

# Project Title: Art Gallery Database Management System (AGDMS)

## 1. PROJECT DEFINITION

**1.1. Project Objective:** To design and implement a comprehensive database management system for art galleries that manages artists, artworks, customers, employees, exhibitions, orders, and ticketing operations while maintaining complete audit trails and price history tracking.

**1.2. Scope:**

- Management of artist profiles and artwork portfolios
- Customer relationship and purchase history tracking
- Employee records and role management
- Exhibition planning and scheduling
- Order processing with detailed line items
- Ticket sales and event management
- Price history tracking for artworks
- Audit logging for accountability

## 2. DESCRIPTION

**2.1. System Overview**

The Art Gallery Database Management System is designed to streamline operations of art galleries by providing a centralized platform for managing all aspects of gallery operations. The system maintains comprehensive records of artists, their artworks, customer interactions, employee activities, exhibitions, sales transactions, and ticket bookings.

**2.2. Key Features**

- **Artist Management:** Complete profiles including biography, contact information, and portfolio tracking
- **Artwork Cataloging:** Detailed records with ownership history and price tracking over time
- **Customer Relationship Management:** Purchase history, preferences, and contact details
- **Employee Management:** Staff records, roles, and activity tracking
- **Exhibition Planning:** Event scheduling, artwork assignments, and visitor management
- **Order Processing:** Multi-item order handling with detailed line-item tracking
- **Ticketing System:** Event ticket sales and visitor tracking
- **Audit Trails:** Complete logging of all data modifications for accountability

## 2.3 Target Users

- Gallery Managers
- Sales Staff
- Curators
- Administrative Personnel
- Customers (for purchase history)

## 3. ENTERPRISE DATA MODEL

### 3.1. Business Rules

**BR1:** Each Artist must have a unique identifier and can create multiple artworks.

**BR2:** An Artwork is created by exactly one Artist but may be displayed in multiple Exhibitions.

**BR3:** Each Customer can place multiple Orders, but each Order belongs to one Customer.

**BR4:** An Order must contain at least one OrderLine, and each OrderLine references exactly one Artwork.

**BR5:** Each Exhibition can display multiple Artworks, and an Artwork can participate in multiple Exhibitions.

**BR6:** Tickets are issued for specific Exhibitions, and each Ticket is purchased by one Customer.

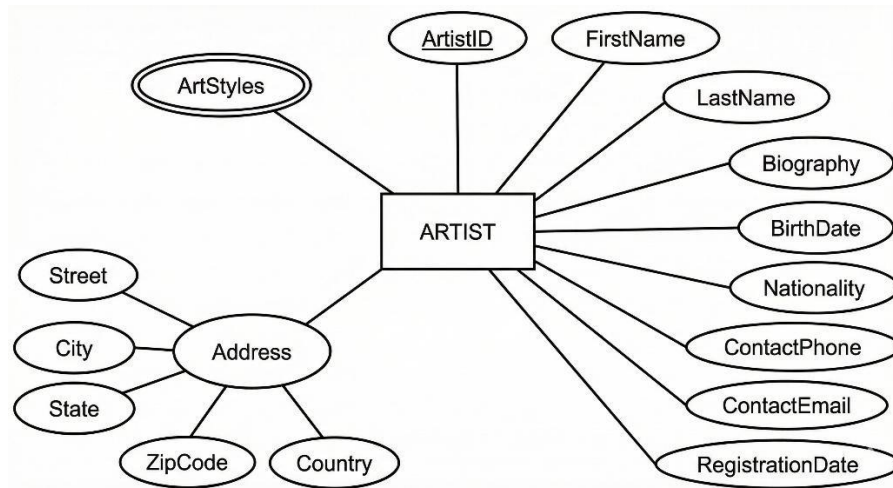
**BR7:** Employees manage Exhibitions, with each Exhibition managed by at least one Employee.

**BR8:** Artworks maintain a price history to track value changes over time.

**BR9:** Each OrderLine records the sale price at the time of purchase, which may differ from current artwork price.

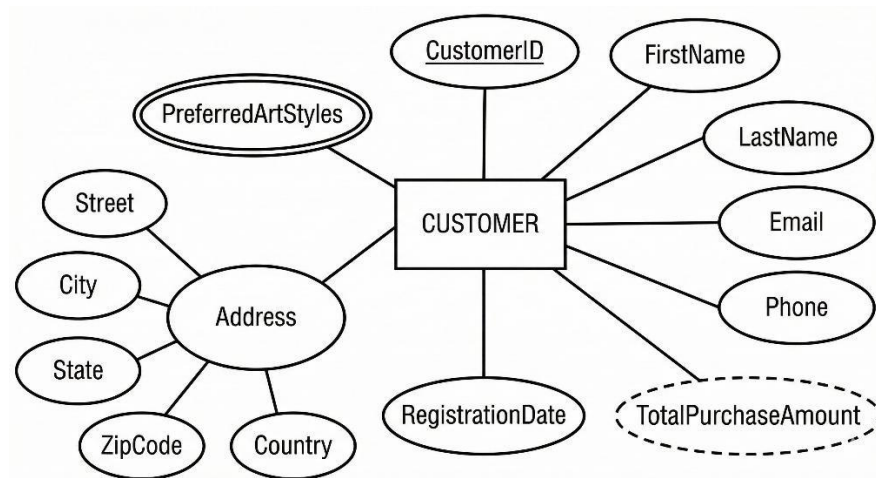
**BR10:** Customers, Artists, and Employees must have valid contact information.





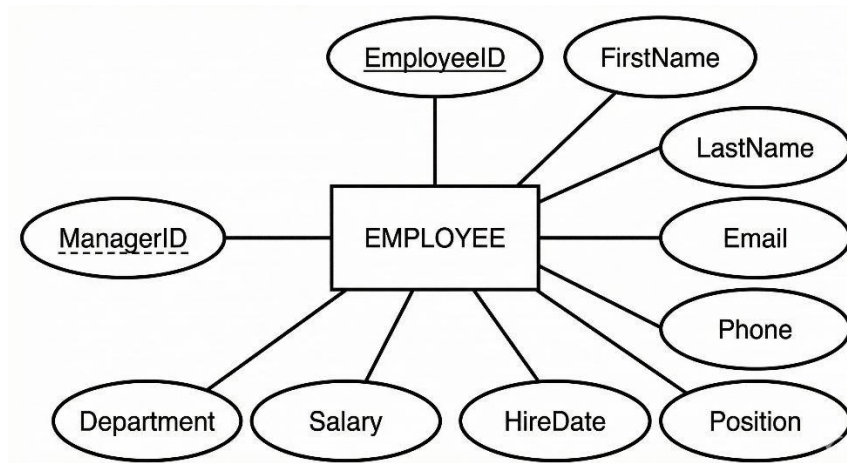
## ENTITY 2: CUSTOMER

- CustomerID (Primary Key), FirstName, LastName, Email, Phone
- Address (Composite: Street, City, State, ZipCode, Country), RegistrationDate, PreferredArtStyles (Multivalued) - *Violates 1NF*, TotalPurchaseAmount (Derived) - *Should be calculated*



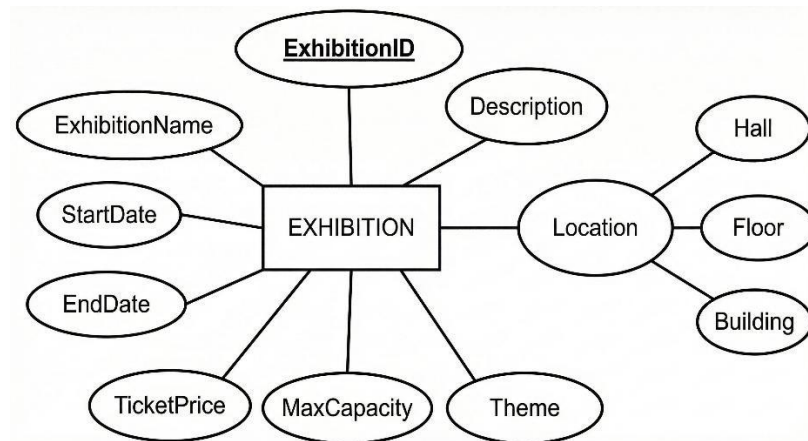
## ENTITY 3: EMPLOYEE

- EmployeeID (Primary Key), FirstName, LastName, Email, Phone, Position, HireDate, Salary, Department, ManagerID (Foreign Key - Unary Relationship)



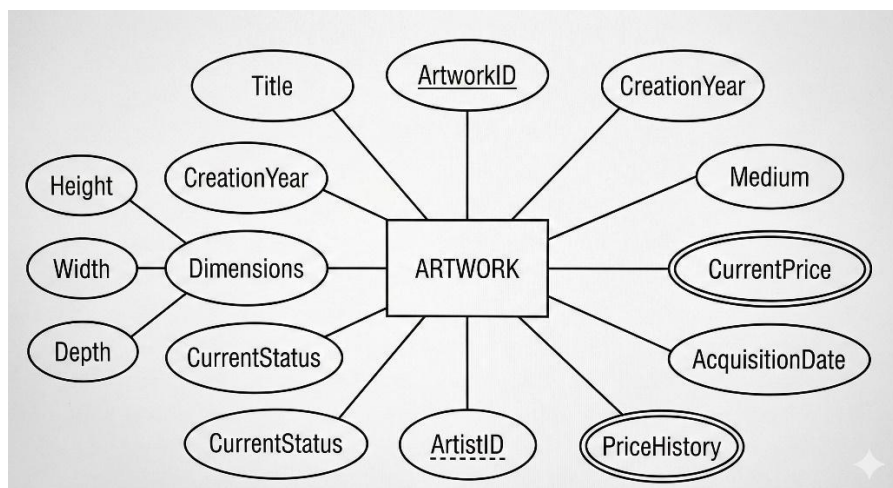
#### ENTITY 4: EXHIBITION

- ExhibitionID (Primary Key), ExhibitionName, Description, StartDate, EndDate, Location (Composite: Hall, Floor, Building), TicketPrice, MaxCapacity, Theme



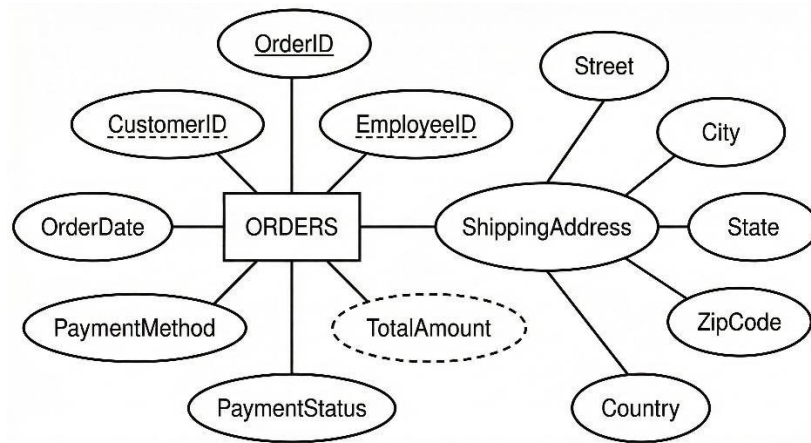
#### ENTITY 5: ARTWORK

- ArtworkID (Primary Key), Title, CreationYear, Medium, Dimensions (Composite: Height, Width, Depth), CurrentPrice, CurrentStatus (Available, Sold, On Exhibition), AcquisitionDate, ArtistID (Foreign Key), Style, PriceHistory (Multivalued: Date, Price) - *Violates 1NF*



