

Lab6:-

```
CREATE TABLE SALESMAN(
    SALESMANID INT,
    SNAME VARCHAR(30),
    CITY VARCHAR(30),
    COMMISSION VARCHAR(30),
    PRIMARY KEY (SALESMANID));

CREATE TABLE CUSTOMER(
    CUSTOMERID INT,
    SALESMANID INT,
    CNAME VARCHAR(30),
    CITY VARCHAR(20),
    GRADE INT,
    PRIMARY KEY (CUSTOMERID),
    FOREIGN KEY (SALESMANID) REFERENCES SALESMAN(SALESMANID)ON DELETE SET NULL);

CREATE TABLE ORDERS(
    ORDERNO INT,
    CUSTOMERID INT,
    SALESMANID INT,
    PURCHASEAMT INT,
    ORDERDATE DATE,
    PRIMARY KEY (ORDERNO),
    FOREIGN KEY (SALESMANID) REFERENCES SALESMAN(SALESMANID) ON DELETE CASCADE,
    FOREIGN KEY (CUSTOMERID) REFERENCES CUSTOMER(CUSTOMERID)ON DELETE CASCADE);
```

```
INSERT INTO SALESMAN VALUES(1000,'JOHN','BANGALORE','25%');
```

```
INSERT INTO SALESMAN VALUES(2000,'RAVI','BANGALORE','20%');
```

```
INSERT INTO SALESMAN VALUES(3000,'KUMAR','MYSORE','15%');
```

```
INSERT INTO SALESMAN VALUES(4000,'SMITH','DELHI','30%');
```

```
INSERT INTO SALESMAN VALUES(5000,'HARSHA','HYDERABAD','15%');
```

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

- Schemas:** lab6
- Tables:** salesman
- Query Editor (Query 1):** Contains the SQL code used to insert data into the salesman table.
- Result Grid:** Displays the data inserted into the salesman table, showing five rows of salesperson information.
- Output:** Shows the history of actions taken, including the use of the lab6 database and the execution of the SELECT query.

SALESMANID	SNAME	CITY	COMMISSION
1000	JOHN	BANGALORE	25%
2000	RAVI	BANGALORE	20%
3000	KUMAR	MYSORE	15%
4000	SMITH	DELHI	30%
5000	HARSHA	HYDERABAD	15%

Action Output:

#	Time	Action	Message	Duration / Fetch
30	20:32:42	use lab6	0 row(s) affected	0.000 sec
31	20:32:42	SELECT * FROM salesman LIMIT 0, 1000	5 row(s) returned	0.016 sec / 0.000 sec

```
INSERT INTO CUSTOMER VALUES(10,1000,'PREETHI','BANGALORE',100);
```

```
INSERT INTO CUSTOMER VALUES(11,1000,'VIVEK','MANGALORE',300);
```

```
INSERT INTO CUSTOMER VALUES(12,2000,'BHASKAR','CHENNAI',400);
```

```
INSERT INTO CUSTOMER VALUES(13,2000,'CHETAN','BANGALORE',200);
```

```
INSERT INTO CUSTOMER VALUES(14,3000,'MAMTHA','BANGALORE',400);
```

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

145 lab1 lab2 lab3 lab4 lab5

Tables: aircraft certified employees flights

Views

Stored Procedures

Functions

lab6

Tables: customer orders salesman

Views

Stored Procedures

Functions

Administration Schemas

Information

Schema: lab6

customer 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
26	20:30:30	use lab6	0 row(s) affected	0.000 sec
27	20:30:30	SELECT * FROM customer LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

```
INSERT INTO ORDERS VALUES(50,10,1000,5000,'2017-05-04');
```

```
INSERT INTO ORDERS VALUES(51,10,2000,450,'2017-01-20');
```

```
INSERT INTO ORDERS VALUES(52,13,2000,1000,'2017-02-24');
```

```
INSERT INTO ORDERS VALUES(53,14,3000,3500,'2017-04-13');
```

```
INSERT INTO ORDERS VALUES(54,12,2000,550,'2017-03-09');
```

MySQL Workbench

Local instance MySQL8.0 X

File Edit View Query Database Server Tools Scripting Help

Navigators

SCHEMAS

- 145
- lab1
- lab2
- lab3
- lab4
- lab5
- lab6
 - Tables
 - aircraft
 - certified
 - employees
 - flights
 - Views
 - Stored Procedures
 - Functions

Administration Schemas

Information

Schema: lab6

orders 2 X

Output

Action Output	#	Time	Action	Message	Duration / Fetch
	28	20.31.11	use lab6	0 row(s) affected	0.000 sec
	29	20.31.11	SELECT * FROM orders LIMIT 0, 1000	5 row(s) returned	0.235 sec / 0.000 sec

Result Grid

ORDENO	CUSTOMERID	SALESMANID	PURCHASEAMT	ORDERDATE
50	10	1000	5000	2017-05-04
51	10	2000	450	2017-01-20
52	13	2000	1000	2017-02-24
53	14	3000	3500	2017-04-13
54	12	2000	550	2017-03-09
NULL	NULL	NULL	NULL	NULL

SQLAdditions

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

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

SELECT GRADE, COUNT(*) FROM CUSTOMER GROUP BY GRADE HAVING GRADE > (SELECT AVG(GRADE)FROM CUSTOMER WHERE CITY = 'BANGALORE');

MySQL Workbench Screenshot:

Query 1 (Result Grid):

GRADE	COUNT(*)
300	1
400	2

Action Output (Output Tab):

#	Time	Action	Message	Duration / Fetch
32	20:34:40	use lab6	0 row(s) affected	0.000 sec
33	20:34:40	SELECT GRADE, COUNT(*) FROM CUSTOMER GROUP BY GRADE HAVING GRADE > (SELECT AVG(GRADE) FROM CUSTOMER WHERE CITY = 'BANGALORE')	2 row(s) returned	0.031 sec / 0.000 sec

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

SELECT SALESMANID, SNAME FROM SALESMAN A WHERE 1 < (SELECT COUNT(*) FROM CUSTOMER WHERE SALESMANID=A.SALESMANID);

MySQL Workbench Screenshot:

Query 1 (Result Grid):

GRADE	COUNT(*)
300	1
400	2

Action Output (Output Tab):

#	Time	Action	Message	Duration / Fetch
32	20:34:40	use lab6	0 row(s) affected	0.000 sec
33	20:34:40	SELECT SALESMANID, SNAME FROM SALESMAN A WHERE 1 < (SELECT COUNT(*) FROM CUSTOMER WHERE SALESMANID=A.SALESMANID)	2 row(s) returned	0.031 sec / 0.000 sec

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

```
SELECT SALESMAN.SALESMANID,SNAME, CNAME, COMMISSION FROM SALESMAN, CUSTOMER WHERE  
SALESMAN.CITY = CUSTOMER.CITY UNION
```

```
SELECT SALESMANID, SNAME, 'NO MATCH', COMMISSION FROM SALESMAN WHERE NOT CITY = ANY  
(SELECT CITY FROM CUSTOMER) ORDER BY 2 DESC;
```

The screenshot shows the MySQL Workbench interface. In the top-left, the Navigator pane displays the database schema, including tables like aircraft, certified, employees, flights, customer, orders, and salesman. The current schema is set to lab6. In the center, the Query Editor window contains two queries:

```
1 •  
2 • 'NO MATCH', COMMISSION FROM SALESMAN WHERE NOT CITY = ANY (SELECT CITY FROM CUSTOMER) ORDER BY 2 DESC;
```

The Result Grid pane below shows the output of the second query:

SALESMANID	SNAME	CNAME	COMMISSION
4000	SMITH	NO MATCH	30%
2000	RAVI	PREETHI	20%
2000	RAVI	CHETRAN	20%
2000	RAVI	MAMTHA	20%
3000	KUMAR	NO MATCH	15%
1000	JOHN	PREETHI	25%
1000	JOHN	CHETRAN	25%
1000	JOHN	MAMTHA	25%
5000	HARSHA	NO MATCH	15%

