Art Museum Database Overview

Group 3: Sarah Morris, Daquan Morrison, Sugeun Chae, Kevon Darton, John West Jr.

IST 659 Fall 2023

Introduction/High-Level Overview

The Artwork Database is a database built for a newly opened imaginary museum in New York. The museum contains 50 of the most historically significant paintings in the world, each of which were recently donated by our generous museum donors. Prior to the creation of this database, our museum did not have a way to track all information about each artwork, its artist, the value of the piece, and the donor of the piece of art. Therefore, we elected to create a database with 4 tables titled Artwork, Artists, Art Value, and Donors. Each table can be joined to each other using foreign keys, and each table is in 3NF[1] with no duplicate rows, partial or transitive dependencies. Overall, the database will function as an inventory management system, a customer relationship management system, and a store of all information regarding artists/artworks for any future exhibits.

Objectives

The business objectives for the creation of the database are multifold. Firstly, the creation of the database is inventory management, so we can know which paintings are being displayed at a given time. The Database is also a CRM tool. By having the contact information and up to date information on our donors, we are able to reach out to them to notify them of museum events, and to send them thank you cards for their generous contributions to our museum. Without a compiled donors table, we do not know who has donated what or how to reach them. Another core business objective of the museum is to construct exhibits for our visitors. By having all information (date, nationality, themes, medium, etc) of our art and artists, we can create exhibits

[1] When reviewing our project, we found that we had missed separating our "name" columns into two different columns (first name and last name) for our donors and artworks table. Other than this oversight (caught in Week 10 after the normalization unit), the tables are atomic and do not contain partial or transitive dependencies.

from paintings that share similar themes or historical periods, which we were unable to do before. Finally, it is critical that we can update our database or change information as needed in case we have new donors or want to add new information to our tables.

Technical Glossary for Business Partners

Term	**Definition**
Check Constraint	A range or limitation set on data to ensure data integrity
Foreign Key (FK)	A column of values in a table that serve as a primary key in another table, or a "key" to allow joining of tables
Referential Integrity	Refers to consistency in data across tables, data is not contradictory
Unique Constraint (U)	Ensures that all data in the column is unique and there are no duplicates.
3NF	The third normal form (format for our tables), which states that all columns are wholly and solely dependent on the primary key. This prevents future data anomalies or errors with input/deletions.
Primary Key (PK)	A column that is a distinctive, unique record for each row to ensure data integrity.
Required (R)	Specifies that the column cannot have non-null values.
Composite (C)	Specifies that the column is composed of 1 of more components.
View	A virtual table based on a result set.
Function	A stored query that receives inputs to produce an output of information.
Trigger	A function that automatically executes when a specific DML (data manipulation) event occurs in the database.
Procedure	Set of code that is saved and stored to run on command.

Business Rules Pertaining to the Entire Dataset

Referential Integrity: Ensuring foreign key relationships are maintained correctly across all tables.

Unique Constraints: Ensuring that certain combinations of fields (like artist name and title in Artworks) remain unique across the dataset.

Data Completeness: Non-null constraints on essential fields ensure that critical data is not missing from the records.

Business Rules Pertaining to Tables

Artists Table:

Primary Key Constraint: artist_id is a unique identifier for each artist.

Data Integrity: Non-null constraints on essential fields like artist_first_name, artist_last_name, and artist_nationality.

Valid Date Range: Ensuring that artist_date_of_birth is less than artist_date_of_death if both are provided.

Artworks Table:

Primary Key Constraint: artwork_id is a unique identifier for each artwork.

Foreign Key Constraint: artwork_artist_id references artist_id in the Artists table, ensuring each artwork is associated with a valid artist.

Unique Title per Artist: Each artist cannot have multiple artworks with the same title.

Data Integrity and Validity: Non-null constraints on essential fields like artwork_title, artwork_date, artwork_artist_id, artwork_period, artwork_genre, and artwork_medium.

Valid Artwork Date: The artwork creation date (artwork_date) must be realistic, not in the future and preferably not before a certain historical year (e.g., 1000).

Art Value Table:

Primary Key Constraint: art_value_id as a unique identifier for each artwork's valuation entry.

Data Type Integrity: Using the MONEY data type for art_estimated_value to ensure proper formatting and handling of monetary values.

Valid Appraisal Status: Ensuring value_art_status is within a predefined set of acceptable values (e.g., 'Priceless, 'High-End', 'Moderate').

Donor Table:

Primary Key Constraint: donor_id is a unique surrogate key for each donor.

Data Type Integrity: Using MONEY data type for donor_donated_amount ensures proper formatting of monetary values. Donor_donated_amount is the sum of the values of all artworks in the artworks table with the donor_id.

