**Art Gallery Database Management System**

**Project Report**

**Group 10**

**A person standing in a room with several paintings on the wall

Description automatically generated with low confidence**

**Presented by**

**Venkata Sai Bala Krishna Batchu - Netid:im2250**

**Chandratej Kurella - ja8071**

**Nagella Sai Rohith Reddy – tx8218**

**Rahul Chada -Hn8870**

**PROJECT Analysis & Life Cycle**

Diagram

Description automatically generated

When it comes to development of project in a Software development for a product or organization- Data base implementation becomes the primary aspect in storing the users of customers or any data.

The general dbms project flow is as mentioned above flow where we can see the stages of a dbms project Starting from User requirements to Database implementation to maintenance.

**TABLE OF CONTENTS**

**TITLES PAGE NO.**

**ABSTRACT** II

1. **INTRODUCTION** 1
   1. INTRODUCTION TO SQL 1
2. **REQUIREMENT SPECIFICATION 3**

2.1 SOFTWARE REQUIREMENTS 3

1. **OBJECTIVE OF THE PROJECT 4**
2. **IMPLEMENTATION 5**

4.1 ER DIAGRAM 5

4.2 MAPPING OF THE ER SCHEMA TO RELATIONS 7

4.3 CREATION OF TABLES 12

4.4 INSERTION OF Data 15

4.7 CREATION OF Joins, Views 18

4.8 CREATION OF STORED PROCEDURES 19

1. **Database management system maintenance &Further scope for improvement**

Abstract

This project intends to include various features related to an art gallery i.e. information about gallery, auction, artists, artwork, customers (the one who bought the paintings) etc to design and implementation of a Art Gallery Management. The main aim of this Art Gallery Management System is to implement efficient way of tracking and keeping details of art and paintings in art gallery also help art lovers know about different arts and colors used, the art work included, the artists. People who wish to buy this artwork can register and buy at an auction. Our Database stores data of the Transactions made at auction. Auction management staff data.

**1) INTRODUCTION:**

1.1 INTRODUCTION TO SQL:

As part of developing a Relational database as part of this course work we are going to using SQL programming language.

SQL which is an abbreviation for **Structured Query Language** is a language to request data from a database, to add, update, or remove data within a database, or to manipulate the metadata of the database.

Sometimes SQL is characterized as *non-procedural* because procedural languages generally, require the details of the operations to be specified, such as opening and closing tables, loading and searching indexes, or flushing buffers and writing data to file systems. Therefore, SQL is designed at a higher conceptual level of operation than procedural languages.

**2) REQUIREMENT SPECIFICATION**

**2.1 SOFTWARE REQUIREMENTS**

Operating System : Any Pc with atleast 8 gb RAM, 128 gb Hardrive memory

Database : MYSQL

**General description**

* Every year NEW IMAGE ART COMPANY organize an art exhibition in Los Angeles. The objective of the exhibition is to promote art and make artists ideas reach public. Company offers services like auction to establish a common platform for artists and art buyers. Exhibition features top art works of all kinds which also includes work done by renowned artists. The exhibition is open to all visitors and choose to participate in auction. However, participation in auction requires user registration. Information collected from Art Gallery is monitored and stored in our databases that includes information about Gallery, Customers, Staff, auctioneers, Artists and more.
  + Client has approached us to design a database for the Art Gallery Exhibition.
  + Gallery management software is designed to help gallery owners and operators with the daily tasks of running and maintaining an art gallery.
  + even performing important business functions such as billing and invoicing.
  + Gallery management software can track all artwork bought and sold by the gallery, maintain a database of contact information, help plan and execute exhibitions, and create a digital catalog.

**User Requirements**

* User requirements of this project is to develop database and display information of the GalleryInfo, Customers, Artists, Staff, Transaction, Auction and Artwork.
* To maintain the record and atomicity by using proper table to save records so that no one can access it and only one person purchases only one Artwork.
* Artist is identified with artist\_ID, and other attributes of this entity is Name, Age, Birthplace as other attributes.
* Artwork is developed by artist and is identified by art\_id. And other attributes are Artist\_name, year it is developed, its unique art title, type of art and colors used and room\_id.
* Artwork is purchased by customers, and is identified by customer\_registration\_id, and other attributes in the entity are, Name, Address, Phone number, Amount spent and interest.
* Gallery features art work and is identified Room\_id. Other entities are Room\_id, No\_of\_Arts, Cameras\_installed and staff\_id.
* Staff maintain gallery and identified with staff\_id, and other attributes are staff\_Name, Address, Phone and designation.Auction is managed by staff, and identified by Auction\_id, other attributes are start\_date, End\_date, Start\_time, End\_time,Bidprice ,Customer\_Registration\_id and art\_id.
* Transaction has attributes of art\_id, customer\_registration\_id.

