**Virtual Art Gallery Shema DDL and DML**

mysql> show databases;

+--------------------+

| Database |

+--------------------+

| Banking\_system |

| HMBank |

| information\_schema |

| mysql |

| performance\_schema |

| practice |

| ProjectManagement |

| sys |

| testdb |

+--------------------+

9 rows in set (0.03 sec)

mysql> create database VirtualArtGallery;

Query OK, 1 row affected (0.01 sec)

mysql> use VirtualArtGallery;

Database changed

mysql> CREATE TABLE Artists (

-> ArtistID INT PRIMARY KEY,

-> Name VARCHAR(255) NOT NULL,

-> Biography TEXT,

-> Nationality VARCHAR(100));

Query OK, 0 rows affected (0.01 sec)

mysql> desc Artists;

A screenshot of a computer

Description automatically generated

mysql> CREATE TABLE Categories (

-> CategoryID INT PRIMARY KEY,

-> Name VARCHAR(100) NOT NULL);

Query OK, 0 rows affected (0.01 sec)

mysql> desc Categories;

A screenshot of a computer

Description automatically generated

mysql> CREATE TABLE Artworks (

-> ArtworkID INT PRIMARY KEY,

-> Title VARCHAR(255) NOT NULL,

-> ArtistID INT,

-> CategoryID INT,

-> Year INT,

-> Description TEXT,

-> ImageURL VARCHAR(255),

-> FOREIGN KEY (ArtistID) REFERENCES Artists (ArtistID),

-> FOREIGN KEY (CategoryID) REFERENCES Categories (CategoryID));

Query OK, 0 rows affected (0.01 sec)

mysql> desc Artworks;

A screenshot of a computer

Description automatically generated

mysql> CREATE TABLE Exhibitions (

-> ExhibitionID INT PRIMARY KEY,

-> Title VARCHAR(255) NOT NULL,

-> StartDate DATE,

-> EndDate DATE,

-> Description TEXT);

Query OK, 0 rows affected (0.01 sec)

mysql> desc Exhibitions;

A screenshot of a computer

Description automatically generated

mysql> CREATE TABLE ExhibitionArtworks (

-> ExhibitionID INT,

-> ArtworkID INT,

-> PRIMARY KEY (ExhibitionID, ArtworkID),

-> FOREIGN KEY (ExhibitionID) REFERENCES Exhibitions (ExhibitionID),

-> FOREIGN KEY (ArtworkID) REFERENCES Artworks (ArtworkID));

Query OK, 0 rows affected (0.01 sec)

mysql> desc ExhibitionArtworks;

A screenshot of a computer

Description automatically generated

mysql> INSERT INTO Artists (ArtistID, Name, Biography, Nationality) VALUES

-> (1, 'Pablo Picasso', 'Renowned Spanish painter and sculptor.', 'Spanish'),

-> (2, 'Vincent van Gogh', 'Dutch post-impressionist painter.', 'Dutch'),

-> (3, 'Leonardo da Vinci', 'Italian polymath of the Renaissance.', 'Italian'),

-> (4, 'Michelangelo', 'Italian sculptor and painter.', 'Italian'),

-> (5, 'Andy Warhol', 'American pop artist.', 'American');

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> select \* from Artists;

A screenshot of a computer

Description automatically generated

mysql> INSERT INTO Categories (CategoryID, Name) VALUES

-> (1, 'Painting'),

-> (2, 'Sculpture'),

-> (3, 'Photography');

Query OK, 3 rows affected (0.01 sec)

Records: 3 Duplicates: 0 Warnings: 0

mysql> select \* from Categories;

A screenshot of a computer

Description automatically generated

mysql> INSERT INTO Artworks (ArtworkID, Title, ArtistID, CategoryID, Year, Description, ImageURL) VALUES

-> (1, 'Starry Night', 2, 1, 1889, 'A famous painting by Vincent van Gogh.', 'starry\_night.jpg'),

-> (2, 'Mona Lisa', 3, 1, 1503, 'The iconic portrait by Leonardo da Vinci.', 'mona\_lisa.jpg'),

-> (3, 'Guernica', 1, 1, 1937, 'Pablo Picasso\'s powerful anti-war mural.', 'guernica.jpg'),

-> (4, 'The Creation of Adam', 4, 1, 1512, 'Michelangelo’s famous fresco.', 'creation\_adam.jpg'),

-> (5, 'Campbell’s Soup Cans', 5, 1, 1962, 'Pop art masterpiece by Warhol.', 'soup\_cans.jpg'),

