

IN-LAB

Use the tables and data in Experiment – 5, In-Lab section and work on the following queries

Case Study 1 - Transport Department

1. Display the list of customers available in a branch.

The screenshot shows the MySQL Workbench interface. The 'SCHEMAS' pane on the left lists various databases, with 'practical5' selected. The 'Query' editor in the center contains the following SQL query:

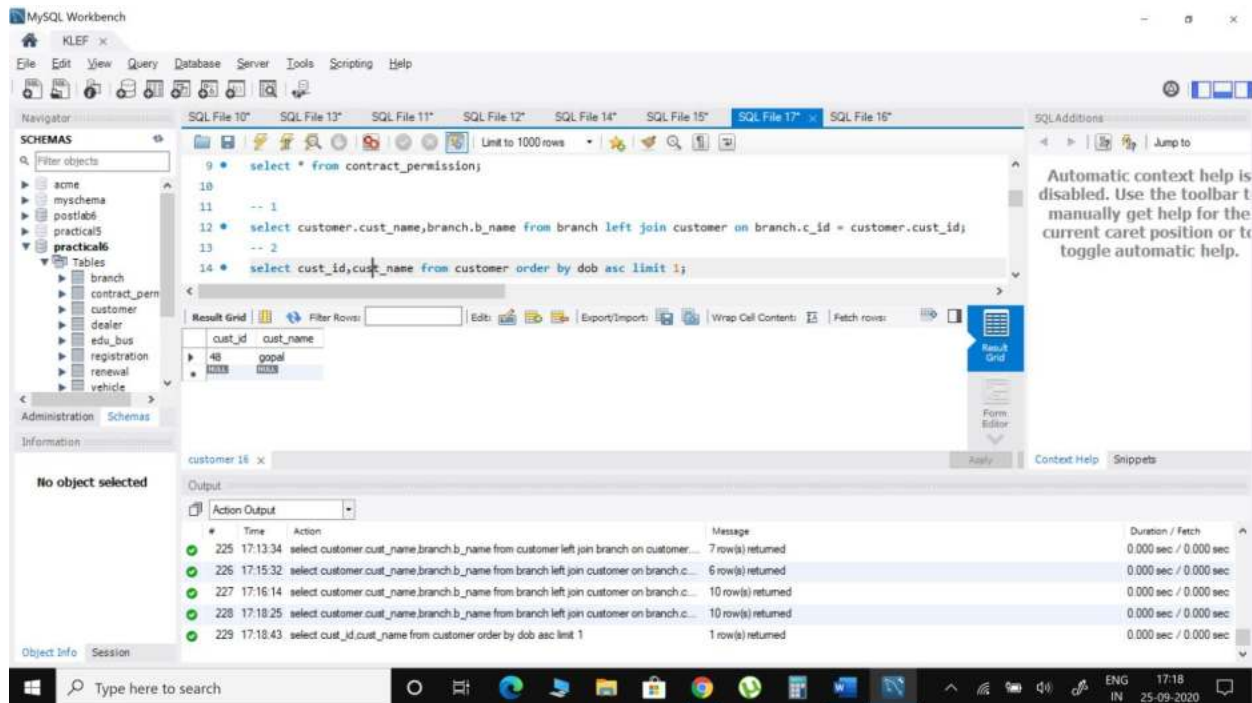
```
select * from contract_permission;  
  
-- 1  
select customer.cust_name, branch.b_name from branch left join customer on branch.c_id = customer.cust_id;  
  
-- 2
```

The 'Result Grid' displays the results of the second query, showing a list of customers and their associated branches:

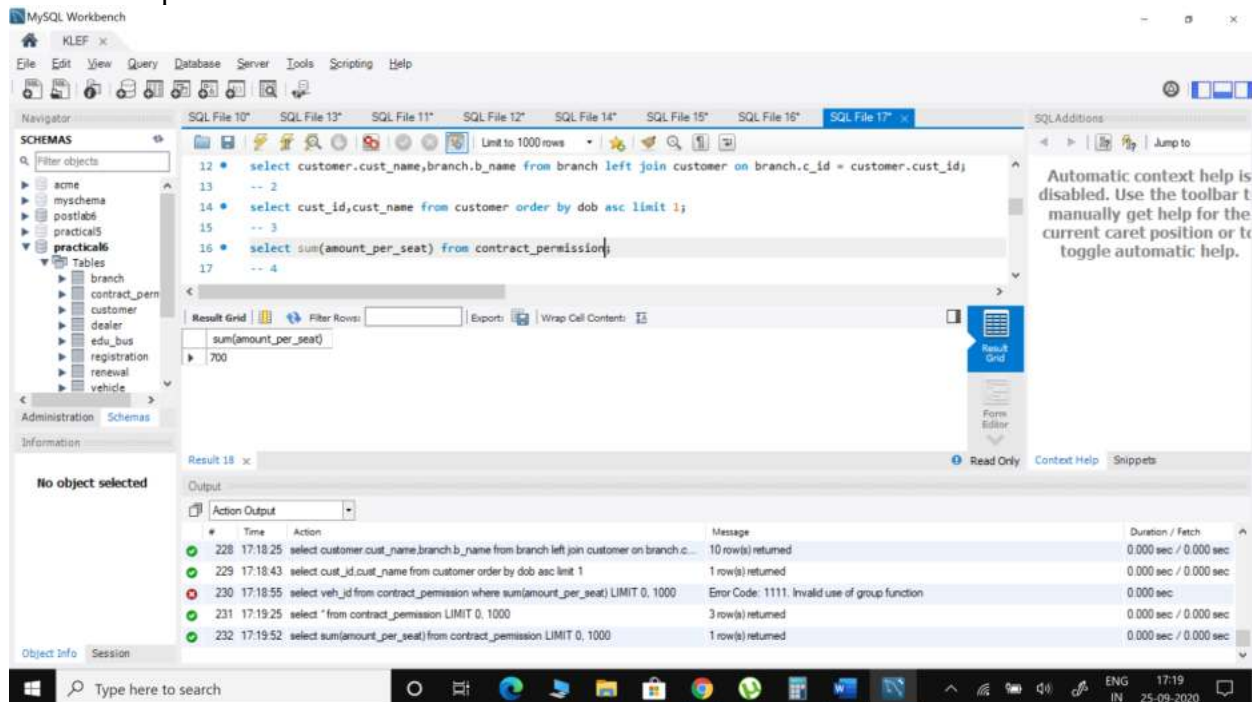
cust_name	b_name
raju	lukatpally
raju	madhapur
hari	hitech city
giri	miyapur
ramu	raju nagar
ramu	pride

The 'Output' pane at the bottom shows the execution log, indicating that the query was executed successfully and returned 7 rows.