Data Analysis

Data Entities, Attributes, Relationships:

Relationships					
Relationship	**Entity**	**Entity to**	**Rule**	**Min**	**Max**
Draw	Artist	Artwork	Draw	1	M
	Artwork	Artist	Drawed by	1	1
Estimate	Artwork	Art Value	Estimate	1	1
	Art Value	Artwork	Estimate	1	1
Donate	Donor	Art Value	Donate	0	M
	Art Value	Donor	Donated by	1	1

Table: artworks		
Column Name	**Data Domain**	**Comments**
artwork_id	PK, RU int	Will be the primary key
artwork_artist_id	FK (artists table), RU int	The artist who drew the artwork
artwork_exhibit_id	FK (exhibit table), RU int	The exhibit that the artwork is exhibited
artwork_title	RC varchar(100)	The title of the artwork
artwork_date	R date	The year of the artwork was drawn
artwork_period	R varchar(50)	The period of the artwork
artwork_genre	R varchar(50)	The genre of the artwork
artwork_medium	R varchar(50)	The medium of the artwork
Constraint Name	**Type**	**Comments**
pk_artwork_id	Primary key on artwork_id	Enforces PK over surrogate key on table
fk_artwork_artist_id	Foreign key references artists table	The artist who drawed the artwork
ck_artwork_before_insert	Check constraint or trigger	Make sure the date of the artwork is not in the future

Table: artists		
Column Name	**Data Domain**	**Comments**
artist_id	PK int	Will be the primary key

artist_first_name	UC varchar(50)	First name of the artist
artist_last_name	UC varchar(50)	Last name of the artist
artist_date_of_birth	R date	The date of birth of the artist
artist_date of death	R date	The date of death of the artist
artist_nationality	R varchar(50)	The nationality of the artist
artist_number_paintings _lifetime	R int	The number of paintings of the artist for lifetime
Constraint Name	**Type**	**Comments**
pk_artist_id	Primary key on artist_id	Enforces PK over surrogate key on table
u_artist_name	Unique on artist first name and last name	Enforces natural key to establish entity integrity

Table: donors

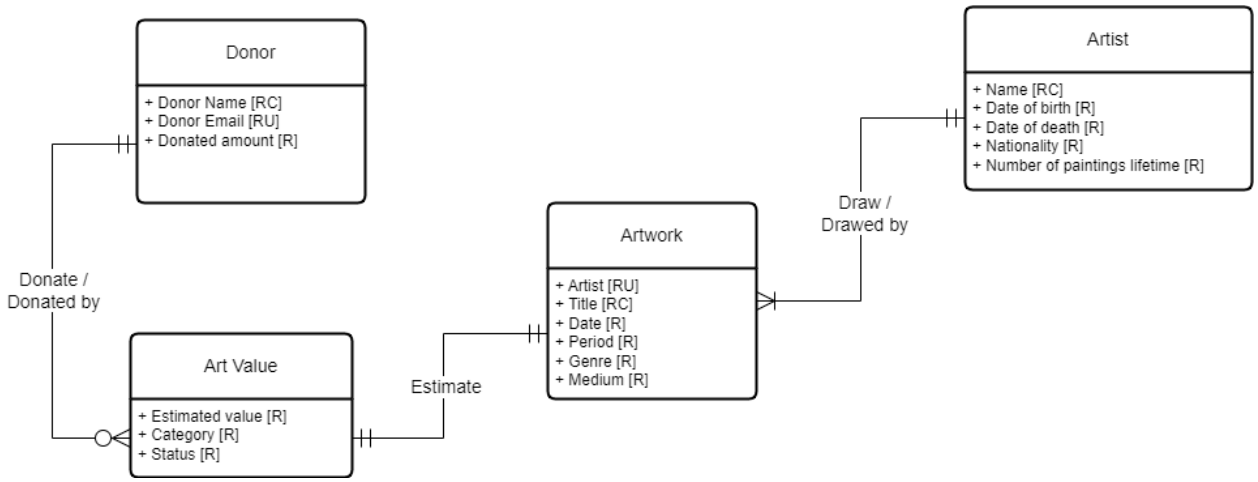
Column Name	Data Domain	Comments
donor_id	PK int	Will be the primary key
donor_email	RU varchar(50)	The email of the donor, Natural key
donor_first_name	R varchar(50)	First name of the donor
donor_last_name	R varchar(50)	Last name of the donor
donor_donated_amount	R money	The amount of donation by the donor
Constraint Name	**Type**	**Comments**
pk_donor_id	Primary key on donor_id	Enforces PK over surrogate key on table
u_donor_email	Unique on donor_email	Enforces natural key to establish entity integrity

