LIBRARY MANAGEMENT PROJECT

BY
A.ANIL KRISHNA
D.GANESH PATNAIK
U.BHAVANA
V.KALYAN KUMAR

TABLE OF CONTENTS

INTRODUCTION

A library is a collection of organized information and resources which is made accessible to a well-defined community for borrowing or reference sake. The collection of the resources and information are provided in digital or physical format in either a building/room or in a virtual space or even both. Library's resources and collections may include newspapers, books, films, prints, maps, CDs, tapes, videotapes, microform, database etc. The main aim of this system is to develop a new programmed system that will conveying ever lasting solution to the manual base operations and to make available a channel through which staff can maintain the record easily and customers can access the information about the library at whatever place they might find themselves.

Library Management System allows the user to store the book details and the customer details. The system is strong enough to withstand regressive yearly operations under conditions where the database is maintained and cleared over a

certain time of span. The implementation of the system in the organization will considerably reduce data entry, time and also provide readily calculated reports.

OBJECTIVE: - It keeps track of all the information about the books in the library, their cost, status and total number of books available in the Library. The user will find it easy in this automated system rather than using the manual writing system. The system contains a database where all the information will be stored safely.

✓ *2. Data types and its description*

1. Integer: one optional sign character (+ or -) followed by at least one digit
(0-9). Leading and trailing blanks are ignored. No other character is
allowed.

2. Varchar: It is used to store alpha numeric characters. In this data type
we can set the maximum number of characters up to 8000 ranges by
defaults SQL server will set the size to 50 characters range.

3. Date: The DATE data type accepts date values. No parameters are
required when declaring a DATE data type. Date values should be
specified in the form: YYYY-MM-DD. However, Point Base will also
accept single digits entries for month and day values.

4. Time: The TIME data type accepts time values. No parameters are
required when declaring a TIME data type. Date values should be
specified in the form: HH:MM: SS. An optional fractional value can be
used to represent nanoseconds.

3. Data Requirements

Entities

- BRANCH
- EMPLOYEE
- CUSTOMER
- ISSUE STATUS
- RETURN STATUS
- BOOKS

Attributes

- BRANCH
- Manager_id
- Branch_id
- Address
- Contact_no
- Branch_h_no
- Street
- City
- State
- Zipcode
- CUSTOMER
- Customer_id
- Books_issued
- Name
- Address

- **ISSUE STATUS**
- **Issue_book_name**
- **Issue_id**
- **Issue_date**
- **ISBN**
- **Customer_id**
- **RETURN STATUS**
- **Return_id**
- **Return_date**
- **Customer_id**
- **Return_book_name**
- **ISBN**
- **BOOKS**
- **ISBN**
- **Title**
- **Category**
- **Rental_price**
- **Author**
- **Publisher**
- **Status**

