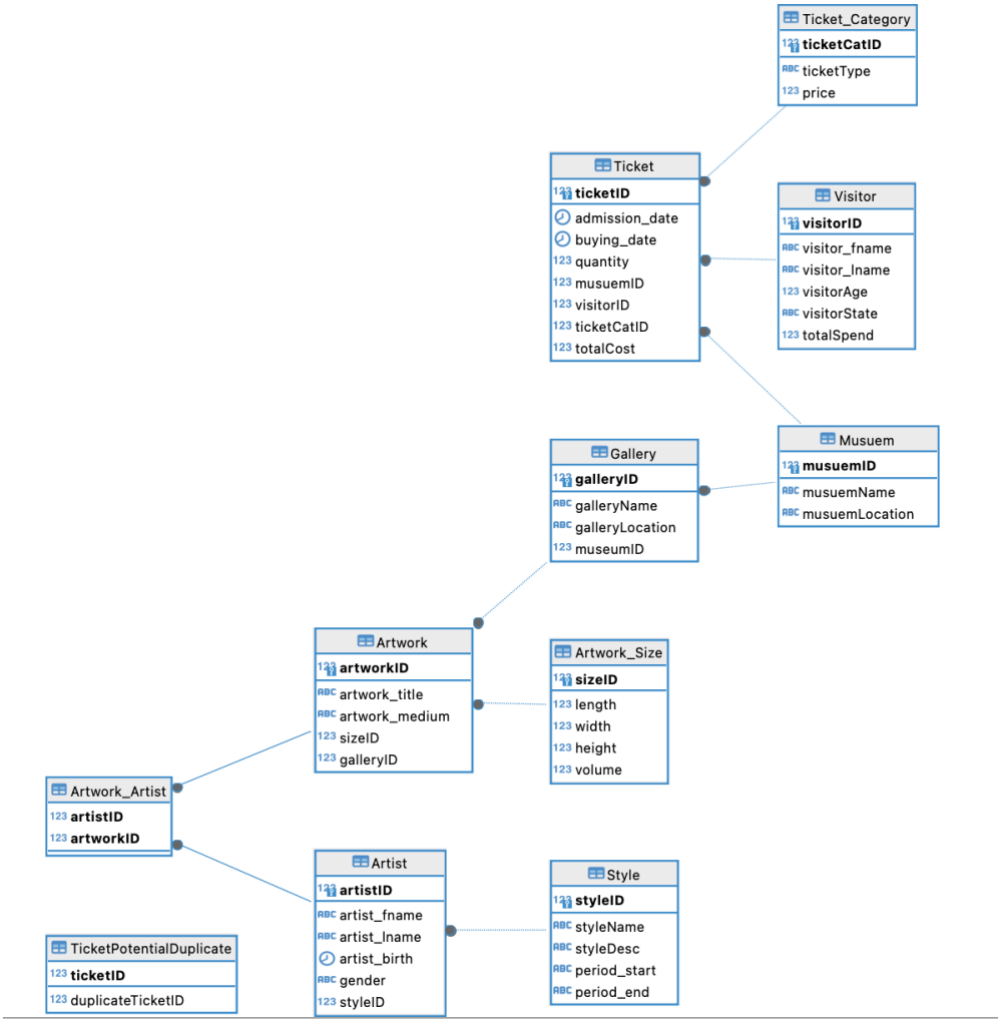


# MUSEUM DATASE



## STORED PROCEDURE 1: Artist Addition Procedure with Duplication and Date of Birth Validation

```
CREATE OR ALTER PROCEDURE Updating_Artist
    @artistID INT,
    @artist_fname VARCHAR(50),
    @artist_lname VARCHAR(50),
    @artist_birth DATE,
    @gender VARCHAR(50),
    @styleID INT
AS
BEGIN
    -- Check if AritstID already exists
    IF EXISTS (SELECT 1 FROM [dbo].[Artist] WHERE [artistID] = @artistID)
    BEGIN;
        THROW 51000, '@artistID already exists in the database.', 1;
    END;

    IF (@artist_birth > GETDATE())
    BEGIN;
        THROW 61283, '@artist_birth cannot be in the future.', 1;
    END;

    -- Insert the new artist
    BEGIN TRY
        BEGIN TRANSACTION G1;
        INSERT INTO dbo.Artist (artist_fname, artist_lname, artist_birth, gender, styleID)
        VALUES (@artist_fname, @artist_lname, @artist_birth, @gender, @styleID);
        COMMIT TRANSACTION G1;
        PRINT 'Artist added successfully.';
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION G1;
        THROW 51583, 'Error inserting data in the database.', 1;
    END CATCH
END;
```

