Online Art Gallery

Developers: Alexander Ashmore, Samuel Friedman

A Database Design Group Project – Fall 2022

Table of Contents

Overview

The Online Art Gallery was designed by Alexander Ashmore and Samuel Friedman. The purpose of this terminal-based website is to allow customers to view art pieces stored in real life art galleries online and for artists to be able to be able to track their art held by these galleries.

Customers can see different artworks and their artists, so that they can plan to see the art piece in person for viewing or for the sake of purchasing in person. Customers also have the option of following their favorite artists so that they can easily see their work.

Artists benefit from the website by being able to track all their art that may be in different galleries. They get information on its location or if the art piece has been bought by some customer. Artists also are given the opportunity to share their information with customers so that customers can more easily search for art that was made by artist with certain criteria.

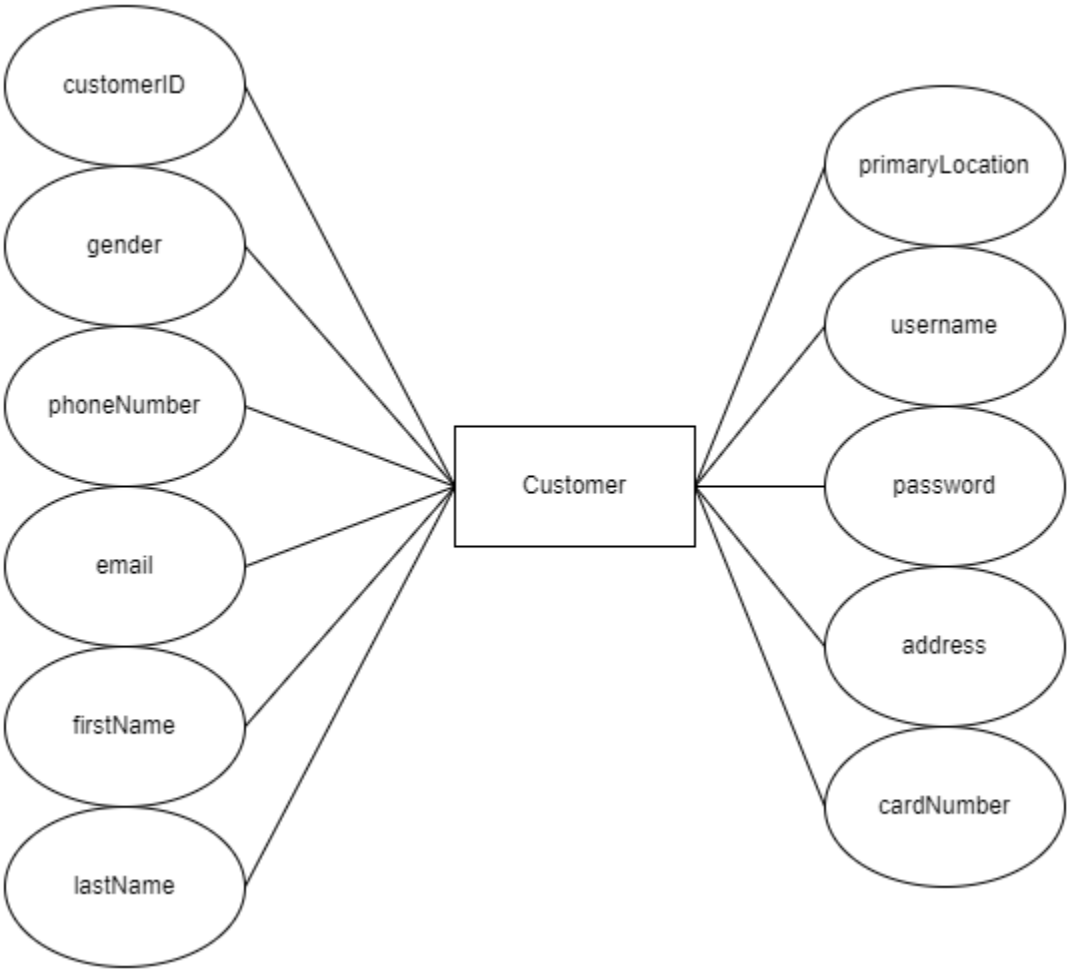
Database Design

The following diagrams provide a high-level introduction to our database. They are provided as a reference while learning about the client's functionality in the next section.

