

NAME:NEELAM H GODIHAL

USN:1BM19CS220

### **PROGRAM 1: INSURANCE DATABASE**

Consider the Insurance database given below. The data types are specified.

PERSON (driver\_id: String, name: String, address: String)

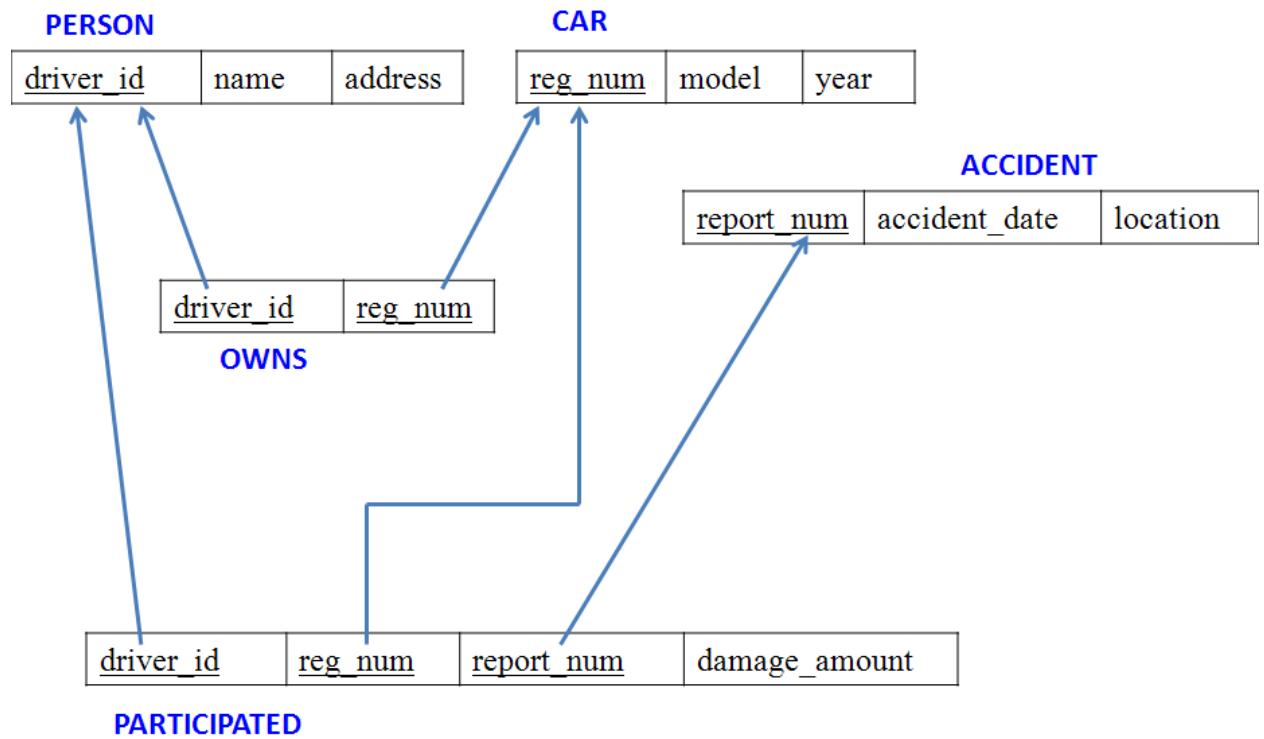
CAR (reg\_num: String, model: String, year: int)

ACCIDENT (report\_num: int, accident\_date: date, location: String)

OWNS (driver\_id: String, reg\_num: String)

PARTICIPATED (driver\_id: String, reg\_num: String, report\_num: int, damage\_amount: int)

#### **Schema diagram**



#### **Tables**

**PERSON**

<u>driver_id</u>	name	address
A01	Richard	Srinivas nagar
A02	Pradeep	Rajaji nagar
A03	Smith	Ashok nagar
A04	Venu	N R Colony
A05	Jhon	Hanumanth nagar

**CAR**

<u>reg_num</u>	model	year
KA052250	Indica	1990
KA031181	Lancer	1957
KA095477	Toyota	1998
KA053408	Honda	2008
KA041702	Audi	2005

**OWNS**

<u>driver_id</u>	<u>reg_num</u>
A01	KA052250
A02	KA053408
A03	KA031181
A04	KA095477
A05	KA041702

**ACCIDENT**

<u>report_num</u>	<u>accident_date</u>	<u>location</u>
11	01-JAN-03	Mysore Road
12	02-FEB-04	South end Circle
13	21-JAN-03	Bull temple Road
14	17-FEB-08	Mysore Road
15	04-MAR-05	Kanakpura Road

**PARTICIPATED**

<u>driver_id</u>	<u>reg_num</u>	<u>report_num</u>	<u>damage_amount</u>
A01	KA052250	11	10000
A02	KA053408	12	50000
A03	KA095477	13	25000
A04	KA031181	14	3000
A05	KA041702	15	5000

**1.Create the above tables by properly specifying the primary keys and the foreign keys.**  
create database insurance1;

use insurance1;

```
create table person(
    driver_id varchar(10) NOT NULL,
    name varchar(30),
    address varchar(40),
    primary key (driver_id)
);
desc person;
```

MySQL Workbench

Local instance MySQL80

**Query 1:**

```

146 • create table person(
147     driver_id varchar(10) NOT NULL,
148     name varchar(30),
149     address varchar(40),
150     primary key (driver_id)
151 );
152 • desc person;

```

**Result Grid:**

Field	Type	Null	Key	Default	Extra
driver_id	varchar(10)	NO	PRI		
name	varchar(30)	YES			
address	varchar(40)	YES			

**Result 3:**

#	Time	Action	Message	Duration / Fetch
1	09:41:08	show databases	6 row(s) returned	0.000 sec / 0.000 sec
2	09:42:00	create database insurance	1 row(s) affected	0.421 sec
3	09:42:09	show databases	7 row(s) returned	0.000 sec / 0.000 sec
4	09:42:26	use insurance	0 row(s) affected	0.000 sec
5	09:45:01	create table person(driver_id varchar(10) NOT NULL, name varchar(30), address varchar(40), primary key (driver_id))	0 row(s) affected	0.766 sec
6	09:45:34	desc person	3 row(s) returned	0.000 sec / 0.000 sec

Query Completed

```

create table car(
    reg_num varchar(10) NOT NULL,
    model varchar(10),
    year int,
    primary key(reg_num)
);
desc car;

```

MySQL Workbench

Local instance MySQL80

**Query 1:**

```

154 • create table car(
155     reg_num varchar(10) NOT NULL,
156     model varchar(10),
157     year int,
158     primary key(reg_num)
159 );
160 • desc car;

```

**Result Grid:**

Field	Type	Null	Key	Default	Extra
reg_num	varchar(10)	NO	PRI		
model	varchar(10)	YES			
year	int	YES			

**Result 4:**

#	Time	Action	Message	Duration / Fetch
3	09:42:09	show databases	7 row(s) returned	0.000 sec / 0.000 sec
4	09:42:26	use insurance	0 row(s) affected	0.000 sec
5	09:45:01	create table person(driver_id varchar(10) NOT NULL, name varchar(30), address varchar(40), primary key (driver_id))	0 row(s) affected	0.766 sec
6	09:45:34	desc person	3 row(s) returned	0.000 sec / 0.000 sec
7	09:48:34	create table car(reg_num varchar(10) NOT NULL, model varchar(10), year int, primary key(reg_num))	0 row(s) affected	0.032 sec
8	09:48:49	desc car	3 row(s) returned	0.375 sec / 0.000 sec

Query Completed

```

create table accident(
    report_num int NOT NULL,
    accident_date date,

```

```

        location varchar(20),
        primary key(report_num)
);
desc accident;

```

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database structure with 'MANAGEMENT', 'INSTANCE', and 'PERFORMANCE' sections.
- Query Editor (Query 1):** Displays the SQL code for creating the 'accident' table and describing it. The table has columns: report\_num (int, NOT NULL, primary key), accident\_date (date), and location (varchar(20)).
- Result Grid:** Shows the structure of the 'accident' table with columns: Field, Type, Null, Key, Default, and Extra. The primary key is 'report\_num'.
- Output (Result 5):** Shows the history of actions taken during the session, including the creation of the 'person', 'car', and 'accident' tables, and their descriptions. It includes columns: Action Output, Message, and Duration / Fetch.
- System Tray:** Shows battery level (69%), network status, and system date/time (26-04-2021 09:52 AM).

```

create table owns(
        driver_id varchar(10) NOT NULL,
        reg_num varchar(10) NOT NULL,
        primary key(driver_id,reg_num),
        foreign key(driver_id) REFERENCES person(driver_id),
        foreign key(reg_num) REFERENCES car(reg_num)
);
desc owns;

```

MySQL Workbench Screenshot showing the creation of the 'participated' table and its description.

```

171     driver_id varchar(10) NOT NULL,
172     reg_num varchar(10) NOT NULL,
173     primary key(driver_id,reg_num),
174     foreign key(driver_id) REFERENCES person(driver_id),
175     foreign key(reg_num) REFERENCES car(reg_num)
176   );
177 * desc participated;

```

The 'Result Grid' shows the table structure:

Field	Type	Null	Key	Default	Extra
driver_id	varchar(10)	NO	PRI		
reg_num	varchar(10)	NO	PRI		

The 'Result 6' tab shows the history of operations:

#	Time	Action	Message	Duration / Fetch
8	09:48:49	desc car	3 row(s) returned	0.375 sec / 0.000 sec
9	09:52:13	create table accident(report_num int NOT NULL, accident_date date, location varchar(20), primary key(report_num))	0 row(s) affected	0.407 sec
10	09:52:26	desc accident	3 row(s) returned	0.016 sec / 0.000 sec
11	09:57:56	create table owns(driver_id varchar(10) NOT NULL, report_num varchar(10) NOT NULL, primary key(driver_id,report_num))	Error Code: 1072. Key column 'reg_num' doesn't exist in table	0.375 sec
12	09:58:34	create table owns(driver_id varchar(10) NOT NULL, reg_num varchar(10) NOT NULL, primary key(driver_id,reg_num))	0 row(s) affected	0.390 sec
13	09:58:51	desc owns	2 row(s) returned	0.000 sec / 0.000 sec

create table participated(

```

        driver_id varchar(10) NOT NULL,
        reg_num varchar(10) NOT NULL,
        report_num int NOT NULL,
        damage_amount int,
        primary key (driver_id,reg_num,report_num),
        foreign key(driver_id) references person(driver_id),
        foreign key(reg_num) references car(reg_num),
        foreign key(report_num) references accident(report_num)
    );

```

desc participated;

MySQL Workbench Screenshot showing the creation of the 'participated' table and its description.

```

183     damage_amount int,
184     primary key (driver_id,reg_num,report_num),
185     foreign key(driver_id) references person(driver_id),
186     foreign key(reg_num) references car(reg_num),
187     foreign key(report_num) references accident(report_num)
188   );
189 * desc participated;

```

The 'Result Grid' shows the table structure:

Field	Type	Null	Key	Default	Extra
driver_id	varchar(10)	NO	PRI		
reg_num	varchar(10)	NO	PRI		
report_num	int	NO	PRI		
damage_amount	int	YES			

The 'Result 7' tab shows the history of operations:

#	Time	Action	Message	Duration / Fetch
11	09:57:56	create table owns(driver_id varchar(10) NOT NULL, report_num varchar(10) NOT NULL, primary key(driver_id,report_num))	Error Code: 1072. Key column 'reg_num' doesn't exist in table	0.375 sec
12	09:58:34	create table owns(driver_id varchar(10) NOT NULL, reg_num varchar(10) NOT NULL, primary key(driver_id,reg_num))	0 row(s) affected	0.390 sec
13	09:58:51	desc owns	2 row(s) returned	0.000 sec / 0.000 sec
14	10:02:24	create table accident(driver_id varchar(10) NOT NULL, reg_num varchar(10) NOT NULL, report_num int NOT NULL, primary key(driver_id,reg_num,report_num))	Error Code: 1050. Table 'accident' already exists	0.016 sec
15	10:02:42	create table participated(driver_id varchar(10) NOT NULL, reg_num varchar(10) NOT NULL, report_num int NOT NULL, primary key(driver_id,reg_num,report_num))	0 row(s) affected	0.219 sec
16	10:03:00	desc participated	4 row(s) returned	0.016 sec / 0.000 sec

## 2.Enter at least five tuples for each relation.

insert into person(driver\_id,name,address)

values ('A01','Richard','Srinivas nagar'),

      ('A02','Pradeep','Rajaji nagar'),

      ('A03','Smith','Ashok nagar'),

      ('A04','Venu','N R Colony'),

      ('A05','Jhon','Hanumanth Nagar');

select \* from person;

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains two queries:
  - Query 1: An insert statement into the 'person' table with driver\_id, name, and address values for five rows (A01 to A05).
  - Query 2: A select statement from the 'person' table.
- Result Grid:** Displays the results of the select query, showing five rows of data:

driver_id	name	address
A01	Richard	Srinivas nagar
A02	Pradeep	Rajaji nagar
A03	Smith	Ashok nagar
A04	Venu	N R Colony
A05	Jhon	Hanumanth Nagar

- Output Window:** Shows the execution log with various actions and their messages and durations.

insert into car(reg\_num,model,year)

values ('KA052250','Indica','1990'),

      ('KA031181','Lancer','1957'),

      ('KA095477','Toyota',1998),

      ('KA053408','Honda',2008),

      ('KA041702','Audi',2005);

select \* from car;

MySQL Workbench Screenshot:

```

200 values ('KA052250','Indica','1990'),
201 ('KA031181','Lancer','1957'),
202 ('KA095477','Toyota','1998'),
203 ('KA053408','Honda','2008'),
204 ('KA041702','Audi','2005');
205 • select * from car;

```

Result Grid:

reg_num	model	year
KA031181	Lancer	1957
KA041702	Audi	2005
KA052250	Indica	1990
KA053408	Honda	2008
KA095477	Toyota	1998
NULL	NULL	NULL

car 9 x

Action Output:

- 16 10:03:00 Action Message 0.016 sec / 0.000 sec
- 17 10:06:29 insert into person(driver\_id,name,address) values ('A01','Richard','Srinivas nagar'), ('A02','Pradeep','Rajaj nagar') 4 row(s) returned 0.125 sec
- 18 10:06:38 insert into person(driver\_id,name,address) values ('A01','Richard','Srinivas nagar'), ('A02','Pradeep','Rajaj nagar') 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.015 sec
- 19 10:06:59 select \* from person LIMIT 0, 1000 5 row(s) returned 0.000 sec / 0.000 sec
- 20 10:47:48 insert into car(reg\_num,model,year) values ('KA052250','Indica','1990'), ('KA031181','Lancer','1957'), ('KA095477','Toyota','1998') 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.391 sec
- 21 10:47:52 select \* from car LIMIT 0, 1000 5 row(s) returned 0.000 sec / 0.000 sec

Object Info Session Query Completed

```

insert into owns(driver_id,reg_num)
values ('A01','KA052250'),
       ('A02','KA053408'),
       ('A03','KA031181'),
       ('A04','KA095477'),
       ('A05','KA041702');

select * from owns;

```

MySQL Workbench Screenshot:

```

208 values ('A01','KA052250'),
209 ('A02','KA053408'),
210 ('A03','KA031181'),
211 ('A04','KA095477'),
212 ('A05','KA041702');
213 • select * from owns;

```

Result Grid:

driver_id	reg_num
A03	KA031181
A05	KA041702
A01	KA052250
A02	KA053408
A04	KA095477
NULL	NULL

owns 10 x

Action Output:

- 18 10:06:38 insert into person(driver\_id,name,address) values ('A01','Richard','Srinivas nagar'), ('A02','Pradeep','Rajaj nagar') 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.015 sec
- 19 10:06:59 select \* from person LIMIT 0, 1000 5 row(s) returned 0.000 sec / 0.000 sec
- 20 10:47:48 insert into car(reg\_num,model,year) values ('KA052250','Indica','1990'), ('KA031181','Lancer','1957'), ('KA095477','Toyota','1998') 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.391 sec
- 21 10:47:52 select \* from car LIMIT 0, 1000 5 row(s) returned 0.000 sec / 0.000 sec
- 22 10:51:13 insert into accident(report\_num,accident\_date,location) values ('A01','KA052250'), ('A02','KA053408'), ('A03','KA031181'), ('A04','KA095477'), ('A05','KA041702') 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.015 sec
- 23 10:51:19 select \* from accident LIMIT 0, 1000 5 row(s) returned 0.000 sec / 0.000 sec

Object Info Session Query Completed

```

insert into accident (report_num,accident_date,location)
values ('11','2003-01-01','Mysore Road'),
       ('12','2004-02-02','South end circle'),

```

```
(13,'2002-01-21','Bull Temple Road'),
(14,'2008-02-17','Mysore road'),
(15,'2005-03-04','Kanakpura Road');
```

select \* from accident;

report_num	accident_date	location
11	2003-01-01	Mysore Road
12	2004-02-02	South end circle
13	2002-01-21	Bull Temple Road
14	2008-02-17	Mysore road
15	2005-03-04	Kanakpura Road

```
insert into participated(driver_id,reg_num,report_num,damage_amount)
values ('A01','KA052250','11','10000'),
('A02','KA053408',12,50000),
('A03','KA095477',13,25000),
('A04','KA031181',14,3000),
('A05','KA041702',15,5000);
```

select \* from participated;

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

No object selected

participated 12 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
22	10:51:13	Insert into `own`(`driver_id`, `reg_num`) values ('A01', 'KA052250'), ('A02', 'KA053408'), ('A03', 'KA031181'), ('A04', 'KA041702')	5 row(s) affected. Records: 5 Duplicates: 0 Warnings: 0	0.019 sec
23	10:51:19	select * from `own` LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
24	10:53:51	Insert into `accident`(`report_num`, `accident_date`, `location`) values ('11', '2003-01-01', 'Mysoe Road'), ('12', '2004-02-01', 'Mysore Road')	5 row(s) affected. Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
25	10:53:57	select * from `accident` LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
26	10:57:16	Insert into `participated`(`driver_id`, `reg_num`, `report_num`, `damage_amount`) values ('A01', 'KA052250', '11', '10000'), ('A02', 'KA053408', '12', '25000'), ('A03', 'KA05477', '13', '25000'), ('A04', 'KA031181', '14', '3000'), ('A05', 'KA041702', '15', '5000')	5 row(s) affected. Records: 5 Duplicates: 0 Warnings: 0	0.359 sec
27	10:57:21	select * from `participated` LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

### 3.Demonstrate how you

- a.Update the damage amount to 25000 for the car with a specific reg-num(example 'K A053408') for which the accident report number was 12.  
update participated set damage\_amount=25000 where reg\_num='KA053408' and report\_num=12;  
select \* from participated;

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

No object selected

participated 13 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
22	10:51:13	Insert into `own`(`driver_id`, `reg_num`) values ('A01', 'KA052250'), ('A02', 'KA053408'), ('A03', 'KA031181'), ('A04', 'KA041702')	5 row(s) affected. Records: 5 Duplicates: 0 Warnings: 0	0.019 sec
23	10:51:19	select * from `own` LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
24	10:53:51	Insert into `accident`(`report_num`, `accident_date`, `location`) values ('11', '2003-01-01', 'Mysoe Road'), ('12', '2004-02-01', 'Mysore Road')	5 row(s) affected. Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
25	10:53:57	select * from `accident` LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
26	10:57:16	Insert into `participated`(`driver_id`, `reg_num`, `report_num`, `damage_amount`) values ('A01', 'KA052250', '11', '10000'), ('A02', 'KA053408', '12', '25000'), ('A03', 'KA05477', '13', '25000'), ('A04', 'KA031181', '14', '3000'), ('A05', 'KA041702', '15', '5000')	5 row(s) affected. Records: 5 Duplicates: 0 Warnings: 0	0.359 sec
27	10:57:21	select * from `participated` LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
28	11:00:25	update participated set damage_amount=25000 where reg_num='KA053408' and report_num=12	1 row(s) affected. Rows matched: 1. Changed: 1. Warnings: 0	0.000 sec
29	11:00:39	select * from `participated` LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

### b.Add a new accident to the database.

insert into accident  
values (16,'2008-03-15','Domlur');  
select \* from accident;

```

234 • select * from participated;
235
236 • select * from accident;
237 • insert into accident
238 values (16,'2008-03-15','Domlur');
239 • select * from accident;

```

report_num	accident_date	location
11	2003-01-01	Mysore Road
12	2004-02-02	South end circle
13	2002-01-21	Bull Temple Road
14	2008-02-17	Mysore road
15	2005-03-04	Kanakpura Road
16	2008-03-15	Domlur
...	...	...

