Fall
2023

Art Gallery Management Database

UNDER THE GUIDANCE OF PROF. KANNAN SRIKANTH

GROUP 11:
SATHYANAND KODI
ANANT TYAGI
SAI CHARAN KAKARALA
PAVAN KANTIPUDI
SIBI CHAKRAVARTHY VASUDEVAN

Meeting Log

S.no	Meeting date / time	Members present	Items discussed
1	Date: 19/09/2023 Time: 13:50sec	SAI CHARAN KAKARALA PAVAN KANTIPUDI SATHYANAND KODI ANANT TYAGI	General Topic Discussion
2	Date: 19/09/2023 Time: 1 Hr. 32 mins	SAI CHARAN KAKARALA PAVAN KANTIPUDI SATHYANAND KODI ANANT TYAGI	Project Scope and Objectives
3	Date: 23/09/2023 Time: 1 Hr. 38 mins	SAI CHARAN KAKARALA PAVAN KANTIPUDI SATHYANAND KODI ANANT TYAGI SIBI CHAKRAVARTHY VASUDEVAN	Benefits and Executive Summary
4	Date: 24/09/2023 Time: 20mins	SAI CHARAN KAKARALA PAVAN KANTIPUDI SATHYANAND KODI ANANT TYAGI SIBI CHAKRAVARTHY VASUDEVAN	Project Overview and Submission
5	Date: 14/10/2023 Time: 40mins	SAI CHARAN KAKARALA PAVAN KANTIPUDI SATHYANAND KODI ANANT TYAGI SIBI CHAKRAVARTHY VASUDEVAN	Relational Data Model layout and design
6	Date: 24/10/2023 Time: 30mins	SAI CHARAN KAKARALA PAVAN KANTIPUDI SATHYANAND KODI ANANT TYAGI SIBI CHAKRAVARTHY VASUDEVAN	Stored Procedures, Triggers, Functions and Views implementation
7	Date: 17/11/2023 Time: 40mins	SAI CHARAN KAKARALA PAVAN KANTIPUDI SATHYANAND KODI ANANT TYAGI SIBI CHAKRAVARTHY VASUDEVAN	Group Video ideas
8	Date: 29/11/2023 Time: 30mins	SAI CHARAN KAKARALA PAVAN KANTIPUDI SATHYANAND KODI ANANT TYAGI SIBI CHAKRAVARTHY VASUDEVAN	Group Video Shooting
9	Date: 10/12/2023 Time: 1 Hr. 40mins	SAI CHARAN KAKARALA PAVAN KANTIPUDI SATHYANAND KODI ANANT TYAGI SIBI CHAKRAVARTHY VASUDEVAN	Final project's overall review and group video presentation submission

Team Member's Contribution

- Database Designing & ERR Diagram – SATHYANAND KODI
- .SQL file creation – ANANT TYAGI
- Set 1 table creation and Data Entry – PAVAN KANTIPUDI
- Set 2 table creation and Data Entry - SIBI CHAKRAVARTHY VASUDEVAN
- Set 3 table creation and Data Entry - PAVAN KANTIPUDI and SAI CHARAN KAKARALA
- Query creation for Stored Procedure and View – SAI CHARAN KAKARALA
- Project Write Up – ANANT TYAGI
- Report Consolidation – SATHYANAND KODI

ART GALLERY MANAGEMENT DATABASE

The Art Gallery Management Database is a complete system meticulously crafted to transform how art galleries operate. Its main objective is to streamline and automate processes to improve gallery staff productivity, enhance visitor experiences, and protect priceless art treasures. This comprehensive solution takes care of everything from monitoring artist profiles and categorizing artwork to organizing exhibitions, keeping track of inventories, and processing financial transactions.

This database is a powerhouse for gallery administrators since it centralizes data and makes difficult tasks simple. It improves management in the background while also giving visitors a more interesting and tailored experience. Additionally, it is essential to the preservation of the gallery's priceless art treasures and offers insightful reporting and analytics. Essentially, the Art Gallery Management Database stands as a pivotal tool, fostering excellence in gallery operations and the overall art appreciation journey.

Key Features:

Artwork Catalog: Administrators of galleries can catalog and classify paintings using the database. The artist's name, title, medium, dimensions, date of production, and history of acquisitions are all included in detail. This extensive library makes it easier to find, arrange, and retrieve artwork.

Artist profiles: Information about artists, including biographical information, contact information, and a portfolio of their works, is kept in specific profiles. This function aids in the management of interactions with artists and gives viewers knowledge about the people who created the works of art.

