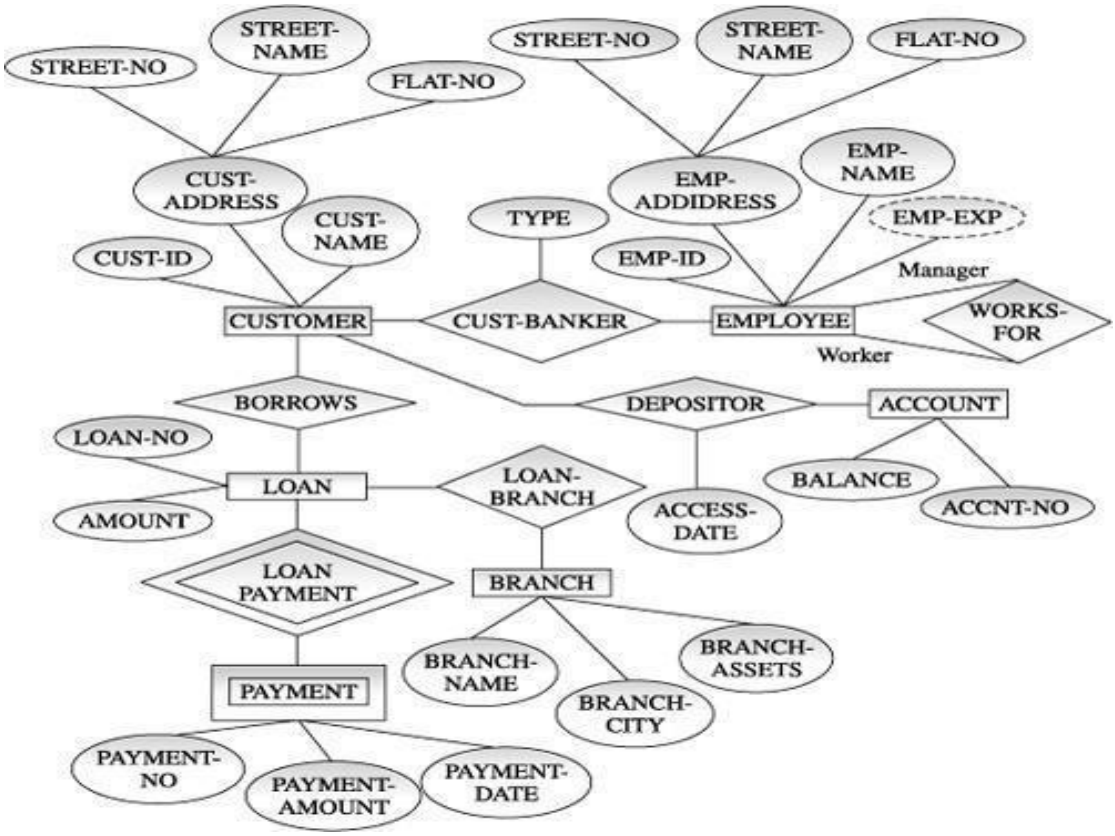
Practical 1

Example of ER diagram



Practical 2(a)

Creating Table:-

mysql> create table client_master(client_no int,client_name varchar(20),address varchar(50),city varchar(10),pincode int,state varchar(20), bal_due float,primary key(client_no));

Inserting values in Table:-

mysql> insert into client_master values('001','abhi','nasik','nasik','422004','MH','5000')

mysql> insert into client_master

values('002','piyu','nasik','nasik','422004','MH','10000');mysql> insert into

client_master values('003','abd','nasik','nasik','422003','MH','5000'); mysql> insert into

client_master values('004','abd','nasik','nasik','422003','MH','5000'); mysql> insert into

client_master values('005','abc','nasik','nasik','422003','MH','5000'); Viewing Created

Table:-

mysql> select * from client_master;

client_no	client_name	address	city	bal_due	pincode	state
1	abhi	5000		nasik	nasik	422004	MH
2	piyu	10000		nasik	nasik	422004	MH
3	abd	5000		nasik	nasik	422003	MH
4	abd	5000		nasik	nasik	422003	MH
5	abc	5000		nasik	nasik	422003	MH

5 rows in set (0.00 sec)

mysql> select client_name,client_no from client_master;

client_name	client_no
abhi	1
piyu	2
abd	3
abd	4
abc	5

5 rows in set (0.00 sec)

Creating view on Table:-

```
mysql> create view client as select client_no,client_name from c_master;
```

Viewing created view:-

```
mysql> select * from client;
+-----------+-------------+
| client_no | client_name |
+-----------+-------------+
| 5 | abc |
| 3 | abd |
| 1 | abhi |
| 4 | nut |
| 2 | piyu |
| 6 | xyz |
 +-----------+-------------+6
rows in set (0.23 sec)
```

Creating and Viewing Index:-

```
mysql> create index client_search on client_master(client_no);
Query OK, 0 rows affected (0.42 sec)
Records: 0
Duplicates: 0
Warnings: 0


mysql> create index client_find on client_master(client_name,city);affected
(0.41 sec)
Records: 0
Duplicates: 0
```

Practical 2(B)

Creating Table

mysql> create table Employee(emp_no int,emp_name varchar(20),date date,position varchar(20));

Inserting values in the Table:-

mysql> insert into Employee values('01','abc','2018-07-11','clerk','50000')
mysql> insert into Employee values('02','abhi','2018-05-11','ceo','150000');mysql> insert into Employee values('04','aqwgy','2018-06-21','te','10000');mysql> insert into Employee values('03','xyz','2018-05-21','hr','100000'); mysql> insert into Employee values('05','sfhjfh','2018-07-21','gt','12000');

