Assignment 2

TITLE: Design and Develop SQL DDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence, Synonym

---

```
mysql> use Abhi;
Database changed
mysql> show tables;
Empty set (0.02 sec)
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));
Query OK, 0 rows affected (0.51 sec)
mysql> select * from client_master;
Empty set (0.02 sec)
mysql> insert into client_master
values('001','abhi','nasik','nasik','422004','MH','5000');
Query OK, 1 row affected (0.14 sec)
mysql> insert into client_master
values('002','piyu','nasik','nasik','422004','MH','10000');
Query OK, 1 row affected (0.09 sec)
mysql> insert into client_master
values('003','abd','nasik','nasik','422003','MH','5000');
Query OK, 1 row affected (0.06 sec)
mysql> insert into client_master
values('004','abd','nasik','nasik','422003','MH','5000');
Query OK, 1 row affected (0.05 sec)
mysql> insert into client_master
values('005','abc','nasik','nasik','422003','MH','5000');

Query OK, 1 row affected (0.06 sec)
mysql> select * from client_master;
+-----------+-------------+---------+-------+---------+-------
+---------+
| client_no | client_name | address | city | pincode | state
| bal_due |
+-----------+-------------+---------+-------+---------+-------
+---------+
| 1 | abhi | nasik | nasik | 422004 | MH
| 5000 |
| 2 | piyu | nasik | nasik | 422004 | MH
| 10000 |
```

```
| 3 | abd | nasik | nasik | 422003 | MH
| 5000 |
| 4 | abd | nasik | nasik | 422003 | MH
| 5000 |
| 5 | abc | nasik | nasik | 422003 | MH
| 5000 |
+-----------+-------------+---------+-------+---------+-------
+---------+
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)
mysql> insert into client_master
values('006','xyz','nasik','nasik','422004','MH','6000');
Query OK, 1 row affected (0.15 sec)
mysql> select client_name,client_no from client_master;
+-------------+-----------+
| client_name | client_no |
+-------------+-----------+
| abhi | 1 |
| piyu | 2 |
| abd | 3 |
| abd | 4 |
| abc | 5 |
| xyz | 6 |
+-------------+-----------+
6 rows in set (0.08 sec)
mysql> create table product_master(product_no int,description
varchar(20),profit_per float,unit_measure varchar(10),quantity
int,reorder int,sell_price float,cost_price float,primary
key(product_no));
Query OK, 0 rows affected (0.77 sec)
mysql> insert into product_master
values('001','shampoo','1','one','4','2','10','15');
Query OK, 1 row affected (0.17 sec)
mysql> insert into product_master
```

```
values('002','oil','13','one','4','2','11','16');
Query OK, 1 row affected (0.06 sec)
mysql> alter table client_master add telephone_no int;
Query OK, 0 rows affected (1.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> select * from client_master;
+-----------+-------------+---------+-------+---------+-------
+---------+--------------+
| client_no | client_name | address | city | pincode | state
| bal_due | telephone_no |
+-----------+-------------+---------+-------+---------+-------
+---------+--------------+
| 1 | abhi | nasik | nasik | 422004 | MH
| 5000 | NULL |
| 2 | piyu | nasik | nasik | 422004 | MH
| 10000 | NULL |
| 3 | abd | nasik | nasik | 422003 | MH
| 5000 | NULL |
| 4 | abd | nasik | nasik | 422003 | MH
| 5000 | NULL |
| 5 | abc | nasik | nasik | 422003 | MH
| 5000 | NULL |
| 6 | xyz | nasik | nasik | 422004 | MH
| 6000 | NULL |
+-----------+-------------+---------+-------+---------+-------
+---------+--------------+
6 rows in set (0.00 sec)
mysql> select * from product_master;
+------------+-------------+------------+--------------
+----------+---------+------------+------------+
| product_no | description | profit_per | unit_measure |
quantity | reorder | sell_price | cost_price |
+------------+-------------+------------+--------------
+----------+---------+------------+------------+
| 1 | shampoo | 1 | one |
4 | 2 | 10 | 15 |
| 2 | oil | 13 | one |
4 | 2 | 11 | 16 |
+------------+-------------+------------+--------------
+----------+---------+------------+------------+
2 rows in set (0.00 sec)
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 table auto(roll_no int NOT NULL
AUTO_INCREMENT,name varchar(20),primary key(roll_no));
Query OK, 0 rows affected (0.36 sec)
mysql> select * from auto;
Empty set (0.01 sec)
mysql> insert into auto values('1','abc');
Query OK, 1 row affected (0.07 sec)
mysql> insert into auto values('2','adc');
Query OK, 1 row affected (0.08 sec)
mysql> alter table auto auto_increment=100;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> select * from auto;
+---------+------+
| roll_no | name |
+---------+------+
| 1 | abc |
| 2 | adc |
+---------+------+
2 rows in set (0.00 sec)
mysql> insert into auto values(null,'abd');
Query OK, 1 row affected (0.05 sec)
mysql> select * from auto;
+---------+------+
| roll_no | name |
+---------+------+
| 1 | abc |
| 2 | adc |
| 100 | abd |
+---------+------+
3 rows in set (0.00 sec)
mysql> insert into auto values(null,'reh');
Query OK, 1 row affected (0.06 sec)
mysql> select * from auto;
+---------+------+
| roll_no | name |
+---------+------+
| 1 | abc |
| 2 | adc |
| 100 | abd |
| 101 | reh |
+---------+------+
4 rows in set (0.00 sec)
mysql> update client_master set client_name="nut" where
```

```
client_no='4';
Query OK, 1 row affected (0.09 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> select * from client_master;
+-----------+-------------+---------+-------+---------+-------
+---------+--------------+
| client_no | client_name | address | city | pincode | state
| bal_due | telephone_no |
+-----------+-------------+---------+-------+---------+-------
+---------+--------------+
| 1 | abhi | nasik | nasik | 422004 | MH
| 5000 | NULL |
| 2 | piyu | nasik | nasik | 422004 | MH
| 10000 | NULL |
| 3 | abd | nasik | nasik | 422003 | MH
| 5000 | NULL |
| 4 | nut | nasik | nasik | 422003 | MH
| 5000 | NULL |
| 5 | abc | nasik | nasik | 422003 | MH
| 5000 | NULL |
| 6 | xyz | nasik | nasik | 422004 | MH
| 6000 | NULL |
+-----------+-------------+---------+-------+---------+-------
+---------+--------------+
6 rows in set (0.00 sec)
mysql> create index client_find on
client_master(client_name,city); Query OK, 0 rows
affected (0.41 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> show tables;
+---------------+
| Tables_in_Abhi |
+---------------+
| auto |
| client_master |
| product_master |
+---------------+
3 rows in set (0.08 sec)
mysql> select * from product_master;
+------------+-------------+------------+--------------
+----------+---------+------------+------------+
| product_no | description | profit_per | unit_measure |
quantity | reorder | sell_price | cost_price |
+------------+-------------+------------+--------------
```

```
+----------+---------+------------+------------+
| 1 | shampoo | 1 | one |
4 | 2 | 10 | 15 |
| 2 | oil | 13 | one |
4 | 2 | 11 | 16 |
+------------+-------------+------------+-------------
+----------+---------+------------+------------+
2 rows in set (0.00 sec)
mysql> desc product_master;
+--------------+-------------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| product_no | int(11) | NO | PRI | NULL | |
| description | varchar(20) | YES | | NULL | |
| profit_per | float | YES | | NULL | |
| unit_measure | varchar(10) | YES | | NULL | |
| quantity | int(11) | YES | | NULL | |
| reorder | int(11) | YES | | NULL | |
| sell_price | float | YES | | NULL | |
| cost_price | float | YES | | NULL | |
+--------------+-------------+------+-----+---------+-------+
8 rows in set (0.05 sec)
mysql> alter table client_master rename to c_master;
Query OK, 0 rows affected (0.25 sec)
mysql> insert into product_master
values('003','nutela','15','three','40','5','110','123');
Query OK, 1 row affected (0.05 sec)
mysql> alter table product_master modify sell_price
float(10,2);
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> desc product_master;
+--------------+-------------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| product_no | int(11) | NO | PRI | NULL | |
| description | varchar(20) | YES | | NULL | |
| profit_per | float | YES | | NULL | |
| unit_measure | varchar(10) | YES | | NULL | |
| quantity | int(11) | YES | | NULL | |
| reorder | int(11) | YES | | NULL | |
| sell_price | float(10,2) | YES | | NULL | |
| cost_price | float | YES | | NULL | |
+--------------+-------------+------+-----+---------+-------+
```