In the bottom-right, the Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
34	20:39:48	use lab6	0 row(s) affected	0.000 sec
35	20:39:48	SELECT SALESMAN.SALESMANID,SNAME, CNAME, COMMISSION FROM SALESMAN, CUSTOMER WH...	9 row(s) returned	0.000 sec / 0.000 sec

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

```
CREATE VIEW VIEWSALESMAN AS
```

```
SELECT B.ORDERDATE, A.SALESMANID, A.SNAME FROM SALESMAN A, ORDERS B WHERE A.SALESMANID  
= B.SALESMANID AND B.PURCHASEAMT=(SELECT max(PURCHASEAMT) FROM ORDERS C WHERE  
C.ORDERDATE = B.ORDERDATE);
```

```
SELECT * FROM VIEWSALESMAN;
```

The screenshot shows the MySQL Workbench interface. In the top-left, the Navigator pane displays the database schema with tables like aircraft, certified, employees, flights, customer, orders, and salesman under schemas lab5 and lab6. The Query 1 tab contains the SQL command: `use lab6;` and `SELECT * FROM VIEWSALESMAN;`. The main area shows a Result Grid with data from the VIEWSALESMAN table:

ORDERDATE	SALESMANID	SNAME
2017-05-04	1000	JOHN
2017-01-20	2000	RAVI
2017-02-24	2000	RAVI
2017-04-13	3000	KUMAR
2017-03-09	2000	RAVI

The bottom section shows the Action Output log:

#	Time	Action	Message	Duration / Fetch
38	20:42:18	use lab6	0 row(s) affected	0.000 sec
39	20:42:18	SELECT * FROM VIEWSALESMAN LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

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

`DELETE FROM SALESMAN WHERE SALESMANID=1000;`

`SELECT * FROM SALESMAN;`

```

use lab6;
DELETE FROM SALESMAN WHERE SALESMANID=1000;
SELECT * FROM SALESMAN;

```

SALESMANID	SNAME	CITY	COMMISSION
2000	RAVI	BANGALORE	20%
3000	KUMAR	MYSORE	15%
4000	SMITH	DELHI	30%
5000	HARSHA	HYDERABAD	15%
NULL	NULL	NULL	NULL

Action Output

#	Time	Action	Message	Duration / Fetch
41	20:43:19	DELETE FROM SALESMAN WHERE SALESMANID=1000	1row(s) affected	0.375 sec
42	20:43:19	SELECT * FROM SALESMAN LIMIT 0, 1000	4row(s) returned	0.000 sec / 0.000 sec

Lab7:-

CREATE TABLE PUBLISHER

(PNAME VARCHAR (20),

PHONE DOUBLE,

ADDRESS VARCHAR (20),

PRIMARY KEY (PNAME));

CREATE TABLE BOOK

(BOOKID INT PRIMARY KEY,

TITLE VARCHAR (20),

PUBYEAR VARCHAR (20),

PUBLISHERNAME VARCHAR(20),

FOREIGN KEY (PUBLISHERNAME) REFERENCES PUBLISHER(PNAME) ON DELETE CASCADE);

CREATE TABLE BOOKAUTHORS

(

BOOKID INT,

AUTHORNAME VARCHAR (20),

```
FOREIGN KEY (BOOKID) REFERENCES BOOK(BOOKID)ON DELETE CASCADE,  
PRIMARY KEY (BOOKID,AUTHORNAME));  
  
CREATE TABLE LIBRARYBRANCH  
(BRANCHID INT PRIMARY KEY,  
BRANCHNAME VARCHAR(30),  
ADDRESS VARCHAR (30));  
  
CREATE TABLE BOOKCOPIES  
(NOOFCOPIES INT,  
BOOKID INT,  
BRANCHID INT,  
FOREIGN KEY (BOOKID) REFERENCES BOOK(BOOKID) ON DELETE CASCADE,  
FOREIGN KEY (BRANCHID) REFERENCES LIBRARYBRANCH(BRANCHID)ON DELETE CASCADE);  
  
CREATE TABLE BOOKLENDING  
(DATE_OUT DATE,  
DUE_DATE DATE,  
BOOKID INT,  
BRANCHID INT,  
CARDNO INT,  
FOREIGN KEY (BOOKID) REFERENCES BOOK(BOOKID) ON DELETE CASCADE,  
FOREIGN KEY (BRANCHID) REFERENCES LIBRARYBRANCH(BRANCHID)ON DELETE CASCADE,  
PRIMARY KEY (CARDNO,BOOKID,BRANCHID));
```

```
INSERT INTO PUBLISHER VALUES('MCGRAW-HILL',9989876587,'BANGALORE');  
INSERT INTO PUBLISHER VALUES('PEARSON',98899876565,'NEW DELHI');  
INSERT INTO PUBLISHER VALUES('RANDOM HOUSE',7455679345,'HYDERABAD');  
INSERT INTO PUBLISHER VALUES('HACHETTE LIVRE',8970862340,'CHENNAI');  
INSERT INTO PUBLISHER VALUES('GRUPO PLANETA',7756120238,'BANGALORE');
```

MySQL Workbench

Local instance MySQL8.0 X

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

145
lab1
lab2
lab3
lab4
lab5
lab6
lab7

Tables

book
bookauthors
bookcopies
booklending
librarybranch
publisher
Views
Stored Procedures
Functions

lt;
sakila
sys
world

Administration Schemas

Information

Schema: lab7

Query 1

1 • use lab7;
2 • select * from publisher;

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

PNAME	PHONE	ADDRESS
GRUPO PLANETA	7756120238	BANGALORE
HACHETTE LIVRE	8970862340	CHENNAI
MCGRAW-HILL	9599876587	BANGALORE
PEARSON	9889976565	NEW DELHI
RANDOM HOUSE	7455679245	HYDERABAD
HILL	HILL	HILL

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

Output

Action Output

#	Time	Action	Message	Duration / Fetch
95	23.12.39	use lab7	0 row(s) affected	0.000 sec
96	23.12.39	select * from publisher LIMIT 0, 1000	5 row(s) returned	0.109 sec / 0.000 sec

Object Info Session

INSERT INTO BOOK VALUES (1,'DBMS','JAN-2017', 'MCGRAW-HILL');

INSERT INTO BOOK VALUES (2,'ADBMS','JUN-2016', 'MCGRAW-HILL');

INSERT INTO BOOK VALUES (3,'CN','SEP-2016', 'PEARSON');

INSERT INTO BOOK VALUES (4,'CG','SEP-2015', 'GRUPO PLANETA');

INSERT INTO BOOK VALUES (5,'OS','MAY-2016', 'PEARSON');

MySQL Workbench

Local instance MySQL8.0 X

File Edit View Query Database Server Tools Scripting Help

Navigat or

SCHEMAS

Filter objects

145
lab1
lab2
lab3
lab4
lab5
lab6
lab7

Tables

book
bookauthors
bookcopies
booklending
librarybranch
publisher

Views

Stored Procedures

Functions

ltl
sakila
sys
world

Administration Schemas

Information

Schema: lab7

Query 1

1 • use lab7;
2 • select * from books;

Result Grid

BOOKID	TITLE	PUBLISHYEAR	PUBLISHERNAME
1	DBMS	JAN-2017	MCGRAW-HILL
2	ADAMS	JUN-2016	MCGRAW-HILL
3	CN	SEP-2016	PEARSON
4	CG	SEP-2015	GRUPO PLANETA
5	OS	MAY-2016	PEARSON

book 8 X

Output

Action Output

#	Time	Action	Message	Duration / Fetch
85	23:08:27	use lab7	0 row(s) affected	0.016 sec
86	23:08:27	select * from book	LIMIT 0, 1000	0.000 sec / 0.000 sec