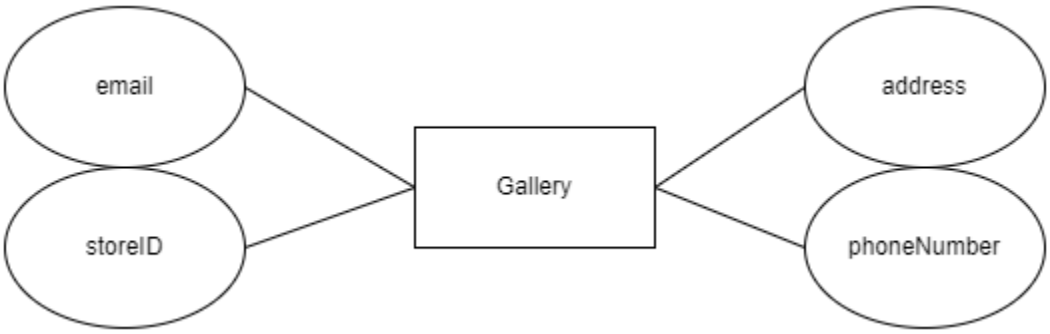
Entity Relationship Diagrams

Entities

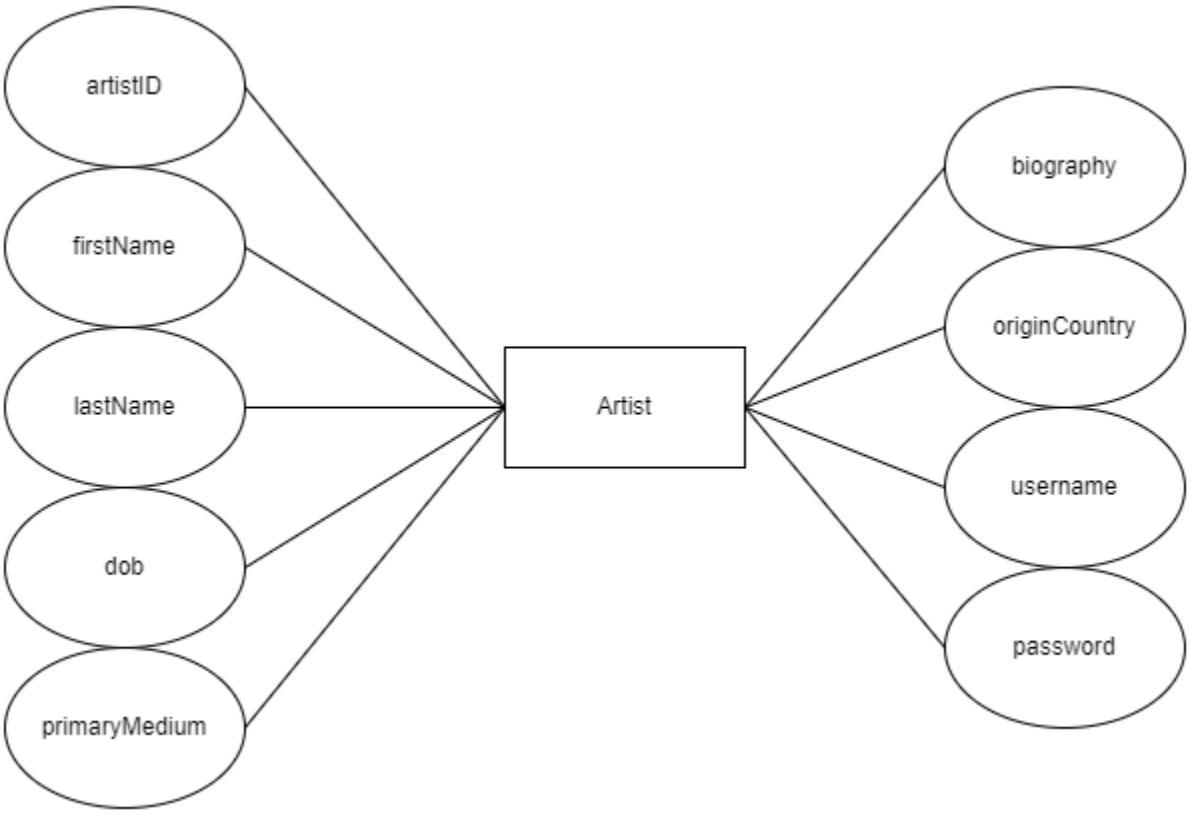
Customers



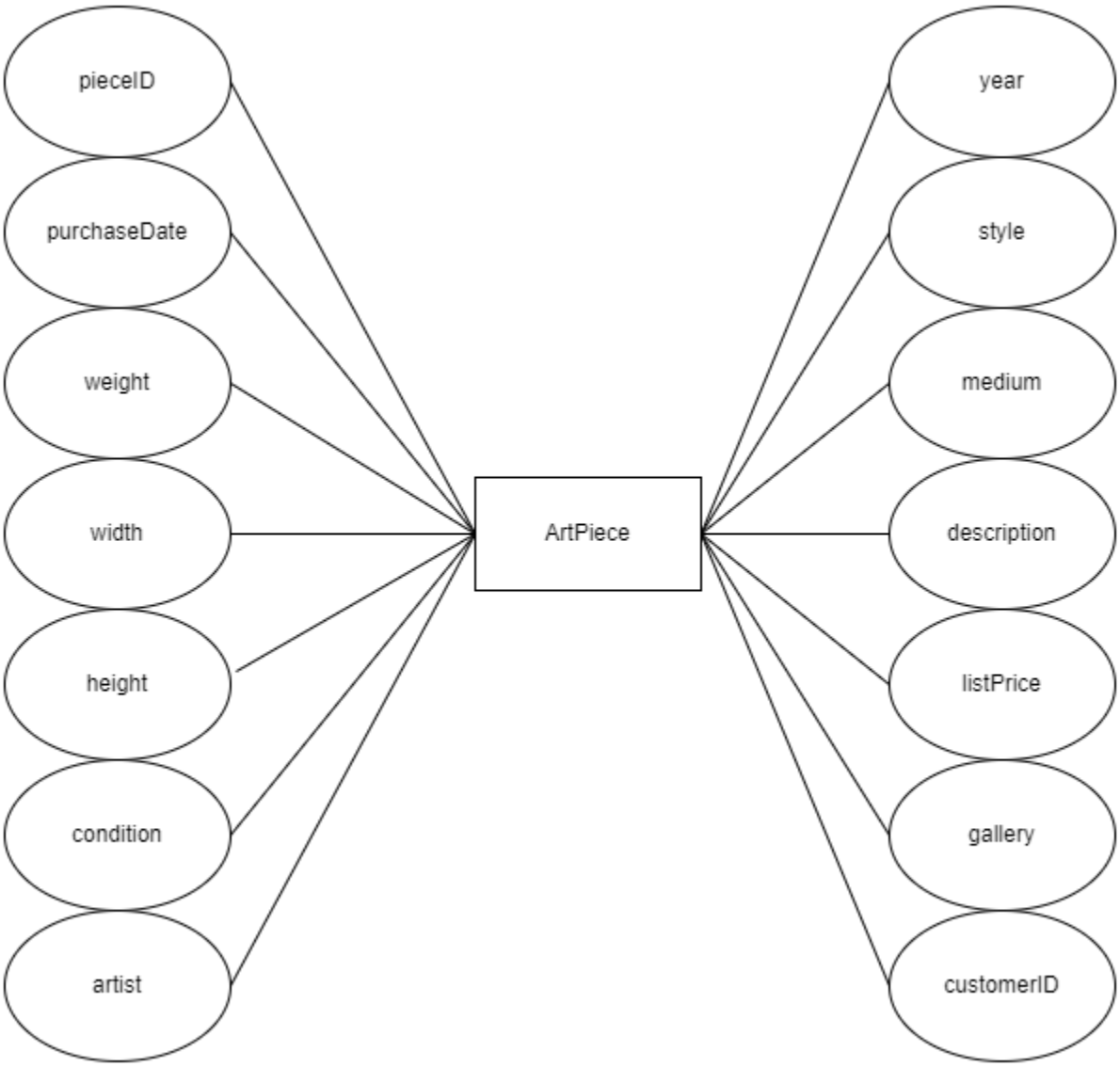
Gallery



Artist



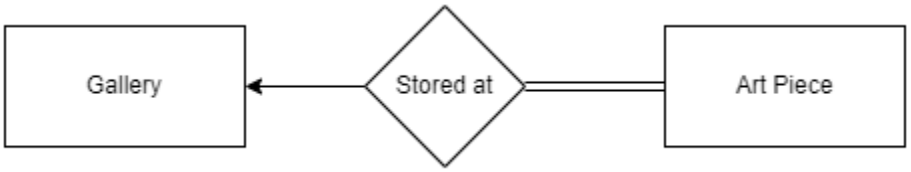
Art Piece


pieceID
purchaseDate
weight
width
height
condition
artist
ArtPiece
year
style
medium
description
listPrice
gallery
customerID

