DBMS PROJECT

TOPIC: ART GALLERY MANAGEMENT SYSTEM

Group Project:

Vuppala Vamsi -RA1811004010296

L.Sai Surya Sashank - RA1811027010087

Contents

1. INTRODUCTION:

Art Gallery Management System:

Objective: *To create an Art Gallery Management System that keeps record of artists, their paintings, art gallery details, exhibition details and showcases pictures of paintings to the customers.*

Abstract: *This project intends to include various features related to an art gallery i.e. information about gallery, exhibition, artists, their paintings, customers (the one who bought the paintings) etc. The main aim is to help art lovers know about different exhibition details organised by different organisations, the art work included, the artists. People who wish to buy this artwork can very easily contact the artist. There is a login page for administrator, artists and customers. Artists can login through their account and include details about new paintings made by them. Customers can login through their account and see different paintings available for selling purpose. Administrators can include details about the new art gallery , upcoming exhibition details and the exhibition artists.*

The activities of store that the system will automate are:

1.login

2. Entry of new art gallery

3. Entry of new artist

4. Entry of new Painting

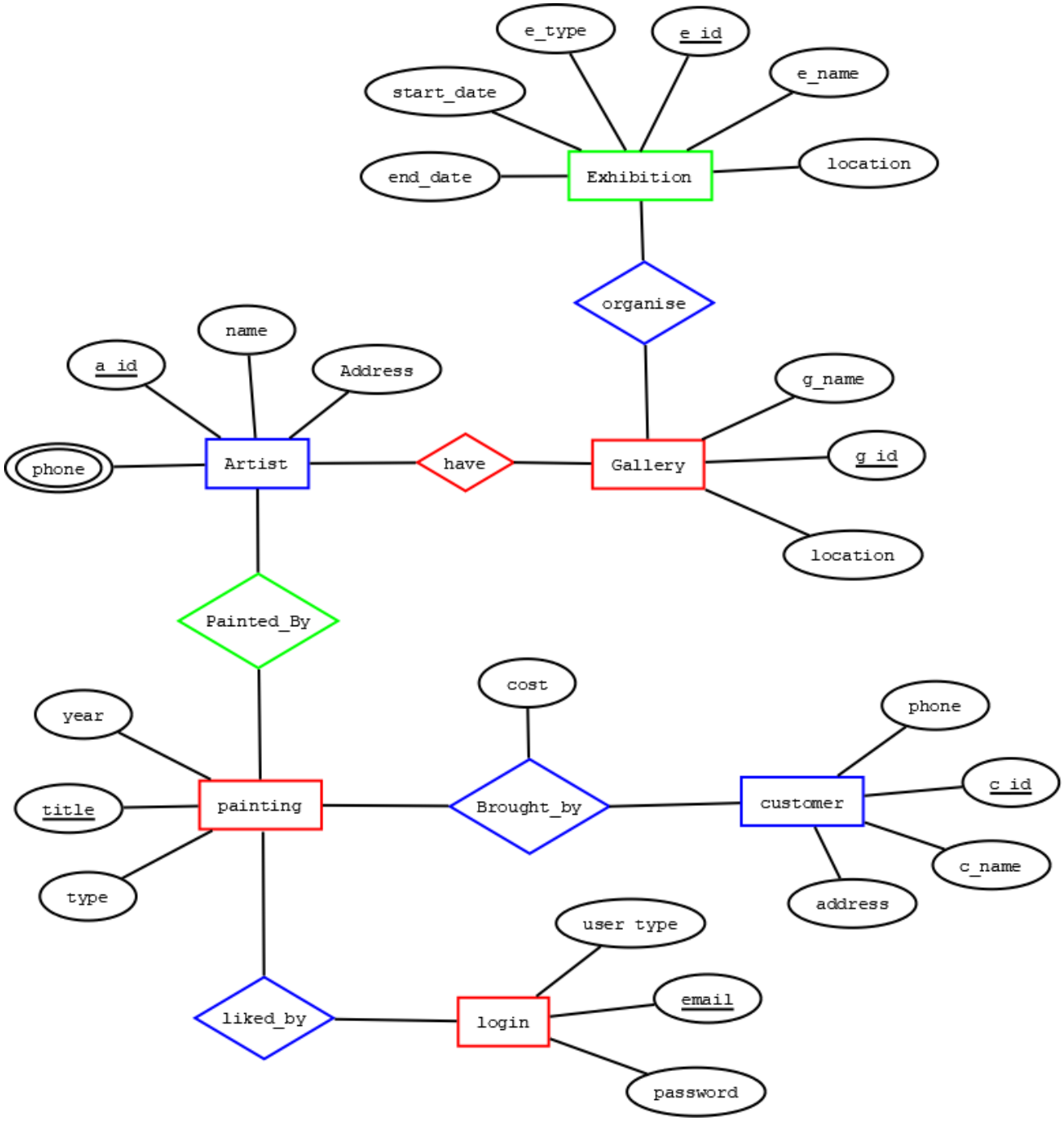
5. Entry of upcoming exhibition

6. Accessing contact details of the gallery as well as artists and their paintings.

2.Schemas:

- Artist(a_id:integer,name:char(10),address:varchar(20),phone:integer);
- Exhibition(e_id:integer,e_name:char(11),location:varchar(13),start_date:date,end_date:date);
- gallery(g_id:integer,g_name:varchar(14),location:char(25));
- customer(c_id:integer,c_name:varchar(16),address:varchar(11),phone:integer);
- painting(year:integer,type:char(17));
- login(email:varchar(238),usertype:char(11),password:char(14));
- painted_by(a_id:integer,title:char(16));
- Brought_by(c_id:integer,title:char(16),cost:integer);
- have(a_id:integer,g_id:integer);
- organise(e_id:integer,g_id:integer);

3.ER-DIAGRAM


e_type
e_id
e_name
start_date
end_date
Exhibition
location
organise
name
a_id
Address
g_name
phone
Artist
have
Gallery
g_id
location
Painted_By
cost
year
phone
c_id
title
painting
Brought_by
customer
c_name
type
address
user type
email
liked_by
login
password

RELATIONAL DATABASE SCHEMA

entities	Relationship	entity	Cardinality ratio
Artist	have	gallery	Many to one
Artist	Painted_by	painting	Many to many
Painting	liked_by	login	One to many
Painting	Brought_by	customer	One to one
Gallery	organise	exhibition	One to one

4.IMPLEMENTATION

CONVERTING ER MODEL TO TABLES

1 . Artist

Create: create table Artist(a_id integer, name char(10), address varchar(20),phone integer,Primary Key(a_id));

Insert:

```
insert into Artist values(201,'Reshma','Bangalore',8007060432);
insert into Artist values(202,'Mounish','chennai',8160416293);
insert into Artist values(203,'Tejaswi','visakhapatnam',8290458104);
insert into Artist values(204,'avan','ranchi',9156390152);
insert into Artist values(205,'Mehak','Hyderabad',99347104384);
insert into Artist values(206,'Vardhan','Bihar,9104529107);
```

SELECT COMMAND: **select * from Artist;**

OUTPUT:

A_ID	NAME	ADDRESS	PHONE
201	Reshma	Bangalore	8007060432
202	Mounish	chennai	8160416293
203	Tejaswi	visakhapatnam	8290458104
204	van	Ranchi	9156390152
205	Mehak	Hyderabad	99347104384
206	Vardhan	Bihar	9104529107

2.Customer

Create:create table customer(c_id integer, c_name char(16), address varchar(11),phone integer ,primary key(c_id));

Insert :

insert into customer values(101,'sruthi','vizag',9166567093);
insert into customer values(102,'Micheal','Hyderabad',9178546709);
insert into customer values(103,'Neha','chennai',8208456709);
insert into customer values(104,'preethi','Hyderabad',876543012);
insert into customer values(105,'Rahul','vizag',9176739028);
insert into customer values(106,'Racheal','Hyderabad',9130542960);

Select command:select * from customer;

OUTPUT:

C_ID	C_NAME	ADDRESS	PHONE
101	sruthi	vizag	9166567093
102	Micheal	Hyderabad	9178546709
103	Neha	chennai	8208456709
104	preethi	Hyderabad	876543012
105	Rahul	vizag	9176739028
106	Racheal	Hyderabad	9130542960

3.Gallery

Create:create table gallery(g_id integer, g_name char(55),location char(25),primary key(g_id));

INSERT:

insert into gallery values(212,'Resh art gallery','Bangalore');
insert into gallery values(222,'Mouni art gallery','chennai');
insert into gallery values(232,'Teja art gallery','visakhapatnam');
insert into gallery values(242,'pavan art gallery','Ranchi');
insert into gallery values(252,'sharma art gallery','Hyderabad');
insert into gallery values(262,'Varun art gallery','Banglore');

SELECT: select * from gallery;

OUTPUT:

G_ID	G_NAME	LOCATION
212	Resh art gallery	Bangalore
222	Mouni art gallery	chennai
232	Teja art gallery	visakhapatnam
242	avan art gallery	Ranchi
252	sharma art gallery	Hyderabad
262	Varun art gallery	Banglore

4.Exhibition

Create:create table Exhibition(e_id integer, e_name varchar(40),location varchar(13), start_date date, end_date date, Primary key(e_id));

INSERT:

insert into Exhibition values(12,'THE ART GALLERY EXHIBITION','chennai','20-november-2020','25-november-2020');

SELECT: select* from Exhibition;

OUTPUT:

E_ID	E_NAME	LOCATION	START_DATE	END_DATE
12	THE ART GALLERY EXHIBITION	chennai	20-NOV-20	25-NOV-20

5.Painting

Create: create table painting(title char(31) ,year integer, type char(41),Primary key(title));

INSERT:

insert into painting values('radha krishna','2019','pasRealism art');

insert into painting values('budha','2020','Photorealism Art');

insert into painting values('krishna','2019','Abstract Art.');

insert into painting values('radha','2020','Photorealism Art');

insert into painting values('forest','2018','Pop Art.');

insert into painting values('the joy of painting','2020','Photorealism Art');

SELECT: select *from painting;

OUTPUT:

TITLE	YEAR	TYPE
radha krishna	2019	pasRealism art
budha	2020	Photorealism Art
krishna	2019	Abstract Art.
radha	2020	Photorealism Art
forest	2018	Pop Art.
the joy of painting	2020	Photorealism Art

6.Login

Create: create table login(email varchar(238), usertype char(11), password char(14),primary key(email));

INSERT:

insert into login values('teja2@gmail.com','teju','Teju@24');

insert into login values('mehak3@gmail.com','sharma','mehak@23');

insert into login values('vardhan2@gmail.com','vardhan','vardhu@14');

insert into login values('mounish2@gmail.com','mani','mouni@44');

insert into login values('pavan2@gmail.com','adii','adiii@33');

insert into login values('resh2@gmail.com','reshma','reshu@24');

SELECT :select * from login;

OUTPUT:

EMAIL	USERTYPE	PASSWORD
teja2@gmail.com	teju	Teju@24
mehak3@gmail.com	sharma	mehak@23
vardhan2@gmail.com	vardhan	vardhu@14
mounish2@gmail.com	mani	mouni@44
avan2@gmail.com	adii	adiii@33
resh2@gmail.com	reshma	reshu@24

RELATIONSHIPS TABLES:

1.Painted_By:

CREATE:create table painted_by(a_id integer, title char(31), PRIMARY KEY(a_id,title),FOREIGN KEY(a_id) REFERENCES Artist(a_id),FOREIGN KEY(title) REFERENCES painting(title));

INSERT:

insert into painted_by values(201,'radha krishna');
insert into painted_by values(202,'budha');
insert into painted_by values(203,'krishna');
insert into painted_by values(204,'radha ');
insert into painted_by values(205,'forest');
insert into painted_by values(206,'the joy of painting');

SELECT: Select * from painted_by;

OUTPUT:

A_ID	TITLE
201	radha krishna
202	budha
203	krishna
204	radha
205	forest
206	the joy of painting

Using ALTER Command:

create table painted_by(a_id integer, title char(31));
ALTER table painted_by add constraint pk PRIMARY KEY(a_id,title),
ALTER table painted_by add constraint fk FOREIGN KEY(a_id) REFERENCES Artist(a_id);
ALTER table painted_by add constraint pfk1 FOREIGN KEY(title) REFERENCES painting(title));

2.Organise:

Create:create table organise(e_id integer, g_id integer, PRIMARY KEY(e_id,g_id), FOREIGN KEY(e_id) REFERENCES Exhibition(e_id),FOREIGN KEY(g_id) REFERENCES gallery(g_id));

INSERT:

insert into organise values(12,212);

insert into organise values(12,222);
insert into organise values(12,232);
insert into organise values(12,242);
insert into organise values(12,252);
insert into organise values(12,262);

SELECT:select * from organise;

OUTPUT:

E_ID	G_ID
12	212
12	222
12	232
12	242
12	252
12	262

3.BROUGHT_BY:

CREATE:create table brought_by(title char(31), c_id integer, cost integer, PRIMARY KEY(title,c_id),FOREIGN KEY(title) REFERENCES painting(title),FOREIGN KEY(c_id) REFERENCES customer(c_id));

INSERT:

insert into brought_by values('radha krishna',101,3000);
insert into brought_by values('budha',102,4000);
insert into brought_by values('krishna',103,20000);
insert into brought_by values('radha ',104,5000);
insert into brought_by values('forest',105,8000);
insert into brought_by values('the joy of painting',106,5000);

SELECT:

select * from brought_by;

OUTPUT:

TITLE	C_ID	COST
radha krishna	101	3000

budha	102	4000
krishna	103	20000
radha	104	5000
forest	105	8000
the joy of painting	106	5000

4.Have:

Create:create table have(a_id integer,g_id integer,PRIMARY KEY(a_id,g_id),FOREIGN KEY(a_id) REFERENCES Artist(a_id),FOREIGN KEY(g_id) REFERENCES gallery(g_id));

INSERT

insert into have values(201,212);
insert into have values(202,222);
insert into have values(203,232);
insert into have values(204,242);
insert into have values(205,252);
insert into have values(206,262);

SELECT:select * from have;

OUTPUT:

A_ID	G_ID
201	212
202	222
203	232
204	242
205	252
206	262

COMMANDS

DDL COMMANDS

- *CREATE*
- *ALTER*
- *DROP*
- *TRUNCATE*

ALTER TABLE command to add a New Column:

Alter table Artist add age integer;

Output:

A_ID	NAME	ADDRESS	PHONE	AGE
201	Reshma	Bangalore	8007060432	-
202	Mounish	chennai	8160416293	-
203	Tejaswi	visakhapatnam	8290458104	-
204	avan	Ranchi	9156390152	-
205	Mehak	Hyderabad	99347104384	-
206	ardhan	Banglore	9104529107	-

ALTER TABLE command to DROP COLUMN :

ALTER TABLE Artist DROP COLUMN age;

Output:

```
Table altered.
```

A_ID	NAME	ADDRESS	PHONE
201	Reshma	Bangalore	007060432
202	Mounish	chennai	160416293
203	Tejaswi	visakhapatnam	290458104
204	avan	Ranchi	156390152
205	Mehak	Hyderabad	9347104384

206	Vardhan	Banglore	104529107

Download CSV

6 rows selected.

- ***USING ALTER Command to MODIFY column address***

  ALTER TABLE customer modify address varchar(30);

- ***USING ALTER Command RENAMEthe column name:***

  ALTER TABLE Customer **RENAME COLUMN** address to addr;

OUTPUT FOR ALTER COMMAND:

Table altered.

Table altered.

C_ID	C_NAME	ADDR	PHONE
101	sruthi	vizag	9166567093
102	Micheal	Hyderabad	9178546709
103	Neha	chennai	8208456709
104	preethi	Hyderabad	876543012
105	Rahul	vizag	9176739028
106	Racheal	Hyderabad	9130542960

TRUNCATE TABLE command:

TRUNCATE TABLE Artist;

OUTPUT:

```
Table truncated.
```

Data Manipulation Language

- INSERT
- UPDATE
- DELETE

UPDATE COMMAND:

UPDATE gallery
set g_name='tejuu art gallery'
where g_id=232;

➔ OUTPUT:

G_ID	G_NAME	LOCATION
212	Resh art gallery	Bangalore
222	Mouni art gallery	chennai
232	tejuu art gallery	visakhapatnam
242	avan art gallery	Ranchi
252	sharma art gallery	Hyderabad
262	Varun art gallery	Banglore

.

1 row(s) updated.

DELETE COMMAND:

DELETE FROM login
WHERE email ='teja2@gmail.com';

OUTPUT:

EMAIL	USERTYP	PASSWORD
mehak3@gmail.co	sharma	mehak@23
vardhan2@gmail.com	vardhan	vardhu@14
mounish2@gmail.com	mani	mouni@44
pavan2@gmail.com	adii	adiii@33
reshu2@gmail.com	reshma	reshu@24

Select commands

SQL SELECT DISTINCT Statement

```
select distinct address
from Artist;
```

OUTPUT:

ADDRESS
Hyderabad
Ranchi
Bangalore
chennai
visakhapatnam
Banglore

SQL WHERE CLAUSE:

```
SELECT * FROM Artist

WHERE name='Tejaswi';
```

OUTPUT:

A_ID	NAME	ADDRESS	PHONE
203	Tejaswi	visakhapatnam	8290458104

SQL AND, OR and NOT Operators:

SELECT * FROM Artist
WHERE *name='Reshma'* AND *address='Bangalore'*;

Output:

A_ID	NAME	ADDRESS	PHONE
201	Reshma	Bangalore	8007060432

The OR operator displays a record if any of the conditions separated by OR is TRUE.

SELECT * FROM Artist
WHERE *name='Mehak'* or *address='delhi'*;

Output:

A_ID	NAME	ADDRESS	PHONE
205	Mehak	Hyderabad	99347104384

ORDER BY CLAUSE:

The ORDER BY keyword is used to sort the result-set in ascending or descending order.

*SELECT * FROM Artist*

ORDER BY name, a_id;

OUTPUT:

A_ID	NAME	ADDRESS	PHONE
204	avan	Ranchi	9156390152
205	Mehak	Hyderabad	99347104384
202	Mounish	chennai	8160416293
201	Reshma	Bangalore	8007060432
203	Tejaswi	visakhapatnam	8290458104
206	Vardhan	Banglore	9104529107

SQL MIN() and MAX() FunctionSThe MIN() function returns the smallest value of the selected column.

```
select min(c_id)
from customer;
```

output:

MiN(C_ID)
101

The MAX() function returns the largest value of the selected column.

```
select max(c_id)
from customer;
```

OUTPUT:

MAX(C_ID)
106

SQL COUNT(), AVG() and SUM() FunctionsThe COUNT() function returns the number of rows that matches a specified criterion.

```
SELECT COUNT(c_id)
FROM customer
```

OUTPUT

COUNT(C_ID)
6

The AVG() function returns the average value of a numeric column.

