A DBMS Project Report on

CREATIQ-ONLINE ART SYSTEM

By:

HRITUJA KHATAVKAR

RASHI DHIR

ROHAN ATHAWADE

Index

INTRODUCTION:

Art of any kind is worth being experienced by more than just the creator. A method for this to transpire is usually developed by galleries, which organize art exhibitions and auctions at a given date and place. This also unfortunately means that customers and visitors must be present at the venue in order to admire the original works of art, which must be put on display.

Often, during the run of the exhibition, customers find themselves edging towards admiring a creation so much that they wish to acquire it. For this to take place, they must first contact the manager of the gallery, ask whether it is on sale or not, then follow the complex procedure further to finally purchase the work.

On the flipside however, it is a task numerously more times difficult for the art gallery to maintain and keep a record of artistic creations, storage and availability, events being hosted, and tracking the purchases being made by customers.

It is then clearly evident that there is a dire need for a system that allows this to take place more seamlessly and without the actual hassle of having to visit a gallery to buy a painting (and often discovering that it has already been sold to someone else), or to manually keep a record of an entire gallery. At such a situation, automating the entire process, from exhibiting to selling art, can be done with the help of an Online Art System.

An Online Art System is an online website that exhibits, sells and collects art works and art pieces (not necessarily paintings exclusively). The core purpose of an Online Art System is to allow customers to explore their artistic hobbies and interests simply at home, without the trouble of having to travel back and forth between real exhibitions and auctions. While the Online Art System streamlines the process of buying a creation (paintings, sculptures and handicrafts), it also is a platform that showcases these creations in the form of real-life exhibitions and events that are organized from time to time. This same system also doubles as a software created for the organizers and handlers of the gallery as a means to maintain and preserve the gallery and its works itself by storing and analyzing metadata about the creations, exhibiting the works on the website, as well as allowing and maintaining track of the transactions that have taken place between the customer and themselves regarding a creation that was bought.

Therefore, the purpose of this project is to depict how an art gallery might employ such a system and store, organize and make use of the data that is needed for the functioning of this system. In essence, this project aims to replicate a database that captures the organizational

hierarchy/complex of the data that is required in general, as well as the data that the various functions offered and their responses may demand and collect in order to fully function.

The database and its working described further is a scaled attempt in sorting data used from all perspectives, views and operation of the system from all types of users, not only restricted to customers that are willing to purchase a creation.

PROBLEM STATEMENT:

To provide an online platform to manage the buying, selling and storage of artistic creations provided by creators/other platforms or auctions, and to organize events to exhibit these products.

OBJECTIVES:

The broad objective of this system is to streamline the process of exhibiting, admiring and acquiring art pieces from at the extreme comfort of the user.

Specifically, the objective of the database and diagram depicted below is to describe and showcase the organization of the data required to run such a system in reality. The database depicts many modules, and their objectives can be understood as:

- To allow users to login to the system portal as customer, admin, etc.
- To allow users to explore paintings, sculptors or handicrafts exhibited on the website in the storage of the organizing gallery.
- To allow users to participate or register for upcoming events.
- To allow users to purchase or request a purchase of a particular item or creation.
- To allow art galleries to showcase their collections online.
- To allow art galleries to place these collections and pieces of art on sale.
- To allow art galleries to keep a record of events being organized.
- To allow art galleries to keep a record of art pieces being collected.
- To allow art galleries to keep a record of the specifications and details about the type of creation, it’s creator and its buyer or seller.
- To allow art galleries to keep a record of the total number and specifications of creations that they host, and its status.
- To allow art galleries to provide a means for customers to pay for and acquire artistic creations they wish to possess.

FUNCTIONAL REQUIREMENTS:

The functionality of the entire system is broadly divided between two basic working perspectives and actual applications of the system, that is, from the point of the user (that may be assumed to be either a customer or an artist/creator) and that of the gallery organizers.

Therefore, it follows that the functional requirements of the system are divided into the User Module and the Organizer Module. The functional requirements that arise from doing so are described further:

- **USER MODULE**

  The user module will be used by mainly two types of users: customers and artists/creators.

  - *Customer:*
    1. Login/Register: the user should be able to login using the portal provided or register to be a suggested type of user (customer).
    2. View creations: the user after logging in should be able to view various pieces of work provided by artists in the gallery, or all works by a particular artist/gallery.
    3. View upcoming events: the user should be able to view a list of upcoming events and exhibitions being organized.
    4. Register for events: the user should be able to register for upcoming events being organized.
    5. Pay for Creations: the user should be able to select and pay for creations they would like to possess through the payments module.

  - *Artist:*
    1. Login/Register: the artist should be able to login using the portal provided or register to be a suggested type of user (artist).
    2. View creations: the artist after logging in should be able to view various pieces of work provided by other artists in the gallery, or all works by a particular artist/gallery.
    3. View upcoming events: the artist should be able to view a list of upcoming events and exhibitions being organized.

4. Participate in upcoming events: the artist should be able to take part in provide creations for upcoming online exhibitions and events being organized.
5. Register for galleries: the artist should be able to register for one or more than one gallery of their choice.
6. Submit creations: the artist should be able to submit their own creations for display at a gallery or an exhibition.

❖ **ORGANIZER MODULE**

The Organizer module consists of a various number of people that may use the system from the "organizational perspective". For example, this may include the gallery manager, the events manager, creations manager, etc.

1. Login/Register: the organizer should be able to login using the portal provided or register for the same.
2. Start a gallery: the organizer should be able to start a gallery of their own post registration under which exhibitions and selling/buying of creations will be possible.
3. Upload Creations: the organizer should be able to upload creations that are new depending on the type of creation it is (sculpture, painting or handicraft).
4. Add artists to gallery: the organizer should be able to add artists to galleries.
5. Organize exhibitions: the organizer should be able to organize exhibitions or other events and upload them onto the website.
6. Accept payments: the organizer should be able to accept payments made by the customers for a given creation.
7. Queue Sold Creations: the organizer should be able to declare a creation as sold and remove it from the displayed creations.
8. Add artists to exhibitions: the organizer should be able to add participating artists to exhibitions.
9. Add customers/users to exhibitions: the organizer should be able to add registering customers/users to upcoming events and exhibitions.