8 rows in set (0.00 sec)
mysql> create view client as select client_no,client_name from
c_master;
Query OK, 0 rows affected (0.05 sec)
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)
mysql>

TITLE: Design at least 10 SQL queries for suitable database application using SQL DML statements: Insert, Select, Update, Delete with operators, functions, and set operator.

```
mysql> show databases;
+--------------------+
| Database |
+--------------------+
| information_schema |
| A |
| Abhi |
| PVG |
| RENUKA |
| mysql |
| nishant |
| performance_schema |
| renuka |
| sys |
| time |
+--------------------+
11 rows in set (0.11 sec)
mysql> use Abhi;
Reading table information for completion of table and column
names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> create table Employee(emp_no int,emp_name
varchar(20),date date,position varchar(20));
Query OK, 0 rows affected (0.75 sec)
mysql> alter table Employee add salary int;
Query OK, 0 rows affected (0.68 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> insert into Employee values('01','abc','2018-07-
11','clerk','50000');
Query OK, 1 row affected (0.08 sec)
mysql> insert into Employee values('02','abhi','2018-05-
11','ceo','150000');
Query OK, 1 row affected (0.08 sec)
mysql> insert into Employee values('03','xyz','2018-05-
21','hr','100000');
Query OK, 1 row affected (0.04 sec)
mysql> insert into Employee values('04','aqwgy','2018-06-
```

21','te','10000');
Query OK, 1 row affected (0.03 sec)
mysql> insert into Employee values('05','sfhjfh','2018-07-
21','gt','12000');

Output:

Query OK, 1 row affected (0.03 sec)
mysql> create table TE(emp_no int,emp_name varchar(20),join_date date,position varchar(20),salary int);
Query OK, 0 rows affected (0.36 sec)
mysql> insert into TE values('01','abc','2018-07-11','clerk','50000');Query OK, 1 row affected (0.03 sec)
mysql> insert into TE values('02','abhi','2018-05-11','ceo','150000'); Query OK, 1 row affected (0.04 sec)
mysql> insert into TE values('03','xyz','2018-05-21','hr','100000'); Query OK, 1 row affected (0.04 sec)
mysql> insert into TE values('04','aqwgy','2018-06-21','te','10000'); Query OK, 1 row affected (0.05 sec)
mysql> insert into TE values('05','sfhjfh','2018-07-21','gt','12000'); Query OK, 1 row affected (0.04 sec)
mysql> select * from TE;