Relations

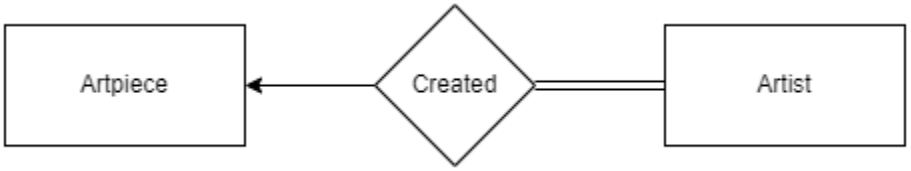
Purchased

Stored at



Favorite artist

Created



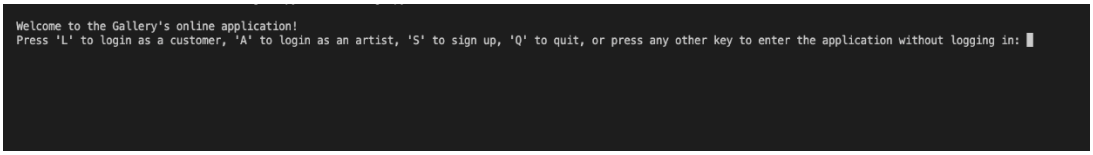
Has primary location

Client Functionality

This section discusses the several types of pages that can be reached by the three types of users: customers, artists, and unregistered users. Customers have the option of seeing their profile, updating the related data, and viewing their favorite store. Artists can also view their profile and update the relevant data, but they can also view their created artworks. Everyone can search for new stores and search for artists and art pieces, but customers can also save their favorite artists and view their works.

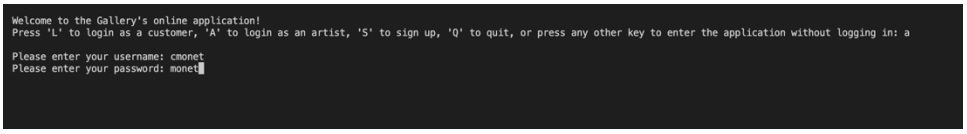
The pages are shown and described below:

1. Welcome Screen:



   This page welcomes the user to login as a customer, artists, or skip the login process all together.

2. Login Page:

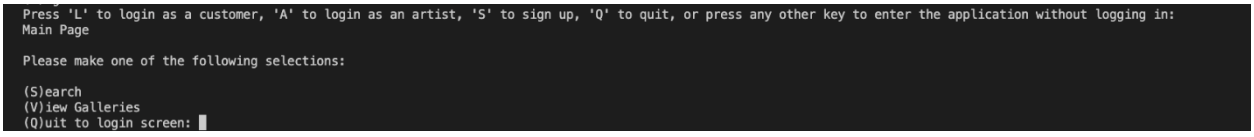


   This page is shown when a user attempts to log in as a customer or artist.

3. Main Menu:

   This page is shown to logged in artists and customers. It invites them to search for artists and art pieces, to view galleries, and to view their profile

4. Main Menu (No Login):

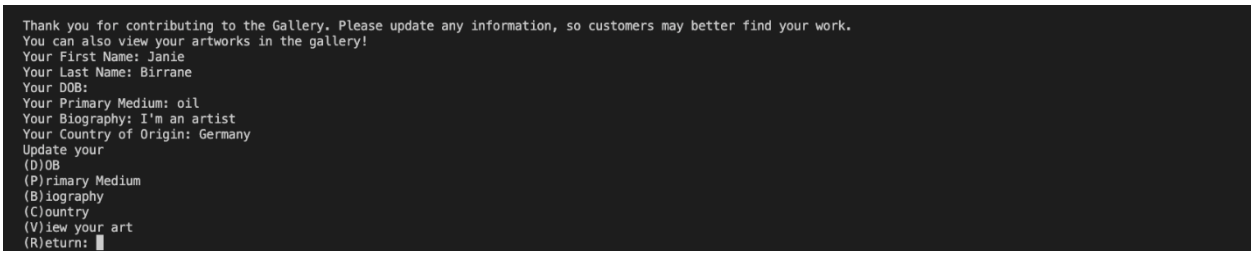


   The Page displayed here gives the generic user the option to see galleries and perform the same search for artists and art pieces.

5. Profile Page (Customer):

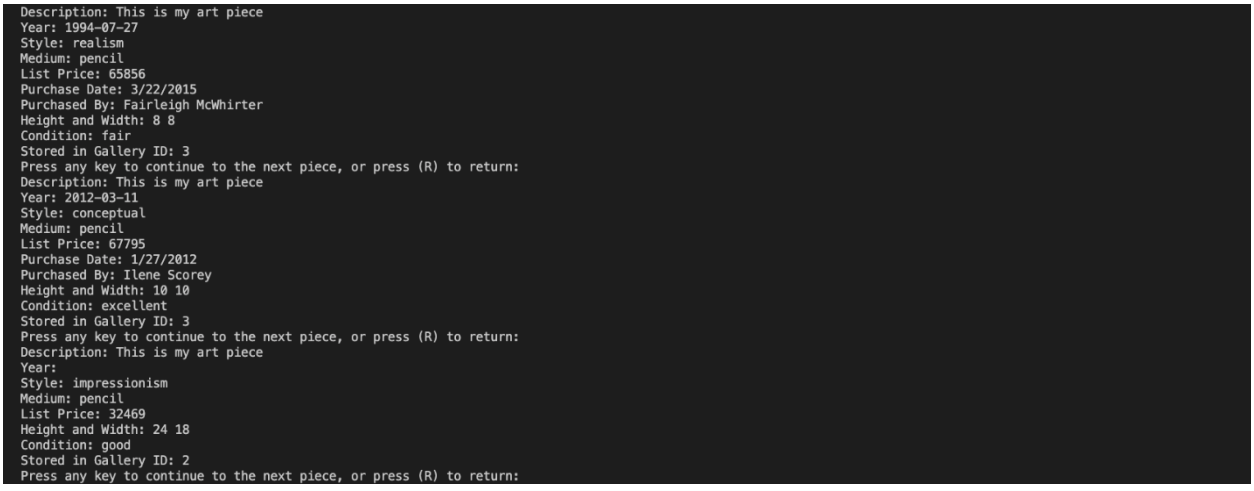
The customer can see their related information, and update key information for usage of the system.

6. Profile Page (Artist):



Here, the artist can see their outward facing profile information and update it as necessary. They can also see the artworks they have held in the gallery.

7. View My Art (Artist):



This page allows the artist to view their art held in the gallery, one by one. It gives the relevant information on the art piece, as well as data on if it sold.

8. Search for Gallery:

```
View (F)avorite gallery, (S)earch for a gallery, or press any key to return: s

Please enter part of the address to search for: lutz

There are 1 gallery/galleries matching your search:
Email: mrowthorn1@apache.org
Address: 82 Buhler Plaza Lutz FL
Phone Number: 308-672-0833

(F)avorite store, or press any key to see the next gallery:
```

All users can search for a gallery by using part of the address (road name, city, state, etc.). Customers can select F to favorite the store they searched for.