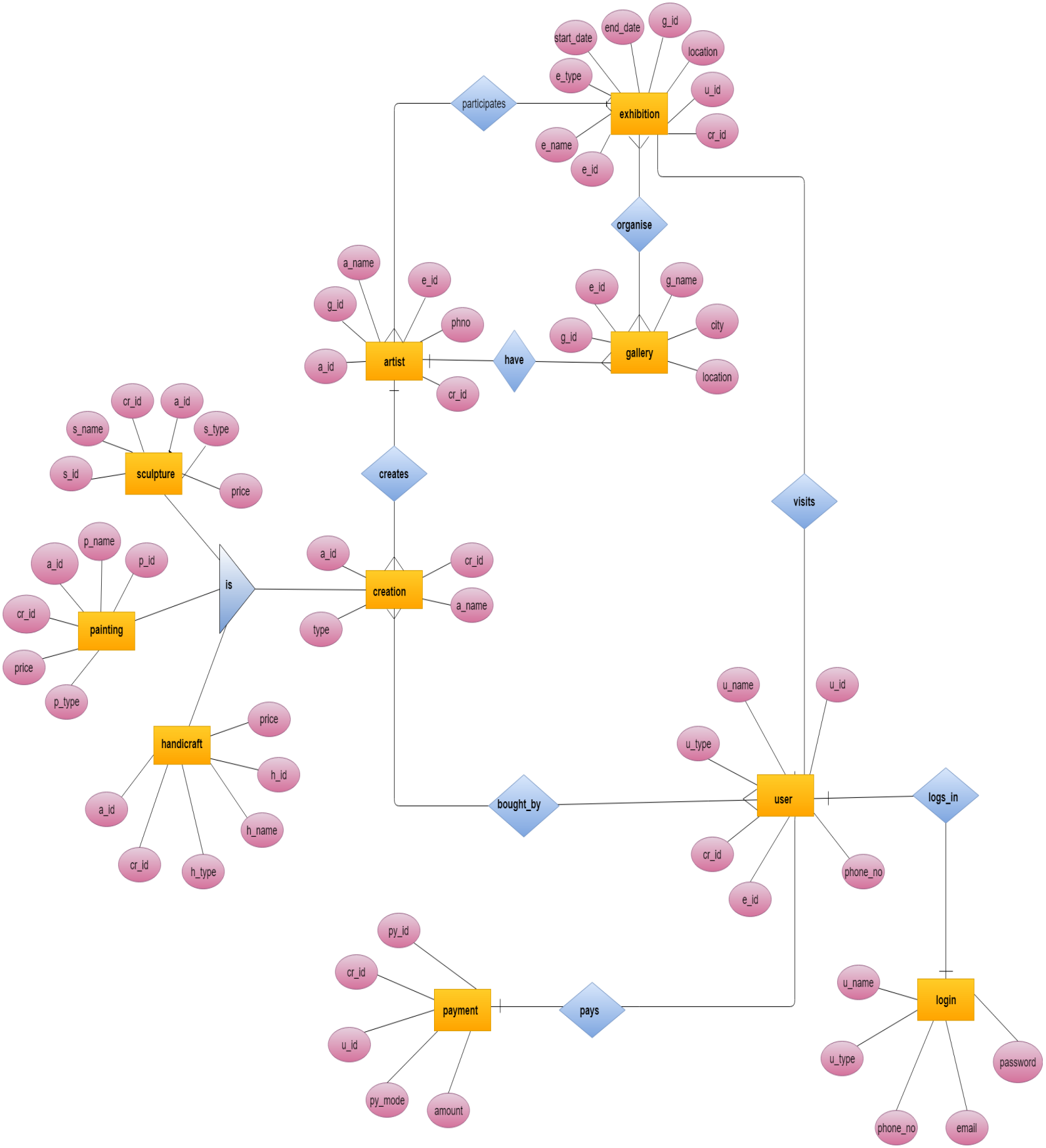
ENTITES AND RELATIONSHIPS:

The relationships between the various entities in our database is described in the table below. The first and third column state the entities in question and the second column describes the relationship between them:

ENTITY	RELATIONSHIP	ENTITY
Gallery	A gallery can have multiple artists whose creations are on exhibition and sale from their side.	Artist
Gallery	A gallery may organize multiple exhibitions, but a particular exhibition belongs to one gallery only.	Exhibition
Artist	An artist may create multiple creations, but a creation belongs to only one artist.	Creation
Artist	An artist can participate in multiple exhibitions, and an exhibition can host multiple artists' work.	Exhibitions
Creation	A creation can be of any type; either in the form of a painting, sculpture or handicraft, but one these is all essentially a singe creation.	Painting,Sculpture, Handicraft
Creation	Multiple creations can be bought by a single user, and a single user can own a single creation.	User
User	A single user may pay using the payments portal, and a	Payment

	payment belongs to only one user.	
User	A single user has a singular set of login credentials at the login portal, but the login portal has many users.	Login
User	A user may register to many exhibitions, and an exhibition may have many users registering.	Exhibition

E-ER DIAGRAM:


start_date
end_date
g_id
location
e_type
u_id
participates
exhibition
cr_id
e_name
e_id
organise
a_name
e_id
g_id
phno
a_id
artist
have
cr_id
e_id
g_name
g_id
city
gallery
location
cr_id
a_id
s_name
s_type
s_id
sculpture
price
creates
visits
p_name
a_id
p_id
cr_id
painting
price
p_type
is
a_id
cr_id
creation
a_name
type
price
handicraft
h_id
a_id
h_name
cr_id
h_type
u_name
u_id
u_type
user
logs_in
bought_by
cr_id
phone_no
e_id
py_id
cr_id
payment
pays
u_id
py_mode
amount
u_name
login
u_type
password
phone_no
email

RELATIONAL SCHEMA:

EXHIBITION					
e_id	e_name	location	e_type	start_date	end_date

GALLERY			
g_id	g_name	g_url	location

ORGANIZE	
e_id	*g_id*

ARTIST			
a_id	a_name	address	phone_no

HAVE	
a_id	*g_id*

HANDICRAFT					
h_id	h_name	cr_id	a_id	h_type	price

SCULPTURE					
s_id	s_name	*cr_id*	*a_id*	s_type	price

PAINTING					
pr_id	p_name	*cr_id*	*a_id*	p_type	price

USER		
u_id	u_name	phone_no

LOGIN					
l_id	u_id	u_type	email	password	phone_no

PAYMENT				
py_id	py_mode	*cr_id*	*u_id*	amount

CREATES	
a_id	**cr_id**

VISITS	
e_id	*u_id*

LOGS_IN	
u_id	*l_id*

PAYS	
u_id	*py_id*

PARTICIPATE	
e_id	*a_id*

BOUGHT_BY	
cr_id	*u_id*

IS	
spec_id	spec_name

PRIMARY KEY – BOLD

FOREIGN KEY - ITALICS

ANOMALIES IN THE RELATIONAL SCHEMA AND NORMALIZATION:

Insertion Anomaly – Insertion anomaly is when new information needs to be added in any relation but the data cannot be entered due to constraints.

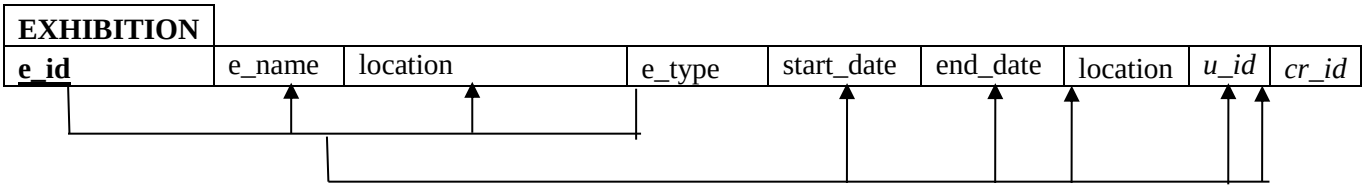
Deletion Anomaly—The deletion anomaly occurs when the deletion of some information causes the unintended deletion of other information.

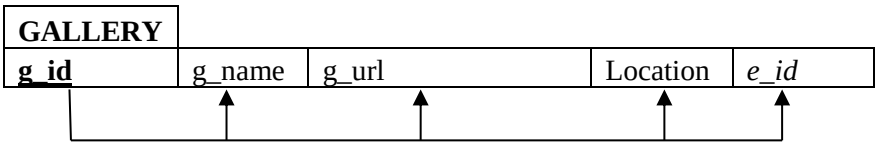
Updation Anomaly – The updation anomaly occurs when the updating of some data causes unintended inconsistency in other data.

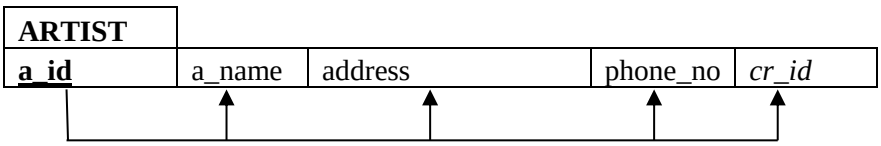
There are NO updation, insertion or deletion anomalies in this relational schema.

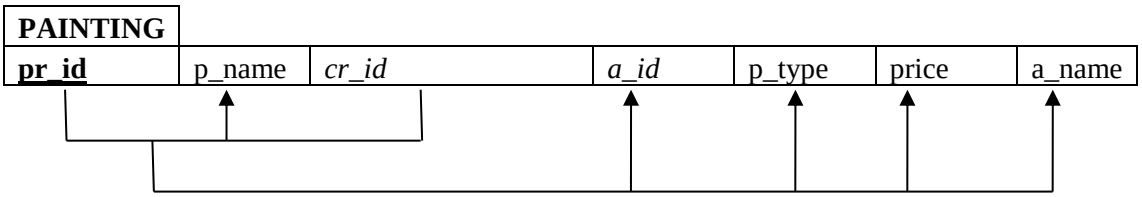
Hence the schema is Normalized up to 3NF.

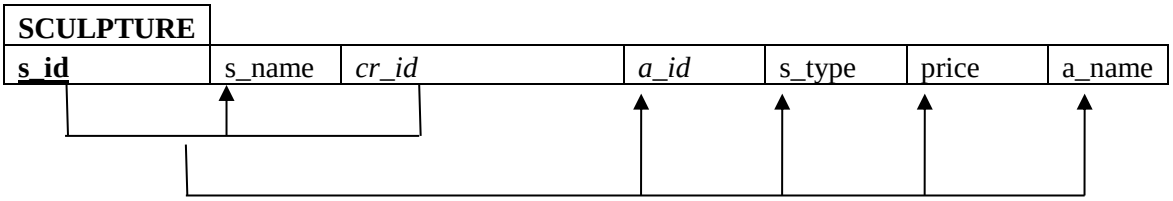
FUNCTIONAL DEPENDENCIES AND FUNCTIONAL DEPENDENCY CHART:

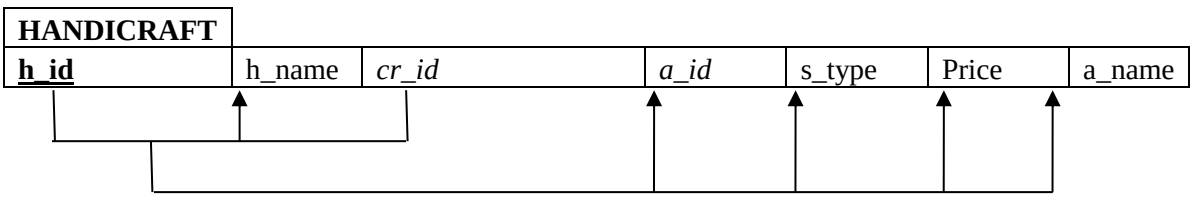


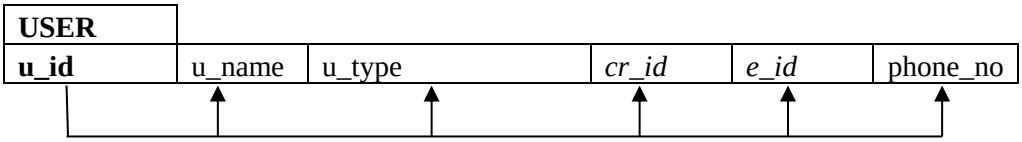


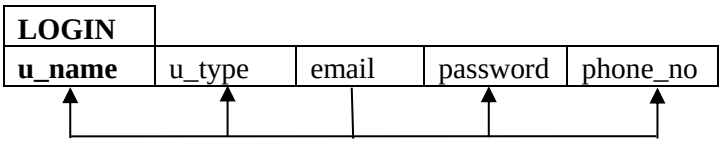


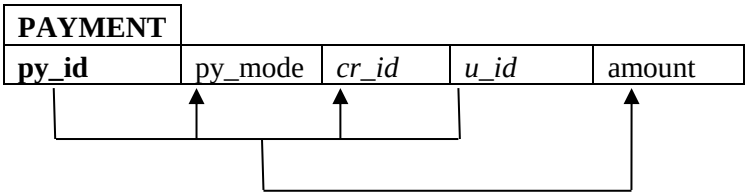






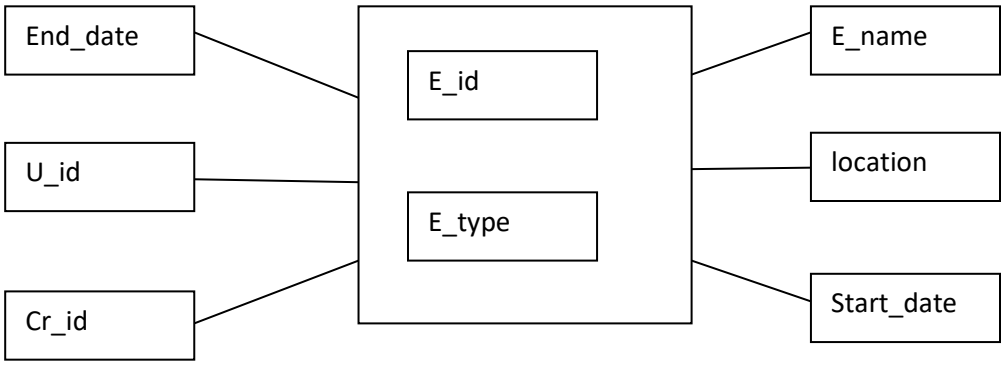

USER
u_id
u_name
u_type
cr_id
e_id
phone_no


LOGIN
u_name
u_type
email
password
phone_no

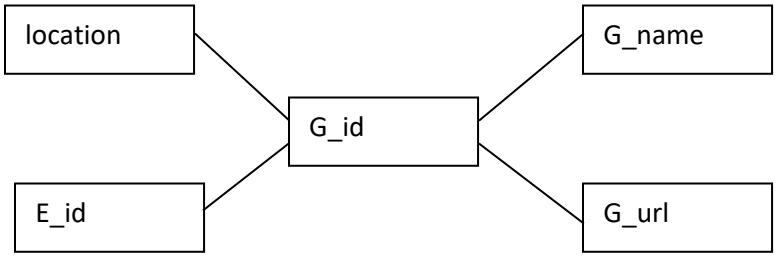

PAYMENT
py_id
py_mode
cr_id
u_id
amount

DEPENDENCY CHARTS:

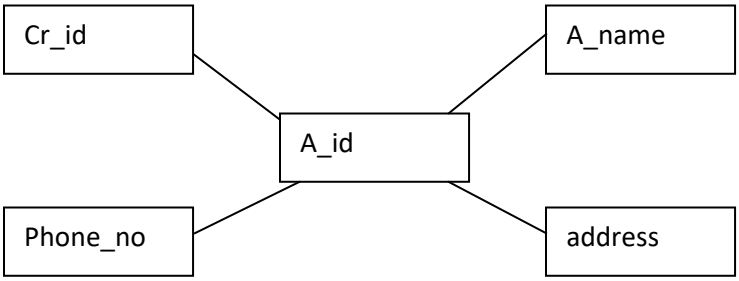
Exhibition



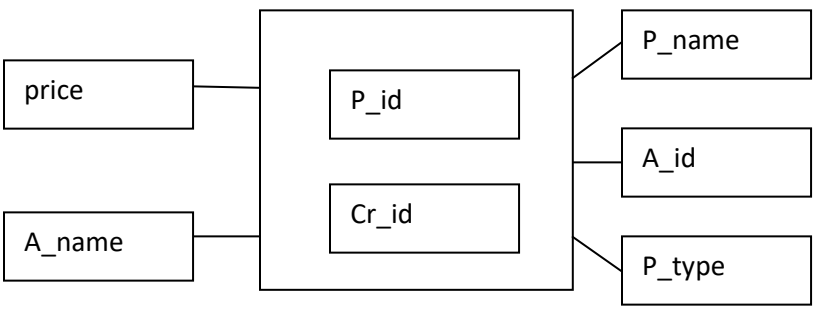
Gallery



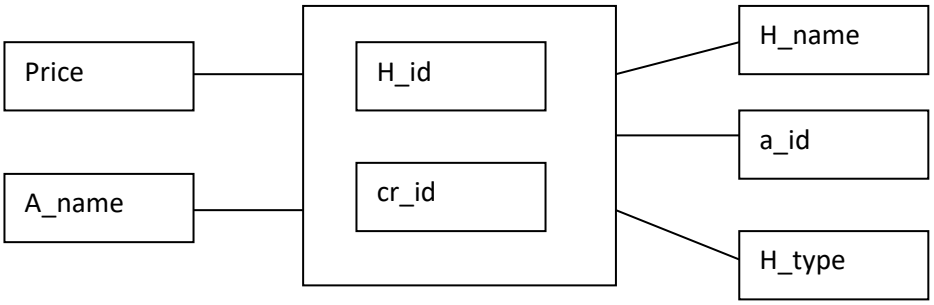
Artist



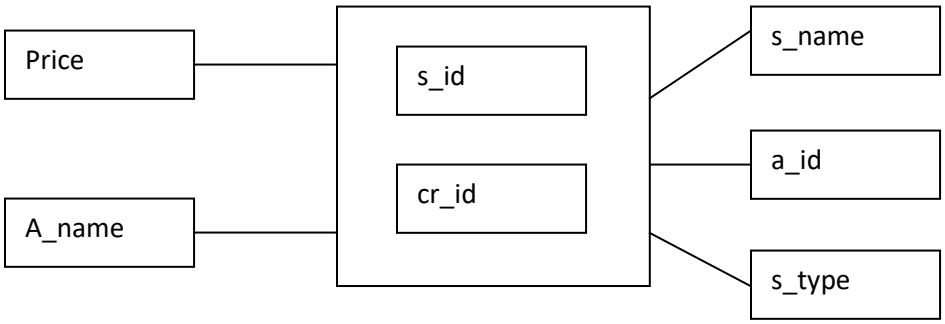
Painting



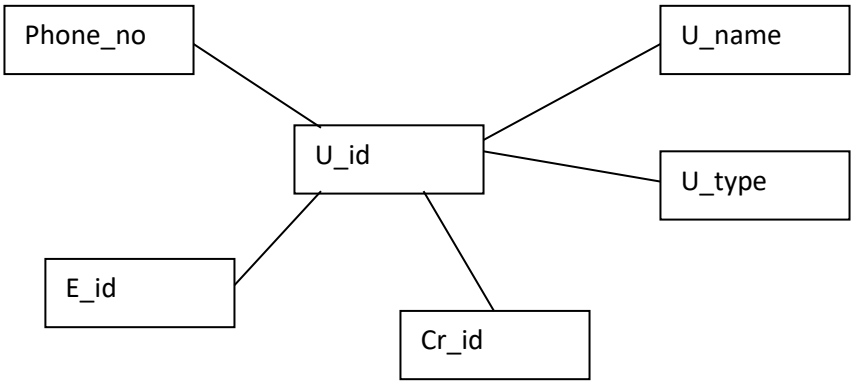
Handicraft



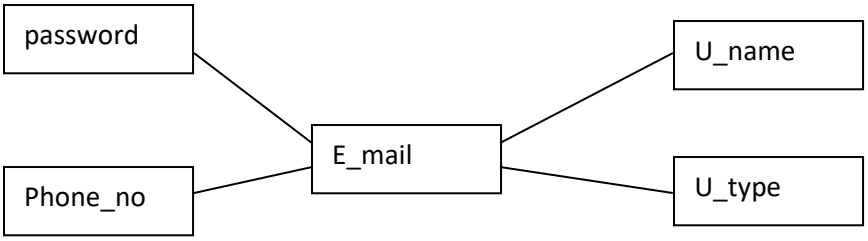
Sculpture



User



Login



IMPLEMENTING THE TABLES IN MYSQL:

```
create table EXHIBITION (e_id int primary
key,
 e_name varchar(225),
location varchar(225),
 e_type varchar(225),
start_date datetime,
 end_date datetime);


create table GALLERY(g_id int primary key,
 g_name varchar(225),
 g_url varchar(225),
 location varchar(225),
 e_id int,
foreign key (e_id) references exhibition(e_id));


create table ORGANIZE(e_id int,
 g_id int,
foreign key (e_id) references exhibition(e_id),
foreign key (g_id) references gallery(g_id)
);


create table ARTIST(a_id int primary key,
 a_name varchar(225),
 address varchar(225),
 phone_no varchar(225));


create table participate(e_id int,
 a_id int,
foreign key (e_id) references exhibition(e_id),
foreign key (a_id) references artist(a_id)
);


create tables CREATES(cr_id int primary key,
a_id int,
foreign key (a_id) references artist(a_id)
);


CREATE TABLE USER
(
u_id int primary key,
u_name varchar(225),
phone_no varchar(225)
);


CREATE TABLE SCULPTURE
( s_id int primary key,
s_name varchar(225),
cr_id int,
a_id int,
foreign key (cr_id) references artist(cr_id),
foreign key (a_id) references artist(a_id),
```

```
price float
);

CREATE TABLE HANDICRAFT
( h_id int primary key,
h_name varchar(225),
cr_id int ,
a_id int,
foreign key (cr_id) references artist(cr_id),
foreign key (a_id) references artist(a_id),
h_type varchar(25),
price float
);

CREATE TABLE PAINTING
( p_id int primary key,
p_name varchar(225),
cr_id int ,
a_id int,
foreign key (cr_id) references artist(cr_id),
foreign key (a_id) references artist(a_id),
p_type varchar(25),
price float
);

CREATE TABLE LOGIN(
l_id int primary key,
u_id int,
foreign key (u_id) references user(u_id),
u_type varchar(225),
email varchar(225),
password varchar(60),
phone_no int
);

CREATE TABLE LOGS_IN(
l_id int,
u_id int,
foreign key (l_id) references login(l_id),
foreign key (u_id) references user(u_id)
);

CREATE TABLE PAYS(
py_id int,
u_id int,
foreign key (py_id) references payment(py_id),
foreign key (u_id) references user(u_id)
);

CREATE TABLE HAVE(
a_id int,
```

g_id int,

foreign key (a_id) references artist(a_id),

foreign key (g_id) references gallery(g_id)

);

create table **IS_A(**spec_id int primary key, spec_name varchar(225));

create table bought_by(cr_id int,

u_id int,

foreign key (cr_id) references creates(cr_id),

foreign key (u_id) references user(u_id)

);

create table **VISITS**(e_id int,

u_id int,

foreign key (e_id) references exhibition(e_id),

foreign key (u_id) references user(u_id)