Object Info Session

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

```
INSERT INTO BOOKAUTHORS VALUES (1,'NAVATHE');

INSERT INTO BOOKAUTHORS VALUES (2,'NAVATHE');

INSERT INTO BOOKAUTHORS VALUES (3,'TANENBAUM');

INSERT INTO BOOKAUTHORS VALUES (4,'EDWARD ANGEL');

INSERT INTO BOOKAUTHORS VALUES (5,'GALVIN');
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the Navigator with the 'Schemas' tab selected, showing the 'lab7' schema which contains tables like book, bookauthors, bookcopies, booklending, librarybranch, and publisher. The main area has a 'Query 1' tab open with the following SQL code:

```
1 • use lab7;
2 • select * from bookauthors;
```

The results are displayed in a 'Result Grid' table:

BOOKID	AUTHORNAME
1	NAVATHE
2	NAVATHE
3	TANENBAUM
4	EDWARD ANGEL
5	GALVIN
6	MAULI

The bottom section shows the 'bookauthors 9' output pane with the following log entries:

#	Time	Action	Message	Duration / Fetch
87	23.09.12	use lab7	0 row(s) affected	0.000 sec
88	23.09.12	select * from bookauthors LIMIT 0, 1000	5 row(s) returned	0.031 sec / 0.000 sec

```
INSERT INTO LIBRARYBRANCH VALUES (10,'RR NAGAR','BANGALORE');
```

```
INSERT INTO LIBRARYBRANCH VALUES (11,'RNSIT','BANGALORE');
```

```
INSERT INTO LIBRARYBRANCH VALUES (12,'RAJAJI NAGAR', 'BANGALORE');
```

```
INSERT INTO LIBRARYBRANCH VALUES (13,'NITTE','MANGALORE');
```

```
INSERT INTO LIBRARYBRANCH VALUES (14,'MANIPAL','UDUPI');
```

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

SCHEMAS: lab7

Tables: book, bookauthors, bookcopies, booklending, librarybranch, publisher

Views

Stored Procedures

Functions

It.

sakila

sys

world

Administration Schemas Information Schema: lab7

Query 1

```
1 • use lab7;
2 • select * from librarybranch;
```

Result Grid

BRANCHID	BRANCHNAME	ADDRESS
10	RK NAGAR	BANGALORE
11	RINST	BANGALORE
12	RAJAJI NAGAR	BANGALORE
13	NITTE	MANGALORE
14	MANDAL	UDUPI
NULL	NULL	NULL

SQLAdditions

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

librarybranch 12

Action Output

#	Time	Action	Message	Duration / Fetch
93	23.11.21	use lab7	0 row(s) affected	0.000 sec
94	23.11.21	select * from librarybranch LIMIT 0, 1000	5 row(s) returned	0.016 sec / 0.000 sec

Object Info Session

INSERT INTO BOOKCOPIES VALUES (10, 1, 10);

INSERT INTO BOOKCOPIES VALUES (5, 1, 11);

INSERT INTO BOOKCOPIES VALUES (2, 2, 12);

INSERT INTO BOOKCOPIES VALUES (5, 2, 13);

INSERT INTO BOOKCOPIES VALUES (7, 3, 14);

INSERT INTO BOOKCOPIES VALUES (1, 5, 10);

INSERT INTO BOOKCOPIES VALUES (3, 4, 11);

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

145
lab1
lab2
lab3
lab4
lab5
lab6
lab7

Tables: book, bookauthors, bookcopies, booklending, librarybranch, publisher

Views, Stored Procedures, Functions, etc., sakila, sys, world

Administration Schemas Information

Schema: lab7

Query 1

```
1 • use lab7;
2 • select * from bookcopies;
```

Result Grid

NOFCOPIES	BOOKID	BRANCHID
10	1	10
5	1	11
2	2	12
5	2	13
7	3	14
1	5	10
3	4	11

bookcopies 10 x

Action Output

#	Time	Action	Message	Duration / Fetch
89	23.09.52	use lab7	0 row(s) affected	0.000 sec
90	23.09.52	select * from bookcopies LIMIT 0,1000	7 row(s) returned	0.078 sec / 0.000 sec

Object Info Session

SQLAdditions

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

INSERT INTO BOOKLENDING VALUES ('2017-01-01','2017-06-01', 1, 10, 101);

INSERT INTO BOOKLENDING VALUES ('2017-01-11','2017-03-11', 3, 14, 101);

INSERT INTO BOOKLENDING VALUES ('2017-02-21','2017-04-21', 2, 13, 101);

INSERT INTO BOOKLENDING VALUES ('2017-03-15','2017-07-15', 4, 11, 101);

INSERT INTO BOOKLENDING VALUES ('2017-04-12','2017-05-12', 1, 11, 104);

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigat... SCHEMAS Filter objects

- 145
- lab1
- lab2
- lab3
- lab4
- lab5
- lab6
- lab7**

Tables

- book
- bookauthors
- bookcopies
- booklending
- librarybranch
- publisher

Views

Stored Procedures

Functions

I1t

sakila

sys

world

Administration Schemas

Information

Schema: lab7

Query 1

```
1 • use lab7;
2 • select * from booklending;
```

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

DATE_OUT	DUE_DATE	BOOKID	BRANCHID	CARDNO
2017-01-01	2017-06-01	1	10	101
2017-02-21	2017-04-21	2	13	101
2017-01-11	2017-03-11	3	14	101
2017-03-15	2017-07-15	4	11	101
2017-04-12	2017-05-12	1	11	104
NULL	NULL	NULL	NULL	NULL

SQLAdditions

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

booklending11 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
91	23:10:30	use lab7	0 row(s) affected	0.000 sec
92	23:10:30	select * from booklending LIMIT 0, 1000	5 row(s) returned	0.032 sec / 0.000 sec

Object Info Session

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

```
SELECT B.BOOKID, B.TITLE, B.PUBLISHERNAME, A.AUTHORNAME, C.NOOCOPIES, L.BRANCHID FROM
BOOK B, BOOKAUTHORS A, BOOKCOPIES C, LIBRARYBRANCH L WHERE B.BOOKID=A.BOOKID AND
B.BOOKID=C.BOOKID AND L.BRANCHID=C.BRANCHID;
```

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigat... SCHEMAS Filter objects

- 145
- lab1
- lab2
- lab3
- lab4
- lab5
- lab6
- lab7**

Tables

- book
- bookauthors
- bookcopies
- booklending
- librarybranch
- publisher

Views

Stored Procedures

Functions

I1t

sakila

sys

world

Administration Schemas

Information

Schema: lab7

Query 1

```
1 • use lab7;
2 • SELECT B.BOOKID, B.TITLE, B.PUBLISHERNAME, A.AUTHORNAME, C.NOOCOPIES, L.BRANCHID FROM BOOK B, BOOKAUTHORS A, BOOKCOPIES C, LIBRARYBRANCH L WHERE B.BOOKID=A.BOOKID AND B.BOOKID=C.BOOKID AND L.BRANCHID=C.BRANCHID;
```

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

BOOKID	TITLE	PUBLISHERNAME	AUTHORNAME	NOOCOPIES	BRANCHID
1	DBMS	MCGRAW-HILL	NAVATHE	10	10
1	DBMS	MCGRAW-HILL	NAVATHE	5	11
2	ADBS	MCGRAW-HILL	NAVATHE	2	12
2	ADBS	MCGRAW-HILL	NAVATHE	5	13
3	CN	PEARSON	TANENBAUM	7	14
4	CG	GRUPO PLANETA	EDWARD ANGEL	3	11
5	OS	PEARSON	GALVIN	1	10

SQLAdditions

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

Result 14 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
97	23:13:50	use lab7	0 row(s) affected	0.000 sec
98	23:13:50	SELECT B.BOOKID, B.TITLE, B.PUBLISHERNAME, A.AUTHORNAME, C.NOOCOPIES, L.BRANCHID FROM BOOK B, BOOKAUTHORS A, BOOKCOPIES C, LIBRARYBRANCH L WHERE B.BOOKID=A.BOOKID AND B.BOOKID=C.BOOKID AND L.BRANCHID=C.BRANCHID;	7 row(s) returned	0.016 sec / 0.000 sec

Object Info Session

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