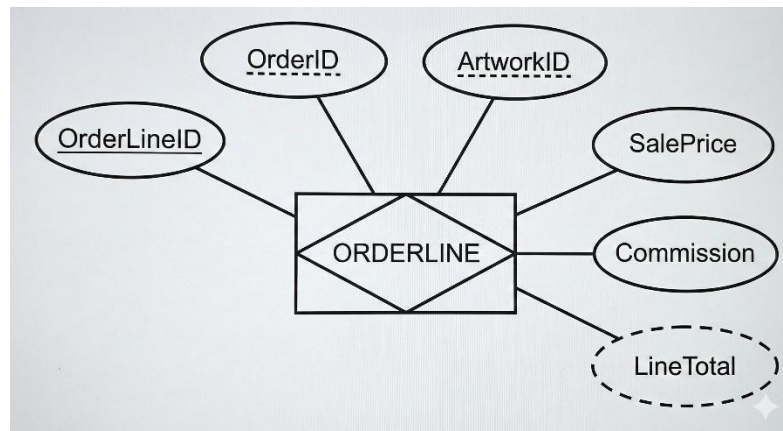
## ENTITY 6: ORDERS

- OrderID (Primary Key), OrderDate, CustomerID (Foreign Key), EmployeeID (Foreign Key), TotalAmount (Derived) - *Should be calculated from OrderLines*, PaymentMethod, PaymentStatus, ShippingAddress (Composite: Street, City, State, ZipCode, Country)



## ENTITY 7: ORDERLINE (Associative Entity)

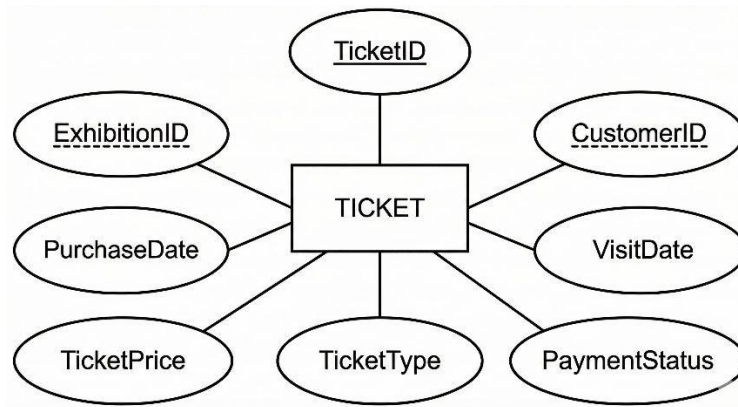
- OrderLineID (Primary Key), OrderID (Foreign Key), ArtworkID (Foreign Key), SalePrice, Commission, LineTotal (Derived:  $\text{SalePrice} + \text{Commission}$ ) - *Should be calculated*



## ENTITY 8: TICKET

- TicketID (Primary Key), ExhibitionID (Foreign Key), CustomerID (Foreign Key), PurchaseDate, VisitDate, TicketPrice, TicketType (Adult, Student, Senior), PaymentStatus

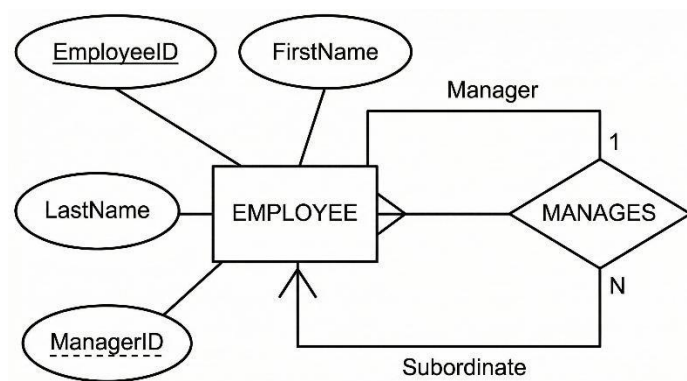




### 4.3. Relationship ER Diagrams

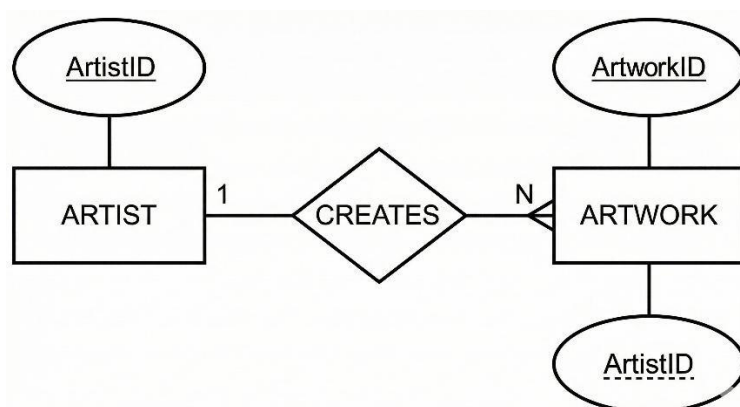
#### Relationship 1: UNARY - Employee Manages Employee

**Relationship Type:** Unary (Recursive) **Cardinality:** One-to-Many (1:N) **Description:** An Employee (Manager) supervises multiple Employees



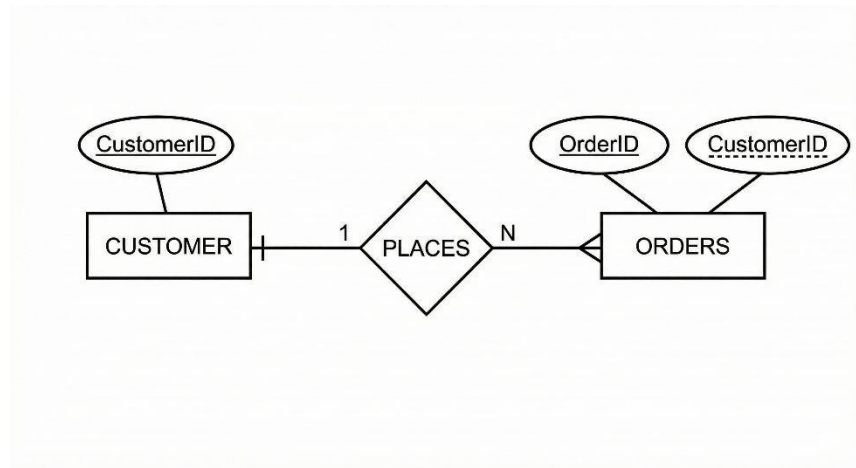
#### Relationship 2: BINARY - Artist Creates Artwork

**Relationship Type:** Binary **Cardinality:** One-to-Many (1:N) **Description:** One Artist creates many Artworks; each Artwork is created by one Artist



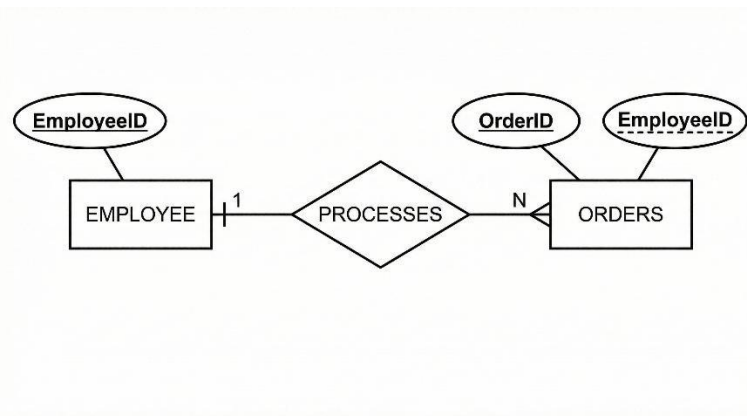
#### Relationship 3: BINARY - Customer Places Orders

**Relationship Type:** Binary **Cardinality:** One-to-Many (1:N) **Description:** One Customer places many Orders; each Order belongs to one Customer



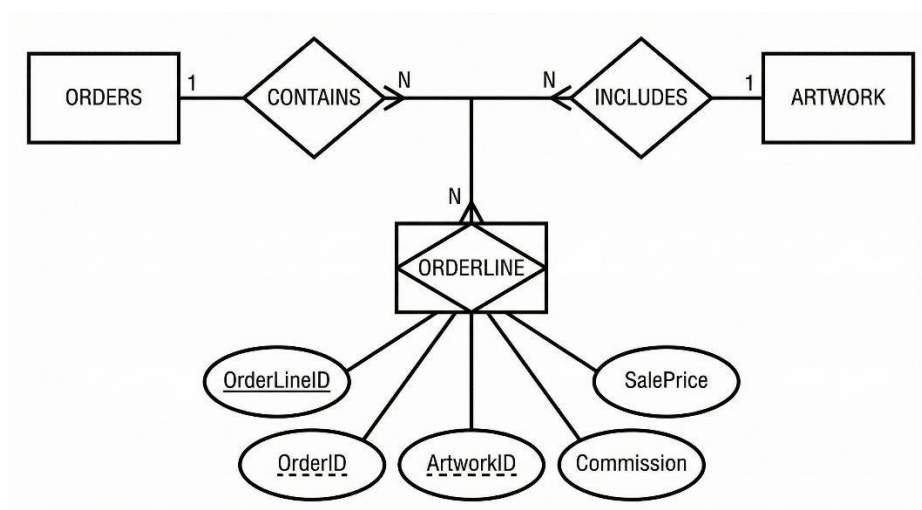
#### Relationship 4: BINARY - Employee Processes Orders

**Relationship Type:** Binary **Cardinality:** One-to-Many (1:N) **Description:** One Employee processes many Orders; each Order is processed by one Employee



#### Relationship 5: TERNARY - Orders Contains Artwork (via OrderLine)

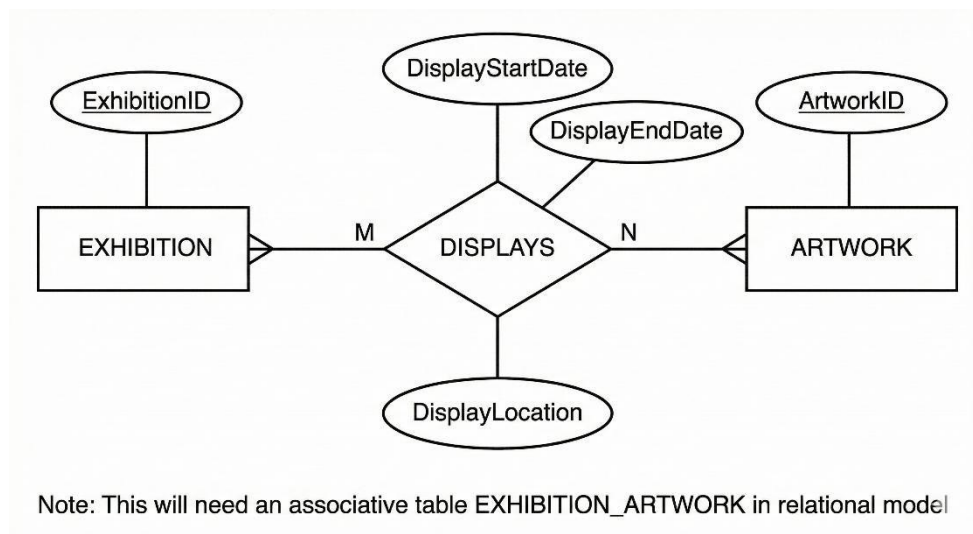
**Relationship Type:** Binary through Associative Entity **Cardinality:** Many-to-Many (M:N) resolved through OrderLine **Description:** Orders contain many Artworks; Artworks can be in many Orders (through OrderLine)