Management of Exhibitions: The system makes it possible to organize and administer art exhibitions. Users can build up display layouts, select exhibition themes, assign artwork to exhibitions, and schedule exhibitions. Dates, descriptions, and participating artists are included in the exhibition information.

Inventory tracking: Whether an artwork is on view, in storage, or out on loan, gallery employees may keep track of its whereabouts and current state. Real-time updates aid in avoiding double or excessive booking of artwork for shows.

Sales and Transactions: The database manages sales and transactions, such as those for memberships, event tickets, and artwork. It creates invoices, monitors payment progress, and keeps account of previous purchases made by customers.

Customer relations: Information about visitors, such as contact information and preferences, is captured in customer profiles. Based on the interests and purchasing patterns of the clients, this information may be used to send newsletters, invitations to events, and customized suggestions.

Preservation and Security: The system includes security features to safeguard priceless artwork. To guarantee the long-term protection of the gallery's holdings, it contains functions like climate control monitoring, insurance tracking, and maintenance scheduling.

Analytics and Reporting: Extensive reporting capabilities offer information on gallery operations, sales patterns, well-liked shows, and clientele statistics. Administrators of galleries can use this information to organize next exhibitions and make educated judgments.

Accessibility: The database is available to authorized workers from a variety of locations, encouraging flexibility and collaboration. It can also be expanded to offer restricted access to customers and artists for activities, such making artwork suggestions or ordering tickets online.

User Management: Administrators can manage user rights and responsibilities to make sure that only authorized users can carry out tasks inside the database. As a result, data security is improved and unauthorized alterations are avoided.

Gallery Promotion and Marketing:

- Manage marketing campaigns and promotions.
- Track the effectiveness of marketing efforts and return on investment (ROI).

Backup and Disaster Recovery:

- Regularly back up the database to prevent data loss.
- Develop a disaster recovery plan to ensure data integrity in case of system failures or data breaches.

Compliance:

- Ensure that the project complies with legal and ethical standards, including copyright and art market regulations.

Benefits:

Efficiency: By streamlining daily operations and lowering administrative costs, the Art Gallery Management Database frees up staff to concentrate on curatorial and customer interaction tasks.

Better Visitor Experience: A well-organized gallery, timely exhibition information, and individualized interactions benefit visitors and improve their overall gallery experience.

Artwork Preservation: The system includes tools for monitoring the locations of artwork, climate management, and maintenance schedules to make sure priceless works of art are preserved.

Making Data-Driven Decisions: Analytics and reporting technologies enable gallery managers to plan upcoming shows based on previous trends and other data-driven decisions.

Artist Engagement: By upholding artist biographies, promoting their work, and enabling direct conversation, the database strengthens connections with artists.

In summary, the Art Gallery Management Database is a powerful and comprehensive solution that revolutionizes how art galleries operate. It is a vital tool for both gallery administrators and art fans since it makes administrative jobs simpler, improves visiting experiences, and safeguards priceless works of art. This database provides doors to a world where art thrives and creativity flourishes with its wide range of capabilities, from classifying artworks and managing exhibitions to conserving the rich tapestry of artists and their inventions. It is more than just a management system; it is the embodiment of a commitment to excellence in art curation, preservation, and appreciation.

Relational Data Model:

Relationship between Artists and Artworks Tables:

ArtistID	Name	Birthdate	Nationality

ArtworkID	CreationDate	Medium	Dimensions	ArtistID

The relationship between these tables is established using the "ArtistID" field in the "Artworks" table, which references the unique identifier of the artist in the "Artists" table, allowing you to connect specific artworks to their respective artists

Relationship between Sales, Artworks and Customers Tables:

SaleID	SaleDate	ArtworkID	CustomerID	SalePrice

ArtworkID	CreationDate	Medium	Dimensions	ArtistID

CustomerID	Name	ContactInformation

These tables are designed to track and manage information about customers, their purchases (sales), and the artworks they buy. The relationships between the tables help organize and query data related to sales transactions, including details about the artworks sold and the customers involved in each transaction.

Relationship between ExhibitionArtworkMapping, Exhibitions and Artworks Tables:

ExhibitionID	ArtworkID

ExhibitionID	ExhibitionName	StartDate	EndDate

ArtworkID	CreationDate	Medium	Dimensions	ArtistID

These tables and their relationships allow you to organize and manage information about art exhibitions, the artworks displayed in those exhibitions, and details about the artworks and artists. The "ExhibitionArtworkMapping" table acts as a bridge to establish the many-to-many relationship between exhibitions and artworks, enabling you to track which artworks are featured in each exhibition and vice versa.