-> (6, 'David', 4, 2, 1504, 'Michelangelo’s marble statue.', 'david.jpg'),

-> (7, 'The Thinker', 4, 2, 1904, 'Famous sculpture by Rodin.', 'thinker.jpg'),

-> (8, 'The Last Supper', 3, 1, 1495, 'Leonardo’s large mural painting.', 'last\_supper.jpg'),

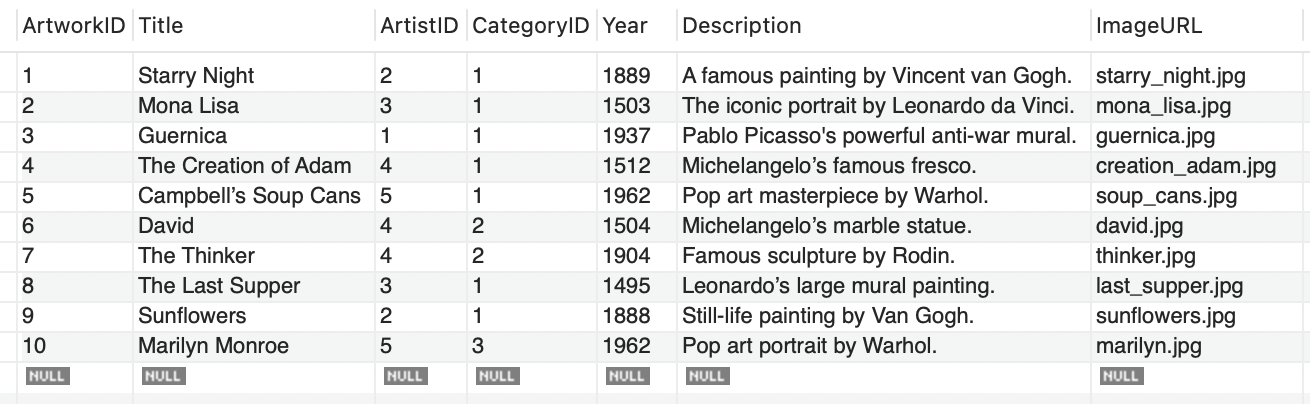
-> (9, 'Sunflowers', 2, 1, 1888, 'Still-life painting by Van Gogh.', 'sunflowers.jpg'),

-> (10, 'Marilyn Monroe', 5, 3, 1962, 'Pop art portrait by Warhol.', 'marilyn.jpg');

Query OK, 10 rows affected (0.01 sec)

Records: 10 Duplicates: 0 Warnings: 0

mysql> select \* from Artworks;



mysql> INSERT INTO Exhibitions (ExhibitionID, Title, StartDate, EndDate, Description) VALUES

-> (1, 'Modern Art Masterpieces', '2023-01-01', '2023-03-01', 'A collection of modern art masterpieces.'),

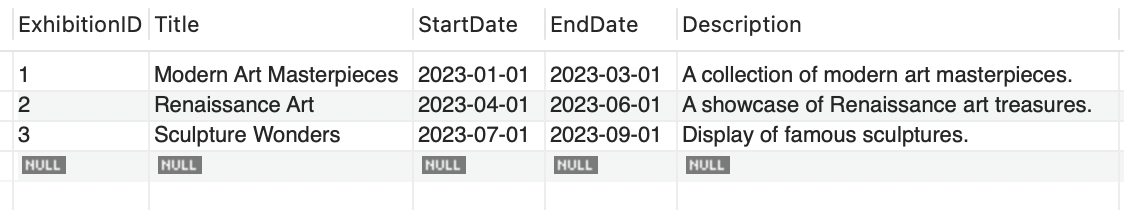
-> (2, 'Renaissance Art', '2023-04-01', '2023-06-01', 'A showcase of Renaissance art treasures.'),

-> (3, 'Sculpture Wonders', '2023-07-01', '2023-09-01', 'Display of famous sculptures.');

Query OK, 3 rows affected (0.00 sec)

Records: 3 Duplicates: 0 Warnings: 0

mysql> select \* from Exhibitions;



mysql> INSERT INTO ExhibitionArtworks (ExhibitionID, ArtworkID) VALUES

-> (1, 1),

-> (1, 2),

-> (1, 3),

-> (1, 5),

-> (1, 10),

-> (2, 2),

-> (2, 4),

-> (2, 8),

-> (3, 6),

-> (3, 7);

Query OK, 10 rows affected (0.00 sec)