9. View Favorite Gallery (Customer):

```
View (F)avorite gallery, (S)earch for a gallery, or press any key to return: f
Email: wgriffe0@odnoklassniki.ru
Address: 41922 Fisk Place, Tampa FL
Phone Number: 8204798474
```

This page allows the customer to view the data of the gallery that they have stored as their favorite.

10. Search Page:

```
Search:
(A)rtist
(P) for Art Piece
(R) to return:
```

This page is the landing page for all searches. Users can search for an artist, or an artwork.

11. Search Artist Page:

```
Search Artist by:
(A)ll artists
(F)irst name
(L)ast name
(M)edium
(R) to return:
```

Here, there is the landing page of the artist search. Users can view all artists, search for artists matching a first or last name, or search an artist by primary medium.

12. Search All Artists Page:

```
Please type one of the options to order all artists by:
firstName
lastName
primaryMedium: firstName


Name: Aloin Leal
Primary Medium: stone
Biography: I'm an artist
Nationality: Spain
(S)ave this artist, or press (C) to stop the search, or any other key to continue:


Name: Alvira Koeppe
Primary Medium: oil
Biography: I'm an artist
Nationality: Spain
(S)ave this artist, or press (C) to stop the search, or any other key to continue:
```

The artist search is demonstrated here. The user has the option to choose the ORDER BY option. If the user is a customer (as shown above), they are given the option to save an artist as a favorite artist. Please note that the other search options for artists are not shown, as they function similarly, with only an added WHERE condition.

13. Art Piece Search Landing Page:

This landing page allows for the viewing of all artworks or viewing artworks that match specific criteria. The Favorite Artist tab only appears when the user is signed in as a customer.

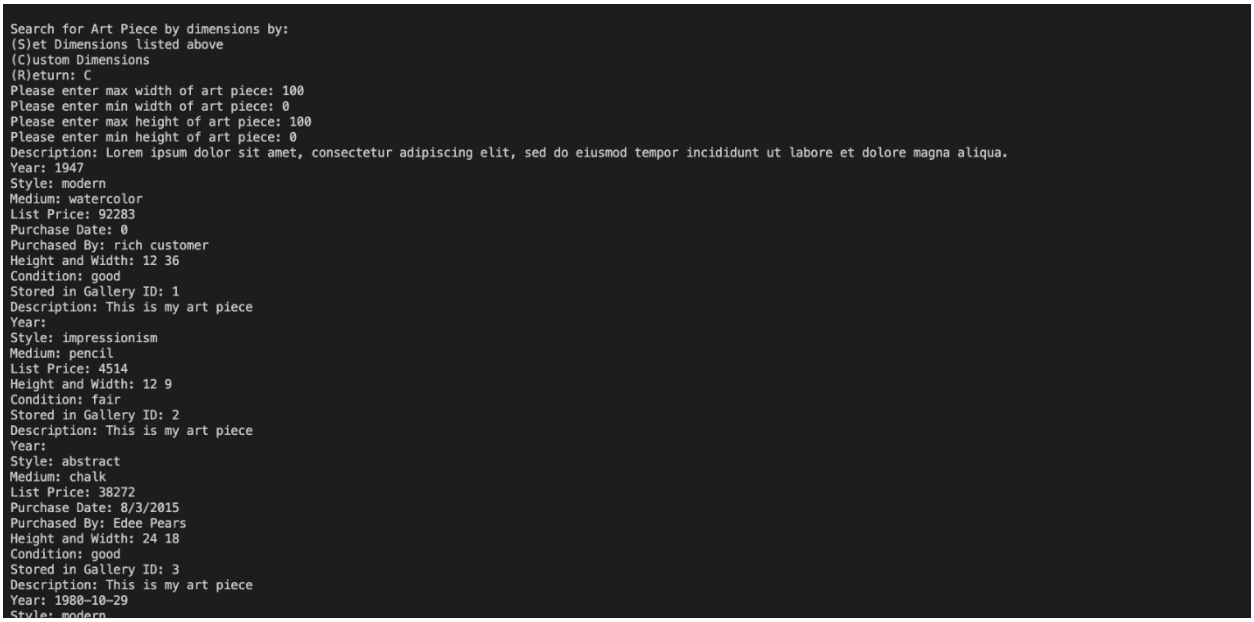
14. All Art Piece Search:

When viewing the art pieces, the user has the option to sort by specific criteria. This allows the user to see every art piece that the gallery holds, and to see key information about it, including the location of the gallery where it is being displayed. Please note that many of the search pages will be omitted, since they display similar information, but with an added ORDER BY filter.

15. Art Piece by Dimension Page:

Any user can find an art piece that matches a specific set of dimensions. The above search looks for a 12 x 36 artwork.

16. Art Piece by Dimension Range Page:



This page works similarly to the one above, but instead of looking for a specific set of dimensions, it looks for a range of dimensions

17. Price Range Search Page:

```
Search Art Piece by:
(E) for all artpieces
(A)rtist
(G)allery
(C)ondition
(S)tyle
(M)edium
(W)eight
(D)imensions
(P)rice
(R) to return: p

Choose an art piece between selected prices
Please enter the max price of the art piece: 10000
Please enter the min price of the art piece: 50
Description: This is my art piece
Year:
Style: impressionism
Medium: pencil
List Price: 4514
Height and Width: 12 9
Condition: fair
Stored in Gallery ID: 2
Description: This is my art piece
Year: 1996-01-18
Style: modern
Medium: oil
List Price: 7181
Height and Width: 48 36
Condition: excellent
Stored in Gallery ID: 5
```

This page is used to search for art in a specific price range.

18. Favorite Artist Page (customer):

```
Search Art Piece by:
(E) for all artpieces
(A)rtist
(G)allery
(C)ondition
(S)tyle
(M)edium
(W)eight
(D)imensions
(P)rice
(F)avorite Artist
(R) to return: f
Press enter to see the art pieces by your favorite artists:
Description: This is my art piece
Year: 1985-10-29
Style: modern
Medium: stone
List Price: 54053
Height and Width:
Condition: fair
Stored in Gallery ID: 5
Description: This is my art piece
Year: 1987-02-14
Style: minimalism
Medium: stone
List Price: 75327
Height and Width:
Condition: fair
Stored in Gallery ID: 3
Description: This is my art piece
Year: 1995-08-27
Style: pop
Medium: clay
List Price: 83716
Height and Width:
Condition: fair
Stored in Gallery ID: 1
Description: This is my art piece
Year:
Style: impressionism
Medium: clay
List Price: 70746
Height and Width:
Condition: excellent
Stored in Gallery ID: 2
```

This final search page is only available to customers. It displays all art pieces that were created by an artist favorited by the customer.