Relationship between ExhibitionArtworkSchedule, Exhibitions and Artworks Tables:

ExhibitionID	ArtworkID	ScheduleDate

ExhibitionID	ExhibitionName	StartDate	EndDate

ArtworkID	CreationDate	Medium	Dimensions	ArtistID

These tables and their relationships allow you to manage and track information about art exhibitions, their schedules, the artworks featured in each exhibition, and details about the artworks and artists involved in the exhibitions. The junction tables ("ExhibitionArtworkMapping" and "ExhibitionArtworkSchedule") help establish the complex relationships between exhibitions and artworks, as well as their scheduling within exhibitions.

Relationship between ExhibitionStaffAssignment, Exhibitions and GalleryStaff Tables:

ExhibitionID	StaffID	AssignmentDate

ExhibitionID	ExhibitionName	StartDate	EndDate

StaffID	Name	ContactInformation	Role

These tables and their relationships allow you to manage and track information related to art exhibitions, the staff members assigned to those exhibitions, and details about the gallery staff. The "ExhibitionStaffAssignment" table establishes the association between exhibitions and staff assignments, while the "Exhibitions" and "GalleryStaff" tables provide additional context and details about exhibitions and staff members, respectively. This structure helps in organizing and recording staff assignments for each exhibition event.

Relationship between ArtworkLoan, Artworks and Customers Tables:

Here lender ID refers to customerID

LoanID	ArtworkID	LoanDate	ReturnDate	LenderID

ArtworkID	CreationDate	Medium	Dimensions	ArtistID

CustomerID	Name	ContactInformation

These tables and their relationships allow you to manage and track information related to artwork loans, the details of the loaned artworks, and customer information. The "ArtworkLoan" table serves as a bridge between loaned artworks and customers, facilitating the recording and tracking of artwork loans.

Relationship between EmployeeShifts, and GalleryStaff Tables:

ShiftID	StaffID	ShiftDate	StartTime	EndTime

StaffID	Name	ContactInformation	Role

These tables and their relationship allow you to track and manage information about employee shifts within a gallery setting. The "EmployeeShifts" table records the details of each shift, including which staff member was assigned to it, and the "GalleryStaff" table stores information about the gallery staff members, including their names, contact information, and roles. The foreign key relationship between "EmployeeShifts" and "GalleryStaff" connects shift records to the staff members who performed those shifts.

Relationship between ExhibitionAttendance, Exhibitions and Customers Tables:

AttendanceID	ExhibitionID	CustomerID	CheckInTime

ExhibitionID	ExhibitionName	StartDate	EndDate

CustomerID	Name	ContactInformation

These tables and their relationships enable you to keep track of attendance at art exhibitions by recording which customers attended specific exhibitions and the check-in times. The "Exhibitions" table provides information about the exhibitions themselves, and the "Customers" table stores customer details. The foreign key relationships connect attendance records to the corresponding exhibitions and customers, allowing you to manage attendance data effectively.

Relationship between ArtworkRestoration and Artworks Tables:

RestorationID	ArtworkID	RestorationDate	Cost

ArtworkID	CreationDate	Medium	Dimensions	ArtistID

These tables and their relationship allow you to track and manage information about the restoration of artworks, including details about the restoration process (such as the date and cost), and to associate each restoration with a specific artwork from the "Artworks" table. This enables you to maintain a record of the restoration history of individual artworks.

Relationship between MembershipRenewal and Membership Tables:

RenewalID	MemberID	RenewalDate	RenewalAmount

MemberID	MemberName	JoinDate	MembershipType

These tables and their relationship allow you to manage and track membership information, including member details and membership renewals. The "MembershipRenewal" table links renewal records to individual members in the "Membership" table, making it possible to record and keep track of when memberships are renewed and the associated renewal amounts.

Relationship between ArtworkReviews, Artworks and Customers Tables:

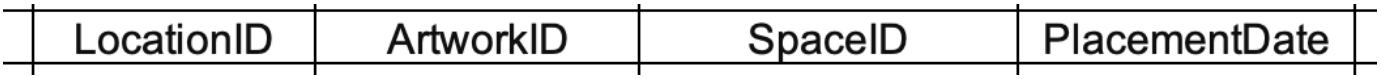
ReviewID	ArtworkID	CustomerID	ReviewText	Rating

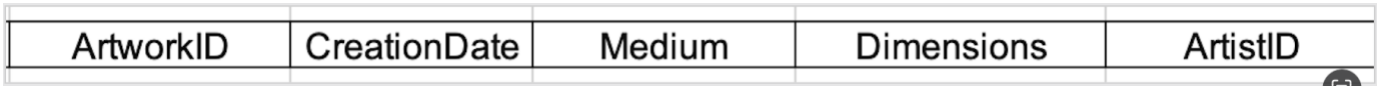
ArtworkID	CreationDate	Medium	Dimensions	ArtistID

