TABLE OF CONTENTS

|  |  |
| --- | --- |
| 1 | SYNOPSIS |
| 2 | PROJECT AIM AND OBJECTIVES |
| 3 | ABOUT THE PROJECT |
| 4 | DEVELOPMENT ENVIRONMENT |
| 5 | TABLE DESIGN |
| 6 | CODING |
| 7 | OUTPUT |
| 8 | BIBILIOGRAPHY |

SYNOPSIS

As the world becomes increasingly computer-dependent, this project introduces the Art Gallery Management System. It alleviates burdensome manual handling of artist records, artwork inventory, exhibitions, and sales, streamlining operations for art galleries and small exhibitions.With its user-friendly interface and secure data storage, the system improves data organization, reduces errors, and facilitates informed decision-making. This project aims to modernize art gallery operations, making management tasks simpler and more efficient.

The system provides easy access to data through various modules, enhancing productivity and efficiency. Developed using Python and MySQL, it offers a comprehensive solution for art gallery management.

**PROJECT AIM AND OBJECTIVES**

**AIM**

Develop a user-friendly Art Gallery Management System for efficient data handling and retrieval.

**OBJECTIVES**

* **Data Management**: Store detailed information about artists and artworks.
* **Efficient Retrieval**: Enable easy search and display of artworks.
* **Data Integrity**: Record and document sales transactions accurately.
* **Update/Delete Operations**: Allow updating and deleting artist information.
* **Exhibition Management**: Manage and schedule exhibitions.
* **User-Friendly Interface**: Provide an intuitive interface for easy interaction.

**ABOUT THE PROJECT**

The Art Gallery Management System is a notable contribution to the field of art galleries, allowing for easy retrieval and management of data related to artists, artworks, exhibitions, and sales transactions. This user-friendly program provides a comprehensive set of features that ensures efficient data handling.

This program can be used to store detailed information about artists, including their names, bios, and contact details. It also manages artwork information such as titles, mediums, sizes, creation dates, and descriptions. Users can add artworks and search for them by both their titles and the artist ID, making the retrieval process simple and efficient.

Furthermore, the program allows for updating artist information and deleting artists by either their ID or name. It displays available artworks in a well-organized table format and records sales transactions, ensuring all sales are properly documented. When an artwork is sold, the system handles the transaction by recording the sale details and maintaining data integrity.The program also manages exhibition schedules by allowing the addition of new exhibitions with details such as titles, start dates, end dates, and descriptions. This feature ensures that all exhibition information is readily available and easy to manage. When an exhibition ends, the system can remove the details from the current table and store them in another table for record-keeping.

Overall, this system ensures that all critical operations related to the management of an art gallery are handled efficiently and effectively, providing a comprehensive solution for data management. The user-friendly interface and robust features make it a valuable tool for any art gallery.

**DEVLOPMENT ENVIRONMENT**

**HARDWARE**

PROCESSOR : Intel(R) Core(TM) i5-10400 CPU @ 2.90GHz

RAM : 8.00 GB (7.83 GB usable)

HARD DISK :

CD ROM :

MONITOR :

KEYBOARD :

MOUSE :

**SOFTWARE**

**ABOUT PYTHON**

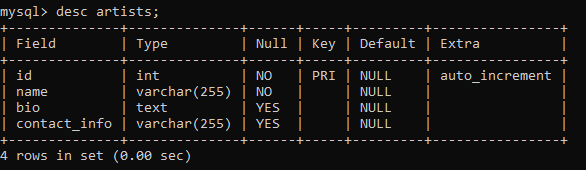
Python is an object-oriented programming (OOP) language. It was created by Guido Van Rossum and released in 1991. It is used for developing desktop GUI applications, websites, and web applications.