```
SELECT CARDNO FROM BOOKLENDING WHERE DATE_OUT BETWEEN '2017-01-01' AND '2017-07-01'
```

```
GROUP BY CARDNO HAVING Count(*)>3;
```

The screenshot shows the MySQL Workbench interface. The 'Navigator' pane on the left displays the database schema, including tables like book, bookauthors, bookcopies, booklending, librarybranch, and publisher. The 'Query 1' pane contains the SQL query:

```
1 • use lab7;
2 • SELECT CARDNO FROM BOOKLENDING WHERE DATE_OUT BETWEEN '2017-01-01' AND '2017-07-01' GROUP BY CARDNO HAVING Count(*)>3;
```

The 'Result Grid' pane shows the results of the query, which is currently empty. The 'Output' pane at the bottom shows the action log:

#	Time	Action	Message	Duration / Fetch
99	23.15.10	use lab7	0 row(s) affected	0.000 sec
100	23.15.10	SELECT CARDNO FROM BOOKLENDING WHERE DATE_OUT BETWEEN 2017-01-01 AND 2017-07-01 ...	1 row(s) returned	0.016 sec / 0.000 sec

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

```
DELETE FROM BOOK WHERE BOOKID='3';
```

```
SELECT * FROM BOOK;
```

```

use lab7;
DELETE FROM BOOK WHERE BOOKID='3';
SELECT * FROM BOOK;

```

Result Grid:

BOOKID	DBMS	PUBYEAR	PUBLISHERNAME
1	DBMS	JAN-2017	MCGRAW-HILL
2	DBMS	JUN-2016	MCGRAW-HILL
4	CG	SEP-2015	GRUPO PLANETA
5	OS	MAY-2016	PEARSON

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

CREATE VIEW V_PUBLICATION AS SELECT PUBYEAR FROM BOOK;

SELECT * FROM V_PUBLICATION;

```

use lab7;
CREATE VIEW V_PUBLICATION AS SELECT PUBYEAR FROM BOOK;
SELECT * FROM V_PUBLICATION;

```

Result Grid:

PUBYEAR
JAN-2017
JUN-2016
SEP-2015
MAY-2016

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

```
CREATE VIEW V_BOOKS AS SELECT B.BOOKID, B.TITLE, C.NOOFCOPIES FROM BOOK B, BOOKCOPIES C,  
LIBRARYBRANCH L WHERE B.BOOKID=C.BOOKID AND C.BRANCHID=L.BRANCHID;
```

```
SELECT * FROM V_BOOKS;
```

The screenshot shows the MySQL Workbench interface. In the left sidebar, under the 'Schemas' section, the 'lab7' schema is selected. The 'Tables' section lists 'book', 'bookauthors', 'bookcopies', 'booklending', 'librarybranch', and 'publisher'. The 'Views' section contains the newly created 'V_BOOKS'. The 'Query 1' tab at the top has the SQL code for creating the view and selecting from it. Below the query, the 'Result Grid' shows the results of the SELECT statement:

BOOKID	TITLE	NOOFCOPIES
1	DBMS	10
1	DBMS	5
2	ADDBMS	2
2	ADDBMS	5
4	CG	3
5	OS	1

The 'Output' tab at the bottom shows the execution log with two entries:

#	Time	Action	Message	Duration / Fetch
108	23:18:00	CREATE VIEW V_BOOKS AS SELECT B.BOOKID, B.TITLE, C.NOOFCOPIES FROM BOOK B, BOOKCOPIES C, LIBRARYBRANCH L WHERE B.BOOKID=C.BOOKID AND C.BRANCHID=L.BRANCHID;	0 row(s) affected	0.297 sec
109	23:18:00	SELECT * FROM V_BOOKS LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

Lab8:-

-- creating table

```
create table student(
```

```
regno varchar(10),
```

```
name varchar(30),
```

```
major varchar(10),
```

```
bdate date,
```

```
constraint stu_reg primary key (regno)
```

```
);
```

```
create table course (
```

```
course int ,
```

```

cname varchar(30),
dept varchar (30),
constraint cou_cou primary key (course)
);
create table enroll (
regno varchar(10) ,
cname varchar (30),
sem int ,
marks int,
constraint en_reg foreign key (regno) references student(regno)
);
create table book_adoption (
course int,
sem int,
book_isbn int,
constraint book_cou foreign key (course) references course(course) on delete cascade on update
cascade ,
constraint book_book foreign key (book_isbn) references text(book_isbn) on delete cascade on update
cascade
);
create table text (
book_isbn int,
book_title varchar(30),
publisher varchar(30),
author varchar (30),
constraint book_book primary key (book_isbn)
);
-- ii. Inserting tuples
insert into student values("CS01", "PRANAV", "DS", "1986-03-12");

```

```

insert into student values("IS02", "PRATEEK", "USP", "1987-12-23");
insert into student values("EC03", "SAURAB", "SNS", "1985-04-17");
insert into student values("CS03", "ARKA", "DBMS", "1987-01-01");
insert into student values("TC05", "PRANSHU", "EC", "1986-10-06");

```

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

- Navigator:** Shows the database schema with the current schema set to **lab8**. It lists tables like **book_adoption**, **course**, **enroll**, **student**, and **text**.
- Query Editor:** Contains two queries:
 - use lab8;
 - select* from student;
- Result Grid:** Displays the results of the second query, showing the following data:

regno	name	major	bdate
CS01	PRANAV	DS	1986-03-12
CS03	ARKA	DBMS	1987-01-01
EC03	SAURAB	SNS	1985-04-17
IS02	PRATEEK	USP	1987-12-23
TC05	PRANSHU	EC	1986-10-06
- Output:** Shows the action history with two entries:

#	Time	Action	Message	Duration / Fetch
155	23:29:22	use lab8	0 row(s) affected	0.000 sec
156	23:29:22	select* from student LIMIT 0, 1000	5 row(s) returned	0.015 sec / 0.000 sec

```

insert into course values(11,"DS","CS");
insert into course values(22,"USP","IS");
insert into course values(33,"SNS","EC");
insert into course values(44,"DBMS","CS");
insert into course values(55,"EC","TC");

```

The screenshot shows the MySQL Workbench interface. In the top-left, the Navigator pane displays the schema structure under 'lab8'. The 'Tables' section lists 'book_adoption', 'course', 'enroll', 'student', and 'text'. The 'Views' section lists 'lt'. The 'Stored Procedures' and 'Functions' sections are empty. The 'Administration' and 'Schemas' tabs are also visible. The main area contains a 'Query 1' tab with the following SQL code:

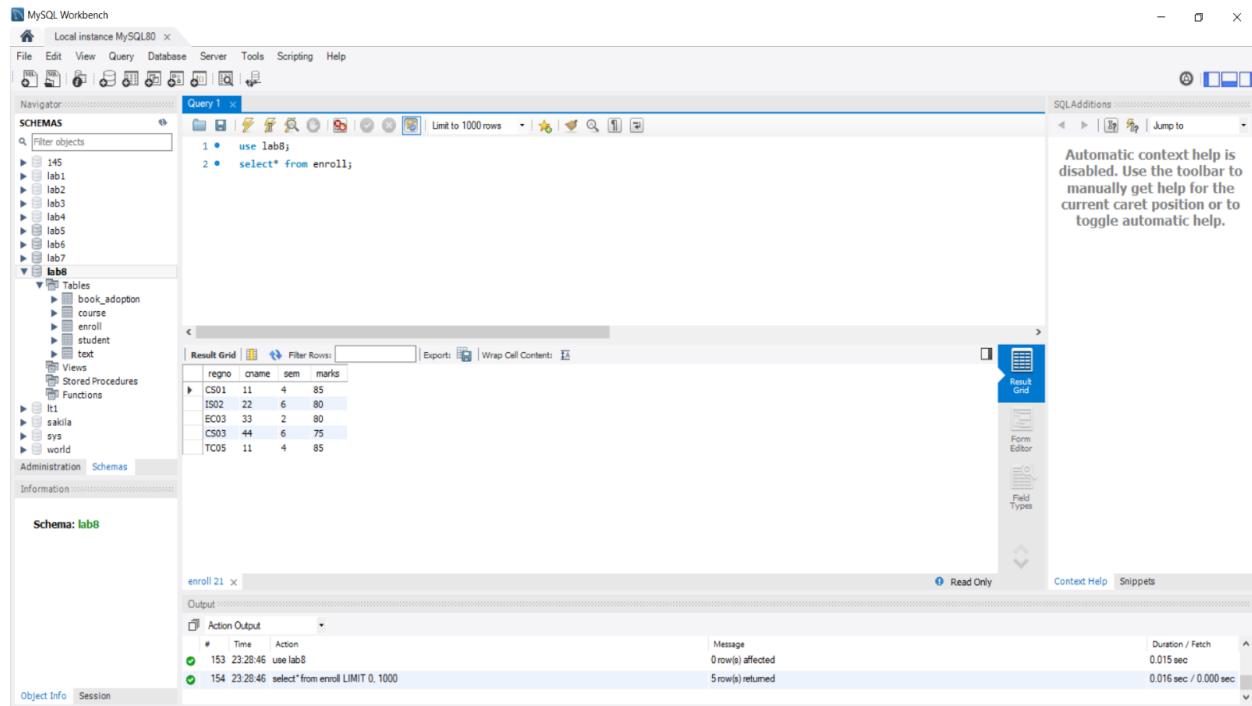
```
1 • use lab8;
2 • select* from course;
```

Below the code is a 'Result Grid' showing the data from the 'course' table:

course	name	dept
11	DS	CS
22	USP	IS
33	SNS	EC
44	DBMS	CS
55	EC	TC
	NAME	NAME

On the right side of the interface, there is a 'SQL Additions' panel with the message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help." Below this are three buttons: 'Result Grid', 'Form Editor', and 'Field Types'.

