Case Study: Virtual Art Gallery

Project Objective:

To create a Virtual Art Gallery System that enables art enthusiasts to explore and manage artworks, artists, users, and galleries interactively.

Implementation Roadmap:

Design Database

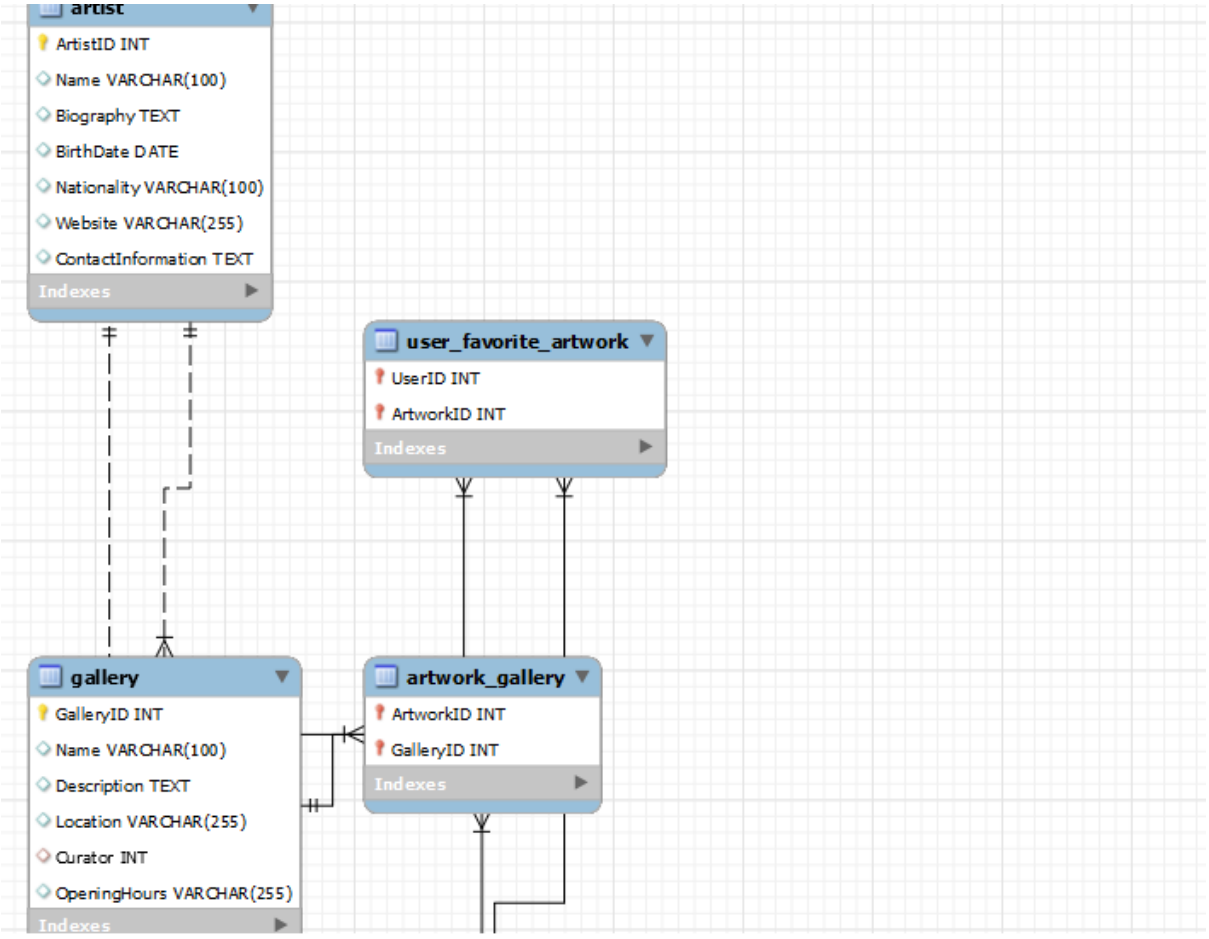
Create SQL tables (Artwork, Artist, User, Gallery) based on the provided schema.