Action Output

#	Time	Action	Message	Duration / Fetch
28	11:00:25	update participated set damage_amount=25000 where reg_num='KA053408' and report_num=12	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
29	11:00:39	select * from participated LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
30	11:02:08	select * from accident LIMIT 0, 1000	5 row(s) returned	0.016 sec / 0.000 sec
31	11:04:05	insert into accident(16, '2008-03-15', 'Domlur')	Error Code: 1064 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 'insert into accident(16, '2008-03-15', 'Domlur')' at line 1	0.000 sec
32	11:04:42	insert into accident values (16, '2008-03-15', 'Domlur')	1 row(s) affected	0.015 sec
33	11:04:52	select * from accident LIMIT 0, 1000	6 row(s) returned	0.015 sec / 0.000 sec

#### 4.Find the total number of people who owned cars that involved in accidents in 2008.

```
select count(distinct driver_id) CNT from participated a, accident b where
a.report_num=b.report_num and b.accident_date like '2008%';
```

```

132 • select * from participated;
133
134 • select * from accident;
135 • insert into accident values(16,'2008-03-15','Domlur');
136 • select * from accident;
137
138 • select count(distinct driver_id) CNT from participated a, accident b where a.report_num=b.report_num and b.accident_date like '2008%';

```

CNT
1

Action Output

#	Time	Action	Message	Duration / Fetch
38	10:46:41	insert into accident values(16, '2008-03-15', 'Domlur')	1 row(s) affected	0.015 sec
39	10:46:43	select * from accident LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
40	10:47:19	select count(distinct driver_id) CNT from participated a, accident b where a.report_num=b.report_num and b.accident_date like '2008%'	1 row(s) returned	0.015 sec / 0.000 sec
41	10:54:35	select count(distinct driver_id) CNT from participated a, accident b where a.report_num=b.report_num and b.accident_date like '2008%'	1 row(s) returned	0.000 sec / 0.000 sec
42	10:56:06	select count(distinct driver_id) CNT from participated a, accident b where a.report_num=b.report_num and b.accident_date like '2008%'	1 row(s) returned	0.000 sec / 0.000 sec
43	10:59:39	select count(distinct driver_id) CNT from participated a, accident b where a.report_num=b.report_num and b.accident_date like '2008%'	1 row(s) returned	0.000 sec / 0.000 sec

#### 5.Find the number of accidents in which cars belonging to a specific model (example )were involved.

```
select count(report_num) CNT from car c,participated p where c.reg_num=p.reg_num and
model='Lancer';
```

```

MySQL Workbench
File Edit View Query Database Server Tools Help
Navigator: Local instance MySQL80 x
Query 1: SQL File 4*
134 • select * from accident;
135 • insert into accident values(16,'2008-03-15','Domur');
136 • select * from accident;
137
138 • select count(distinct driver_id) CNT from participated a, accident b where a.report_num=b.report_num and b.accident_date like '2008%';
139
140 • select count(report_num) CNT from car c,participated p where c.reg_num=p.reg_num and model='Lancer';

```

Result Grid: CNT

CNT
1

Result 24 x

Action Output

#	Time	Action	Message	Duration / Fetch
39	10:46:43	select * from accident LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
40	10:47:19	select count(distinct driver_id) CNT from participated a, accident b where a.report_num=b.report_num and b.ac... 1 row(s) returned		0.015 sec / 0.000 sec
41	10:54:35	select count(distinct driver_id) CNT from participated a, accident b where a.report_num=b.report_num and b.ac... 1 row(s) returned		0.000 sec / 0.000 sec
42	10:56:06	select count(distinct driver_id) CNT from participated a, accident b where a.report_num=b.report_num and b.ac... 1 row(s) returned		0.000 sec / 0.000 sec
43	10:59:39	select count(distinct driver_id) CNT from participated a, accident b where a.report_num=b.report_num and b.ac... 1 row(s) returned		0.000 sec / 0.000 sec
44	10:59:50	select count(report_num) CNT from car c,participated p where c.reg_num=p.reg_num and model='Lancer' LIM... 1 row(s) returned		0.016 sec / 0.000 sec

Object Info Session

Query Completed

## EXTRA QUERIES:

Update the appropriate table for the following details

1. Mr. Smith meet with an accident when he travelled in "Audi (KA05MC001)" car on 1 st march 2019 at "Bull temple road". Police registered a case with number 16 recorded damage amount with 75000. Mr. Smith has bought the car in the year 2018.

insert into person  
values ('A06','Smith','JP nagar');

select \* from person;

```

MySQL Workbench
File Edit View Query Database Server Tools Help
Navigator: Local instance MySQL80 (inst... x Local instance MySQL80 (bank) x
Query 1: v
241 • show databases;
242 • use insurance1;
243
244 • insert into person
245   values ('A06','Smith','JP nagar');
246
247 • select * from person;

```

Result Grid: driver\_id name address

driver_id	name	address
A01	Richard	Srinivas nagar
A02	Pradeep	Rajeev nagar
A03	Smith	Ashok nagar
A04	Venu	N.R Colony
A05	Jhon	Harumathi Nagar
A06	Smith	JP nagar
A07	Smith	BBM

person 3 x

Action Output

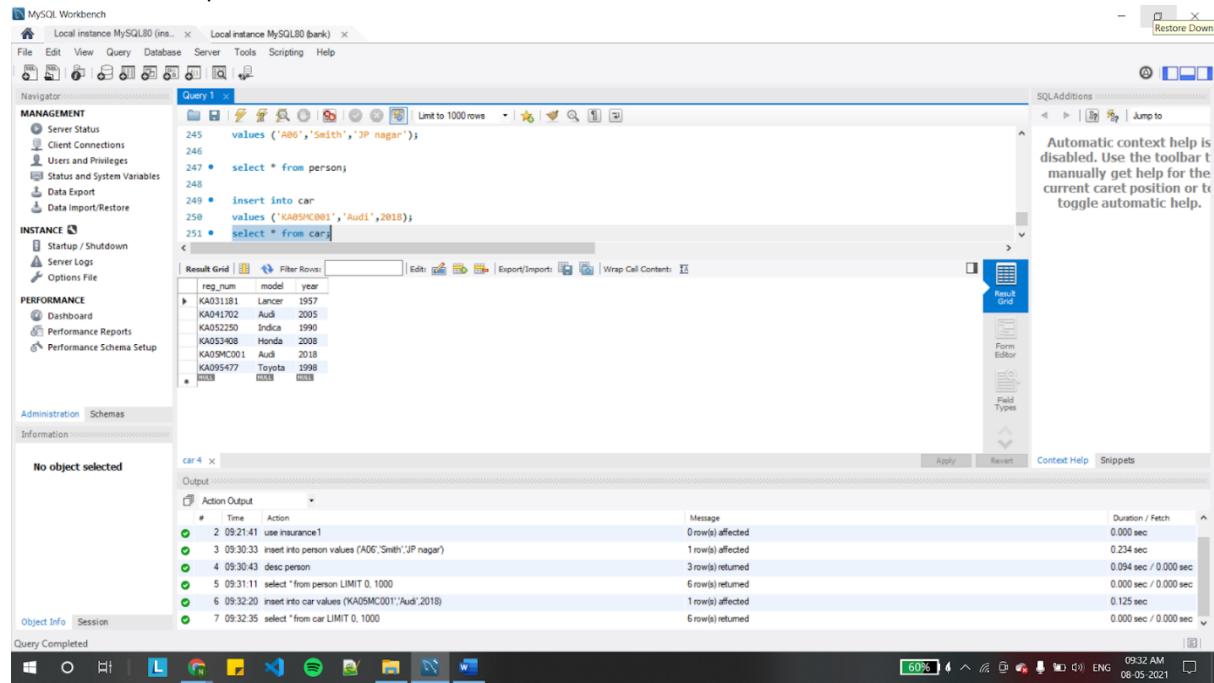
#	Time	Action	Message	Duration / Fetch
1	09:21:29	show databases	8 row(s) returned	0.000 sec / 0.000 sec
2	09:21:41	use insurance1	0 row(s) affected	0.000 sec
3	09:30:33	insert into person values ('A06','Smith','JP nagar')	1 row(s) affected	0.234 sec
4	09:30:43	desc person	3 row(s) returned	0.094 sec / 0.000 sec
5	09:31:11	select * from person LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

insert into car

values ('KA05MC001','Audi',2018);  
 select \* from car;



```

245 values ('A06','Smith','JP nagar');
246
247 • select * from person;
248
249 • insert into car
250     values ('KA05MC001','Audi',2018);
251 • select * from car;
    
```

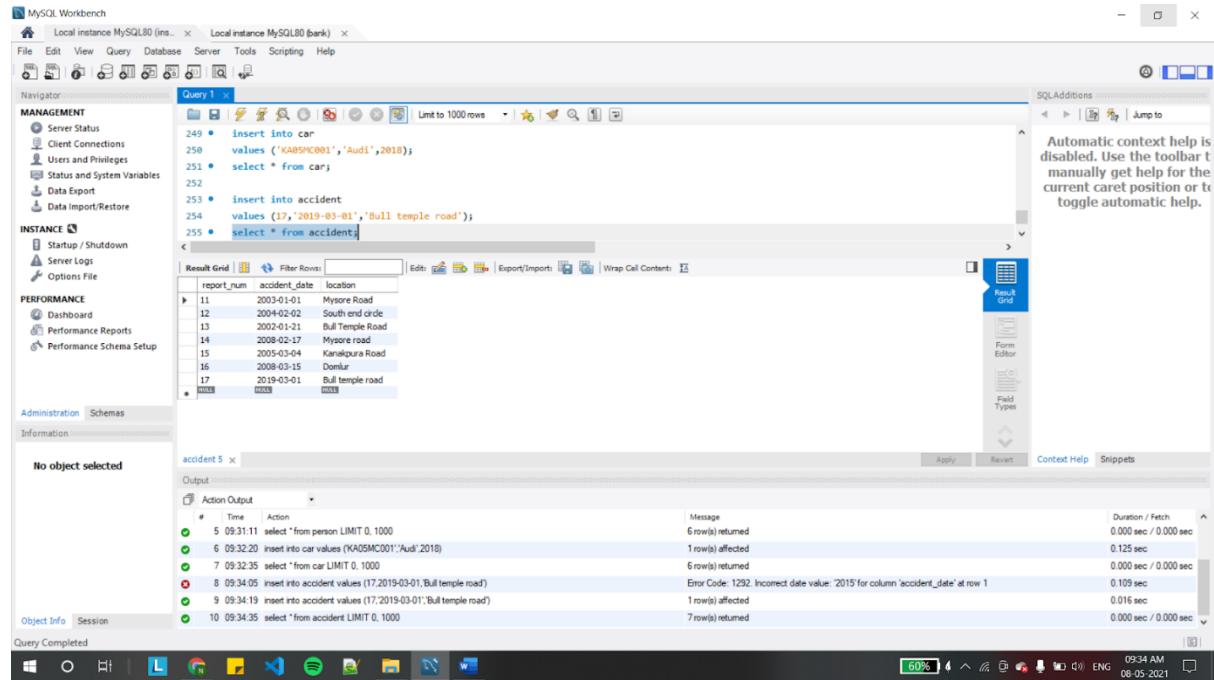
The screenshot shows the MySQL Workbench interface. The left sidebar has sections for MANAGEMENT, INSTANCE, and PERFORMANCE. The central area has a 'Query 1' tab with the above SQL code. Below it is a 'Result Grid' showing the following data:

reg_num	model	year
KA031181	Lancer	1957
KA041702	Audi	2005
KA052250	Indica	1990
KA053408	Honda	2009
KA059C001	Audi	2018
KA095477	Toyota	1998
NULL	NULL	NULL

Below the result grid is an 'Output' section showing the execution log:

#	Time	Action	Message	Duration / Fetch
2	09:21:41	use insurance1	0 row(s) affected	0.000 sec
3	09:30:33	insert into person values ('A06','Smith','JP nagar')	1 row(s) affected	0.234 sec
4	09:30:43	desc person	3 row(s) returned	0.094 sec / 0.000 sec
5	09:31:11	select * from person LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
6	09:32:20	insert into car values ('KA05MC001','Audi',2018)	1 row(s) affected	0.125 sec
7	09:32:35	select * from car LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

insert into accident  
 values (17,'2019-03-01','Bull temple road');  
 select \* from accident;



```

249 • insert into car
250     values ('KA05MC001','Audi',2018);
251 • select * from car;
252
253 • insert into accident
254     values (17,'2019-03-01','Bull temple road');
255 • select * from accident;
    
```

The screenshot shows the MySQL Workbench interface. The left sidebar has sections for MANAGEMENT, INSTANCE, and PERFORMANCE. The central area has a 'Query 1' tab with the above SQL code. Below it is a 'Result Grid' showing the following data:

report_num	accident_date	location
11	2003-01-01	Mysore Road
12	2004-02-02	South end circle
13	2002-01-21	Bull Temple Road
14	2008-02-17	Mysore road
15	2005-03-04	Karikalpura Road
16	2008-03-15	Donlur
17	2019-03-01	Bull temple road
NULL	NULL	NULL

Below the result grid is an 'Output' section showing the execution log:

#	Time	Action	Message	Duration / Fetch
5	09:31:11	select * from person LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
6	09:32:20	insert into car values ('KA05MC001','Audi',2018)	1 row(s) affected	0.125 sec
7	09:32:35	select * from car LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
8	09:34:05	insert into accident values (17,'2019-03-01','Bull temple road')	Error Code: 1292: Incorrect date value: '2015' for column 'accident_date' at row 1	0.109 sec
9	09:34:19	insert into accident values (17,'2019-03-01','Bull temple road')	1 row(s) affected	0.016 sec
10	09:34:35	select * from accident LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

insert into participated  
 values ('A06','KA05MC001',17,75000);  
 select \* from participated;

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

**MANAGEMENT**

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

**INSTANCE**

- Startup / Shutdown
- Server Logs
- Options File

**PERFORMANCE**

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

participated 6 x

Action Output

#	Time	Action	Message	Duration / Fetch
7	09:32:35	select * from car LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
8	09:34:05	insert into accident values (17, '2019-03-01', 'Bull temple road')	Error Code: 1292: Incorrect date value: '2015' for column 'accident_date' at row 1	0.105 sec
9	09:34:19	insert into accident values (17, 2019-03-01, 'Bull temple road')	1 row(s) affected	0.016 sec
10	09:34:35	select * from accident LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
11	09:35:54	insert into participated values ('A06', 'KA05MC001', 17, 75000)	1 row(s) affected	0.391 sec
12	09:35:58	select * from participated LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

2. Print driver\_id and reg\_number of the owner who owns the car "indica".

```
select driver_id,reg_num
from owns
where reg_num=
    select reg_num
        from car
        where model='Indica'
);
```

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

**MANAGEMENT**

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

**INSTANCE**

- Startup / Shutdown
- Server Logs
- Options File

**PERFORMANCE**

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

owns 8 x

Action Output

#	Time	Action	Message	Duration / Fetch
9	09:34:19	insert into accident values (17, '2019-03-01', 'Bull temple road')	1 row(s) affected	0.016 sec
10	09:34:35	select * from accident LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
11	09:35:54	insert into participated values ('A06', 'KA05MC001', 17, 75000)	1 row(s) affected	0.391 sec
12	09:35:58	select * from participated LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
13	09:38:49	select driver_id,reg_num from owns where reg_num=(select reg_num from car where model='Indica') LIMIT 0, 1000	1 row(s) returned	0.360 sec / 0.000 sec
14	09:39:05	select driver_id,reg_num from owns where reg_num=(select reg_num from car where model='Indica') LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec

Object Info Session

Query Completed

3. List the reg\_nums of the cars which has involved in the accident between the period 2003 to 2005.

```

select reg_num
from participated
where report_num IN(
    select report_num
    from accident
    where accident_date between '2003-01-01' and '2005-12-31'
);

```

The screenshot shows the MySQL Workbench interface with a query editor containing the provided SQL code. The result grid displays the following data:

reg_num
KA052250
KA052408
KA041702

4. List total no of accident happened in Mysore.  
 select count(report\_num) AS no\_of\_acc  
 from accident  
 where location like '%Mysore%';

The screenshot shows the MySQL Workbench interface with a query editor containing the provided SQL code. The result grid displays the following data:

no_of_acc
2

5. Find the customer whose damage amount is greater than the average damage amount.

```
select driver_id  
from participated  
where damage_amount > (  
    select avg(damage_amount)  
    from participated  
)
```

The screenshot shows the MySQL Workbench interface. In the central Query Editor window, the following SQL code is displayed:

```
280  
281 • select driver_id  
282     from participated  
283     where damage_amount > [ ]  
284         select avg(damage_amount)  
285             from participated  
286     )
```

The result grid shows the output of the query:

driver_id
A02
A03
A06

Below the result grid, the Output pane displays the execution log:

Action	Time	Message	Duration / Fetch
15	09:49:13	select reg_num from participated where report_num IN( select report_num from accident where accident_date like '2008%' )	0.000 sec / 0.000 sec
16	09:51:43	select reg_num from participated where report_num IN( select report_num from accident where accident_date like '2008%' ) AS no_of_acc from accident where location='Myrtle' LIMIT 0, 1000	0.000 sec / 0.000 sec
17	09:54:16	select count(report_num) AS no_of_acc from accident where location='Myrtle' LIMIT 0, 1000	1 row(s) returned
18	09:55:40	select count(report_num) AS no_of_acc from accident where location like '%Myrtle%' LIMIT 0, 1000	1 row(s) returned
19	09:57:22	select count(report_num) AS no_of_acc from accident where location like '%Myrtle%' LIMIT 0, 1000	1 row(s) returned
20	10:08:38	select driver_id from participated where damage_amount > ( select avg(damage_amount) from participated ) UNION	3 row(s) returned

6. Find the total number of people who owned cars that were involved in accidents in 2008.

```
select count(driver_id)  
from participated  
where report_num IN (  
    select report_num  
    from accident  
    where accident_date like '2008%'  
)
```

The screenshot shows the MySQL Workbench interface. In the top-left corner, there are two tabs: "Local instance MySQL80 (ins...)" and "Local instance MySQL80 (bank)". The main window has a toolbar with various icons. On the left, a "Navigator" pane displays sections like MANAGEMENT, INSTANCE, and PERFORMANCE. Below the navigator is an "Administration" section with tabs for Schemas and Information, currently showing "No object selected". The central area contains a "Query 1" tab with the following SQL code:

```

288 • select count(driver_id)
289   from participated
290   where report_num IN (
291     select report_num
292       from accident
293      where accident_date like '2008%'
294

```

Below the code is a "Result Grid" showing the output of the query: "count(driver\_id)" with a value of 1. To the right of the result grid is a vertical toolbar with icons for Result Grid, Form Editor, and Field Types. A tooltip message says: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help." At the bottom of the interface, there is a status bar showing "55%", "10:21 AM", "ENG", and the date "08-05-2021".

7. Delete the Audi belonging to “Jhon”.

**Referential integrity is violated.Hence,it cannot be deleted.**

NAME:NEELAM H GODIHAL

USN:1BM19CS220

## PROGRAM 2: BANKING ENTERPRISE DATABASE

Consider the following database for a banking enterprise.

**Branch** (branch-name: String, branch-city: String, assets: real)

**BankAccount**(accno: int, branch-name: String, balance: real)

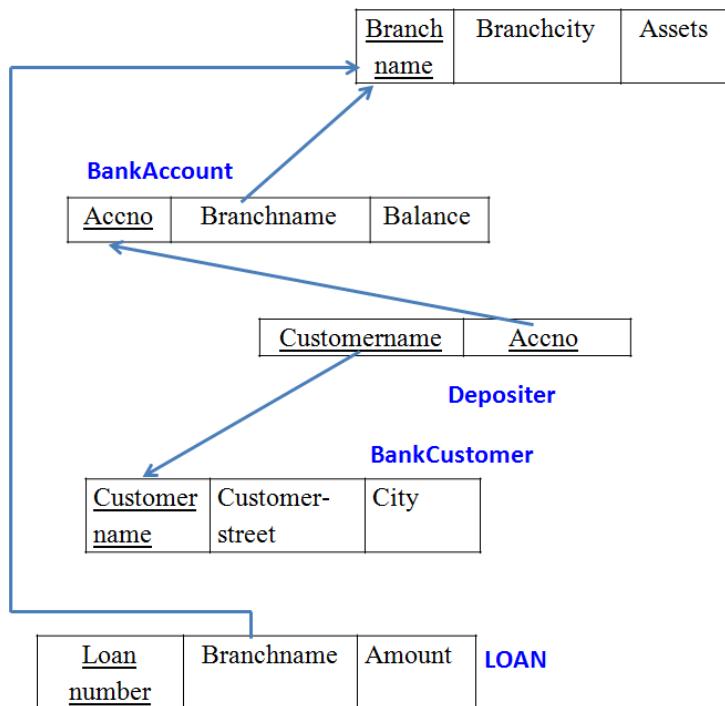
**BankCustomer** (customer-name: String, customer-street: String, customer-city: String)

**Depositer**(customer-name: String, accno: int)

**Loan** (loan-number: int, branch-name: String, amount: real)

**INTRODUCTION:** This database is developed for supporting banking facilities. Details of the branch along with the accounts and loans handled by them are recorded. Also details of the depositors of the corresponding branches are maintained.