2. Create a mysql query to know the older of all the customers



3. Write mysql query to calculate the total amount generated by giving contract permission for amount per seat



4. Create a query to display all the type of vehicles present

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

```

13 -- 2
14 select cust_id, cust_name from customer order by dob asc limit 1;
15 -- 3
16 select sum(amount_per_seat) from contract_permission;
17 -- 4
18 select distinct veh_type from vehicle;

```

The Results window displays the output of the last query:

veh_type
2-Wheeler
3-Wheeler
4-Wheeler

The Output window shows the execution log:

#	Time	Action	Message	Duration / Fetch
229	17:18:43	select cust_id, cust_name from customer order by dob asc limit 1	1 row(s) returned	0.000 sec / 0.000 sec
230	17:18:55	select veh_id from contract_permission where sum(amount_per_seat) LIMIT 0, 1000	Error Code: 1111. Invalid use of group function	0.000 sec
231	17:19:25	select * from contract_permission LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
232	17:19:52	select sum(amount_per_seat) from contract_permission LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
233	17:20:16	select distinct veh_type from vehicle LIMIT 0, 1000	3 row(s) returned	0.016 sec / 0.000 sec

5. Write mysql query to display all the cities present in a given state.

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

```

16 select sum(amount_per_seat) from contract_permission;
17
18 distinct veh_type from vehicle;
19
20 distinct customer.city, customer.state from customer inner join edu_bus inner join branch inner join Dealer;
21

```

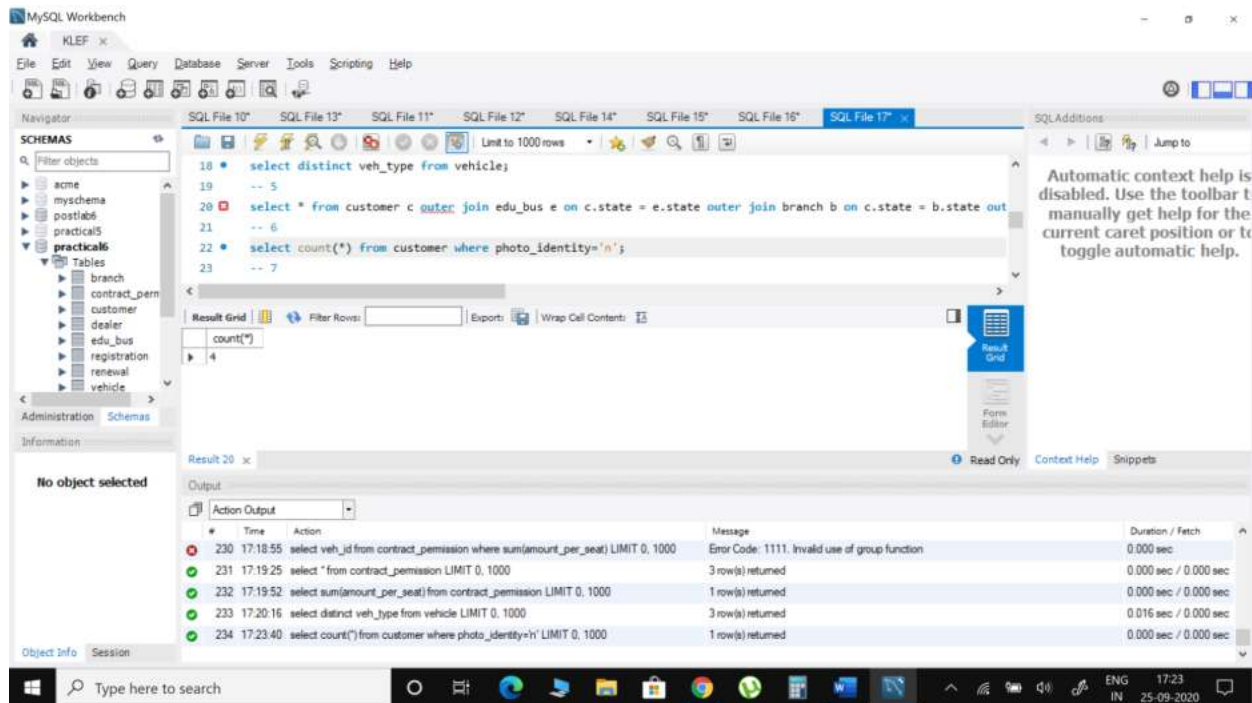
The Results window displays the output of the last query:

city	state
Guntur	Andhra Pradesh
Perambur	Tamil Nadu
Hyderabad	Telangana
Vijayawada	Andhra Pradesh

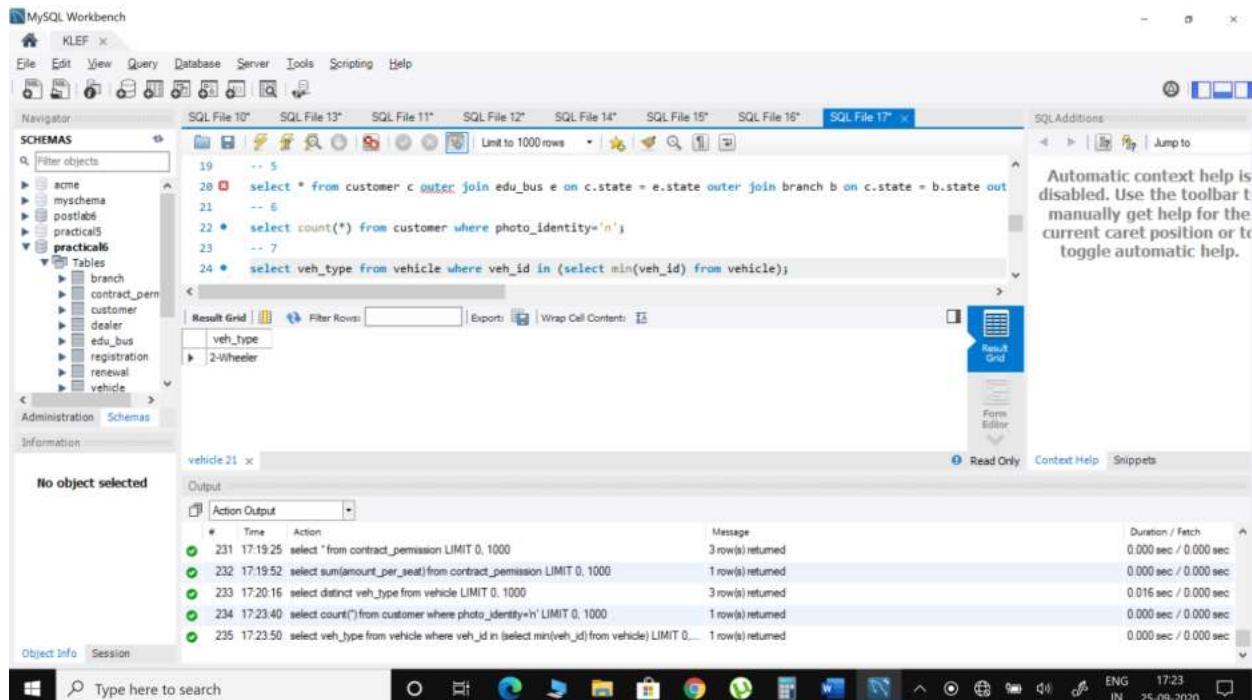
The Output window shows the execution log:

#	Time	Action	Message	Duration / Fetch
247	17:26:32	select * from customer c join edu_bus e join dealer d group by c.state LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
248	17:27:08	select c.state, count(distinct c.city) from customer c join edu_bus e join dealer d group by c.state LIMIT 0, 1000	3 row(s) returned	0.015 sec / 0.000 sec
249	17:27:18	Drop view Present_Customer	0 row(s) affected	0.015 sec
250	17:27:29	select * from Present_Customer LIMIT 0, 1000	Error Code: 1146. Table 'practical6.present_customer' doesn't exist	0.000 sec
251	17:29:07	select distinct customer.city, customer.state from customer inner join edu_bus inner join branch inner join Dealer	4 row(s) returned	0.000 sec / 0.000 sec

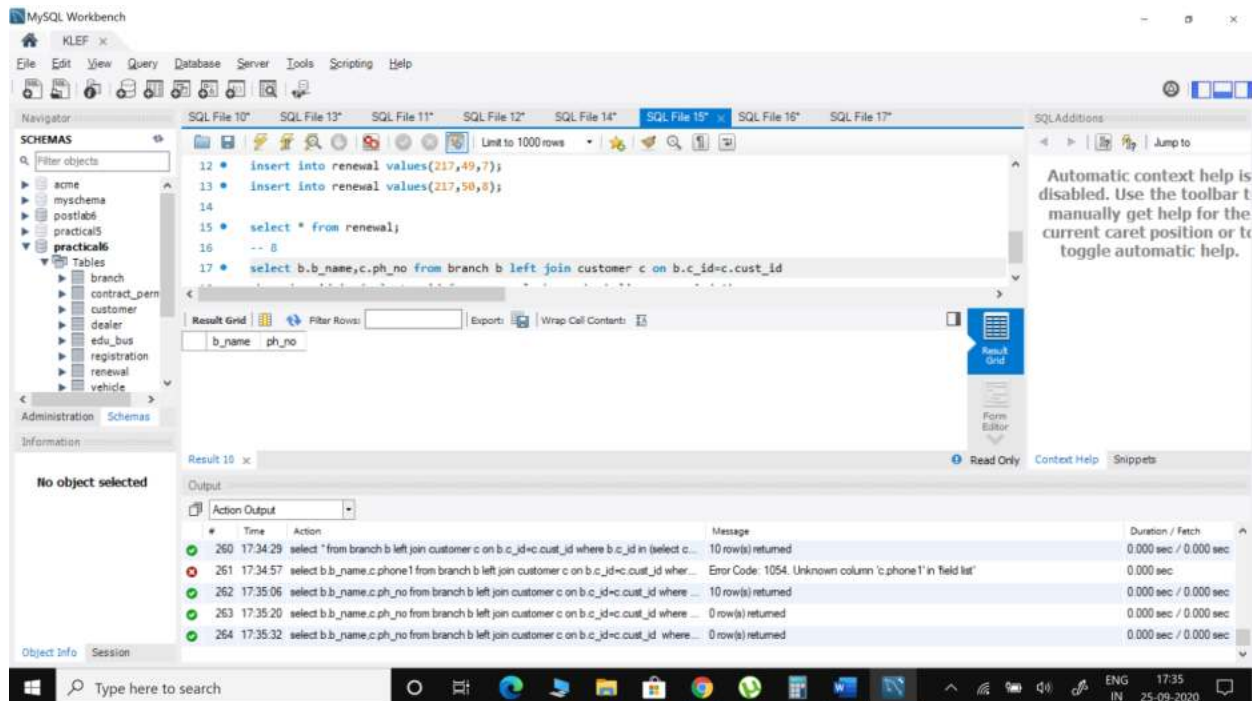
6. Display the number of vehicles of customers who are not having photo identity.



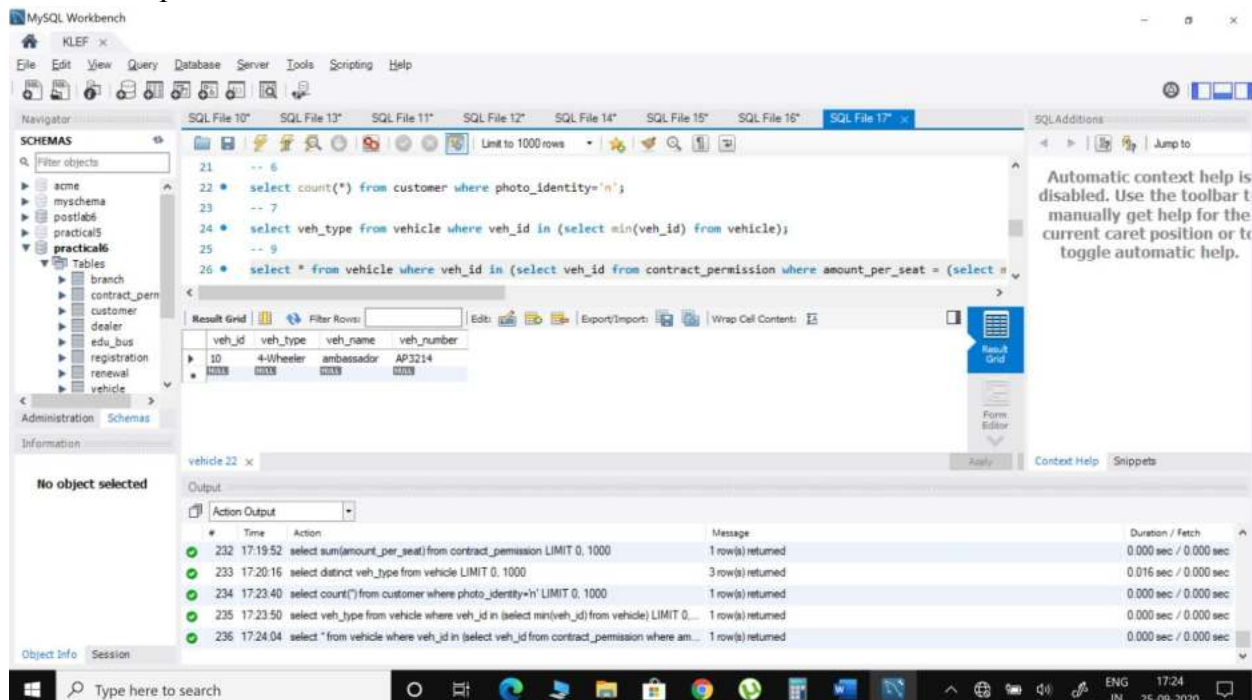
7. Write mysql statement to search for vehicle type which is having the vehicle id as the smallest number.