```
insert into enroll values("CS01", 11, 4, 85);
insert into enroll values("IS02", 22, 6, 80);
insert into enroll values("EC03", 33, 2, 80);
insert into enroll values("CS03", 44, 6, 75);
insert into enroll values("TC05", 11, 4, 85);
```



```
insert into text values(1, "DS AND C", "PRINCETON", "PADMA REDDY");
insert into text values(2, "FUNDAMENTALS OF DS", "SPRINGER", "GODSE");
insert into text values(3, "FUNDAMENTALS OF DBMS", "SPRINGER", "NAVATHE");
insert into text values(4, "SQL", "PRINCETON", "FOLEY");
insert into text values(5, "ELECTRONIC CIRCUITS", "TMH", "ELMASRI");
```

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

SCHEMAS: lab8

Tables: book_adoption, course, enroll, student, text

VIEWS: lab8

Stored Procedures: lab8

FUNCTIONS: lab8

TRIGGERS: lab8

ROUTINES: lab8

FUNCTIONS: lab8

Information: Schema: lab8

Object Info Session

Query 1

```
use lab8;
select* from text;
```

Result Grid

book_idn	book_title	publisher	author
1	DS AND C	PRINCETON	PADMA REDDY
2	FUNDAMENTALS OF DS	SPRINGER	GODSE
3	FUNDAMENTALS OF DBMS	SPRINGER	NAVATHE
4	SQL	PRINCETON	FOLEY
5	ELECTRONIC CIRCUITS	TMH	ELMASRI

SQL Additions

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

text 23

Action Output

#	Time	Action	Message	Duration / Fetch
157	23:30:07	use lab8	0 row(s) affected	0.000 sec
158	23:30:07	select* from text LIMIT 0, 1000	5 row(s) returned	0.031 sec / 0.000 sec

Output

Apply Revert Context Help Snippets

```
insert into book_adoption values(11, 4 ,1);
insert into book_adoption values(11, 4 ,2);
insert into book_adoption values(44, 6 ,3);
insert into book_adoption values(44, 6 ,4);
insert into book_adoption values(55, 2 ,5);
```

MySQL Workbench Screenshot showing the 'book_adoption' table structure and data.

Schema: lab8

Table: book_adoption

Query 1:

```

1 • use lab8;
2 • select* from book_adoption;
    
```

Result Grid:

course	sem	book_idn
11	4	1
11	4	2
44	6	3
44	6	4
55	2	5
22	6	6

Action Output:

#	Time	Action	Message	Duration / Fetch
149	23:27:31	use lab8	0 row(s) affected	0.000 sec
150	23:27:31	select* from book_adoption LIMIT 0, 1000	5 row(s) returned	0.594 sec / 0.000 sec

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

insert into text values(6, "RELATIONAL ALGEBRA", "ABS", "HUGH DARWIN");

insert into book_adoption values(22, 6 ,6);

MySQL Workbench Screenshot showing the 'book_adoption' table structure and data after insertion.

Schema: lab8

Table: book_adoption

Query 1:

```

1 • use lab8;
2 • select* from book_adoption;
    
```

Result Grid:

course	sem	book_idn
11	4	1
11	4	2
44	6	3
44	6	4
55	2	5
22	6	6
22	6	7

Action Output:

#	Time	Action	Message	Duration / Fetch
162	23:31:37	use lab8	0 row(s) affected	0.000 sec
163	23:31:37	select* from book_adoption LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

-- 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.

```
select b.book_isbn from book_adoption b where b.course in (select c.course from course c where c.dept="CS") and (select count(*) from book_adoption be where be.course=b.course)>=2 ;
```

```
insert into course values(66,"OS","CS");
```

```
insert into text values(7, "OPERATING SYSTEM", "MC GRAW", "ABRAHAM SILBERSCHATZ");
```

```
insert into book_adoption values(66, 6 ,7);
```

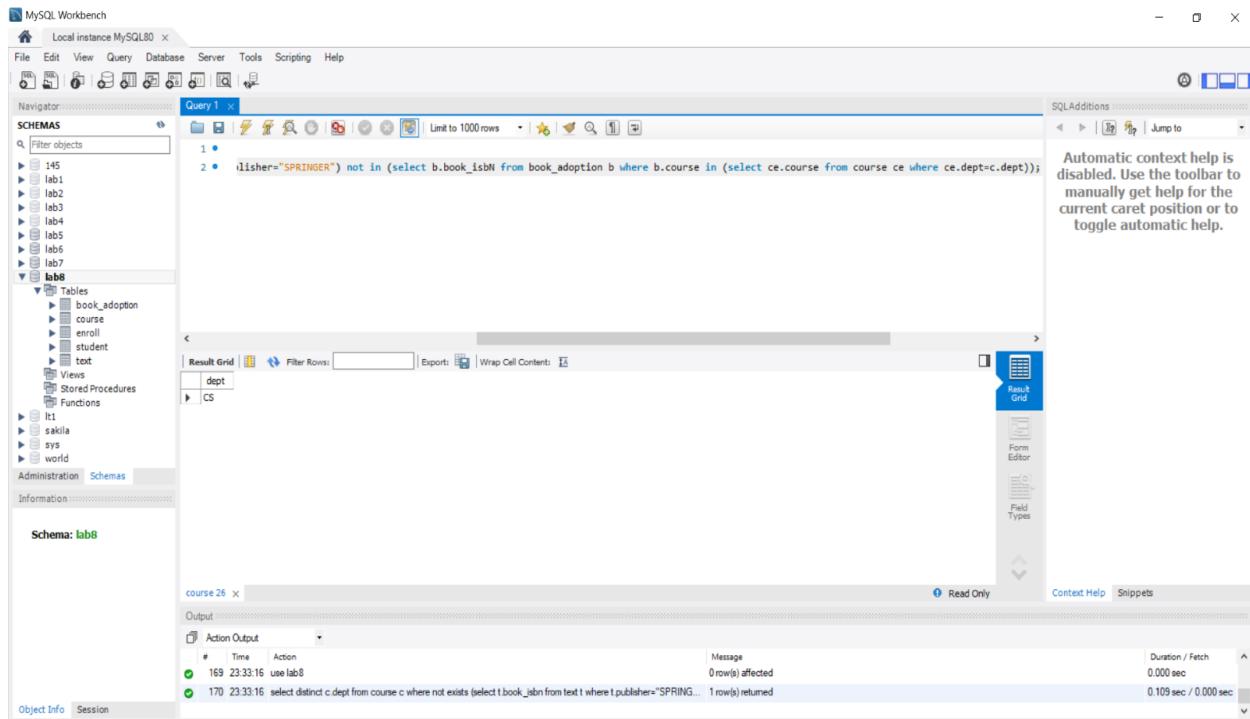
The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema for 'lab8'. It includes tables like book_adoption, course, enroll, student, and text, along with views, stored procedures, and functions.
- Query Editor:** Contains five numbered SQL statements:
 - use lab8;
 - select b.book_isbn from book_adoption b where b.course in (select c.course from course c where c.dept="CS") and (select count(*) from book_adoption be where be.course=b.course)>=2 ;
 - insert into course values(66,"OS","CS");
 - insert into text values(7, "OPERATING SYSTEM", "MC GRAW", "ABRAHAM SILBERSCHATZ");
 - insert into book_adoption values(66, 6 ,7);
- Result Grid:** Displays the results of the last query, showing the 'book_isbn' column with values 1, 2, 3, and 4.
- Output:** Shows the log of actions taken:

Action	Time	Message	Duration / Fetch
insert into text values(7, "OPERATING SYSTEM", "MC GRAW", "ABRAHAM SILBERSCHATZ")	23:32:24	1 row(s) affected	0.109 sec
insert into book_adoption values(66, 6 ,7)	23:32:24	1 row(s) affected	0.125 sec

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