#### Relationship 6: BINARY - Exhibition Displays Artwork

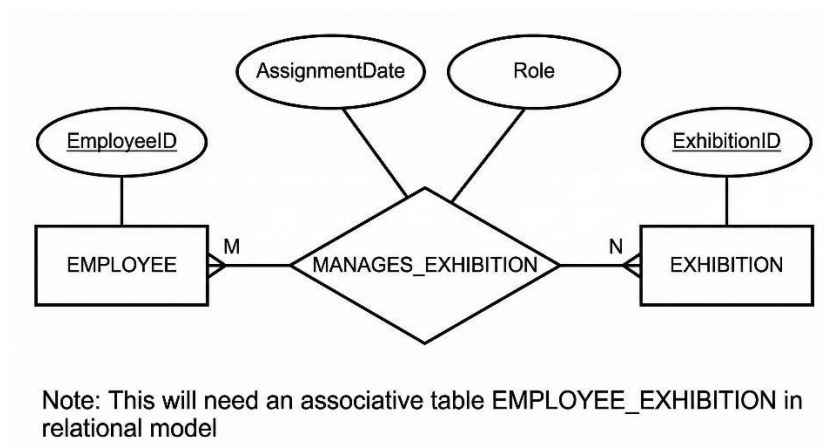


**Relationship Type:** Binary **Cardinality:** Many-to-Many (M:N) **Description:** Exhibitions display many Artworks; Artworks can be in many Exhibitions



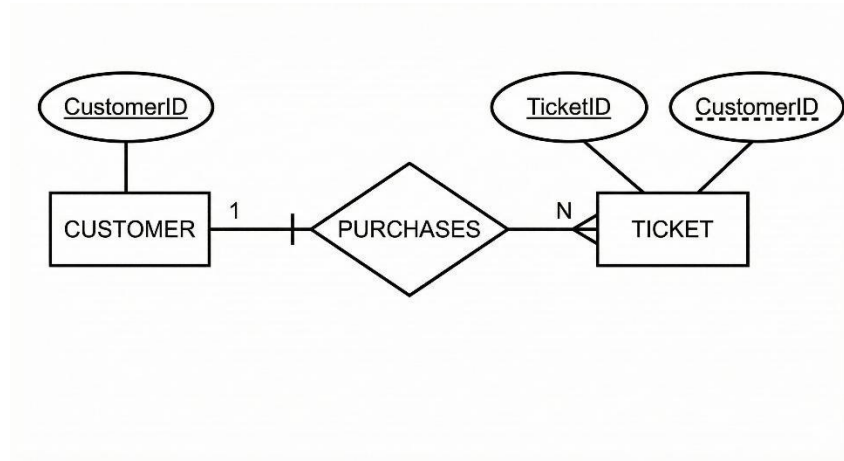
#### Relationship 7: BINARY - Employee Manages Exhibition

**Relationship Type:** Binary **Cardinality:** Many-to-Many (M:N) **Description:** Employees manage many Exhibitions; Exhibitions are managed by many Employees



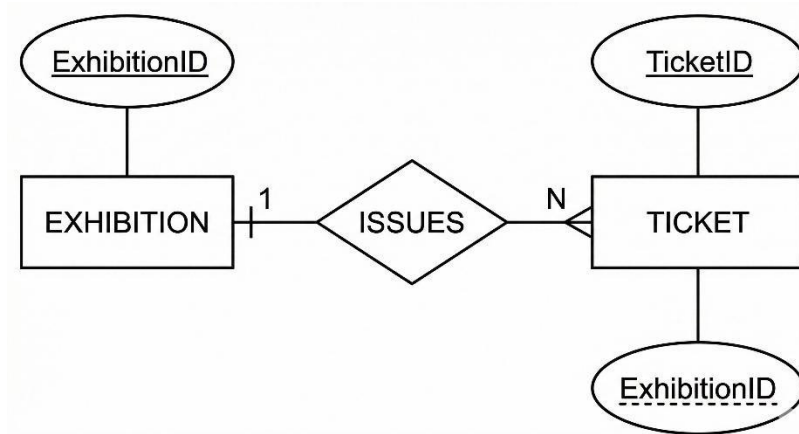
#### Relationship 8: BINARY - Customer Purchases Ticket

**Relationship Type:** Binary **Cardinality:** One-to-Many (1:N) **Description:** One Customer purchases many Tickets; each Ticket belongs to one Customer



#### Relationship 9: BINARY - Exhibition Issues Ticket

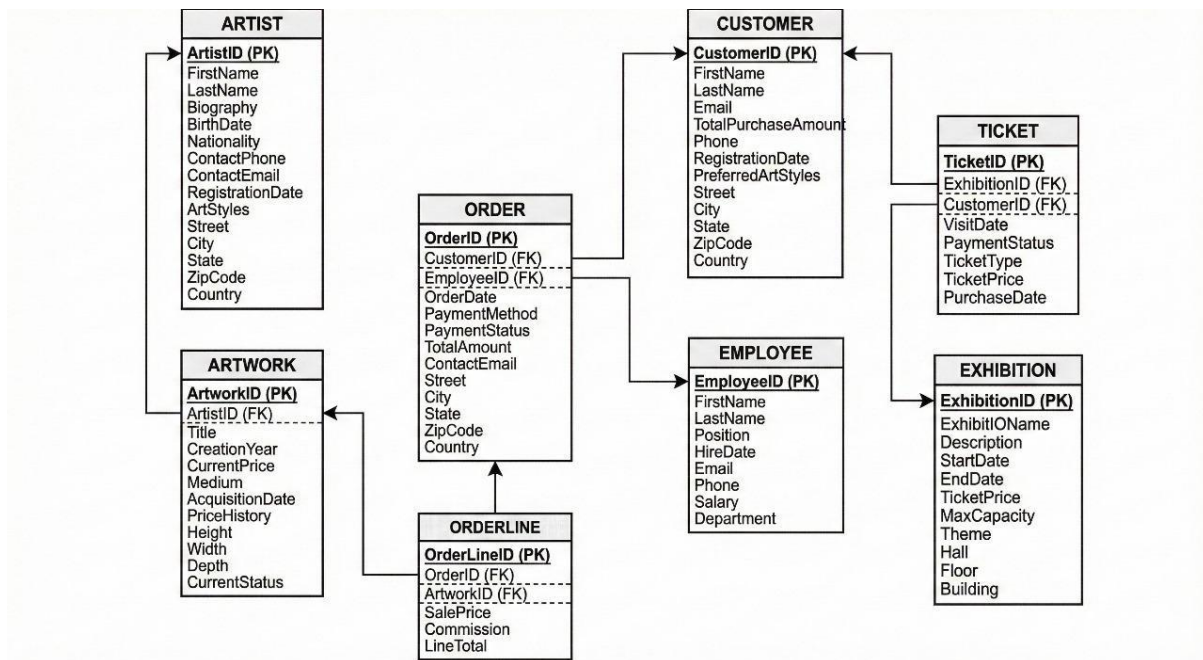
**Relationship Type:** Binary **Cardinality:** One-to-Many (1:N) **Description:** One Exhibition has many Tickets; each Ticket is for one Exhibition



#### 4.4. Complete ER Diagram (Before Normalization)



## 5. TRANSFORM ER INTO RELATIONAL MODEL



## 6. NORMALIZATION

### 1NF FORM:

Splitting artist ,customer and artworkinto separate tables because they contain multi valued attributes

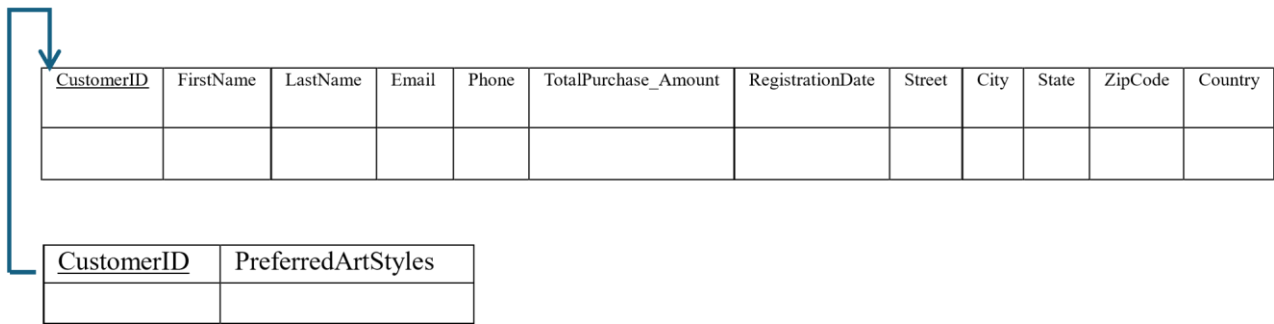
#### ARTIST

<u>ArtistID</u>	FirstName	LastName	Biography	BirthDate	Nationality	ContactPhone	ContactEmail	RegistrationDate	Street	City	State	ZipCode	Country

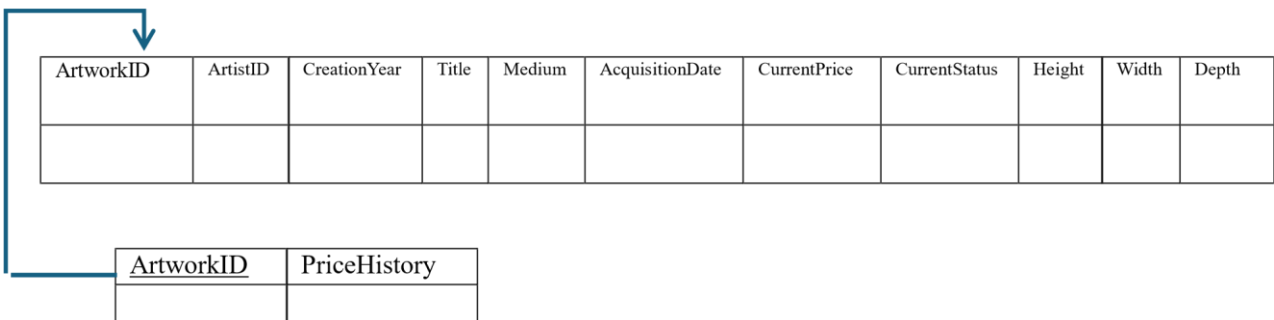
#### ARTIST\_ARTSTYLE

<u>ArtistID</u>	ArtStyle

#### CUSTOMER



## ARTWORK



## 2NF FORM:

We will **remove** partial dependencies from tables to get them in 2NF form. Our tables don't have partial dependencies so they are already in 2NF form.

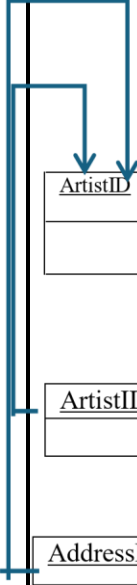
## 3NF FORM:

We will **remove** transitive dependencies(non-key attribute depending on another non key attribute) from tables.