SELECT AVG(*c_id*)

FROM *customer*;

OUTPUT:

(C_ID)
103.5

The SUM() function returns the total sum of a numeric column.

SELECT SUM(c_id)

FROM customer;

OUTPUT:

SUM(C_ID)
621

VIEWS

CREATE VIEW:

```
CREATE VIEW Artist_VIEW AS
SELECT name,address
FROM Artist;
```

SELECT:

```
select *  from Artist_view;
```

OUTPUT:

```
View created.
```

NAME	ADDRESS
Reshma	Bangalore
Mounish	chennai
Tejaswi	visakhapatnam
avan	Ranchi
Mehak	Hyderabad
Vardhan	Banglore

DELETE VIEW :

```
DELETE FROM Artist_View
 WHERE name = 'Vardhan';
```

SELECT:

```
select *  from Artist_view;
```

OUTPUT:

NAME	ADDRESS
Reshma	Bangalore
Mounish	chennai
Tejaswi	visakhapatnam
avan	Ranchi
Mehak	Hyderabad

DROP VIEWS:

DROP VIEW Artist_view;

OUTPUT:

View dropped

Queries:

1. write an sql query to display the title of the painting which is published in the most recent year?

Query:

```
select title from painting
where year=(select max(year) from painting);
```

OUTPUT:

TITLE
budha
radha
the joy of painting

2. Write an sql query to display the gallery name whose id is second highest?

Query using nested subquery:

```
select g_name from gallery
where g_id in( select max(g_id) from gallery where g_id <> (select max(g_id) from gallery));
```

OUTPUT:

G_NAME
sharma art gallery

3.Write an sql query to SQL BETWEEN Operator:

SELECT g_id

FROM gallery
WHERE g_id BETWEEN 212 AND 242;

OUTPUT:

G_ID
212
222
232
242

*4.*Write an sql query to *NOT BETWEEN:*

SELECT g_id
FROM gallery
WHERE g_id NOT BETWEEN 212 AND 242;

OUTPUT:

G_ID
252
262

5.Write an sql query to *SQL LIKE Operator:*

SELECT * FROM Customer
WHERE c_name LIKE 'R%';

OUTPUT:

C_ID	C_NAME	ADDRESS	PHONE
105	Rahul	vizag	9176739028
106	Rache	Hyderab	9130542960

	al	ad	

6.Write an sql query to SQL IN Operator

SELECT * FROM Customer

WHERE address IN ('vizag', 'chennai');

OUTPUT:

C_ID	C_NAME	ADDRESS	PHONE
101	sruthi	vizag	9166567093
103	Neha	chennai	8208456709
105	Rahul	vizag	9176739028

7.Write an sql query to NOT IN OPERATOR:

SELECT * FROM Customer

WHERE address NOT IN ('vizag', 'chennai');

output:

C_ID	C_NAME	ADDRESS	PHONE
102	Micheal	Hyderabad	9178546709
104	preethi	Hyderabad	876543012
106	Racheal	Hyderabad	9130542960

8. Write an sql query to ORDER BY CLAUSE:

```
SELECT * FROM Artist
  ORDER BY name;
```

OUTPUT:

A_ID	NAME	ADDRESS	PHONE
204	avan	Ranchi	9156390152
205	Mehak	Hyderabad	99347104384
202	Mounish	chennai	8160416293
201	Reshma	Bangalore	8007060432
203	Tejaswi	visakhapatnam	8290458104
206	Vardhan	Banglore	9104529107

PL/ SQL

PL/SQL stands for "Procedural Language extensions to the Structured Query Language".

1. *WELCOME TO ART GALLERY:*

```
begin
dbms_output.put_line('welcome to art gallery');
End;
```

OUTPUT:

```
Statement processed.
welcome to art gallery
```

2.factorial of given number:

```
declare
num int :=21;
```

```
fact int :=1;
temp int;
begin
if(num>20)then
temp:=num;
while(num>0)
loop
 fact := fact * num;
 num := num-1;
end loop;
dbms_output.put_line('number less than 20');
  end if;
  end;
```

OUTPUT:

```
Statement processed.
number less than 20
```

3.Area of perimeter:

```
declare
area float;
perimeter float;
radius int:=5;
pi float:=3.14;
begin
area :=pi * radius * radius;
perimeter :=2 * pi * radius;
dbms_output.put_line('Area ='||area);
```

```
dbms_output.put_line('perimeter ='||perimeter);
end;
```

OUTPUT:

```
Statement processed.
Area =78.5
perimeter =31.4
```

4.pattern:

```
declare
i int;
j int;
n int:=5;
begin
for i in 1..n loop
for j in 1..i loop
dbms_output.put('*');
end loop;
dbms_output.new_line;
end loop;
End;
```