This project allows three types of users:

* Staff
* Artist
* Customer

Scope of the project:

This project will help people to show their artistic skills. It can be used to make people aware of the upcoming exhibition i.e. its start date, end date, location. Also include artists whose paintings are available; people could contact them if they wish to learn from them or if they want to buy their art piece.

**4) IMPLEMENTATION:**

To Implement the Database based on User requirements, we need to develop an ERD and then a Relation Model, use the SQL commands to develop the relation database in MYSQL Software server.

Commonly used statements are grouped into the following categories

**Data Definition Language (DDL)**

CREATE

ALTER, DROP

**Data Manipulation Language (DML)**

INSERT - Used to Insert values into a entity.

UPDATE - Used to update the records or entity tables.

DELETE - Used to delete certain records or tables.

Joins are used to retrieve the required data by joining multiple tables based on criteria like inner or left or full.

Views are used for the viewing multiple tables, or a table and it retrieves data from original tables as views doesn’t have data.

Procedures are like Functions used in a programming language, where we define certain instructions and call those functions when required.

4.1 ER DIAGRAM:

* An **entity-relationship model (ER Model)** describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.
* An entity may be defined as a thing capable of an independent existence that can be uniquely identified. An entity is an abstraction from the complexities of a domain.
* entity-relationship (ER) diagram, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases.

Cardinality Relation

The cardinality rules are used to emphasize the connectivity of entities by using relations or lines and mentioning like one to one or one to many relations,etc.

Assumptions:

* One art can be purchased from an artist by gallery.
* One art can be sold to an at least one customer.
* Customer registration ID is validated for Auction
* One unique art will be present in one gallery room only.

Also, the Primary and Foreign key constraints where we join tables based on these keys, to retrieve data.

Diagram, schematic

Description automatically generated**Entity Relationship Diagram**

4.2 MAPPING OF THE ER SCHEMA TO RELATIONS

**Relation Model (Logical Database Design)**

Diagram

Description automatically generated with low confidence

**(Implementation in MySQL) code of various queries join,views,procedures, screenshots**

4.3 CREATION OF TABLES: **Using DDL**

**Art work table:**

create table artwork(

art\_id int(11) not null,

year\_developed int(11),

unique\_art\_title varchar(30),

type\_of\_art varchar(30),

colors\_used varchar(20),

artist\_id int(18),

room\_id int(11),

constraint artwork\_pk primary key (art\_id),

constraint artwork\_fk1 foreign key (artist\_id) REFERENCES artists(artist\_id),

constraint artwork\_fk2 foreign key (room\_id) REFERENCES gallery\_info(room\_id)

);

**Artwork**

A screenshot of a computer

Description automatically generated with medium confidence

**Output:**

**Graphical user interface, application

Description automatically generated**

**Artist Table:**

create table artists(

artist\_id int(18) not null,

artist\_name varchar(30),

artist\_age int(11),

artist\_birthplace varchar(30),

constraint artists\_pk primary key (artist\_id));

Graphical user interface, text

Description automatically generated

);

**Output:**

**Graphical user interface, table

Description automatically generated with medium confidence**

**Auction Table:**

create table auction(

auction\_id int(11) not null,

auc\_startdate date,

auc\_enddate date,

min\_bidprice double,

auc\_starttime time,

auc\_endtime time,

customer\_registration\_id int(11),

art\_id int(11),

constraint auction\_pk primary key (auction\_id),

CONSTRAINT auction\_fk1 foreign key(customer\_registration\_id) references customers(customer\_registration\_id),

CONSTRAINT auction\_fk2 foreign key(art\_id) references artwork(art\_id)

);

Graphical user interface, table

Description automatically generated

**Output:**

**Table

Description automatically generated**

**Customer Table:**

create table customers(

customer\_registration\_id int(11) not null,

customer\_name varchar(25),

customer\_Address varchar(35),

customer\_phone\_number int(11),

customer\_amount\_spent double,

customer\_interest varchar(30),

constraint customers\_pk primary key (customer\_registration\_id)

);

Graphical user interface, text

Description automatically generated with medium confidence

**Output:**

**Table

Description automatically generated**

**Gallery Info Table:**

create table gallery\_info(

room\_id int(11) not null,

no\_of\_arts int(11),

cameras\_installed int(11),

staff\_id int(11),

constraint gallery\_info\_pk primary key (room\_id),

constraint gallery\_info\_fk1 foreign key (staff\_id) REFERENCES staff(staff\_id)

);

Graphical user interface, text

Description automatically generated

**Output:**

**Table

Description automatically generated**

**Staff Table:**

create table staff(

staff\_id int(11) not null,

staff\_name varchar(30) not null,

Staff\_Address varchar(35),

staff\_phone int(11),

staff\_designation varchar(25),

constraint staff\_pk primary key (staff\_id)

);