Viewing the Created Table:-

mysql> select * from Employee;

+.....+.....+.....+.....+.....+

| emp_no | emp_name | date | position | salary |

+.....+.....+.....+.....+.....+

| 1 | abc | 2018-07-11 | clerk | 50000 |

| 2 | abhi | 2018-05-11 | ceo | 150000 |

| 3 | xyz | 2018-05-21 | hr | 100000 |

| 4 | aqwgy | 2018-06-21 | te | 10000 |

| 5 | sfhjfh | 2018-07-21 | gt | 12000 |

+.....+.....+.....+.....+.....+5

rows in set (0.00 sec)

Updating the Table:-

mysql> update Employee set emp_name='gjgj' where emp_no='5';

Query OK, 1 row affected (0.13 sec) Rows matched: 1 Changed: 1 Warnings: 0

Viewing the Updated Table:-

```
mysql> select * from Employee;
+---------+-----------+--------------+-----------+---------+
| emp_no | emp_name | date | position | salary |
+---------+-----------+--------------+-----------+---------+
| 1 | abc | 2018-07-11 | clerk | 50000 |
| 2 | abhi | 2018-05-11 | ceo | 150000 |
| 3 | xyz | 2018-05-21 | hr | 100000 |
| 4 | aqwgy | 2018-06-21 | te | 10000 |
| 5 | gjgj | 2018-07-21 | gt | 12000 |
+---------+-----------+--------------+-----------+---------+5
rows in set (0.00 sec)
```

SQL Operators(IN and NOT IN):-

```
mysql> select distinct emp_no from Employee where emp_no in(select emp_no from Employee);
 +---------+
| emp_no |
 +---------+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
 +---------+
 5 rows in set (0.03 sec)
```

```
mysql> select distinct emp_name from Employee where emp_name in(select emp_name from
Employee);
+-----------+
| emp_name |
 +-----------+
| abc |
| abhi |
| xyz |
| aqwgy |
```

| gjgj |
+.....+
5 rows in set (0.00 sec)

SQL Functions(Aggregate):-

mysql> select min(salary) from Employee;
+.....+
| min(salary) |
+.....+
| 10000 |
+.....+
1 row in set (0.04 sec)

mysql> select max(salary) from Employee;

+.....+
| max(salary) |
+.....+
| 150000 |
+.....+
1 row in set (0.00 sec)

mysql> select sum(salary) from Employee;

+.....+

| sum(salary) |
+.....+
| 322000 |
1 row in set (0.00 sec)

mysql> select avg(salary) from Employee;

+.....+
| avg(salary)
| +.....+
| 64400.0000 |
+.....+
1 row in set (0.00 sec)

mysql> select count(salary) from Employee;
+.....+
| count(salary)
| +.....+

| 5 |
+----------+
1 row in set (0.00 sec)

SQL Functions(Scalar):-

mysql> select lcase(emp_name) from Employee;
+----------+
| lcase(emp_name)
| +----------+
| ABC|
| ABHI |
| XYZ |
| AQWGY |
| GJGJ |
+----------+
5 rows in set (0.00 sec)

mysql> select ucase(emp_name) from Employee;
+----------+
| ucase(emp_no) |
+----------+
| abc |
| abhi |
| xyz |
| aqwgy |
| gjgj |
+----------+
5 rows in set (0.00 sec)

mysql> select mid(emp_no,1,2) from Employee;
+----------+
| mid(emp_no,1,2) |
+----------+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
+----------+
5 rows in set (0.00 sec)

Practical 3

Creating Table

```
mysql> create table capital(cap_no int,cap_name varchar(20),state_no int,primary key(cap_no));

mysql> create table state(state_no int,state_name varchar(20),state_code int,capital
varchar(20),primarykey(state_no));
```

Inserting values into Capital Table:-

```
mysql> insert into capital values('01','MH','01');
mysql> insert into capital values('02','RAJ','02');
mysql> insert into capital values('03','GOA','03');
mysql> insert into capital values('04','GUJ','04');
mysql> insert into capital values('05','KAR','05');
```

Inserting values into Capital Table:-

```
mysql> insert into state values('01','MH','01','MUM');
mysql> insert into state values('02','RAJ','02','JAI');
mysql> insert into state values('03','GOA','03','PAN');
mysql> insert into state values('04','GUJ','04','SUR');
mysql> insert into state values('05','KAR','05','BAN');
```

Viewing Created Tables:-

```
mysql> select * from capital;
+---------+-----------+-----------+
| cap_no | cap_name | state_no |
+---------+-----------+-----------+| 1 | MH | 1 |
| 2 | RAJ | 2 |
| 3 | GOA | 3 |
| 4 | GUJ | 4 |
| 5 | KAR | 5 |
+---------+-----------+-----------+5
rows in set (0.01 sec)

mysql> select * from state;
+-----------+-------------+-------------+----------+
| state_no | state_name | state_code | capital |
+-----------+-------------+-------------+----------+
| 1 | MH | 1 | MUM |
| 2 | RAJ | 2 | JAI |
| 3 | GOA | 3 | PAN |
| 4 | GUJ | 4 | SUR |
```

| 5 | KAR | 5 | BAN |
+.....+.....+.....+.....+

Inner Join

mysql> select capital.cap_no, state.state_no from capital inner join state on capital.cap_no=state.state_no;

+.....+.....+
| cap_no | state_no |
+.....+.....+
1	1
2	2
3	3
4	4
5	5
+.....+.....+
5 rows in set (0.06 sec)

mysql> select capital.cap_no, state.state_no from capital inner join state on capital.cap_no=state.state_no;