The above pages represent the interactions with the art gallery program. Using SQL queries, including inserts, selects, updates, joins, and more, a comprehensive user interface was created. Using the above pages, users can figure out where to find artworks that are relevant to them, where to visit the art gallery, and where to find key information about artists. Artists can share their outward facing profiles as well.

Database Tables

This section describes what data we are storing in our database and how our relations represent this data.

The following tables are included in the database. Refer to the entity relationship diagrams for a higher-level graphical description.

Customers

Customer(
customerID: INTEGER,
gender: VARCHAR(9),
phoneNumber: CHAR(12),
email: VARCHAR(254),
firstName: VARCHAR(20),
lastName: VARCHAR(20),
cardNumber: CHAR(16),
address: VARCHAR(40),
primaryLocation: INTEGER,
username: VARCHAR(20,
password: VARCHAR(50),
)

Foreign keys: *primaryLocation(Gallery.storeID)*

Primary key: *customerID*

NOT NULL*: customerID, email, firstName, lastName, username, password*

This table stores the information associated with each customer. A customer's *customerID* is used by other tables to track customer relations with artists and art pieces. Art pieces can be bought by users A customer can have an artist set as their favorite artist so that they can easily see information pertaining to them. Username and password are case sensitive in all queries.

Gallery

Gallery(
storeID: INTEGER,
email: VARCHAR(254),
address: VARCHAR(40),
phoneNumber: CHAR(12),
)
Foreign keys: *None*
Primary key: *storeID*
NOT NULL*: storeID, email, address, phoneNumber*

Gallery table is used to store information with different galleries. There can be multiple in person gallery locations. This table provides information pertaining to their contact information like email, address, and phone number. storeID is also used in several queries to check if art pieces are held in those stores, or if the store is a primary location of a customer. Supposedly on a fully implemented website, if the gallery is set as the primary location of a customer, the gallery can send updates of new art pertaining to that specific gallery.

Favorite Artist

FavoriteArtist(
customerID: INTEGER,
artistID: INTEGER,
)
Foreign keys: *customerID(Customer.customerID), artistID(Artist.artistID)*
Primary key: *(customerID, artistID)*
NOT NULL*: customerID, artistID*

This table stores the relation between artists and customers. If a customer favorites an artist, the customers' *customerID* will be paired with the artist's *artistID*. Customers can search their favorite artist in the search page, referencing *artistID* and *customerID*. FavoriteArtist table is used as an intermediate source to show other information of chosen artist like name, primary medium, and so forth. Supposedly on a fully implemented website, if the artist is a favorite artist of that customer the gallery can send emails to the customer if the artist has new artwork.

Artist

Artist(
artistID: INTEGER,
firstName: VARCHAR(20),
lastName: VARCHAR(20),
dob: DATE,
primaryMedium: VARCHAR(50),
biography: VARCHAR(400),
originCountry: VARCHAR(50),
username: VARCHAR(20),
password: VARCHAR(50),
)
Foreign keys: *None*
Primary key: *artistID*
NOT NULL*: artistID, firstName, lastName, username, password*

This table stores the information associated with each artist. *artistID* is used in several other queries like checking ownership of art pieces as well as a relation to customers; artist being the customer's favorite artist. This table is very useful for the favorite artist feature, as users can easily reference new art pieces made by their favorite artist so that they may view it or buy it at that gallery location. Username and password are case sensitive in all queries.

Art Piece

ArtPiece(
pieceID: INTEGER,
purchaseDate: DATE,
weight: DECIMAL(8,2),
width: DECIMAL(8,2),
height: DECIMAL (8,2),
condition: VARCHAR(50),
year: DATE,
style: VARCHAR(50),
medium: VARCHAR(50),
description: VARCHAR(400),
listPrice: DECIMAL(11,2),
gallery: INTEGER,
artist: INTEGER,
customerID: INTEGER,
)
Foreign keys: *gallery(Gallery.storeID), artist(Artist.artistID)*
Primary key: *pieceID*
NOT NULL*: pieceID, listPrice, gallery, artist*

This table stores the information associated with each art piece. It is used to identify if the artwork is currently owned by a customer or if it is still up for sale. This is identified by checking if customerID and purchaseDate is NULL or not. If it has a value, that means that it is owned by a customer. The table also stores which gallery the artwork is held at so that users may search up this information. Customers can either view the art pieces online or in person, while artists can know which gallery locations their art is stored at or if customers have bought their piece.

SQL Queries

Signup

When a user enters the database, they are given the option of continuing without signing up, signing up, or logging in. If the user chooses to sign up, they are given the option of signing up as an artist or a customer.

Signup as an Artist

The user is then prompted to enter both a username and password. The username, '<*username*>', should be of 20 characters or smaller. Password should be a maximum length of 50 characters. The query below serves the purpose of checking if the username is taken or not. If the query returns a value, that means that the username already exists. The user is then prompted to enter a username that does not exist.

```
SELECT * FROM Artist WHERE username = '<username>';
```

The query below serves the purpose of attaching an identification number as a random integer of 13 digits or less. This query is run in a loop assigning a random integer and checking if the query returns a value. If a value is returned, that means the artistID, '<*RandomGenID*>', is already taken and the loop will run again until an unused value is found.

```
SELECT COUNT(*) FROM Artist WHERE artistID = '<RandomGenID>';
```

Once an unused integer for artistID is found, the below query is run to put the artist into the system. Values for artistID, username, password, first name, and last name are gathered before and entered in this query.

```
INSERT INTO Artist (artistID, username, password, firstName, lastName) VALUES (?,?,?,?,?);
```

Signup as a Customer

The user is then prompted to enter both a username and password. The username, '<*username*>', should be of 20 characters or smaller. Password should be a maximum length of 50 characters. The query below serves the purpose of checking if the username is taken or not. If the query returns a value, that means that the username already exists. The user is then prompted to enter a username that does not exist.

```
SELECT * FROM Customer WHERE username = '<username>';
```

The query below serves the purpose of attaching an identification number as a random integer of 13 digits or less. This query is run in a loop assigning a random integer and checking if the query returns a value. If a value is returned, that means the customerID, '<*RandomGenID*>', is already taken and the loop will run again until an unused value is found.

```
SELECT COUNT(*) FROM Artist WHERE artistID = '<RandomGenID>';
```

Once an unused integer for artistID is found, the below query is run to put the customer into the system. Values for customerID, username, password, first name, and last name are gathered before and entered in this query.

```
INSERT INTO Customer (customerID, username, password, firstName, lastName, email) VALUES
(?,?,?,?,?,?);
```

Login

Users have the choice of logging in as an artist or a customer. The process for both is very similar, checking username and password in the database in either the Customer table or Artist table. There is a length check for both password and username to ensure the data is being entered properly. If it is not, the user has 5 login attempts until returning to the main page.