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table

Description automatically generated with medium confidence

**Output:**

**Graphical user interface, table

Description automatically generated**

Transaction Table:

create table transaction(

art\_id int(11),

customer\_registration\_id int(11),

constraint transaction\_pk primary key (art\_id,customer\_registration\_id),

CONSTRAINT transaction\_fk1 foreign key(art\_id) references artwork(art\_id),

CONSTRAINT transaction\_fk2 foreign key(customer\_registration\_id) references customers(customer\_registration\_id)

);

Graphical user interface, application

Description automatically generated

**Output**

Table

Description automatically generated

**Auction Management**

create table auction\_management\_staff(

auction\_id int(11),

staff\_id int(11),

constraint auction\_mng\_pk primary key (auction\_id,staff\_id),

CONSTRAINT auction\_mng\_fk1 foreign key(auction\_id) references auction(auction\_id),

CONSTRAINT auction\_mng\_fk2 foreign key(staff\_id) references staff(staff\_id)

);

Graphical user interface, application, table, Teams

Description automatically generated

Output:

Table

Description automatically generated

Inserting values commands can be found in sql file

**Joins**

1. **Join1 customers and transactions**

SELECT customers.customer\_name,customers.customer\_interest,transaction.art\_id

FROM customers

JOIN transaction ON customers.customer\_registration\_id=transaction.customer\_registration\_id;

**Output**

Table

Description automatically generated

**2)Join2 galleryinfo and staff**

SELECT room\_id,no\_of\_arts,cameras\_installed

FROM gallery\_info

LEFT JOIN staff

ON gallery\_info.staff\_id = staff.staff\_id;

**Output**

Table

Description automatically generated

**3)Join3 artists and artwork**

SELECT art\_id,type\_of\_art

FROM artists

Right JOIN artwork

ON artists.artist\_id = artwork.artist\_id;

**Output**

Table

Description automatically generated

**Views**

**1)View and inner join to view customer attributes**

CREATE VIEW customerAuction

AS

SELECT customer\_phone\_number,

customer\_amount\_spent

FROM customers

INNER JOIN

Auction USING (customer\_registration\_id);

**Output**

Table

Description automatically generated

**2)View2 to view auction table**

CREATE VIEW test AS SELECT \* FROM auction;

**Output**

Table

Description automatically generated

Table

Description automatically generated

**3)View3 to view artwork,auction tables**

CREATE VIEW auctArt AS

SELECT L1.year\_developed, L1.type\_of\_art,L2.auction\_id,L2.min\_bidprice

FROM artwork L1, auction L2;

**Output**

Table

Description automatically generated

Table

Description automatically generated

**Procedure**

**1)Procedure1 to create customers attributes**

DELIMITER //

CREATE PROCEDURE Getcustomers()

BEGIN

select customer\_name,customer\_interest,customer\_phone\_number from customers;

END //

DELIMITER ;

CALL Getcustomers();

**Output**

Table

Description automatically generated

**2)Procedure2**

Procedure to calculate the no of years old the artwork is

DELIMITER $$

CREATE PROCEDURE art()

BEGIN

SELECT \*, year(curdate())-year\_developed as artold from artwork;

END$$

**Output**

Graphical user interface, application

Description automatically generated

**3)Procedure3 procedure to create customer amount spent on artwork**

DELIMITER $$

CREATE PROCEDURE amountspent()

BEGIN

SELECT customer\_registration\_id,customer\_amount\_spent

FROM customers

ORDER BY customer\_amount\_spent;

END$$

call amountspent();

**Output**

Table

Description automatically generated

**TRIGGER**

**Trigger ON artists TABLE TO CHANGING NAME TO UPPERCASE**

The trigger is made such that when a new record is inserted into a artists table, it automatically changes the lowercase name into uppercase in the backend.

DELIMITER $$

CREATE TRIGGER UPPERCASE

BEFORE INSERT on artists

FOR EACH ROW

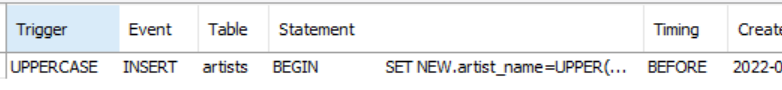
BEGIN

SET NEW.artist\_name=UPPER(NEW.artist\_name);

END$$

**Output**





**DATABASE MAINTENANCE**

After development of database, it’s also more important to maintain the database and its data by maintaining the duplicate database in any other server incase of server crash or data rollback issues.

**Scope to Increase in Future**

We can still more increase the scope of this project in future with the additional following entities if required by business.

Digital art Like NFT entity in future.

Tax Audits entity in future to help Millionaires who bought art to exempt the art purchase money with their business revenue.

**ACKNOWLEDGEMENT**

We express our deep and sincere thanks to our professor **yuan yuan gao** for her extensive support all the time and helping us to understand the DBMS concepts in a very new different way where we got to learn many new things.