Advantages and Features of Python:

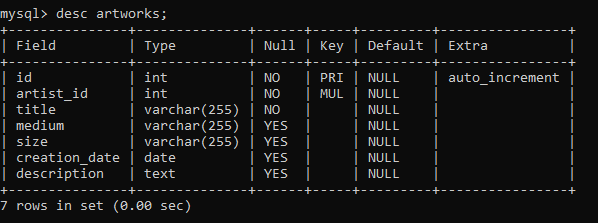
* Simple: Python is a simple and minimalistic language. The pseudo-code nature of Python is one of its greatest strengths.
* Easy to Learn: Python has an extraordinarily simple syntax, making it easier to learn.
* Free and Open Source: One can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs.
* High-Level Language: While writing programs in Python, one does not need to worry about low-level details such as managing the memory used by the program.
* Portable: Python can work on many platforms, such as Linux and Windows.
* Interpreted: Python does not require compilation to binary. The program is run directly from the source code.
* Extensible: A Python code can be written in C or C++ language that can be compiled in C/C++ language.
* Embeddable: Python can be embedded within C/C++ to give scripting capabilities for the program's users.
* Extensive Libraries: The Python Standard Library is a collection of script modules accessible to a Python program, simplifying the programming process and removing the need to rewrite commonly used commands. They can be used by calling/importing them at the beginning of a script**.**

**TABLE DESIGN**

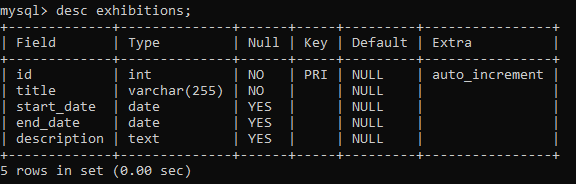
**Table : Artists**



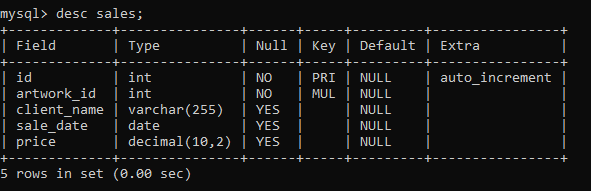
**Table : Artworks**



**Table : Exhibitions**

****

**Table : Sales**

****

**CODING:**

CREATE DATABASE art\_gallery;

USE art\_gallery;

CREATE TABLE artists (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

bio TEXT,

contact\_info VARCHAR(255));

CREATE TABLE artworks (

id INT AUTO\_INCREMENT PRIMARY KEY,

artist\_id INT NOT NULL,

title VARCHAR(255) NOT NULL,

medium VARCHAR(255),

size VARCHAR(255),

creation\_date DATE,

description TEXT,

FOREIGN KEY (artist\_id) REFERENCES artists(id));

CREATE TABLE exhibitions (

id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(255) NOT NULL,

start\_date DATE,

end\_date DATE,description TEXT);

CREATE TABLE sales (

id INT AUTO\_INCREMENT PRIMARY KEY,

artwork\_id INT NOT NULL,

client\_name VARCHAR(255),

sale\_date DATE,

price DECIMAL(10, 2),

FOREIGN KEY (artwork\_id) REFERENCES artworks(id));

import mysql.connector

from tabulate import tabulate

# Database connection

def get\_db\_connection():

return mysql.connector.connect(

host='localhost',

user='root',

password='admin@123', # replace with your MySQL password

database='art\_gallery')

# Function to add an artist

def add\_artist():

name = input("Enter artist's name: ")

bio = input("Enter artist's bio: ")

contact\_info = input("Enter artist's contact info: ")

conn = get\_db\_connection()

cursor = conn.cursor()

cursor.execute(

"INSERT INTO artists (name, bio, contact\_info) VALUES (%s, %s, %s)",

(name, bio, contact\_info) )

conn.commit()

cursor.close()

conn.close()

print("Artist added successfully!")

# Function to add an artwork

def add\_artwork():

artist\_id = int(input("Enter artist ID: "))

title = input("Enter artwork title: ")

medium = input("Enter artwork medium: ")

size = input("Enter artwork size: ")

creation\_date = input("Enter creation date (YYYY-MM-DD): ")

description = input("Enter artwork description: ")

conn = get\_db\_connection()

cursor = conn.cursor()

cursor.execute(

"INSERT INTO artworks (artist\_id, title, medium, size, creation\_date, description) VALUES (%s, %s, %s, %s, %s, %s)", (artist\_id, title, medium, size, creation\_date, description))

conn.commit()

cursor.close()

conn.close()

print("Artwork added successfully!")