);

create table **PAYMENT**(py_id int primary key,

py_mode varchar(225),

cr_id int,

u_id int,

foreign key (e_id) references exhibition(e_id),

foreign key (u_id) references user(u_id),

amount float);

```
mysql> desc artist;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| a_id     | int(11)      | NO   | PRI | NULL    |       |
| a_name   | varchar(225) | YES  |     | NULL    |       |
| address  | varchar(225) | YES  |     | NULL    |       |
| phone_no | varchar(225) | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
4 rows in set (0.18 sec)

mysql> desc bought_by;
+-------+---------+------+-----+---------+-------+
| Field | Type    | Null | Key | Default | Extra |
+-------+---------+------+-----+---------+-------+
| cr_id | int(11) | YES  | MUL | NULL    |       |
| u_id  | int(11) | YES  | MUL | NULL    |       |
+-------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> desc exhibition;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| e_id       | int(11)      | NO   | PRI | NULL    |       |
| e_name     | varchar(225) | YES  |     | NULL    |       |
| location   | varchar(225) | YES  |     | NULL    |       |
| e_type     | varchar(225) | YES  |     | NULL    |       |
| start_date | datetime     | YES  |     | NULL    |       |
| end_date   | datetime     | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> desc creates;
+-------+---------+------+-----+---------+-------+
| Field | Type    | Null | Key | Default | Extra |
+-------+---------+------+-----+---------+-------+
| cr_id | int(11) | NO   | PRI | NULL    |       |
| a_id  | int(11) | YES  | MUL | NULL    |       |
+-------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

```
mysql> desc gallery;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| g_id     | int(11)      | NO   | PRI | NULL    |       |
| g_name   | varchar(225) | YES  |     | NULL    |       |
| g_url    | varchar(225) | YES  |     | NULL    |       |
| location | varchar(225) | YES  |     | NULL    |       |
| e_id     | int(11)      | YES  | MUL | NULL    |       |
+----------+--------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> desc handicraft;
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| h_id   | int(11)      | NO   | PRI | NULL    |       |
| h_name | varchar(225) | YES  |     | NULL    |       |
| cr_id  | int(11)      | YES  | MUL | NULL    |       |
| a_id   | int(11)      | YES  | MUL | NULL    |       |
| h_type | varchar(25)  | YES  |     | NULL    |       |
| price  | float        | YES  |     | NULL    |       |
+--------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> desc painting;
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| p_id   | int(11)      | NO   | PRI | NULL    |       |
| p_name | varchar(225) | YES  |     | NULL    |       |
| cr_id  | int(11)      | YES  | MUL | NULL    |       |
| a_id   | int(11)      | YES  | MUL | NULL    |       |
| p_type | varchar(25)  | YES  |     | NULL    |       |
| price  | float        | YES  |     | NULL    |       |
+--------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

```
mysql> desc sculpture;
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| s_id   | int(11)      | NO   | PRI | NULL    |       |
| s_name | varchar(225) | YES  |     | NULL    |       |
| cr_id  | int(11)      | YES  | MUL | NULL    |       |
| a_id   | int(11)      | YES  | MUL | NULL    |       |
| s_type | varchar(25)  | YES  |     | NULL    |       |
| price  | float        | YES  |     | NULL    |       |
+--------+--------------+------+-----+---------+-------+
6 rows in set (0.03 sec)

mysql> desc have;
+-------+---------+------+-----+---------+-------+
| Field | Type    | Null | Key | Default | Extra |
+-------+---------+------+-----+---------+-------+
| a_id  | int(11) | YES  | MUL | NULL    |       |
| g_id  | int(11) | YES  | MUL | NULL    |       |
+-------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> desc organize;
+-------+---------+------+-----+---------+-------+
| Field | Type    | Null | Key | Default | Extra |
+-------+---------+------+-----+---------+-------+
| e_id  | int(11) | NO   |     | NULL    |       |
| g_id  | int(11) | YES  | MUL | NULL    |       |
+-------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> desc logs_in;
+-------+---------+------+-----+---------+-------+
| Field | Type    | Null | Key | Default | Extra |
+-------+---------+------+-----+---------+-------+
| l_id  | int(11) | YES  | MUL | NULL    |       |
| u_id  | int(11) | YES  | MUL | NULL    |       |
+-------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql>
```

```
mysql> desc login;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| l_id     | int(11)      | NO   | PRI | NULL    |       |
| u_id     | int(11)      | YES  | MUL | NULL    |       |
| u_type   | varchar(225) | YES  |     | NULL    |       |
| email    | varchar(225) | YES  |     | NULL    |       |
| password | varchar(60)  | YES  |     | NULL    |       |
| phone_no | int(11)      | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> desc participate;
+-------+---------+------+-----+---------+-------+
| Field | Type    | Null | Key | Default | Extra |
+-------+---------+------+-----+---------+-------+
| e_id  | int(11) | YES  | MUL | NULL    |       |
| p_id  | int(11) | NO   | PRI | NULL    |       |
+-------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> desc pays;
+-------+---------+------+-----+---------+-------+
| Field | Type    | Null | Key | Default | Extra |
+-------+---------+------+-----+---------+-------+
| py_id | int(11) | YES  | MUL | NULL    |       |
| u_id  | int(11) | YES  | MUL | NULL    |       |
+-------+---------+------+-----+---------+-------+
2 rows in set (0.02 sec)

mysql> desc payment;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| py_id   | int(11)     | NO   | PRI | NULL    |       |
| py_mode | varchar(25) | YES  |     | NULL    |       |
| amount  | int(11)     | YES  |     | NULL    |       |
| cr_id   | int(11)     | YES  | MUL | NULL    |       |
| u_id    | int(11)     | YES  | MUL | NULL    |       |
+---------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

```
mysql> desc user;
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| u_id   | int(11)      | NO   | PRI | NULL    |       |
| u_name | varchar(225) | YES  |     | NULL    |       |
| u_type | varchar(225) | YES  |     | NULL    |       |
| cr_id  | int(11)      | YES  | MUL | NULL    |       |
| e_id   | int(11)      | YES  | MUL | NULL    |       |
+--------+--------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> desc visits;
+-------+---------+------+-----+---------+-------+
| Field | Type    | Null | Key | Default | Extra |
+-------+---------+------+-----+---------+-------+
| e_id  | int(11) | YES  | MUL | NULL    |       |
| u_id  | int(11) | YES  | MUL | NULL    |       |
+-------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

QUERIES:

FIND THE ARTISTS WHO HAVE A SCULPTURE WHOSE PRICE IS MORE THAN 20000.

select a_name from artist where a_id in(select a_id from sculpture where price >20000) group by(a_name);

```
mysql> select a_name from artist where a_id in(select a_id from sculpture where price >20000) group by(a_name);
+-----------------------+
| a_name                |
+-----------------------+
| CHARLES WHITE         |
| SEBASTIO SALGADO      |
| CAO FEI               |
| LILY VAN DER STROKKER |
| WILHELM KUHNERT       |
| BRUCE NAUMANN         |
| TOMMA ABTS            |
| LEONARDO DA VINCI     |
| VAN GOGH              |
| MARIA LOBODA          |
+-----------------------+
10 rows in set (0.00 sec)
```

FIND THE NAMES OF HANDICRAFTS BY A PARTICULAR ARTIST.

select h_name from handicraft where a_id in(select a_id from artist where a_name='cao fei ');

```
mysql> select h_name from handicraft where a_id in(select a_id from artist where a_name='masa lalu ');
Empty set (0.00 sec)

mysql> select h_name from handicraft where a_id in(select a_id from artist where a_name='MASU LALU');
Empty set (0.00 sec)

mysql> select h_name from handicraft where a_id in(select a_id from artist where a_name='cao fei');
+-----------+
| h_name    |
+-----------+
| BALLERINA |
+-----------+
1 row in set (0.00 sec)

mysql> select h_name from handicraft where a_id in(select a_id from artist where a_name='leonardo da vinci');
Empty set (0.00 sec)

mysql> select h_name from handicraft where a_id in(select a_id from artist where a_name='tomma abts');
+----------+
| h_name   |
+----------+
| JUKE BOX |
+----------+
1 row in set (0.00 sec)
```

DISPLAY THE NAMES OF ARTIST AND THE EXHIBITIONS THEY HAVE PARTICIPATED IN.

select a_name from artist where a_id in(select a_id from participate) group by a_name

```
mysql> select a_name from artist where a_id in(select a_id from participate ) group by a_name;
+-----------------------+
| a_name                |
+-----------------------+
| ANDY WARHOL           |
| MARIA LOBODA          |
| CHARLES WHITE         |
| SEBASTIO SALGADO      |
| MASA LALU             |
| MARTINE FRANCK        |
| CAO FEI               |
| QIU ZHIJIE            |
| LILY VAN DER STROKKER |
| LEONARDO DA VINCI     |
| WILHELM KUHNERT       |
| VAN GOGH              |
| DOROTHEA TANNING      |
| BRUCE NAUMANN         |
| TOMMA ABTS            |
+-----------------------+
15 rows in set (0.03 sec)
```

```
mysql> select a_name,e_name from artist a,exhibition e,participate p where a.a_id=p.a_id and p.e_id=e.e_id  ;
+-----------------------+---------------------------------------+
| a_name                | e_name                                |
+-----------------------+---------------------------------------+
| ANDY WARHOL           | Whitney Museum of American Art        |
| MARIA LOBODA          | SCHIRN Kunsthalle                     |
| CHARLES WHITE         | Bildmuseet, Umeå                      |
| SEBASTIO SALGADO      | Museum für Gestaltung Zürich          |
| MASA LALU             | Museum of Modern and Contemporary Art |
| MARTINE FRANCK        | Fondation Henri Cartier-Bresson       |
| CAO FEI               | Kunsthistorisches Museum Wien         |
| QIU ZHIJIE            | SCHIRN Kunsthalle                     |
| LILY VAN DER STROKKER | WILDERNESS                            |
| LEONARDO DA VINCI     | Ullens Center for Contemporary Art    |
| WILHELM KUHNERT       | Stedelijk                             |
| VAN GOGH              | KING OF THE ANIMALS                   |
| DOROTHEA TANNING      | Galleria Borghese                     |
| BRUCE NAUMANN         | Musée de la Civilisation              |
| TOMMA ABTS            | The Museum of Modern Art, MoMA PS1    |
| SEBASTIO SALGADO      | The Art Institute of Chicago          |
| MARTINE FRANCK        | Guggenheim Bilbao                     |
| QIU ZHIJIE            | Palais de Tokyo                       |
| LEONARDO DA VINCI     | Centre Pompidou                       |
| WILHELM KUHNERT       | Philadelphia Museum of Art            |
+-----------------------+---------------------------------------+
20 rows in set (0.02 sec)
```

DISPLAY DETAILS OF ARTISTS PARTICIPATING IN EXHIBITIONS HELD IN NEW YORK.

*select * from artist where a_id in (select a_id from participate where e_id from exhibition where location="New York"*

```
mysql> select * from artist where a_id in(select a_id from participate where e_id in(select e_id from exhibition where location='new york'));
+------+-------------+---------+----------+
| a_id | a_name      | address | phone_no |
+------+-------------+---------+----------+
|  301 | ANDY WARHOL | ADR1    | PHN01    |
|  315 | TOMMA ABTS  | ADR15   | PHN015   |
+------+-------------+---------+----------+
2 rows in set (0.00 sec)
```

DISPLAY THE NAME OF USERS WHO HAVE LOGGED IN THE DATABASE AND ARE ARTISTS.

Select u_name, u_type from users u, login l, logs_in ln were u.u_id = l.u_id and l.l_id = ln.l_id and u_type = 'artist'

```
mysql> select u_name,u_type from users u,login l,logs_in ln where u.u_id=l.u_id and l.l_id=ln.l_id and u_type='artist';
+---------+--------+
| u_name  | u_type |
+---------+--------+
| UNAME2  | artist |
| UNAME4  | artist |
| UNAME6  | artist |
| UNAME8  | artist |
| UNAME10 | artist |
| UNAME12 | artist |
| UNAME14 | artist |
| UNAME16 | artist |
| UNAME18 | artist |
| UNAME20 | artist |
+---------+--------+
10 rows in set (0.11 sec)
```

DETAILS OF DIFFERENT TYPES OF USERS.