### Schema Diagram



### Sample Table data

**Branch**

BRANCHNAME	BRANCHCITY	ASSESTS
SBI_Chamrajpet	Bangalore	50000
SBI_ResidencyRoad	Bangalore	10000
SBI_ShivajiRoad	Bombay	20000
SBI_ParliamentRoad	Delhi	10000
SBI_Jantarmantar	Delhi	20000

**BankAccount**

ACCNO	BRANCHNAME	BALANCE
1	SBI_Chamrajpet	2000
2	SBI_ResidencyRoad	5000
3	SBI_ShivajiRoad	6000
4	SBI_ParliamentRoad	9000
5	SBI_Jantarmantar	8000
6	SBI_ShivajiRoad	4000
8	SBI_ResidencyRoad	4000
9	SBI_ParliamentRoad	3000
10	SBI_ResidencyRoad	5000
11	SBI_Jantarmantar	2000

**BankCustomer**

CUSTOMERNAME	CUSTOMERSTREET	CUSTOMERCITY
Avinash	Bull_Temple_Road	Bangalore
Dinesh	Bannergatta_Road	Bangalore
Mohan	NationalCollege_Road	Bangalore
Nikil	Akbar_Road	Delhi
Ravi	Prithviraj_Road	Delhi

**Depositer**

CUSTOMERNAME	ACCNO
Avinash	1
Dinesh	2
Nikil	4
Ravi	5
Avinash	8
Nikil	9
Dinesh	10
Nikil	11

**Loan**

LOANNUMBER	BRANCHNAME	AMOUNT
1	SBI_Chamrajpet	1000
2	SBI_ResidencyRoad	2000
3	SBI_ShivajiRoad	3000
4	SBI_ParliamentRoad	4000
5	SBI_Jantarmantar	5000

1. Create the above tables by properly specifying the primary keys and the foreign keys.

create database bank;  
use bank;

```
create table branch(
    branch_name varchar(255),
    branch_city varchar(20),
    assets real,
    primary key(branch_name)
);
```

```
desc branch;
```

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: MANAGEMENT INSTANCE PERFORMANCE Administration Schemas Information No object selected

```

42     branch_city varchar(20),
43     assets real,
44     primary key(branch_name)
45   );
46
47 • desc branch;
48

```

Result Grid Export Wrap Cell Content

Field	Type	Null	Key	Default	Extra
branch_name	varchar(255)	NO	PRI	NULL	
branch_city	varchar(20)	YES	MUL	NULL	
assets	double	YES		NULL	

SQLAdditions SQL Editor Field Types

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result 3 x Read Only Context Help Snippets

Action Output

#	Time	Action	Message	Duration / Fetch
2	08:38:11	drop database bank	3 row(s) affected	0.063 sec
3	08:38:19	show databases	7 row(s) returned	0.016 sec / 0.000 sec
4	08:39:02	create database bank	1 row(s) affected	0.015 sec
5	08:39:05	use bank	0 row(s) affected	0.000 sec
6	08:39:30	create table branch(branch_name varchar(255), branch_city varchar(20), assets real, primary key(bran...)	0 row(s) affected	0.110 sec
7	08:39:43	desc branch	3 row(s) returned	0.000 sec / 0.000 sec

Object Info Session Query Completed

```

create table bank_account(
    acc_no int,
    branch_name varchar(255),
    balance real,
    primary key(acc_no),
    foreign key(branch_name) references branch(branch_name)
);

```

desc bank\_account;

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: MANAGEMENT INSTANCE PERFORMANCE Administration Schemas Information No object selected

```

53     primary key(acc_no),
54     foreign key(branch_name) references branch(branch_name)
55   );
56
57 • desc bank_account;
58

```

Result Grid Export Wrap Cell Content

Field	Type	Null	Key	Default	Extra
acc_no	int	NO	PRI	NULL	
branch_name	varchar(255)	YES	MUL	NULL	
balance	double	YES		NULL	

SQLAdditions SQL Editor Field Types

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result 4 x Read Only Context Help Snippets

Action Output

#	Time	Action	Message	Duration / Fetch
4	08:39:02	create database bank	1 row(s) affected	0.019 sec
5	08:39:05	use bank	0 row(s) affected	0.000 sec
6	08:39:30	create table bank_account(acc_no int, branch_name varchar(255), balance real, primary key(acc_no)...)	0 row(s) affected	0.110 sec
7	08:39:43	desc branch	3 row(s) returned	0.000 sec / 0.000 sec
8	08:44:00	create table bank_account(acc_no int, branch_name varchar(255), balance real, primary key(acc_no)...)	0 row(s) affected	0.031 sec
9	08:44:06	desc bank_account	3 row(s) returned	0.000 sec / 0.000 sec

Object Info Session Query Completed

create table bank\_customer(

```

        customer_name varchar(30),
customer_street varchar(50),
customer_city varchar(20),
primary key(customer_name)
);

```

desc bank\_customer;

The screenshot shows the MySQL Workbench interface with multiple tabs at the top: Local instance MySQL80, Local instance MySQL80 (bank), Local instance MySQL80 (supplier), Local instance MySQL80 (flight), and Local instance MySQL80 (student). The main area displays a SQL editor with the following code:

```

58
59 • create table bank_customer(
60     customer_name varchar(30),
61     customer_street varchar(50),
62     customer_city varchar(20),
63     primary key(customer_name)
64 );
65 • use bank;
66 • desc bank_customer;

```

Below the code, the Results pane shows the table structure:

Field	Type	Null	Key	Default	Extra
customer_name	varchar(30)	NO	PRI		
customer_street	varchar(50)	YES			
customer_city	varchar(20)	YES			

The Output pane shows the execution history:

Action	Time	Action	Message	Duration / Fetch
1	03:38:18	desc bank_customer	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCHEMAS list in the ...	0.000 sec
2	03:38:25	use bank	0 row(s) affected	0.016 sec
3	03:38:27	desc bank_customer	3 row(s) returned	0.016 sec / 0.000 sec

```

create table depositer(
        customer_name varchar(30),
acc_no int,
primary key(customer_name,acc_no),
foreign key(customer_name) references bank_customer(customer_name),
foreign key(acc_no) references bank_account(acc_no)
);

```

desc depositer;

The screenshot shows the MySQL Workbench interface with the following details:

- Query 1:** Contains the SQL code for creating the bank schema and the depositor table.
- Result Grid:** Shows the structure of the depositor table with columns: customer\_name (varchar(30), NO, PRI, NULL) and acc\_no (int, NO, PRI, NULL).
- Result 5:** Shows the history of actions taken during the schema creation, including table creation and the creation of the depositor table.

```

CREATE SCHEMA bank;
USE bank;
CREATE TABLE bank_account(
    acc_no int,
    branch_name varchar(25),
    balance real,
    primary key(acc_no),
    foreign key(branch_name) references branch(branch_name)
);
CREATE TABLE depositor(
    customer_name varchar(30),
    acc_no int,
    primary key(customer_name,acc_no),
    foreign key(customer_name) references bank_customer(customer_name),
    foreign key(acc_no) references bank_account(acc_no)
);
desc depositor;

```

```

create table loan(
    loan_num int,
    branch_name varchar(30),
    amount real,
    primary key(loan_num),
    foreign key(branch_name) references branch(branch_name)
);

```

```
desc loan;
```

The screenshot shows the MySQL Workbench interface with the following details:

- Query 1:** Contains the SQL code for creating the loan table.
- Result Grid:** Shows the structure of the loan table with columns: loan\_num (int, NO, PRI, NULL), branch\_name (varchar(30), YES, MUL, NULL), and amount (double, YES).
- Result 6:** Shows the history of actions taken during the schema creation, including table creation and the creation of the loan table.

```

CREATE TABLE loan(
    loan_num int,
    branch_name varchar(30),
    amount real,
    primary key(loan_num),
    foreign key(branch_name) references branch(branch_name)
);
desc loan;

```

## 2. Enter at least five tuples for each relation.

```

insert into branch
values ('SBI_Chamrajpet','Bangalore',50000),
       ('SBI_ResidencyRoad','Bangalore',10000),
       ('SBI_ShivajiRoad','Bombay',20000),
       ('SBI_ParliamentRoad','Delhi',10000),
       ('SBI_Jantarmantar','Delhi',20000);

```

```
select * from branch;
```

The screenshot shows the MySQL Workbench interface with a query editor window open. The query is:

```

90   ('SBI_ResidencyRoad','Bangalore',10000),
91   ('SBI_ShivajiRoad','Bombay',20000),
92   ('SBI_ParliamentRoad','Delhi',10000),
93   ('SBI_Jantarmantar','Delhi',20000);
94
95 • select * from branch;
96

```

The results grid displays the following data:

branch_name	branch_city	assets
SBI_Chamrajpet	Bangalore	50000
SBI_Jantarmantar	Delhi	20000
SBI_ResidencyRoad	Bangalore	10000
SBI_ShivajiRoad	Bangalore	10000
SBI_ParliamentRoad	Bombay	20000

The status bar at the bottom right indicates the session was completed at 09:25 AM on 15-05-2021.

```

insert into bank_account
values (1,'SBI_Chamrajpet',2000),
       (2,'SBI_ResidencyRoad',5000),
       (3,'SBI_ShivajiRoad',6000),
       (4,'SBI_ParliamentRoad',9000),
       (5,'SBI_Jantarmantar',8000);

```

```
select * from bank_account;
```

MySQL Workbench Screenshot showing the creation of the bank\_accounts table:

```

99     (2,'SBI_ResidencyRoad',5000),
100    (3,'SBI_ShivajiRoad',6000),
101    (4,'SBI_ParliamentRoad',9000),
102    (5,'SBI_JantarMantar',8000);
103
104 • select * from bank_account;
105

```

The result grid shows the following data:

acc_no	branch_name	balance
1	SBI_Champet	2000
2	SBI_ResidencyRoad	5000
3	SBI_ShivajiRoad	6000
4	SBI_ParliamentRoad	9000
5	SBI_JantarMantar	8000
*	MAX	MAX

Action Output log:

- 15 09:16:24 des loan 3 row(s) returned
- 16 09:25:07 insert into branch values ('SBI\_Champet','Bangalore',50000), ('SBI\_ResidencyRoad','Bangalore',10000), ... 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.125 sec
- 17 09:25:13 select \* from branch LIMIT 0, 2000 5 row(s) returned
- 18 09:25:57 insert into loan values (''), (''), (''), (''). Error Code: 1064. 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 '' at line 1 0.078 sec
- 19 09:32:32 insert into bank\_account values (1,SBI\_Champet,2000), (2,SBI\_ResidencyRoad,5000), (3,SBI\_Shiv... 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.000 sec
- 20 09:32:52 select \*from bank\_account LIMIT 0, 2000 5 row(s) returned 0.000 sec / 0.000 sec

```

insert into bank_customer
values ('Avinash','Bull Temple Road','Bangalore'),
       ('Dinesh','Bannergatta Road','Bangalore'),
       ('Mohan','National College Road','Bangalore'),
       ('Nikil','Akbar Road','Delhi'),
       ('Ravi','Prithviraj Road','Delhi');

```

```
select * from bank_customer;
```

MySQL Workbench Screenshot showing the creation of the bank\_customer table:

```

108    ('Dinesh','Bannergatta Road','Bangalore'),
109    ('Mohan','National College Road','Bangalore'),
110    ('Nikil','Akbar Road','Delhi'),
111    ('Ravi','Prithviraj Road','Delhi');
112
113 • select * from bank_customer;
114

```

The result grid shows the following data:

customer_name	customer_street	customer_city
Avinash	Bull Temple Road	Bangalore
Dinesh	Bannergatta Road	Bangalore
Mohan	National College Road	Bangalore
Nikil	Akbar Road	Delhi
Ravi	Prithviraj Road	Delhi
*	MAX	MAX

Action Output log:

- 17 09:25:13 select \* from branch LIMIT 0, 2000 5 row(s) returned
- 18 09:25:57 insert into loan values (''), (''), (''), (''). Error Code: 1064. 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 '' at line 1 0.078 sec
- 19 09:32:32 insert into bank\_account values (1,SBI\_Champet,2000), (2,SBI\_ResidencyRoad,5000), (3,SBI\_Shiv... 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.000 sec
- 20 09:32:52 select \*from bank\_account LIMIT 0, 2000 5 row(s) returned
- 21 09:38:57 insert into bank\_customer values ('Avinash','Bull Temple Road','Bangalore'), ('Dinesh','Bannergatta Road','Ban... 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.359 sec
- 22 09:39:02 select \*from bank\_customer LIMIT 0, 2000 5 row(s) returned 0.000 sec / 0.000 sec

```

insert into depositer
values ('Avinash',1),

```

```
('Dinesh',2),
('Nikil',4),
('Ravi',5),
('Mohan',3);
```

select \* from depositer;

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the following SQL code and its execution results:
 

```
117 ('Dinesh',2),
118 ('Nikil',4),
119 ('Ravi',5),
120 ('Mohan',3);
121
122 • select * from depositer;
123
```

customer_name	acc_no
Avinash	1
Dinesh	2
Mohan	3
Nikil	4
Ravi	5
*	NULL
- Action Output:** Shows the history of database actions with their times, messages, and durations:
 

Action	Time	Message	Duration / Fetch
19 09:32:32 Insert into bank_account values (1,'SBI_Chamrajpet',2000),(2,'SBI_ResidencyRoad',5000), (3,'SBI_Shiv... 20 09:32:52 select * from bank_account LIMIT 0, 2000 21 09:38:57 Insert into bank_customer values (Avinash,Bull Temple Road','Bangalore'),(Dinesh,Bannerghatta Road','Ban... 22 09:39:02 select * from bank_customer LIMIT 0, 2000 23 09:43:27 Insert into depositer values (Avinash',1),(Dinesh',2), (Nikil',4), (Ravi',5), (Mohan',3) 24 09:43:43 select * from depositer LIMIT 0, 2000		5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 5 row(s) returned 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 5 row(s) returned 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 5 row(s) returned	0.000 sec 0.000 sec / 0.000 sec 0.359 sec 0.000 sec / 0.000 sec 0.015 sec 0.000 sec / 0.000 sec

insert into loan

```
values (1,'SBI_Chamrajpet',1000),
       (2,'SBI_ResidencyRoad',2000),
       (3,'SBI_ShivajiRoad',3000),
       (4,'SBI_ParliamentRoad',4000),
       (5,'SBI_Jantarmantar',5000);
```

select \* from loan;

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Management
  Client Connections
  Users and Privileges
  Status and System Variables
  Data Export
  Data Import/Restore
INSTANCE
  Startup / Shutdown
  Server Logs
  Options File
PERFORMANCE
  Dashboard
  Performance Reports
  Performance Schema Setup
Administration Schemas
Information
No object selected
Query 1 x
127      (2,'SBI_ResidencyRoad',2000),
128      (3,'SBI_ShivajiRoad',3000),
129      (4,'SBI_ParliamentRoad',4000),
130      (5,'SBI_JantarMantar',5000);
131
132 • select * from loan;
133
Result Grid | Filter Rows: 15 | Edit: | Export/Import: | Wrap Cell Content: |
loan_num branch_name amount
1 SBI_Chennajpet 1000
2 SBI_ResidencyRoad 2000
3 SBI_ShivajiRoad 3000
4 SBI_ParliamentRoad 4000
5 SBI_JantarMantar 5000
* 1000 1000
SQLAdditions
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
loan 11 x
Output
Action Output
# Time Action Message Duration / Fetch
21 09:38:57 Insert into bank_customer values ('Avinash','Bull Temple Road','Bangalore'), ('Dinesh','Bannerghatta Road','Ban... 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.359 sec
22 09:39:02 select * from bank_customer LIMIT 0, 2000 5 row(s) returned 0.000 sec / 0.000 sec
23 09:43:27 Insert into depositor values ('Avinash',1), ('Dinesh',2), ('Nikil',4), ('Ravi',5), ('Mohan',3) 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.015 sec
24 09:43:43 select * from depositor LIMIT 0, 2000 5 row(s) returned 0.000 sec / 0.000 sec
25 09:47:31 Insert into loan values (1,'SBI_Chennajpet',1000), (2,'SBI_ResidencyRoad',2000), (3,'SBI_ShivajiRoad',3000), (4,'SBI_ParliamentRoad',4000), (5,'SBI_JantarMantar',5000) 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.000 sec
26 09:47:48 select * from loan LIMIT 0, 2000 5 row(s) returned 0.000 sec / 0.000 sec
Object Info Session
Query Completed

```

### 3. Find all the customers who have at least two accounts at the *Main* branch (ex. **SBI\_ResidencyRoad**).

select c.customer\_name

from bank\_customer c

where exists(

select d.customer\_name,count(d.customer\_name)

from depositer d,bank\_account ba

where(

d.acc\_no=ba.acc\_no and c.customer\_name=d.customer\_name and

ba.branch\_name='SBI\_ResidencyRoad'

)

group by d.customer\_name

having count(d.customer\_name)>=2

);

MySQL Workbench interface showing a query editor with the following SQL code:

```

138     where exists(
139         select d.customer_name,count(d.customer_name)
140         from depositor d,branch b,`bank_account` a
141         where(
142             d.acc_no=ba.acc_no and c.customer_name=d.customer_name and ba.branch_name='SBT_ResidencyRoad'
143         )
144         group by d.customer_name
145         having count(d.customer_name)>2
146     );

```

The result grid shows one row with 'customer\_name' as NULL.

The Performance panel shows the following execution plan details:

Action	Time	Duration / Fetch
1 03:38:18	desc bank_customer	0.000 sec
2 03:38:25	use bank	0.016 sec
3 03:38:27	desc bank_customer	0.016 sec / 0.000 sec
4 03:46:23	select c.customer_name from bank_customer c where exists(select d.customer_name,count(d.customer_name) from depositer ...)	0.016 sec / 0.000 sec

Query Completed at 03:47 AM on 05-06-2021.

#### 4. Find all the customers who have an account at *all* the branches located in a specific city (Ex. Delhi).

select d.customer\_name from depositer d,branch b,`bank\_account` a  
where b.branch\_name=a.branch\_name

AND a.acc\_no=d.acc\_no

and branch\_city='Delhi'

group by d.customer\_name

HAVING COUNT(distinct b.branch\_name)=(  
    SELECT COUNT(branch\_name)  
    FROM branch  
    WHERE branch\_city='Delhi');

MySQL Workbench interface showing a query editor with the following SQL code:

```

148     select d.customer_name from depositer d,branch b,`bank_account` a
149     where b.branch_name=a.branch_name
150     AND a.acc_no=d.acc_no
151     and branch_city='Delhi'
152     group by d.customer_name
153     HAVING COUNT(distinct b.branch_name)=(
154         SELECT COUNT(branch_name)
155         FROM branch
156         WHERE branch_city='Delhi')

```

The result grid shows one row with 'customer\_name' as NULL.

The Performance panel shows the following execution plan details:

Action	Time	Duration / Fetch
1 03:38:18	desc bank_customer	0.000 sec
2 03:38:25	use bank	0.016 sec
3 03:38:27	desc bank_customer	0.016 sec / 0.000 sec
4 03:45:23	select c.customer_name from bank_customer c where exists(select d.customer_name,count(d.customer_name) from depositer ...)	0.016 sec / 0.000 sec
5 03:47:33	select d.customer_name from depositer d,branch b,`bank_account` a where b.branch_name=a.branch_name AND a.acc_no=d.a...	0.015 sec / 0.000 sec

Query Completed at 03:47 AM on 05-06-2021.

## 5. Demonstrate how you delete all account tuples at every branch located in a specific city (Ex. Bombay).