8. Create a mysql query to know the branch name and phone number of a customer who is having license period of 2 years .



9. Display the vehicle details for which maximum amount is paid per seat for contractpermission.



10. Write Co-related nested subquery to know the customer name , phone number, city whose branch name is 'Madhapur'

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

```

23 -- 7
24 select veh_type from vehicle where veh_id in (select min(veh_id) from vehicle);
25 -- 9
26 select * from vehicle where veh_id in (select veh_id from contract_permission where amount_per_seat = (select
27 -- 10
28 select cust_name,ph_no,city from customer where cust_id in (select c_id from branch where b_name = 'madhapur')));

```

The Result Grid shows the following data:

cust_name	ph_no	city
raju	9123456789	Guntur

The Output tab shows the execution log with the following messages:

#	Time	Action	Message	Duration / Fetch
233	17:20:16	select distinct veh_type from vehicle LIMIT 0, 1000	3 row(s) returned	0.016 sec / 0.000 sec
234	17:23:40	select count(*) from customer where photo_identity='n' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
235	17:23:50	select veh_type from vehicle where veh_id in (select min(veh_id) from vehicle) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
236	17:24:04	select * from vehicle where veh_id in (select veh_id from contract_permission where amount_per_seat = (select cust_name,ph_no,city from customer where cust_id in (select c_id from branch where b_name = 'madhapur')))	1 row(s) returned	0.000 sec / 0.000 sec
237	17:24:26	select cust_name,ph_no,city from customer where cust_id in (select c_id from branch where b_name = 'madhapur')	1 row(s) returned	0.000 sec / 0.000 sec

11. Create a view “Present_Customer” with customer name , phone number, state and city of customer and display the view.

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

```

26 select * from vehicle where veh_id in (select veh_id from contract_permission where amount_per_seat = (select
27 -- 10
28 select cust_name,ph_no,city from customer where cust_id in (select c_id from branch where b_name = 'madhapur')));
29 -- 11
30 create view Present_Customer as select cust_name,ph_no,state,city from customer;
31 select * from Present_Customer;

```

The Result Grid shows the following data:

cust_name	ph_no	state	city
raju	9123456789	Andhra Pradesh	Guntur
hari	1122334455	Tamil Nadu	Perambur
gri	8877665544	Telangana	Hyderabad
ramu	7654564321	Andhra Pradesh	Vijayanada
rahu	9999999998	Andhra Pradesh	Guntur
gopi	787777775	Telangana	Hyderabad

The Output tab shows the execution log with the following messages:

#	Time	Action	Message	Duration / Fetch
235	17:23:50	select veh_type from vehicle where veh_id in (select min(veh_id) from vehicle) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
236	17:24:04	select * from vehicle where veh_id in (select veh_id from contract_permission where amount_per_seat = (select cust_name,ph_no,city from customer where cust_id in (select c_id from branch where b_name = 'madhapur')))	1 row(s) returned	0.000 sec / 0.000 sec
237	17:24:26	select cust_name,ph_no,city from customer where cust_id in (select c_id from branch where b_name = 'madhapur')	1 row(s) returned	0.000 sec / 0.000 sec
238	17:24:48	create view Present_Customer as select cust_name,ph_no,state,city from customer;	0 row(s) affected	0.063 sec
239	17:24:49	select * from Present_Customer LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

12. Write mysql query to show indexes on customer table.

The screenshot shows the MySQL Workbench interface with the following SQL queries in the editor:

```

28 * select cust_name,ph_no,city from customer where cust_id in (select c_id from branch where b_name = 'madhapur');
29 -- 11
30 * create view Present_Customer as select cust_name,ph_no,state,city from customer;
31 * select *from Present_Customer;
32 -- 12
33 * show indexes from customer;

```

The **Result Grid** displays the following table:

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment
customer	0	PRIMARY	1	cust_id	A	10				BTREE	
customer	1	v_id	1	v_id	A	10			YES	BTREE	

The **Action Output** pane shows the following results:

#	Time	Action	Message	Duration / Fetch
236	17:24:04	select *from vehicle where veh_id in (select veh_id from contract_permission where am...	1 row(s) returned	0.000 sec / 0.000 sec
237	17:24:26	select cust_name,ph_no,city from customer where cust_id in (select c_id from branch w...	1 row(s) returned	0.000 sec / 0.000 sec
238	17:24:48	create view Present_Customer as select cust_name,ph_no,state,city from customer	0 row(s) affected	0.063 sec
239	17:24:49	select *from Present_Customer LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
240	17:24:59	show indexes from customer	2 row(s) returned	0.000 sec / 0.000 sec

13. Create a query to display the count of dealers from “Andhra Pradesh”

The screenshot shows the MySQL Workbench interface with the following SQL queries in the editor:

```

30 * create view Present_Customer as select cust_name,ph_no,state,city from customer;
31 * select *from Present_Customer;
32 -- 12
33 * show indexes from customer;
34 -- 13
35 * select count(deal_id) from dealer where state = "Andhra Pradesh";