Customer login

The customer enters their username, '<*customerUsername*>', and password, '<*customerPassword*>'. The below query is run. If data is returned, a global variable, loggedIn, is set to TRUE, activeUser is equal to '<*customerUsername*>', and userType is set to "customer". These values keep track of certain permissions and views the customer can have access to.

```
SELECT * FROM Customer WHERE username = '<customerUsername>' AND password =
'<customerPassword>';
```

Artist login

The artist enters their username, '<*artistUsername* >', and password, '<*artistPassword* >'. The below query is run. If data is returned, a global variable, loggedIn, is set to TRUE, activeUser is equal to '<*artistUsername* >', and userType is set to "artist". These values keep track of certain permissions and views the artist can have access to.

```
SELECT * FROM Artist WHERE username = '<artistUsername>' AND password =
'<artistPassword>';
```

Profile

Profiles can only be accessed by the user logged in. If user is not logged in and is browsing as a “guest”, they cannot access this page.

Customer profile

The queries that can be run in customer profile only modify the existing data for that customer. They are not able to modify their customerID.

Updates

The customer can modify their gender, phone number, first name, last name, card number, and address. Each query checks to see if the wanted modified attribute matches the requirements set by the specific table, and if so the UPDATE query will be run.

Updating Gender

The character length for gender should be 9. The attribute gender is overwritten with the new value, ‘<*newGender*>’, where customerID matches the ID of the current customer. The query is below:

> UPDATE Customer SET gender = ‘<*newGender*>’ where customerID = ‘<*currentUserID*>’;

Updating Phone Number

The character length for phoneNumber should be 9 or less. The attribute phoneNumber is overwritten with the new value, ‘<*newphoneNumber*>’, where customerID matches the ID of the current customer. The query is below:

> UPDATE Customer set phoneNumber = ‘<*newphoneNumber*>’ where customerID = ‘<*currentUserID*>’;

Updating First Name

The character length for firstName should be 20 or less. The attribute firstName is overwritten with the new value, ‘<*newFirstName*>’, where customerID matches the ID of the current customer. The query is below:

> UPDATE Customer SET firstName = ‘<*newFirstName*>’ WHERE customerID = ‘<*currentUserID*>’;

Updating Last Name

The character length for last name should be 50 or less. The attribute lastName is overwritten with the new value, ‘<*newlastName*>’, where customerID matches the ID of the current customer. The query is below:

> UPDATE Customer SET lastName = ‘<*newlastName*>’ WHERE customerID = ‘<*currentUserID*>’;

Updating Card Number

The character length for card number should be 16. The attribute cardNumber is overwritten with the new value‘<*newcardNumber*>’, where customerID matches the ID of the current customer. The query is below:

> UPDATE Customer SET cardNumber = ‘<*newcardNumber*>’ WHERE customerID = ‘<*currentUserID*>’;

Updating Address

The character length for address should be 40 or less. The attribute address is overwritten with the new value, '<*newaddress*>', where customerID matches the ID of the current customer. The query is below:

> UPDATE Customer SET address = '<*newaddress*>' WHERE customerID = '<*currentUserID*>';

Artist profile

The queries that can be run in artist profile only modify the existing data for that artist. They are not able to modify their artistID. Artists are also able to view their current artwork and see what galleries they are held at, as well as if the artwork has been bought.

Updates

The artist can modify their day of birth, primary medium, biography, or country of origin. Each query checks to see if the wanted modified attribute matches the requirements set by the specific table, and if so the UPDATE query will be run.

Updating Date of Birth

The format for day of birth should follow the DATE data type in the format of YYY-MM-DD. If not, it will print "Invalid Date". The attribute DOB is overwritten with the new value, '<*newDOB* >', where artistID matches the ID of the current artist. The query is below:

> UPDATE Artist SET DOB = '<*newDOB*>' WHERE artistID = '<*currentArtistID*>';

Updating Primary Medium

The character length for primary medium should be 50 or less. The attribute primaryMedium is overwritten with the new value, '<*newprimaryMedium* >', where artistID matches the ID of the current artist. The query is below:

> UPDATE Artist SET primaryMedium = '<*newprimaryMedium*>' WHERE artistID = '<*currentArtistID*>';

Updating Biography

The character length for biography should be 400 or less. The attribute primaryMedium is overwritten with the new value, '<*biography* >', where artistID matches the ID of the current artist. The query is below:

> UPDATE Artist SET biography = '<*biography*>' WHERE artistID = '<*currentArtistID*>';

Updating Country

The character length for country should be 50 or less. The attribute originCountry is overwritten with the new value, '<*originCountry* >', where artistID matches the ID of the current artist. The query is below:

> UPDATE Artist SET originCountry = '<*originCountry*>' WHERE artistID = '<*currentArtistID*>';

View My Art

The below query gathers all information related to that artist. If nothing was returned, the user is told "There were no artworks found". For each artwork under that artistID, the artwork's description, year, style, medium, list price, heigh and width, condition , gallery is listed. The query is below:

SELECT * FROM ArtPiece WHERE artist = '<*currentArtistID*>';

For each artwork found, it is checked if it has an assigned customerID. If so, the day of purchase and the customer who purchased it is shown as well. The query is below.

SELECT firstName, lastName FROM Customer WHERE customerID = '<*customerWhoBoughtArt*>';

Search

All users, whether they are artists, customers, or guests, can search the database for artists or art pieces. Most of the queries below are very similar, searching by the said criteria and returning almost all the attributes.

Searching by Artist

Users may search for artists by either seeing all artists, searching by first name, last name, or by their primary medium.

Saving Artist by Favorite Artist

This option is only available to the userType "customer". For that customer logged in, they can save the currently shown artist as their favorite artist while seeing search results. They can then view this by searching for their favorite artist, explained in a section below.

The below query gets the customerID of the current user by checking the username, '<*currentUser*>', and searching through the Customer table.

SELECT customerID FROM Customer WHERE username = '<*currentUser*>';

The below query checks if that artist is already saved as that user's favorite artist. If so, it will return "Already saved!" If not, it will continue to the next query.

SELECT COUNT(*) FROM FavoriteArtist WHERE customerID = = '<*currentUserID*>' AND artistID = '<*currentlyViewedArtist*>';

The query below will insert the values of customerID, '<*currentUser*>', and artistID, '<*currentlyViewedArtist*>', into the FavoriteArtist table, storing the relationship between the customer and artist. The customer can now see this artist's work easily by searching for their favorite artists.