```
delete from bank_account  
where branch_name IN(  
    select branch_name  
    from branch  
    where branch_city='Bombay'  
);
```

The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```
149 where b.branch_name=a.branch_name  
150 AND a.acc_no=d.acc_no  
151 AND branch_city='Delhi'  
152 group by d.customer_name  
153 HAVING COUNT(DISTINCT b.branch_name)=(  
154     SELECT COUNT(branch_name)  
155         FROM branch  
156         WHERE branch_city='Delhi');  
157 • use bank;  
158 • delete from bank_account  
159 where branch_name IN(  
160     select branch_name  
161     from branch  
162     where branch_city='Bombay'  
163 );
```

The 'PERFOR' tab in the left sidebar is selected. The 'Action Output' section in the bottom right shows the following log entries:

#	Time	Action	Message	Duration / Fetch
1	08:14:51	use bank	0 row(s) affected	0.000 sec
2	08:14:54	delete from bank_account where branch_name IN(select branch_name from branch where branch_city='Bombay')	Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('bank'.'depositor', CONSTRAINT 'depositor_ibfk_2' FOREIGN KEY ('acc_no') REFERENCES 'bank_account' ('acc_no'))	0.015 sec
3	08:19:38	delete from bank_ccount where branch_name IN(select distinct branch_name from branch where branch_city = 'Bombay')	Error Code: 1146. Table 'bank.bank_ccount' doesn't exist	0.000 sec
4	08:19:57	delete from bank_account where branch_name IN(select distinct branch_name from branch where branch_city = 'Bombay')	Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('bank'.'depositor', CONSTRAINT 'depositor_ibfk_2' FOREIGN KEY ('acc_no') REFERENCES 'bank_account' ('acc_no'))	0.000 sec
5	08:26:14	delete from bank_account where branch_name IN(select branch_name)	Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('bank'.'depositor', CONSTRAINT 'depositor_ibfk_2' FOREIGN KEY ('acc_no') REFERENCES 'bank_account' ('acc_no'))	0.000 sec

The status bar at the bottom right indicates the system is at 29% battery, 08:26 AM, 05-06-2021.

NAME:NEELAM H GODIHAL

USN:1BM19CS220

### **PROGRAM 3: SUPPLIER DATABASE**

Consider the following schema:

SUPPLIERS(sid: integer, sname: string, address: string)

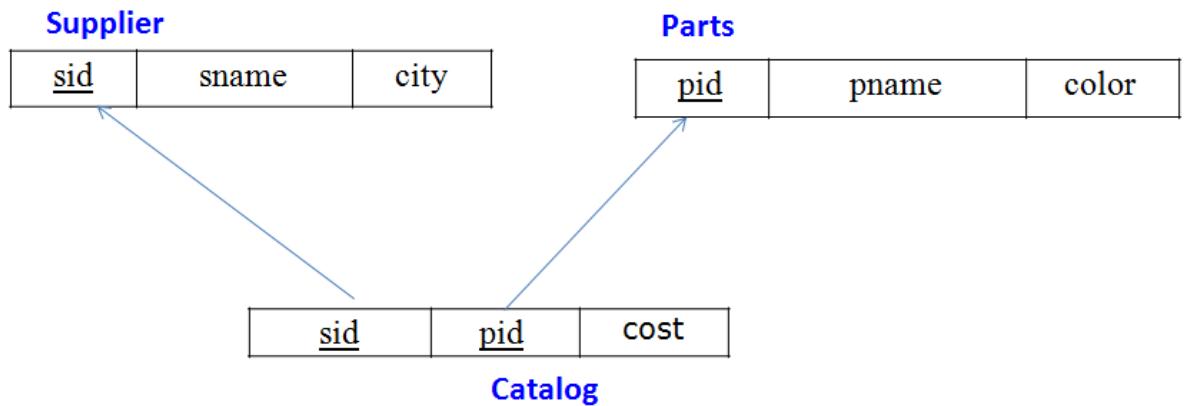
PARTS(pid: integer, pname: string, color: string)

CATALOG(sid: integer, pid: integer, cost: real)

The Catalog relation lists the prices charged for parts by Suppliers.

Write the following queries in SQL:

#### **Schema Diagram**



#### **Table Data**

SUPPLIERS		
SID	SNAME	CITY
-----		
10001	Acme Widget	Bangalore
10002	Johns	Kolkata
10003	Vimal	Mumbai
10004	Reliance	Delhi

PARTS		
PID	PNAME	COLOR
-----		
20001	Book	Red
20002	Pen	Red
20003	Pencil	Green
20004	Mobile	Green
20005	Charger	Black

CATALOG		
SID	PID	COST
-----		
10001	20001	10
10001	20002	10
10001	20003	30
10001	20004	10
10001	20005	10
10002	20001	10
10002	20002	20
10003	20003	30
10004	20003	40

## TABLES:

### 1.supplier

The screenshot shows the MySQL Workbench interface with three tabs open: Local instance MySQL80 (bank), Local instance MySQL80 (sup...), and Local instance MySQL80 (part...). The left sidebar contains navigation panels for MANAGEMENT, INSTANCE, PERFORMANCE, and Administration. The central pane displays a SQL editor with the following code:

```
42 • insert into sup
43   values (10001,'Acme Widget','Bangalore'),
44           (10002,'Johns','Kolkata'),
45           (10003,'Vimal','Mumbai'),
46           (10004,'Reliance','Delhi');
47 • select * from sup;
```

The Result Grid shows the following data:

sid	sname	city
10001	Acme Widget	Bangalore
10002	Johns	Kolkata
10003	Vimal	Mumbai
10004	Reliance	Delhi

The Output pane shows the execution history:

#	Time	Action	Message	Duration / Fetch
9	07:19:19	insert into parts values('Book','Red'), ('20002','Pen','Red'), ('20003','Pencil','Green'), ('20004','Mobile','Black')	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.015 sec
10	07:23:25	insert into catalog values(.,.,.,.,.,.,.,.,.)	Error Code: 1064. 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 '' at line 1	0.000 sec
11	07:25:57	insert into catalog values(10001,20001,10), (10001,20002,10), (10001,20003,30), (10001,20004,10), (10001,20005,30)	9 row(s) affected Records: 9 Duplicates: 0 Warnings: 0	0.000 sec
12	07:26:20	desc sup	3 row(s) returned	0.015 sec / 0.000 sec
13	07:26:34	show table sup	Error Code: 1064. 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 'show table sup' at line 1	0.000 sec
14	07:27:05	select * from sup LIMIT 0, 2000	4 row(s) returned	0.016 sec / 0.000 sec
15	07:27:32	select * from parts LIMIT 0, 2000	5 row(s) returned	0.000 sec / 0.000 sec

### 2.parts

The screenshot shows the MySQL Workbench interface with three tabs open: Local instance MySQL80 (bank), Local instance MySQL80 (sup...), and Local instance MySQL80 (part...). The left sidebar contains navigation panels for MANAGEMENT, INSTANCE, PERFORMANCE, and Administration. The central pane displays a SQL editor with the following code:

```
50 • insert into parts
51   values ('20001','Book','Red'),
52           ('20002','Pen','Red'),
53           ('20003','Pencil','Green'),
54           ('20004','Mobile','Green'),
55           ('20005','Charger','Black');
56 • select * from parts;
```

The Result Grid shows the following data:

pid	pname	color
20001	Book	Red
20002	Pen	Red
20003	Pencil	Green
20004	Mobile	Green
20005	Charger	Black

The Output pane shows the execution history:

#	Time	Action	Message	Duration / Fetch
10	07:23:25	insert into catalog values(.,.,.,.,.,.,.,.,.)	Error Code: 1064. 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 '' at line 1	0.000 sec
11	07:25:57	insert into catalog values(10001,20001,10), (10001,20002,10), (10001,20003,30), (10001,20004,10), (10001,20005,30)	9 row(s) affected Records: 9 Duplicates: 0 Warnings: 0	0.000 sec
12	07:26:20	desc sup	3 row(s) returned	0.015 sec / 0.000 sec
13	07:26:34	show table sup	Error Code: 1064. 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 'show table sup' at line 1	0.000 sec
14	07:27:05	select * from sup LIMIT 0, 2000	4 row(s) returned	0.016 sec / 0.000 sec
15	07:27:32	select * from parts LIMIT 0, 2000	5 row(s) returned	0.000 sec / 0.000 sec

### 3.catalog

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator: Local instance MySQL80 Local instance MySQL80 (bank) Local instance MySQL80 (sup_...
Query 1:
62      (10001,20004,10),
63      (10001,20005,10),
64      (10002,20001,10),
65      (10002,20002,20),
66      (10003,20003,30),
67      (10004,20003,40);
68 • select * from catalog;
Result Grid:
sd pd cost
10001 20001 10
10001 20002 10
10001 20003 30
10001 20004 10
10001 20005 10
10002 20001 10
10002 20002 20
10003 20003 30
10004 20003 40
catalog 9:
Output:
Action Output:
# Time Action Message Duration / Fetch
11 07:25:57 Insert into catalog values(10001,20001,10), (10001,20002,10), (10001,20003,30), (10001,20004,10)... 9 row(s) affected Records: 9 Duplicates: 0 Warnings: 0 0.000 sec
12 07:26:20 desc sup 3 row(s) returned 0.015 sec / 0.000 sec
13 07:26:34 show table sup Error Code: 1064: You have an error in your SQL syntax; check the manual that corresponds to your MySQL se... 0.000 sec
14 07:27:05 select * from sup LIMIT 0, 2000 4 row(s) returned 0.016 sec / 0.000 sec
15 07:27:32 select * from parts LIMIT 0, 2000 5 row(s) returned 0.000 sec / 0.000 sec
16 07:27:59 select * from catalog LIMIT 0, 2000 9 row(s) returned 0.000 sec / 0.000 sec
Object Info Session
Query Completed

```

## QUERIES:

1.Find the pnames of parts for which there is some supplier.

select distinct p.pname

from parts p,catalog c

where p.pid=c.pid;

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator: Local instance MySQL80 Local instance MySQL80 (bank) Local instance MySQL80 (sup_...
Query 1:
68 • select * from catalog;
69
70 #queries
72 • select distinct p.pname
73 from parts p,catalog c
74 where p.pid=c.pid;
Result Grid:
pname
Book
Pen
Pencil
Mobile
Result 12:
Output:
Action Output:
# Time Action Message Duration / Fetch
14 07:27:05 select * from sup LIMIT 0, 2000 4 row(s) returned 0.015 sec / 0.000 sec
15 07:27:32 select * from parts LIMIT 0, 2000 5 row(s) returned 0.000 sec / 0.000 sec
16 07:27:59 select * from catalog LIMIT 0, 2000 9 row(s) returned 0.000 sec / 0.000 sec
17 07:30:45 select * from catalog LIMIT 0, 2000 9 row(s) returned 0.000 sec / 0.000 sec
18 07:43:39 select p.pname from parts p,catalog c where p.pid=c.pid LIMIT 0, 2000 9 row(s) returned 0.000 sec / 0.000 sec
19 07:44:13 select distinct p.pname from parts p,catalog c where p.pid=c.pid LIMIT 0, 2000 5 row(s) returned 0.016 sec / 0.000 sec
Object Info Session
Query Completed

```

## 2.Find the snames of suppliers who supply every part.

```
select s.sname  
from sup s  
where s.sid IN (  
    select c.sid  
    from catalog c  
    group by c.sid having count(c.pid)=  
        select count(p.pid)  
        from parts p)  
);
```

The screenshot shows the MySQL Workbench interface. The query editor window contains the SQL code for finding suppliers who supply every part. The results grid shows a single row with the supplier name 'Acme Widget'. The performance monitor window at the bottom displays two actions: 'use supplier' and the execution of the query, which took 0.093 seconds.

Action	Time	Action	Message	Duration / Fetch
1	03:24:59	use supplier	0 row(s) affected	0.000 sec
2	03:25:03	select sname from sup s where s.sid IN ( select c.sid from catalog c group by c.sid having count(c.pid)= ( select count(p.pid) from parts p) )	1 row(s) returned	0.093 sec / 0.000 sec

## 3.Find the snames of suppliers who supply every red part.

```
select sid,sname  
from sup  
where sid IN (  
    select distinct c.sid
```

```

from catalog c
where c.pid IN (
    select p.pid
        from catalog c,parts p
        where p.color='Red'
)
);

```

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```

80     where sid IN (
81         select distinct c.sid
82         from catalog c
83         where c.pid IN (
84             select p.pid
85             from catalog c,parts p
86             where p.color='Red'
87         )
88     );

```

The results pane shows a table with columns 'sid' and 'sname'. The data is:

sid	sname
10002	Johns
10001	Acme Widget

The status bar at the bottom indicates "Query Completed".

#### 4. Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.

```

select p.pname
        from parts p
        where pid IN(
            select distinct c.pid
                from catalog c
                where sid=(

                    select s.sid

```

```

from sup s
where sname='Acme Widget'
)
);

```

The screenshot shows the MySQL Workbench interface. The Query Editor window contains the following SQL code:

```

113 where pid IN(
114     select distinct c.pid
115     from catalog c
116     where sid=(
117         select s.sid
118         from sup s
119         where sname='Acme Widget'
120     )
121 );

```

The Performance Monitor window below shows the execution plan and statistics for the query:

Action	Time	Action	Message	Duration / Fetch
1	03:24:59	use supplier	0 row(s) affected	0.000 sec
2	03:25:03	select sname from sup s where s.sid IN ( select c.sid from catalog c group by c.sid having count(c.pid)= ( select count(p.pid) fro... ) )	1 row(s) returned	0.093 sec / 0.000 sec
3	03:25:16	select sid,sname from sup where sid IN (select district c.sid from catalog c where c.pid IN ( select p.pid from catalog c.parts p wh... ) )	2 row(s) returned	0.000 sec / 0.000 sec
4	03:25:25	select p.pname from parts p where pid IN(select distinct c.pid from catalog c where sid=(select s.sid from sup s where snam...))	5 row(s) returned	0.000 sec / 0.000 sec

**5. Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part).**

```

select c.sid
from catalog c
where c.cost>(
    select avg(cost)
    from catalog
    where pid=c.pid
);

```

```

128 });
129
130 • use supplier;
131
132
133
134 #q5
135 • select c.sid
136   from catalog c
137   where c.cost>(
138       select avg(cost)
139         from catalog
140        where pid=c.pid
141   );

```

No object selected

catalog 2 x

Action	Time	Message	Duration / Fetch
1 08:16:33 select c.sid from catalog c where c.cost>( select avg(cost) from catalog where pid=c.pid ) LIMIT 0, 2000		Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...	0.016 sec
2 08:16:45 use supplier		0 row(s) affected	0.000 sec
3 08:18:08 select c.sid from catalog c where c.cost>( select avg(cost) from catalog where pid=c.pid ) LIMIT 0, 2000		2 row(s) returned	0.109 sec / 0.000 sec
4 08:29:00 select c.sid from catalog c where c.cost>( select avg(cost) from catalog where pid=c.pid ) LIMIT 0, 2000		2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session Query Completed

**6. For each part, find the sname of the supplier who charges the most for that part.**  
select p.pid, s.sname

from parts p,suppliers s,catalog c

where c.pid=p.pid and c.sid=s.sid and c.cost=

select max(c1.cost)

from catalog c1

where c1.pid=p.pid

);

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports

Administration Schemas

Information

No object selected

Object Info Session

Query 1 x

```
131 }j
132
133 #q6
134 • select p.pid, s.sname
135   from parts p,sup s,catalog c
136   where c.pid=p.pid and c.sid=s.sid and c.cost=(
137     select max(c1.cost)
138     from catalog c1
139     where c1.pid=p.pid
140 )
```

Result Grid Filter Rows Export Wrap Cell Content

pid	sname
20001	Acme Widget
20004	Acme Widget
20005	Acme Widget
20001	Johns
20002	Johns
20003	Reliance

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result 17 x

Action Output

#	Time	Action	Message	Duration / Fetch
19	21:06:17	select c1.sid from catalog c1 where c1.cost = ( select max(c2.cost) from catalog c2 where c1.pid=c2.pid ) ... 6 rows(s) returned		0.016 sec / 0.000 sec
20	21:10:27	select c1.sid from catalog c1,parts p where p.pid=c1.pid and c1.cost = (select max(c2.cost) from catalog c2 ... 6 row(s) returned		0.000 sec / 0.000 sec
21	21:16:18	select c1.sid from catalog c1,parts p where p.pid=c1.pid and c1.cost = (select max(c2.cost) from catalog c2 ... 6 row(s) returned		0.000 sec / 0.000 sec
22	21:42:12	select p.pid,s.sname from parts p,sup s,catalog c where c.pid=p.pid and c.sid=s.sid and c.cost=( select max(c1... 6 row(s) returned		0.390 sec / 0.000 sec
23	21:42:38	select distinct p.pid,distinct s.sname from parts p,sup s,catalog c where c.pid=p.pid and c.sid=s.sid and c.cost<... Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...		0.000 sec
24	21:42:54	select p.pid, s.sname from parts p,up s,catalog c where c.pid=p.pid and c.sid=s.sid and c.cost<( select max(c1... 6 row(s) returned		0.000 sec / 0.000 sec

Read Only Context Help Snippets

Object Info Session

Query Completed

75% 09:44 PM 23-05-2021

**NAME:NEELAM H GODIHAL**

**USN:1BM19CS220**

**PROGRAM 4: STUDENT FACULTY DATABASE**

Consider the following database for student enrollment for course :

STUDENT(snum: integer, sname:string, major: string, lvl: string, age: integer)

CLASS(cname: string, meetsat: time, room: string, fid: integer)

ENROLLED(snum: integer, cname:string)

FACULTY(fid: integer, fname:string, deptid: integer)

The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class. Level(lvl) is a two character code with 4 different values (example: Junior: JR etc)

Write the following queries in SQL.

No duplicates should be printed in any of the answers.

**TABLES:**

1.student

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

MANA

```

44 • insert into student
45     values (1,'Jhon','CS','SR',19),
46             (2,'Smith','CS','JR',20),
47             (3,'Jacob','CV','SR',20),
48             (4,'Tom','CS','JR',20),
49             (5,'Rahul','CS','JR',20),
50             (6,'Rita','CS','SR',21);

```

INSTAN

```

52 • select * from students;
53

```

PERFO

Result Grid | Filter Rows | Edits | Export/Import | Wrap Cell Content | Result Grid | Form Editor

srno	sname	major	lv	age
1	Jhon	CS	SR	19
2	Smith	CS	JR	20
3	Jacob	CV	SR	20
4	Tom	CS	JR	20
5	Rahul	CS	JR	20
6	Rita	CS	SR	21

student 8

Output

Action Output

#	Time	Action	Message	Duration / Fetch
13	00:37:36	desc class	4 row(s) returned	0.016 sec / 0.000 sec
14	00:39:51	create table enrolled( snum int, cname varchar(30), primary key(snum,cname), foreign key(snum)references student(snu... Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for ... 0.016 sec	0 row(s) affected	0.016 sec
15	00:40:01	create table enrolled( snum int, cname varchar(30), primary key(snum,cname), foreign key(snum)references student(snu... 0.047 sec	2 row(s) returned	0.016 sec / 0.000 sec
16	00:40:15	desc enrolled	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0 0.015 sec	0.015 sec
17	00:44:17	insert into student values (1,Jhon,CS,SR,19), (2,Smith,CS,JR,20), (3,Jacob,CV,SR,20), (4,Tom,CS,JR,20), ... 6 row(s) returned	6 row(s) returned	0.000 sec / 0.000 sec
18	00:44:34	select * from student LIMIT 0, 2000		

Query Completed

Windows Taskbar: 77%, 12:44 AM, ENG, 04-06-2021, 3

## 2.faculty

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

MANA

```

53
54 • insert into faculty
55     values (12,'Harish',1000),
56             (12,'MV',1000),
57             (13,'Mira',1001),
58             (14,'Shiva',1002),
59             (15,'Nupur',1000);

```

INSTAN

```

60
61 • select * from faculty;

```

PERFO

Result Grid | Filter Rows | Edits | Export/Import | Wrap Cell Content | Result Grid | Form Editor

fd	fname	dept
11	Harish	1000
12	MV	1000
13	Mira	1001
14	Shiva	1002
15	Nupur	1000

faculty 9

Output

Action Output

#	Time	Action	Message	Duration / Fetch
15	00:40:01	create table enrolled( snum int, cname varchar(30), primary key(snum,cname), foreign key(snum)references student(snu... 0 row(s) affected	0.047 sec	
16	00:40:15	desc enrolled	2 row(s) returned	0.016 sec / 0.000 sec
17	00:44:17	insert into student values (1,Jhon,CS,SR,19), (2,Smith,CS,JR,20), (3,Jacob,CV,SR,20), (4,Tom,CS,JR,20), ... 6 row(s) returned Records: 6 Duplicates: 0 Warnings: 0 0.015 sec	6 row(s) returned	0.000 sec / 0.000 sec
18	00:44:34	select * from student LIMIT 0, 2000	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.015 sec	5 row(s) returned
19	00:48:20	insert into faculty values (11,Harish,1000), (12,MV,1000), (13,Mira,1001), (14,Shiva,1002), (15,Nupur,1000)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.015 sec	
20	00:48:38	select * from faculty LIMIT 0, 2000	5 row(s) returned	0.000 sec / 0.000 sec

Query Completed

Windows Taskbar: 75%, 12:48 AM, ENG, 04-06-2021, 3

## 3.class

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator Query 1

MANA