# Function to add an exhibition

def add\_exhibition():

title = input("Enter exhibition title: ")

start\_date = input("Enter start date (YYYY-MM-DD): ")

end\_date = input("Enter end date (YYYY-MM-DD): ")

description = input("Enter exhibition description: ")

conn = get\_db\_connection()

cursor = conn.cursor()

cursor.execute(

"INSERT INTO exhibitions (title, start\_date, end\_date, description) VALUES (%s, %s, %s, %s)",(title, start\_date, end\_date, description))

conn.commit()

cursor.close()

conn.close()

print("Exhibition added successfully!")

# Function to add a sale

def add\_sale():

artwork\_id = int(input("Enter artwork ID: "))

client\_name = input("Enter client name: ")

sale\_date = input("Enter sale date (YYYY-MM-DD): ")

price = float(input("Enter sale price: "))

conn = get\_db\_connection()

cursor = conn.cursor()

cursor.execute(

"INSERT INTO sales (artwork\_id, client\_name, sale\_date, price) VALUES (%s, %s, %s, %s)",(artwork\_id, client\_name, sale\_date, price) )

conn.commit()

cursor.close()

conn.close()

print("Sale recorded successfully!")

# Function to delete an artist by ID or name

def delete\_artist():

choice = input("Delete by (1) ID or (2) Name? Enter choice: ")

conn = get\_db\_connection()

cursor = conn.cursor()

if choice == '1':

artist\_id = int(input("Enter artist ID to delete: "))

cursor.execute("DELETE FROM artists WHERE id = %s", (artist\_id,))

elif choice == '2':

name = input("Enter artist's name to delete: ")

cursor.execute("DELETE FROM artists WHERE name = %s", (name,))

else:

print("Invalid choice.")

return

conn.commit()

cursor.close()

conn.close()

print("Artist deleted successfully!")

# Function to search and display artist by ID or name

def search\_artist():

choice = input("Search by (1) ID or (2) Name? Enter choice: ")

conn = get\_db\_connection()

cursor = conn.cursor()

if choice == '1':

artist\_id = int(input("Enter artist ID to search: "))

cursor.execute("SELECT \* FROM artists WHERE id = %s", (artist\_id,))

elif choice == '2':

name = input("Enter artist's name to search: ")

cursor.execute("SELECT \* FROM artists WHERE name = %s", (name,))

else:

print("Invalid choice.")

return

artist = cursor.fetchone()

if artist:

print(tabulate([artist], headers=["ID", "Name", "Bio", "Contact Info"], tablefmt="pretty"))

else:

print("Artist not found.")

cursor.close()

conn.close()

# Function to display all details in a table format

def display\_all\_details():

conn = get\_db\_connection()

cursor = conn.cursor()

print("\nArtists:")

cursor.execute("SELECT \* FROM artists")

artists = cursor.fetchall()

print(tabulate(artists, headers=["ID", "Name", "Bio", "Contact Info"], tablefmt="pretty"))

print("\nArtworks:")

cursor.execute("SELECT \* FROM artworks")

artworks = cursor.fetchall()

print(tabulate(artworks, headers=["ID", "Artist ID", "Title", "Medium", "Size", "Creation Date", "Description"], tablefmt="pretty"))

print("\nExhibitions:")

cursor.execute("SELECT \* FROM exhibitions")

exhibitions = cursor.fetchall()

print(tabulate(exhibitions, headers=["ID", "Title", "Start Date", "End Date", "Description"], tablefmt="pretty"))

print("\nSales:")

cursor.execute("SELECT \* FROM sales")

sales = cursor.fetchall()

print(tabulate(sales, headers=["ID", "Artwork ID", "Client Name", "Sale Date", "Price"], tablefmt="pretty"))

cursor.close()

conn.close()