## ARTIST:

In this case, ArtistID is the only key, so 3NF focuses on **removing derived/composite attributes**, such as the Address.

Thus, Address must be separated into its own table.



<u>ArtistID</u>	FirstName	LastName	Biography	BirthDate	Nationality	ContactPhone	ContactEmail	RegistratiopnDate

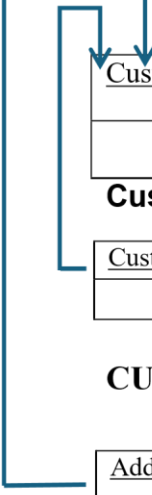
### ARTIST\_ARTSTYLE

<u>ArtistID</u>	ArtStyle

### ARTIST\_ADDRESS

<u>AddressID</u>	<u>ArtistID</u>	Street	City	State	ZipCode	Country

### CUSTOMER



<u>CustomerID</u>	FirstName	LastName	Email	Phone	TotalPurchase_Amount	RegistratiopnDate


### Customer\_Preference

<u>CustomerID</u>	PreferredArtStyles

### CUSTOMER\_Location

<u>AddressID</u>	<u>CustomerID</u>	Street	City	State	ZipCode	Country

### EXHIBITION




<u>ExhibitionID</u>	Description	Theme	ExhibitionName	StartDate	EndDate	TicketPrice	MaxCapacity

### EXHIBITION\_LOCATION

<u>ExhibitioID</u>	Hall	Floor	Building




## ARTWORK




<u>ArtworkID</u>	ArtistID	CreationYear	Title	Medium	AcquisitionDate	CurrentPrice	CurrentStatus

## ARTWORK\_RANGE




<u>ArtworkID</u>	PriceHistory

## ARTWORK\_DIMENSIONS




<u>ArtworkId</u>	Height	Width	Depth

## Orders



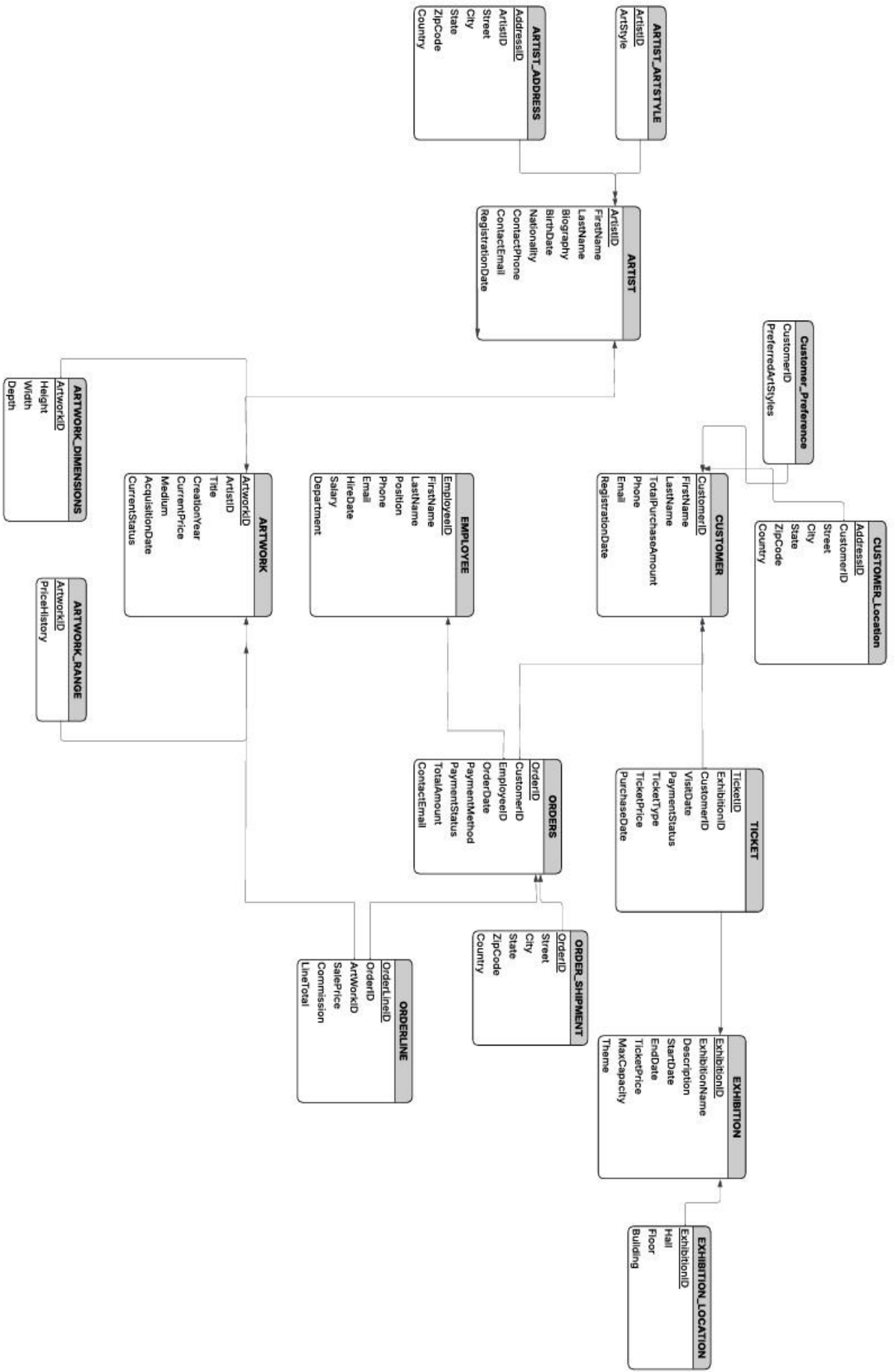
<u>OrderID</u>	EmployeeID	CustomerID	TotalAmount	PaymentMethod	PaymentStatus	OrderDate

## ORDER\_SHIPMENT



<u>AddressID</u>	<u>OrderID</u>	Street	City	State	ZipCode	Country

## Final Normalized Form:



## 7.Physical Model

**TABLE 1: ARTIST**

**Description:** Stores artist profile information and contact details

Column Name	Data Type	Size	PK	FK	Not Null	Unique	Default
ArtistID	INT	-	✓	-	✓	✓	AUTO_INCREMENT
FirstName	VARCHAR	50	-	-	✓	-	-
LastName	VARCHAR	50	-	-	✓	-	-
Biography	TEXT	-	-	-	-	-	NULL
BirthDate	DATE	-	-	-	-	-	NULL
Nationality	VARCHAR	50	-	-	-	-	NULL
ContactPhone	VARCHAR	20	-	-	-	-	NULL
ContactEmail	VARCHAR	100	-	-	✓	✓	-
RegistrationDate	TIMESTAMP	-	-	-	-	-	CURRENT_TIMESTAMP

### Indexes:

- PRIMARY KEY (ArtistID)
- INDEX idx\_artist\_lastname (LastName)
- INDEX idx\_artist\_email (ContactEmail)

**TABLE 2: ARTIST\_ARTSTYLE**

**Description:** Multivalued attribute - stores multiple art styles per artist

Column Name	Data Type	Size	PK	FK	Not Null	Unique	Default	Constraints
ArtistID	INT	-	✓	✓	✓	-	-	-
ArtStyle	VARCHAR	50	✓	-	✓	-	-	-

### Foreign Keys:

- ArtistID REFERENCES ARTIST(ArtistID) ON DELETE CASCADE ON UPDATE CASCADE

### Indexes:

- PRIMARY KEY (ArtistID, ArtStyle)

## TABLE 3: ARTIST\_ADDRESS

**Description:** One-to-one relationship - artist physical address

Column Name	Data Type	Size	PK	FK	Not Null	Unique	Default	Constraints
AddressID	INT	-	✓	-	✓	✓	AUTO_INCREMENT	-
ArtistID	INT	-	-	✓	✓	✓	-	-
Street	VARCHAR	100	-	-	-	-	NULL	-
City	VARCHAR	50	-	-	-	-	NULL	-
State	VARCHAR	50	-	-	-	-	NULL	-
ZipCode	VARCHAR	10	-	-	-	-	NULL	-
Country	VARCHAR	50	-	-	-	-	NULL	-

### Foreign Keys:

- ArtistID REFERENCES ARTIST(ArtistID) ON DELETE CASCADE ON UPDATE CASCADE

### Indexes:

- PRIMARY KEY (AddressID)
- UNIQUE (ArtistID)
- INDEX idx\_artist\_city (City)

## TABLE 4: CUSTOMER

**Description:** Stores customer information and registration details

Column Name	Data Type	Size	PK	FK	Not Null	Unique	Default
CustomerID	INT	-	✓	-	✓	✓	AUTO_INCREMENT
FirstName	VARCHAR	50	-	-	✓	-	-
LastName	VARCHAR	50	-	-	✓	-	-
Email	VARCHAR	100	-	-	✓	✓	-
Phone	VARCHAR	20	-	-	-	-	NULL
TotalPurchase_Amount	DECIMAL	12,2	-	-	-	-	0.00
RegistrationDate	TIMESTAMP	-	-	-	-	-	CURRENT_TIMESTAMP

#### Indexes:

- PRIMARY KEY (CustomerID)
- INDEX idx\_customer\_email (Email)
- INDEX idx\_customer\_lastname (LastName)

#### Constraints:

- CHECK (TotalPurchase\_Amount >= 0)

## TABLE 5: CUSTOMER\_PREFERENCE

**Description:** Multivalued attribute - customer preferred art styles

Column Name	Data Type	Size	PK	FK	Not Null	Unique	Default	Constraints
CustomerID	INT	-	✓	✓	✓	-	-	-
PreferredArtStyle	VARCHAR	50	✓	-	✓	-	-	-

#### Foreign Keys:

- CustomerID REFERENCES CUSTOMER(CustomerID) ON DELETE CASCADE ON UPDATE CASCADE

#### Indexes:

- PRIMARY KEY (CustomerID, PreferredArtStyle)

## TABLE 6: CUSTOMER\_LOCATION

**Description:** One-to-one relationship - customer address

Column Name	Data Type	Size	PK	FK	Not Null	Unique	Default	Constraints
AddressID	INT	-	✓	-	✓	✓	AUTO_INCREMENT	-
CustomerID	INT	-	-	✓	✓	✓	-	-
Street	VARCHAR	100	-	-	-	-	NULL	-
City	VARCHAR	50	-	-	-	-	NULL	-
State	VARCHAR	50	-	-	-	-	NULL	-
ZipCode	VARCHAR	10	-	-	-	-	NULL	-
Country	VARCHAR	50	-	-	-	-	NULL	-

### Foreign Keys:

- CustomerID REFERENCES CUSTOMER(CustomerID) ON DELETE CASCADE ON UPDATE CASCADE

### Indexes:

- PRIMARY KEY (AddressID)
- UNIQUE (CustomerID)
- INDEX idx\_customer\_city (City)

## TABLE 7: EMPLOYEE

**Description:** Stores employee information with hierarchical management structure



Column Name	Data Type	Size	PK	FK	Not Null	Unique	Default	Constraints
EmployeeID	INT	-	✓	-	✓	✓	AUTO_INCREMENT	-
FirstName	VARCHAR	50	-	-	✓	-	-	-
LastName	VARCHAR	50	-	-	✓	-	-	-
Email	VARCHAR	100	-	-	✓	✓	-	Valid email format
Phone	VARCHAR	20	-	-	-	-	NULL	-
Position	VARCHAR	50	-	-	-	-	NULL	-
HireDate	DATE	-	-	-	-	-	NULL	-
Salary	DECIMAL	10,2	-	-	-	-	NULL	> 0
Department	VARCHAR	50	-	-	-	-	NULL	-
ManagerID	INT	-	-	✓	-	-	NULL	-