```

66 ('class1', '2012-11-15 10:15:16', 'R128', 14),
67 ('class2', '2012-11-15 10:15:20', 'R2', 12),
68 ('class3', '2012-11-15 10:15:25', 'R3', 11),
69 ('class4', '2012-11-15 20:15:20', 'R4', 14),
70 ('class5', '2012-11-15 20:15:20', 'R3', 15),

```

INSTAP

	name	meets_at	room	fid
class1	2012-11-15 10:15:16	R1	14	
class2	2012-11-15 10:15:20	R2	12	
class3	2012-11-15 10:15:25	R3	11	
class4	2012-11-15 20:15:20	R4	14	
class5	2012-11-15 20:15:20	R3	15	
class6	2012-11-15 13:20:20	R3	14	
class7	2012-11-15 10:10:10	R3	14	

PERFOR

Information

Result Grid Filter Rows Edits Export/Import Wrap Cell Content

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid Form Editor Field Types Query Stats

Context Help Snippets

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
1	10:07:23	insert into class values ('class1','2012-11-15 10:15:16',R1,14), ('class2','2012-11-15 10:15:20',R2,12), ('class3','2012-11-15 10:15:25',R3,11), ('class4','2012-11-15 20:15:20',R4,14), ('class5','2012-11-15 20:15:20',R3,15)	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCHEMAS list in the left pane.	0.000 sec
2	10:07:31	use student	0 row(s) affected	0.000 sec
3	10:07:36	insert into class values ('class1','2012-11-15 10:15:16',R1,14), ('class2','2012-11-15 10:15:16',R128,14), ('class3','2012-11-15 10:15:20',R2,12), ('class4','2012-11-15 20:15:20',R3,11), ('class5','2012-11-15 20:15:20',R4,14)	8 row(s) affected Records: 8 Duplicates: 0 Warnings: 0	0.016 sec
4	10:07:51	select * from class LIMIT 0, 2000	0 row(s) returned	0.000 sec / 0.000 sec

Query Completed

97% 10:08 AM ENG 04-06-2021 (3)

#### 4.enrolled

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator Query 1

```

84 ('class1'),
85 ('class2'),
86 ('class3'),
87 ('class4'),
88 select * from enrolled;

```

INSTAP

	shnum	name
1	1	class1
2	2	class1
3	3	class3
4	4	class3
5	5	class4
1	1	class5
2	2	class5
3	3	class5
4	4	class5
5	5	class5

enrolled 2

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
2	10:07:31	use student	0 row(s) affected	0.000 sec
3	10:07:36	insert into class values ('class1','2012-11-15 10:15:16',R1,14), ('class2','2012-11-15 10:15:16',R128,14), ('class3','2012-11-15 10:15:20',R2,12), ('class4','2012-11-15 20:15:20',R3,11), ('class5','2012-11-15 20:15:20',R4,14)	8 row(s) affected Records: 8 Duplicates: 0 Warnings: 0	0.016 sec
4	10:07:51	select * from class LIMIT 0, 2000	8 row(s) returned	0.000 sec / 0.000 sec
5	10:14:58	insert into enrolled values ('1', '1', '1', '1', '1', '1')	Error Code: 1064. 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', '1', '1', '1', '1', '1' at line 1	0.000 sec
6	10:17:53	insert into enrolled values ('1,class1'), ('2,class1'), ('3,class3'), ('4,class3'), ('5,class4'), ('1,class5'), ('2,class5')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec
7	10:18:08	select * from enrolled LIMIT 0, 2000	10 row(s) returned	0.000 sec / 0.000 sec

Query Completed

97% 10:19 AM ENG 04-06-2021 (3)

## QUERIES:

**1. Find the names of all Juniors (level = JR) who are enrolled in a class taught by “Harish”.**

```
select s.sname
from student s,enrolled e
where s.snum=e.snum and cname IN(
    select c1.cname
    from class c1,faculty f1
    where c1.fid=f1.fid and fname='Harish'
) and lvl='JR';
```

The screenshot shows the MySQL Workbench interface with several tabs open. The main query editor window contains the SQL code for finding Juniors enrolled in classes taught by Harish. The results grid shows one row with the name 'Tom'. Below the results, the 'Output' pane displays the execution log, which includes actions like inserting into 'enrolled' and 'class' tables, and a warning message about an error in the SQL syntax. The status bar at the bottom right shows the date and time as 04-06-2021 10:30 AM.

**2. Find the names of all classes that either meet in room R128 or have five or more Students enrolled.**

```
select c cname
from class c
where c.room='R128' OR c cname IN (
    select e cname
    from enrolled e
    group by e cname
    having count(*)>=5
);
```

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator Query 1 ×
MANA Local instance MySQL80 × Local instance MySQL80 × Local instance MySQL80 × Local instance MySQL80 × Local instance MySQL80 (stu...
INSTA
PERFO
INFOR
Result Grid Filter Rows: | Export/Import | Wrap Cell Content: | Result Grid
Output Action Output
# Time Action Message Duration / Fetch
4 10:07:51 select * from class LIMIT 0, 2000 8 row(s) returned 0.000 sec / 0.000 sec
5 10:14:58 insert into enrolled values (''), (''), (''), (''), ('')
6 10:17:53 insert into enrolled values (1,class1), (2,class1), (3,class3), (4,class3), (5,class4), (1,class5), (2,class5)
7 10:18:08 select * from enrolled LIMIT 0, 2000 10 row(s) returned 0.000 sec / 0.000 sec
8 10:30:49 select s.sname from student s, enrolled e where s.snum=e.snum and cname IN(select c1.cname from class c1,faculty f1 where ...
9 10:41:30 select c.cname from class c where c.room='R120' OR c.cname IN( select e.cname from enrolled e group by e.cname h...
< > 9 10:41:30 select c.cname from class c where c.room='R120' OR c.cname IN( select e.cname from enrolled e group by e.cname h...
Query Completed

```

**3. Find the names of all students who are enrolled in two classes that meet at the same time.**

```

select distinct s.sname
from student s
where s.snum IN(
    select e1.snum
    from enrolled e1,enrolled e2,class c1,class c2
    where e1.snum=e2.snum and e1 cname <> e2 cname
    and e1 cname=c1 cname
    and e2 cname=c2 cname and c1.meets_at=c2.meets_at
);

```

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator Query 1 ×
MANA Local instance MySQL80 × Local instance MySQL80 × Local instance MySQL80 × Local instance MySQL80 × Local instance MySQL80 (stu...
INSTA
PERFO
INFOR
Result Grid Filter Rows: | Export/Import | Wrap Cell Content: | Result Grid
Output Action Output
# Time Action Message Duration / Fetch
5 10:14:58 insert into enrolled values (''), (''), (''), (''), ('')
6 10:17:53 insert into enrolled values (1,class1), (2,class1), (3,class3), (4,class3), (5,class4), (1,class5), (2,class5)
7 10:18:08 select * from enrolled LIMIT 0, 2000 10 row(s) returned 0.000 sec / 0.000 sec
8 10:30:49 select s.sname from student s, enrolled e where s.snum=e.snum and cname IN(select c1.cname from class c1,faculty f1 where ...
9 10:41:30 select c.cname from class c where c.room='R120' OR c.cname IN( select e.cname from enrolled e group by e.cname h...
< > 10 11:57:01 select distinct s.sname from student s where s.snum IN(select e1.snum from enrolled e1,enrolled e2,class c1,class c2 wher...
Query Completed

```

**4.Find the names of faculty members who teach in every room in which some class is taught.**

```
select fname
from faculty where fid=(  
    select c1.fid  
    from class c1  
    group by c1.fid  
    having count(distinct room)=(  
        select count(distinct c.room)  
        from class c)  
);
```

The screenshot shows the MySQL Workbench interface. The SQL editor window contains the following query:

```
120
121 #q4
122
123 • select fname
124   from faculty where fid=(
125     select c1.fid
126     from class c1
127     group by c1.fid
128     having count(distinct room)=(
129       select count(distinct c.room)
130         from class c)
131 );
```

The results grid shows the output:

fname
Shiva

The status bar at the bottom right indicates the date and time as 04-06-2021 02:21 PM.

**5.Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.**

```
select distinct f.fname
from faculty f
where 5>(  
    select count(e.snum)
    from class c,enrolled e
    where c cname=e cname and c fid=f fid
);
```

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

MANA

```

133 #q5
134 • use student;
135 • select distinct f.fname
136   from faculty f
137   where s>(
138     select count(e.snum)
139       from class c,enrolled e
140      where c.cname=e.cname and c.fid=f.fid
141 );

```

INSTAP

PERFOR

Information

Result Grid | Filter Rows: [ ] Export: [ ] Wrap Cell Content: [ ]

faculty 17 x

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
29	23:26:47	select age as AGE from student	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for...	0.000 sec
30	23:26:53	select age as AGE from student LIMIT 0, 2000	6 row(s) returned	0.015 sec / 0.000 sec
31	23:27:03	select age AGE from student LIMIT 0, 2000	6 row(s) returned	0.000 sec / 0.000 sec
32	23:27:10	select age me from student LIMIT 0, 2000	6 row(s) returned	0.000 sec / 0.000 sec
< ->	33 03:00:25	use student	0 row(s) affected	0.000 sec
< ->	34 03:00:28	select distinct f fname from faculty f where s>( select count(e.snum) from class c,enrolled e ... where c.cname=e.cname and c.fid=f.fid )	4 row(s) returned	0.000 sec / 0.000 sec

Query Completed

## 6.Find the names of students who are not enrolled in any class.

```

select distinct s.sname
from student s
where s.snum NOT IN(
    select e.snum
    from enrolled e
);

```

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

MANA

```

116
117   from enrolled e,student s, class c1, class c2
118   where e1.snum=e2.snum and e1.cname <> e2.cname
119   and e1.cname=c1.cname
120   and e2.cname=c2.cname and c1.meets_at=c2.meets_at
121
122 #q6
123
124 • select distinct s.sname
125   from student s
126   where s.snum NOT IN(
127     select e.snum
128   );

```

INSTAP

PERFOR

Information

Result Grid | Filter Rows: [ ] Export: [ ] Wrap Cell Content: [ ]

student 6 x

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
6	10:17:53	Insert into enrolled values (1,class1), (2,class1), (3,class3), (4,class3), (5,class4), (1,class5), (2,class5)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec
7	10:18:08	select * from enrolled LIMIT 0, 2000	10 row(s) returned	0.000 sec / 0.000 sec
8	10:30:49	select s.sname from student s where s.snum=e.snum and cname IN(select c1.cname from class c1,faculty f1 where ...)	1 row(s) returned	0.000 sec / 0.000 sec
9	10:41:30	select c.cname from class c where c.room=R126 OR c.cname IN (select e.cname ... from enrolled e group by e.cname ...)	2 row(s) returned	0.000 sec / 0.000 sec
10	11:57:01	select distinct s.sname from student s where s.snum IN(select e1.snum ... from enrolled e1,student s1, class c1, class c2 where ...)	1 row(s) returned	0.015 sec / 0.000 sec
< ->	11 12:05:23	select distinct s.sname from student s where s.snum NOT IN(select e.snum ... from enrolled e ) LIMIT 0, 2000	1 row(s) returned	0.000 sec / 0.000 sec

Query Completed

## 7.For each age value that appears in Students, find the level value that appears most often. For example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR).

```

SELECT S.age, S.lvl
FROM student S
GROUP BY S.age, S.lvl
HAVING S.lvl IN (
    SELECT S1.lvl FROM student S1 WHERE S1.age = S.age GROUP BY S1.lvl,
S1.age
    HAVING COUNT(*) >= ALL (
        SELECT COUNT(*)
        FROM student S2
        WHERE S1.age = S2.age
        GROUP BY S2.lvl, S2.age)
);

```

The screenshot shows the MySQL Workbench interface. The query editor pane contains the SQL code provided above. The results pane shows a table with two rows:

age	lvl
19	SR
20	JR
21	SR

The output pane at the bottom displays the following log entries:

#	Action	Time	Message	Duration / Fetch
21	SELECT S.age, S.lvl FROM student S GROUP BY S.age, S.lvl HAVING S.lvl IN (	16:58:29	SELECT S1 lvl FROM student S1 WHERE ... Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for ...	0.000 sec
22	SELECT S.age, S.lvl FROM student S GROUP BY S.age, S.lvl HAVING S.lvl IN (SELECT S1 lvl FROM student S1 WHERE S...	17:06:46	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for ...	0.000 sec
23	use student;	17:07:26	0 row(s) affected	0.000 sec
24	SELECT S.age, S.lvl FROM student S GROUP BY S.age, S.lvl HAVING S.lvl IN (SELECT S1 lvl FROM student S1 WHERE S...	17:07:31	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for ...	0.000 sec
25	SELECT S.age, S.lvl FROM student S GROUP BY S.age, S.lvl HAVING S.lvl IN (SELECT S1 lvl FROM student S1 WHERE S...	17:12:04	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for ...	0.000 sec
26	SELECT S.age, S.lvl FROM student S GROUP BY S.age, S.lvl HAVING S.lvl IN (SELECT S1 lvl FROM student S1 WHERE S...	17:13:33	3 row(s) returned	0.016 sec / 0.000 sec

NAME:NEELAM H GODIHAL

USN:1BM19CS220

## PROGRAM 5: AIRLINE FLIGHT DATABASE

Consider the following database that keeps track of airline flight information:

FLIGHTS(flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time, price: integer)

AIRCRAFT(aid: integer, aname: string, cruisingrange: integer)

CERTIFIED(eid: integer, aid: integer)

EMPLOYEES(eid: integer, ename: string, salary: integer)

Note that the Employees relation describes pilots and other kinds of employees as well; Every pilot is certified for some aircraft, and only pilots are certified to fly.

Write each of the following queries in SQL.

## TABLES:

### 1.flights

The screenshot shows the MySQL Workbench interface with the 'Query' tab active. The code entered is:

```
9    distance int,
10   departs timestamp,
11   arrives timestamp,
12   price int
13  );
14
15 • desc flights;
```

The 'Result Grid' pane displays the structure of the 'flights' table:

Field	Type	Null	Key	Default	Extra
flno	int	NO	PRI		
from	varchar(30)	NO			
to	varchar(30)	NO			
distance	int	YES			
departs	timestamp	YES			
arrives	timestamp	YES			
price	int	YES			

The 'Output' pane shows the results of the 'desc flights;' command:

#	Time	Action	Message	Duration / Fetch
1	19:59:30	create database flight	1 row(s) affected	0.562 sec
2	19:59:40	use flight	0 row(s) affected	0.031 sec
3	20:17:22	create table flights(flno int primary key, from varchar(30) not null, to varchar(30) not null, distance int, departs timestamp, arrives timestamp, price int);	0 row(s) affected	3.500 sec
4	20:18:54	desc flights	7 row(s) returned	0.140 sec / 0.000 sec

### 2.aircraft

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: MANAGEMENT INSTANCE PERFORMANCE Information No object selected

Query 1:

```

46      (103,'647',800),
47      (104,'Dreamliner',10000),
48      (105,'Boeing',3500),
49      (106,'707',1500),
50      (107,'Dream',120000);
51
52 • select * from aircraft;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Contents: |

aid	aname	cruising_range
101	747	3000
102	Boeing	900
103	647	800
104	Dreamliner	10000
105	Boeing	3500
106	707	1500
107	Dream	120000

SQLAdditions | Jump to | Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

aircraft 5:

Action Output

#	Time	Action	Message	Duration / Fetch
8	20:25:20	desc employee	3 row(s) returned	0.129 sec / 0.000 sec
9	20:31:28	create table certified(eid int, aid int, primary key(eid,aid), foreign key(aid) references aircraft(aid), fore...	Error Code: 3734. Failed to add the foreign key constraint. Missing column 'eid' for constraint 'certified_ibfk_1' in...	0.203 sec
10	20:32:07	create table certified(eid int, aid int, primary key(eid,aid), foreign key(aid) references aircraft(aid), fore...	0 row(s) affected	0.438 sec
11	20:32:20	desc certified	2 row(s) returned	0.093 sec / 0.000 sec
12	20:52:52	Insert into aircraft values(101,747,3000), (102,Boeing',900), (103,647,800), (104,Dreamliner',10000...)	7 row(s) affected Records: 7 Duplicates: 0 Warnings: 0	0.422 sec
13	20:53:31	select * from aircraft LIMIT 0, 2000	7 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

Windows Taskbar: 48% ENG 08:53 PM 22-05-2021

### 3.employee

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: MANAGEMENT INSTANCE PERFORMANCE Administration Schemas Information No object selected

Query 1:

```

57      (703,'C',150000),
58      (704,'D',90000),
59      (705,'E',40000),
60      (706,'F',50000),
61      (707,'G',90000);
62
63 • select * from employee;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Contents: |

eid	ename	salary
701	A	50000
702	B	100000
703	C	150000
704	D	90000
705	E	40000
706	F	60000
707	G	90000

SQLAdditions | Jump to | Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

employee 6:

Action Output

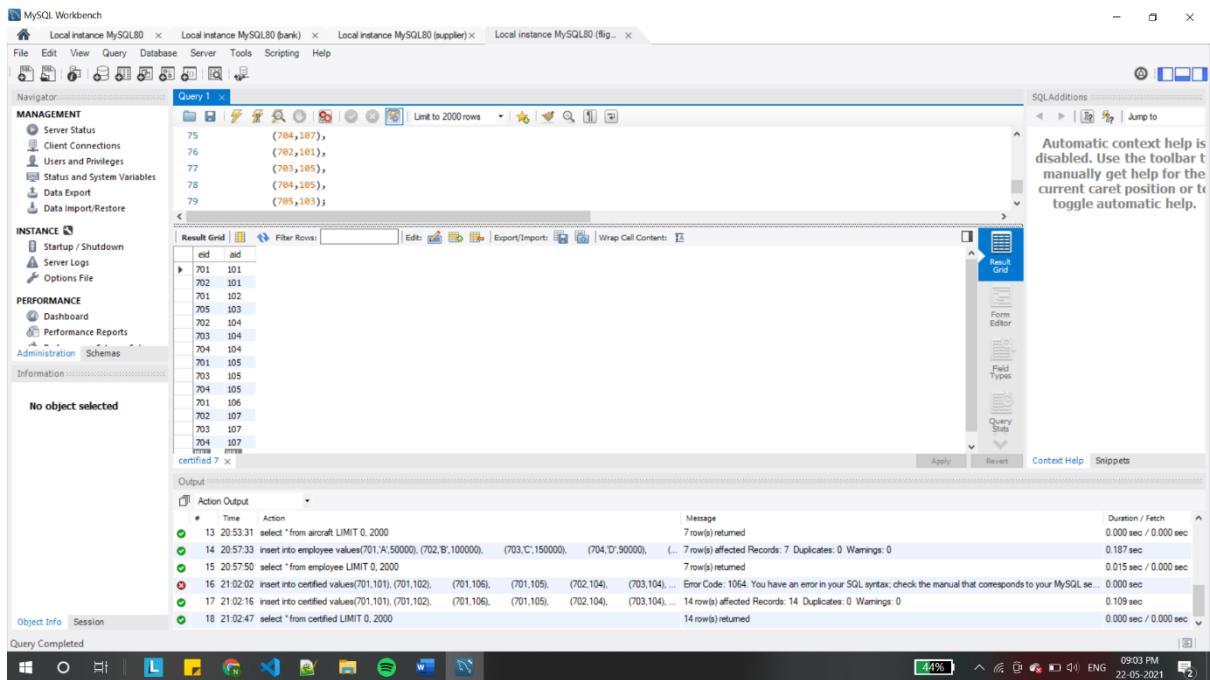
#	Time	Action	Message	Duration / Fetch
10	20:32:07	create table certified(eid int, aid int, primary key(eid,aid), foreign key(aid) references aircraft(aid), fore...	0 row(s) affected	0.438 sec
11	20:32:20	desc certified	2 row(s) returned	0.093 sec / 0.000 sec
12	20:52:52	Insert into aircraft values(101,747,3000), (102,Boeing',900), (103,647,800), (104,Dreamliner',10000...)	7 row(s) affected Records: 7 Duplicates: 0 Warnings: 0	0.422 sec
13	20:53:31	select * from aircraft LIMIT 0, 2000	7 row(s) returned	0.000 sec / 0.000 sec
14	20:57:33	insert into employee values(701,'A',50000), (702,'B',100000), (703,'C',150000), (704,'D',90000), (...)	7 row(s) affected Records: 7 Duplicates: 0 Warnings: 0	0.187 sec
15	20:57:50	select * from employee LIMIT 0, 2000	7 row(s) returned	0.015 sec / 0.000 sec

Object Info Session

Query Completed

Windows Taskbar: 46% ENG 08:57 PM 22-05-2021

### 4.certified



## QUERIES:

**1. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.**