Table: artvalue

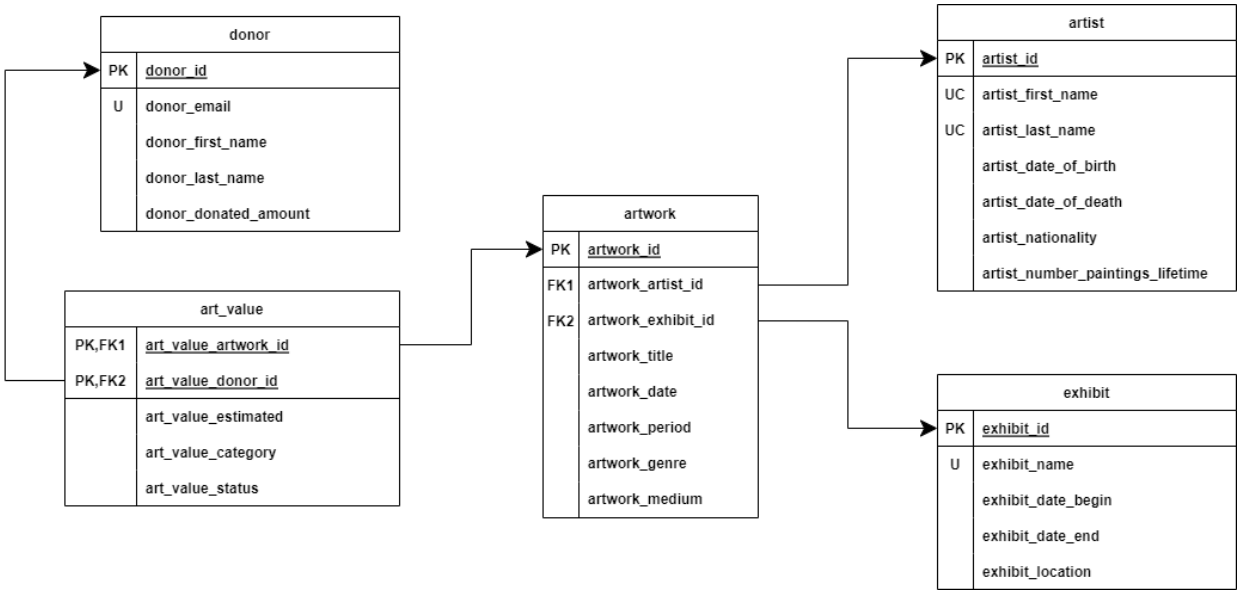
Column Name	Data Domain	Comments
art_value_artwork_id	PK, FK int	Will be the primary key, Foreign key(artwork_id)
art_value_donor_id	PK, FK int	Will be the primary key, Foreign key(donor_id)
art_value_art_estimated	R varchar(50)	The estimated value of the artwork

value_art_category	R check constraint, varchar (15)	The category of the artwork
value_art_status	R varchar(50)	The status of the artwork, Available or not
Constraint Name	**Type**	**Comments**
fk_value_artwork_id	Foreign key references artwork table	The artwork which is valued
fk_value_donor_id	Foreign key references donor table	The donor who donated the artwork

Conceptual Model Diagram



Logical Model Diagram



External Data Model

In all we created four tables populated with data: artists, artwork, artvalue, and donors. There is also a fifth table that we included in our logical data model (exhibits) that is intended to be populated with information as the museum creates more exhibits. Our tables are stored in a database called “art” (under the master database in Azure Data Studio).

Art Database Creation

```
-- Create a new database called 'DatabaseName'
-- Connect to the 'master' database to run this snippet
--USE master
--GO
-- Create the new database if it does not exist already
IF NOT EXISTS (
    SELECT [name]
        FROM sys.databases
        WHERE [name] = 'art'
)
drop database if exists art
go
CREATE DATABASE art
GO
-- creating the donors table under art database
use art
GO
```

Artworks

	artwork…	artwork_arti…	artwork_ar…	artwork_t…	artwork_d…	artwork_per…	artwork_ge…	artwork_medi…
1	3	3	Diego Rivera	Paisaje zapat…	1915	Cubism	Pastoral	Oil on canv
2	4	4	Claude Monet	The Water Lil…	1899	Impressionism	Landscape	Oil on canv
3	5	5	Rene Magritte	The Lovers	1928	Surrealism	Symbolic	Oil on canv
4	6	6	Salvador Dali	Dream Caused …	1944	Surrealism	Symbolic	Oil on canv
5	7	7	Edouard Manet	Le Déjeuner s…	1863	Realism	Counterculture	Oil on canv
6	8	8	Andrei Rublev	Trinity	1425	Medieval	Russian Orthod…	Tempera on

```
CREATE TABLE Artworks (
    artwork_id INT PRIMARY KEY,
    artwork_artist_id INT,
    artwork_artist VARCHAR(255),
    artwork_title VARCHAR(255),
    artwork_date INT,
    artwork_period VARCHAR(100),
    artwork_genre VARCHAR(100),
    artwork_medium VARCHAR(100),
    FOREIGN KEY (artwork_artist_id) REFERENCES Artists(artist_id)
);


INSERT INTO Artworks (artwork_id, artwork_artist_id, artwork_artist, artwork_title, artwork_date, artwork_period, artwork_genre, artwork_medium) V
(3, 3, 'Diego Rivera', 'Paisaje zapatista', 1915, 'Cubism', 'Pastoral', 'Oil on canvas'),
(4, 4, 'Claude Monet', 'The Water Lily Pond', 1899, 'Impressionism', 'Landscape', 'Oil on canvas'),
```

Artists

	artis…	artist_first…	artist_last…	artist_date_of…	artist_date_of_…	artist_natio…
1	1	Amedeo	Modigliani	1884	1920	Italian
2	2	Vasiliy	Kandinskiy	1866	1944	Russian
3	3	Diego	Rivera	1886	1957	Mexican
4	4	Claude	Monet	1840	1926	French
5	5	Rene	Magritte	1898	1967	Belgian
6	6	Salvador	Dali	1904	1989	Spanish
7	7	Edouard	Manet	1832	1883	French
8	8	Andrei	Rublev	1360	1430	Russian

```
60  drop table if exists artists
61  go
62  create table artists(
63      artist_id int primary key,
64      artist_first_name varchar(255),
65      artist_last_name varchar(255),
66      artist_date_of_birth INT,
67      artist_date_of_death INT,
68      artist_nationality varchar(100)
69
70  )
71
72
73  INSERT INTO Artists (artist_id, artist_first_name, artist_last_name, artist_date_of_birth, artist_date_of_death, artist_nationality) VALUES
74
75  (1, 'Amedeo', 'Modigliani', 1884, 1920, 'Italian'),
76
77  (2, 'Vasiliy', 'Kandinskiy', 1866, 1944, 'Russian'),
78
79  (3, 'Diego', 'Rivera', 1886, 1957, 'Mexican'),
80
81  (4, 'Claude', 'Monet', 1840, 1926, 'French'),
```