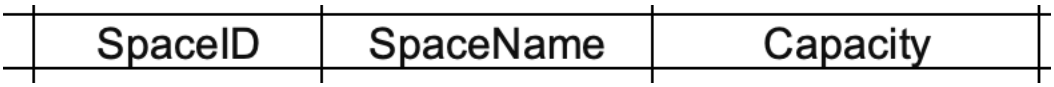
CustomerID	Name	ContactInformation

These tables and their relationships allow you to manage and track information related to artwork reviews, the details of the reviewed artworks, and customer information. The "ArtworkReviews" table serves as a bridge between reviews, artworks, and customers, facilitating the recording and retrieval of reviews associated with specific artworks and their respective authors.

Relationship between ArtworkGalleryLocations, Artworks and GallerySpacesTables:







These tables and their relationships allow you to manage and track the locations of artworks within gallery spaces, store details about the artworks, and maintain information about the gallery spaces. The "ArtworkGalleryLocations" table serves as a bridge between artworks, gallery spaces, and the placement dates, facilitating the organization and retrieval of location-related data for artworks within gallery settings.

Assumptions/Notes About Data Entities and Relationships:

Artists:

Each artist has a unique ArtistID.

Artists may have multiple artworks associated with them in the Artworks table.

Artworks:

Each artwork has a unique ArtworkID.

Artworks are created by artists, and the ArtistID serves as a foreign key to link them.

Artworks can belong to one or more categories defined in the ArtworkCategories table.

Artworks can be scheduled for display in exhibitions using the ExhibitionArtworkSchedule table.

Artworks can be reviewed and rated by customers in the ArtworkReviews table.

The placement of artworks within gallery spaces is recorded in the ArtworkGalleryLocations table.

Exhibitions:

Each exhibition has a unique ExhibitionID.

Artworks can be part of multiple exhibitions through the ExhibitionArtworkMapping table.

Staff members can be assigned to work on specific exhibitions using the ExhibitionStaffAssignment table.

Attendance records for exhibitions are logged in the ExhibitionAttendance table.

GalleryStaff:

Each staff member has a unique StaffID.

Staff members have roles within the gallery, such as curator, security, or receptionist.

Customers:

Each customer has a unique CustomerID.

Customers can purchase artworks, and their transactions are recorded in the Sales table.

Customers can also provide reviews and ratings for artworks in the ArtworkReviews table.

Membership information for customers is stored in the Membership table.

Sales:

Each sale has a unique SaleID.

Sales transactions include the sale date, the artwork sold (via ArtworkID), the customer (via CustomerID), and the sale price.

Sales can be associated with specific transactions recorded in the Transactions table.

ArtworkCategories:

Categories for artworks are defined with unique CategoryIDs.

Artworks can belong to multiple categories, and categories have names and descriptions.

ExhibitionArtworkMapping:

This table establishes a many-to-many relationship between exhibitions and artworks by mapping ExhibitionID and ArtworkID pairs.

It allows multiple artworks to be associated with multiple exhibitions.

Transactions:

Financial transactions are recorded with unique TransactionIDs.

Transactions include information such as transaction type, transaction date, and amount.

GallerySpaces:

Each gallery space has a unique SpaceID.

Gallery spaces have names and capacity information.

ExhibitionArtworkSchedule:

Artworks can be scheduled for specific dates within exhibitions using this table.

It associates ExhibitionID, ArtworkID, and ScheduleDate.

ExhibitionStaffAssignment:

Staff members can be assigned to work on specific exhibitions using this table.

It links ExhibitionID, StaffID, and AssignmentDate.

ArtworkLoan:

Records loans of artworks, including LoanID, ArtworkID, LoanDate, ReturnDate, and LenderID.

Typically, the lender is a customer.

EmployeeShifts:

Tracks shifts worked by gallery staff, with ShiftID, StaffID, ShiftDate, StartTime, and EndTime.

ExhibitionAttendance:

Logs attendance records for exhibitions, including AttendanceID, ExhibitionID, CustomerID, and CheckInTime.

ArtworkRestoration:

Records information about the restoration of artworks, including RestorationID, ArtworkID, RestorationDate, and Cost.

Membership:

Stores data about gallery memberships, including MemberID, MemberName, JoinDate, and MembershipType.