+.....+.....+
| cap_no | state_no |
+.....+.....+
|
5 |
5 |
+.....+.....+
1 row in set (0.00 sec)

Outer Join

mysql> select capital.cap_no, state.state_no from capital left join state on capital.cap_no=state.state_no;

+.....+.....+
| cap_no | state_no |
+.....+.....+
1	NULL
2	NULL
3	NULL
4	NULL
5	5
+.....+.....+
5 rows in set (0.00 sec)

mysql> select capital.cap_no, state.state_no from capital right join state on capital.cap_no=state.state_no;

+---------+----------+
| cap_no | state_no |
+---------+----------+
|
5 |
5 || NULL | 46 |
NULL	58
NULL	78
NULL	458
NULL	489
+---------+----------+
6 rows in set (0.00 sec)

Combining Joins

mysql> select capital.cap_no,capital.cap_name,state.capital,state.state_no from capital left join state on capital.cap_no=state.state_no union selectcapital.cap_no,capital.cap_name,state.capital,state.state_no from capital right join state on capital.cap_no=state.state_no;

+---------+----------+---------+----------+
| cap_no | cap_name | capital | state_no |
+---------+----------+---------+----------+
1	MH	NULL	NULL
2	RAJ	NULL	NULL
3	GOA	NULL	NULL
4	GUJ	NULL	NULL
5	KAR	BHO	5
NULL	NULL	PAN	46
NULL	NULL	JAI	58
NULL	NULL	MUM	78
NULL	NULL	BAN	458
NULL	NULL	SUR	489
+---------+----------+---------+----------+10
rows in set (0.00 sec)

Nested Queries

```
mysql> select * from state where state_no=(select state_no from state where
state_name='MH');
+----------+------------+------------+---------+
| state_no | state_name | state_code | capital |
+----------+------------+------------+---------+
|
78 | MH
|
1 | MUM
|
+----------+------------+------------+---------+1
row in set (0.06 sec)

mysql> select * from state where state_no=(select state_no from state where
state_name='GUJ');
+----------+------------+------------+---------+
| state_no | state_name | state_code | capital |
+----------+------------+------------+---------+
|
489 | GUJ
|
4 | SUR
|
+----------+------------+------------+---------+1
row in set (0.00 sec)

mysql> select * from state where state_no=(select capital.state_no from capital
where cap_name='KAR');
+----------+------------+------------+---------+
| state_no | state_name | state_code | capital |
+----------+------------+------------+---------+
|
5 | MP
|
5 | BHO
|
+----------+------------+------------+---------+1
row in set (0.00 sec)
```

Practical 4

Creating Table

```
mysql> create table Borrower(Roll_no int,Name varchar(20),DateofIssue date,NameofBook
varchar(20),Status varchar(10));

mysql> create table Fine(Roll_no int, DateofIssue date, Amount int);
```

Viewing Table:-

```
mysql> select * from Borrower;
-> //
+---------+---------+-------------+---------+---------+
| roll_no | name
| DOI
| book_name | status
|
+---------+---------+-------------+---------+---------+
| 12 | patel | 2018-07-01 | xyz | issued |
| 14 | shinde | 2018-06-01 | oop | issued |
| 16 | bhangale | 2018-05-01 | coa | returned |
| 18 | rebello | 2018-06-15 | toc | returned |
| 20 | patil | 2018-05-15 | mp | issued
|
+---------+---------+-------------+---------+---------+5
rows in set (0.00 sec)
```

PL/SQL Block Code

```
mysql> create procedure B(roll_new int,book_name varchar(20))
-> begin
-> declare X integer;
-> declare continue handler for not found
-> begin
-> select 'NOT FOUND';
-> end;
-> select datediff(curdate(),DOI) into X from Borrower
where roll_no=roll_new;
->
if (X>15&&X<30)
-> then
-> insert into Fine values(roll_new,curdate(),(X*5));
-> end if;
-> if (X>30)
-> then
-> insert into Fine values(roll_new,curdate(),(X*50));
```

```
-> end if;
-> update Borrower set status='returned' where
roll_no=roll_new;
-> end;
-> //
Query OK, 0 rows affected (0.02 sec)
```

Extracting values

```
mysql> call B(12,'xyz');-> //
Query OK, 1 row affected (0.42 sec)

mysql> select * from Fine;//
+---------+-------------+--------+
| roll_no | fine_date
| amount |
+---------+-------------+--------+
|
12 | 2018-07-28 |
135 |
+---------+-------------+--------+1
row in set (0.00 sec)

mysql> select * from Borrower;//
+---------+----------+-------------+-----------+----------+
| roll_no | name
| DOI
| book_name | status
|
+---------+----------+-------------+-----------+----------+
| 12 | patel | 2018-07-01 | xyz | returned |
| 14 | shinde | 2018-06-01 | oop | issued
| 16 | bhangale | 2018-05-01 | coa | returned |
| 18 | rebello | 2018-06-15 | toc | returned |
| 20 | patil | 2018-05-15 | mp | issued
||
+---------+----------+-------------+-----------+----------+5
rows in set (0.00 sec)

mysql> call B(20,'patil');
-> //
Query OK, 1 row affected (0.35 sec)
mysql> select * from Fine;//
+---------+-------------+--------+
| roll_no | fine_date
| amount |
+---------+-------------+--------+
|
```