Donors

	Donor ID Numb…	Donor Name	Donor Ema…	Donation Contribut…	Rewards Sta…
1	1	Perla Riley	priley@hotmai…	21100000.00	Platinum Status
2	2	Alicia Marquez	amqz@gmail.com	48000000.00	Platinum Status
3	3	Rowan Mccarthy	mcrowan@hotma…	278000000.00	Diamond Status
4	4	Allison Solom…	alisonsolo@ho…	522000000.00	Diamond Status
5	5	Lamont Morris…	lamontmorriso…	1498500000.00	Diamond Status
6	6	Willow Valent…	willowval@gma…	428000000.00	Diamond Status
7	7	London Charles	londoncharles…	10000000.00	Gold Status
8	8	Tanya Wall	tanyawall@yah…	858000.00	Gold Status
9	9	Melina Clay	claymelina@ho…	12000000.00	Platinum Status

```sql
-- creating the donors table under art database
use art
GO
drop table if exists donors
GO
create TABLE donors (
    donor_id int not null,
    donor_first_name varchar(50) not null,
    donor_last_name varchar(50) not null,
    donor_email varchar(50) not null,
    donor_donated_amount money not NULL,
    CONSTRAINT pk_donors_donor_id primary KEY (donor_id)
)
-- inserting data
insert into donors
    (donor_id, donor_first_name, donor_last_name, donor_email, donor_donated_amount)
    values
    (1, 'Perla', 'Riley', 'priley@hotmail.com', 21100000),
    (2, 'Alicia', 'Marquez', 'amqz@gmail.com', 48000000),
    (3, 'Rowan', 'Mccarthy', 'mcrowan@hotmail.com', 278000000),
    (4, 'Allison', 'Solomon', 'alisonsolo@hotmail.com', 522000000),
    (5, 'Lamont', 'Morrison', 'lamontmorrison@gmail.com', 1498500000),
    (6, 'Willow', 'Valentine', 'willowval@gmail.com', 428000000),
    (7, 'London', 'Charles', 'londoncharles@hotmail.com', 10000000),
```

Art Value

	art_value_id	art_estimated_value	art_value_category	art_value_status	donor_id
1	1	100000000.00	*NULL*	Available to Auction	11
2	2	40000000.00	*NULL*	Unavailable to Auction	14
3	3	14000000.00	*NULL*	Unavailable to Auction	19
4	4	70400000.00	*NULL*	Unavailable to Auction	20
5	5	59000000.00	*NULL*	Unavailable to Auction	3
6	6	10000000.00	*NULL*	Unavailable to Auction	7

```sql
drop table if exists artvalue
go
CREATE TABLE artvalue (
    art_value_id INT PRIMARY KEY,
    art_estimated_value MONEY,
    art_value_category VARCHAR(50),
    art_value_status VARCHAR(50),
    donor_id INT
);

INSERT INTO artvalue (art_value_id, art_estimated_value, art_art_value_status, donor_id) VALUES
( 1, 100000000, 'Available to Auction', 11),
(2, 40000000, 'Unavailable to Auction', 14),
(3, 14000000, 'Unavailable to Auction', 19),
(4, 70400000, 'Unavailable to Auction', 20),
(5, 59000000, 'Unavailable to Auction', 3),
(6, 10000000, 'Unavailable to Auction', 7),
```

Data Logic

► Function for sorting data by nationality of artists

Purpose:

This function is designed to get data based on nationality of artists

```
--- Making the view shows the information artist and artwork by artist's nationality ---
GO
drop view if exists v_artist_nationality

GO
create view v_artist_nationality AS
   select a.artist_first_name + " " + a.artist_last_name as artist_name,
         a.artist_date_of_birth, a.artist_date_of_death, a.artist_nationality,
         w.artwork_title, w.artwork_date, v.value_art_estimated
      from artists as a
         join artworks as w on a.artist_id = w.artwork_artist_id
         join artvalue as v on w.artwork_id = v.value_art_id
      order by a.artist_nationality

--- Making the function to search the artwork and artist information by artist's nationality ---
GO
drop function if exists f_search_nationality

GO
create function f_search_nationality (
   @keyword as varchar(50)
   ) RETURNS table as
   return
      select * from v_artist_nationality where artist_nationality = @keyword
go

--- Checking the function ---
select * from dbo.f_search_nationality('Italian')
```

► Trigger for Updating Art Value Category

Purpose:

This trigger is designed to ensure the proper category of art value when the art value is updated.

```
-- Category update trigger function --
drop trigger if exists t_category_update
GO
```

```
create trigger t_category_update on dbo.artvalue
after update
as begin
   update dbo.artvalue
   set value_art_category = 'Moderate'
   from dbo.artvalue
   where value_art_estimated <= 10000000

   update dbo.artvalue
   set value_art_category = 'High-End'
   from dbo.artvalue
   where value_art_estimated between 10000001 and 49999999

   update dbo.artvalue
   set value_art_category = 'Priceless'
   from dbo.artvalue
   where value_art_estimated >= 50000000
END
GO

update artvalue set value_art_estimated = '9999999' where value_art_id=2
select * from artvalue
```