RELATIONSHIPS - CARDINALITY

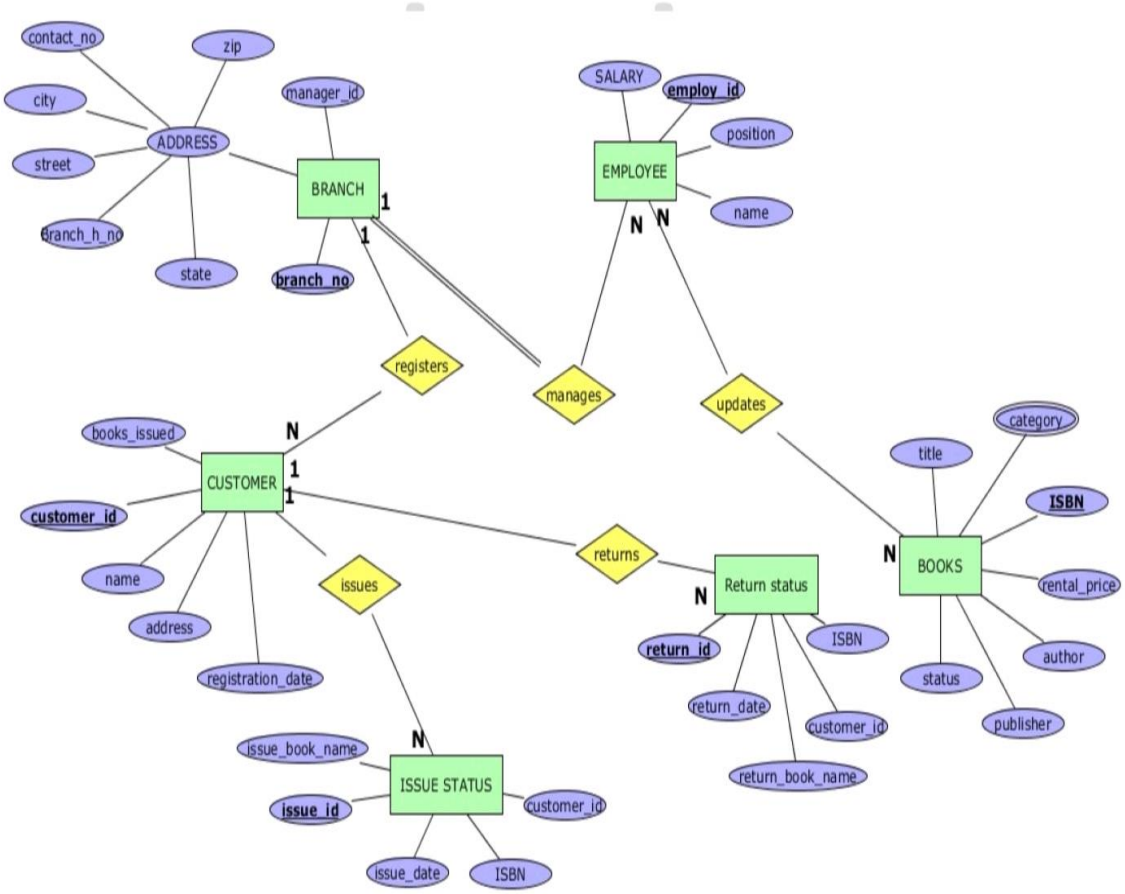
- **MANAGER manages the BRANCH (1 - N)**

- **CUSTOMER registers in the respective BRANCH (N – 1)**
- **CUSTOMER issues BOOKS (1 – N)**
- **CUSTOMER returns BOOKS (N – 1)**
- **EMPLOYEE updates BOOKS (N – N)**

4. Entity-Relationship-DIAGRAM

Entity Relationship Diagram is used in modern database software engineering to illustrate logical structure of database. It is a relational schema database modelling method used to Model a system and approach. This approach commonly used in database design. The diagram created using this method is called ER-diagram.

The ER-diagram depicts the various relationships among entities, considering each object as entity. Entity is represented as rectangle shape and relationship represented as diamond shape. It depicts the relationship between data object.



5. SCHEMA DIAGRAM

A schema is the structure behind data organization. It is a visual representation of how different table relationships enable the schema's underlying mission business rules for which the database is created. Database schema defines its entities and the relationship among them.

It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful.

Schema diagrams have an important function because they force database developers to transpose ideas to paper. This provides an overview of the entire database, while facilitating future database administrator work.

