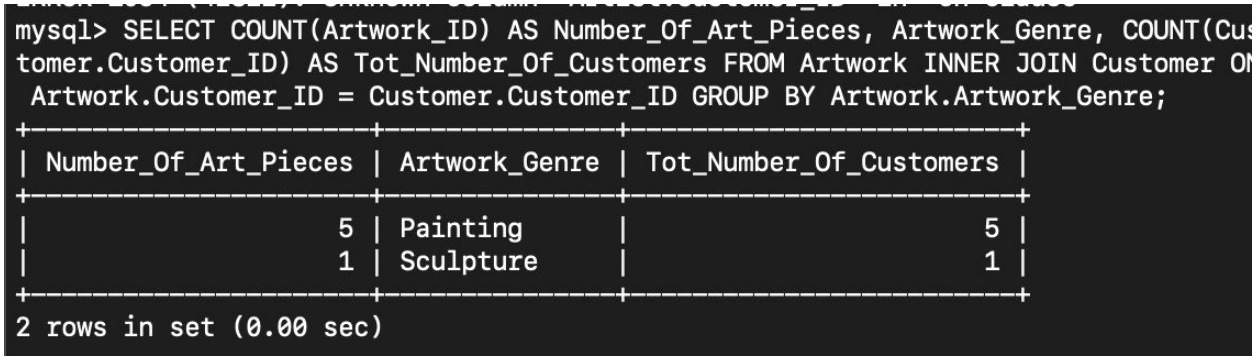
```
SELECT COUNT(Artwork_ID) AS Number_Of_Art_Pieces,
Artwork_Genre, COUNT(Artist.Artist_ID) AS
Tot_Number_Of_Artists FROM Artwork INNER JOIN Artist ON
Artwork.Artist_ID = Artist.Artist_ID GROUP BY
Artwork.Artwork_Genre;
```

```
mysql> SELECT COUNT(Artwork_ID) AS Number_Of_Art_Pieces, Artwork_Genre,
COUNT(Artist.Artist_ID) AS Tot_Number_Of_Artists FROM Artwork INNER JOIN
 Artist ON Artwork.Artist_ID = Artist.Artist_ID GROUP BY Artwork.Artwork
_Genre;
+----------------------+---------------+-----------------------+
| Number_Of_Art_Pieces | Artwork_Genre | Tot_Number_Of_Artists |
+----------------------+---------------+-----------------------+
|                    7 | Painting      |                     7 |
|                    5 | Sculpture     |                     5 |
+----------------------+---------------+-----------------------+
2 rows in set (0.00 sec)
```

9. **Show Preferred Genre of Art purchased by all of the Customers** (Extra Query)

   Command:

```
SELECT COUNT(Artwork_ID) AS Number_Of_Art_Pieces,
Artwork_Genre, COUNT(Customer.Customer_ID) AS
Tot_Number_Of_Customers FROM Artwork INNER JOIN
Customer ON Artwork.Customer_ID = Customer.Customer_ID
GROUP BY Artwork.Artwork_Genre;
```



```
mysql> SELECT COUNT(Artwork_ID) AS Number_Of_Art_Pieces, Artwork_Genre, COUNT(Cus
tomer.Customer_ID) AS Tot_Number_Of_Customers FROM Artwork INNER JOIN Customer ON
 Artwork.Customer_ID = Customer.Customer_ID GROUP BY Artwork.Artwork_Genre;
+----------------------+---------------+--------------------------+
| Number_Of_Art_Pieces | Artwork_Genre | Tot_Number_Of_Customers |
+----------------------+---------------+--------------------------+
|                    5 | Painting      |                        5 |
|                    1 | Sculpture     |                        1 |
+----------------------+---------------+--------------------------+
2 rows in set (0.00 sec)
```

Conclusion

- The MySQL Database to manage the Art Galleries were successfully executed after extensive planning.
- It can be seen that the extensive planning did ensure that the Database System is actually useful and can be used to extract a lot of helpful and useful information.
- The Test Queries assigned to the project all worked successfully and a few more extra test queries were added to show the information that can be extracted from the Database even with the limited amount of Data actually in the Database. (It contains about 7-13 Entries in each Table).
- The Section of Database Creation can be followed to create the same system on another machine or the script bundled with this project can be executed to automatically create the tables. (File name : ArtGallery_TableCreation.sql)
- The Test Entries can also be inserted automatically using the second script bundled with this project. (File name : ArtGallery_Data_Insertion.sql)
- Lastly this project was overall a success because it successfully created a system to help a group of Art Galleries to keep track of the Employees of each gallery and those who purchased (Customers) or created (Artists) the Artwork.

References

- **MySQL Documentation** - https://dev.mysql.com/doc/
- **SQL section of W3Schools** - https://www.w3schools.com/sql/
- **Fundamentals of Database Systems 7th Edition** (Book)