Select u_name, u_type from users u, login l, logs_in ln where u.u_id = l.u_id and l.l_id = ln.l_id group by(u_type)

```
mysql> select u_name,u_type from users u,login l,logs_in ln where u.u_id=l.u_id and l.l_id=ln.l_id group by(u_type);
+--------+--------+
| u_name | u_type |
+--------+--------+
| UNAME1 | client |
| UNAME2 | artist |
+--------+--------+
2 rows in set (0.00 sec)
```

DISPLAY VISITORS WHO HAVE VISITED EXHIBITIONS IN AMSTERDAM.

Select u_name, location, e_name from users u, exhibition e, visits v where u.u_id = v.u_id and e.e_id = v.e_id and location = 'amsterdam'

```
mysql> select u_name,location,e_name from users u,exhibition e,visits v where u.u_id=v.u_id and e.e_id=v.e_id and location='amsterdam';
+---------+-----------+-----------------------------------+
| u_name  | location  | e_name                            |
+---------+-----------+-----------------------------------+
| UNAME10 | AMSTERDAM | Ullens Center for Contemporary Art |
| UNAME11 | AMSTERDAM | Ullens Center for Contemporary Art |
| UNAME12 | AMSTERDAM | Stedelijk                         |
+---------+-----------+-----------------------------------+
3 rows in set (0.00 sec)
```

FUNCTION TO MULTIPLY:

```
mysql> delimiter $$
mysql> create function multiply2(n1 int,n2 int)
    -> returns int
    -> begin
    -> declare result int;
    -> set result=n1*n2;
    -> return result;
    -> end $$
Query OK, 0 rows affected (0.58 sec)

mysql> select multiply2(4,6) \g
+----------------+
| multiply2(4,6) |
+----------------+
|             24 |
+----------------+
1 row in set (0.01 sec)
```

UPDATE TRIGGER ON TABLE ARTIST:

```
mysql> select * from trig;
+-----+------+-------------------+--------+---------+----------+---------------------+
| tid | a_id | a_name            | action | address | phone_no | changed_on          |
+-----+------+-------------------+--------+---------+----------+---------------------+
|   1 |  310 | LEONARDO DA VINCI | update | ADR10   | PHNO10   | 2018-11-13 23:30:25 |
+-----+------+-------------------+--------+---------+----------+---------------------+
1 row in set (0.00 sec)
```

PROCEDURE TO DISPLAY NAMES OF USERS WHO HAVE MADE A PAYMENT OF MORE THAN 30000:

```
mysql> delimiter $$
mysql> create procedure xyz()
    -> begin
    -> select u_name from users where u_id in(select u_id from payment where amount>30000);
    -> end $$
ERROR 1304 (42000): PROCEDURE xyz already exists
mysql> delimiter $$
mysql> create procedure abc()
    -> begin
    -> select u_name from users where u_id in(select u_id from payment where amount>30000);
    -> end $$
Query OK, 0 rows affected (0.17 sec)

mysql> call abc() \g
+---------+
| u_name  |
+---------+
| UNAME3  |
| UNAME4  |
| UNAME6  |
| UNAME7  |
| UNAME10 |
| UNAME16 |
| UNAME17 |
+---------+
7 rows in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

VIEWS:

View for handicraft name and price that's over 15,000:

create view hand_name_price as select h_name, price from handicraft where price>15000;

```
mysql> select * from sculpture_name_price;
+------------------+--------+
| s_name           | price  |
+------------------+--------+
| LOREN MADSEN     | 120000 |
| MOUNT FEAR       | 132000 |
| WORRY            | 100000 |
| NATHALIE MEIBACH | 150000 |
| EXPLOSION        | 130000 |
| ANDREAS FISSCHER | 140000 |
| BRIAN SIEMEN     | 123000 |
| LOOK             | 167000 |
| SMILES           | 154000 |
| AWAY             | 120000 |
| HENREICH         | 130000 |
| CINDERELLA       | 150000 |
| CHAOS            | 155000 |
| SEE              | 165000 |
| SHINE            | 172000 |
| SOUND OF THE POEM | 163000 |
| MOUNT DREAD      | 128000 |
| SHIMMER          | 139000 |
| GLIMMER          | 124000 |
| NINE             | 136000 |
+------------------+--------+
20 rows in set (0.00 sec)
```

View for painting name and price that's over 15,000:

create view painting_name_price as select p_name, price from painting where price>15000;

```
mysql> select * from painting_name_price;
+-----------------------------------+--------+
| p_name                            | price  |
+-----------------------------------+--------+
| Self-Portrait with a Glass of Wine |  16500 |
| Deer beside a Lake                | 155000 |
| The Archangel Michael             |  19900 |
| Albarello                         |  19900 |
| Ceramic Floor                     | 173000 |
| The Flood                         |  40000 |
| Chimney breast                    |  16500 |
| Chimney breast                    |  24500 |
| Stag Hunt                         |  35000 |
| Orpheus and Eurydice              |  24600 |
| The Rape of Proserpine            |  15620 |
| Landscape at Castiglioncello      |  15700 |
+-----------------------------------+--------+
12 rows in set (0.00 sec)
```

View for sculpture name and price that's over 15,000:

create view sculpture_name_price as select s_name, price from sculpture where price>15000

```
mysql> select * from hand_name_price;
+-----------------------------------+-------+
| h_name                            | price |
+-----------------------------------+-------+
| WOODEN PHOTO FRAME HORN/BONE DECOR | 34000 |
| BRASS HANDICRAFT BOX              | 35000 |
| TREE OF LIFE                      | 23000 |
| BUDDHA                            | 87000 |
| SERENDIPITY                       | 17000 |
| MAGNIFICACO                       | 16000 |
| UNDER THE UMBRELLA                | 19000 |
| LA VIE EN ROSE                    | 18000 |
| JUKE BOX                          | 15500 |
| TREASURE OF LIFE                  | 16500 |
| SATIN DESIGNS                     | 22000 |
| HEARTITUDE                        | 23000 |
| INDIA KA KHAZANA                  | 23500 |
| THE WARLI                         | 30000 |
| HIM                               | 27000 |
+-----------------------------------+-------+
15 rows in set (0.00 sec)
```