6. CREATING DATABASE USING MYSQL

```
mysql> create database libproject;
mysql> use libproject;
Database changed

mysql> CREATE TABLE BOOKS(
        ISBN int(100) not null,
        book_title varchar(50) not null,
        category varchar(50) not null,
        rental_price int(10) not null,
        status varchar(50),
        author varchar(50) not null,
        publisher varchar(50) not null,
        primary key(ISBN)) ;
Query OK, 0 rows affected (1.44 sec)
mysql> CREATE TABLE EMPLOYEE(
            employ_id int(10) not null,
            branchid int not null,
            employ_name varchar(50) not null,
            position varchar(30) not null,
            salary int(10) not null,
             primary key(employ_id),
        constraint  f1 foreign key (branchid) references
branch(branch_no));
Query OK, 0 rows affected (0.35 sec)
mysql> create table customer(
        customer_id int(10) not null,
        customer_name varchar(50),
        customer_address varchar(100) not null,
```

```
        registration_date date not null,
        rent int ,
        primary key(customer_id));
Query OK, 0 rows affected (0.38 sec)
mysql> create table branch(
        branch_no int(10) not null,
manager_id int(10) not null,
branch_address varchar(100) not null,
contact_no int(10) not null,
primary key(branch_no));
Query OK, 0 rows affected (0.49 sec)
mysql> create table issue_status(
        issue_id int(10) not null,
        issued_cust int(10) not null,
        employ_id int not null,
        issued_book_name varchar(50) not null,
        issue_date date not null,
        isbn_book int(10) not null,
        primary key(issue_id),
        constraint f3 foreign key(employ_id) references
EMPLOYEE(employ_id),
        constraint foreign key(isbn_book) references
BOOKS(ISBN),constraint foreign key(issued_cust) references
customer(customer_id));
Query OK, 0 rows affected (0.66 sec)
mysql> create table return_status(
        return_id int(10) not null,
        return_cust int(10) not null,
        employ_id int not null,
```

```
        returned_book_name varchar(50) not null,
        return_date date not null,
        isbn_book2 int(10) not null,
        primary key(return_id),
        constraint  f4 foreign key(employ_id) references
EMPLOYEE(employ_id),
        constraint foreign key(isbn_book2) references
        BOOKS(ISBN),
        constraint foreign key(return_cust)
        references issue_status(issued_cust));
Query OK, 0 rows affected (0.39 sec)



mysql> show tables;
+---------------------+
| Tables_in_libproject |
+---------------------+
| books               |
| branch              |
| customer            |
| employee            |
| issue_status        |
| return_status       |
+---------------------+
```

6 rows in set (0.12 sec)

desc books;

```
+-------------+------------+------+-----+---------+-------+
| Field       | Type       | Null | Key | Default | Extra |
+-------------+------------+------+-----+---------+-------+
| ISBN        | int(100)   | NO   | PRI | NULL    |       |
| book_title  | varchar(50) | NO   |     | NULL    |       |
| category    | varchar(50) | NO   |     | NULL    |       |
| rental_price | int(10)    | NO   |     | NULL    |       |
| status      | varchar(50) | YES  |     | NULL    |       |
| author      | varchar(50) | NO   |     | NULL    |       |
| publisher   | varchar(50) | NO   |     | NULL    |       |
+-------------+------------+------+-----+---------+-------+
```

7 rows in set (0.06 sec)

mysql> desc branch;

```
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
```

```
| branch_no      | int(10)      | NO   | PRI | NULL    |       |
| manager_id     | int(10)      | NO   |     | NULL    |       |
| branch_address | varchar(100) | NO   |     | NULL    |       |
| contact_no     | int(10)      | NO   |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> desc customer;
+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| customer_id      | int(10)      | NO   | PRI | NULL    |       |
| customer_name    | varchar(50)  | YES  |     | NULL    |       |
| customer_address | varchar(100) | NO   |     | NULL    |       |
| registration_date | date        | NO   |     | NULL    |       |
| rent             | int(11)      | YES  |     | NULL    |       |
+------------------+--------------+------+-----+---------+-------+
5 rows in set (0.05 sec)

mysql> desc employee;
```

```
+------------+------------+------+-----+---------+-------+
| Field      | Type       | Null | Key | Default | Extra |
+------------+------------+------+-----+---------+-------+
| employ_id  | int(10)    | NO   | PRI | NULL    |       |
| branchid   | int(11)    | NO   | MUL | NULL    |       |
| employ_name | varchar(50) | NO  |     | NULL    |       |
| position   | varchar(30) | NO   |     | NULL    |       |
| salary     | int(10)    | NO   |     | NULL    |       |
+------------+------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> desc issue_status;
+-----------------+------------+------+-----+---------+-------+
| Field           | Type       | Null | Key | Default | Extra |
+-----------------+------------+------+-----+---------+-------+
| issue_id        | int(10)    | NO   | PRI | NULL    |       |
| issued_cust     | int(10)    | NO   | MUL | NULL    |       |
| employ_id       | int(11)    | NO   | MUL | NULL    |       |
| issued_book_name | varchar(50) | NO  |     | NULL    |       |
| issue_date      | date       | NO   |     | NULL    |       |
```

| isbn_book | int(10) | NO | MUL | NULL | |

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

6 rows in set (0.00 sec)

mysql> desc return_status;