MembershipRenewal:

Tracks membership renewals with RenewalID, MemberID, RenewalDate, and RenewalAmount.

ArtworkReviews:

Collects reviews and ratings for artworks, including ReviewID, ArtworkID, CustomerID, ReviewText, and Rating.

ArtworkGalleryLocations:

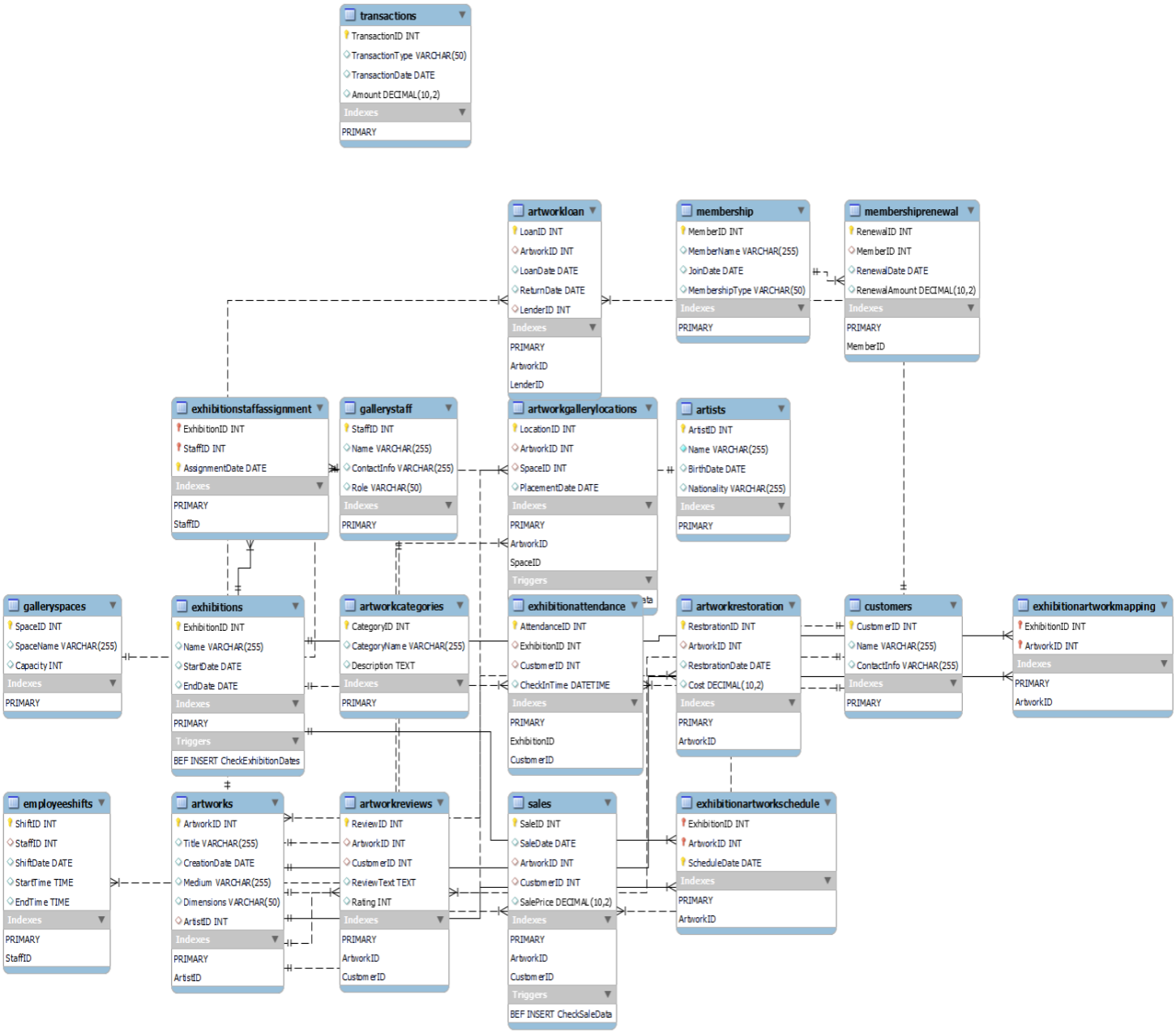
Records the placement of artworks within gallery spaces, with LocationID, ArtworkID, SpaceID, and PlacementDate.

These assumptions and notes provide an overview of the data entities, their attributes, and the relationships between them in the database schema. They help clarify how the data is structured and how different tables are connected.