12 | 2018-07-28 |
135 ||
20 | 2018-07-28 |
3700 |
+----------+----------+----------+2
rows in set (0.00 sec)
mysql> select * from Borrower;//
+----------+----------+----------+----------+----------+
| roll_no | name
| DOI
| book_name | status
|
+----------+----------+----------+----------+----------+
| 12 | patel | 2018-07-01 | xyz | returned |
| 14 | shinde | 2018-06-01 | oop | issued
16	bhangale	2018-05-01	coa	returned
18	rebello	2018-06-15	toc	returned
20	patil	2018-05-15	mp	returned
+----------+----------+----------+----------+----------+5
rows in set (0.00 sec)

Practical 5

Creating Table

```
mysql> create table marks(roll_no int,name varchar(20),total_marks
varchar(20));mysql> create table result(roll_no int,name varchar(20),class
varchar(20));
```

Inserting values in the Table

```
mysql> insert into marks values('1','Abhi','1400');
mysql> insert into marks values('2','piyush','980');
mysql> insert into marks values('3','hitesh','880');
mysql> insert into marks values('4','ashley','820');
mysql> insert into marks values('5','partik','740');
mysql> insert into marks values('6','patil','640');
```

Creating Procedure

```
mysql> create procedure proc_result(in marks int,out class
char(20))
-> begin
-> if(marks<1500&&marks>990)
-> then
-> set class='Distincton';
-> end if;
-> if(marks<989&&marks>890)
-> then
-> set class='First Class';
-> end if;
-> if(marks<889&&marks>825)
-> then
-> set class='Higher Second Class';
-> end if;
-> if(marks<824&&marks>750)
-> then
-> set class='Second Class';-> end if;if(marks<749&&marks>650)
-> then
-> set class='Passed';
-> end if;
```

```
-> if(marks<649)
-> then
-> set class='Fail';
-> end if;
-> end;
-> //
Query OK, 0 rows affected (0.00 sec)
```

Creating Function

```
mysql> create function final_result3(R1 int)
-> returns int
-> begin
-> declare fmarks integer;
-> declare grade varchar(20);
-> declare stud_name varchar(20);
-> select marks.total_marks,marks.name into
fmarks,stud_name from marks where marks.roll_no=R1;
-> call proc_grade(fmarks,@grade);
-> insert into result values(R1,stud_name,@grade);
-> return R1;
-> end
```

Calling Function

```
mysql> select final_result3(2);
-> //
+----------------------+
| final_result3(2) |
+----------------------+|2 |
+----------------------+
1 row in set (0.05 sec)

mysql> select final_result3(5);//
+----------------------+
| final_result3(5) |
+----------------------+
|
5 |
+----------------------+
```

1 row in set (0.05 sec)

Viewing table after performing all Operations

mysql> select * from result;

-> //

```
+.....+.....+.....+
| roll_no | name
| class
|+.....+.....+.....+
| 1 | NULL | Distincton |
| 1 | Abhi | Distincton |
| 1 | Abhi | Distincton |
| 2 | piyush | First Class | 3 | hitesh | Higher Second Class |
| 4 | ashley | Second Class |
| 5 | partik | Passed |
|
+.....+.....+.....+7
```

rows in set (0.00 sec)

Practical 6

Creating Table

```
mysql> create table o_rollcall(roll_no int,name varchar(20),address

varchar(20));mysql> create table n_rollcall(roll_no

int,namevarchar(20),addressvarchar(20));
```

Inserting values in the Table:-

```
mysql> insert into o_rollcall values('1','Hitesh','Nandura');
mysql> insert into o_rollcall values('2','Piyush','MP');
mysql> insert into o_rollcall values('3','Ashley','Nsk');
mysql> insert into o_rollcall values('4','Kalpesh','Dhule');
mysql> insert into o_rollcall values('5','Abhi','Satara');
```

Creating Cursor:-

```
mysql> create procedure p3(in r1 int)
-> begin
-> declare r2 int;
-> declare exit_loop boolean;
-> declare c1 cursor for select roll_no from o_rollcall
where roll_no>r1;
-> declare continue handler for not found set
exit_loop=true;
-> open c1;
-> e_loop:loop
-> fetch c1 into r2;
-> if not exists(select * from n_rollcall where
roll_no=r2)
-> then
-> insert into n_rollcall select * from o_rollcall whereroll_no=r2;
-> end if;
-> if exit_loop
-> then
-> close c1;
-> leave e_loop;
-> end if;
-> end loop e_loop;-> end
-> //
```

Extracting values in n_rollcall table using cursor

```
 mysql> call p3(3);
-> //
Query OK, 0 rows affected (0.10 sec)
mysql> select * from n_rollcall;
-> //
+----------+----------+----------+
| roll_no | name
| address |
+----------+----------+----------+
| 4 | Kalpesh | Dhule |
| 5 | Abhi |
| Satara
+----------+----------+----------+2
rows in set (0.00 sec)
```

Inserting values to check merging:-

```
mysql> insert into o_rollcall values('6','Patil','Kolhapur');

mysql> call p3(4);
-> //
Query OK, 0 rows affected (0.05 sec)
mysql> select * from n_rollcall;
-> //
+----------+----------+----------+
| roll_no | name
| address
+----------+----------+----------+
| 4 | Kalpesh | Dhule |
| 5 | Abhi | Satara |
| 1 | Hitesh | Nandura |
| 2 | Piyush | MP |
| 3 | Ashley | Nsk |
| 6 | Patil | Kolhapur |
+----------+----------+----------+6
rows in set (0.00 sec)
```

Practical 7

Creating Table

```
mysql> create table borrower2(roll_no int,name varchar(20),date_of_issue date,book varchar2);
```