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

| Field | Type | Null | Key | Default | Extra |

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

| return_id | int(10) | NO | PRI | NULL | |

| return_cust | int(10) | NO | MUL | NULL | |

| employ_id | int(11) | NO | MUL | NULL | |

| returned_book_name | varchar(50) | NO | | NULL |
|

| return_date | date | NO | | NULL | |

| isbn_book2 | int(10) | NO | MUL | NULL | |

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

6 rows in set (0.00 s)

insert into books values (123,"DBMS","CSE",5,"available","Mc Graw Hill","Tata");

Query OK, 1 row affected (0.23 sec)

mysql> insert into books values
(124,"DBMS","CSE",5,"available","Stanley","Tata");

Query OK, 1 row affected (0.11 sec)

mysql> insert into books values
(125,"CG","CSE",5,"available","Stanley","Tata");

Query OK, 1 row affected (0.17 sec)

mysql> insert into books values
(126,"DS","CSE",5,"available","Stanley","Tata");

Query OK, 1 row affected (0.09 sec)

mysql> insert into books values
(127,"DBMS","CSE",5,"available","William","Tata");

Query OK, 1 row affected (0.07 sec)

mysql> insert into books values
(128,"DS","CSE",6,"available","William","Tata");

Query OK, 1 row affected (0.21 sec)

mysql> insert into books values
(129,"CG","CSE",6,"available","William","Tata");

Query OK, 1 row affected (0.13 sec)

mysql> select * from books;

```
+------+-----------+----------+-------------+-----------+-------------+-----------+
| ISBN | book_title | category | rental_price | status    | author      | publisher |
+------+-----------+----------+-------------+-----------+-------------+-----------+
|  123 | DBMS      | CSE      |           5 | available | Mc Graw Hill | Tata      |
|  124 | DBMS      | CSE      |           5 | available | Stanley     | Tata      |
|  125 | CG        | CSE      |           5 | available | Stanley     | Tata      |
|  126 | DS        | CSE      |           5 | available | Stanley     | Tata      |
|  127 | DBMS      | CSE      |           5 | available | William     | Tata      |
|  128 | DS        | CSE      |           6 | available | William     | Tata      |
|  129 | CG        | CSE      |           6 | available | William     | Tata      |
```

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

7 rows in set (0.00 sec)

insert into branch values(1,25,"b-block",8498905320);

ERROR 1264 (22003): Out of range value for column 'contact_no' at row 1

mysql> insert into branch values(1,25,"b-block",849890532);

Query OK, 1 row affected (0.21 sec)

mysql> insert into branch values(3,20,"c-block",849890534);

Query OK, 1 row affected (0.14 sec)

mysql> insert into branch values(2,21,"c-block",849890535);

Query OK, 1 row affected (0.20 sec)

mysql> insert into branch values(5,4"b-block",849840535);

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '"b-block",849840535)' at line 1

mysql> insert into branch values(5,4,"b-block",849840535);

Query OK, 1 row affected (0.12 sec)

```
mysql> select * from branch;
+-----------+------------+---------------+------------+
| branch_no | manager_id | branch_address | contact_no |
+-----------+------------+---------------+------------+
|         1 |         25 | b-block        |  849890532 |
|         2 |         21 | c-block        |  849890535 |
|         3 |         20 | c-block        |  849890534 |
|         5 |          4 | b-block        |  849840535 |
+-----------+------------+---------------+------------+
4 rows in set (0.00 sec)

insert into employee values(104,1,"Ravi","clerk",2000);
Query OK, 1 row affected (0.22 sec)

mysql> insert into employee
values(105,1,"Ram","clerk",3000);
Query OK, 1 row affected (0.16 sec)

mysql> insert into employee
values(106,2,"Rahim","clerk",3000);
```

```
Query OK, 1 row affected (0.13 sec)

mysql> insert into employee
values(108,2,"Anil","librarian",30000);
Query OK, 1 row affected (0.09 sec)

mysql> insert into employee
values(107,5,"sunil","clerk",4000);
Query OK, 1 row affected (0.17 sec)

mysql> select * from employee;
+-----------+----------+-------------+-----------+--------+
| employ_id | branchid | employ_name | position  | salary |
+-----------+----------+-------------+-----------+--------+
|       104 |        1 | Ravi        | clerk     |   2000 |
|       105 |        1 | Ram         | clerk     |   3000 |
|       106 |        2 | Rahim       | clerk     |   3000 |
|       107 |        5 | sunil       | clerk     |   4000 |
|       108 |        2 | Anil        | librarian |  30000 |
+-----------+----------+-------------+-----------+--------+
5 rows in set (0.00 sec)
```