Scenarios covered for the database:

1. Created 20 tables.
2. Developed 10 complex queries.
3. Used 10 Joins with the combinations of (Left Join, Right Join, Inner Join and Cross Join).
4. Included Self-Join.
5. Included Sub-Queries.
6. Included 3 Functions.
7. Included 3 Procedures.
8. Included 3 triggers.

E-R Diagram:



Design of the Database:

Table Name	**Primary Key**	**Foreign Key**	**Non-key attributes**	**# of Rows in Table**
Artists	ArtistID	NA	Name Birthdate Nationality	10

Table Name	**Primary Key**	**Foreign Key**	**Non-key attributes**	**# of Rows in Table**
Artworks	ArtworkID	ArtistID	Title CreationDate Medium Dimensions	10

Table Name	**Primary Key**	**Foreign Key**	**Non-key attributes**	**# of Rows in Table**
Exhibitions	ExhibitionID	NA	ExhibitionName StartDate EndDate	10

Table Name	**Primary Key**	**Foreign Key**	**Non-key attributes**	**# of Rows in Table**
GalleryStaff	StaffID	NA	Name ContactInformation Role	10

Table Name	**Primary Key**	**Foreign Key**	**Non-key attributes**	**# of Rows in Table**
Customers	CustomerID	NA	Name ContactInformation	10

Table Name	Primary Key	Foreign Key	Non-key attributes	# of Rows in Table
Sales	SaleID	ArtworkID CustomerID	SaleDate SalePrice	10

Table Name	Primary Key	Foreign Key	Non-key attributes	# of Rows in Table
ArtworkCategories	CategoryID	NA	CategoryName Description	10

Table Name	Primary Key	Foreign Key	Non-key attributes	# of Rows in Table
ExhibitionArtworkMapping	NA	ExhibitionID ArtworkID	NA	10

Table Name	Primary Key	Foreign Key	Non-key attributes	# of Rows in Table
Transactions	TransactionID	NA	TransactionType TransactionDate Amount	10

Table Name	Primary Key	Foreign Key	Non-key attributes	# of Rows in Table
GallerySpaces	SpaceID	NA	SpaceName Capacity	10

Table Name	Primary Key	Foreign Key	Non-key attributes	# of Rows in Table
ExhibitionArtworkSchedule	SpaceID	ExhibitionID ArtworkID	ScheduleDate	10

Table Name	Primary Key	Foreign Key	Non-key attributes	# of Rows in Table
ExhibitionStaffAssignment	SpaceID	ExhibitionID StaffID	AssignmentDate	10

Table Name	Primary Key	Foreign Key	Non-key attributes	# of Rows in Table
ArtworkLoan	LoanID	ArtworkID LenderID	LoanDate ReturnDate	10

Table Name	Primary Key	Foreign Key	Non-key attributes	# of Rows in Table
EmployeeShifts	ShiftID	StaffID	ShiftDate StartTime EndTime	10

Table Name	Primary Key	Foreign Key	Non-key attributes	# of Rows in Table
ExhibitionAttendance	AttendanceID	ExhibitionID CustomerID	CheckInTime	10

Table Name	Primary Key	Foreign Key	Non-key attributes	# of Rows in Table
ArtworkRestoration	RestorationID	ArtworkID	RestorationDate Cost	10

Table Name	Primary Key	Foreign Key	Non-key attributes	# of Rows in Table
Membership	MemberID	NA	MemberName JoinDate MembershipType	10

Table Name	Primary Key	Foreign Key	Non-key attributes	# of Rows in Table
MembershipRenewal	RenewalID	MemberID	RenewalDate RenewalAmount	10

Table Name	Primary Key	Foreign Key	Non-key attributes	# of Rows in Table
ArtworkReviews	ReviewID	ArtworkID CustomerID	ReviewText Rating	10

Table Name	Primary Key	Foreign Key	Non-key attributes	# of Rows in Table
ArtworkGalleryLocations	LocationID	ArtworkID SpaceID	PlacementDate	10

Third Normal Form (3NF) Database:

All tables have a primary key.

There are no partial dependencies (attributes depend on the entire primary key).

There are no transitive dependencies (non-key attributes depend on other non-key attributes).