Records: 10 Duplicates: 0 Warnings: 0

mysql> select \* from ExhibitionArtworks;

A table with numbers on it

Description automatically generated

1. Retrieve the names of all artists along with the number of artworks they have in the gallery, and list them in descending order of the number of artworks.

mysql> SELECT a.Name, COUNT(b.ArtworkID) AS ArtworkCount

-> FROM Artists a

-> LEFT JOIN Artworks b ON a.ArtistID = b.ArtistID

-> GROUP BY a.Name

-> ORDER BY ArtworkCount DESC;

A screenshot of a computer

Description automatically generated

2.List the titles of artworks created by artists from 'Spanish' and 'Dutch' nationalities, and order them by the year in ascending order.

mysql> SELECT a.Title, a.Year, b.Name, b.Nationality

-> FROM Artworks a

-> JOIN Artists b ON a.ArtistID = b.ArtistID

-> WHERE b.Nationality IN ('Spanish', 'Dutch')

-> ORDER BY a.Year ASC;

A screenshot of a phone

Description automatically generated

3. Find the names of all artists who have artworks in the 'Painting' category, and the number of artworks they have in this category.

mysql> SELECT a.Name, COUNT(ar.ArtworkID) AS PaintingCount

-> FROM Artists a

-> JOIN Artworks ar ON a.ArtistID = ar.ArtistID

-> JOIN Categories c ON ar.CategoryID = c.CategoryID

-> WHERE c.Name = 'Painting'

-> GROUP BY a.Name

-> ORDER BY PaintingCount DESC;

A screenshot of a list of paintings

Description automatically generated

4. List the names of artworks from the 'Modern Art Masterpieces' exhibition, along with their artists and categories.

mysql> SELECT ar. Title, a.Name AS Artist, c.Name AS Category

-> FROM Artworks ar

-> JOIN ExhibitionArtworks ea ON ar.ArtworkID = ea.ArtworkID

-> JOIN Exhibitions e ON ea.ExhibitionID = e.ExhibitionID

-> JOIN Artists a ON ar.ArtistID = a.ArtistID

-> JOIN Categories c ON ar.CategoryID = c.CategoryID

-> WHERE e.Title = 'Modern Art Masterpieces';

A screenshot of a computer

Description automatically generated

5. Find the artists who have more than two artworks in the gallery.

mysql> SELECT a.Name, COUNT(ar.ArtworkID) AS ArtworkCount

-> FROM Artists a

-> JOIN Artworks ar ON a.ArtistID = ar.ArtistID

-> GROUP BY a.Name

-> HAVING COUNT(ar.ArtworkID) > 2;

A screenshot of a phone

Description automatically generated

6. Find the titles of artworks that were exhibited in both 'Modern Art Masterpieces' and 'Renaissance Art' exhibitions.

mysql> SELECT ar.Title

-> FROM Artworks ar

-> JOIN ExhibitionArtworks ea ON ar.ArtworkID = ea.ArtworkID

-> JOIN Exhibitions e ON ea.ExhibitionID = e.ExhibitionID

-> WHERE e.Title IN ('Modern Art Masterpieces', 'Renaissance Art')

-> GROUP BY ar.Title

-> HAVING COUNT(DISTINCT e.Title) = 2;

A screenshot of a phone

Description automatically generated

7. Find the total number of artworks in each category.

mysql> SELECT c.Name AS Category, COUNT(ar.ArtworkID) AS TotalArtworks

-> FROM Categories c

-> LEFT JOIN Artworks ar ON c.CategoryID = ar.CategoryID

-> GROUP BY c.Name;

A screenshot of a computer

Description automatically generated

8. List artists who have more than 3 artworks in the gallery.

mysql> SELECT a.Name, COUNT(ar.ArtworkID) AS ArtworkCount

-> FROM Artists a

-> JOIN Artworks ar ON a.ArtistID = ar.ArtistID

-> GROUP BY a.Name

-> HAVING COUNT(ar.ArtworkID) > 3;

Empty set (0.00 sec)

A screenshot of a computer

Description automatically generated

9. Find the artworks created by artists from a specific nationality (e.g., Spanish).

mysql> SELECT ar.Title, a.Name, a.Nationality

-> FROM Artworks ar

-> JOIN Artists a ON ar.ArtistID = a.ArtistID

-> WHERE a.Nationality = 'Spanish';