emp_no	emp_name	join_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 (0.04 sec)
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)
mysql> update TE set emp_name='gjgj' where emp_no='5';

```
Query OK, 1 row affected (0.13 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> select * from TE;
+--------+----------+------------+----------+--------+
| emp_no | emp_name | join_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)
mysql> select * from Employee union select * from TE;
+--------+----------+------------+----------+--------+
| 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 | gjgj | 2018-07-21 | gt | 12000 |
+--------+----------+------------+----------+--------+
6 rows in set (0.01 sec)
mysql> select * from Employee union all select * from TE;
+--------+----------+------------+----------+--------+
| 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 |
| 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 |
+--------+----------+------------+----------+--------+
10 rows in set (0.00 sec)
mysql> select distinct emp_no from Employee where emp_no
in(select emp_no from TE);
+--------+
```

```
| emp_no |
+--------+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
+--------+
5 rows in set (0.03 sec)
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)
mysql> select * from TE;
+--------+----------+------------+----------+--------+
| emp_no | emp_name | join_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)
mysql> select distinct emp_name from Employee where emp_name
in(select emp_name from TE);
+----------+
| emp_name |
+----------+
| abc |
| abhi |
| xyz |
| aqwgy |
+----------+
4 rows in set (0.00 sec)
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)
mysql> select * from TE;
+--------+----------+------------+----------+--------+
| emp_no | emp_name | join_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)
mysql> select distinct emp_name from Employee where emp_name
in(select emp_name from TE);
+----------+
| emp_name |
+----------+
| abc |
| abhi |
| xyz |
| aqwgy |
+----------+
4 rows in set (0.00 sec)
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)
mysql> select lcase(emp_no) from Employee;
+---------------+
| lcase(emp_no) |
+---------------+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
+---------------+
5 rows in set (0.00 sec)
mysql> select ucase(emp_no) from Employee;
+---------------+
| ucase(emp_no) |
+---------------+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
+---------------+
5 rows in set (0.00 sec)
```

```
mysql> select lcase(salary) from Employee;
+---------------+
| lcase(salary) |
+---------------+
| 50000 |
| 150000 |
| 100000 |
| 10000 |
| 12000 |
+---------------+
5 rows in set (0.00 sec)
mysql> select mid(emp_no,1,3) from Employee;
+-----------------+
| mid(emp_no,1,3) |
+-----------------+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
+-----------------+
5 rows in set (0.01 sec)
mysql> select mid(emp_no,1,3) from Employee;
+-----------------+
| mid(emp_no,1,3) |
+-----------------+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
+-----------------+
5 rows in set (0.00 sec)
mysql> select mid(emp_no,1,5) from Employee;
+-----------------+
| mid(emp_no,1,5) |
+-----------------+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
+-----------------+
5 rows in set (0.00 sec)
```

```
mysql> select mid(salary,1,3) from Employee;
+-----------------+
| mid(salary,1,3) |
+-----------------+
| 500 |
| 150 |
| 100 |
| 100 |
| 120 |
+-----------------+
5 rows in set (0.00 sec)
mysql> select mid(salary,1,5) from Employee;
+-----------------+
| mid(salary,1,5) |
+-----------------+
| 50000 |
| 15000 |
| 10000 |
| 10000 |
| 12000 |
+-----------------+
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)
mysql>
```

Assignment 3

TITLE: Design at least 10 SQL queries for suitable database application using SQL DML statements: all types of Join, Sub-Query and View.

```
mysql> show databases;
+--------------------+
| Database |
+--------------------+
| information_schema |
| A |
| Abhi |
| COMPUTER |
| H |
| PVG |
| RENUKA |
| mysql |
| nishant |
| nishantl |
| performance_schema |
| renuka |
| sys |
| time |
+--------------------+
14 rows in set (0.21 sec)
mysql> use Abhi;
Reading table information for completion of table and column
names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> show tables;
+----------------+
| Tables_in_Abhi |
+----------------+
| Employee |
| TE |
| auto |
| c_master |
| product_master |
+----------------+
5 rows in set (0.00 sec)
mysql> create table _master(product_no int,description
```

```
varchar(20),profit_per float,unit_measure varchar(10),quantity
int,reorder int,sell_price float,cost_price float,primary
key(product_no));
Query OK, 0 rows affected (0.55 sec)
mysql> create table customer(cust_no int,cust_name
varchar(20),cust_add varchar(20),phone_no int,primary
key(cust_no));
Query OK, 0 rows affected (0.28 sec)
mysql> create table capital(cap_no int,cap_name
varchar(20),state_no int,primary key(cap_no));
Query OK, 0 rows affected (0.27 sec)
mysql> create table state(state_no int,state_name
varchar(20),state_code int,capital varchar(20),primary
key(state_no));
Query OK, 0 rows affected (0.28 sec)
mysql> insert into capital values('01','MH','01');
Query OK, 1 row affected (0.12 sec)
mysql> insert into capital values('02','RAJ','02');
Query OK, 1 row affected (0.04 sec)
mysql> insert into capital values('03','GOA','03');
Query OK, 1 row affected (0.05 sec)
mysql> insert into capital values('04','GUJ','04');
Query OK, 1 row affected (0.05 sec)
mysql> insert into capital values('05','KAR','05');
Query OK, 1 row affected (0.04 sec)
mysql> insert into state values('01','MH','01','MUM');
Query OK, 1 row affected (0.03 sec)
mysql> insert into state values('02','RAJ','02','JAI');
Query OK, 1 row affected (0.03 sec)
mysql> insert into state values('03','GOA','03','PAN');
Query OK, 1 row affected (0.04 sec)
mysql> insert into state values('04','GUJ','04','SUR');
Query OK, 1 row affected (0.04 sec)
mysql> insert into state values('05','KAR','05','BAN');
```

Query OK, 1 row affected (0.03 sec)
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 |
+----------+------------+------------+---------+
5 rows in set (0.00 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 |
+--------+----------+
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
+--------+----------+
5 rows in set (0.06 sec)
mysql> UPDATE state SET state_no="78" where state_no='1';
Query OK, 1 row affected (0.04 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> UPDATE state SET state_no="58" where state_no='2';
Query OK, 1 row affected (0.04 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> UPDATE state SET state_no="46" where state_no='3';
Query OK, 1 row affected (0.03 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> UPDATE state SET state_no="489" where state_no='4';
Query OK, 1 row affected (0.05 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> UPDATE state SET state_no="458" where state_no='5';
Query OK, 1 row affected (0.03 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> insert into state values('05','MP','05','BHO');
Query OK, 1 row affected (0.03 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)
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 left
join state on capital.cap_no=state.state_name;
+--------+----------+
| cap_no | state_no |
+--------+----------+
| 1 | NULL |
| 2 | NULL |
| 3 | NULL |
| 4 | NULL |
| 5 | NULL |
+--------+----------+
5 rows in set, 20 warnings (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)
```