► **Trigger for Data Integrity on Artwork Insertion**

Purpose:

This trigger is designed to ensure the integrity of data when a new artwork is added to the Artworks table.

```
CREATE TRIGGER trg_ArtworkBeforeInsert
        ON Artworks
        INSTEAD OF INSERT
        AS
        BEGIN
                IF EXISTS (SELECT 1 FROM inserted WHERE artwork_date >
        YEAR(GETDATE()))
                BEGIN
                RAISERROR ('Artwork date cannot be in the future', 16, 1);
                RETURN;
                END
```

```
        IF NOT EXISTS (SELECT 1 FROM Artists WHERE artist_id IN (SELECT
artwork_artist_id FROM inserted))
        BEGIN
        RAISE ERROR ('Artist does not exist', 16, 1);
        RETURN;
        END
        INSERT INTO Artworks (artwork_id, artwork_artist_id, artwork_artist,
artwork_title, artwork_date, artwork_period, artwork_genre, artwork_medium)
        SELECT artwork_id, artwork_artist_id, artwork_artist, artwork_title,
artwork_date, artwork_period, artwork_genre, artwork_medium FROM inserted;
END;
```

► Procedure to Retrieve Artworks by Century

Purpose:

This stored procedure allows users to retrieve artworks based on the century in which they were created.

```
CREATE PROCEDURE GetArtworksByCentury
    @century INT
AS
BEGIN
    DECLARE @startYear INT, @endYear INT
    SET @startYear = (@century - 1) * 100
    SET @endYear = @century * 100 - 1
    SELECT artwork_id, artwork_artist, artwork_title, artwork_date, artwork_period,
artwork_genre, artwork_medium
    FROM Artworks
    WHERE artwork_date BETWEEN @startYear AND @endYear;
END;
```

► Foreign Key Constraint to Link Artworks with Artists

Purpose:

Establishes a referential integrity constraint between Artworks and Artists. It ensures that each artwork in the Artworks table is associated with a valid artist from the Artists table.

```
ALTER TABLE Artworks
ADD CONSTRAINT FK_Artworks_Artists
FOREIGN KEY (artwork_artist_id) REFERENCES Artists(artist_id);
```

► **Modifying Columns to Enforce Non-Null Values**

Purpose:

This alteration ensures that critical data fields in the Artworks table, such as the title, date, and artist ID, cannot be null.

```
ALTER TABLE Artworks
MODIFY artwork_title VARCHAR(255) NOT NULL,
MODIFY artwork_date INT NOT NULL,
MODIFY artwork_artist_id INT NOT NULL;
```

► **Artist Table**

Purpose:

This procedure pulls detailed information about an artist based on their artist_id.

```
CREATE PROCEDURE GetArtistDetails
    @artist_id INT
AS
BEGIN
    SELECT artist_id, artist_first_name, artist_last_name, artist_date_of_birth,
artist_date_of_death, artist_nationality
    FROM Artists
    WHERE artist_id = @artist_id;
END;
```

```
EXEC GetArtistDetails @artist_id = 1;
```

This procedure allows updates to an artist's details (ex. name, birth and death dates, and nationality)

```
CREATE PROCEDURE UpdateArtistInfo
    @artist_id INT,
    @first_name VARCHAR(255),
    @last_name VARCHAR(255),
    @dob INT,
    @dod INT,
    @nationality VARCHAR(100)
AS
BEGIN
    UPDATE Artists
    SET artist_first_name = @first_name,
    artist_last_name = @last_name,
    artist_date_of_birth = @dob,
    artist_date_of_death = @dod,
    artist_nationality = @nationality
    WHERE artist_id = @artist_id;
END;

EXEC UpdateArtistInfo
    @artist_id = 1,
    @first_name = 'NewFirstName',
    @last_name = 'NewLastName',
    @dob = 1880,
    @dod = 1920,
```

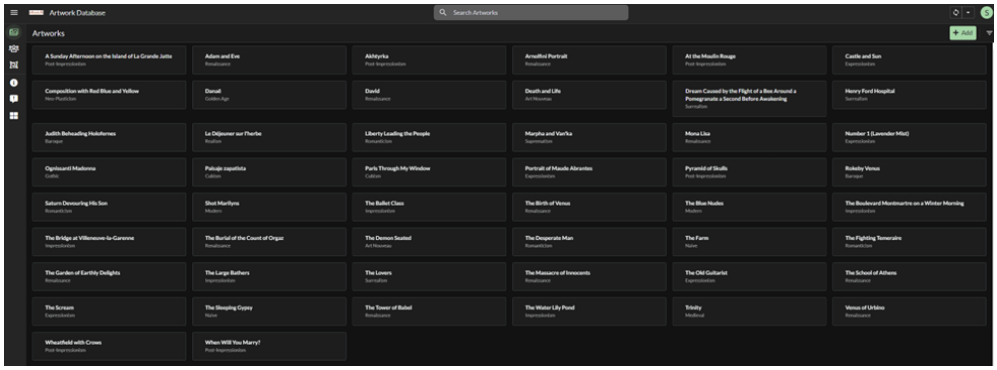
```
    @nationality = 'NewNationality';
```