insert into customer values (1,"Anil","Ongole","2008-11-09",0);

Query OK, 1 row affected (0.23 sec)

mysql> insert into customer values (2,"Kalyan","Vizag","2008-11-09",0);

Query OK, 1 row affected (0.17 sec)

mysql> insert into customer values (3,"Ravi","Vizag","2009-11-09",0);

Query OK, 1 row affected (0.22 sec)

mysql> insert into customer values (4,"Ganesh","Vizag","2009-11-09",0);

Query OK, 1 row affected (0.14 sec)

mysql> select * from customers;

ERROR 1146 (42S02): Table 'libr.customers' doesn't exist

mysql> select * from customer;

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

| customer_id | customer_name | customer_address | registration_date | rent |

```
+------------+--------------+-----------------+------------------+------+
|          1 | Anil         | Ongole          | 2008-11-09       |    0 |
|          2 | Kalyan       | Vizag           | 2008-11-09       |    0 |
|          3 | Ravi         | Vizag           | 2009-11-09       |    0 |
|          4 | Ganesh       | Vizag           | 2009-11-09       |    0 |
+------------+--------------+-----------------+------------------+------+
4 rows in set (0.00 sec)
```

Triggers for issue_status

```
create trigger checkissue before insert on issue_status
   -> for each row
   -> begin
   -> if((select status from books where
ISBN=new.isbn_book)="unavailable")
   -> then
   -> signal sqlstate '45000'
   -> set message_text="Sorry The book is not available";
   -> end if;
   -> end;
   -> //
```

Query OK, 0 rows affected (0.28 sec)

create trigger maxissue before insert on issue_status

-> for each row

-> begin

-> if((select count(isbn_book) as c from issue_status i ,books b where isbn_book=ISBN and status="unavailable" group by issued_cust having issued_cust=new.issued_cust)>3)

-> then

-> signal sqlstate '46000'

-> set message_text="Max books have been already alloted to the customer";

-> end if;

-> end;

-> //

Query OK, 0 rows affected (0.54 sec)

create trigger issue before insert on issue_status

-> for each row

-> begin

-> update books set status="unavailable" where ISBN=new.isbn_book;

-> end;

-> //

Query OK, 0 rows affected (0.10 sec)

insert into issue_status values (1040,2,104,"DBMS","2018-04-10",124);

-> //

Query OK, 1 row affected (0.17 sec)

Triggers on return_status

| STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION |
root@localhost | utf8mb4 | utf8mb4_0900_ai_ci |
utf8mb4_0900_ai_ci |

| ret | INSERT | return_status | begin

update books set status="available" where ISBN=new.isbn_book2;

end |
BEFORE | 2018-10-20 13:23:09.71 |
STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION |
root@localhost | utf8mb4 | utf8mb4_0900_ai_ci |
utf8mb4_0900_ai_ci |

Procedure for fine or rent

create procedure fine(

-> INOUT RE DATE,INOUT ISS DATE,INOUT REN INT,INOUT CUST INT)

-> BEGIN

-> SET REN=DATEDIFF(RE,ISS)*REN;

-> UPDATE customer set rent=rent+REN where customer_id=CUST;

-> end//

Query OK, 0 rows affected (0.21 sec)

7. TEST QUERIES

1. FIND THE EMPLOYEE WHO ISSUES LARGE NUMBER OF BOOKS

A. Create view maxemp as(select employ_id,count(issue_id) as c from issue_status group by employ_id);
select employ_id,employ_name from employee where employ_id in(select employ_id from maxemp where c=(select max(c) as m from maxemp));