Inserting values in the Table:-

```
mysql>insert into borrower2 values('1','nick','2018-06-10','wings_of_fire','avaliable','APJ');
mysql> insert into borrower2 values('2','mira','2018-05-11','leaves_life','not_avaliable','borwarkar');
mysql> insert into borrower2 values('3','rina','2018-02-12','unusal','avaliable','johar');
mysql> insert into borrower2 values('4','harsha','2018-06-20','skylimit','avaliable','ingale');
mysql> insert into borrower2 values('5','tej','2018-04-20','highway','not_avaliable','klm');
```

Inserting Trigger:-

```
mysql> delimiter //
mysql> create trigger library after insert on borrower1 for
each row
-> begin
-> insert into audit1
values(new.roll_no,new.name,new.date_of_issue,new.book_name,ne
w.status,new.author,current_timestamp);
-> end;
-> //Query OK, 0 rows affected (0.10 sec)
mysql> insert into borrower1 values('6','xyz','2018-09-
06','aaa','avaliable','xxx');
-> //
Query OK, 1 row affected (0.07 sec)
```

Viewing borrower table and audit table after creation of trigeger:-

```
mysql> select * from borrower1;
-> //
+----------+---------+-------------------+
+-----------------+------------+
| roll_no | name
| author
|
| date_of_issue | book_name
| status
+----------+---------+-------------------+
+-----------------+------------+
|
| APJ
```

1 | nick
|
| 2018-06-10
| wings_of_fire | avaliable
|
2 | mira
| 2018-05-11
not_avaliable | borwarkar | | leaves_life |
|
3 | rina
| johar
| | unusal | avaliable
|
4 | harsha | 2018-06-20
| ingale
| | skylimit | avaliable
|
5 | tej
| 2018-04-20
not_avaliable | klm
| | highway |
|
| xxx | aaa | avaliable6 |
xyz
|
| 2018-02-12
| 2018-09-06
+----------+----------+--------------------+
+--------------------+------------+6
rows in set (0.00 sec)

mysql> select * from audit1;
-> //
+----------+-------+--------------------+------------+
+--------+-------------------- +| roll_no | name | date_of_issue | book_name | status
author | ts
||
+----------+-------+--------------------+------------+
+----------+--------------------------+
|
6 | xyz | 2018-09-06
| 2018-08-29 15:46:13 |
| aaa
| avaliable | xxx

+----------+------+----------------+----------+
+---------+-------------------------+1
row in set (0.00 sec)

Updating Trigger

mysql> create trigger library1 after update on borrower1 for each row
->
begin
->
insert into audit1 values(new.roll_no,new.name,new.date_of_issue,new.book_name,ne
w.status,new.author,current_timestamp);
-> end;
-> //

Performing record update to check whether update operation in performed or not

mysql> update borrower1 set roll_no='8',book_name='leaf' where name='xyz';
-> //
Query OK, 1 row affected (0.04 sec)
Rows matched: 1
Changed: 1
Warnings: 0

mysql> select *from borrower1;
-> //
+----------+---------+----------------+
+----------------+-----------+
| roll_no | name
| author
|
| date_of_issue | book_name
| status+--------+---------+----------------+
+----------------+-----------+
|
| APJ
1 | nick
|
| 2018-06-10
| wings_of_fire | avaliable
|
2 | mira

| 2018-05-11
not_avaliable | borwarkar | | leaves_life |
|
3 | rina
| johar
| | unusal | avaliable
|
4 | harsha | 2018-06-20
| ingale
| | skylimit | avaliable
|
5 | tej
| 2018-04-20
not_avaliable | klm
| | highway |
|
| xxx | leaf | avaliable8 |
xyz
|
| 2018-02-12
| 2018-09-06
+----------+----------+----------+
+----------+----------+6
rows in set (0.00 sec)

Practical 8

Output :-

```
sl2-pc5@sl2pc5-HP-Compaq-4000-Pro-SFF-PC:~$
mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with
; or \g.
Your MySQL connection id is 42
Server version: 5.5.61-0ubuntu0.14.04.1 (Ubuntu)
Copyright (c) 2000, 2018, Oracle and/or its
affiliates. All rights reserved.
Oracle is a registered trademark of Oracle
Corporation and/or itsaffiliates. Other names may be trademarks
oftheir respective
owners.
Type 'help;' or '\h' for help. Type '\c' to
clear the current input statement.

mysql> create database info;
Query OK, 1 row affected (0.03 sec)

mysql> use info;
Database changed

mysql> create table result (stud_RollNoint,stud_Name varchar(20),stud_Dept
varchar(20));
Query OK, 0 rows affected (0.08 sec)

mysql> select *from result;
+-------------+-----------+ ---------- +| stud_RollNo | stud_Name | stud_Dept |
+.....+.....+.....+
|
1 | abc
| comp
|
+.....+.....+.....+1
row in set (0.00 sec)
```

//ADD DATA

```
mysql> select *from result;
+-------------+-----------+-----------+
| stud_RollNo | stud_Name | stud_Dept |
+-------------+-----------+-----------+
|
1 | abc
| comp
|
|
2 | harsha
| comp
|
|
3 | tej
| comp
|
|
4 | rina
| mech
|
+-------------+-----------+-----------+4 rows in set (0.00 sec)
```

//DELETE DATA

```
mysql> select *from result;
+-------------+-----------+ ---------- +| stud_RollNo | stud_Name | stud_Dept |
+-------------+-----------+-----------+
|
2 | harsha
| comp
|
|
3 | tej
| comp
|
|
4 | rina
| mech
|
+-------------+-----------+-----------+
```