This procedure retrieves all artists of a specified nationality

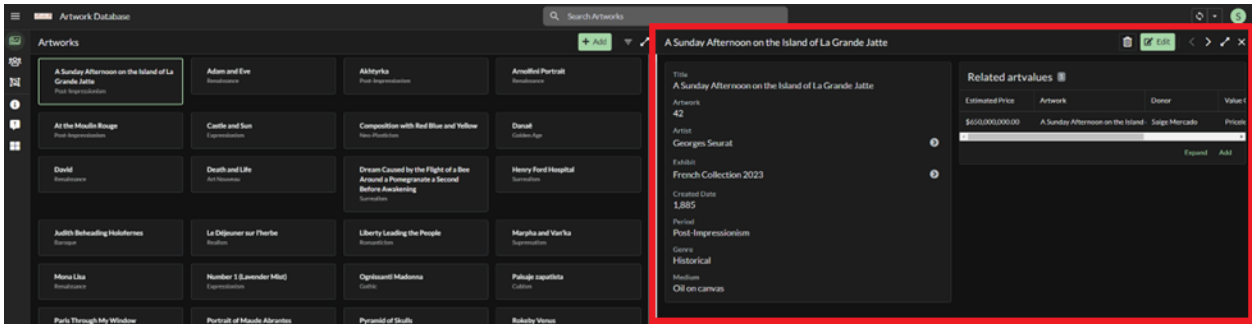
```
CREATE PROCEDURE GetArtistsByNationality
    @nationality VARCHAR(100)
AS
BEGIN
    SELECT artist_id, artist_first_name, artist_last_name, artist_date_of_birth,
artist_date_of_death
    FROM Artists
    WHERE artist_nationality = @nationality;
END;

EXEC GetArtistsByNationality @nationality = 'Italian'
```