```
select distinct a.aname
```

```
from aircraft a
```

```
where aid IN (
```

```
    select c.aid
```

```
    from certified c,employee e
```

```
    where c.eid=e.eid and salary>80000
```

```
);
```

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1

```

MANA: 95
      (107,'Bangalore','Frankfurt',8000,'2005-05-13 07:15:31','2005-05-13 22:15:31',60000),
      (108,'Bangalore','Frankfurt',8500,'2005-05-13 07:15:31','2005-05-13 23:15:31',75000),
      (105,'Kolkata','Delhi',3400,'2005-05-13 07:15:31','2005-05-13 09:15:31',70000);

INSTAP: 99 • select * from flights;
100
101 • use flight;
102
103 • select distinct a.aname
104   from aircraft a

```

PERFO: Result Grid | Filter Rows | Export: | Wrap Cell Content: | SQLAdditions |

Output: **aircraft 6**

aname
747
Dreamliner
Boeing
Dream

Action Output

#	Time	Action	Message	Duration / Fetch
3	18:51:25	select a.aname from aircraft a.certified c.employee e where c.eid=e.eid and e.salary>80000 LIMIT 0, 2000	63 row(s) returned	0.032 sec / 0.000 sec
4	18:52:23	select a.aname from aircraft a.certified c.employee e where c.eid=e.eid and e.salary>80000 LIMIT 0, 2000	63 row(s) returned	0.000 sec / 0.000 sec
5	18:57:58	select distinct a.aname from aircraft a where aid IN (select c.aid from certified c.employee e where c.eid=e.eid and salary>... 4 row(s) returned		0.000 sec / 0.000 sec
6	19:00:57	select distinct a.aname from aircraft a where aid IN (select c.aid from certified c.employee e where c.eid=e.eid and salary>... 4 row(s) returned		0.000 sec / 0.000 sec
7	19:01:50	select distinct a.aname from aircraft a where aid IN (select c.aid from certified c.employee e where c.eid=e.eid and salary>... 4 row(s) returned		0.000 sec / 0.000 sec
8	19:02:05	select distinct a.aname from aircraft a where aid IN (select c.aid from certified c.employee e where c.eid=e.eid and salary>... 4 row(s) returned		0.000 sec / 0.000 sec

85% | 8:47 PM | ENG | 03-06-2021 | 3)

2. For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruising range of the aircraft for which she or he is certified.

select c.eid,max(a.cruising\_range)

from certified c,aircraft a

where c.aid=a.aid

group by c.eid

having count(\*)>3;

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1

```

MANA: 105
      from aircraft a
      where aid IN (
      106       select c.aid
      107         from certified c.employee e
      108         where c.eid=e.eid and salary>80000
      109     );
      110
      111
      112 #q2
      113 • select c.eid,max(a.cruising_range)
      114   from certified c.aircraft a

```

PERFO: Result Grid | Filter Rows | Export: | Wrap Cell Content: | SQLAdditions |

Output: **Result 7**

eid	max(a.cruising_range)
701	3500

Action Output

#	Time	Action	Message	Duration / Fetch
4	18:52:23	select a.aname from aircraft a.certified c.employee e where c.eid=e.eid and e.salary>80000 LIMIT 0, 2000	63 row(s) returned	0.000 sec / 0.000 sec
5	18:57:58	select distinct a.aname from aircraft a where aid IN (select c.aid from certified c.employee e where c.eid=e.eid and salary>... 4 row(s) returned		0.000 sec / 0.000 sec
6	19:00:57	select distinct a.aname from aircraft a where aid IN (select c.aid from certified c.employee e where c.eid=e.eid and salary>... 4 row(s) returned		0.000 sec / 0.000 sec
7	19:01:50	select distinct a.aname from aircraft a where aid IN (select c.aid from certified c.employee e where c.eid=e.eid and salary>... 4 row(s) returned		0.000 sec / 0.000 sec
8	19:02:05	select distinct a.aname from aircraft a where aid IN (select c.aid from certified c.employee e where c.eid=e.eid and salary>... 4 row(s) returned		0.000 sec / 0.000 sec
9	19:47:27	select c.eid,max(a.cruising_range) from certified c.aircraft a where c.eid=a.aid group by c.eid having count(*)>3 LIMIT 0, 2000	1 row(s) returned	0.000 sec / 0.000 sec

90% | 8:47 PM | ENG | 03-06-2021 | 3)

**3. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.**

```
select distinct e.ename  
from employee e  
where e.salary < (  
    select min(f.price)  
    from flights f  
    where f.from_ = 'Bangalore' AND f.to_ = 'Frankfurt'  
);
```

The screenshot shows the MySQL Workbench interface. In the Query Editor, there are two queries. The first query is:

```
113 • select c.eid,max(a.cruising_range)  
114   from certified c,aircraft a  
115   where c.eid=a.aid  
116   group by c.eid  
117   having count(*)>3;  
118  
119 #q3  
120  
121 • select distinct e.ename  
122   From emolovee e
```

The second query is:

```
121 • select distinct e.ename  
From emolovee e
```

In the Results Grid, the output is:

ename
A
E

Below the results grid is the Output window, which shows the execution log:

Action	Time	Action	Message	Duration / Fetch
9	19:47:27	select c.eid,max(a.cruising_range) from certified c,aircraft a where c.eid=a.aid group by c.eid having count(*)>3 LIMIT 0, 2000	1 row(s) returned	0.000 sec / 0.000 sec
10	19:53:03	select e.ename from employee e where e.salary < (select min(f.price) from flights f where f.from_ = 'Bangalore' AND f.to_ = 'Frankfurt')	Error Code: 1054. Unknown column 'f.from' in 'where clause'	0.016 sec
11	19:53:30	select distinct e.ename from employee e where e.salary < (select min(f.price) from flights f where f.from_ = 'Bangalore' AND f.to_ = 'Frankfurt')	Error Code: 1054. Unknown column 'f.from' in 'where clause'	0.000 sec
12	19:54:31	select distinct e.ename from employee e where e.salary < (select min(f.price) from flights f where f.from_ = 'Bangalore' AND f.to_ = 'Frankfurt')	Error Code: 1054. Unknown column 'f.from' in 'where clause'	0.000 sec
13	20:13:13	select distinct e.ename from employee e where e.salary < ANY(select min(f.price) from flights f where f.from_ = 'Bangalore' AND f.to_ = 'Frankfurt')	Error Code: 1054. Unknown column 'f.from' in 'where clause'	0.000 sec
14	20:14:35	select distinct e.ename from employee e where e.salary < (select min(f.price) from flights f where f.from_ = 'Bangalore' AND f.to_ = 'Frankfurt')	2 row(s) returned	0.000 sec / 0.000 sec

**4. For all aircraft with cruisingrange over 1000 Kms, find the name of the aircraft and the average salary of all pilots certified for this aircraft.**

```
select a.aname,a.cruising_range,avg(e.salary)  
from aircraft a,employee e,certified c  
where c.eid = e.eid and c.aid = a.aid
```

The screenshot shows the MySQL Workbench interface. In the top-left pane, there are five tabs labeled "Local instance MySQL80". The main area contains a query editor with the following SQL code:

```

138     from class c)
139  ); #rong
140
141 #q4m
142
143 • use flight;
144 • select a.aname,a.cruising_range,avg(e.salary)
145 • from aircraft a,employee e,certified c
146 where c.eid = e.eid and c.aid = a.aid
147 group by a.aname having a.cruising_range > 1000 ;

```

The results grid shows the following data:

aname	cruising_range	avg(salary)
700	1500	50000.0000
Boeing	3500	85000.0000
747	3000	75000.0000
Dream	120000	113333.3333
Dreamliner	10000	113333.3333

The bottom pane shows the "Output" tab with the following log entries:

#	Time	Action	Message	Duration / Fetch
3	08:10:59	select a.aname,a.cruisingrange,avg(e.salary) from aircraft a,employee e,certified c where c.eid = e.eid and c.aid = a.aid	gr... Error Code: 1146. Table 'flight.employee' doesn't exist	0.015 sec
4	08:11:25	select a.aname,a.cruisingrange,avg(e.salary) from aircraft a,employee e,certified c where c.eid = e.eid and c.aid = a.aid	gro... Error Code: 1054. Unknown column 'a.cruisingrange' in field list'	0.000 sec
5	08:11:36	use flight	0 row(s) affected	0.000 sec
6	08:11:42	select a.aname,a.cruisingrange,avg(e.salary) from aircraft a,employee e,certified c where c.eid = e.eid and c.aid = a.aid	gro... Error Code: 1054. Unknown column 'a.cruisingrange' in field list'	0.000 sec
7	08:11:51	use flight	0 row(s) affected	0.000 sec
8	08:11:56	select a.aname,a.cruising_range,avg(e.salary) from aircraft a,employee e,certified c where c.eid = e.eid and c.aid = a.aid	gr... 5 row(s) returned	0.000 sec / 0.000 sec

## 5.Find the names of pilots certified for some Boeing aircraft.

select distinct e.ename

from aircraft a2,certified c,employee e

where c.eid=e.eid and c.aid IN (

select a1.aid

from aircraft a1

where aname like 'Boeing%'

);

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator Query 1 SQLAdditions

MANA: 129 • select a2.ename,avg(e.salary)  
from aircraft a2,employee e  
130     where a2.aid IN(select a1.aid  
131         from aircraft a1  
132         where cruising\_range>1000);  
133  
134

INSTAP: 135  
136     #q5  
137 • select distinct e.ename  
138     from aircraft a2,certified c,employee e  
139

PERFO:

Result Grid | Filter Rows | Export | Wrap Cell Content:

ename
D
C
A

Result Grid Form Editor Field Types Context Help Snippets

Output:

Action	Time	Action	Message	Duration / Fetch
14	20:14:35	select distinct e.ename from employee e where e.salary<( select min(f.price) from flights f where f.from_="Bangalore" AND f.to_="Delhi")	2 row(s) returned	0.000 sec / 0.000 sec
15	23:04:39	select a2.ename,avg(e.salary) from aircraft a2,employee e where a2.aid IN(select a1.aid from aircraft a1 where cruising_range>1000)	1 row(s) returned	0.031 sec / 0.000 sec
16	23:24:31	select e.ename from aircraft a2,certified c,employee e where c.eid=e.eid and c.aid IN (select a1.aid from aircraft a1 where a1.ename='D')	Error Code: 1242. Subquery returns more than 1 row	0.000 sec
17	23:25:16	select e.ename from aircraft a2,certified c,employee e where c.eid=e.eid and c.aid IN (select a1.aid from aircraft a1 where a1.ename='D')	28 row(s) returned	0.000 sec / 0.000 sec
18	23:25:34	select distinct e.ename from aircraft a2,certified c,employee e where c.eid=e.eid and c.aid IN (select a1.aid from aircraft a1 where a1.ename='D')	3 row(s) returned	0.016 sec / 0.000 sec
19	23:26:15	select distinct e.ename from aircraft a2,certified c,employee e where c.eid=e.eid and c.aid IN (select a1.aid from aircraft a1 where a1.ename='D')	3 row(s) returned	0.000 sec / 0.000 sec

86% 11:26 PM ENG 03-06-2021 3)

## 6.Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.

select a.aid

from aircraft a

where a.cruising\_range >(

    select min(f.distance)

    from flights f

    where f.from\_="Bangalore" AND f.to\_="Delhi"

);

The screenshot shows the MySQL Workbench interface. In the top navigation bar, there are five tabs: Local instance MySQL80, Local instance MySQL80, Local instance MySQL80, Local instance MySQL80 (fig...), and Local instance MySQL80 (student). The main area has a toolbar with various icons. Below the toolbar is a 'Navigator' section with 'MANA', 'INSTAP', and 'PERFOR' tabs. The 'INSTAP' tab is selected, showing a tree view of tables: flights, aircraft, and flight\_employee. The 'Result Grid' pane displays a table with columns 'aid' and 'aid' containing values 101, 104, 105, and 107. The 'Performance' pane at the bottom shows a timeline of actions with their times, messages, and durations. A message in the top right corner says: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.'

```

153
154 • use flight;
155
156 • select * from flights;
157
158 • select a.aid
  from aircraft a
 where a.cruising_range >(
   select min(f.distance)
 )

```

aid	aid
101	
104	
105	
107	

aircraft 7 x

Action Output

#	Time	Action	Message	Duration / Fetch
9	17:35:00	select a.aid from aircraft a where a.cruising_range >( select min(f.distance) from flights f where f.from_=>"Bangalore" AND f.to_=>"D..."	4 row(s) returned	0.015 sec / 0.000 sec
10	18:06:51	select depart from flights f where f.to_=>"Delhi" and arrives<="18:00:00" and f.from_=>"Bangalore" union (select depart from fl..."	1 row(s) returned	0.000 sec / 0.000 sec
11	02:29:47	select depart from flights f where f.to_=>"Kolkata" and arrives<="18:00:00" and f.from_=>"Bangalore" union select depart fr..."	1 row(s) returned	0.016 sec / 0.000 sec
12	02:34:15	select a2.aname,avg(e.salary) from aircraft a2,employee e where a2.aid IN(select a.aid from aircraft a1 where cruising_range>..."	1 row(s) returned	0.000 sec / 0.000 sec
13	02:38:50	SELECT Temp.name, Temp.AvgSalary FROM (SELECT A.aid, A.aname AS name, AVG(E.salary) AS AvgSalary FROM Aircraft A, Employee E WHERE A.aid = E.aid) AS Temp	Error Code: 1146. Table 'flight.employee' doesn't exist	0.000 sec
14	02:43:35	select a.aid from aircraft a where a.cruising_range > ( select min(f.distance) from flights f where f.from_=>"Bangalore" AND f.to_=>"D..."	4 row(s) returned	0.000 sec / 0.000 sec

Query Completed

**7.A customer wants to travel from Bangalore to Kolkata with no more than two changes of flight. List the choice of departure times if the customer wants to arrive in Kolkata by 6 p.m.**

select departs

from flights f

where f.to\_="Kolkata" and arrives<="18:00:00" and f.from\_="Bangalore" union (select departs from flights

where from\_="Bangalore" and to\_ in(select from\_ from flights where flno in(select flno from flights f where f.to\_="Delhi" and f.from\_ in(select distinct(f.to\_)

```
from flights f where f.from_="Bangalore") and arrives<="18:00:00"));
```

The screenshot shows the MySQL Workbench interface. The main window has several tabs at the top: Local instance MySQL80, Local instance MySQL80, Local instance MySQL80, Local instance MySQL80 (flights), and Local instance MySQL80 (student). Below the tabs is a toolbar with various icons. The central area contains a query editor with the following code:

```
154 select min(f.distance)
155 from flights f
156 where f.from_="Bangalore" AND f.to_="Delhi"
157 ;
158
159 #q7
160
161 select departs
162 from flights f
163 where f.to_="Kolkata" and arrives<"18:00:00" and f.from_="Bangalore" union (select departs from flights
164 where from_="Bangalore" and to_ in(select from_ from flights where fno in(select fno from flights f where f.to_="Delhi" and f.from_ in(select distinct(f.to_)
165 from flights f where f.from_="Bangalore") and arrives<"18:00:00")));
166
167
168
```

The results grid shows one row of data:

departs
2005-05-13 07:15:31

Below the results grid is a "Result 5" section with an "Output" tab. It displays the execution log:

#	Time	Action	Message	Duration / Fetch
6	17:28:08	select * from flights LIMIT 0, 2000	6 row(s) returned	0.016 sec / 0.000 sec
7	17:29:18	select * from flights LIMIT 0, 2000	6 row(s) returned	0.000 sec / 0.000 sec
8	17:29:26	select a.aid from aircraft a where a.cruising_range > (select min(distance) from flights f where f.from_="Bangalore" AND f.to_ = ...)	Error Code: 1054. Unknown column 'Bangalore' in 'Where clause'	0.000 sec
9	17:35:00	select a.aid from aircraft a where a.cruising_range > (select min(distance) from flights f where f.from_="Bangalore" AND f.to_ = ...)	4 row(s) returned	0.015 sec / 0.000 sec
10	18:06:51	select departs from flights f where f.to_="Delhi" and arrives<"18:00:00" and f.from_="Bangalore" union (select departs from flights f where f.to_ in(select distinct(f.to_)	1 row(s) returned	0.000 sec / 0.000 sec
11	02:29:47	select departs from flights f where f.to_="Kolkata" and arrives<"18:00:00" and f.from_="Bangalore" union (select departs from flights f where f.to_ in(select distinct(f.to_)	1 row(s) returned	0.016 sec / 0.000 sec

The status bar at the bottom shows "Query Completed".

**Name:Neelam Godihal**  
**USN:1BM19CS220**

## **Program 6 : Order Database**

Consider the following schema for Order Database:

**SALESMAN (Salesman\_id, Name, City, Commission)**

**CUSTOMER (Customer\_id, Cust\_Name, City, Grade, Salesman\_id)**

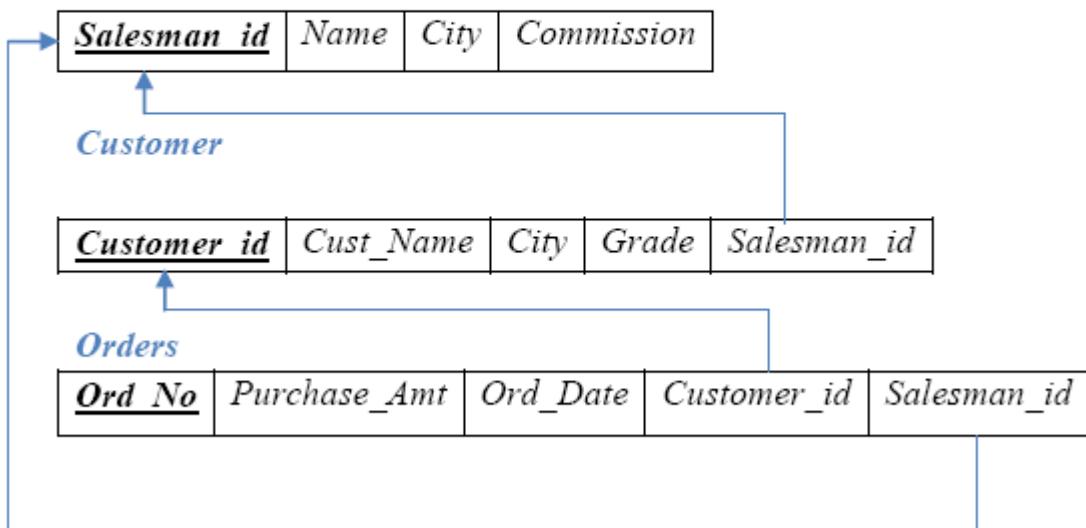
**ORDERS (Ord\_No, Purchase\_Amt, Ord\_Date, Customer\_id, Salesman\_id)**

Write SQL queries to

1. Count the customers with grades above Bangalore's average.
2. Find the name and numbers of all salesmen who had more than one customer.
3. List all salesmen and indicate those who have and don't have customers in their cities  
(Use UNION operation.)
4. Create a view that finds the salesman who has the customer with the highest order of a day.
5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

### **Schema Diagram**

*Salesman*



### **Creation of tables and entering tuple values:**

```
show databases;
```

```
create database order1;
```

```
use order1;
```

```
create table salesman(  
    salesman_id int primary key,  
    name varchar(30),  
    city varchar(30),  
    commission varchar(10)  
);
```

```
desc salesman;
```

```
create table customer(  
    customer_id int primary key,  
    cust_name varchar(30),  
    city varchar(30),  
    grade int(3),  
    salesman_id int,  
    foreign key(salesman_id) references salesman(salesman_id) on delete set NULL      #gonna  
delete later thats why  
);
```

```
desc customer;
```

```
create table orders(  
    ord_no int(5) primary key,  
    purchase_amt real,  
    ord_date date,  
    customer_id int,  
    salesman_id int,  
    foreign key(customer_id) references customer(customer_id),  
    foreign key(salesman_id) references salesman(salesman_id)  
);
```

```
desc orders;
```

```
insert into salesman  
values (1000,'John','Bangalore','25%'),  
       (2000,'Ravi','Bangalore','20%'),  
       (3000,'Kumar','Mysore','15%'),  
       (4000,'Smith','Delhi','30%'),  
       (5000,'Harsha','Hyderabad','15%');
```

```
select * from salesman;
```

```
insert into customer  
values (10,'Preethi','Bangalore',100,1000),
```

```

(11,'Vivek','Mangalore',300,1000),
(12,'Bhaskar','Chennai',400,2000),
(13,'Chethan','Bangalore',200,2000),
(14,'Mamatha','Bangalore',400,3000);

```

```
select * from customer;
```

```
insert into orders
```

```

values (50,5000,'2017-05-04',10,1000),
       (51,450,'2017-01-20',10,2000),
       (52,1000,'2017-02-24',13,2000),
       (53,3500,'2017-04-13',14,3000),
       (54,550,'2017-03-09',12,2000);

```

```
select * from orders;
```

## 1. Count the customers with grades above Bangalore's average.