Practical 9

Creating Collection

```
> db.createCollection('Student');
{ "ok" : 1 }
```

Inserting values into collection

```
> db.Student.insert({'Rno':'1','Name':'Piyush','Class':'TE COMP'});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({'Rno':'2','Name':'Abhi','Class':'TE COMP'});
WriteResult({ "nInserted" : 1 })
>db.Student.insert({'Rno':'3','Name':'Ashley','Class':'TE COMP'});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({'Rno':'4','Name':'Hitesh','Class':'TE COMP'});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({'Rno':'5','Name':'Pratik','Class':'TE COMP'});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({'Rno':'6','Name':'Pratik','Class':'TE COMP'});
WriteResult({ "nInserted" : 1 })
```

Finding Record in Collection

```
> db.Student.find();
{ "_id" : ObjectId("5ba1d618f5bbacd4ad81568d"), "Rno" : "1",
"Name" : "Piyush", "Class" : "TE COMP" }
{ "_id" : ObjectId("5ba1d625f5bbacd4ad81568e"), "Rno" : "2",
"Name" : "Abhi", "Class" : "TE COMP" }
{ "_id" : ObjectId("5ba1d63af5bbacd4ad81568f"), "Rno" : "3",
"Name" : "Ashley", "Class" : "TE COMP" }
{ "_id" : ObjectId("5ba1d647f5bbacd4ad815690"), "Rno" : "4",
"Name" : "Hitesh", "Class" : "TE COMP" }
{ "_id" : ObjectId("5ba1d65ef5bbacd4ad815691"), "Rno" : "5",
"Name" : "Pratik", "Class" : "TE COMP" }
{ "_id" : ObjectId("5ba1d66df5bbacd4ad815692"), "Rno" : "6",
"Name" : "Pratik", "Class" : "TE COMP" }
```

Finding Record in pretty format

```
> db.Student.find().pretty();
{
"_id" : ObjectId("5ba1d618f5bbacd4ad81568d"),
```

"Rno" : "1",
"Name" : "Piyush",
"Class" : "TE COMP"
}
{
"_id" : ObjectId("5ba1d625f5bbacd4ad81568e"),
"Rno" : "2","Name" : "Abhi",
"Class" : "TE COMP"
}
{
"_id" : ObjectId("5ba1d63af5bbacd4ad81568f"),
"Rno" : "3",
"Name" : "Ashley",
"Class" : "TE COMP"
}
{
"_id" : ObjectId("5ba1d647f5bbacd4ad815690"),
"Rno" : "4",
"Name" : "Hitesh",
"Class" : "TE COMP"}
{
"_id" : ObjectId("5ba1d65ef5bbacd4ad815691"),
"Rno" : "5",
"Name" : "Pratik",
"Class" : "TE COMP"
}
{
"_id" : ObjectId("5ba1d66df5bbacd4ad815692"),
"Rno" : "6",
"Name" : "Pratik",
"Class" : "TE COMP"
}

Updating Collection

```
db.Student.update({'Name':'Hitesh'},{$set: {'Name':'Henry'}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1})
> db.Student.find().pretty();
{
"_id" : ObjectId("5b8fad4ef00832a0a50b5036"),
"Rno" : "1",
"Name" : "Piyush",
```

"Class" : "TE COMP"
}{
"_id" : ObjectId("5b8fad62f00832a0a50b5037"),
"Rno" : "2",
"Name" : "Abhi",
"Class" : "TE COMP"
}{
"_id" : ObjectId("5b8fad70f00832a0a50b5038"),
"Rno" : "3",
"Name" : "Ashley",
"Class" : "TE COMP"
}{
"_id" : ObjectId("5b8fad7ff00832a0a50b5039"),
"Rno" : "4",
"Name" : "Henry",
"Class" : "TE COMP"
}{ "_id" :
ObjectId("5b8fad8df00832a0a50b503a"),
"Rno" : "5",
"Name" : "Pratik",
"Class" : "TE COMP"
}{
"_id" : ObjectId("5b8fada4f00832a0a50b503b"),"Rno" : "6",
"Name" : "Pratik",
"Class" : "TE COMP"
}
> db.Student.remove({'ADD':'MP'});
WriteResult({ "nRemoved" : 1 })
> db.Student.find().pretty();
{
"_id" : ObjectId("5b8fad62f00832a0a50b5037"),
"Rno" : "2",
"Name" : "Abhi",
"Class" : "TE COMP"
}{
"_id" : ObjectId("5b8fad70f00832a0a50b5038"),
"Rno" : "3",
"Name" : "Ashley",
"Class" : "TE COMP"
}{
"_id" : ObjectId("5b8fad7ff00832a0a50b5039"),
"Rno" : "4",
"Name" : "Henry",
"Class" : "TE COMP"

}
{
"_id" : ObjectId("5b8fad8df00832a0a50b503a"),
"Rno" : "5",
"Name" : "Pratik",
"Class" : "TE COMP"
} {"_id" :
ObjectId("5b8fada4f00832a0a50b503b"),"Rno" :
"6",
"Name" : "Pratik",
"Class" : "TE COMP"
}

Saving Collection

```
>db.Student.save({_id:ObjectId("5b8fad4ef00832a0a50b5036"),"RNO
":"1","NAME":"PIYUSH","CLASS":"TE COMP","ADD":"MP"});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1})
```

Practical 10

CREATE COLLECTION WEBSITE

```
> db.createCollection('website');
{ "ok" : 1 }
```