```
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 (0.00 sec)
mysql> select * from state;
+----------+------------+------------+---------+
| state_no | state_name | state_code | capital |
+----------+------------+------------+---------+
| 5 | MP | 5 | BHO |
| 46 | GOA | 3 | PAN |
| 58 | RAJ | 2 | JAI |
| 78 | MH | 1 | MUM |
| 458 | KAR | 5 | BAN |
| 489 | GUJ | 4 | SUR |
+----------+------------+------------+---------+
6 rows in set (0.00 sec)
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 (0.00 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)
mysql> select
```

```
capital.cap_no,capital.cap_name,state.capital,state.state_no
from capital inner join state on
capital.cap_no=state.state_no;
+--------+----------+---------+----------+
| cap_no | cap_name | capital | state_no |
+--------+----------+---------+----------+
| 5 | KAR | BHO | 5 |
+--------+----------+---------+----------+
1 row in set (0.00 sec)
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;
+--------+----------+---------+----------+
| 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 |
+--------+----------+---------+----------+
5 rows in set (0.00 sec)
mysql> select
capital.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 |
+--------+----------+---------+----------+
| 5 | KAR | BHO | 5 |
| NULL | NULL | PAN | 46 |
| NULL | NULL | JAI | 58 |
| NULL | NULL | MUM | 78 |
| NULL | NULL | BAN | 458 |
| NULL | NULL | SUR | 489 |
+--------+----------+---------+----------+
6 rows in set (0.00 sec)
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 select
capital.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)
mysql> select * from capital c1, state s1 where
c1.cap_no=s1.state_no;
+--------+----------+----------+----------+------------
+------------+---------+
| cap_no | cap_name | state_no | state_no | state_name |
state_code | capital |
+--------+----------+----------+----------+------------
+------------+---------+
| 5 | KAR | 5 | 5 | MP |
5 | BHO |
+--------+----------+----------+----------+------------
+------------+---------+
1 row in set (0.00 sec)
mysql> select * from capital c1, state s1 where c1.cap_no!
=s1.state_no;
+--------+----------+----------+----------+------------
+------------+---------+
| cap_no | cap_name | state_no | state_no | state_name |
state_code | capital |
+--------+----------+----------+----------+------------
+------------+---------+
| 1 | MH | 1 | 5 | MP |
5 | BHO |
| 2 | RAJ | 2 | 5 | MP |
5 | BHO |
| 3 | GOA | 3 | 5 | MP |
5 | BHO |
| 4 | GUJ | 4 | 5 | MP |
5 | BHO |
```

| 1 | MH | 1 | 46 | GOA |
3 | PAN |
| 2 | RAJ | 2 | 46 | GOA |
3 | PAN |
| 3 | GOA | 3 | 46 | GOA |
3 | PAN |
| 4 | GUJ | 4 | 46 | GOA |
3 | PAN |
| 5 | KAR | 5 | 46 | GOA |
3 | PAN |
| 1 | MH | 1 | 58 | RAJ |
2 | JAI |
| 2 | RAJ | 2 | 58 | RAJ |
2 | JAI |
| 3 | GOA | 3 | 58 | RAJ |
2 | JAI |
| 4 | GUJ | 4 | 58 | RAJ |
2 | JAI |
| 5 | KAR | 5 | 58 | RAJ |
2 | JAI |
| 1 | MH | 1 | 78 | MH |
1 | MUM |
| 2 | RAJ | 2 | 78 | MH |
1 | MUM |
| 3 | GOA | 3 | 78 | MH |
1 | MUM |
| 4 | GUJ | 4 | 78 | MH |
1 | MUM |
| 5 | KAR | 5 | 78 | MH |
1 | MUM |
| 1 | MH | 1 | 458 | KAR |
5 | BAN |
| 2 | RAJ | 2 | 458 | KAR |
5 | BAN |
| 3 | GOA | 3 | 458 | KAR |
5 | BAN |
| 4 | GUJ | 4 | 458 | KAR |
5 | BAN |
| 5 | KAR | 5 | 458 | KAR |
5 | BAN |
| 1 | MH | 1 | 489 | GUJ |
4 | SUR |
| 2 | RAJ | 2 | 489 | GUJ |
4 | SUR |

```
| 3 | GOA | 3 | 489 | GUJ |
4 | SUR |
| 4 | GUJ | 4 | 489 | GUJ |
4 | SUR |
| 5 | KAR | 5 | 489 | GUJ |
4 | SUR |
+--------+----------+----------+----------+------------
+------------+---------+
29 rows in set (0.00 sec)
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='MH');
Empty set (0.00 sec)
mysql> select * from state where state_no=(select
capital.state_no from capital where cap_name='GUJ');
Empty set (0.00 sec)
mysql> select * from state where state_no=(select
capital.state_no from capital where cap_name='RAJ');
Empty 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)
mysql>
```

Assignment 4

TITLE: Unnamed PL/SQL code block: Use of Control structure and Exception handling is mandatory. Write a PL/SQL block of code for the following requirements:-

Schema:

1. Borrower(Rollin, Name, DateofIssue, NameofBook, Status)
2. Fine(Roll_no,Date,Amt)

· Accept roll_no & name of book from user.

· Check the number of days (from date of issue), if days are between 15 to 30 then fine amount will be Rs 5per day.

· If no. of days>30, per day fine will be Rs 50 per day & for days less than 30, Rs. 5 per day.

· After submitting the book, status will change from I to R.

· If condition of fine is true, then details will be stored into fine table.

Frame the problem statement for writing PL/SQL block inline with above statement.