#### Foreign Keys:

- ManagerID REFERENCES EMPLOYEE(EmployeeID) ON DELETE SET NULL ON UPDATE CASCADE (Self-referencing)

#### Indexes:

- PRIMARY KEY (EmployeeID)
- INDEX idx\_employee\_email (Email)
- INDEX idx\_employee\_manager (ManagerID) **Constraints:**
- CHECK (Salary > 0)

## TABLE 8: EXHIBITION

**Description:** Stores exhibition event information and scheduling

Column Name	Data Type	Size	PK	FK	Not Null	Unique	Default	Constraints
ExhibitionID	INT	-	✓	-	✓	✓	AUTO_INCREMENT	-
ExhibitionName	VARCHAR	100	-	-	✓	-	-	-
Description	TEXT	-	-	-	-	-	NULL	-
Theme	VARCHAR	100	-	-	-	-	NULL	-
StartDate	DATE	-	-	-	✓	-	-	-
EndDate	DATE	-	-	-	✓	-	-	EndDate >= StartDate
TicketPrice	DECIMAL	8,2	-	-	-	-	NULL	>= 0
MaxCapacity	INT	-	-	-	-	-	NULL	> 0

#### Indexes:

- PRIMARY KEY (ExhibitionID)
- INDEX idx\_exhibition\_dates (StartDate, EndDate)

#### Constraints:

- CHECK (EndDate >= StartDate)
- CHECK (TicketPrice >= 0)
- CHECK (MaxCapacity > 0)

## TABLE 9: EXHIBITION\_LOCATION

**Description:** One-to-one relationship - exhibition venue details

Column Name	Data Type	Size	PK	FK	Not Null	Unique	Default	Constraints
ExhibitionID	INT	-	✓	✓	✓	✓	-	-
Hall	VARCHAR	50	-	-	-	-	NULL	-
Floor	VARCHAR	20	-	-	-	-	NULL	-
Building	VARCHAR	50	-	-	-	-	NULL	-

#### Foreign Keys:

- ExhibitionID REFERENCES EXHIBITION(ExhibitionID) ON DELETE CASCADE ON UPDATE CASCADE

#### Indexes:

- PRIMARY KEY (ExhibitionID)

## TABLE 10: ARTWORK

**Description:** Stores artwork catalog and inventory information

Column Name	Data Type	Size	PK	FK	Not Null	Unique	Default	Constraints
ArtworkID	INT	-	✓	-	✓	✓	AUTO_INCREMENT	-
ArtistID	INT	-	-	✓	✓	-	-	-
Title	VARCHAR	100	-	-	✓	-	-	-
CreationYear	INT	-	-	-	-	-	NULL	1000-Current Year
Medium	VARCHAR	50	-	-	-	-	NULL	-
AcquisitionDate	DATE	-	-	-	-	-	NULL	-
CurrentPrice	DECIMAL	12,2	-	-	-	-	NULL	>= 0
CurrentStatus	ENUM	-	-	-	-	-	'Available'	Available/Sold Exhibition
Style	VARCHAR	50	-	-	-	-	NULL	-

### Foreign Keys:

- ArtistID REFERENCES ARTIST(ArtistID) ON DELETE RESTRICT ON UPDATE CASCADE

### Indexes:

- PRIMARY KEY (ArtworkID)
- INDEX idx\_artwork\_artist (ArtistID)
- INDEX idx\_artwork\_status (CurrentStatus)
- INDEX idx\_artwork\_title (Title)

### Constraints:

- CHECK (CurrentPrice >= 0)
- CHECK (CreationYear >= 1000 AND CreationYear <= YEAR(CURRENT\_DATE))

## TABLE 11: ARTWORK\_DIMENSIONS

**Description:** One-to-one relationship - artwork physical measurements

Column Name	Data Type	Size	PK	FK	Not Null	Unique	Default	Constraints
ArtworkID	INT	-	✓	✓	✓	✓	-	-
Height	DECIMAL	8,2	-	-	-	-	NULL	> 0
Width	DECIMAL	8,2	-	-	-	-	NULL	> 0
Depth	DECIMAL	8,2	-	-	-	-	NULL	>= 0

### Foreign Keys:

- ArtworkID REFERENCES ARTWORK(ArtworkID) ON DELETE CASCADE ON UPDATE CASCADE

### Indexes:

- PRIMARY KEY (ArtworkID)

### Constraints:

- CHECK (Height > 0)
- CHECK (Width > 0)
- CHECK (Depth >= 0)

## TABLE 12: ARTWORK\_RANGE

**Description:** Tracks historical price changes for artworks

Column Name	Data Type	Size	PK	FK	Not Null	Unique	Default	Constraints
ArtworkID	INT	-	-	✓	✓	-	-	-
Price	DECIMAL	12,2	-	-	✓	-	-	>= 0

### Foreign Keys:

- ArtworkID REFERENCES ARTWORK(ArtworkID) ON DELETE CASCADE ON UPDATE CASCADE

**Indexes:**

- PRIMARY KEY (PriceHistoryID)
- INDEX idx\_pricehistory\_artwork (ArtworkID)
- INDEX idx\_pricehistory\_date (EffectiveDate)

**Constraints:**

- CHECK (Price >= 0)

## TABLE 13: ORDERS

**Description:** Stores customer purchase orders

Column Name	Data Type	Size	PK	FK	Not Null	Unique	Default
OrderID	INT	-	✓	-	✓	✓	AUTO_INCREMENT
CustomerID	INT	-	-	✓	✓	-	-
EmployeeID	INT	-	-	✓	✓	-	-
OrderDate	TIMESTAMP	-	-	-	-	-	CURRENT_TIMESTAMP
TotalAmount	DECIMAL	12,2	-	-	-	-	0.00
PaymentMethod	VARCHAR	50	-	-	-	-	NULL
PaymentStatus	ENUM	-	-	-	-	-	'Pending'

**Foreign Keys:**

- CustomerID REFERENCES CUSTOMER(CustomerID) ON DELETE RESTRICT ON UPDATE CASCADE
- EmployeeID REFERENCES EMPLOYEE(EmployeeID) ON DELETE RESTRICT ON UPDATE CASCADE

**Indexes:**

- PRIMARY KEY (OrderID)
- INDEX idx\_order\_customer (CustomerID)
- INDEX idx\_order\_employee (EmployeeID)
- INDEX idx\_order\_date (OrderDate)

**Constraints:**

- CHECK (TotalAmount >= 0)

## TABLE 14: ORDER\_SHIPMENT

**Description:** One-to-one relationship - order delivery address

Column Name	Data Type	Size	PK	FK	Not Null	Unique	Default	Constraints
AddressID	INT	-	✓	-	✓	✓	AUTO_INCREMENT	-
OrderID	INT	-	-	✓	✓	✓	-	-
Street	VARCHAR	100	-	-	-	-	NULL	-
City	VARCHAR	50	-	-	-	-	NULL	-
State	VARCHAR	50	-	-	-	-	NULL	-
ZipCode	VARCHAR	10	-	-	-	-	NULL	-
Country	VARCHAR	50	-	-	-	-	NULL	-

### Foreign Keys:

- OrderID REFERENCES ORDERS(OrderID) ON DELETE CASCADE ON UPDATE CASCADE

### Indexes:

- PRIMARY KEY (AddressID)
- UNIQUE (OrderID)

## TABLE 15: ORDERLINE

**Description:** Associative entity - order line items with sale details

Column Name	Data Type	Size	PK	FK	Not Null	Unique	Default	Constraints
OrderLineID	INT	-	✓	-	✓	✓	AUTO_INCREMENT	-
OrderID	INT	-	-	✓	✓	-	-	-
ArtworkID	INT	-	-	✓	✓	-	-	-
SalePrice	DECIMAL	12,2	-	-	✓	-	-	> 0
Commission	DECIMAL	12,2	-	-	-	-	0.00	>= 0
LineTotal	DECIMAL	12,2	-	-	-	-	COMPUTED	SalePrice + Commission



**Foreign Keys:**

- OrderID REFERENCES ORDERS(OrderID) ON DELETE CASCADE ON UPDATE CASCADE
- ArtworkID REFERENCES ARTWORK(ArtworkID) ON DELETE RESTRICT ON UPDATE CASCADE

**Indexes:**

- PRIMARY KEY (OrderLineID)
  - INDEX idx\_orderline\_order (OrderID)
  - INDEX idx\_orderline\_artwork (ArtworkID)
- Constraints:**
- CHECK (SalePrice > 0)
  - CHECK (Commission >= 0)

## TABLE 16: TICKET

**Description:** Stores exhibition ticket sales and visitor information

Column Name	Data Type	Size	PK	FK	Not Null	Unique	Default
TicketID	INT	-	✓	-	✓	✓	AUTO_INCREMENT
ExhibitionID	INT	-	-	✓	✓	-	-
CustomerID	INT	-	-	✓	✓	-	-
PurchaseDate	TIMESTAMP	-	-	-	-	-	CURRENT_TIMESTAMP
VisitDate	DATE	-	-	-	-	-	NULL
TicketPrice	DECIMAL	8,2	-	-	✓	-	-
TicketType	ENUM	-	-	-	-	-	'Adult'
PaymentStatus	ENUM	-	-	-	-	-	'Pending'

**Foreign Keys:**

- ExhibitionID REFERENCES EXHIBITION(ExhibitionID) ON DELETE RESTRICT ON UPDATE CASCADE

- CustomerID REFERENCES CUSTOMER(CustomerID) ON DELETE RESTRICT ON UPDATE CASCADE

#### Indexes:

- PRIMARY KEY (TicketID)
- INDEX idx\_ticket\_exhibition (ExhibitionID)
- INDEX idx\_ticket\_customer (CustomerID)
- INDEX idx\_ticket\_visitdate (VisitDate) **Constraints:**
- CHECK (TicketPrice > 0)