## EXEC Updating\_Artist

```
@artistID = 34,  
@artist_fname = 'Amrita',  
@artist_lname = 'Shergill',  
@artist_birth = '2025-01-30',  
@gender = 'Female',  
@styleID = 6;
```

	artistID	artist_fname	artist_lname	artist_birth	gender	styleID
1	19	Vincent	van Gogh	1853-03-30	Male	1
2	20	Auguste	Rodin	1840-11-12	Male	2
3	21	Pablo	Picasso	1881-10-25	Male	3
4	22	Leonardo	da Vinci	1452-04-15	Male	4
5	23	Judy	Chicago	1939-07-20	Female	5
6	24	Joseph	Stella	1877-06-13	Male	6
7	25	Katsushika	Hokusai	1760-10-31	Male	1
8	26	Charles	Eames	1907-06-17	Male	2
9	27	Festo	Festo	1971-01-01	N/A	3
1...	28	Andy	Warhol	1928-08-06	Male	4
1...	30	Henry Fairfie...	Osborn	1857-08-08	Male	4
1...	31	William	Scheele	1920-02-03	Male	4
1...	32	Peter	Eisenman	1932-08-11	Male	6
1...	33	Amrita	Shergill	1913-01-30	Female	6
1...	NULL	NULL	NULL	NULL	NULL	NULL

## STORED PROCEDURE 2: Artwork Table Update Procedure with Constraint Validation

```
SET IDENTITY_INSERT dbo.Artwork ON;
```

## CREATE PROCEDURE Update\_Artwork

```
@artworkID INT,  
@artwork_title VARCHAR(MAX),  
@artwork_medium VARCHAR(MAX),  
@sizeID INT,  
@galleryID INT
```

AS

BEGIN

```
-- Check if Artwork already in the database
```

```

IF EXISTS (SELECT 1 FROM [dbo].[Artwork] WHERE [artworkID] = @artworkID)
BEGIN;
    THROW 51002, '@artworkID already exists in the database.', 1;
END;

-- Check if Artwork title is not null
IF @artwork_title IS NULL
BEGIN;
    THROW 51003, '@artwork_title cannot be unnamed. Insert a title or say Untitled', 1;
END;

-- Insert the new artwork
BEGIN TRY
    BEGIN TRANSACTION G2;
    INSERT INTO dbo.Artwork (artworkID, artwork_title, artwork_medium, sizeID, galleryID)
    VALUES (@artworkID, @artwork_title, @artwork_medium, @sizeID, @galleryID)
    COMMIT TRANSACTION G2;
    PRINT 'Artwork added successfully.';
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION G2;
    THROW 51584, 'Error inserting data in the database.', 1;
END CATCH
END;

EXEC Update_Artwork
    @artworkID = 30,
    @artwork_title = 'Three Girls',
    @artwork_medium = 'Oil on Canvas',
    @sizeID = 10,
    @galleryID = 6;

```

	artworkID	artwork_title	artwork_medium	sizeID	galleryID
1	13	Starry Night	Oil on Canvas	6	6
2	14	The Thinker	Bronze Sculpt...	7	6
3	15	Les Demoisell...	Oil on Canvas	7	6
4	16	Mona Lisa	Oil on Panel	9	6
5	17	The Dinner Pa...	Mixed Media I...	8	7
6	18	The Brooklyn ...	Oil on Canvas	9	7
7	19	The Great Wav...	Woodblock Pri...	10	7
8	21	Chair	Molded Plywood	8	8
9	22	Robotic Hand	Metal and Ele...	9	8
1...	23	Campbell's So...	Acrylic and S...	10	9
1...	24	Guernica	Oil on Canvas	8	9
1...	25	Tyrannosaurus...	Fossilized Bo...	10	10
1...	26	The Blue Whal...	Plastic and M...	9	10
1...	27	Children's Me...	Mixed Media I...	6	11
1...	30	Three Girls	Oil on Canvas	10	6
1...	NULL	NULL	NULL	NULL	NULL

## TRIGGER 1: Duplicate Ticket Purchase Monitoring

-- Potential Duplicate Ticket Purchase Trigger - If the same user purchases the same quantity and ticket category for the same day, this is a potential duplicate transaction that the user did mistakenly. We will add this to a Ticket\_Potential\_Duplicate table to be audited by museum operators.