# Main function to interact with the user

def main():

while True:

print("\nArt Gallery Management System")

print("1. Add Artist")

print("2. Add Artwork")

print("3. Add Exhibition")

print("4. Add Sale")

print("5. Display All Details")

print("6. Delete Artist")

print("7. Search Artist by ID or Name")

print("8. Exit")

choice = input("Enter your choice: ")

if choice == '1':

add\_artist()

elif choice == '2':

add\_artwork()

elif choice == '3':

add\_exhibition()

elif choice == '4':

add\_sale()

elif choice == '5':

display\_all\_details()

elif choice == '6':

delete\_artist()

elif choice == '7':

search\_artist()

elif choice == '8':

break

else:

print("Invalid choice. Please try again.")

if \_\_name\_\_ == '\_\_main\_\_':

main()

**OUTPUT**

Art Gallery Management System

1. Add Artist

2. Add Artwork

3. Add Exhibition

4. Add Sale

5. Display All Details

6. Delete Artist

7. Search Artist by ID or Name

8. Exit

Enter your choice: 1

Enter artist's name: Pablo Picasso

Enter artist's bio: Spanish painter and sculptor

Enter artist's contact info: picasso@artgallery.com

Artist added successfully!

Art Gallery Management System

1. Add Artist

2. Add Artwork

3. Add Exhibition

4. Add Sale

5. Display All Details

6. Delete Artist

7. Search Artist by ID or Name

8. Exit

Enter your choice: 2

Enter artist ID: 1

Enter artwork title: Les Demoiselles d'Avignon

Enter artwork medium: Oil on canvas

Enter artwork size: 243.9 cm x 233.7 cm

Enter creation date (YYYY-MM-DD): 1907-06-19

Enter artwork description: One of Picasso's most famous works.

Artwork added successfully!

Art Gallery Management System

1. Add Artist

2. Add Artwork

3. Add Exhibition

4. Add Sale

5. Display All Details

6. Delete Artist

7. Search Artist by ID or Name

8. Exit

Enter your choice: 3

Enter exhibition title: Picasso's Masterpieces

Enter start date (YYYY-MM-DD): 2024-06-01

Enter end date (YYYY-MM-DD): 2024-09-30

Enter exhibition description: A collection of Picasso's most celebrated works.

Exhibition added successfully!

Art Gallery Management System

1. Add Artist

2. Add Artwork

3. Add Exhibition

4. Add Sale

5. Display All Details

6. Delete Artist

7. Search Artist by ID or Name

8. Exit

Enter your choice: 4

Enter artwork ID: 1

Enter client name: John Doe

Enter sale date (YYYY-MM-DD): 2024-07-15

Enter sale price: 5000000

Sale recorded successfully!

Art Gallery Management System

1. Add Artist

2. Add Artwork

3. Add Exhibition

4. Add Sale

5. Display All Details

6. Delete Artist

7. Search Artist by ID or Name

8. Exit

Enter your choice: 5

Art Gallery Management System

1. Add Artist

2. Add Artwork

3. Add Exhibition

4. Add Sale

5. Display All Details

6. Delete Artist

7. Search Artist by ID or Name

8. Exit

Enter your choice: 6

Delete by (1) ID or (2) Name? Enter choice: 1

Enter artist ID to delete: 1

Artist deleted successfully!

Art Gallery Management System

1. Add Artist

2. Add Artwork

3. Add Exhibition

4. Add Sale

5. Display All Details

6. Delete Artist

7. Search Artist by ID or Name

8. Exit

Enter your choice: 7

Search by (1) ID or (2) Name? Enter choice: 2

Enter artist's name to search: Pablo Picasso

+----+----------------+---------------------------------+----------------------+

| ID | Name | Bio | Contact Info |

+----+----------------+---------------------------------+----------------------+

| 1 | Pablo Picasso | Spanish painter and sculptor | picasso@artgallery.com|

+----+----------------+---------------------------------+----------------------+

Art Gallery Management System

1. Add Artist

2. Add Artwork

3. Add Exhibition

4. Add Sale

5. Display All Details

6. Delete Artist

7. Search Artist by ID or Name

8. Exit

Enter your choice: 8