OUTPUT

```
Statement processed.
*
**
***
****
```

*****

5.Fibonacci series

```
declare
a number := 0;
b number := 1;
temp number;
n number:=6;
i number;
begin
  dbms_output.put_line('fibonacci series is :');
  dbms_output.put_line(a);
  dbms_output.put_line(b);
  for i in 2..n
  loop
   temp:=a + b;
   a:=b;
   b:=temp;
   dbms_output.put_line(temp);
   end loop;
End;
```

OUTPUT:

```
Statement processed.
```

```
fibonacci series is :
0
1
1
2
3
5
8
```

6.CURSORS

```
Create table salary(sid int,name varchar(20),age int,sal int);
insert into salary values(1,'tejaswi',19,21000);
insert into salary values(2,'mehak',21,51000);
insert into salary values(3,'mounish',20,20000);
insert into salary values(4,'reshma',34,34000);
insert into salary values(5,'pavan',45,45000);
insert into salary values(6,'rahul',33,33000);
declare
c1 int;
begin
update salary set sal=33333 where age>30;
c1:=sql%rowcount;
dbms_output.put_line(c1||'rows updated successfully');
end;
```

OUTPUT :

```
Statement processed.
3rows updated successfully
```

7.LENGTH OF A STRING :

```
DECLARE
    test_string string(20) := 'ART GALLERY';
Begin
    dbms_output.put_line(length(test_string));
End;
```

OUTPUT:

```
Statement processed.
11
```

8.CONCAT:

```
DECLARE
    test_string string(20) := 'ART GALLERY';
    test_string2 string(20) := 'management system';
BEGIN

    dbms_output.put_line(concat(test_string,test_string2));
End;
```

OUTPUT:

```
Statement processed.
```

ART GALLERY management system

FUNCTION:

Creation of function:

```
Create  or replace f unction welcome_msg_f un(p_name IN
VARCHAR2)
RETURN VARCHAR2
IS
BEGIN
RETURN('welcome'||p_name);
End;
```

OUTPUT:

Function created

Declare function

```
declare
invite_msg VARCHAR(250);
BEGIN
invite_msg:= welcome_msg_f un('art gallery');
dbms_output.put_line(invite_msg);
End;
```

OUTPUT:

Statement processed.

welcome art gallery

NORMALIZATION

Database normalization is the process of organizing the attributes of the database to reduce or eliminate data redundancy (having the same data but at different places)

FIRST NORMAL FORM:

A_ID	NAME	ADDRESS	PHONE
201	Reshma	Bangalore	8007060432,7693027304
202	Mounish	chennai	8160416293,6598983984
203	Tejaswi	visakhapatnam	8290458104,7538050065
204	pavan	Ranchi	9156390152,null
205	Mehak	Hyderabad	99347104384,623985740
206	Vardhan	Bihar	69104529107,null

Form the above category table we can see that it consists of atomicity .We can normalize the table into 1NF

A_ID	NAME	ADDRESS	PHONE	PHONE2
201	Reshma	Bangalore	8007060432	864873982
202	Mounish	chennai	8160416293	7649365905
203	Tejaswi	visakhapatnam	8290458104	654905479
204	pavan	Ranchi	9156390152	-
205	Mehak	Hyderabad	99347104384	7309540454
206	Vardhan	bihar	9104529107	-

So above table is in the form of 1NF

SECOND NORMAL FORM:

A_ID	NAME	ADDRESS	PHONE
201	Reshma	Bangalore	8007060432
202	Mounish	chennai	8160416293
203	Tejaswi	visakhapatnam	8290458104
204	pavan	Ranchi	9156390152
205	Mehak	Hyderabad	99347104384
206	Vardhan	bihar	69104529107

Form the above category table we can see that it consists of atomicity .We can normalize the table into 2NF

A_ID	*PHONE2*
201	*8007060432*
202	*948736585*
203	*8290458104*
204	*-*
205	*-*
206	*9104529107*

So above table is in the form of 2NF

THIRD NORMAL FORM:

Artist table:

A_ID	*NAME*	*ADDRESS*	*LOCATION*	*PHONE*
206	*Vardhan*	*bihar*	*gaya street*	*9104529107*

203	*Tejaswi*	*visakhapatnam*	*yendada*	*8290458104*
205	*Mehak*	*Hyderabad*	*LB nagar*	*99347104384*
204	*pavan*	*ranchi*	*basanti market*	*9156390152*
201	*Reshma*	*Bangalore*	*white field*	*8007060432*
202	*Mounish*	*chennai*	*mint street*	*8160416293*

Form the above category table we can see that it consists of atomicity .We can normalize the table into 3NF

Normalized artist table:

A_ID	NAME	ADDRESS	PHONE
201	Reshma	Bangalore	8007060432
202	Mounish	chennai	8160416293
203	Tejaswi	visakhapatnam	8290458104
204	pavan	Ranchi	9156390152
205	Mehak	Hyderabad	99347104384
206	Vardhan	bihar	69104529107