```

select count(customer_id)
from customer c2
where grade>(
    select avg(c1.grade)
    from customer c1
    where city='Bangalore'
);

```

The screenshot shows the MySQL Workbench interface. The query editor window contains the following SQL code:

```

65
66 • q1
67 •     select count(customer_id)
68 •         from customer c2
69 •     where grade>(
70 •         select avg(c1.grade)
71 •             from customer c1
72 •             where city='Bangalore'
73 •     );

```

The results grid shows the output of the query:

	count(customer_id)
	3

The status bar at the bottom indicates "Query Completed".

## 2. Find the name and numbers of all salesmen who had more than one customer.

```

select name,salesman_id
from salesman
where salesman_id IN (
    select salesman_id

```

```

from customer
group by salesman_id
having count(customer_id)>1
);

```

The screenshot shows the MySQL Workbench interface with multiple tabs for different databases. The main window displays a query in the SQL editor:

```

76
77 • select name,salesman_id
78   from salesman
79   where salesman_id IN (
80     select salesman_id
81       from customer
82       group by salesman_id
83       having count(customer_id)>1
84   );

```

The results grid shows the following data:

	name	salesman_id
John	1000	
Ravi	2000	
***	8008	

The status bar at the bottom right indicates "Query Completed".

### 3. List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation.)

```

select salesman.salesman_id, name, cust_name, commission
from salesman, customer
where salesman.city = customer.city
union
select salesman_id, name, 'no match', commission
from salesman
where not city = any
(select city
from customer)
order by 2 desc;

```

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigation Bar
Query 1
MANA
INSTAN
Result Grid | Filter Rows | Export | Wrap Cell Content |
Output
Action Output
# Time Action
13 01:14:26 SELECT SALESMAN_ID, NAME, CUST_NAME, COMMISSION FROM SALESMAN, CUSTOMER WHERE SALESMAN.CI...
14 01:16:59 select salesman_id, name, cust_name, commission from salesman, customer where salesman.city = ...
15 01:17:13 Use order1
16 01:17:19 select name,a.salesman_id,cust_name,a.commission from salesman a,customer c where s.city=c.c...
17 01:19:34 select salesman_id, name, cust_name, commission from salesman, customer where salesman.city = ...
18 01:19:54 select salesman_id, name, cust_name, commission from salesman, customer where salesman.city = ...

```

Result Grid:

salesman_id	name	cust_name	commission
4000	Smith	no match	20%
2000	Ravi	Preethi	20%
2000	Ravi	Chethan	20%
2000	Ravi	Hematha	20%
3000	Kumar	no match	15%
1000	John	Preethi	25%
1000	John	Chethan	25%
1000	John	Hematha	25%
5000	Harsha	no match	15%

Result 9:

Action	Time	Action	Message	Duration / Fetch
13	01:14:26	SELECT SALESMAN_ID, NAME, CUST_NAME, COMMISSION FROM SALESMAN, CUSTOMER WHERE SALESMAN.CI...	Error Code: 1146. Table 'order1.customer1' doesn't exist	0.000 sec
14	01:16:59	select salesman_id, name, cust_name, commission from salesman, customer where salesman.city = ...	Error Code: 1146. Table 'order1.customer1' doesn't exist	0.000 sec
15	01:17:13	Use order1	0 row(s) affected	0.000 sec
16	01:17:19	select name,a.salesman_id,cust_name,a.commission from salesman a,customer c where s.city=c.c...	Error Code: 1052. Column 'salesman_id' in IN/ALL/ANY subquery is ambiguous	0.000 sec
17	01:19:34	select salesman_id, name, cust_name, commission from salesman, customer where salesman.city = ...	Error Code: 1052. Column 'salesman_id' in field list is ambiguous	0.000 sec
18	01:19:54	select salesman_id, name, cust_name, commission from salesman, customer where salesman.city = ...	9 row(s) returned	0.125 sec / 0.000 sec

Query Completed

#### 4. Create a view that finds the salesman who has the customer with the highest order of a day.

```

create view salesman_with_highest_sale as
select o2.ord_date,s.salesman_id,s.name
from salesman s,orders o2
where s.salesman_id=o2.salesman_id and o2.purchase_amt=(
    select max(purchase_amt)
    from orders o1
    where o1.ord_date=o2.ord_date
);      #for every type of a query

```

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigation Bar
Query 1
MANA
INSTAN
Result Grid | Filter Rows | Export | Wrap Cell Content |
Output
Action Output
# Time Action
118 01:19:34 select * from salesman_with_highest_sale;
119 01:19:54 select * from salesman_with_highest_sale;
120 01:21:50 select distinct salesman_id, name, cust_name, commission from salesman, customer where salesman.city = ...
121 01:22:24 select salesman_id, name, cust_name, commission from salesman, customer where salesman.city = ...
20 01:23:23 create view salesman_with_highest_sale as select o2.ord_date,s.salesman_id,s.name from salesman.s...
21 01:23:23 Table 'salesman_with_highest_sale' already exists
22 01:23:28 select * from salesman_with_highest_sale LIMIT 0,2000

```

Result Grid:

ord_date	salesman_id	name
2017-01-04	3000	John
2017-01-20	2000	Ravi
2017-02-24	2000	Ravi
2017-04-13	3000	Kumar
2017-03-09	2000	Ravi

Result 12:

Action	Time	Action	Message	Duration / Fetch
17	01:19:34	select salesman_id, name, cust_name, commission from salesman, customer where salesman.city = ...	Error Code: 1052. Column 'salesman_id' in field list is ambiguous	0.000 sec
18	01:19:54	select * from salesman_with_highest_sale;	9 row(s) returned	0.125 sec / 0.000 sec
19	01:21:50	select distinct salesman_id, name, cust_name, commission from salesman, customer where salesman.city = ...	9 row(s) returned	0.000 sec / 0.000 sec
20	01:22:24	select salesman_id, name, cust_name, commission from salesman, customer where salesman.city = ...	9 row(s) returned	0.000 sec / 0.000 sec
21	01:23:23	create view salesman_with_highest_sale as select o2.ord_date,s.salesman_id,s.name from salesman.s...	Error Code: 1050. Table 'salesman_with_highest_sale' already exists	0.109 sec
22	01:23:28	select * from salesman_with_highest_sale LIMIT 0,2000	5 row(s) returned	0.141 sec / 0.000 sec

Query Completed

5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

```
delete from Salesman  
where salesman_id = 1000;
```

```
select * from Salesman;  
select * from Orders
```

	order_id	purchase_amt	order_date	customer_id	salesman_id
▶	50	5000	2017-05-04	10	NULL
	51	450	2017-01-20	10	2000
	52	1000	2017-02-24	13	2000
	53	3500	2017-04-13	14	3000
*	54	550	2017-03-09	12	2000
	NULL	NULL	NULL	NULL	NULL

**Name:Neelam Godihal**  
**USN:1BM19CS220**

## **Program 7 : Book Database**

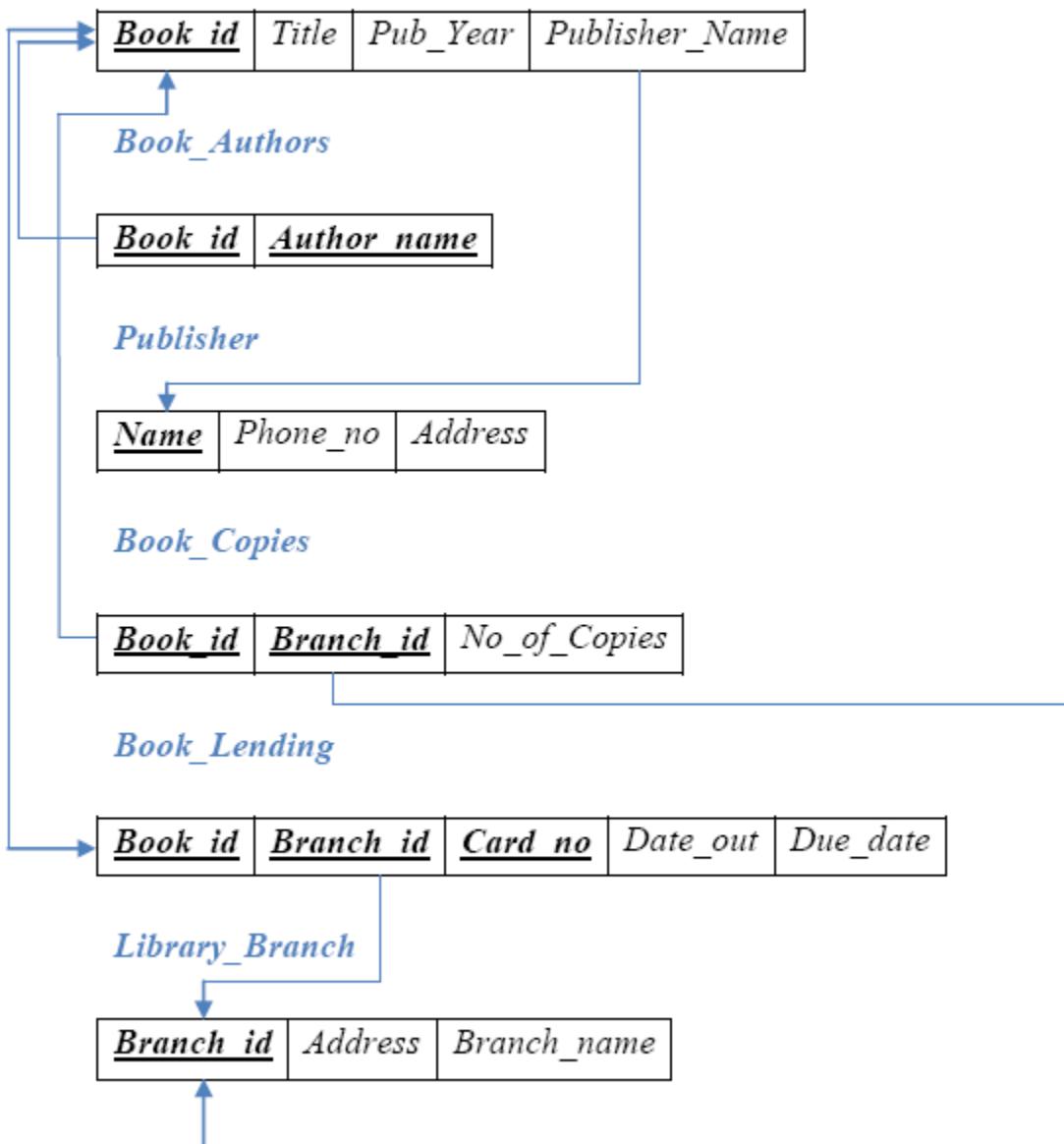
BOOK (Book\_id, Title, Publisher\_Name, Pub\_Year)  
BOOK\_AUTHORS (Book\_id, Author\_Name)  
PUBLISHER (Name, Address, Phone)  
BOOK\_COPIES (Book\_id, Branch\_id, No-of\_Copies)  
BOOK\_LENDING (Book\_id, Branch\_id, Card\_No, Date\_Out, Due\_Date)  
LIBRARY\_BRANCH (Branch\_id, Branch\_Name, Address)

### **Write SQL queries to**

1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, etc.
2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017
3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.
4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.
5. Create a view of all books and its number of copies that are currently available in the Library.

## Schema Diagram

*Book*



### **Creation of tables and entering tuple values:**

```
create database librarydb;
use librarydb;

create table Book(
    book_id int not null,
    title varchar(10) not null,
    pub_year varchar(20) not null,
    publisher_name varchar(20) not null,
    primary key(book_id)
);

create table BookAuthors(
    book_id int not null,
    author_name varchar(20) not null,
    primary key(book_id,author_name),
    foreign key(book_id) references Book(book_id) on delete cascade
);

create table Publisher(
    p_name varchar(20) not null,
    phone_no varchar(10) not null,
    address varchar(20) not null,
    primary key(p_name)
);

create table LibraryBranch(
    branch_id int not null,
    address varchar(20) not null,
    branch_name varchar(20) not null,
    primary key(branch_id)
);

create table BookCopies(
    book_id int not null,
    branch_id int not null,
    no_of_copies int not null,
    primary key(book_id,branch_id),
    foreign key(book_id) references Book(book_id) on delete cascade,
    foreign key(branch_id) references LibraryBranch(branch_id) on delete cascade
);

create table Card(
    card_no int not null,
    primary key(card_no)
);

create table BookLending(
    book_id int not null,
```

```
branch_id int not null,  
card_no int not null,  
date_out date not null,  
due_date date not null,  
primary key(book_id,branch_id,card_no),  
foreign key(book_id) references Book(book_id) on delete cascade,  
foreign key(branch_id) references LibraryBranch(branch_id) on delete cascade,  
foreign key(card_no) references Card(card_no) on delete cascade  
);
```

```
insert into Publisher(p_name,phone_no,address)  
values ('McGraw Hill','9989076587','Bangalore'),  
       ('Pearson','9889076565','New Delhi'),  
       ('Random House','7455679345','Hyderabad'),  
       ('Hachette Livre','8970862340','Chennai'),  
       ('Grupo Planeta','7756120238','Bangalore');
```

```
insert into Book(book_id,title,pub_year,publisher_name)  
values (1,'DBMS','2017-01','McGraw Hill'),  
       (2,'ADBMS','2016-06','McGraw Hill'),  
       (3,'CN','2016-09','Pearson'),  
       (4,'CG','2015-09','Grupo Planeta'),  
       (5,'OS','2016-05','Pearson');
```

```
insert into BookAuthors(author_name,book_id)  
values ('Navathe',1),  
       ('Navathe',2),  
       ('Tanenbaum',3),  
       ('Edward Angel',4),  
       ('Galvin',5);
```

```
insert into LibraryBranch(branch_id,branch_name,address)  
values (10,'RR Nagar','Bangalore'),  
       (11,'RNSIT','Bangalore'),  
       (12,'Rajajinagar','Bangalore'),  
       (13,'Nitte','Mangalore'),  
       (14,'Manipal','Udupi');
```

```
insert into BookCopies(book_id,branch_id,no_of_copies)  
values (1,10,10),  
       (1,11,5),  
       (2,12,2),  
       (2,13,5),  
       (3,14,7),  
       (5,10,1),  
       (4,11,3);
```

```
insert into Card(card_no)  
values (100),  
       (101),
```

(102),  
(103),  
(104);

```
insert into BookLending(date_out,due_date,book_id,branch_id,card_no)
values ('2017-01-01','2017-06-01',1,10,101),
       ('2017-01-11','2017-03-11',3,14,101),
       ('2017-02-21','2017-04-21',2,13,101),
       ('2017-03-15','2017-07-15',4,11,101),
       ('2017-04-12','2017-05-12',1,11,104);
```

### **QUERIES:**

- 1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, etc.**

Select

```
b.book_id,b.title,ba.author_name,b.publisher_name,b.pub_year,bc.no_of_copies,lb.branch_name
from Book b,BookCopies bc,LibraryBranch lb,BookAuthors ba
where b.book_id = bc.book_id and lb.branch_id = bc.branch_id and b.book_id =
ba.book_id;
```

book_id	title	author_name	publisher_name	pub_year	no_of_copies	branch_name
1	DBMS	Navathe	McGraw Hill	2017-01	10	RR Nagar
1	DBMS	Navathe	McGraw Hill	2017-01	5	RNSIT
2	ADBMS	Navathe	McGraw Hill	2016-06	2	Rajajinagar
2	ADBMS	Navathe	McGraw Hill	2016-06	5	Nitte
4	CG	Edward Angel	Grupo Planeta	2015-09	3	RNSIT
5	OS	Galvin	Pearson	2016-05	1	RR Nagar

- 2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017.**

```
select card_no
from BookLending where date_out between '2017-01-01' and '2017-06-30'
group by card_no
having count(book_id) > 3;
```

- 3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.**

```
delete from Book where title = 'CN';
select * from Book;
```

	book_id	title	pub_year	publisher_name
▶	1	DBMS	2017-01	McGraw Hill
	2	ADBMS	2016-06	McGraw Hill
	4	CG	2015-09	Grupo Planeta
	5	OS	2016-05	Pearson
◀	HULL	HULL	HULL	HULL

**4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.**

```
create view book_dates as
select pub_year from Book;
select * from book_dates;
```

	pub_year
▶	2017-01
	2016-06
	2015-09
	2016-05

**5.Create a view of all books and its number of copies that are currently available in the Library.**

```
create view book_view as
select b.book_id,b.title,lb.branch_name,bc.no_of_copies
from Book b,BookCopies bc,Librarybranch lb
where b.book_id = bc.book_id and lb.branch_id = bc.branch_id;
select * from book_view;
```

	book_id	title	branch_name	no_of_copies
▶	1	DBMS	RR Nagar	10
	1	DBMS	RNSIT	5
	2	ADBMS	Rajajinagar	2
	2	ADBMS	Nitte	5
	4	CG	RNSIT	3
	5	OS	RR Nagar	1

---

## **Program 8:**

Consider the following database of student enrollment in courses & books adopted for each course.

STUDENT (regno: string, name: string, major: string, bdate:date)

COURSE (course #:int, cname:string, dept:string)

ENROLL ( regno:string, course#:int, sem:int, marks:int)

BOOK \_ ADOPTION (course# :int, sem:int, book-ISBN:int)

TEXT (book-ISBN:int, book-title:string, publisher:string, author:string)

Database applications laboratory GCEM DEPARTMENT OF CSE Page - 5 - 5<sup>th</sup> semester i. Create the above tables by properly specifying the primary keys and the foreign keys.

ii. Enter at least five tuples for each relation.

iii. Demonstrate how you add a new text book to the database and make this book be adopted by some department.

iv. Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the ‘CS’ department that use more than two books.

v. List any department that has all its adopted books published by a specific publisher.

vi. Generate suitable reports.

### **Creating tables and entering tuples:**

```
create database student2;
use student2;
```

```
create table Student(
    regno varchar(10) not null,
    sname varchar(20) not
    null, major varchar(20) not
    null, bdate date not null,
    primary key(regno)
);
```

```
create table Course(
    courseno int not null,
    cname varchar(20) not
    null, dept varchar(20) not
    null, primary
    key(courseno)
);
```

```
create table Enroll(
    regno varchar(10) not null,
    courseno int not null,
    sem int not null,
    marks int not null,
    primary key(regno,courseno),
    foreign key(regno) references Student(regno) on delete cascade,
    foreign key(courseno) references Course(courseno) on delete
    cascade
);
```

```

create table Text(
    book_ISBN int not null,
    book_title varchar(20) not null,
    publisher varchar(20) not null,
    author varchar(20) not null,
    primary key(book_ISBN)
);

create table BookAdoption(
    courseno int not null,
    sem int not null,
    book_ISBN int not null,
    primary key(courseno,book_ISBN),
    foreign key(courseno) references Course(courseno) on delete
    cascade, foreignkey(book_ISBN)references Text(book_ISBN)on
    deletecascade
);

insert into Student(regno,sname,major,bdate)
values
('1pe11cs001','a','jr','1993-10-25'),
('1pe11cs002','b','sr','1993-09-24'),
('1pe11cs003','c','sr','1993-11-27'),
('1pe11cs004','d','sr','1993-04-13'),
('1pe11cs005','e','jr','1994-08-24');

insert into Course(courseno,cname,dept)
values
(111,'OS','CSE'),
(112,'EC','CSE'),
(113,'SS','ISE'),
(114,'DBMS','CSE'),
(115,'SIGNALS','ECE');

insert into Enroll(regno,courseno,sem,marks)
values
('1pe11cs001',115,3,100),
('1pe11cs002',114,5,100),
('1pe11cs003',113,5,100),
('1pe11cs004',111,5,100),
('1pe11cs005',112,3,100);

insert into Text(book_ISBN,book_title,publisher,author)
values
(10,'DATABASE
SYSTEMS','PEARSON','SCHIELD'),
(900,'OPERATING SYS','PEARSON','LELAND'),
(901,'CIRCUITS','HALL INDIA','BOB'),
(902,'SYSTEM SOFTWARE','PETERSON','JACOB'),

```

```
(903,'SCHEDULING','PEARSON','PATIL'),
(904,'DATABASE SYSTEMS','PEARSON','JACOB'),
(905,'DATABASE MANAGER','PEARSON','BOB'),
(906,'SIGNALS','HALL INDIA','SUMIT');
```

```
insert into BookAdoption(courseno,sem,book_ISBN)
values
(111,5,900),
(111,5,903),
(111,5,904),
(112,3,901),
(113,3,10),
(114,5,905),
(113,5,902),
(115,3,906);
```

```
update Text set publisher = 'PEARSON' where book_ISBN = 907;
```

### **Queries:**

**3) Demonstrate how you add a new text book to the database and make this book be adopted by some department.**

```
insert into Text values (907,'English literature','Random
House','ABCDE'); insert into BookAdoption values (112,5,907);
```

**4) Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the ‘CSE’ department that use more than two books.**

```
select ba.courseno,ba.book_ISBN,t.book_title
from BookAdoption ba,Text t,Course c
where ba.book_ISBN = t.book_ISBN and ba.courseno =
c.courseno and c.dept = 'CSE'
and 2 < (select count(ba1.courseno) from BookAdoption ba1,Course c1 where
ba1.courseno = c1.courseno and ba1.sem = ba.sem and c.dept = 'CSE')
order by t.book_title;
```

	courseno	book_ISBN	book_title
▶	112	901	CIRCUITS
	114	905	DATABASE MANAGER
	111	904	DATABASE SYSTEMS
	112	907	English literature
	111	900	OPERATING SYS
	111	903	SCHEDULING

**5) List any department that has all its adopted books published by a specific publisher.**

```
select c.dept from Course c,BookAdoption ba  
where c.courseno = ba.courseno  
group by c.dept  
having count(ba.book_ISBN) = (select count(book_ISBN) from Text where publisher  
= 'Pearson');
```

dept
CSE

#### 6) Generate suitable reports.