A screenshot of a phone

Description automatically generated

10. List exhibitions that feature artwork by both Vincent van Gogh and Leonardo da Vinci.

mysql> SELECT e.Title

-> FROM Exhibitions e

-> JOIN ExhibitionArtworks ea ON e.ExhibitionID = ea.ExhibitionID

-> JOIN Artworks ar ON ea.ArtworkID = ar.ArtworkID

-> JOIN Artists a ON ar.ArtistID = a.ArtistID

-> WHERE a.Name IN ('Vincent van Gogh', 'Leonardo da Vinci')

-> GROUP BY e.Title

-> HAVING COUNT(DISTINCT a.Name) = 2;

A close-up of a screen

Description automatically generated

11. Find all the artworks that have not been included in any exhibition.

mysql> SELECT ar.Title

-> FROM Artworks ar

-> LEFT JOIN ExhibitionArtworks ea ON ar.ArtworkID = ea.ArtworkID

-> WHERE ea.ExhibitionID IS NULL;

A screenshot of a computer

Description automatically generated

12. List artists who have created artworks in all available categories.

mysql> SELECT a.Name

-> FROM Artists a

-> JOIN Artworks ar ON a.ArtistID = ar.ArtistID

-> JOIN Categories c ON ar.CategoryID = c.CategoryID

-> GROUP BY a.Name

-> HAVING COUNT(DISTINCT ar.CategoryID) = (SELECT COUNT(\*) FROM Categories);

Empty set (0.02 sec)

A screenshot of a cell phone

Description automatically generated

13. List the total number of artworks in each category.

mysql> SELECT c.Name AS Category, COUNT(ar.ArtworkID) AS TotalArtworks

-> FROM Categories c

-> LEFT JOIN Artworks ar ON c.CategoryID = ar.CategoryID

-> GROUP BY c.Name;

A screenshot of a computer

Description automatically generated

14. Find the artists who have more than 2 artworks in the gallery.

mysql> SELECT a.Name, COUNT(ar.ArtworkID) AS ArtworkCount

-> FROM Artists a

-> JOIN Artworks ar ON a.ArtistID = ar.ArtistID

-> GROUP BY a.Name

-> HAVING COUNT(ar.ArtworkID) > 2;

A screenshot of a phone

Description automatically generated

15. List the categories with the average year of artworks they contain, only for categories with more than 1 artwork.

mysql> SELECT c.Name AS Category, AVG(ar.Year) AS AvgYear

-> FROM Categories c

-> JOIN Artworks ar ON c.CategoryID = ar.CategoryID

-> GROUP BY c.Name

-> HAVING COUNT(ar.ArtworkID) > 1;

A screenshot of a calculator

Description automatically generated

16. Find the artworks that were exhibited in the 'Modern Art Masterpieces' exhibition.

mysql> SELECT ar.Title

-> FROM Artworks ar

-> JOIN ExhibitionArtworks ea ON ar.ArtworkID = ea.ArtworkID

-> JOIN Exhibitions e ON ea.ExhibitionID = e.ExhibitionID

-> WHERE e.Title = 'Modern Art Masterpieces';

A screenshot of a list

Description automatically generated

17. Find the categories where the average year of artworks is greater than the average year of all artworks.

mysql> SELECT c.Name AS Category, AVG(ar.Year) AS AvgYear

-> FROM Categories c

-> JOIN Artworks ar ON c.CategoryID = ar.CategoryID

-> GROUP BY c.Name

-> HAVING AVG(ar.Year) > (SELECT AVG(Year) FROM Artworks);

A screenshot of a phone

Description automatically generated

18. List the artworks that were not exhibited in any exhibition.

mysql> SELECT ar.Title

-> FROM Artworks ar

-> LEFT JOIN ExhibitionArtworks ea ON ar.ArtworkID = ea.ArtworkID

-> WHERE ea.ExhibitionID IS NULL;

A screenshot of a computer

Description automatically generated

19. Show artists who have artworks in the same category as "Mona Lisa."

mysql> SELECT DISTINCT a.Name

-> FROM Artists a

-> JOIN Artworks ar ON a.ArtistID = ar.ArtistID

-> WHERE ar.CategoryID = (SELECT CategoryID FROM Artworks WHERE Title = 'Mona Lisa');

A screenshot of a computer

Description automatically generated

20. List the names of artists and the number of artworks they have in the gallery.

mysql> SELECT a.Name, COUNT(ar.ArtworkID) AS ArtworkCount

-> FROM Artists a

-> LEFT JOIN Artworks ar ON a.ArtistID = ar.ArtistID

-> GROUP BY a.Name

-> ORDER BY ArtworkCount;

A screenshot of a computer

Description automatically generated