CONCLUSION

- **SQL database management application which is very well used in the modern world in organising and manipulating a database.**

- **Though SQL doesn't have the GUI interface like Microsoft access is having and they all manage the database comfortable.**

- **Depending on the user or users, if an organisation has multiple users then they should go for SQL server based application.**

- **This project shows how to create tables in SQL and how to create simple data manipulation language and data definition language with how to execute them.**

- **It also shows how relationships are established with the concepts of primary and foreign key within a table.**

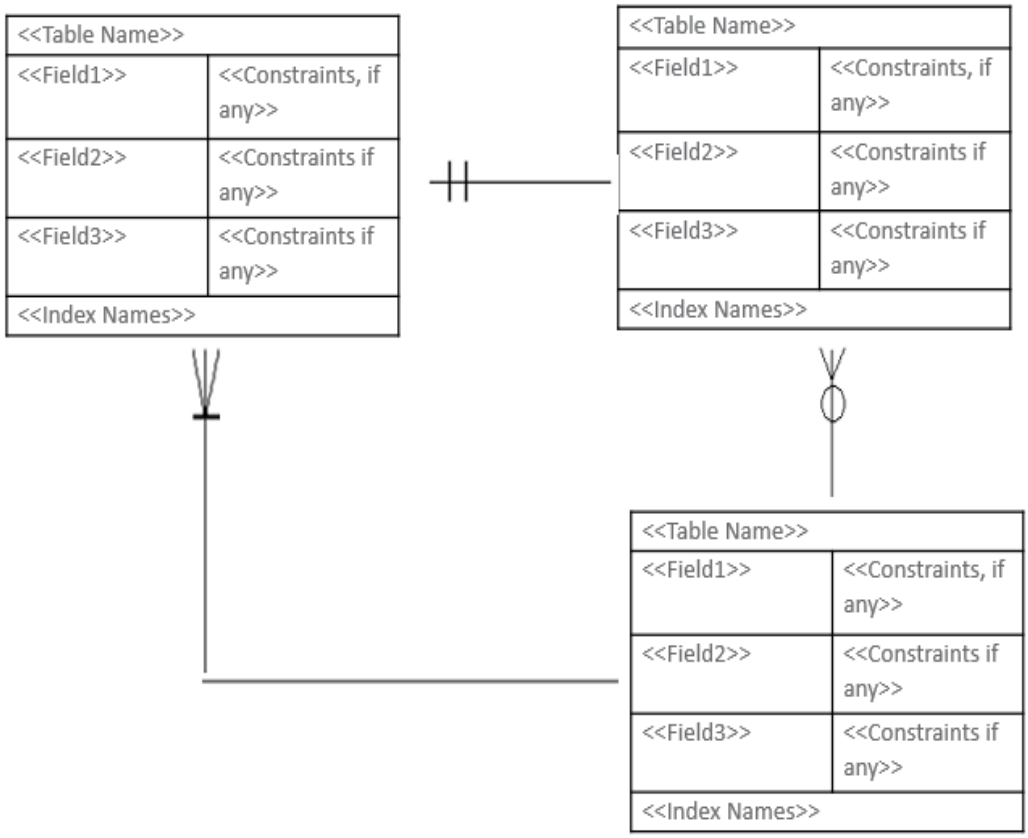
Lastly, the project shows how queries are created in SQL server, queries like the create command, view, update, alter etc.

INSTA DB PROJECT

- **The project is to design a relational database that allows you to store and manage pictures from multiple users of the system in line with the guidelines given below: :**
    - **The database will store information about users of the system, primarily their names and email addresses, and pictures they post.**
    - **Database stores captions of pictures, picture path, their date of posting, who posted them. A picture can be posted by only one user.**
    - **Pictures can be tagged for – Art, Science, Music, History and Engineering. Multiple tags are allowed for the same picture.**
    - **Like functionality is also supported i.e. a user can like multiple pictures.**
    - **Use unique ids for primary keys wherever applicable, e.g. picture id, user id etc.**
    - **Normalize the tables to sufficient depth so that anomalies and data duplication can be avoided to a larger extent**