```
select distinct c.dept from course c where not exists (select t.book_isbn from text t where t.publisher="SPRINGER") not in (select b.book_isbn from book_adoption b where b.course in (select ce.course from course ce where ce.dept=c.dept));
```



Lab9:-

```

create table actor(
    act_id int,
    act_name varchar(30),
    act_gender varchar(1),
    constraint act_id primary key (act_id)
);

create table director(
    dir_id int,
    dir_name varchar(30),
    dir_phone varchar(30),
    constraint dir_id primary key (dir_id)
);

create table movies(
    mov_id int,
    mov_title varchar(30),

```

```
mov_year varchar(4),
mov_lang varchar(12),
dir_id int ,
constraint mov_id primary key (mov_id),
constraint mov_id foreign key (dir_id) references director(dir_id)
);
create table movie_cast (
act_id int,
mov_id int,
role varchar(10),
constraint mov_pr primary key(act_id ,mov_id),
constraint mov_act foreign key (act_id) references actor(act_id),
constraint mov_mov foreign key (mov_id) references movies(mov_id)
);
create table rating (
mov_id int ,
rev_stars varchar(25),
constraint rat primary key (mov_id),
constraint rat_mov foreign key (mov_id) references movies(mov_id));
```

```
insert into actor values (301,'anushka','f');
insert into actor values (302,'prabhas','m');
insert into actor values (303,'punith','m');
insert into actor values (304,'jeremy','m');
```

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

- Navigator:** Shows the schema **lab9** selected, containing tables: actor, director, movie_cast, movies, rating.
- Query Editor:** Displays the following SQL code:


```
1 • use lab9;
2 • select* from actor;
```
- Result Grid:** Shows the results of the query:

act_id	act_name	act_gender
301	anushka	f
302	prabhas	m
303	punith	m
304	jeremy	m
NULL	NULL	NULL
- Action Output:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
34	12:29:05	use lab9	0 row(s) affected	0.000 sec
35	12:29:05	select* from actor LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

INSERT INTO DIRECTOR VALUES (60,"RAJAMOULI", 8751611001);

INSERT INTO DIRECTOR VALUES (61,"HITCHCOCK", 7766138911);

INSERT INTO DIRECTOR VALUES (62,"FARAN", 9986776531);

INSERT INTO DIRECTOR VALUES (63,"STEVEN SPIELBERG", 8989776530);

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

- Navigator:** Shows the schema **lab9** selected, containing tables: actor, director, movie_cast, movies, rating.
- Query Editor:** Displays the following SQL code:


```
1 • use lab9;
2 • select* from director;
```
- Result Grid:** Shows the results of the query:

dr_id	dr_name	dr_phone
60	RAJAMOULI	8751611001
61	HITCHCOCK	7766138911
62	FARAN	9986776531
63	STEVEN SPIELBERG	8989776530
NULL	NULL	NULL
- Action Output:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
36	12:29:55	use lab9	0 row(s) affected	0.000 sec
37	12:29:55	select* from director LIMIT 0, 1000	4 row(s) returned	0.047 sec / 0.000 sec

```

INSERT INTO MOVIES VALUES (1001,"BAHUBALI-2", 2017, "TELAGU", 60);

INSERT INTO MOVIES VALUES (1002,"BAHUBALI-1", 2015, "TELAGU", 60);

INSERT INTO MOVIES VALUES (1003,"AKASH", 2008, "KANNADA", 61);

INSERT INTO MOVIES VALUES (1004,"WAR HORSE", 2011, "ENGLISH", 63);

```

mov_id	mov_title	mov_year	mov_lang	dr_id
1001	BAHUBALI-2	2017	TELAGU	60
1002	BAHUBALI-1	2015	TELAGU	60
1003	AKASH	2008	KANNADA	61
1004	WAR HORSE	2011	ENGLISH	63

```

INSERT INTO MOVIE_CAST VALUES (301, 1002, "HEROINE");

INSERT INTO MOVIE_CAST VALUES (301, 1001, "HEROINE");

INSERT INTO MOVIE_CAST VALUES (303, 1003, "HERO");

INSERT INTO MOVIE_CAST VALUES (303, 1002, "GUEST");

INSERT INTO MOVIE_CAST VALUES (304, 1004, "HERO");

INSERT INTO MOVIE_CAST VALUES (301, 1005, "HEROINE");

INSERT INTO MOVIE_CAST VALUES (303, 1005, "HERO");

```

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

- Navigator:** Shows the schema lab9, which contains tables: actor, director, movie_cast, movies, rating.
- Query Editor:** Displays the query: `use lab9;` `select* from movie_cast;`
- Result Grid:** Shows the data from the movie_cast table:

act_id	mov_id	role
301	1001	HEROINE
301	1002	HEROINE
303	1002	GUEST
303	1003	HERO
304	1004	HERO
NULL	NULL	NULL
- Output Panel:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
38	12:31:23	use lab9	0 row(s) affected	0.000 sec
39	12:31:23	select* from movie_cast LIMIT 0, 1000	5 row(s) returned	0.547 sec / 0.000 sec

INSERT INTO RATING VALUES (1001, 4);

INSERT INTO RATING VALUES (1002, 2);

INSERT INTO RATING VALUES (1003, 5);

INSERT INTO RATING VALUES (1004, 4);

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

- Navigator:** Shows the schema lab9, which contains tables: actor, director, movie_cast, movies, rating.
- Query Editor:** Displays the query: `use lab9;` `select* from rating;`
- Result Grid:** Shows the data from the rating table:

mov_id	rev_stars
1001	4
1002	2
1003	5
1004	4
NULL	NULL
- Output Panel:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
42	12:32:25	use lab9	0 row(s) affected	0.000 sec
43	12:32:25	select* from rating LIMIT 0, 1000	4 row(s) returned	0.016 sec / 0.000 sec

-- i. List the titles of all movies directed by "Hitchcock".