INSERT INTO FavoriteArtist (customerID, artistID) VALUES (?,?);

Searching Artist by All Artists

Users can search for all artists in the case that they don't know any artists specifically to look at. This query will return the name, primary medium, biography, and nationality of the artist. If the user is a customer, the user will be prompted to go through and choose "Saving Favorite Artist" section or continue to next artist in the list until there are no longer any artists in the database. The user can choose how the data is organized by firstName, lastName, or primaryMedium shown as '*<dataOrder >*'. The query is below:

SELECT * FROM Artist ORDER BY {'*<dataOrder>*'} ;

Searching Artist by First name

This query will return the name, primary medium, biography, and nationality of the artist. If the user is a customer, the user will be prompted to go through and choose "Saving Favorite Artist" section or continue to next artist in the list until there are no longer any artists in the database. The user can search for artist by their first name, *'<"%"+artistFirstName+"%">'*. The LIKE operator will search for any instance where the user's input for the artist's first name matches and will be returned and ordered by the artist's last name. The query is below:

SELECT * FROM Artist WHERE firstName LIKE *'<"%"+artistFirstName+"%">'* ORDER BY lastName;

Searching Artist by Last name

This query will return the name, primary medium, biography, and nationality of the artist. If the user is a customer, the user will be prompted to go through and choose "Saving Favorite Artist" section or continue to next artist in the list until there are no longer any artists in the database. The user can search for artist by their last name, *'<"%"+artistLastName+"%">'*. The LIKE operator will search for any instance where the user's input for the artist's last name matches and will be returned and ordered by the artist's first name. The query is below:

SELECT * FROM Artist WHERE lastName LIKE *'<"%"+artistLastName +"%">'*ORDER BY firstName;

Searching Artist by Medium

This query will return the name, primary medium, biography, and nationality of the artist. If the user is a customer, the user will be prompted to go through and choose "Saving Favorite Artist" section or continue to next artist in the list until there are no longer any artists in the database. The user can search for artists by their primary medium, *'<"%"+artistLastName+"%">'*. The LIKE operator will search for any instance where the user's input for the artist's primary medium matches and will be returned and ordered by the artist's last name, then first name. The query is below:

SELECT * FROM Artist WHERE primaryMedium LIKE *'<"%"+medium+"%">'*ORDER BY lastName, firstName;

Searching by Art Piece

If purchased

For each search, the below query will be run to check if the art piece is purchased. The query returns the first name, and last name of the customer who owns the piece of art if the attribute customerID,

‘*<customerWhoPurchased>*’, of the current piece of art is NOT NULL. If it is NOT NULL, that means someone bought the art piece.

SELECT firstName, lastName FROM Customer WHERE customerID = ‘*<customerWhoPurchased>*’;

Searching Art Piece by All Artworks

Users can search for all artworks by either purchaseDate, weight, width, height, condition, style, medium, listPrice, or style. This query will return the description, year, style, medium, list price, height and width, condition, and location of art piece. If the art piece is purchased, it will go through the process of “If Purchased” and return purchase date and purchased by, as well as the earlier attributes. The art pieces will show up ordered by the chosen attribute. The query is below:

SELECT * FROM ArtPiece ORDER BY {*‘<chosenSearchBy>’*}

Searching Art Piece by Artist

Users can search art pieces by the artists who made them. Users can enter either the first name or last name of the artist and the query will check all values that contain the input data. This is done by the LIKE operator. If it is remotely similar, that data will be returned. If the art piece is purchased, it will go through the process of “If Purchased” and return purchase date and purchased by, as well as the earlier attributes. The art pieces will show up ordered by pieceID. The query is below:

SELECT * FROM ArtPiece JOIN Artist ON ArtPiece.artist = Artist.artistID WHERE firstName LIKE *‘<"%"+chosenFirstName+"%">’* OR lastName LIKE *‘<"%"+chosenLastName+"%">’* ORDER BY pieceID;

Searching Art Piece by Gallery

Users can search art pieces by the galleries they are held at. Users can enter the address of the gallery and the query will search for anything that contains that input data. This is done by the LIKE operator. If it is remotely similar, that data will be returned. If the art piece is purchased, it will go through the process of “If Purchased” and return purchase date and purchased by, as well as the earlier attributes. The art pieces will show up ordered by pieceID. The query is below:

SELECT * FROM ArtPiece INNER JOIN gallery ON ArtPiece.gallery = Gallery.storeID WHERE address LIKE *‘<"%"+chosenAddress+"%">’* ORDER BY pieceID;

Searching Art Piece by Condition

Users can search art pieces by their condition. Users are presented with the common art conditions: excellent, good, fair. Users can enter the condition of the art piece by either choosing one of the listed above or their own condition they want to check for. The query will search for anything that contains the searched item, *‘<"%"+chosenCondition+"%">’* .This is done by the LIKE operator. If it is remotely similar, that data will be returned. If the art piece is purchased, it will go through the process of “If Purchased” and return purchase date and purchased by, as well as the earlier attributes. The art pieces will show up ordered by pieceID. The query is below:

SELECT * FROM ArtPiece WHERE condition LIKE *‘<"%"+chosenCondition+"%">’* ORDER BY pieceID;

Searching Art Piece by Style

Users can search art pieces by their art style. Users are presented with the common art styles at the gallery: minimalism, pop, impressionism, conceptual, abstract, modern, realism. Users can enter the style of the art piece by either choosing one of the listed above or their own style they want to check for. The query will search for anything that contains the searched item, *'<"%"+chosenStyle+"%">'* .This is done by the LIKE operator. If it is remotely similar, that data will be returned. If the art piece is purchased, it will go through the process of "If Purchased" and return purchase date and purchased by, as well as the earlier attributes. The art pieces will show up ordered by pieceID. The query is below:

SELECT * FROM ArtPiece WHERE style LIKE *'<"%"+chosenStyle+"%">'* ORDER BY pieceID;

Searching Art Piece by Medium

Users can search art pieces by their medium. Users are presented with the common art mediums at the gallery: oil, acrylic, watercolor, charcoal, pastel, chalk, ink, pencil, clay, metal, stone, mixed media. Users can enter the medium of the art piece by either choosing one of the listed above or their own medium they want to check for. The query will search for anything that contains the searched item, *'<"%"+chosenMedium+"%">'* .This is done by the LIKE operator. If it is remotely similar, that data will be returned. If the art piece is purchased, it will go through the process of "If Purchased" and return purchase date and purchased by, as well as the earlier attributes. The art pieces will show up ordered by pieceID. The query is below:

SELECT * FROM ArtPiece WHERE medium LIKE *'<"%"+chosenMedium+"%>'* ORDER BY pieceID;

Searching Art Piece by Weight

Users can search art pieces by their weight. Users will choose the maximum weight of the art piece and the query will return associated information if it matches that condition. If the art piece is purchased, it will go through the process of "If Purchased" and return purchase date and purchased by, as well as the earlier attributes. The art pieces will show up ordered by pieceID. The query is below:

SELECT * FROM ArtPiece WHERE weight <= *'<chosenWeight>* ' ORDER BY pieceID;

Searching Art Piece by Dimensions

Art pieces can be search for dimensions by two methods: set dimensions and by a range of dimensions. Both check attributes of weight and height as conditions.