```

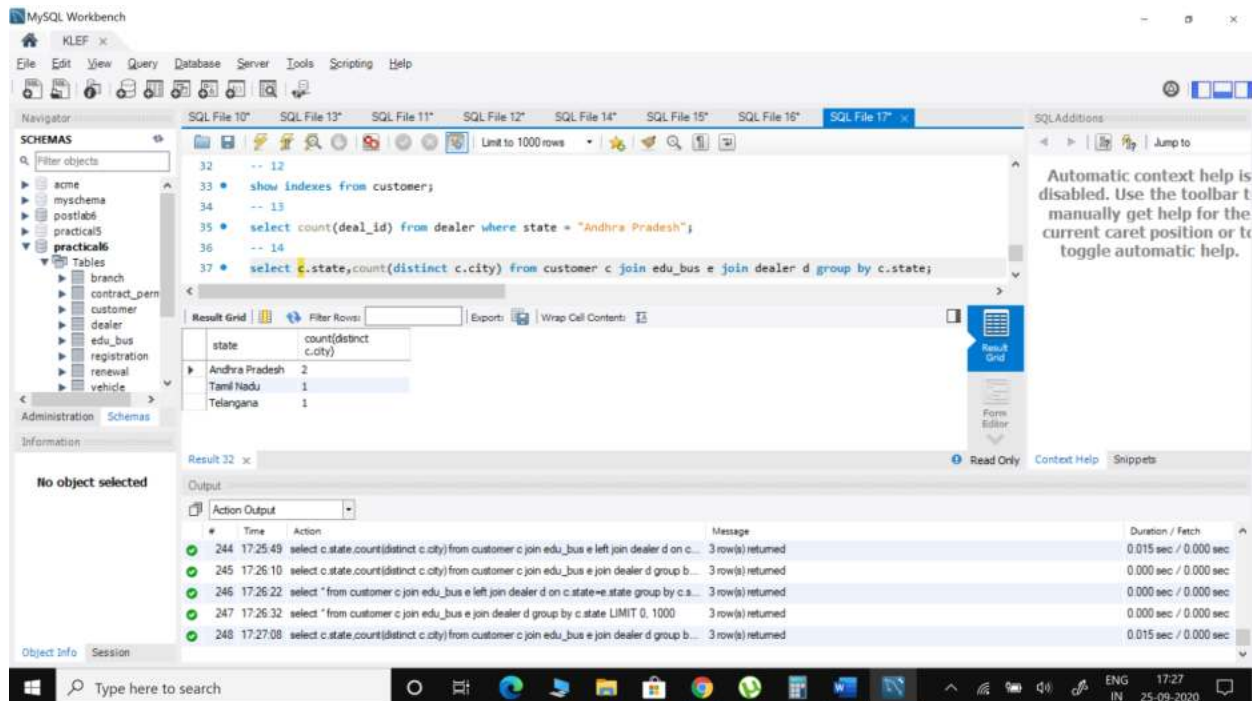
The **Result Grid** displays the following table:

count(deal_id)
4

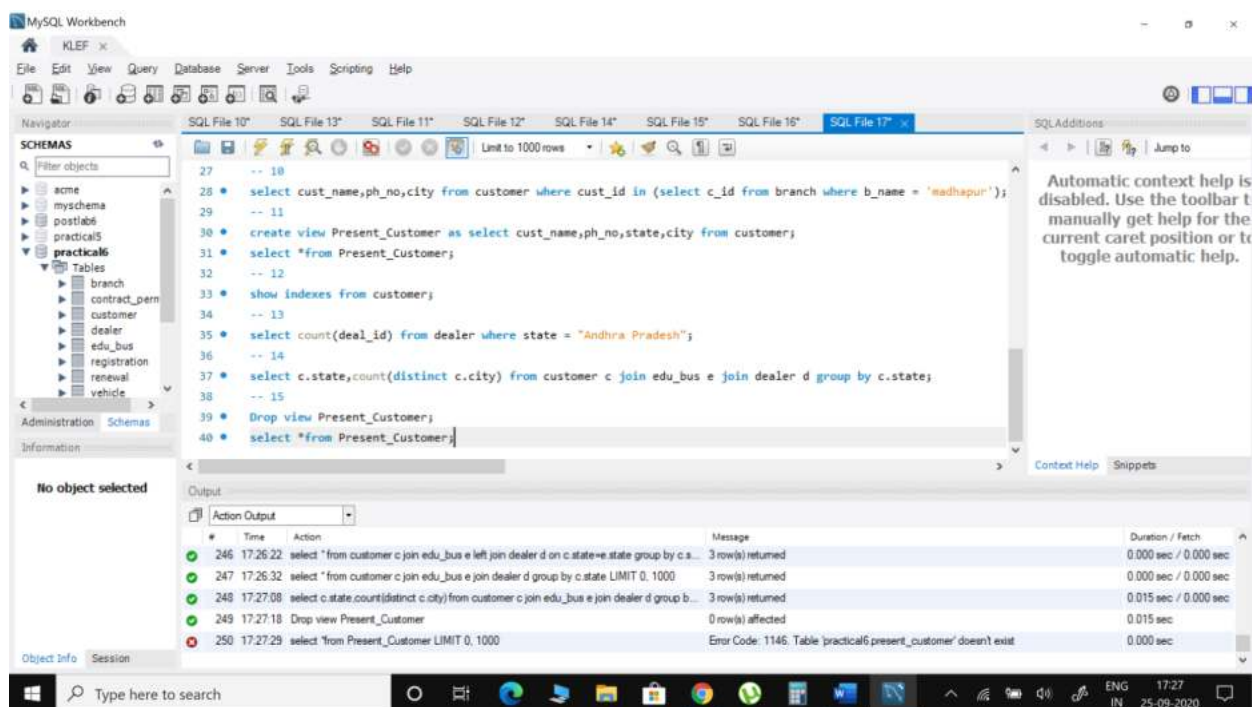
The **Action Output** pane shows the following results:

#	Time	Action	Message	Duration / Fetch
237	17:24:26	select cust_name,ph_no,city from customer where cust_id in (select c_id from branch w...	1 row(s) returned	0.000 sec / 0.000 sec
238	17:24:48	create view Present_Customer as select cust_name,ph_no,state,city from customer	0 row(s) affected	0.063 sec
239	17:24:49	select *from Present_Customer LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
240	17:24:59	show indexes from customer	2 row(s) returned	0.000 sec / 0.000 sec
241	17:25:07	select count(deal_id) from dealer where state = "Andhra Pradesh" LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

14. Display the number of cities in each state



15. Drop the view "Present_Customer"



Select *from Present_Customer; // gives error because we dropped the view

Use the tables and data in Experiment – 5, In-Lab section and work on the following queries

Case Study 4 – KL University ERP

- 1) Write a query to display the names, annual salary of all the faculty based on decreasing order of their annual salary.