```
select m.mov_title from movies m where dir_id = (select dir_id from director where dir_name = "HITCHCOCK");
```

The screenshot shows the MySQL Workbench interface. In the top-left pane, the Navigator displays the schema 'lab9' with its tables: actor, director, movie_cast, movies, rating, and rating. The 'Tables' section is expanded. In the top-right pane, the SQL Editor contains the following query:

```
1 • use lab9;
2 • select m.mov_title from movies m where dir_id = (select dir_id from director where dir_name = "HITCHCOCK");
```

The bottom-right pane shows the results of the query in a Result Grid:

mov_title
AKASH

The bottom-left pane shows the Output window with the following log entries:

#	Time	Action	Message	Duration / Fetch
43	12:32:25	select` from rating LIMIT 0, 1000	4 row(s) returned	0.016 sec / 0.000 sec
44	12:33:00	select m.mov_title from movies m where dir_id = (select dir_id from director where dir_name = "HITCHCOCK") Li...	1 row(s) returned	0.016 sec / 0.000 sec

-- ii. Find the movie names where one or more actors acted in two or more movies.

```
select mo.mov_title from movies mo where mo.mov_id in (select c.mov_id from movie_cast c where c.act_id in (select m.act_id from movie_cast m group by m.act_id having count(*)>=2));
```

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

```

1 •
2 •    mo.mov_id in (select c.mov_id from movie_cast c where c.act_id in (select m.act_id from movie_cast m group by m.act_id having count(*)>2))

```

The result grid displays the following data:

mov_title
BAHUBALI-2
BAHUBALI-1
AKASH

Output window:

Action	Time	Action	Message	Duration / Fetch
45	12:33:38	use lab9	0 row(s) affected	0.000 sec
46	12:33:38	select mo.mov_title from movies mo where mo.mov_id in (select c.mov_id from movie_cast c where c.act_id in ...)	3 row(s) returned	0.000 sec / 0.000 sec

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

```

select distinct(m.act_id) from movie_cast m join movie_cast mo on m.act_id=mo.act_id and
m.mov_id!=mo.mov_id where m.mov_id in (select mo.mov_id from movies mo where
mo.mov_year>=2015) and mo.mov_id in (select mo.mov_id from movies mo where
mo.mov_year<2000);

```

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

```

1 •
2 •    mov_id in (select mo.mov_id from movies mo where mo.mov_year>=2015) and mo.mov_id in (select mo.mov_id from movies mo where mo.mov_year<2000)

```

The result grid displays the following data:

act_id

Output window:

Action	Time	Action	Message	Duration / Fetch
53	12:36:12	use lab9	0 row(s) affected	0.000 sec
54	12:36:12	select distinct(m.act_id) from movie_cast m join movie_cast mo on m.act_id=mo.act_id and m.mov_id!=mo.mov...	0 row(s) returned	0.000 sec / 0.000 sec

-- iv. 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,r.rev_stars from movies m join rating r where m.mov_id=r.mov_id group by m.mov_title having min(r.rev_stars)>=1 order by r.rev_stars desc ;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the Navigator with Schemas (lab9 selected), Tables (actor, director, movie_cast, movies, rating), Views, Stored Procedures, Functions, and sys. The main area contains a Query Editor window with the following content:

```
1 •
2 • r.rev_stars from movies m join rating r where m.mov_id=r.mov_id group by m.mov_title having min(r.rev_stars)>=1 order by r.rev_stars desc ;
```

The Result Grid shows the following data:

mov_title	rev_stars
AKASH	5
BAHUBALI-2	4
WAR HORSE	4
BAHUBALI-1	2

The Output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
55	12:37:08	use lab9	0 row(s) affected	0.000 sec
56	12:37:08	select m.mov_title,r.rev_stars from movies m join rating r where m.mov_id=r.mov_id group by m.mov_title having ... 4 row(s) returned		0.000 sec / 0.000 sec

-- v. Update rating of all movies directed by "Steven Spielberg" to 5.

```
update rating set rev_stars = 5 where mov_id in (select m.mov_id from movies m where m.dir_id in (select dir_id from director where dir_name="STEVEN SPIELBERG" ));
```

```
select * from rating;
```

The screenshot shows the MySQL Workbench interface. In the top-left, the Navigator pane displays the schema structure of the 'lab9' database, including tables like actor, director, movie_cast, movies, rating, and views. The top-right pane, titled 'SQLAdditions', contains a note about context help being disabled. The central area features a 'Query 1' editor with the following SQL code:

```

1 • use lab9;
2 • update rating set rev_stars = 5 where mov_id in (select m.mov_id from movies m where m.dir_id in (select dir_id from director where dir_name=
3 • select * from rating);

```

Below the editor is a 'Result Grid' showing the following data:

mov_id	rev_stars
1001	4
1002	2
1003	5
1004	5
1005	NULL

The bottom-right pane, titled 'rating 13', shows the 'Action Output' log with two entries:

#	Time	Action	Message	Duration / Fetch
58	12:37:48	update rating set rev_stars = 5 where mov_id in (select m.mov_id from movies m where m.dir_id in (select dir_id ...)	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.390 sec
59	12:37:48	select * from rating LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

Lab10:-

```

create table student (
    usn varchar(10),
    sname varchar(10),
    address varchar (30),
    phone varchar(10),
    gender varchar(1),
    constraint stu_usn primary key (usn)
);

```

```
create table semsec (
```

```
    ssid varchar(5),
    sem varchar(2),
```

```
sec varchar(1),
constraint sem_ssid primary key (ssid)
);

create table class(
usn varchar(10),
ssid varchar(5),
constraint class_usn primary key (usn),
constraint class_usn foreign key (usn) references student (usn) on delete cascade on update cascade,
constraint class_ssid foreign key (ssid) references semsec(ssid) on delete cascade on update cascade
);

create table subjects(
subcode varchar(10),
title varchar (20),
sem int,
credits int,
constraint sub_sub primary key (subcode)
);

create table iamarks(
usn varchar(10),
subcode varchar(10),
ssid varchar(5),
test1 int,
test2 int,
test3 int,
finalia int,
constraint ia primary key (usn,subcode,ssid),
```

constraint ia_usn foreign key (usn) references student(usn) on delete cascade on update cascade ,
 constraint ia_subcode foreign key (subcode) references subjects(subcode) on delete cascade on update cascade ,
 constraint ia_ssid foreign key (ssid) references semsec(ssid) on delete cascade on update cascade
);

```

INSERT INTO STUDENT VALUES ('1BM19CS020','AKSHAY','BELAGAVI', 8877881122,'M');

INSERT INTO STUDENT VALUES ('1BM19CS062','SANDHYA','BENGALURU', 7722829912,'F');

INSERT INTO STUDENT VALUES ('1BM19CS091','TEESHA','BENGALURU', 7712312312,'F');

INSERT INTO STUDENT VALUES ('1BM19CS066','SUPRIYA','MANGALURU', 8877881122,'F');

INSERT INTO STUDENT VALUES ('1BM20CS010','ABHAY','BENGALURU', 9900211201,'M');

INSERT INTO STUDENT VALUES ('1BM20CS032','BHASKAR','BENGALURU', 9923211099,'M');

INSERT INTO STUDENT VALUES ('1BM20CS025','ASMI','BENGALURU', 7894737377,'F');

INSERT INTO STUDENT VALUES ('1BM20CS011','AJAY','TUMKUR', 9845091941,'M');
  
```

MySQL Workbench - Local instance MySQL80

Query 1

```

1 • use lab10;
2 • select * from student;
  
```

Result Grid

usn	sname	address	phone	gender
1BM19CS020	AKSHAY	BELAGAVI	8877881122	M
1BM19CS062	SANDHYA	BENGALURU	7722829912	F
1BM19CS066	SUPRIYA	MANGALURU	8877881122	F
1BM19CS091	TEESHA	BENGALURU	7712312312	F
1BM20CS010	ABHAY	BENGALURU	9900211201	M
1BM20CS011	AJAY	TUMKUR	9845091941	M
1BM20CS025	ASMI	BENGALURU	7894737377	F
1BM20CS032	BHASKAR	BENGALURU	9923211099	M

student 4

Action Output

Time	Action	Message	Duration / Fetch
9 12:46:34	use lab10	0 row(s) affected	0.000 sec
10 12:46:34	select * from student LIMIT 0, 1000	8 row(s) returned	0.031 sec / 0.000 sec