Searching Art Piece by Set dimensions

Users can search art pieces by their dimensions. Users are prompted with the standard dimensions of the art gallery. The user can choose a set width, '<*chosenWidth*>, and height, '<*chosenHeight*>, to search by for the artwork. If the art piece is purchased, it will go through the process of "If Purchased" and return purchase date and purchased by, as well as the earlier attributes. The art pieces will show up ordered by pieceID. The query is below:

SELECT * FROM ArtPiece WHERE width = '<*chosenWidth*>' AND height = '<*chosenHeight*>' ORDER BY pieceID;

Searching Art Piece by Range of dimensions

Users can search art pieces by their dimensions by a custom range. Users are prompted with the standard dimensions of the art gallery. The user can choose a set max width, '<*chosenMaxWidth*>', min width, '<*chosenMinWidth* >', max height, '<*chosenMaxHeight*>', and min height, '<*chosenMinHeight*>' ,to search by for the artwork. The query checks for data between these values. If the art piece is purchased, it will go through the process of "If Purchased" and return purchase date and purchased by, as well as the earlier attributes. The art pieces will show up ordered by pieceID. The query is below:

> SELECT * FROM ArtPiece WHERE (width BETWEEN '<*chosenMinWidth* >'AND '<*chosenMaxWidth*>') AND (height BETWEEN '<*chosenMinHeight*>' AND '<*chosenMaxHeight*>') ORDER BY pieceID;

Searching Art Piece by Price

Users can search art pieces by their listing price by a custom range. The user can choose a set max price, *'<chosenMaxPrice>'*, and minimum price, *'<chosenMinPrice>'* ,to search by for the artwork. The query checks for data between these values. If the art piece is purchased, it will go through the process of "If Purchased" and return purchase date and purchased by, as well as the earlier attributes. The art pieces will show up ordered by pieceID. The query is below:

> SELECT * FROM ArtPiece WHERE listPrice BETWEEN *'<chosenMaxPrice>'* AND *'<chosenMinPrice>* *'* ORDER BY listPrice;

Favorite artists

Users can see their favorite artists' works here. Only userTypes of customers can see this query. The query below fetches all data for the artwork where there is a relation between the customer, '<*currentCustomer*>', and the artist. f the art piece is purchased, it will go through the process of "If Purchased" and return purchase date and purchased by, as well as the earlier attributes. The art pieces will show up ordered by artistID. The query is below:

> SELECT * FROM ArtPiece JOIN (SELECT artistID FROM FavoriteArtist WHERE customerID = '<*currentCustomer*>') ON artistID = artist ORDER BY artistID;

Gallery

Gallery queries allow users to search for galleries and see their primary/favorite gallery.

View Favorite Gallery

Only customers can view their favorite gallery. If you are not logged in as a customer, you are prompted with line that tells you can only view favorite galleries as a customer. Artists are not able to set their own favorite gallery. If the query pulls no data, there is no primary location/ favorite gallery set for the customer. The customer can only have one favorite gallery. The query is below:

> SELECT * FROM Gallery WHERE storeID = (SELECT primaryLocation FROM Customer WHERE username = '<*currentUser*>');

Search Gallery by Address

All users: artists, customers, guests can search for galleries by address. Users can enter an input,'<*galleryString*>', to search for all addresses that contain the input. If data is found, the gallery's email, address, and phone number are returned. The user, if a customer, is asked if they would like to set the gallery as their primary location/favorite gallery and goes through "Set primary location". The query is below:

```
SELECT * FROM Gallery WHERE address LIKE '<galleryString>';
```

Set Gallery as Primary Location/Favorite Gallery

For each returned gallery and if the user is a customer, they are prompted if they would like to set the gallery as their primary location/ favorite store. This is a modification query, overwriting the data of primaryLocation '<*currentGallery*>' for the customer, '<*currentCustomer*>'.

```
UPDATE Customer SET primaryLocation = '<currentGallery>' WHERE username =
'<currentCustomer>';
```

Database Initialization Queries

The following SQL queries are used to initialize the database tables:

```
CREATE TABLE Customer(
   customerID INT NOT NULL
   gender VARCHAR(9),
   phoneNumber CHAR(12),
   email VARCHAR(254) NOT NULL,
   firstName VARCHAR(20) NOT NULL,
   lastName VARCHAR(20) NOT NULL,
   cardNumber CHAR(16),
   address VARCHAR(40),
   primaryLocation INT,
   username VARCHAR(20) NOT NULL,
   password VARCHAR(50) NOT NULL,
   PRIMARY KEY (customerID),
   FOREIGN KEY(primaryLocation) REFERENCES Gallery(StoreID)
);

CREATE TABLE Gallery(
   storeID INT NOT NULL,
   email VARCHAR(254) NOT NULL,
   address VARCHAR(40) NOT NULL,
   phoneNumber CHAR(12) NOT NULL,
   PRIMARY KEY(storeID)
);
```

```
CREATE TABLE FavoriteArtist(
   customerID INT NOT NULL,
   artistID INT NOT NULL,
   PRIMARY KEY(customerID, artistID),
   FOREIGN KEY(customerID) REFERENCES Customer,
   FOREIGN KEY(artistID) REFERENCES Artist
);

CREATE TABLE Artist(
   artistID INT NOT NULL,
   firstName VARCHAR(20) NOT NULL,
   lastName VARCHAR(20) NOT NULL,
   dob DATE,
   primaryMedium VARCHAR(50),
   biography VARCHAR(400),
   originCountry VARCHAR(50),
   username VARCHAR(20) NOT NULL,
   password VARCHAR(50) NOT NULL,
   PRIMARY KEY(artistID)
);

CREATE TABLE ArtPiece(
   pieceID INT NOT NULL,
   purchaseDate DATE,
   weight DECIMAL(8,2),
   width DECIMAL(8,2),
   height DECIMAL (8,2),
   condition VARCHAR(50),
   year DATE,
   style VARCHAR(50),
   medium VARCHAR(50),
   description VARCHAR(400),
   listPrice DECIMAL(11,2) NOT NULL,
   gallery INT NOT NULL,
   artist INT NOT NULL,
   customerID INT,
   PRIMARY KEY(pieceID),
   FOREIGN KEY(gallery) REFERENCES Gallery,
   FOREIGN KEY(artist) REFERENCES Artist
);
```