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following queries:

```
6 * select * from OTHER_FEE;
7
8 -- 1
9 * select FID,FNAME,SALARY from Faculty order by SALARY desc;
10
11 -- 2
11 * select FID,FNAME,SALARY from Faculty order by SALARY desc limit 1 offset 2;
```

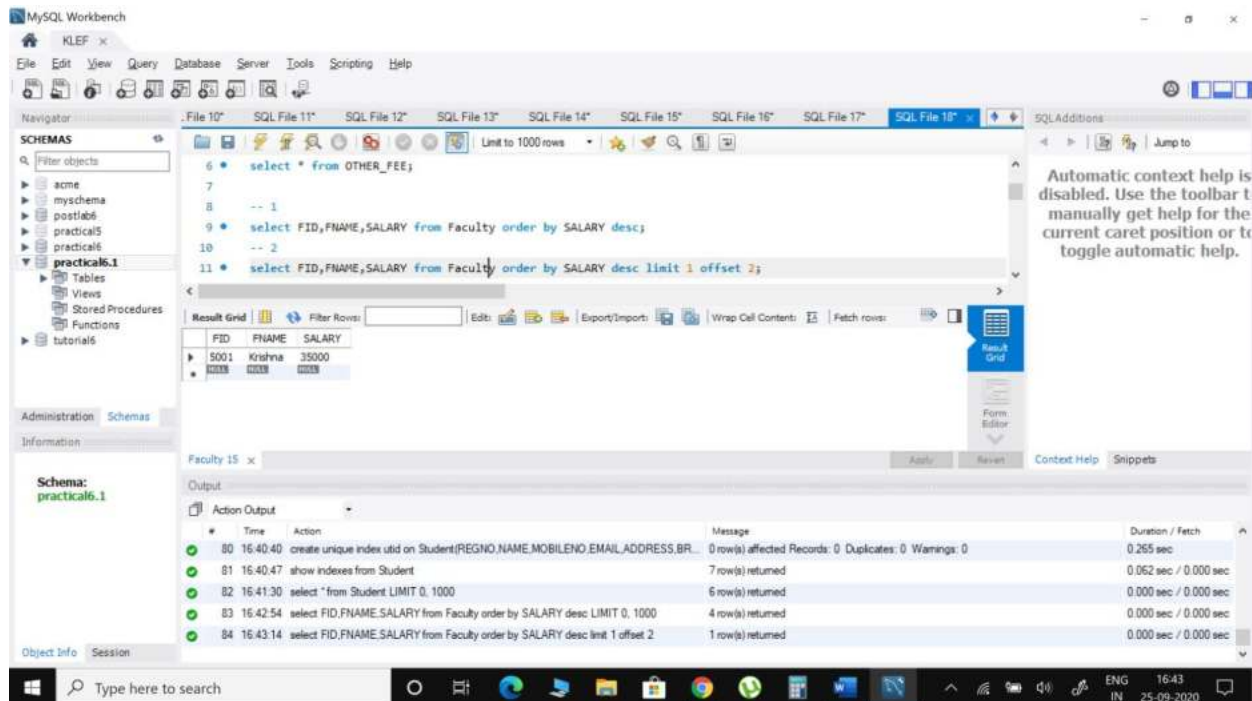
The Result Grid displays the following data:

FID	FNAME	SALARY
5002	Hari	75000
5003	Mohan	40000
5001	Krishna	35000
5004	Giri	30000

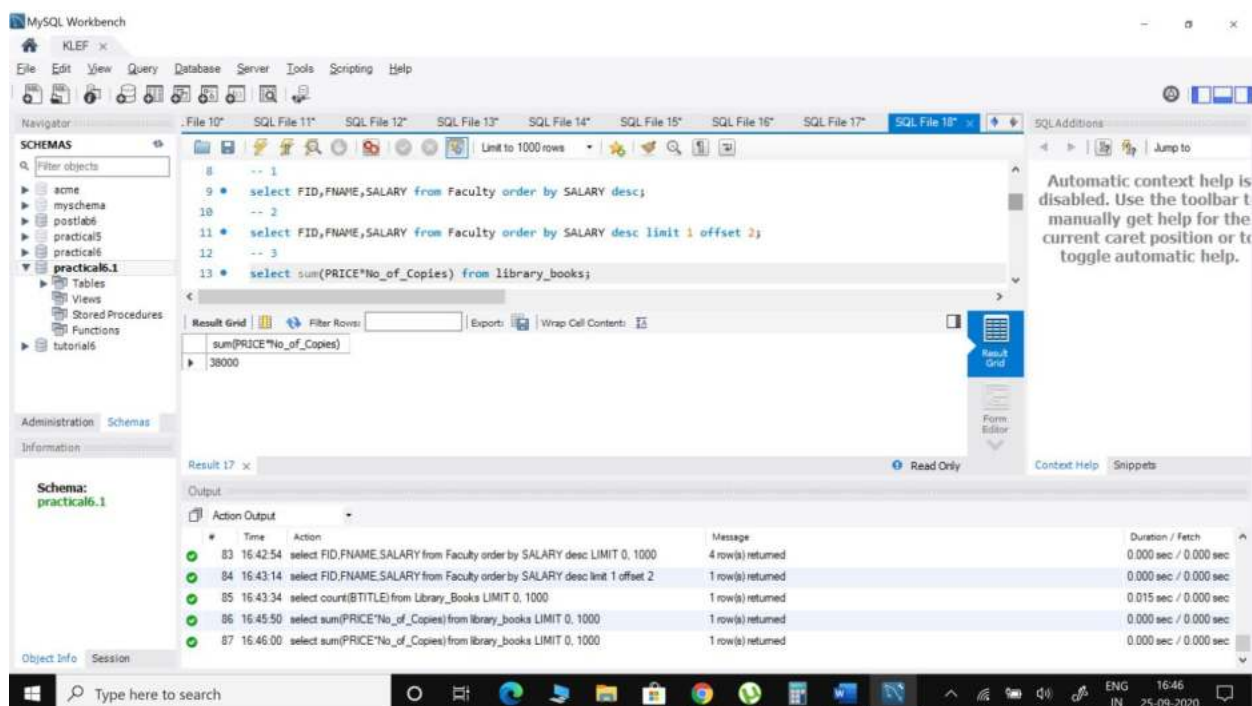
The Output tab shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
79	16:40:27	select count(REGNO) from student_details where crame in (select crame from student...	1 row(s) returned	0.031 sec / 0.000 sec
80	16:40:40	create unique index utid on Student(REGNO,NAME,MOBILENO,EMAILADDRESS,BR...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.265 sec
81	16:40:47	show indexes from Student	7 row(s) returned	0.062 sec / 0.000 sec
82	16:41:30	select * from Student LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
83	16:42:54	select FID,FNAME,SALARY from Faculty order by SALARY desc LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

- 2) Write a query to get the third highest salary of a faculty



- 3) Write a query to display the total value of books available in the Library using aggregate function.