- **Go through the queries to know more about database design. E.g. query"Who (user id) has liked the most pictures?"tells that database must store user ids of users liking the pictures.**

- **What needs to be done:**
  - **Identify tables, respective fields and relationships among tables**
  - **Create tables**
  - **Insert data into tables**
  - **Write SQL to address the following queries**
- **Development Approach:**
  - **After identifying tables, respective fields and relationship among tables, learners should create database design diagram in the following format and discuss with the faculty.**
  - **Incorporate the feedback from faculty on correctness of database design and proceed with creation of tables and data insertion.**
  - **Identify the appropriate operations and express them in SQL.**




```
create table instauser(
id int primary key,
email varchar(50) unique not null,
name varchar(50) not null
);
create table pictures(
pid int primary key,
path varchar(50) unique not null,
caption varchar(50) default 'NO CAPTION',
id int not null,
postdate date not null,
constraint f1 foreign key (id) references instauser(id)
);
```

```
create table tags(
pid int not null,
tag varchar(50)  not null,
constraint f2 foreign key (pid) references pictures(pid),
constraint f3 primary key(pid,tag)
);
create table likes(
pid int not null,
id int  not null,
constraint f5 foreign key (id) references instauser(id),
constraint f4 foreign key (pid) references pictures(pid),
constraint f6 primary key(pid,id)
);
insert into instauser values (1,'anl@gmail.com','anil');
insert into instauser values (2,'vamsi@gmail.com','vamsi');
insert into instauser values (3,'ravi@gmail.com','ravi');
insert into instauser values
(4,'kalyan@gmail.com','kalyan');
insert into instauser values (5,'sagar@gmail.com','sagar');
insert into instauser values (6,'nitesh@gmail.com','nitesh');
 select * from instauser
    ;

        ID EMAIL
```

--------- ----------

NAME

----

1 anl@gmail.com

anil

2 vamsi@gmail.com

vamsi

3 ravi@gmail.com

ravi

ID EMAIL

--------- ----------

NAME

----

4 kalyan@gmail.com

kalyan

5 sagar@gmail.com

sagar

6 nitesh@gmail.com

nitesh

6 rows selected.

```
insert into pictures values (1,'c/photos/n.jpg','chill',1,'24-MAR-18');

insert into pictures values (6,'c/photos/kl.jpg','chill bro',2,'21-JAN-18');

insert into pictures values (7,'c/photos/pp.jpg','Happy time',2,'21-FEB-18');

insert into pictures values (2,'c/photos/p.jpg','chill bro',1,'21-JAN-18');

insert into pictures values (3,'c/photos/k.jpg','chill',4,'2-SEP-18');

insert into pictures values (4,'c/photos/ll.jpg','chill',3,'24-MAR-17');

insert into pictures values (5,'c/photos/o.jpg','chill time',1,'24-FEB-18');

select * from pictures;
```

```
       PID PATH
---------- --------------------------------------------------
CAPTION                                             ID POSTDATE
-------------------------------------------------- ---------- ---------
         1 c/photos/n.jpg
chill                                                1 24-MAR-18

         2 c/photos/p.jpg
```

chill bro 1 21-JAN-18

3 c/photos/k.jpg

chill 4 02-SEP-18

PID PATH

---------- ----------

CAPTION ID POSTDATE

---------- ---------- ---------

4 c/photos/ll.jpg

chill 3 24-MAR-17

5 c/photos/o.jpg

chill time 1 24-FEB-18

6 c/photos/kl.jpg

chill bro 2 21-JAN-18

PID PATH

---------- ----------

CAPTION ID POSTDATE

---------- ---------- ---------

7 c/photos/pp.jpg

Happy time 2 21-FEB-18

7 rows selected.

```
insert into tags values (1,'Art');
insert into tags values (1,'Science');
insert into tags values (1,'History');
insert into tags values (2,'Art');
insert into tags values (3,'Art');
insert into tags values (2,'History');
insert into tags values (4,'History');
insert into tags values (5,'History');
```

```
 select * from tags;

       PID TAG
---------- --------------------------------------------------
         1 Art
         1 Science
         1 History
         2 Art
         3 Art
         2 History
```

4 History

5 History

8 rows selected.

```
insert into likes values (1,1);
insert into likes values (1,2);
insert into likes values (2,1);
insert into likes values (1,3);
insert into likes values (3,1);
insert into likes values (1,4);
insert into likes values (1,5);
insert into likes values (5,1);
insert into likes values (4,1);
```

```
select * from likes;
```

PID	ID
1	1
1	2
2	1
1	3
3	1

1	4
1	5
5	1
4	1

9 rows selected.