## 8. SQL IMPLEMENTATION:

### -- Art Gallery Database Schema

```
CREATE DATABASE ArtGalleryDB;
USE ArtGalleryDB;
```

```
-- Create ARTIST table
```

```
CREATE TABLE ARTIST (
    ArtistID INT PRIMARY KEY IDENTITY(1,1),
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Biography TEXT,
    BirthDate DATE,
    Nationality VARCHAR(50),
    ContactPhone VARCHAR(20),
    ContactEmail VARCHAR(100),
    RegistrationDate DATE
);
```

```
-- Create CUSTOMER table
```

```
CREATE TABLE CUSTOMER (
    CustomerID INT PRIMARY KEY IDENTITY(1,1),
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    TotalPurchaseAmount DECIMAL(10,2),
    Phone VARCHAR(20),
    Email VARCHAR(100),
    RegistrationDate DATE
);
```

```

-- Create EMPLOYEE table
CREATE TABLE EMPLOYEE (
    EmployeeID INT PRIMARY KEY IDENTITY(1,1),
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Position VARCHAR(50),
    Phone VARCHAR(20),
    Email VARCHAR(100),
    HireDate DATE,
    Salary DECIMAL(10,2),
    Department VARCHAR(50)
);

-- Create EXHIBITION table
CREATE TABLE EXHIBITION (
    ExhibitionID INT PRIMARY KEY IDENTITY(1,1),
    ExhibitionName VARCHAR(100),
    Description TEXT,
    StartDate DATE,
    EndDate DATE,
    TicketPrice DECIMAL(10,2),
    MaxCapacity INT,
    Theme VARCHAR(100)
);

-- Create ARTWORK table
CREATE TABLE ARTWORK (
    ArtworkID INT PRIMARY KEY IDENTITY(1,1),
    Title VARCHAR(100),
    CreationYear INT,
    CurrentPrice DECIMAL(10,2),
    Medium VARCHAR(50),
    AcquisitionDate DATE,
    CurrentStatus VARCHAR(50),
    FOREIGN KEY (ArtistID) REFERENCES ARTIST(ArtistID)
);

-- Create ORDERS table
CREATE TABLE ORDERS (
    OrderID INT PRIMARY KEY IDENTITY(1,1),
    CustomerID INT,
    EmployeeID INT,
    OrderDate DATE,
    PaymentMethod VARCHAR(50),
    PaymentStatus VARCHAR(50),
    TotalAmount DECIMAL(10,2),
    ContactEmail VARCHAR(100),
    FOREIGN KEY (CustomerID) REFERENCES CUSTOMER(CustomerID),
    FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE(EmployeeID)
);

-- Create ORDERLINE table
CREATE TABLE ORDERLINE (
    OrderLineID INT PRIMARY KEY IDENTITY(1,1),
    OrderID INT,
    ArtworkID INT,
    SalePrice DECIMAL(10,2),
    Commission DECIMAL(10,2),

```

```

        LineTotal DECIMAL(10,2),
        FOREIGN KEY (OrderID) REFERENCES ORDERS(OrderID),
        FOREIGN KEY (ArtWorkID) REFERENCES ARTWORK(ArtworkID)
    );

-- Create TICKET table
CREATE TABLE TICKET (
    TicketID INT PRIMARY KEY IDENTITY(1,1),
    ExhibitionID INT,
    CustomerID INT,
    VisitDate DATE,
    PaymentStatus VARCHAR(50),
    TicketType VARCHAR(50),
    TicketPrice DECIMAL(10,2),
    PurchaseDate DATE,
    FOREIGN KEY (ExhibitionID) REFERENCES EXHIBITION(ExhibitionID),
    FOREIGN KEY (CustomerID) REFERENCES CUSTOMER(CustomerID)
);

-- Create ARTIST_ARTSTYLE table
CREATE TABLE ARTIST_ARTSTYLE (
    ArtistID INT,
    ArtStyle VARCHAR(50),
    PRIMARY KEY (ArtistID, ArtStyle),
    FOREIGN KEY (ArtistID) REFERENCES ARTIST(ArtistID)
);

-- Create ARTIST_ADDRESS table
CREATE TABLE ARTIST_ADDRESS (
    AddressID INT PRIMARY KEY IDENTITY(1,1),
    ArtistID INT,
    Street VARCHAR(100),
    City VARCHAR(50),
    State VARCHAR(50),
    ZipCode VARCHAR(20),
    Country VARCHAR(50),
    FOREIGN KEY (ArtistID) REFERENCES ARTIST(ArtistID)
);

-- Create Customer_Preference table
CREATE TABLE Customer_Preference (
    CustomerID INT PRIMARY KEY,
    PreferredArtStyles VARCHAR(200),
    FOREIGN KEY (CustomerID) REFERENCES CUSTOMER(CustomerID)
);

-- Create CUSTOMER_Location table
CREATE TABLE CUSTOMER_Location (
    AddressID INT PRIMARY KEY IDENTITY(1,1),
    CustomerID INT,
    Street VARCHAR(100),
    City VARCHAR(50),
    State VARCHAR(50),
    ZipCode VARCHAR(20),
    Country VARCHAR(50),
    FOREIGN KEY (CustomerID) REFERENCES CUSTOMER(CustomerID)
);

-- Create EXHIBITION_LOCATION table

```

```
CREATE TABLE EXHIBITION_LOCATION (
    ExhibitionID INT PRIMARY KEY,
    Hall VARCHAR(50),
    Floor VARCHAR(20),
    Building VARCHAR(50),
    FOREIGN KEY (ExhibitionID) REFERENCES EXHIBITION(ExhibitionID)
);
```

```
-- Create ARTWORK_RANGE table
CREATE TABLE ARTWORK_RANGE (
    ArtworkID INT PRIMARY KEY,
    PriceHistory TEXT,
    FOREIGN KEY (ArtworkID) REFERENCES ARTWORK(ArtworkID)
);
```

```
-- Create ARTWORK_DIMENSIONS table
CREATE TABLE ARTWORK_DIMENSIONS (
    ArtworkID INT PRIMARY KEY,
    Height DECIMAL(10,2),
    Width DECIMAL(10,2),
    Depth DECIMAL(10,2),
    FOREIGN KEY (ArtworkID) REFERENCES ARTWORK(ArtworkID)
);
```

```
-- Create ORDER_SHIPMENT table
CREATE TABLE ORDER_SHIPMENT (
    OrderID INT PRIMARY KEY,
    Street VARCHAR(100),
    City VARCHAR(50),
    State VARCHAR(50),
    ZipCode VARCHAR(20),
    Country VARCHAR(50),
    FOREIGN KEY (OrderID) REFERENCES ORDERS(OrderID)
);
```

GO

-- STORED PROCEDURES

-- 1. Add New Artist

```
CREATE PROCEDURE sp_AddArtist
    @FirstName VARCHAR(50),
    @LastName VARCHAR(50),
    @Biography TEXT,
    @BirthDate DATE,
    @Nationality VARCHAR(50),
    @ContactPhone VARCHAR(20),
    @ContactEmail VARCHAR(100)
AS
BEGIN
    INSERT INTO ARTIST (FirstName, LastName, Biography, BirthDate,
        Nationality, ContactPhone, ContactEmail, RegistrationDate)
    VALUES (@FirstName, @LastName, @Biography, @BirthDate,
        @Nationality, @ContactPhone, @ContactEmail, GETDATE());
END;
GO
```

-- 2. Add New Artwork

```

CREATE PROCEDURE sp_AddArtwork
@ArtistID INT,
@Title VARCHAR(50),
@CreationYear INT,
@CurrentPrice DECIMAL(10,2),
@Medium VARCHAR(50),
@CurrentStatus VARCHAR(50)
AS
BEGIN
INSERT INTO ARTWORK (ArtistID, Title, CreationYear,
CurrentPrice, Medium, AcquisitionDate, CurrentStatus)
VALUES (@ArtistID, @Title, @CreationYear, @CurrentPrice,
@Medium, GETDATE(), @CurrentStatus);
END;
GO

-- 3. Create New Order
CREATE PROCEDURE sp_CreateOrder
@CustomerID INT,
@EmployeeID INT,
@PaymentMethod VARCHAR(50),
@ContactEmail VARCHAR(100)
AS
BEGIN
INSERT INTO ORDERS (CustomerID, EmployeeID, OrderDate,
PaymentMethod, PaymentStatus, TotalAmount, ContactEmail)
VALUES (@CustomerID, @EmployeeID, GETDATE(), @PaymentMethod,
'Pending', 0.00, @ContactEmail);

SELECT SCOPE_IDENTITY() AS NewOrderID;
END;
GO

-- 4. Add Order Line Item
CREATE PROCEDURE sp_AddOrderLine
@OrderID INT,
@ArtworkID INT,
@SalePrice DECIMAL(10,2),
@CommissionRate DECIMAL(5,2)
AS
BEGIN
DECLARE @Commission DECIMAL(10,2);
DECLARE @LineTotal DECIMAL(10,2);

SET @Commission = @SalePrice * (@CommissionRate / 100);
SET @LineTotal = @SalePrice + @Commission;

INSERT INTO ORDERLINE (OrderID, ArtWorkID, SalePrice,
Commission, LineTotal)
VALUES (@OrderID, @ArtworkID, @SalePrice, @Commission,
@LineTotal);
END;
GO

-- 5. Register Customer

```

```

CREATE PROCEDURE sp_RegisterCustomer
    @FirstName VARCHAR(50),
    @LastName VARCHAR(50),
    @Phone VARCHAR(20),
    @Email VARCHAR(100)
AS
BEGIN
    INSERT INTO CUSTOMER (FirstName, LastName, TotalPurchaseAmount,
    Phone, Email, RegistrationDate)
    VALUES (@FirstName, @LastName, 0.00, @Phone, @Email, GETDATE());

    SELECT SCOPE_IDENTITY() AS NewCustomerID;
END;
GO

-- 6. Purchase Ticket
CREATE PROCEDURE sp_PurchaseTicket
    @ExhibitionID INT,
    @CustomerID INT,
    @VisitDate DATE,
    @TicketType VARCHAR(50)
AS
BEGIN
    DECLARE @TicketPrice DECIMAL(10,2);

    SELECT @TicketPrice = TicketPrice
    FROM EXHIBITION
    WHERE ExhibitionID = @ExhibitionID;

    INSERT INTO TICKET (ExhibitionID, CustomerID, VisitDate,
    PaymentStatus, TicketType, TicketPrice, PurchaseDate)
    VALUES (@ExhibitionID, @CustomerID, @VisitDate, 'Paid',
    @TicketType, @TicketPrice, GETDATE());
END;
GO

-- 7. Get Artist Artworks
CREATE PROCEDURE sp_GetArtistArtworks
    @ArtistID INT
AS
BEGIN
    SELECT * FROM ARTWORK WHERE ArtistID = @ArtistID;
END;
GO

-- 8. Get Customer Order History
CREATE PROCEDURE sp_GetCustomerOrders
    @CustomerID INT
AS
BEGIN
    SELECT O.*, OL.ArtworkID, OL.SalePrice, A.Title
    FROM ORDERS O
    JOIN ORDERLINE OL ON O.OrderID = OL.OrderID
    JOIN ARTWORK A ON OL.ArtworkID = A.ArtworkID
    WHERE O.CustomerID = @CustomerID;
END;
GO

-- 9. Update Artwork Status

```



```

CREATE
PROCEDURE sp_UpdateArtworkStatus
@ArtworkID INT,
@NewStatus VARCHAR(50)
AS
BEGIN
UPDATE ARTWORK
SET CurrentStatus = @NewStatus
WHERE ArtworkID = @ArtworkID;
END;
GO

-- 10. Get Exhibition Statistics
CREATE PROCEDURE sp_GetExhibitionStats
@ExhibitionID INT
AS
BEGIN
SELECT
    E.ExhibitionName,
    COUNT(T.TicketID) AS TotalTicketsSold,
    SUM(T.TicketPrice) AS TotalRevenue,
    E.MaxCapacity,
    (E.MaxCapacity - COUNT(T.TicketID)) AS RemainingCapacity
FROM EXHIBITION E
LEFT JOIN TICKET T ON E.ExhibitionID = T.ExhibitionID
WHERE E.ExhibitionID = @ExhibitionID
GROUP BY E.ExhibitionName, E.MaxCapacity;
END;
GO