-- Creating TicketPotentialDuplicate table

```
CREATE TABLE [dbo].[TicketPotentialDuplicate] (
    [ticketID] INT NOT NULL,
    [duplicateTicketID] INT NOT NULL,
    CONSTRAINT [PK_TicketPotentialDuplicate] PRIMARY KEY CLUSTERED ([ticketID] ASC)
);
```

```
CREATE OR ALTER TRIGGER dbo.potentialDuplicateTransaction ON dbo.Ticket FOR INSERT
```

AS

```
-- Declaring variables to store inserted row information
declare @visitorID int, @quantity int, @ticketCatID int, @ticketID int, @admission_date date, @duplicateTicketID
int, @musuemID int

-- Extracting inserted row information
SELECT @visitorID = i.visitorID, @quantity = i.quantity, @ticketCatID = i.ticketCatID, @ticketID = i.ticketID,
@admission_date = i.admission_date, @musuemID = i.musuemID FROM Inserted i

-- Querying Ticket table to see if duplicate transactions exist for the inserted row
SELECT @duplicateTicketID = ticketID FROM dbo.Ticket WHERE visitorID = @visitorID AND quantity =
@quantity AND ticketCatID = @ticketCatID AND @admission_date = admission_date AND @musuemID =
musuemID AND NOT ticketID = @ticketID

-- If duplicate transaction exists, adding the information to a new TicketPotentialDuplicate table for museum
operators to audit
IF @duplicateTicketID IS NOT NULL
BEGIN
    INSERT INTO [Museum].[dbo].[TicketPotentialDuplicate] (ticketID, duplicateTicketId) VALUES (@ticketID,
@duplicateTicketID);
END
GO

INSERT INTO [Museum].[dbo].[Ticket] (admission_date, buying_date, quantity, musuemID, visitorID, ticketCatID)
VALUES
('2024-02-17', '2024-02-09', 3, 1, 1001, 200);

INSERT INTO [Museum].[dbo].[Ticket] (admission_date, buying_date, quantity, musuemID, visitorID, ticketCatID)
VALUES
('2024-03-22', '2024-02-12', 8, 1, 1005, 202);
```

	ticketID	duplicateTick...
1	5262	5260
2	5263	5261
3	NULL	NULL

## TRIGGER 2: Gallery Data Consistency Enforcement

-- TRIGGER 2: Gallery Data Consistency Trigger: If a gallery is attempted to be deleted that has artwork attached, the delete should fail.

```
CREATE OR ALTER TRIGGER dbo.preventGalleryDeleteWithArtwork ON dbo.Gallery INSTEAD OF DELETE
AS
```

-- Declaring variables to store inserted row information

```
declare @galleryID int
```

-- Extracting inserted row information

```
SELECT @galleryID = d.galleryID FROM DELETED d
```

-- Querying ARTWORK table and throwing error if there is an artwork associated with it

```
IF EXISTS (SELECT * FROM Artwork WHERE galleryID = @galleryID)
```

```
BEGIN;
```

```
    THROW 67283, 'Cannot delete Gallery that still has Artwork assets attached. Please reassign Artwork to other galleries before attempting delete.', 1;
```

```
END;
```

```
DELETE FROM Gallery WHERE galleryID = @galleryID;
```

```
GO
```

-- Checking to see if the trigger works when we have an artwork associated with a gallery

```
DELETE FROM Gallery WHERE galleryID = 6;
```

Results	Messages	Query Plan	Plan Tree	Top Operations
	(1 row affected) (1 row affected) (1 row affected) (1 row affected) Msg 67283, Level 16, State 1, Procedure preventGalleryDeleteWithArtwork, Line 16 Cannot delete Gallery that still has Artwork assets attached. Please reassign Artwork to other galleries before attempting delete. Total execution time: 00:00:00.120			

## COMPUTATIONAL COLUMN 1: To Calculate Total Cost

-- Computed Column 1: For Total Cost

```
CREATE FUNCTION dbo.getTicketPrice(@ticketCatID int)
```