Queries:

1. Display Picture ids of pictures posted by user id "1"

A. select pid from pictures where id=1;

PID
1
2
5

2.Display captions of pictures posted by "2". Display 'No caption' if caption is not having any value.

A. select pid,caption from pictures where id=2;

PID	CAPTION

6 chill bro

7 Happy time

3.Which pictures (picture ids) and by which users (user ids) have been posted in last 1 year?

A.select pid ,id from pictures where postdate > '16-SEP-17';

PID	ID
1	1
2	1
3	4
5	1
6	2
7	2

6 rows selected.

4.Which picture/s (picture ids) has received maximum likes?

A. create view maxlikes as (select pid,count(id) as c from likes group by pid) ;

select pid,c as likecount from maxlikes where c=(select max(c) as l from maxlikes);

PID	LIKECOUNT
1	5

5. Display all picture ids in descending order of the likes they have got. Also display total no. of likes each picture has received

A. create view maxlikes as (select pid,count(id) as c from likes group by pid) ;

select pid,c as likecount from maxlikes order by c desc;

PID	LIKECOUNT
1	5
5	1
4	1
2	1
3	1

6. Display picture ids of pictures with more than 3 likes.

A. create view maxlikes as (select pid,count(id) as c from likes group by pid) ;

select pid from maxlikes where c>3;

PID
1

7.Who (user id) has liked the most pictures?

A. create view maxliked as (select id,count(pid) as c from likes group by id) ;

select id from maxliked where c=(select max(c) as l from maxliked);

ID

1

8. Which day of the week users are posting the maximum no. of pictures

A. create view week as (select pid ,TO_CHAR(postdate,'DY') as weekday from pictures);

create view daypost as(select weekday,count(pid)as c from week group by weekday);

select weekday,c as posts from daypost where c=(select max(c) from daypost);

WEE POSTS

--- ----------

SUN 3

9.Get the count of pictures posted during weekends.

A. select count(pid) as weekendposts from pictures where TO_CHAR(postdate,'DY') ='SAT' or TO_CHAR(postdate,'DY')='SUN';

WEEKENDPOSTS

5

10. Find pictures (picture ids) with more than 2 tags.

A. select pid from tags group by pid having count(tag)>2;

PID

1

11. Display pictures with "London" in their caption. Do case insensitive search.

A. select pid from pictures where lower(caption)=lower('London');

no rows selected

12.Which pictures (picture ids) are stored on my "D:" drive?

A. select pid from pictures where path like 'D/%';

no rows selected

13.Display year, pic_id of pictures posted by User id -3 in last 5 years.

A. select TO_CHAR(postdate,'Y') as postyear ,pid from pictures where id=3 and postdate>'16-SEP-13';

POSTYEAR PID

- ----------

17 4

14. Display first and last names of all users who have either have an account on gmail or yahoo.

A. select name from instauser where email like '%@gmail%' or email like'%@yahoo%';

NAME

----

anil

vamsi

ravi

kalyan

sagar

nitesh

6 rows selected.

15.Add new user with user-name, his first & last name and email.

A. insert into instauser values (7,'krishna@gmail.com','krishna');

16. Add a like for Picture id 6 by user id 7.

A. insert into likes values (6,7);

17. Delete all likes done by user id '3'.

A. delete from likes where id=3;

1 row deleted.

18.Update last name for user id 6' to 'Kumar'.

A. update instauser set name='Kumar' where id=6;

1 row updated.

19.Update email address of User "anil"to avvaru@gmail.com.

A. update instauser set email='avvaru@gmail.com' where name='anil';

1 row updated.

20.Add a new tag for "Literature".

A.

insert into tags values(5,'Literature');