```

## -- TRIGGERS

```

-- 1. Update Order Total After OrderLine Insert
CREATE TRIGGER trg_UpdateOrderTotal
ON ORDERLINE
AFTER INSERT
AS
BEGIN
UPDATE ORDERS
SET TotalAmount = (
    SELECT SUM(LineTotal)
    FROM ORDERLINE
    WHERE OrderID = i.OrderID
)
FROM ORDERS O
INNER JOIN inserted i ON O.OrderID = i.OrderID;
END;
GO

-- 2. Update Customer Total Purchase Amount
CREATE TRIGGER trg_UpdateCustomerPurchaseAmount
ON ORDERS
AFTER UPDATE
AS
BEGIN
IF UPDATE(PaymentStatus)
BEGIN

```

```

        UPDATE CUSTOMER
        SET TotalPurchaseAmount = TotalPurchaseAmount +
i.TotalAmount
        FROM CUSTOMER C
        INNER JOIN inserted i ON C.CustomerID = i.CustomerID
        INNER JOIN deleted d ON i.OrderID = d.OrderID
        WHERE i.PaymentStatus = 'Completed' AND d.PaymentStatus <>
'Completed';
    END;
END;
GO

```

```

-- 3. Update Artwork Status on Sale
CREATE TRIGGER trg_UpdateArtworkOnSale
ON ORDERLINE
AFTER INSERT
AS
BEGIN

```

```

    UPDATE ARTWORK
    SET CurrentStatus = 'Sold'
    FROM ARTWORK A
    INNER JOIN inserted i ON A.ArtworkID = i.ArtworkID;
END;
GO

```

```

-- 4. Prevent Order Deletion if Completed
CREATE TRIGGER trg_PreventCompletedOrderDeletion
ON ORDERS
INSTEAD OF DELETE
AS
BEGIN

```

```

    IF EXISTS (SELECT 1 FROM deleted WHERE PaymentStatus =
'Completed')
    BEGIN
        RAISERROR ('cannot delete completed orders', 16, 1);
        ROLLBACK TRANSACTION;
    END
    ELSE
    BEGIN
        DELETE FROM ORDERS WHERE OrderID IN (SELECT OrderID FROM
deleted);    END;
END;
GO

```

```

-- 5. Log Price Changes in Artwork
CREATE TRIGGER trg_LogArtworkPriceChange
ON ARTWORK
AFTER UPDATE
AS
BEGIN

```

```

    IF UPDATE(CurrentPrice)
    BEGIN
        UPDATE ARTWORK_RANGE
        SET PriceHistory = CONCAT(
            ISNULL(PriceHistory, ''),
            FORMAT(GETDATE(), 'yyyy-MM-dd'),
            ': $',
            CAST(i.CurrentPrice AS VARCHAR),
            '; '
        )
    END
END

```

```

        FROM ARTWORK_RANGE AR
        INNER JOIN inserted i ON AR.ArtworkID = i.ArtworkID;
    END;
END;
GO

-- 6. Validate Exhibition Capacity
CREATE TRIGGER trg_validateTicketCapacity
ON TICKET
INSTEAD OF INSERT
AS
BEGIN
    DECLARE @ExhibitionID INT;
    DECLARE @MaxCapacity INT;
    DECLARE @CurrentTickets INT;

    SELECT @ExhibitionID = ExhibitionID FROM inserted;

    SELECT @MaxCapacity = MaxCapacity FROM EXHIBITION WHERE
    ExhibitionID = @ExhibitionID;
    SELECT @CurrentTickets = COUNT(*) FROM TICKET WHERE ExhibitionID
    = @ExhibitionID;

    IF @CurrentTickets >= @MaxCapacity
    BEGIN
        RAISERROR ('Exhibition is at full capacity', 16, 1);
        ROLLBACK TRANSACTION;
    END
    ELSE
    BEGIN
        INSERT INTO TICKET (ExhibitionID, CustomerID, VisitDate,
        PaymentStatus, TicketType, TicketPrice, PurchaseDate)
        SELECT ExhibitionID, CustomerID, VisitDate, PaymentStatus,
        TicketType, TicketPrice, PurchaseDate
        FROM inserted;
    END;
END;
GO

```

## -- USER ACCESS & CONTROL

```

-- Create Database Roles
CREATE ROLE GalleryAdmin;
CREATE ROLE GalleryManager;
CREATE ROLE SalesEmployee;
CREATE ROLE CustomerService;
CREATE ROLE ReadOnlyUser;
GO

```

```

-- 1. GALLERY ADMIN - Full Access
GRANT CONTROL ON DATABASE::ArtGalleryDB TO GalleryAdmin; GO

```

```

-- 2. GALLERY MANAGER - Manage exhibitions, artworks, and view reports
GRANT          , INSERT, SELECTUPDATE ON ARTIST TO GalleryManager;
GRANT          , INSERT, SELECTUPDATE ON ARTWORK TO GalleryManager;
GRANT          , INSERT, SELECTUPDATE ON EXHIBITION TO GalleryManager;
GRANT SELECT ON ORDERS TO GalleryManager;

```

```

GRANT SELECT ON CUSTOMER TO GalleryManager;
GRANT SELECT ON EMPLOYEE TO GalleryManager;
GRANT EXECUTE ON sp_AddArtist TO GalleryManager;
GRANT EXECUTE ON sp_AddArtwork TO GalleryManager;
GRANT EXECUTE ON sp_GetExhibitionStats TO GalleryManager;
GO

```

-- 3. SALES EMPLOYEE - Process orders and manage customers

```

GRANT SELECT ON CUSTOMER TO SalesEmployee;
GRANT SELECT ON ARTIST TO SalesEmployee;
GRANT SELECT, INSERT ON ORDERLINE TO SalesEmployee;
GRANT SELECT, INSERT, UPDATE ON CUSTOMER TO SalesEmployee;
GRANT SELECT, INSERT, UPDATE ON ORDERS TO SalesEmployee;
GRANT EXECUTE, INSERT ON ORDERLINE TO SalesEmployee;
GRANT EXECUTE, INSERT ON ORDERLINE TO SalesEmployee;
GRANT EXECUTE, INSERT ON ORDERLINE TO SalesEmployee;
GO

```

```

-- 4.
GRANT EXECUTE ON sp_CreateOrder TO SalesEmployee;
GRANT EXECUTE ON sp_AddOrderLine TO SalesEmployee;
GRANT EXECUTE ON sp_RegisterCustomer TO SalesEmployee;
GRANT EXECUTE ON sp_PurchaseTicket TO SalesEmployee;

```

CUSTOMER  
SERVICE  
view and  
assist

customers

```

GRANT SELECT ON CUSTOMER TO CustomerService;
GRANT SELECT ON ORDERS TO CustomerService;
GRANT SELECT ON ORDERLINE TO CustomerService;
GRANT SELECT ON TICKET TO CustomerService;
GRANT SELECT ON EXHIBITION TO CustomerService;
GRANT EXECUTE ON sp_GetCustomerOrders TO CustomerService;
GO

```

-- 5. READ ONLY USER - View public information

```

GRANT SELECT ON ARTIST TO ReadOnlyUser;
GRANT SELECT ON ARTWORK TO ReadOnlyUser;
GRANT SELECT ON EXHIBITION TO ReadOnlyUser;
GO

```

-- Create Example Users and Assign Roles

-- Admin User

```

CREATE LOGIN AdminUser WITH PASSWORD = 'Admin@123';
CREATE USER AdminUser FOR LOGIN AdminUser;
ALTER ROLE GalleryAdmin ADD MEMBER AdminUser;
GO

```

-- Manager User

```

CREATE LOGIN ManagerUser WITH PASSWORD = 'Manager@123';
CREATE USER ManagerUser FOR LOGIN ManagerUser;
ALTER ROLE GalleryManager ADD MEMBER ManagerUser;
GO

```

-- Sales User

```

CREATE LOGIN SalesUser WITH PASSWORD = 'Sales@123';

```

```

CREATE USER SalesUser FOR LOGIN SalesUser;
ALTER ROLE SalesEmployee ADD MEMBER SalesUser;
GO

-- Customer Service User
CREATE LOGIN CSUser WITH PASSWORD = 'CS@123';
CREATE USER CSUser FOR LOGIN CSUser;
ALTER ROLE CustomerService ADD MEMBER CSUser;
GO

-- Read Only User
CREATE LOGIN GuestUser WITH PASSWORD = 'Guest@123';
CREATE USER GuestUser FOR LOGIN GuestUser;
ALTER ROLE ReadOnlyUser ADD MEMBER GuestUser;
GO

-- DENY SPECIFIC PERMISSIONS FOR SECURITY

-- Prevent Sales from deleting orders
DENY DELETE ON ORDERS TO SalesEmployee;
GO

-- Prevent Customer Service from modifying data
DENY UPDATE, INSERT, DELETE ON CUSTOMER TO CustomerService;
DENY UPDATE, INSERT, DELETE ON ORDERS TO CustomerService;
GO

-- Prevent ReadOnly from accessing sensitive data
DENY SELECT ON EMPLOYEE TO ReadOnlyUser;
DENY SELECT ON ORDERS TO ReadOnlyUser;
DENY SELECT ON CUSTOMER TO ReadOnlyUser;
GO

```

#### OUTPUTS:

```

SELECT * FROM ARTIST;
SELECT * FROM CUSTOMER;
SELECT * FROM EMPLOYEE;
SELECT * FROM EXHIBITION;
SELECT * FROM ARTWORK;
SELECT * FROM ORDERS;
SELECT * FROM ORDERLINE;
SELECT * FROM TICKET;
SELECT * FROM ARTIST_ARTSTYLE;
SELECT * FROM ARTIST_ADDRESS;
SELECT * FROM Customer_Preference;
SELECT * FROM CUSTOMER_Location;
SELECT * FROM EXHIBITION_LOCATION;

```

SELECT \* FROM ARTWORK\_RANGE;  
 SELECT \* FROM ARTWORK\_DIMENSIONS;  
 SELECT \* FROM ORDER\_SHIPMENT;

	ArtistID	FirstName	LastName	Biography	BirthDate	Nationality	ContactPhone	ContactEmail	RegistrationDate
1	1	Ahmed	Khan	Contemporary calligraphy and Islamic art special...	1985-03-15	Pakistani	0300-1234501	ahmed.khan@email.com	2023-01-10
2	2	Fatima	Ahmed	Miniature painter following Mughal traditions	1990-07-22	Pakistani	0321-1234502	fatima.ahmed@email.com	2023-02-15
3	3	Ali	Hassan	Modern landscape and cityscape artist	1988-11-30	Pakistani	0333-1234503	ali.hassan@email.com	2023-03-20
4	4	Ayesha	Malik	Portrait and figurative painter	1982-05-18	Pakistani	0301-1234504	ayesha.malik@email.com	2023-04-05
5	5	Usman	Raza	Mixed media and installation artist	1995-09-12	Pakistani	0345-1234505	usman.raza@email.com	2023-05-18
6	6	Zainab	Siddiqui	Contemporary textile and embroidery artist	1987-12-08	Pakistani	0312-1234506	zainab.siddiqui@email.com	2023-06-22
7	7	Hamza	Butt	Abstract painter and digital artist	1980-04-25	Pakistani	0335-1234507	hamza.butt@email.com	2023-07-30
8	8	Maryam	Iqbal	Watercolor artist specializing in nature	1992-08-14	Pakistani	0303-1234508	maryam.iqbal@email.com	2023-08-12