INSERT VALUES IN WEBSITE

```
> db.website.insert({'roll':'1','name':'harsh','amount':1000,'u
r l':'www.yahoo.com'});
WriteResult({ "nInserted" : 1 })
>db.website.insert({'roll':'2','name':'jitesh','amount':2000,'url':'www.yahoo.com
'});
WriteResult({ "nInserted" : 1 })
>db.website.insert({'roll':'3','name':'rina','amount':3000,'url':'www.google.com'
});
WriteResult({ "nInserted" : 1 })
>db.website.insert({'roll':'4','name':'ash','amount':4000,'url':'www.gmail.com'})
;
WriteResult({ "nInserted" : 1 })
>db.website.insert({'roll':'5','name':'ash','amount':1000,'url':'www.pvg.com'});
WriteResult({ "nInserted" : 1 })
```

SUM AGGREGATE

```
> db.website.aggregate({$group:{_id:"$name","total":{$sum:"$amount"}}});
{ "_id" : "ash", "total" : 5000 }
{ "_id" : "rina", "total" : 3000 }
{ "_id" : "jitesh", "total" : 2000 }
{ "_id" : "harsh", "total" : 2000 }
```

AVG AGGREGATE

```
> db.website.aggregate({$group:{_id:"$name","total":
{$avg:"$amount"}}});
{ "_id" : "ash", "total" : 2500 }
{ "_id" : "rina", "total" : 3000 }
{ "_id" : "jitesh", "total" : 2000 }
{ "_id" : "harsh", "total" : 1000 }
```

MIN AGGREGATION

```
> db.website.aggregate({$group:{_id:"$name","total":{$min:"$amount"}}});
{ "_id" : "ash", "total" : 1000 }
{ "_id" : "rina", "total" : 3000 }
```

{ "_id" : "jitesh", "total" : 2000 }
{ "_id" : "harsh", "total" : 1000 }

MAX AGGREGATION

> db.website.aggregate({$group:{_id:"$name","total":{$max:"$amount"}}});
{ "_id" : "ash", "total" : 4000 }
{ "_id" : "rina", "total" : 3000 }
{ "_id" : "jitesh", "total" : 2000 }
{ "_id" : "harsh", "total" : 1000 }

FIRST AGGREGATION

> db.website.aggregate({$group:{_id:"$name","total":{$first:"$amount"}}});
{ "_id" : "ash", "total" : 4000 }
{ "_id" : "rina", "total" : 3000 }
{ "_id" : "jitesh", "total" : 2000 }
{ "_id" : "harsh", "total" : 1000 }

LAST AGGREGATION

> db.website.aggregate({$group:{_id:"$name","total":{$last:"$amount"}}});
{ "_id" : "ash", "total" : 1000 }
{ "_id" : "rina", "total" : 3000 }
{ "_id" : "jitesh", "total" : 2000 }
{ "_id" : "harsh", "total" : 1000 }

PUSH AGGREGATION

> db.website.aggregate({$group:{_id:"$name","total":
{$push:"$amount"}}});
{ "_id" : "ash", "total" : [4000, 1000] }
{ "_id" : "rina", "total" : [3000] }
{ "_id" : "jitesh", "total" : [2000] }
{ "_id" : "harsh", "total" : [1000, 1000] }

INDEXING

> db.createCollection('website1');
{ "ok" : 1 }
> db.website1.insert({'r':1,'name':'harsh'});
WriteResult({ "nInserted" : 1 })
> db.website1.find().pretty()
{ "_id" : ObjectId("5ba3509a444926329738012d"), "roll" : 1,
"name" : "harsh" }

```
{ "_id" : ObjectId("5ba35293444926329738012e"), "roll" : 1,
"name" : "harsh" }
> db.website1.createIndex({'name':1})
{ "numIndexesBefore" : 2, "note" : "all indexes already
exist", "ok" : 1 }

> db.website1.createIndex({'name':-1})
{
"createdCollectionAutomatically" : false,
"numIndexesBefore" : 2,
"numIndexesAfter" : 3,
"ok" : 1
}
```

Viewing Index

```
> db.website1.getIndexes()[
{"v" : 1,
"key" : {
"_id" : 1
},
"name" : "_id_",
"ns" : "harsh.website1"
},
{
"v" : 1,
"key" : {
"name" : 1
},
"name" : "name_1", "ns" :
"harsh.website1"
},
{
"v" : 1,
"key" : {
"name" : -1
},
"name" : "name_-1", "ns" :
"harsh.website1"
}]
> db.website1.createIndex({'name':-1})
{ "numIndexesBefore" : 3, "note" : "all indexes already
exist", "ok" : 1 }
```

DROP INDEX

```
> db.website.dropIndex({'name':-1})
{ "nIndexesWas" : 3, "ok" : 1 }>
db.website1.dropIndex({'name':1})
{ "nIndexesWas" : 2, "ok" : 1 }> db.website1.dropIndex({'name':1})
{
"nIndexesWas" : 1,
"ok" : 0,
"errmsg" : "can't find index with key:{ name: 1.0 }"
}
```

GET INDEXING

```
 > db.website1.find().pretty()

{ "_id" : ObjectId("5ba3509a444926329738012d"), "roll" : 1,

"name" : "harsh" }

{ "_id" : ObjectId("5ba35293444926329738012e"), "roll" : 1,

"name" : "harsh" }
```