```
mysql> use Abhi;
Reading table information for completion of table and column
names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> delimiter //
mysql> call B1(1,'TOC')


1 row in set (0.35 sec)
Query OK, 0 rows affected (0.41 sec)
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)
mysql> show tables;
-> //
+----------------+
| Tables_in_Abhi |
+----------------+
| Borrower |
| Employee |
| Fine |
| TE |
| _master |
| auto |
| c_master |
| capital |
| customer |
| orders |
| person |
| product_master |
| state |
+----------------+
13 rows in set (0.00 sec)
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)
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)
mysql>
```

Assignment 5

TITLE: PL/SQL Stored Procedure and Stored Function.

Write a Stored Procedure namely proc_Grade for the categorization of student. If marks scored by students in examination is <=1500 and marks>=990 then student will be placed in distinction category if marks scored are between 989 and900 category is first class, if marks 899 and 825 category is Higher Second Class. Write a PL/SQL block for using procedure created with above requirement.

Stud_Marks(name, total_marks)

Result(Roll,Name, Class)

Frame the separate problem statement for writing PL/SQL Stored Procedure and function, inline with above statement. The problem statement should clearly state the requirements.

```
mysql> use Abhi;
Reading table information for completion of table and column
names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> create table marks(roll_no int,name
varchar(20),total_marks varchar(20));
Query OK, 0 rows affected (0.67 sec)
mysql> create table result(roll_no int,name varchar(20),class
varchar(20));
Query OK, 0 rows affected (0.41 sec)
insert into marks values('1','Abhi',1400')' at line 1
mysql> insert into marks values('1','Abhi','1400');Query OK, 1
row affected (0.04 sec)
mysql> insert into marks values('2','piyush','980');
Query OK, 1 row affected (0.08 sec)
mysql> insert into marks values('3','hitesh','880');
Query OK, 1 row affected (0.08 sec)
mysql> insert into marks values('4','ashley','820');
Query OK, 1 row affected (0.08 sec)
mysql> insert into marks values('5','partik','740');
Query OK, 1 row affected (0.03 sec)
mysql> insert into marks values('6','patil','640');
```

```
Query OK, 1 row affected (0.08 sec)
mysql> delimiter //
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)
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;
-> //
```

```
Query OK, 0 rows affected (0.00 sec)
mysql> select final_result3(2);
-> //
+------------------+
| final_result3(2) |
+------------------+
| 2 |
+------------------+
1 row in set (0.05 sec)
mysql> select final_result3(3);//
+------------------+
| final_result3(3) |
+------------------+
| 3 |
+------------------+
1 row in set (0.04 sec)
mysql> select final_result3(4);//
+------------------+
| final_result3(4) |
+------------------+
| 4 |
+------------------+
1 row in set (0.12 sec)
mysql> select final_result3(5);//
+------------------+
| final_result3(5) |
+------------------+
| 5 |
+------------------+
1 row in set (0.05 sec)
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)
```

Assignment 6

TITLE: Cursors: (All types: Implicit, Explicit, Cursor FOR Loop, Parameterized Cursor) Write a PL/SQL block of code using parameterized Cursor, that will merge the data available in the newly created table N_RollCall with the data available in the table O_RollCall. If the data in the first table already exist in the second table then that data should be skipped.Frame the separate problem statement for writing PL/SQL block to implement all types of Cursors inline with above statement. The problem statement should clearly state the requirements.

```
mysql> use Abhi;
Reading table information for completion of table and column
names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> create table o_rollcall(roll_no int,name
varchar(20),address varchar(20));
Query OK, 0 rows affected (0.28 sec)
mysql> create table n_rollcall(roll_no int,name
varchar(20),address varchar(20));
Query OK, 0 rows affected (0.27 sec)
mysql> insert into o_rollcall('1','Hitesh','Nandura');
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 ''1','Hitesh','Nandura')' at
line 1
mysql> insert into o_rollcall values('1','Hitesh','Nandura');
Query OK, 1 row affected (0.05 sec)
mysql> insert into o_rollcall values('2','Piyush','MP');
Query OK, 1 row affected (0.06 sec)
mysql> insert into o_rollcall values('3','Ashley','Nsk');
Query OK, 1 row affected (0.05 sec)
mysql> insert into o_rollcall values('4','Kalpesh','Dhule');
Query OK, 1 row affected (0.05 sec)
mysql> insert into o_rollcall values('5','Abhi','Satara');
Query OK, 1 row affected (0.04 sec)
mysql> delimiter //
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 where
roll_no=r2;
-> end if;
-> if exit_loop
-> then
-> close c1;
-> leave e_loop;
-> end if;
-> end loop e_loop;
-> end
-> //
Query OK, 0 rows affected (0.00 sec)
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)
mysql> call p3(0);
-> //
Query OK, 0 rows affected (0.22 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 |
+---------+---------+---------+
5 rows in set (0.00 sec)
mysql> insert into o_rollcall values('6','Patil','Kolhapur');
-> //
Query OK, 1 row affected (0.04 sec)
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)
mysql>
```

Assignment 7

TITLE: Database Trigger (All Types: Row level and Statement level triggers, Before and After Triggers). Write a database trigger on Library table. The System should keep track of the records that are being updated or deleted. The old value of updated or deleted records should be added in Library_Audit table.Frame the problem statement for writing Database Triggers of all types, in-line with above statement. The problem statement should clearly state the requirements.