A_id	LOCATION
202	mint street
201	white field
204	basanti market
203	yendada
205	LB nagar
206	gaya street

So above table is in the form of 3NF

IMPLEMENTATION:

INTERFACE

HTML CODE FOR WEB PAGE

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport"  CContent="width=device-width,
nitial-scale=1.0">
<meta http-equiv="X-UA-compatible"  ocontent="ie-edge">
<title>Insertion form</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
<div class="main">
<div class="insert">
<h2>Insert data into artist's table</h2>
<form id="insertion" action="select.php"   method="POST">
<label for="A_ID">A_ID</label>
<div>
<input type="text" name="A_ID" id="A_ID">
</div>
<br>
<label for="NAME">NAME</label>
<div>
```

```
        <input type="text" name="NAME" id="NAME">
    </div>
    <br>
    <label for="ADDRESS">ADDRESS</label>
    <div>


        <br><textarea name="ADDRESS" id="ADDRESS" cols="30"
rows="10"></textarea>
    </div>
    <br>
    <label for="PHONE_NO">PHONE NO</label>
    <div>
        <input type="text" name="PHONE_NO" id="PHONE_NO">
    </div>
    <br>
    <div>
        <button type="submit" class="a"      value="Insert">INSERT</b
button type="reset" class="a"
value="reset">RESET</button>
    </div>
 </form>
</div>
</div>
</body>
 </html>
```

Css Code FOR WEB PAGE

```
 *{
margin: 0;
padding: 0;



body{
background-color: rgb(186, 134, 235);
div.main{
width: 400px;
margin: 100px auto 0px auto;
h2{
text-align: center;
font-family: sans-serif;
padding: 20px;
liv.insert{
background-color: rgba(0,0, 0, 0.5);
width: 100%;
font-size: 18px;
border-radius: 10px;
```

```
 border: 1px solid rgba(255,255,255,0.3);
 box-shadow: 2px 2px 15px rgba(0,0,0,3);
 color: #fff;

form#insertion{
 margin: 40px;

 Label{
 font-family: sans-serif;
 font-size: 18px;
 Input#A_ID{
 border-radius: 5px;

input#NAME{
 border-radius: 5px;
 input#ADDRESS{
 border-radius: 5px;
input#PHONE_NO{
 border-radius: 5px;
```

```
a{
 border-radius: 8px;
 padding:10px;
 margin-right: 5px;
```

WEB PAGE

CONNECTIVITY

INSERT VALUES:

CONNECTIVITY

INSERT VALUES:

```
<?php
$A_ID = $_POST['A_ID'];
 $NAME = $_POST['NAME'];
 $ADDRESS = $_POST['ADDRESS'];
 $PHONE_NO = $_POST['PHONE_NO'];


  $conn= new mysqli('localhost','root','','art');
 if($conn->connect_error) {
     die('connection failed:'.$conn->connect_error);
 }
 else {
  $a = $conn->prepare("insert into artist values(?,?,?,?)");
  $a->bind_param("issi",$A_ID,$NAME,$ADDRESS,$PHONE_NO);
  $a->execute();
  echo "inserted successfully";
  $a->close();
  $conn->close();
 }
?>
```

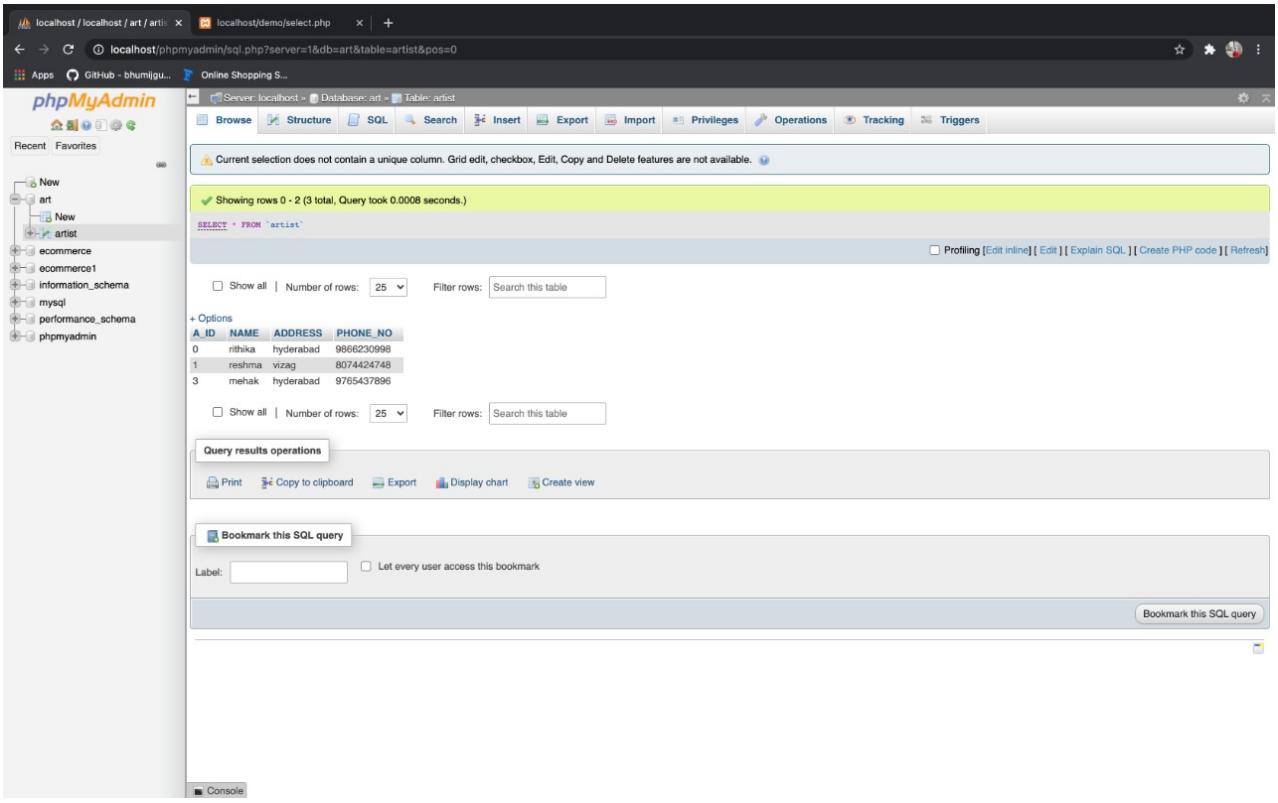
SELECT:

```
<?php
session_start();
    $conn=mysqli_connect("localhost","root","","art");
   $query="SELECT * from artist ";
   $rs=$conn->query($query);
   if($rs->num_rows>0) {
    echo "<table border=2><tr><th style=color:blue;>A_ID</th><th
style=color:blue;>NAME</th><th style=color:blue;>ADDRESS</th><th
style=color:blue;>PHONE_NO</th></tr>";
    while($row=$rs->fetch_assoc()) {
      echo
"<tr><td>".$row['A_ID']."</td><td>".$row['NAME']."</td><td>".$row['ADDR
ESS']."</td><td>".$row['PHONE_NO']."</td></tr>";
    }
    echo "</table>";
}
else {
"0 results";
}
mysqli_close($conn);
?>
```

RESULT:

Insertion of artist details

Fetching of data from database



Using where condition to fetch data

PHP CODE TO WHERE CONDITION

```
<?php
session_start();
    $conn=mysqli_connect("localhost","root","","art");
   $query="SELECT * from artist where A_ID=1 ";
   $rs=$conn->query($query);
   if($rs->num_rows>0) {
    echo "<table border=2><tr><th
style=color:blue;>A_ID</th><th
style=color:blue;>NAME</th><th
style=color:blue;>ADDRESS</th><th
style=color:blue;>PHONE_NO</th></tr>";
```

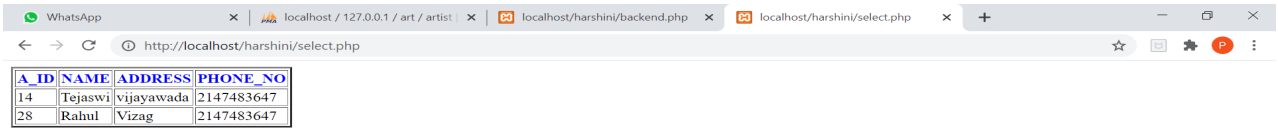
```
    while($row=$rs->fetch_assoc()) {
     echo
"<tr><td>".$row['A_ID']."</td><td>".$row['NAME']."</td><td>".$row['A
DDRESS']."</td><td>".$row['PHONE_NO']."</td></tr>";
    }

  echo "</table>";
}
else {
"o results";
}
mysqli_close($conn);
?>
```

OUTPUT FOR WHERE CONDITION

A_ID	NAME	ADDRESS	PHONE_NO
1	reshma	vizag	8074424748

OUTPUT FOR SELECT COMMAND:



A_ID	NAME	ADDRESS	PHONE_NO
14	Tejaswi	vijayawada	2147483647
28	Rahul	Vizag	2147483647

Conclusion:

This project is built keeping in mind that any number of Customers can login and visit the exhibition. They can like a painting and buy it.They can also view the name of the artist and their respective cost.We can add a new artist,new paintings and we can also update and delete artists and a painting name.We also connected frontend and backend using PHP.Through this connection we can insert values in frontend and store them in database. Thus with this

framework we can save time and manage records efficiently.

THANK YOU