Practical No. 11

Creating Collection

```
> db.createCollection('Journal');

{ "ok" : 1 }
```

Inserting values into collection

```
>db.Journal.insert({'book_id':1,'book_name':'JavacdOOP','amt':500,'status':'Available'});

WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book_id':1,'book_name':'JavaOOP','amt':400,'status':'Not Available'});

WriteResult({ "nInserted" : 1 })

>db.Journal.insert({'book_id':1,'book_name':'Java','amt':300,'status':'Not Available'});

WriteResult({ "nInserted" : 1 })

>db.Journal.insert({'book_id':2,'book_name':'Java','amt':300,'status':'Available'});

WriteResult({ "nInserted" : 1 })>

>db.Journal.insert({'book_id':2,'book_name':'OPP','amt':200,'status':'Available'});

WriteResult({ "nInserted" : 1 })

>db.Journal.insert({'book_id':2,'book_name':'C+','amt':200,'status':'Available'});

 WriteResult({ "nInserted" : 1 })

>db.Journal.insert({'book_id':3,'book_name':'C+','amt':150,'status':'Available'});

 WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book_id':3,'book_name':'C++','amt':200,'status':'Not Available'});

WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book_id':4,'book_name':'OPP C++','amt':300,'status':'Not Available'});

WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book_id':5,'book_name':'OPP C++','amt':400,'status':'Available'});

WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book_id':5,'book_name':'C++','amt':400,'status':'Available'});

WriteResult({ "nInserted" : 1 })
```

```
> db.Journal.insert({'book_id':5,'book_name':'C++ Java','amt':400,'status':'Not Available'});

WriteResult({ "nInserted" : 1 })
```

Applying Map reduce function

```
> var mapfunction=function(){ emit(this.book_id,this.amt)};
> var reducefunction=function(key,value){return Array.sum(value);};
> db.Journal.mapReduce(mapfunction,reducefunction,{'out':'new'});
{
"result" : "new",
"timeMillis" : 49,"counts" : {
"input" : 12,
"emit" : 12,
"reduce" : 4,
"output" : 5
},
"ok" : 1
}
```

Viewing Map reduce function

```
> db.Journal.mapReduce(mapfunction,reducefunction,
{'out':'new'}).find().pretty();
{ "_id" : 1, "value" : 1200 }
{ "_id" : 2, "value" : 700 }
{ "_id" : 3, "value" : 350 }
{ "_id" : 4, "value" : 300 }
{ "_id" : 5, "value" : 1200 }
>>>
db.new.find().pretty();

{ "_id" : 1, "value" : 1200 }
{ "_id" : 2, "value" : 700 }
{ "_id" : 3, "value" : 350 }
{ "_id" : 4, "value" : 300 }
{ "_id" : 5, "value" : 1200 }
>
```

Practical 12

For connecting the Mongodb database connectivity we have use the “Java” as the Front EndLanguage

Java Program for MongoDB database connectivity

```
import java.net.UnknownHostException;
import java.util.Scanner;
import com.mongodb.*;
public class DatabaseConnectivity {
private static void choice_input(){
System.out.println("\n1.insert data into database\n2.update
database
documents\n3.delete database documents\n4.show database
collections\n5.Exit");
}
public static void main(String[] args) {
String key, value;
Scanner scanner = new Scanner(System.in);
int choice;
try {
Mongo mongo = new Mongo("localhost", 27017);DB
db = mongo.getDB("myDb");
DBCollection collection = db.getCollection("dummyColl");
do{
choice_input();
System.out.println("Enter your choice: ");
choice = scanner.nextInt();switch (choice){
case 1:
BasicDBObject document = new BasicDBObject();
String ch;
do{
System.out.println("Enter key: "); key
= scanner.next();
System.out.println("Enter value: ");
value = scanner.next();
document.put(key, value);
System.out.println("Do you want to enter more(y/n)? ");
ch = scanner.next();
} while (!ch.equals("n"));
collection.insert(document);
```

```
break;
case 2:
BasicDBObject searchObj = new BasicDBObject();
System.out.println("Enter searched key: ");
key = scanner.next();
System.out.println("Enter searched value: ");
value = scanner.next();
searchObj.put(key, value);
BasicDBObject newObj = new BasicDBObject();
System.out.println("Enter new key: ");
key = scanner.next();
System.out.println("Enter new value: ");
value = scanner.next();
newObj.put(key, value);
collection.update(searchObj, newObj);
break;
case 3:
System.out.println("Enter removable key: ");key = scanner.next();
System.out.println("Enter removable value: ");
value = scanner.next();
BasicDBObject removableObj = new BasicDBObject();
removableObj.put(key, value);
collection.remove(removableObj);
break;
case 4:
DBCursor cursorDoc = collection.find();
while (cursorDoc.hasNext()) {
System.out.println(cursorDoc.next());
}
break;
case 5:
System.exit(0);
break;
}
} while(choice != 6);
} catch (UnknownHostException | MongoException e) {
e.printStackTrace();
}
}
}
```

Output

```
1.insert data into database
2.update database documents
3.delete database documents
4.show database collections
5.Exit

Enter your choice:
1
Enter key:2
Enter value:
harish
Do you want to enter more(y/n)?N
1.insert data into database
2.update database documents
3.delete database documents
4.show database collections
5.Exit

Enter your choice:
2
Enter searched key:
2
Enter searched value:
harish
Enter new key:
1
Enter new value:
Sam
1. insert data into database
2.update database documents
3.delete database documents
4.show database collections
5.Exit

Enter your choice:
4{
"_id" : { "$oid" : "5bb453bce4b0283ac9d3205d"} , "1" : "sam"}
1.insert data into database
2. update database documents
```

3. delete database documents
4.show database collections
5.Exit

Enter your choice:
3
Enter removable key:
3
Enter removable value:
hari
1. insert data into database
2.update database documents
3.delete database documents
4.show database collections
5.Exit

Enter your choice:
4{
"_id" : { "$oid" : "5bb453bce4b0283ac9d3205d"} , "1" : "sam"}
1.insert data into database
2. update database documents
3.delete database documents
4.show database collections
5.Exit

Enter your choice:
5