- 4) Write a query to display the Regd.no, name and the Average CGPA of each student for the entire 1st year using aggregate and group by clause

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

```

13 * select sum(PRICE*No_of_Copies) from library_books;
14 -- 4
15 * select a.REGID,s.NAME,avg(CGPA) from Acad_Performance a,Student s
16 where s.REGNO = a.REGID and a.YEAR=1 group by s.NAME;
17 -- 5
18 * select count(CCODE),BRANCH from Course where YEAR = 2 group by BRANCH having count(CCODE) >=2;

```

The Results window displays the output of the third query:

REGID	NAME	avg(CGPA)
1000	Hari	9.300000190734863
1001	Jaya	9.199999809265137
1002	Kiran	9.100000381469727
2000	Gopal	9.100000381469727
2001	Kalyan	9.300000190734863
3000	Suresh	9.199999809265137

The Action Output window shows the execution details of the queries.

- 5) Write a query to display the number of courses available in 2nd year branch wise, that contains atleast two courses per branch using Aggregate Functions with GroupBy & Having Clauses

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

```

14 -- 4
15 * select a.REGID,s.NAME,avg(CGPA) from Acad_Performance a,Student s
16 where s.REGNO = a.REGID and a.YEAR=1 group by s.NAME;
17 -- 5
18 * select count(CCODE),BRANCH from Course where YEAR = 2 group by BRANCH having count(CCODE) >=2;
19 -- 6

```

The Results window displays the output of the third query:

count(CCODE)	BRANCH
3	CSE

The Action Output window shows the execution details of the queries.

- 6) Write a query to display the students and the book name issued to him/her in the month of May.

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

```

17 -- 5
18 * select count(CCODE),BRANCH from Course where YEAR = 2 group by BRANCH having count(CCODE) >=2;
19 -- 6
20 * select s.NAME,l.BTITLE,l.ISSUEDATE from Student s,Library_Books l,ISSUE_REGESTER i
21 where i.REGNO = s.REGNO and l.ACCNO = i.ACCNO and ISSUEDATE like 'M/05/%';
22 -- 7

```

The Result Grid shows the following data:

NAME	BTITLE	ISSUEDATE
Gopal	DBMS	01/05/2020
Kiran	DBMS	09/05/2020

The Action Output pane shows the execution of the query, with a message indicating that 1 row(s) were returned.

- 7) Write an SQL query to display the names of students, their branch, the courses they have registered, and the faculty teaching the course.

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

```

20 * select s.NAME,l.BTITLE,l.ISSUEDATE from Student s,Library_Books l,ISSUE_REGESTER i
21 where i.REGNO = s.REGNO and l.ACCNO = i.ACCNO and ISSUEDATE like 'M/05/%';
22 -- 7
23 * select s.name,s.branch,c.CCODE,c.CNAME,f.FNAME from Student s,Course c,Stu_Reg_Courses st,Faculty f
24 where s.REGNO = st.REGNO and c.CCODE = st.COURSECODE and st.FID = f.FID;
25 -- 9

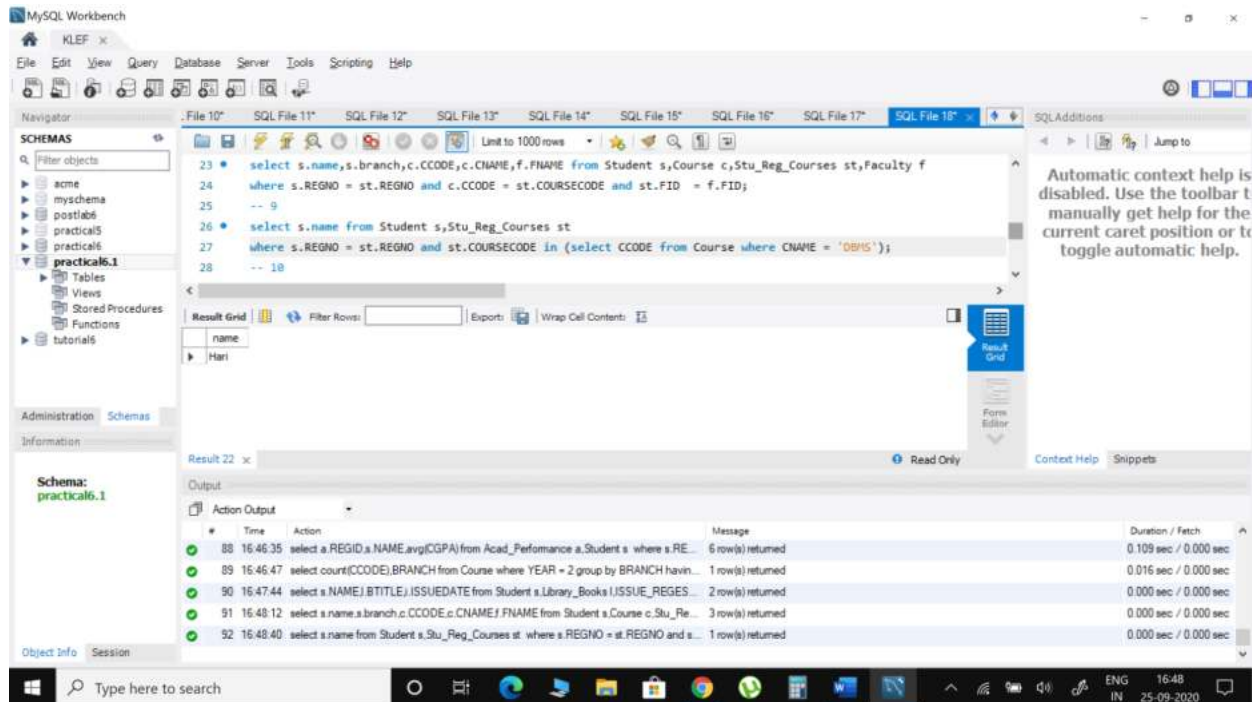
```

The Result Grid shows the following data:

name	branch	CCODE	CNAME	FNAME
Hari	CSE	18CS2101	DBMS	Hari
Jaya	CSE	18CS2102	EP	Krishna
Kiran	CSE	18CS2103	Os	Krishna

The Action Output pane shows the execution of the query, with a message indicating that 3 row(s) were returned.