Here's the modified database design in 3NF:

Tables:

Artists:

ArtistID (Primary Key)

Name

BirthDate

Nationality

ArtworkCategories:

CategoryID (Primary Key)

CategoryName

Artworks:

ArtworkID (Primary Key)

Title

CreationDate

Medium

Dimensions

ArtistID (Foreign Key references Artists)

CategoryID (Foreign Key references ArtworkCategories)

Exhibitions:

ExhibitionID (Primary Key)

Name

StartDate

EndDate

GalleryStaff:

StaffID (Primary Key)

Name

ContactInfo

Role

Customers:

CustomerID (Primary Key)

Name

ContactInfo

Sales:

SaleID (Primary Key)

SaleDate

ArtworkID (Foreign Key references Artworks)

CustomerID (Foreign Key references Customers)

SalePrice

Transactions:

TransactionID (Primary Key)

TransactionType

TransactionDate

Amount

GallerySpaces:

SpaceID (Primary Key)

SpaceName

Capacity

ExhibitionArtworkMapping:

ExhibitionID (Primary Key, Foreign Key references Exhibitions)

ArtworkID (Primary Key, Foreign Key references Artworks)

ExhibitionStaffAssignment:

AssignmentID (Primary Key)

ExhibitionID (Foreign Key references Exhibitions)

StaffID (Foreign Key references GalleryStaff)

ArtworkLoan:

LoanID (Primary Key)

ArtworkID (Foreign Key references Artworks)

LoanDate

ReturnDate

LenderID (Foreign Key references Customers)

EmployeeShifts:

ShiftID (Primary Key)

StaffID (Foreign Key references GalleryStaff)

ShiftDate

StartTime

EndTime

ExhibitionAttendance:

AttendanceID (Primary Key)

ExhibitionID (Foreign Key references Exhibitions)

CustomerID (Foreign Key references Customers)

CheckInTime

ArtworkRestoration:

RestorationID (Primary Key)

ArtworkID (Foreign Key references Artworks)

RestorationDate

Cost

Membership:

MemberID (Primary Key)

MemberName

JoinDate

MembershipType

MembershipRenewal:

RenewalID (Primary Key)

MemberID (Foreign Key references Membership)

RenewalDate

RenewalAmount

ArtworkReviews:

ReviewID (Primary Key)

ArtworkID (Foreign Key references Artworks)

CustomerID (Foreign Key references Customers)

ReviewText

Rating

ArtworkGalleryLocations:

LocationID (Primary Key)

ArtworkID (Foreign Key references Artworks)

SpaceID (Foreign Key references GallerySpaces)

PlacementDate

Now, the tables are in 3NF, and there are no partial or transitive dependencies within the tables. Each table has a well-defined primary key, and the relationships between tables are based on these keys.

Views and Stored Procedures created on Database:

Stored Procedures:

Stored Procedure 1

```
DELIMITER //


CREATE PROCEDURE InsertArtist(
   IN artist_id INT,
   IN artist_name VARCHAR(255),
   IN birth_date DATE,
   IN nationality VARCHAR(255)
)
BEGIN
   INSERT INTO Artists (ArtistID, Name, BirthDate, Nationality)
   VALUES (artist_id, artist_name, birth_date, nationality);
END //


DELIMITER ;


-- Call the InsertArtist procedure to insert a new artist
CALL InsertArtist(11, 'Ronaldo', '1970-02-05', 'Portugese');
select * from Artists;
```

Name: InsertArtist

Purpose: The procedure is designed to insert information about a new artist into the "Artists" table. It takes four input parameters: artist_id, artist_name, birth_date, and nationality. These parameters represent the artist's unique identifier, name, birth date, and nationality, respectively.

SQL Syntax:

It uses the INSERT INTO statement to add a new row to the "Artists" table.

The values for the columns ArtistID, Name, BirthDate, and Nationality are provided by the input parameters.

The procedure is defined within a BEGIN...END block, which is a common structure for MySQL stored procedures.

It uses the DELIMITER command to change the delimiter temporarily, allowing the use of // as a delimiter for the procedure definition.

Finally, it resets the delimiter back to ; after defining the procedure.

Usage: After defining the procedure, you can call it using the CALL statement, passing the required values for the input parameters. In the example provided, the procedure is called with sample values to insert a new artist record into the "Artists" table.

Result: When the procedure is called, it inserts a new artist record into the "Artists" table with the provided information.

Stored Procedure 2

```
DELIMITER //


CREATE PROCEDURE InsertArtwork(
    IN artwork_id INT,
    IN title VARCHAR(255),
    IN creation_date DATE,
    IN medium VARCHAR(255),
    IN dimensions VARCHAR(50),
    IN artist_id INT
)
BEGIN
    INSERT INTO Artworks (ArtworkID, Title, CreationDate, Medium, Dimensions, ArtistID)
    VALUES (artwork_id, title, creation_date, medium, dimensions, artist_id);
END //


DELIMITER ;


-- Call the InsertArtwork procedure to insert a new artwork
CALL InsertArtwork(11, 'Mona Lisa', '1980-07-29', 'Oil on poplar panel', '60 cm x 70 cm', 11);
select * from Artworks;
```

Name: InsertArtwork

Purpose: The procedure is designed to insert information about a new artwork into the "Artworks" table. It takes six input parameters: artwork_id, title, creation_date, medium, dimensions, and artist_id. These parameters represent the artwork's unique identifier, title, creation date, medium, dimensions, and the artist's unique identifier, respectively.