```
mysql> use info;
Reading table information for completion of table and column
names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> create table borrower2(roll_no int,name
varchar(20),date_of_issue date,book
Query OK, 0 rows affected (0.44 sec)
mysql> insert into borrower2 values('1','nick','2018-06-
10','wings_of_fire','avaliable','APJ');
Query OK, 1 row affected (0.07 sec)
mysql> insert into borrower2 values('2','mira','2018-05-
11','leaves_life','not_avaliable','borwarkar');
Query OK, 1 row affected (0.05 sec)
mysql> insert into borrower2 values('3','rina','2018-02-
12','unusal','avaliable','johar');
Query OK, 1 row affected (0.04 sec)
mysql> insert into borrower2 values('4','harsha','2018-06-
20','skylimit','avaliable','ingale');
Query OK, 1 row affected (0.05 sec)
mysql> insert into borrower2 values('5','tej','2018-04-
20','highway','not_avaliable','klm');
Query OK, 1 row affected (0.05 sec)
mysql> select *from borrower1;
+---------+--------+--------------+--------------
+---------------+-----------+
| roll_no | name | date_of_issue | book_name | status
| author |
+---------+--------+--------------+--------------
+---------------+-----------+
```

```
| 1 | nick | 2018-06-10 | wings_of_fire | avaliable
| APJ |
| 2 | mira | 2018-05-11 | leaves_life |
not_avaliable | borwarkar |
| 3 | rina | 2018-02-12 | unusal | avaliable
| johar |
| 4 | harsha | 2018-06-20 | skylimit | avaliable
| ingale |
| 5 | tej | 2018-04-20 | highway |
not_avaliable | klm |
+---------+--------+---------------+---------------
+---------------+-----------+
5 rows in set (0.00 sec)
//INSERT 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)
mysql> select * from borrower1;
-> //
+---------+--------+---------------+---------------
+---------------+-----------+
| roll_no | name | date_of_issue | book_name | status
| author |
+---------+--------+---------------+---------------
+---------------+-----------+
| 1 | nick | 2018-06-10 | wings_of_fire | avaliable
| APJ |
| 2 | mira | 2018-05-11 | leaves_life |
not_avaliable | borwarkar |
| 3 | rina | 2018-02-12 | unusal | avaliable
| johar |
| 4 | harsha | 2018-06-20 | skylimit | avaliable
| ingale |
```

```
| 5 | tej | 2018-04-20 | highway |
not_avaliable | klm |
| 6 | xyz | 2018-09-06 | aaa | avaliable
| xxx |
+---------+--------+---------------+---------------



+---------------+-----------+
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 | aaa | avaliable | xxx
| 2018-08-29 15:46:13 |
+---------+------+---------------+-----------+-----------
+--------+--------------------+
1 row in set (0.00 sec)
// UPDATE TRIGGER
mysql> delimiter //
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;
-> //
Query OK, 0 rows affected (0.08 sec)
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 | date_of_issue | book_name | status
```

| author |
+---------+--------+---------------+---------------
+---------------+-----------+
| 1 | nick | 2018-06-10 | wings_of_fire | avaliable
| APJ |
| 2 | mira | 2018-05-11 | leaves_life |
not_avaliable | borwarkar |
| 3 | rina | 2018-02-12 | unusal | avaliable

| johar |
| 4 | harsha | 2018-06-20 | skylimit | avaliable
| ingale |
| 5 | tej | 2018-04-20 | highway |
not_avaliable | klm |
| 8 | xyz | 2018-09-06 | leaf | avaliable
| xxx |
+---------+--------+---------------+---------------
+---------------+-----------+
6 rows in set (0.00 sec)

Assignment 8

Title of Assignment: Database Connectivity:

Write a program to implement MySQL/Oracle database connectivity with any front end language to implement Database navigation operations (add, delete, edit etc.)

```
import java.awt.*;
import java.awt.event.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.*;
public class student extends JFrame implements
ActionListener{
JFrame f;
JLabel l1, l2,l3,l4;
JTextField t1, t2,t3;
JButton b1, b2, b3, b4, b5;
Connection c;
Statement s;
ResultSet r;
student ()
{try{
f=new JFrame("Student Form");
f.setLayout(null);f.setVisible(true);
f.setSize(700, 500);
l4=new JLabel("Student Management System");
//l4.setBounds(100,01,250,250);
l4.setBounds(100, 30, 400, 30);
f.add(l4);
l4.setForeground(Color.blue);
l4.setFont(new Font("Serif", Font.BOLD,
30));
l1=new JLabel("Stud_RollNo");
l1.setBounds(50, 70, 100, 50);
f.add(l1);
l2=new JLabel("Stud_Name");
l2.setBounds(50, 120, 100, 50);
f.add(l2);
l3=new JLabel("Stud_Dept");
l3.setBounds(50, 170, 100, 50);
f.add(l3);
t1=new JTextField();
```

```
t1.setBounds(150, 90, 100, 30);
f.add(t1);
t2=new JTextField();
t2.setBounds(150, 140, 100, 30);
f.add(t2);t3=new JTextField();
t3.setBounds(150, 190, 100, 30);
f.add(t3);
b1= new JButton("ADD");
b1.setBounds(200, 300, 75, 50);
f.add(b1);
b1.addActionListener(this);
b2= new JButton("EDIT");
b2.setBounds(300, 300, 75, 50);
f.add(b2);
b2.addActionListener(this);
b3= new JButton("DELETE");b3.setBounds(400, 300, 75, 50);
f.add(b3);
b3.addActionListener(this);
b5= new JButton("EXIT");
b5.setBounds(500, 300, 75, 50);
f.add(b5);
b5.addActionListener(this);
Class.forName("com.mysql.jdbc.Driver");
c=DriverManager.getConnection("jdbc:mysql://loca
lhost:3306/info","root","root");
s=c.createStatement();
}catch(Exception e){System.out.println(e);}
}//ends INS Constructor
public void actionPerformed(ActionEvent ae){
try{
if(ae.getSource()==b1){String s1="INSERT
INTO result(stud_RollNo,stud_Name,stud_Dept)
VALUES("+t1.getText()+",'"+t2.getText()
+"','"+t3.getText() + "')";
System.out.println(s1);
s.executeUpdate(s1);
r=s.executeQuery("SELECT * FROM result");
t1.setText("");
t2.setText("");
t3.setText("");
}else if(ae.getSource()==b2){
String s2="UPDATE user1 SET
stud_Name='"+t2.getText()+"' WHERE
stud_RollNo="+t1.getText();
```