- 8) Write a query to display the names of students who have joined the hostel using a Nested Query
- 9) Write a query to display the student name who has registered for DBMS Course using Nested Query



- 10) Write a query to display the titles of the three most expensive books available in the Library.

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

```

25 -- 9
26 select s.name from Student s,Stu_Reg_Courses st
27 where s.REGNO = st.REGNO and st.COURSECODE in (select CCODE from Course where CNAME = 'DBMS');
28 -- 10
29 select BTITLE,PRICE from Library_Books order by PRICE desc limit 3;
30 -- 11

```

The Result Grid shows the following data:

BTITLE	PRICE
Let Us C	600
JavaCompleteReference	500
DBMS	350

The Output tab shows the execution log with the following messages:

#	Time	Action	Message	Duration / Fetch
89	16:46:47	select count(CCODE),BRANCH from Course where YEAR = 2 group by BRANCH havin...	1 row(s) returned	0.016 sec / 0.000 sec
90	16:47:44	select s.NAME, BTITLE, ISSUEDATE from Student s, Library_Books l, ISSUE_REGES...	2 row(s) returned	0.000 sec / 0.000 sec
91	16:48:12	select s.name, s.branch, c.CCODE, c.CNAME, f.FNAME from Student s, Course c, Stu_Re...	3 row(s) returned	0.000 sec / 0.000 sec
92	16:48:40	select s.name from Student s, Stu_Reg_Courses st where s.REGNO = st.REGNO and s...	1 row(s) returned	0.000 sec / 0.000 sec
93	16:48:58	select BTITLE, PRICE from Library_Books order by PRICE desc limit 3	3 row(s) returned	0.000 sec / 0.000 sec

11) Write a query to create a View named Student_Details with the following data: Registration No., Name, Mobile No., Branch, Registered Course Name, Faculty Name

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

```

29 select BTITLE,PRICE from Library_Books order by PRICE desc limit 3;
30 -- 11
31 create view Student_Details as select s.REGNO, s.NAME, s.MOBILENO, s.BRANCH, c.CNAME, f.FNAME
32 from Student s left join stu_reg_courses st on s.REGNO = st.REGNO
33 left join Course c on st.COURSECODE = c.CCODE left join Faculty f on st.FID = f.FID;
34 select * from Student_Details;

```

The Result Grid shows the following data:

REGNO	NAME	MOBILENO	BRANCH	CNAME	FNAME
1000	Hari	9988776555	CSE	DBMS	Hari
1001	Jaya	9876543246	CSE	EP	Krishna
1002	Kiran	7864569878	CSE	Os	Krishna
2000	Gopal	7654328998	ECE	DBMS	DBMS
2001	Kalyan	8765498755	ECE	DBMS	DBMS
3000	Suresh	8067543567	EEE	DBMS	DBMS

The Output tab shows the execution log with the following messages:

#	Time	Action	Message	Duration / Fetch
91	16:48:12	select s.name, s.branch, c.CCODE, c.CNAME, f.FNAME from Student s, Course c, Stu_Re...	3 row(s) returned	0.000 sec / 0.000 sec
92	16:48:40	select s.name from Student s, Stu_Reg_Courses st where s.REGNO = st.REGNO and s...	1 row(s) returned	0.000 sec / 0.000 sec
93	16:48:58	select BTITLE, PRICE from Library_Books order by PRICE desc limit 3	3 row(s) returned	0.000 sec / 0.000 sec
94	16:49:11	create view Student_Details as select s.REGNO, s.NAME, s.MOBILENO, s.BRANCH, c.C...	Error Code: 1050. Table 'Student_Details' already exists	0.000 sec
95	16:49:13	select * from Student_Details LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

- 12) Write a query to find the number of students who have registered for at least one using the view named Student_Details which already has the following columns: Registration No., Name, Mobile No., Branch, Registered Course Name, Faculty Name.

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

```

31 create view Student_Details as select s.REGNO,s.NAME,s.MOBILENO,s.BRANCH,c.CNAME,f.FNAME
32 from Student s left join stu_reg_courses st on s.REGNO = st.REGNO
33 left join Course c on st.COURSECODE = c.CODE left join Faculty f on st.FID = f.FID;
34 select *from Student_Details;
35 -- 12
36 select count(REGNO) from student_details where cname in (select cname from student_details);

```

The Result Grid shows the output of the query:

count(REGNO)
3

The Action Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
92	16:48:40	select s.name from Student s,Stu_Reg_Courses st where s.REGNO = st.REGNO and s...	1 row(s) returned	0.000 sec / 0.000 sec
93	16:48:58	select BTITLE.PRICE from Library_Books order by PRICE desc limit 3	3 row(s) returned	0.000 sec / 0.000 sec
94	16:49:11	create view Student_Details as select s.REGNO.s.NAME.s.MOBILENO.s.BRANCH.c.C...	Error Code: 1050. Table 'Student_Details' already exists	0.000 sec
95	16:49:13	select *from Student_Details LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
96	16:49:33	select count(REGNO) from student_details where cname in (select cname from student...	1 row(s) returned	0.000 sec / 0.000 sec

- 13) Write a query to create a Unique Index value for the Students

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

```

34 select *from Student_Details;
35 -- 12
36 select count(REGNO) from student_details where cname in (select cname from student_details);
37 -- 13
38 create unique index utid on Student(REGNO,NAME,MOBILENO,EMAIL,ADDRESS,BRANCH);
39 show indexes from Student;

```

The Result Grid shows the output of the query:

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment
student	0	PRIMARY	1	REGNO	A	2				BTREE	
student	0	utid	1	REGNO	A	6				BTREE	
student	0	utid	2	NAME	A	6			YES	BTREE	
student	0	utid	3	Mobileno	A	6			YES	BTREE	
student	0	utid	4	EMAIL	A	6			YES	BTREE	

The Action Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
93	16:48:58	select BTITLE.PRICE from Library_Books order by PRICE desc limit 3	3 row(s) returned	0.000 sec / 0.000 sec
94	16:49:11	create view Student_Details as select s.REGNO.s.NAME.s.MOBILENO.s.BRANCH.c.C...	Error Code: 1050. Table 'Student_Details' already exists	0.000 sec
95	16:49:13	select *from Student_Details LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
96	16:49:33	select count(REGNO) from student_details where cname in (select cname from student...	1 row(s) returned	0.000 sec / 0.000 sec
97	16:49:44	show indexes from Student	7 row(s) returned	0.000 sec / 0.000 sec