SQL Syntax:

It uses the INSERT INTO statement to add a new row to the "Artworks" table.

The values for the columns ArtworkID, Title, CreationDate, Medium, Dimensions, and ArtistID are provided by the input parameters.

The procedure is defined within a BEGIN...END block, which is a common structure for MySQL stored procedures.

It uses the DELIMITER command to change the delimiter temporarily, allowing the use of // as a delimiter for the procedure definition.

Finally, it resets the delimiter back to ; after defining the procedure.

Usage: After defining the procedure, you can call it using the CALL statement, passing the required values for the input parameters. In the example provided, the procedure is called with sample values to insert a new artwork record into the "Artworks" table.

Result: When the procedure is called, it inserts a new artwork record into the "Artworks" table with the provided information.

Stored Procedure 3

```
DELIMITER //


CREATE PROCEDURE InsertGalleryStaff(
   IN staff_id INT,
   IN staff_name VARCHAR(255),
   IN contact_info VARCHAR(255),
   IN role VARCHAR(50)
)
BEGIN
   INSERT INTO GalleryStaff (StaffID, Name, ContactInfo, Role)
   VALUES (staff_id, staff_name, contact_info, role);
END //


DELIMITER ;


-- Call the InsertGalleryStaff procedure to insert a new staff member
CALL InsertGalleryStaff(11, 'John Doe', 'johndoe@example.com', 'Curator');
select * from GalleryStaff;
```

Name: InsertGalleryStaff

Purpose: The procedure is designed to insert information about a new gallery staff member into the "GalleryStaff" table. It takes four input parameters: staff_id, staff_name, contact_info, and role. These parameters represent the staff member's unique identifier, name, contact information, and role, respectively.

SQL Syntax:

It uses the INSERT INTO statement to add a new row to the "GalleryStaff" table.

The values for the columns StaffID, Name, ContactInfo, and Role are provided by the input parameters.

The procedure is defined within a BEGIN...END block, which is a common structure for MySQL stored procedures.

It uses the DELIMITER command to change the delimiter temporarily, allowing the use of // as a delimiter for the procedure definition.

Finally, it resets the delimiter back to ; after defining the procedure.

Usage: After defining the procedure, you can call it using the CALL statement, passing the required values for the input parameters. In the example provided, the procedure is called with sample values to insert a new gallery staff member record into the "GalleryStaff" table.

Result: When the procedure is called, it inserts a new gallery staff member record into the "GalleryStaff" table with the provided information.

Views

View 1

```
SELECT
    A.Name AS ArtistName,
    A.BirthDate AS ArtistBirthDate,
    A.Nationality AS ArtistNationality,
    W.Title AS ArtworkTitle,
    W.CreationDate AS ArtworkCreationDate,
    W.Medium AS ArtworkMedium,
    W.Dimensions AS ArtworkDimensions
FROM Artists A
LEFT JOIN Artworks W ON A.ArtistID = W.ArtistID
ORDER BY ArtistName, ArtworkCreationDate;
```

Result: This query will give you a list of artists along with details of their artworks, ordered by artist name and artwork creation date.

View 2

```
SELECT
    E.Name AS ExhibitionName,
    A.Title AS ArtworkTitle,
    A.CreationDate AS ArtworkCreationDate,
    A.Medium AS ArtworkMedium,
    A.Dimensions AS ArtworkDimensions,
    AR.ArtistID AS ArtistID,
    AR.Name AS ArtistName
FROM ExhibitionArtworkMapping EAM
JOIN Exhibitions E ON EAM.ExhibitionID = E.ExhibitionID
JOIN Artworks A ON EAM.ArtworkID = A.ArtworkID
JOIN Artists AR ON A.ArtistID = AR.ArtistID
WHERE E.Name = 'Impressionist Masters'; -- Replace with the desired exhibition name
```

Result: This query retrieves a list of artworks that are part of a specific exhibition, along with details of the artist and artwork.

View 3

```
SELECT
   YEAR(SaleDate) AS SaleYear,
   SUM(SalePrice) AS TotalRevenue
FROM Sales
GROUP BY SaleYear
ORDER BY SaleYear;
```

Result: This query calculates the total sales revenue for each year based on the SaleDate in the Sales table.