```
create view student_view as
```

```
select s.regno,s.sname,s.major,c cname,s.bdate from Student s,Enroll e,course c  
where s.regno = e.regno and c.courseno = e.courseno;
```

```
select * from student_view;
```

	regno	sname	major	cname	bdate
▶	1pe11cs001	a	jr	SIGNALS	1993-10-25
	1pe11cs002	b	sr	DBMS	1993-09-24
	1pe11cs003	c	sr	SS	1993-11-27
	1pe11cs004	d	sr	OS	1993-04-13
	1pe11cs005	e	jr	EC	1994-08-24



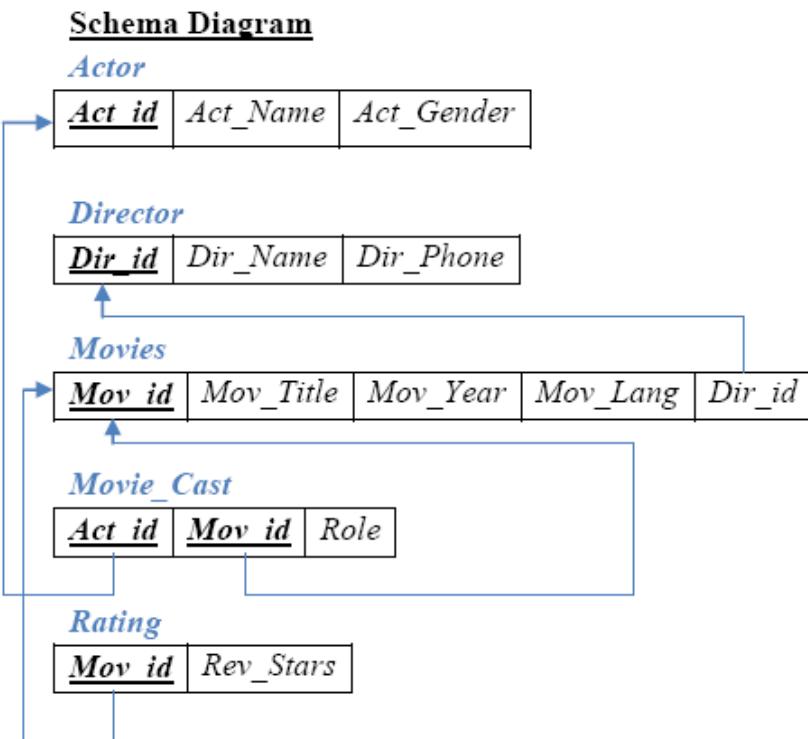
## Program 9: Movie database

Consider the schema for Movie Database:

**ACTOR** (*Act\_id*, *Act\_Name*, *Act\_Gender*)  
**DIRECTOR** (*Dir\_id*, *Dir\_Name*, *Dir\_Phone*)  
**MOVIES** (*Mov\_id*, *Mov\_Title*, *Mov\_Year*, *Mov\_Lang*, *Dir\_id*)  
**MOVIE\_CAST** (*Act\_id*, *Mov\_id*, *Role*)  
**RATING** (*Mov\_id*, *Rev\_Stars*)

Write SQL queries to

1. List the titles of all movies directed by ‘Hitchcock’.
2. Find the movie names where one or more actors acted in two or more movies.
3. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).
4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.
5. Update rating of all movies directed by ‘Steven Spielberg’ to 5.



Creating tables and entering tuples:

```
create database moviesDB;
use moviesDB;
```

```

create table Actor(
    act_id int not null,
    act_name varchar(20) not null,
    act_gender  varchar(1)  not
    null, primary key(act_id)
);

create table Director(
    dir_id int not null,
    dir_name varchar(20) not null,
    dir_phone varchar(10) not null,
    primary key(dir_id)
);

create table Movies(
    mov_id int not null,
    mov_title varchar(20) not null,
    mov_year year not null,
    mov_lang varchar(20) not null,
    dir_id int,
    primary key(mov_id),
    foreign key(dir_id) references Director(dir_id) on delete cascade
);

create table MovieCast(
    act_id int not null,
    mov_id int not null,
    role varchar(20) not null,
    primary
    key(act_id,mov_id),
    foreign key(act_id) references Actor(act_id) on delete cascade,
    foreign key(mov_id) references Movies(mov_id) on delete cascade
);

create table Rating(
    mov_id int not null,
    rev_stars int not null,
    primary key(mov_id),
    foreign key(mov_id) references Movies(mov_id) on delete cascade
);

insert into Actor(act_id,act_name,act_gender)
values(301,'Anushka','F'),
      (302,'Prabhas','M'),
      (303,'Punith','M'),
      (304,'Jermy','M');

```

```

insert into Director(dir_id,dir_name,dir_phone)
values(60,'Rajamouli','8751611001'),
      (61,'Hitchcock','7766138911'),
      (62,'Faran','9986776531),
      (63,'Steven Spielberg',8989776530);

insert into Movies(mov_id,mov_title,mov_year,mov_lang,dir_id)
values(1001,'baahubali 2','2017','Telugu',60),
      (1002,'baahubali 1','2015','Telugu',60),
      (1003,'akash','2008','Kannada',61),
      (1004,'war horse','2011','English',63);

insert into MovieCast(act_id,mov_id,role)
values(301,1002,'Heroine'),
      (301,1001,'Heroine'),
      (303,1003,'Hero'),
      (303,1002,'Guest'),
      (304,1004,'Hero');

insert into Rating(mov_id,rev_stars)
values(1001,4),
      (1002,2),
      (1003,5),
      (1004,4);

```

**1)List the titles of all movies directed by 'Hitchcock'.**

```

select mov_title from Movies,Director
where Movies.dir_id = Director.dir_id and dir_name like 'Hitchcock';

```

	mov_title
▶	akash

**2)Find the movie names where one or more actors acted in two or more movies.**

```

select distinct m.mov_title from Movies m,MovieCast
mc where m.mov_id = mc.mov_id
group by mc.act_id having count(mc.mov_id) >= 2;

```

	mov_title
▶	baahubali 1
	baahubali 2

**3)List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).**

```

select a.act_name,m.mov_title from Actor a,Movies m,MovieCast
mc where m.mov_id = mc.mov_id and a.act_id = mc.act_id
and m.mov_year not between '2000' and '2015';

```

	act_name	mov_title
▶	Anushka	baahubali 2

**4) Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.**

select m.mov\_title,max(r.rev\_stars) from Movies  
m,Rating r where m.mov\_id = r.mov\_id group by  
r.mov\_id;

	mov_title	max(r.rev_stars)
▶	baahubali 2	4
	baahubali 1	2
	akash	5
	war horse	5

**5) Update rating of all movies directed by ‘Steven Spielberg’ to 5. update Rating**

set rev\_stars = 5  
where mov\_id = (select mov\_id from Movies where dir\_id = (select dir\_id from Director where dir\_name = 'Steven Spielberg'));

select \* from Rating;

	mov_id	rev_stars
▶	1001	4
	1002	2
	1003	5
	1004	5
*	NULL	NULL

### Program 10 - College Database

Consider the schema for College Database:

STUDENT (*USN, SName, Address, Phone, Gender*)  
SEMSEC (*SSID, Sem, Sec*)

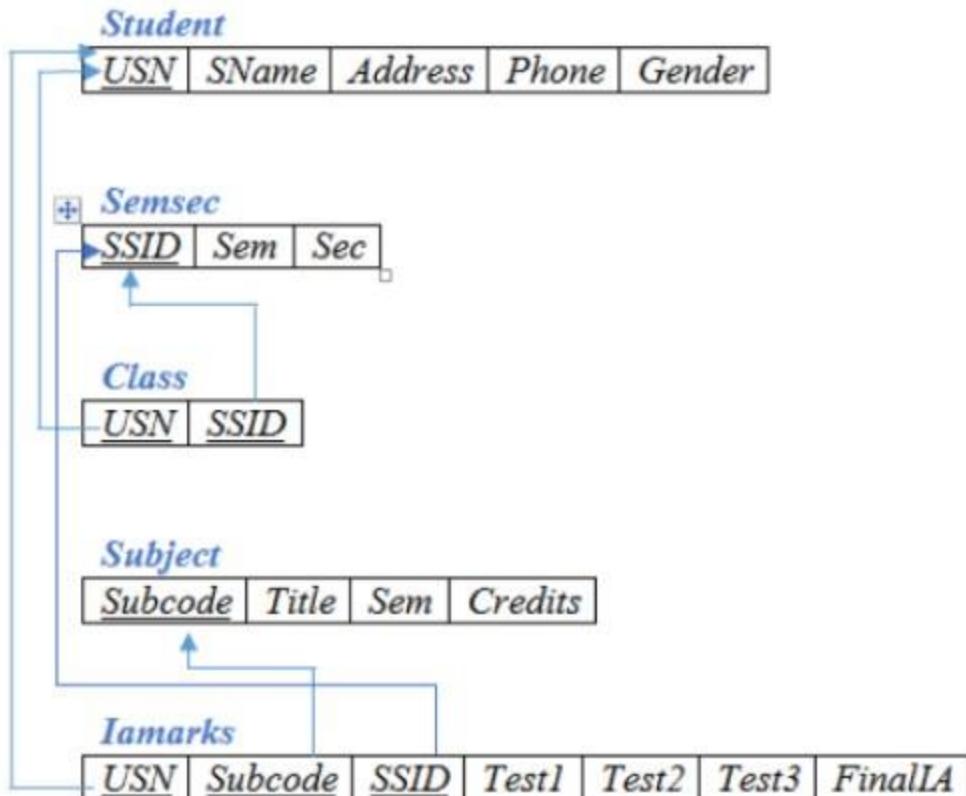
CLASS (*USN, SSID*)

SUBJECT (*Subcode, Title, Sem, Credits*)

IAMARKS (*USN, Subcode, SSID, Test1, Test2, Test3, FinalIA*) Write SQL queries to

1. List all the student details studying in fourth semester ‘C’ section.
2. Compute the total number of male and female students in each semester and in each section.
3. Create a view of Test1 marks of student USN ‘1BI15CS101’ in all subjects.
4. Categorize students based on the following criterion: If FinalIA = 17 to 20 then CAT = ‘Outstanding’  
If FinalIA = 12 to 16 then CAT = ‘Average’ If FinalIA < 12 then CAT = ‘Weak’

Give these details only for 8th semester A, B, and C section students.



#### Creating tables and entering tuple values:

```
create database collegeDB;
use collegeDB;
```

```
create table Student(
```

```

usn varchar(10) not null,
sname varchar(20) not null,
address varchar(20) not null,
phone varchar(10) not null,
gender varchar(1) not null,
primary key(usn)
);

create table SemSec(
    ssid varchar(5) not
null, sem int not null,
section varchar(1) not
null, primary key(ssid)
);

create table Class(
    usn varchar(10) not null,
ssid varchar(5) not null,
primary key(usn,ssid),
foreign key(usn) references Student(usn) on delete
cascade, foreign key(ssid) references SemSec(ssid) on
delete cascade
);

create table Subject(
    subcode varchar(6) not
null, title varchar(10) not
null, sem int not null,
credits int not null,
primary key(subcode)
);

create table IAMarks(
    usn varchar(10) not null,
subcode varchar(6) not
null, ssid varchar(5) not
null, test1 int not null,
test2 int not null,
test3 int not null,
final double not null default ((test1 + test2 +
test3)/3), primary key(usn,subcode,ssid),
foreign key(usn) references Student(usn) on delete cascade,
foreign key(subcode) references Subject(subcode) on delete cascade,
foreign key(ssid) references SemSec(ssid) on delete cascade
);

insert into Student
values ('1RN13CS020','Akshay','Belagavi','8877881122','M'),

```

('1RN13CS052','Sandhya','Bengaluru','7722829912','F'),  
('1RN13CS091','Teesha','Bengaluru','7712312312','F'),  
('1RN13CS066','Supriya','Mangaluru','8877881122','F'),  
('1RN14CS010','Abhay','Bengaluru','9900211201','M'),  
('1RN14CS032','Bhaskar','Bengaluru','9923211099','M'),  
('1RN15CS011','Ajay','Tumkur','9845091341','M'),  
('1RN15CS029','Chitra','Davangere','7696772121','F'),  
('1RN15CS045','Jeeva','Bellary','9944850121','M'),  
('1RN15CS091','Santhosh','Mangaluru','8812332201','M'),  
('1RN16CS045','Ismail','Kalburgi','9900232201','M'),  
('1RN16CS088','Sameera','Shimoga','9905542212','F'),  
('1RN16CS122','Vinayaka','Chikmagalur','8800880011','M'),  
('1RN14CS025','Asmi','Bengaluru','7894737377','F');

insert into SemSec values

('CSE8A',8,'A'),  
('CSE8B',8,'B'),  
('CSE8C',8,'C'),  
('CSE7A',7,'A'),  
('CSE7B',7,'B'),  
('CSE7C',7,'C'),  
('CSE6A',6,'A'),  
('CSE6B',6,'B'),  
('CSE6C',6,'C'),  
('CSE5A',5,'A'),  
('CSE5B',5,'B'),  
('CSE5C',5,'C'),  
('CSE4A',4,'A'),  
('CSE4B',4,'C'),  
('CSE4C',4,'C'),  
('CSE3A',3,'A'),  
('CSE3B',3,'B'),  
('CSE3C',3,'C'),  
('CSE2A',2,'A'),  
('CSE2B',2,'B'),  
('CSE2C',2,'C'),  
('CSE1A',1,'A'),  
('CSE1B',1,'B'),  
('CSE1C',1,'C');

insert into Class values

('1RN13CS020','CSE8A'),  
('1RN13CS052','CSE8A'),  
('1RN13CS066','CSE8B'),  
('1RN13CS091','CSE8C'),  
('1RN14CS010','CSE7A'),  
('1RN14CS032','CSE7A'),

```
('1RN15CS011','CSE4A'),  
('1RN15CS029','CSE4A'),  
('1RN15CS045','CSE4B'),  
('1RN15CS091','CSE4C'),  
('1RN16CS045','CSE3A'),  
('1RN16CS088','CSE3B'),  
('1RN16CS122','CSE3C');
```

```
insert into Subject values  
('10CS81','ACA',8,4),  
('10CS82','SSM',8,4),  
('10CS83','NM',8,4),  
('10CS84','CC',8,4),  
('10CS85','PW',8,4),  
('10CS71','OOAD',7,4),  
('10CS72','ECS',7,4),  
('10CS73','PTW',7,4),  
('10CS74','DWDM',7,4),  
('10CS75','JAVA',7,4),  
('10CS76','SAN',7,4),  
('15CS51','ME',5,4),  
('15CS52','CN',5,4),  
('15CS53','DBMS',5,4),  
('15CS54','ATS',5,4),  
('15CS55','JAVA',5,3),  
('15CS56','AI',5,3),  
('15CS41','M4',4,4),  
('15CS42','SE',4,4),  
('15CS43','DAA',4,4),  
('15CS44','MPMC',4,4),  
('15CS45','OOC',4,3),  
('15CS46','DC',4,3),  
('15CS31','M3',3,4),  
('15CS32','ADE',3,4),  
('15CS33','DSA',3,4),  
('15CS34','CO',3,4),  
('15CS35','USP',3,3),  
('15CS36','DMS',3,3);
```

```
insert into IAMarks(usn,subcode,ssid,test1,test2,test3) values  
('1RN13CS091','10CS81','CSE8C',15,16,18),  
('1RN13CS091','10CS82','CSE8C',12,19,14),  
('1RN13CS091','10CS83','CSE8C',19,15,20),  
('1RN13CS091','10CS84','CSE8C',20,16,19),  
('1RN13CS091','10CS85','CSE8C',15,15,12);
```

```
select * from IAMarks;
```

1) List all the student details studying in fourth semester ‘C’ section.

select s.usn,s.sname from Student s,Class c where s.usn = c.usn and ssid = 'CSE4C';

	usn	sname
▶	1RN15CS091	Santhosh

2) Compute the total number of male and female students in each semester and in each section.

select c ssid,s.gender,count(s.gender) from Student s,Class c where s.usn = c.usn

group by c ssid,s.gender;

	ssid	gender	count(s.gender)
▶	CSE3A	M	1
	CSE3B	F	1
	CSE3C	M	1
	CSE4A	M	1
	CSE4A	F	1
	CSE4B	M	1
	CSE4C	M	1
	CSE7A	M	2
	CSE8A	M	1
	CSE8A	F	1
	CSE8B	F	1
	CSE8C	F	1

3) Create a view of Test1 marks of student USN ‘1BI15CS101’ in all subjects.

create view student\_details\_view as

select \* from IAMarks where usn = '1RN13CS091';

select \* from student\_details\_view;

	usn	subcode	ssid	test1	test2	test3	final
▶	1RN13CS091	10CS81	CSE8C	15	16	18	16.3333333333
	1RN13CS091	10CS82	CSE8C	12	19	14	15
	1RN13CS091	10CS83	CSE8C	19	15	20	18
	1RN13CS091	10CS84	CSE8C	20	16	19	18.3333333333
	1RN13CS091	10CS85	CSE8C	15	15	12	14

4) Categorize students based on the following criterion:

FinalIA = 17 to 20 then CAT = ‘Outstanding’

FinalIA = 12 to 16 then CAT = ‘Average’

FinalIA < 12 then CAT = ‘Weak’

Give these details only for 8th semester A, B, and C section students.

```

select iam.usn,iam.final,'Outstanding'
from IAMarks iam,Student s,Class c
where iam.usn = s.usn and c.usn =
s.usn
and final between 17 and 20 and c.ssid like
'CSE8_ ' union
select iam.usn,iam.final,'Average'
from IAMarks iam,Student s,Class
c
where iam.usn = s.usn and c.usn = s.usn
and final between 12 and 17 and c.ssid like
'CSE8_ ' union
select iam.usn,iam.final,'Average'
from IAMarks iam,Student s,Class
c
where iam.usn = s.usn and c.usn =
s.usn and final < 12 and c.ssid like
'CSE8_ ';

```

	usn	final	Outstanding
▶	1RN13CS091	18.333333333	Outstanding
	1RN13CS091	18	Outstanding
	1RN13CS091	14	Average
	1RN13CS091	15	Average
	1RN13CS091	16.333333333	Average