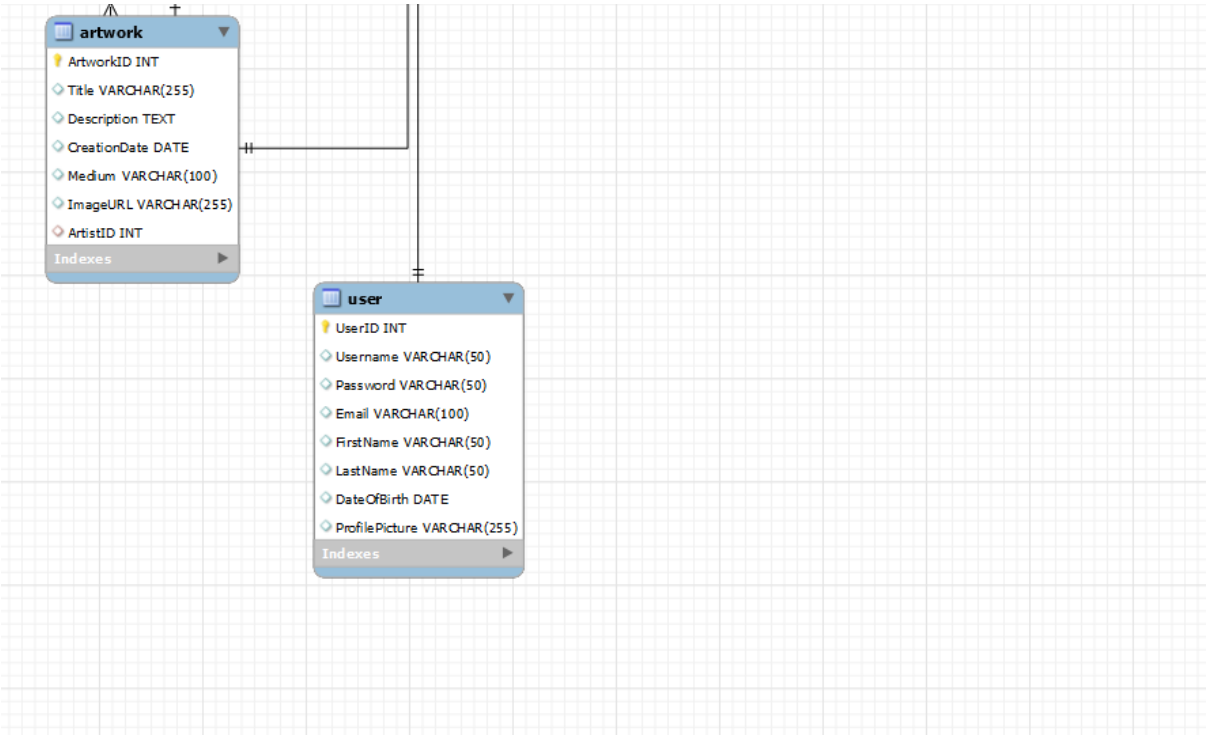
Classes

Develop Python classes for entities (Artwork, Artist, User, Gallery) within the `entity` package.

Define constructors, getters, setters, and necessary methods in these classes.

E-R diagram





Database connection:

Define `create_connection()` Function

This function is responsible for establishing a connection to the MySQL database.

connection parameters. The parameters include:

- **`host:`** The hostname where the database is located (`localhost` in this case).
- **`user`**: The username for accessing the database.
- **`password:`** The password of the current user..
- **`port:`** The port number for MySQL (`3306`).
- **`database`**: The name of the specific database to connect to (‘`vag`’ in this case)
- **`mysql.connector.connect()`** is used to create a connection object.
- If the connection is established successfully, the function returns the connection object.

Implemented functions

`addArtwork()`

- Purpose: Adds a new artwork to the gallery.
- Parameters: Artwork object containing details like title, description, creation date, medium, image URL.

- Return Type: Boolean indicating success or failure.

`removeArtwork()`

- Purpose: Removes an artwork from the gallery.
- Parameters: ArtworkID to identify the artwork to be removed.
- Return Type: Boolean indicating success or failure.

`getArtworkById()`

- Purpose: Retrieves an artwork by its ID.
- Parameters: ArtworkID to identify the artwork to retrieve.
- Return Type: Artwork object containing details of the artwork.

`searchArtworks()`

- Purpose: Searches for artworks based on a keyword.
- Parameters: Keyword to search within artwork titles or descriptions.
- Return Type: List of Artwork objects matching the search criteria.

`addArtworkToFavorite()`

- Purpose: Adds an artwork to a user's list of favorites.
- Parameters: UserID and ArtworkID to link the artwork as a favorite for the user.
- Return Type: Boolean indicating success or failure.

`removeArtworkFromFavorite()`

- Purpose: Removes an artwork from a user's list of favourites.
- Parameters: UserID and ArtworkID to unlink the artwork from the user's favourites.
- Return Type: Boolean indicating success or failure.

Exception handling:

1. `ArtWorkNotFoundException`

This custom exception is designed to handle situations where an invalid or non-existent ArtworkID is encountered while performing operations related to artworks.

2. `UserNotFoundException`

Similar to `ArtWorkNotFoundException`, this exception handles the scenarios where an invalid or non-existent UserID is encountered while handling user-related operations.