RETURNS FLOAT

AS

BEGIN

declare @price float

select @price = price from dbo.Ticket\_Category where @ticketCatID = ticketCatID

RETURN @price

END

ALTER TABLE dbo.Ticket ADD totalCost AS (ROUND(dbo.getTicketPrice(ticketCatID)\*quantity,2));

admission_date	buying_date	quantity	musuemID	visitorID	ticketCatID	totalCost
2024-02-14	2024-02-09	2	1	1001	200	79.98
2024-02-16	2024-02-11	1	2	1002	201	59.99
2024-02-19	2024-02-14	3	3	1003	202	269.97
2024-02-14	2024-02-09	2	1	1001	200	79.98
2024-02-16	2024-02-11	1	2	1002	201	59.99
2024-02-19	2024-02-14	3	3	1003	202	269.97
2024-02-21	2024-02-15	1	4	1004	207	19.99
2024-02-23	2024-02-18	2	5	1005	208	19.98
2024-02-26	2024-02-21	1	1	1006	208	9.99
2024-02-29	2024-02-24	4	2	1007	208	39.96
2024-03-21	2024-02-24	4	1	1006	207	79.96
2024-03-22	2024-01-14	8	1	1005	202	719.92
2024-03-19	2024-01-24	1	1	1004	207	19.99

## COMPUTATIONAL COLUMN 2: For Total Spending by One Visitor

-- Comuted Column 2: For Total Spent by one visitor

CREATE FUNCTION dbo.getTotalSpendForVisitor(@visitorID INT)

RETURNS FLOAT

AS

BEGIN

DECLARE @visitorSpend FLOAT

SELECT @visitorSpend = SUM(totalCost) FROM dbo.Ticket T WHERE T.visitorID = @visitorID

RETURN @visitorSpend

END

ALTER TABLE dbo.Visitor ADD totalSpend AS ROUND(dbo.getTotalSpendForVisitor(visitorID),2);



	visitorID	visitor_fname	visitor_lname	visitorAge	visitorState	totalSpend
1	1001	John	Smith	30	California	2279.43
2	1002	Emily	Johnson	25	New York	739.87
3	1003	Michael	Williams	40	Texas	729.9
4	1004	Sarah	Brown	22	Florida	99.95
5	1005	David	Martinez	35	Arizona	3699.5
6	1006	Jessica	Garcia	28	Illinois	269.93
7	1007	James	Jones	45	Ohio	59.95
8	NULL	NULL	NULL	NULL	NULL	NULL

### COMPLEX QUERY 1:

-- COMPLEX QUERY 1 (JOINS, GROUP BY, AGGREGATE FUNCTIONS, ORDER BY)

-- Names of Museums where visitors over the age of 25 spent more than 40\$ on average on Museum tickets.

SELECT M.museumName, M.museumID, AVG(TC.price) AS AvgAmountSpend

FROM Museum M

JOIN Ticket T ON M.museumID = T.museumID

JOIN Visitor V ON T.visitorID = V.visitorID

JOIN Ticket\_Category TC ON T.ticketCatID = TC.ticketCatID

WHERE V.visitorAge > 25 AND TC.price > 40

GROUP BY M.museumID, M.museumName

ORDER BY AvgAmountSpend DESC;

	museumName	museumID	AvgAmountSpend
1	MoMA	1	89.99
2	Cooper Hewitt, Smithsonian Design Museum	3	89.99
3	American Museum of Natural History	4	59.99

### COMPLEX QUERY 2:

-- COMPLEX QUERY 2 (JOINS, GROUP BY, FILTERING)

-- Artists born after 1800 who have their work showcased at 2nd floor of any gallery

SELECT A.artist\_fname, A.artist\_lname

FROM Artist A

JOIN Artwork\_Artist AA ON A.artistID = AA.artistID

JOIN Artwork AW ON AA.artworkID = AW.artworkID

JOIN Gallery G ON AW.galleryID = G.galleryID

JOIN Musuem M ON G.museumID = M.musuemID

JOIN Ticket T ON M.musuemID = T.musuemID

WHERE A.artist\_birth > '1800-01-01' AND G.galleryLocation = '2nd Floor'

GROUP BY A.artist\_fname, A.artist\_lname

	artist_fname	artist_lname
1	Results grid	Rodin
2	Pablo	Picasso
3	Vincent	van Gogh