Application Demonstration by Google Appsheet

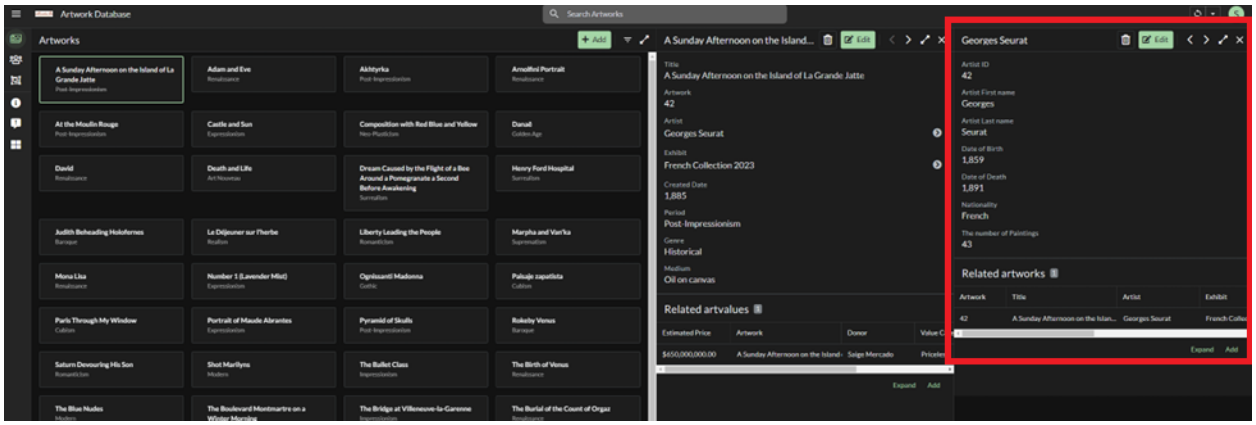
1. The list of Artworks



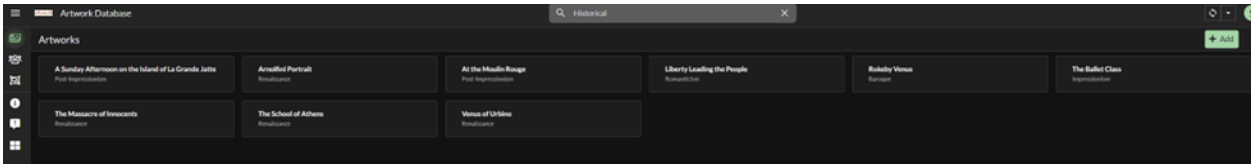
1-1. Artwork Details with Artwork Value



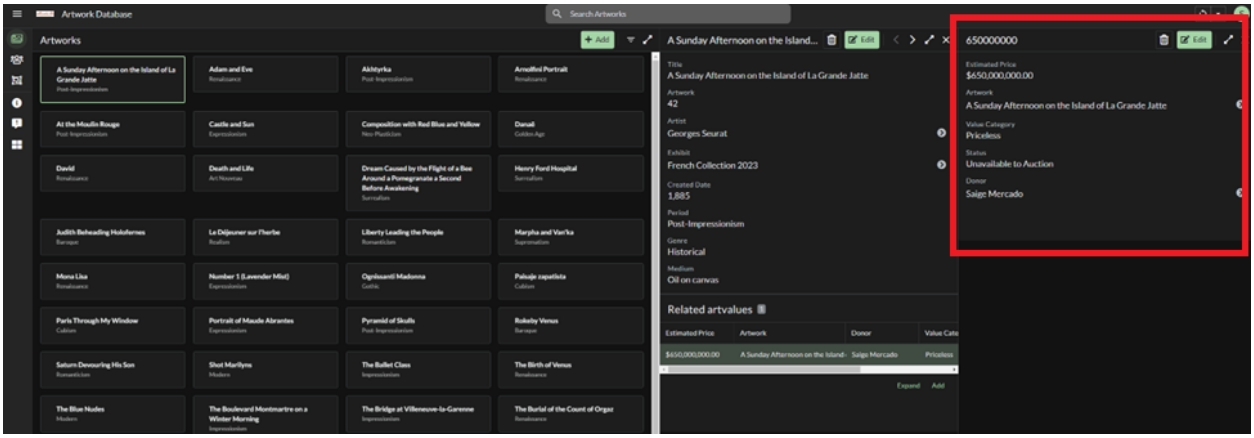
1-2. Artist Details of the artwork



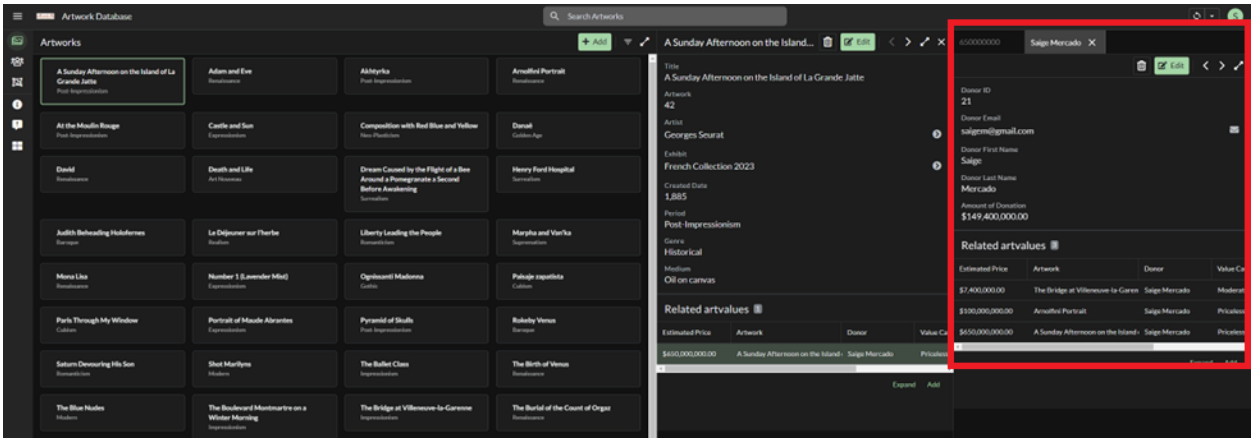
1-3. Search the Artworks by Genre (ex: Historical)