	CustomerID	FirstName	LastName	TotalPurchaseAmount	Phone	Email	RegistrationDate
1	1	Muhammad	Asif	450000.00	0300-2345601	muhammad.asif@email.com	2023-01-15
2	2	Hira	Saleem	285000.00	0321-2345602	hira.saleem@email.com	2023-02-20
3	3	Imran	Haider	680000.00	0333-2345603	imran.haider@email.com	2023-03-10
4	4	Sana	Tariq	175000.00	0301-2345604	sana.tariq@email.com	2023-04-18
5	5	Kamran	Bashir	395000.00	0345-2345605	kamran.bashir@email.com	2023-05-22
6	6	Nadia	Rehman	520000.00	0312-2345606	nadia.rehman@email.com	2023-06-15
7	7	Faisal	Mahmood	125000.00	0335-2345607	faisal.mahmood@email.com	2023-07-08
8	8	Rabia	Nawaz	310000.00	0303-2345608	rabia.nawaz@email.com	2023-08-25

	EmployeeID	FirstName	LastName	Position	Phone	Email	HireDate	Salary	Department
1	1	Hassan	Mirza	Gallery Manager	0300-3456701	hassan.mirza@gallery.com	2020-01-15	95000.00	Management
2	2	Saima	Arif	Sales Associate	0321-3456702	saima.arif@gallery.com	2021-03-20	55000.00	Sales
3	3	Tariq	Hussain	Curator	0333-3456703	tariq.hussain@gallery.com	2019-06-10	85000.00	Curatorial
4	4	Zara	Noor	Customer Servi...	0301-3456704	zara.noor@gallery.com	2022-02-14	48000.00	Customer S...
5	5	Waqas	Ahmed	Sales Associate	0345-3456705	waqas.ahmed@gallery.c...	2021-08-22	55000.00	Sales
6	6	Mahira	Khan	Marketing Man...	0312-3456706	mahira.khan@gallery.com	2020-05-18	78000.00	Marketing
7	7	Shahid	Malik	Exhibition Coor...	0335-3456707	shahid.malik@gallery.com	2021-11-30	65000.00	Operations
8	8	Aiza	Rauf	Sales Associate	0303-3456708	aiza.rauf@gallery.com	2022-07-12	55000.00	Sales

	ExhibitionID	ExhibitionName	Description	StartDate	EndDate	TicketPrice	MaxCapacity	Theme
3	3	Digital Tehzeeb	Modern digital art exploring cultural fusion	2024-03-10	2024-05-10	2500.00	180	Digital Art
4	4	Mitti ki Roshni	Sculptural works in clay and ceramic	2024-04-05	2024-06-05	2000.00	120	Sculpture
5	5	Khat-o-Kitabat	Islamic calligraphy and typography exhibition	2024-05-15	2024-07-15	2200.00	220	Calligraphy
6	6	Chehre Pakista...	Portrait exhibition featuring Pakistani personalities	2024-06-20	2024-08-20	1800.00	160	Portrait
7	7	Shehr-e-Lahore	Urban art celebrating Lahore culture	2024-07-10	2024-09-10	2100.00	190	Urban Art
8	8	Mughal Virsa	Miniature paintings following Mughal tradition	2024-08-15	2024-10-15	2300.00	140	Miniature
9	9	Rang aur Roshni	Mixed media exploring light and color	2024-09-20	2024-11-20	2400.00	170	Mixed Media
10	10	Roshan Musta...	Contemporary works envisioning Pakistan future	2024-10-25	2024-12-25	2600.00	200	Contemporary

	ArtworkID	ArtistID	Title	CreationYear	CurrentPrice	Medium	AcquisitionDate	CurrentStatus
1	1	1	Ayat-ul-Kursi	2023	180000.00	Calligraphy on Canvas	2023-02-15	Sold
2	2	2	Mughal Garden	2023	320000.00	Miniature Painting	2023-03-20	Sold
3	3	3	Badshahi Mo...	2024	125000.00	Oil on Canvas	2024-01-10	Sold
4	4	4	Portrait of a S...	2023	215000.00	Oil on Canvas	2023-05-12	Sold
5	5	5	Urban Dreams	2024	165000.00	Mixed Media	2024-02-08	Sold
6	6	6	Phulkari Herit...	2023	275000.00	Textile Art	2023-06-25	Sold
7	7	7	Abstract Lahore	2024	340000.00	Acrylic on Canvas	2024-03-15	Sold
8	8	8	Kashmir Valley	2023	145000.00	Watercolor	2023-07-18	Sold

	OrderID	CustomerID	EmployeeID	OrderDate	PaymentMethod	PaymentStatus	TotalAmount	ContactEmail
1	1	1	2	2024-01-20	Bank Transfer	Completed	198000.00	muhammad.asif@email.com
2	2	2	5	2024-02-15	Bank Transfer	Completed	352000.00	hira.saleem@email.com
3	3	3	2	2024-03-10	Credit Card	Completed	238500.00	imran.haider@email.com
4	4	4	8	2024-04-05	Cash	Completed	181500.00	sana.tariq@email.com
5	5	5	5	2024-05-12	Bank Transfer	Pending	302500.00	kamran.bashir@email.com
6	6	6	2	2024-06-18	Bank Transfer	Completed	374000.00	nadia.rehman@email.com
7	7	7	8	2024-07-22	Credit Card	Completed	137500.00	faisal.mahmood@email.com
8	8	8	5	2024-08-15	Bank Transfer	Completed	159500.00	rabia.nawaz@email.com



	OrderLineID	OrderID	ArtWorkID	SalePrice	Commission	LineTotal			
1	1	1	1	180000.00	18000.00	198000.00			
2	2	2	2	320000.00	32000.00	352000.00			
3	3	3	4	215000.00	21500.00	236500.00			
4	4	4	5	165000.00	16500.00	181500.00			
5	5	5	6	275000.00	27500.00	302500.00			
6	6	6	7	340000.00	34000.00	374000.00			
7	7	7	3	125000.00	12500.00	137500.00			
8	8	8	8	145000.00	14500.00	159500.00			

	TicketID	ExhibitionID	CustomerID	VisitDate	PaymentStatus	TicketType	TicketPrice	PurchaseDate
1	1	1	1	2024-02-10	Paid	General	2000.00	2024-01-20
2	2	2	2	2024-02-20	Paid	VIP	1500.00	2024-02-01
3	3	3	3	2024-04-15	Paid	General	2500.00	2024-03-15
4	4	4	4	2024-05-10	Paid	Student	2000.00	2024-04-20
5	5	5	5	2024-06-20	Paid	General	2200.00	2024-05-25
6	6	6	6	2024-07-15	Paid	VIP	1800.00	2024-06-30
7	7	7	7	2024-08-05	Paid	General	2100.00	2024-07-15
8	8	8	8	2024-09-10	Paid	Student	2300.00	2024-08-20

	ArtistID	ArtStyle
1	1	Calligraphy
2	1	Islamic Art
3	2	Miniature
4	3	Landscape
5	4	Portrait
6	5	Mixed M...
7	6	Textile Art
8	7	Abstract

	AddressID	ArtistID	Street	City	State	ZipCode	Country
1	1	1	House 45, Street 7, F-10/3	Islamabad	ICT	44000	Pakistan
2	2	2	Plot 123, DHA Phase 5	Lahore	Punjab	54000	Pakistan
3	3	3	Flat 6, Block A, Gulistan-...	Karachi	Sindh	75290	Pakistan
4	4	4	House 78, Cavalry Ground	Lahore	Punjab	54810	Pakistan
5	5	5	Street 12, Sector F-11	Islamabad	ICT	44000	Pakistan
6	6	6	House 34, Gulberg III	Lahore	Punjab	54660	Pakistan
7	7	7	Flat 9, Clifton Block 5	Karachi	Sindh	75600	Pakistan
8	8	8	House 56, Model Town	Lahore	Punjab	54700	Pakistan

	CustomerID	PreferredArtStyles
1	1	Calligraphy, Islamic Art
2	2	Miniature, Traditional
3	3	Landscape, Contem...
4	4	Portrait, Realistic
5	5	Mixed Media, Abstract
6	6	Textile, Traditional
7	7	Abstract, Modern
8	8	Watercolor, Nature

	AddressID	CustomerID	Street	City	State	ZipCode	Country
1	1	1	House 12, Street 5, DHA Phase 4	Rawalpindi	Punjab	46000	Pakistan
2	2	2	Flat 3B, Bahria Town Phase 7	Islamabad	ICT	44000	Pakistan
3	3	3	Plot 67, DHA Phase 8	Lahore	Punjab	54792	Pakistan
4	4	4	House 89, Johar Town	Lahore	Punjab	54782	Pakistan
5	5	5	Street 10, Sector E-11	Islamabad	ICT	44000	Pakistan
6	6	6	Flat 12, Defence View	Karachi	Sindh	75500	Pakistan
7	7	7	House 45, Satellite Town	Rawalpindi	Punjab	46300	Pakistan
8	8	8	Plot 23, Bahria Town	Lahore	Punjab	53720	Pakistan



	ExhibitionID	Hall	Floor	Building			
3	3	Faiz Hall	Ground Floor	West Wing			
4	4	Ghalib Hall	2nd Floor	Main Building			
5	5	Rumi Hall	1st Floor	East Wing			
6	6	Meer Hall	Ground Floor	Main Building			
7	7	Sirshar H...	1st Floor	West Wing			
8	8	Saadat H...	2nd Floor	East Wing			
9	9	Zauq Hall	Ground Floor	North Buildi...			
10	10	Momin H...	1st Floor	South Buildi...			
	ArtworkID	PriceHistory					
3	3	2024-01-10: Rs.125000;					
4	4	2023-05-12: Rs.195000; 2023-11-10: Rs.215000;					
5	5	2024-02-08: Rs.155000; 2024-06-15: Rs.165000;					
6	6	2023-06-25: Rs.250000; 2023-12-20: Rs.275000;					
7	7	2024-03-15: Rs.340000;					
8	8	2023-07-18: Rs.135000; 2023-10-12: Rs.145000;					
9	9	2024-04-20: Rs.400000; 2024-08-15: Rs.425000;					
10	10	2023-08-30: Rs.180000; 2024-02-10: Rs.195000;					
	ArtworkID	Height	Width	Depth			
1	1	91.44	60.96	3.00			
2	2	30.48	40.64	0.50			
3	3	121....	91.44	2.50			
4	4	106....	76.20	2.00			
5	5	91.44	91.44	10.00			
6	6	152....	121...	5.00			
7	7	182....	121....	3.50			
8	8	45.72	60.96	0.20			
	OrderID	Street	City	State	ZipCode	Country	
1	1	House 12, Street 5, DHA Phase 4	Rawalpindi	Punjab	46000	Pakistan	
2	2	Flat 3B, Bahria Town Phase 7	Islamabad	ICT	44000	Pakistan	
3	3	Plot 67, DHA Phase 8	Lahore	Punjab	54792	Pakistan	
4	4	House 89, Johar Town	Lahore	Punjab	54782	Pakistan	
5	5	Street 10, Sector E-11	Islamabad	ICT	44000	Pakistan	
6	6	Flat 12, Defence View	Karachi	Sindh	75500	Pakistan	
7	7	House 45, Satellite Town	Rawalpindi	Punjab	46300	Pakistan	
8	8	Plot 23, Bahria Town	Lahore	Punjab	53720	Pakistan	