```
System.out.println(s2);
s.executeUpdate(s2);
r=s.executeQuery("SELECT * FROM result");
t1.setText("");
t2.setText("");t3.setText("");
}else if(ae.getSource()==b3){
String s3="DELETE FROM result WHERE
stud_RollNo="+t1.getText();
System.out.println(s3);
s.executeUpdate(s3);
r=s.executeQuery("SELECT * FROM result");
t1.setText("");
t2.setText("");
t3.setText("");}else if(ae.getSource()==b5){
System.exit(0); }
}catch(Exception e){System.out.println(e);}
}
public static void main(String args[]){
new student();
}
}
```

---------- **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 of
their 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_RollNo
int,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
|
+-------------+-----------+-----------+
3 rows in set (0.00 sec)

Assignment 9

Title of Assignment: MongoDB Queries:

Design and Develop MongoDB Queries using CRUD operations. (Use CRUD operations, SAVE method, logical operators etc.).

```
sl1-pc6@sl1pc6-HP-dx2480-MT-VP562PA:~$ mongo
MongoDB shell version: 2.6.10
connecting to: test
Server has startup warnings:
2018-09-19T10:13:21.731+0530 [initandlisten]
2018-09-19T10:13:21.731+0530 [initandlisten] ** NOTE: This is
a 32 bit MongoDB binary.
2018-09-19T10:13:21.731+0530 [initandlisten] ** 32 bit
builds are limited to less than 2GB of data (or less with
--journal).
2018-09-19T10:13:21.731+0530 [initandlisten] ** See
http://dochub.mongodb.org/core/32bit
2018-09-19T10:13:21.731+0530 [initandlisten]
> use Abhi;
switched to db Abhi
> db.createCollection('Student');
{ "ok" : 1 }
> 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 })
> 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" }
> 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"
}
> 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"
}
>
db.Student.save({_id:ObjectId("5b8fad4ef00832a0a50b5036"),"RNO
":"1","NAME":"PIYUSH","CLASS":"TE COMP","ADD":"MP"});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1
})
> db.Student.find().pretty();
{
"_id" : ObjectId("5b8fad4ef00832a0a50b5036"),
"RNO" : "1",
"NAME" : "PIYUSH",
"CLASS" : "TE COMP",
```

```
"ADD" : "MP"
}
{
"_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.find({$and:[{"Name":"Piyush"},{"Rno":"2"}]});
> db.Student.find({$and:[{"Name":"Piyush"},
{"Rno":"1"}]}).pretty();
{
"_id" : ObjectId("5ba1d618f5bbacd4ad81568d"),
"Rno" : "1",
"Name" : "Piyush",
"Class" : "TE COMP"
}
> db.Student.find({$and:[{"Name":"Piyush"},
{"Rno":"2"}]}).pretty();
> db.Student.find({$or:[{"Name":"Piyush"},
```

{"Rno":"2"}]}).pretty();
{
"_id" : ObjectId("5ba1d618f5bbacd4ad81568d"),
"Rno" : "1",
"Name" : "Piyush",
"Class" : "TE COMP"
}
{
"_id" : ObjectId("5ba1d625f5bbacd4ad81568e"),
"Rno" : "2",
"Name" : "Abhi",
"Class" : "TE COMP"
}
> db.Student.find({$or:[{"Name":"Piyush"},{"Class":"TE
COMP"}]}).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"
}
> db.Student.find({$nor:[{"Name":"Piyush"},{"Class":"TE
COMP"}]}).pretty();
> db.Student.find({$nor:[{"Name":"Piyush"},
{"Rno":"2"}]}).pretty();
{
"_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"
}
db.Student.find( {"Rno": { $not:{$lt:"3"}}}).pretty();
{
"_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"
}
> db.Student.find( {"Rno": { $eq:"5"}}).pretty();
{
"_id" : ObjectId("5ba1d65ef5bbacd4ad815691"),
"Rno" : "5",
"Name" : "Pratik",
"Class" : "TE COMP"
}
> db.Student.find( {"Rno": { $ne:"5"}}).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("5ba1d66df5bbacd4ad815692"),
"Rno" : "6",
"Name" : "Pratik",
"Class" : "TE COMP"
}
> db.Student.find( {"Rno": { $gt:"5"}}).pretty();
{
"_id" : ObjectId("5ba1d66df5bbacd4ad815692"),
"Rno" : "6",
"Name" : "Pratik",
"Class" : "TE COMP"
}
> db.Student.find( {"Rno": { $gte:"5"}}).pretty();
{
"_id" : ObjectId("5ba1d65ef5bbacd4ad815691"),
"Rno" : "5",
"Name" : "Pratik",
"Class" : "TE COMP"
}
{
"_id" : ObjectId("5ba1d66df5bbacd4ad815692"),
"Rno" : "6",
"Name" : "Pratik",
"Class" : "TE COMP"
}
> db.Student.find( {"Rno": { $lt:"5"}}).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"
}
> db.Student.find( {"Rno": { $lte:"5"}}).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"
}
> db.Student.find( {"Rno": { $lt:"5",$gt:"2"}}).pretty();
{
```

```
"_id" : ObjectId("5ba1d63af5bbacd4ad81568f"),
"Rno" : "3",
"Name" : "Ashley",
"Class" : "TE COMP"
}
{
"_id" : ObjectId("5ba1d647f5bbacd4ad815690"),
"Rno" : "4",
"Name" : "Hitesh",
"Class" : "TE COMP"
}
> db.Student.find( {"Rno": { $lte:"5",$gte:"2"}}).pretty();
{
"_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"
}
> db.Student.find( {"Rno": { $lte:"5",$gt:"2"}}).pretty();
{
"_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"
}
> db.Student.find( {"Rno": { $lt:"5",$gte:"2"}}).pretty();
{
"_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"
}
```

Assignment 10

Title of Assignment: MongoDB – Aggregation and Indexing:

Design and Develop MongoDB Queries using aggregation and indexing with suitable example using MongoDB.

```
//USE MONGODB
sl1-pc5@sl1pc5-dx2480-MT:~$ mongo
MongoDB shell version: 2.6.10
connecting to: test
Server has startup warnings:
2018-09-20T13:00:01.599+0530 [initandlisten]
2018-09-20T13:00:01.599+0530 [initandlisten] ** NOTE: This is
a 32 bit MongoDB binary.
2018-09-20T13:00:01.599+0530 [initandlisten] **
32 bit builds are limited to less than 2GB
of data (or less with --journal).
2018-09-20T13:00:01.599+0530 [initandlisten] **
See http://dochub.mongodb.org/core/32bit
2018-09-20T13:00:01.599+0530 [initandlisten]
//USE DATABASE
> use comp;
switched to db comp
//CREATE COLLECTION WEBSITE
> db.createCollection('website');
{ "ok" : 1 }
//INSERT VALUES IN WEBSITE
>
db.website.insert({'roll':'1','name':'harsh','amount':1000,'ur
l':'www.yahoo.com'});
WriteResult({ "nInserted" : 1 })
>
db.website.insert({'roll':'2','name':'jitesh','amount':2000,'u
rl':'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 ] }
//COUNT AGGREGATION
> db.website.aggregate({$group:{_id:"$name","total":
{$sum:1}}});
{ "_id" : "ash", "total" : 2 }
{ "_id" : "rina", "total" : 1 }
{ "_id" : "jitesh", "total" : 1 }
{ "_id" : "harsh", "total" : 2 }
//ADDTOSET AGGREGATE
> db.website.aggregate({$group:{_id:"$name","total":
{$addToSet:"$amount"}}});
{ "_id" : "ash", "total" : [ 1000, 4000 ] }
{ "_id" : "rina", "total" : [ 3000 ] }
{ "_id" : "jitesh", "total" : [ 2000 ] }
{ "_id" : "harsh", "total" : [ 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 }//CREATE INDEXING
> db.website1.createIndex({'name':-1})
{
"createdCollectionAutomatically" : false,
"numIndexesBefore" : 2,
"numIndexesAfter" : 3,
"ok" : 1
}
> db.website1.getIndexses()
2018-09-20T13:28:09.628+0530 TypeError: Property 'getIndexses'
of object om.website is not a
function
```

```
> 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.getIndexes()
[
```

```
{
"v" : 1,
"key" : {
"_id" : 1
},
"name" : "_id_",
"ns" : "harsh.website1"
}
]
> db.website1.find().pretty()
{ "_id" : ObjectId("5ba3509a444926329738012d"), "roll" : 1,
"name" : "harsh" }
{ "_id" : ObjectId("5ba35293444926329738012e"), "roll" : 1,
"name" : "harsh" }
>
> db.website1.createIndex({'name':1})
{
"createdCollectionAutomatically" : false,
"numIndexesBefore" : 1,
"numIndexesAfter" : 2,
"ok" : 1
}
> db.website1.getIndexes()
[
{
"v" : 1,
"key" : {
"_id" : 1
},
"name" : "_id_",
"ns" : "harsh.website1"
},
{
"v" : 1,
"key" : {"name" : 1
},
"name" : "name_1",
"ns" : "harsh.website1"
}
]
> db.website1.dropIndex({'name':1})
{ "nIndexesWas" : 2, "ok" : 1 }
> db.website1.getIndexes()
[
```

```
{
"v" : 1,
"key" : {
"_id" : 1
},
"name" : "_id_",
"ns" : "harsh.website1"
}
]
> db.website1.createIndex({'name':1,'r':-1})
{
"createdCollectionAutomatically" : false,
"numIndexesBefore" : 1,
"numIndexesAfter" : 2,
"ok" : 1
}
> db.website1.getIndexes()
[
{
"v" : 1,
"key" : {
"_id" : 1
},



"name" : "_id_",
"ns" : "harsh.website1"
},
{
"v" : 1,
"key" : {
"name" : 1,
"r" : -1
},
"name" : "name_1_r_-1",
"ns" : "harsh.website1"
}
] (i-search)`db.website1.insert({'roll':1,'name':'harsh'});':
```

Assignment 11

```
sl1-pc6@sl1pc6-HP-dx2480-MT-VP562PA:~$ mongo
MongoDB shell version: 2.6.10
connecting to: test
Server has startup warnings:
2018-09-26T11:05:29.854+0530 [initandlisten]
2018-09-26T11:05:29.854+0530 [initandlisten] ** NOTE: This is
a 32 bit MongoDB binary.
2018-09-26T11:05:29.855+0530 [initandlisten] ** 32 bit
builds are limited to less than 2GB of data (or less with
--journal).
2018-09-26T11:05:29.855+0530 [initandlisten] ** See
http://dochub.mongodb.org/core/32bit
2018-09-26T11:05:29.855+0530 [initandlisten]
> use Abhi
switched to db Abhi
> db.createCollection('Journal');
{ "ok" : 1 }
> db.Journal.insert({'book_id':1,'book_name':'Javacd
OOP','amt':500,'status':'Available'});
WriteResult({ "nInserted" : 1 })
> db.Journal.insert({'book_id':1,'book_name':'Java
OOP','amt':400,'status':'Not Available'});
WriteResult({ "nInserted" : 1 })
>
db.Journal.insert({'book_id':1,'book_name':'Java','amt':300,'s
tatus':'Not Available'});
WriteResult({ "nInserted" : 1 })
>
db.Journal.insert({'book_id':2,'book_name':'Java','amt':300,'s
tatus':'Available'});
WriteResult({ "nInserted" : 1 })
>
db.Journal.insert({'book_id':2,'book_name':'OPP','amt':200,'st
atus':'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 })
>
>
>
> 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
}
```

```
> 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 }
>
```

Assignment 12

```
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