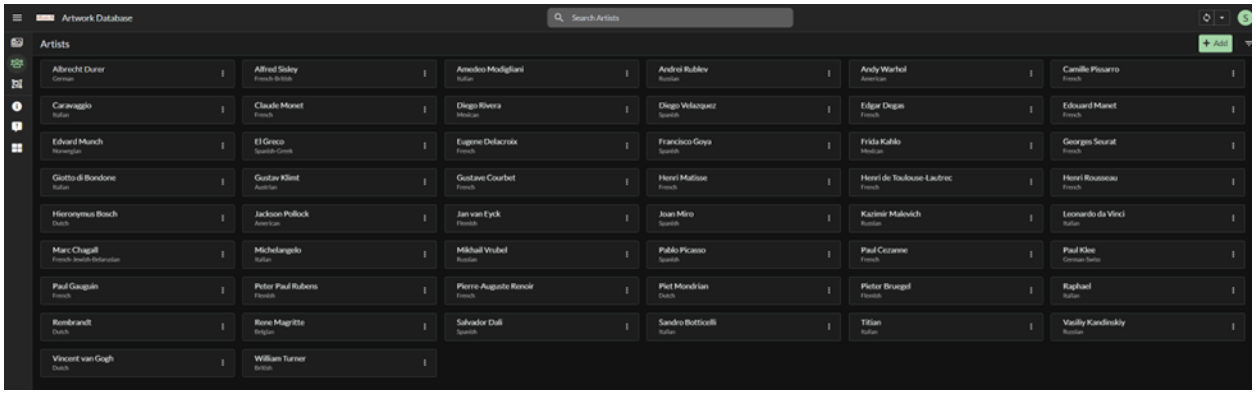
1-4. The Details of Artwork Value



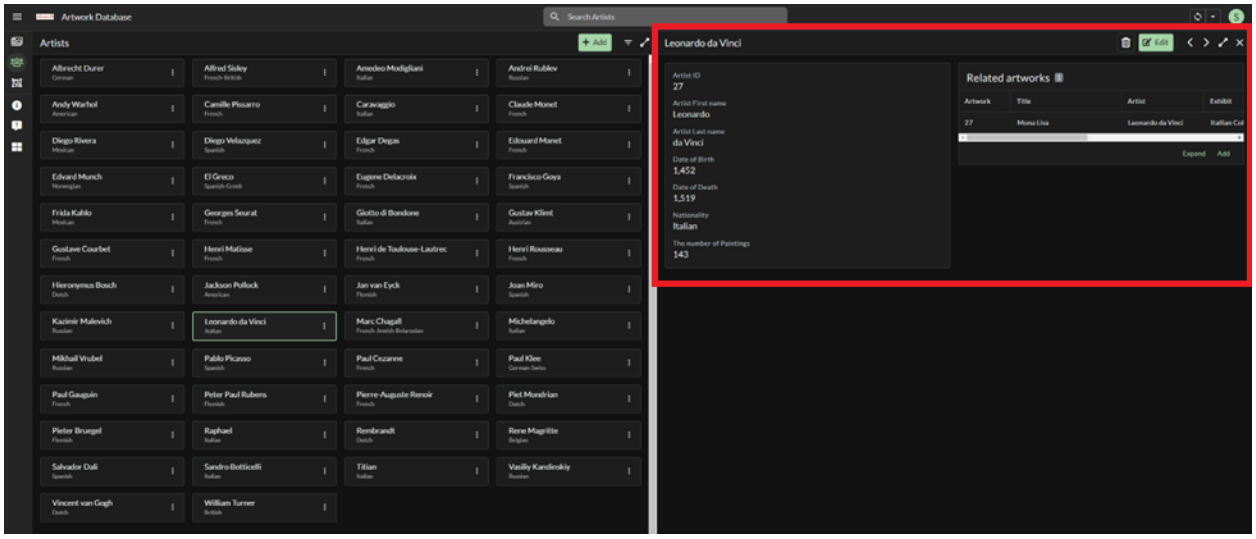
1-5. The Donor Details



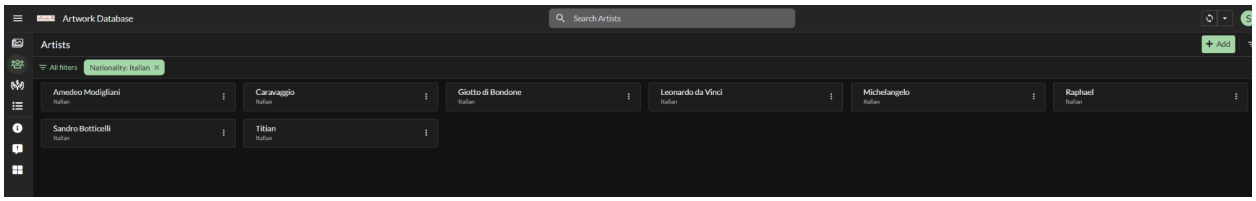
2. The List of Artists



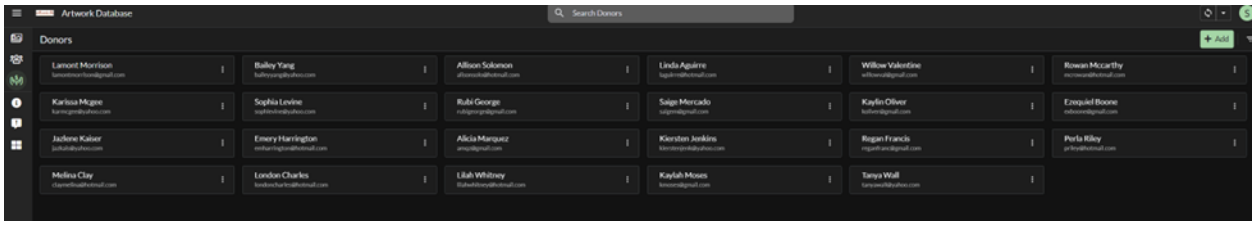
2-1. The Artist Details with the Artwork drew by the Artist



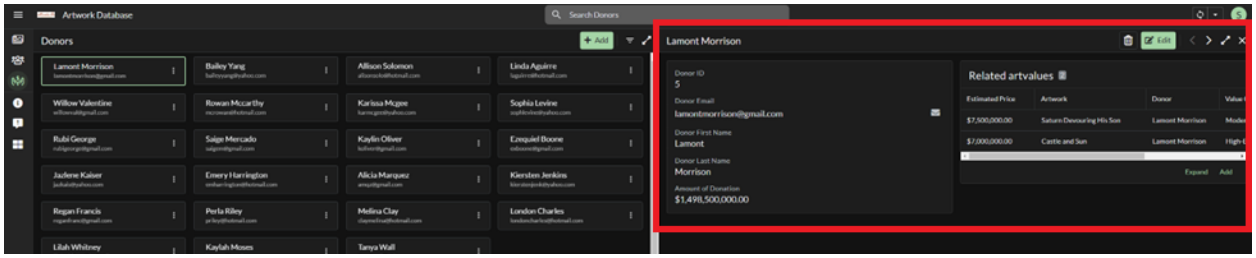
2-2. Filtered by Nationality



3. The List of Donors



3-1. The Details of the Donor



Conclusion

We believe that our database will help improve business operations for our museum across the board, including in customer relationship management, inventory management, auctioning of artwork, and event management. With a functioning donors table that keeps all records of who has donated what, we are able to reach out to our donors to notify them of events, provide updates on their donated paintings, and create a donor rewards program to thank high value donors for their contributions. With functioning art value and artworks table, we can keep track of which paintings we would like to auction in the future and their estimated value. By joining our artworks and artists table, we have information including themes, genres, and country of origin for all of our paintings. This is incredibly important to curate cohesive exhibits and provide educational information for our guests.