```

INSERT INTO SEMSEC VALUES ('CSE4A', 4,'A');

INSERT INTO SEMSEC VALUES ('CSE4B', 4,'B');

INSERT INTO SEMSEC VALUES ('CSE4C', 4,'C');

INSERT INTO SEMSEC VALUES ('CSE6A', 6,'A');

INSERT INTO SEMSEC VALUES ('CSE6B', 6,'B');

INSERT INTO SEMSEC VALUES ('CSE6C', 6,'C');

INSERT INTO SEMSEC VALUES ('CSE8A', 8,'A');

INSERT INTO SEMSEC VALUES ('CSE8B', 8,'B');

INSERT INTO SEMSEC VALUES ('CSE8C', 8,'C');

```

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (selected), Filter objects, lab10 (selected), Tables (semsec is selected), Views, Stored Procedures, Functions.
- Query Editor:** Query 1 (use lab10; select * from semsec;).
- Result Grid:** Shows the data from the semsec table.
- Output Tab:** Action Output (5 rows affected, 9 rows returned), Message (0 row(s) affected, 9 row(s) returned), Duration / Fetch (0.000 sec, 0.109 sec / 0.000 sec).

semsec	sem	sec
CSE4A	4	A
CSE4B	4	B
CSE4C	4	C
CSE6A	6	A
CSE6B	6	B
CSE6C	6	C
CSE8A	8	A
CSE8B	8	B
CSE8C	8	C

```

INSERT INTO CLASS VALUES ('1BM19CS020','CSE8A');

INSERT INTO CLASS VALUES ('1BM19CS062','CSE4A');

INSERT INTO CLASS VALUES ('1BM19CS066','CSE8B');

INSERT INTO CLASS VALUES ('1BM19CS091','CSE8C');

INSERT INTO CLASS VALUES ('1BM20CS010','CSE4A');

INSERT INTO CLASS VALUES ('1BM20CS025','CSE4B');

```

INSERT INTO CLASS VALUES ('1BM20CS032','CSE4C');

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL80, Schemas (lab1, lab10, lab2, lab3, lab4, lab5, lab6, lab7, lab8, lab9, lti, sakila).
- Query Editor:** Query 1 contains the SQL statement:

```
1 • use lab10;
2 • select * from class;
```
- Result Grid:** Shows the results of the SELECT query:

user	ssid
1BM19CS062	CSE4A
1BM20CS010	CSE4A
1BM20CS025	CSE4B
1BM20CS032	CSE4C
1BM19CS020	CSE8A
1BM19CS066	CSE8B
1BM19CS091	CSE8C
NULL	NULL
- Action Output:** Shows the history of actions:

#	Time	Action	Message	Duration / Fetch
1	12:44:55	use lab10	0 row(s) affected	0.000 sec
2	12:44:55	select * from class LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

INSERT INTO SUBJECTS VALUES ('10CS81','ACA', 8, 4);

INSERT INTO SUBJECTS VALUES ('10CS82','SSM', 8, 4);

INSERT INTO SUBJECTS VALUES ('10CS83','NM', 8, 4);

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL80, Schemas (lab1, lab10, lab2, lab3, lab4, lab5, lab6, lab7, lab8, lab9, lti, sakila).
- Query Editor:** Query 1 contains the SQL statement:

```
1 • use lab10;
2 • select * from subjects;
```
- Result Grid:** Shows the results of the SELECT query:

subcode	title	sem	credits
10CS81	ACA	8	4
10CS82	SSM	8	4
10CS83	NM	8	4
NULL	NULL	NULL	NULL
- Action Output:** Shows the history of actions:

#	Time	Action	Message	Duration / Fetch
11	12:47:11	use lab10	0 row(s) affected	0.000 sec
12	12:47:11	select * from subjects LIMIT 0, 1000	3 row(s) returned	0.032 sec / 0.000 sec

```
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES  
('1BM19CS091','10CS81','CSE8C', 15, 16, 18);
```

```
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES  
('1BM19CS091','10CS82','CSE8C', 12, 19, 14);
```

```
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES  
('1BM19CS091','10CS83','CSE8C', 19, 15, 20);
```

The screenshot shows the MySQL Workbench interface with three tabs open:

- Query 1:** Contains the SQL code for inserting data into the IAMARKS table. The result grid shows three rows of data inserted.
- Result Grid:** Displays the data inserted into the IAMARKS table. The columns are usn, subcode, ssid, test1, test2, test3, and finala. The data is as follows:

usn	subcode	ssid	test1	test2	test3	finala
1BM19CS091	10CS81	CSE8C	15	16	18	HULL
1BM19CS091	10CS82	CSE8C	12	19	14	HULL
1BM19CS091	10CS83	CSE8C	19	15	20	HULL
HULL	HULL	HULL	HULL	HULL	HULL	HULL

- Output:** Shows the action history and message log. It includes two entries: "use lab10" and "select * from iamarks LIMIT 0, 1000". The message log indicates 0 row(s) affected and 3 row(s) returned.

-- i. List all the student details studying in fourth semester 'C' section.

```
select * from student s where s.usn in (select c.usn from class c where c ssid in (select ssid from semsec where sem='4' and sec='C'));
```

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator: Local instance MySQL80
SCHEMAS: Filter objects
145 lab1 lab10
Tables: class lamarks semsec student subjects
Views: 
Stored Procedures: 
Functions: 
lab2 lab3 lab4 lab5 lab6 lab7 lab8 lab9 lt1 sakila
Administration Schemas Information
No object selected

Query 1:
1 • use lab10;
2 • select * from student s where s.usn in (select c.usn from class c where c.ssid in (select ssid from semsec where sem='A' and sec='C'));

Result Grid | Filter Rows: | Edits: | Export/Imports: | Wrap Cell Content: 
usn sname address phone gender
1BM20CS032 BHASKAR BENGALURU 9923211099 M
NULL NULL NULL NULL NULL

Output:
Action Output
# Time Action Message Duration / Fetch
13 12:48:08 use lab10 0 row(s) affected 0.000 sec
14 12:48:08 select * from student s where s.usn in (select c.usn from class c where c.ssid in (select ssid from semsec where ... 1 row(s) returned 0.016 sec / 0.000 sec

Object Info Session

```

-- ii. Compute the total number of male and female students in each semester and in each section

select se.sem,se.sec,gender,count(gender) as 'number of students' from class c natural join student s natural join semsec se group by sem,sec,gender;

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator: Local instance MySQL80
SCHEMAS: Filter objects
145 lab1 lab10
Tables: class lamarks semsec student subjects
Views: 
Stored Procedures: 
Functions: 
lab2 lab3 lab4 lab5 lab6 lab7 lab8 lab9 lt1 sakila
Administration Schemas Information
No object selected

Query 1:
1 • use lab10;
2 • select se.sem,se.sec,gender,count(gender) as 'number of students' from class c natural join student s natural join semsec se group by sem,sec;
3 |

Result Grid | Filter Rows: | Export: | Wrap Cell Content: 
sem sec gender number of students
4 A F 1
4 A M 1
4 B F 1
4 C M 1
8 A M 1
8 B F 1
8 C F 1

Output:
Action Output
# Time Action Message Duration / Fetch
15 12:48:38 use lab10 0 row(s) affected 0.000 sec
16 12:48:39 select se.sem,se.sec,gender,count(gender) as 'number of students' from class c natural join student s natural jo... 7 row(s) returned 0.015 sec / 0.000 sec

Object Info Session

```

-- iii. Create a view of Test1 marks of student USN '1BI15CS101' in all subjects.

```
create view marks(test1) as select i.test1 from iamarks i where i.usn='1BM19CS091';
```

The screenshot shows the MySQL Workbench interface. In the left sidebar, under the 'Schemas' section, the 'lab10' schema is selected. Inside 'lab10', there is a 'Views' folder containing a single view named 'test1'. The main query editor window at the top has the SQL command: `SELECT * FROM lab10.marks;`. Below the query editor is a 'Result Grid' showing the data from the 'test1' view, which contains three rows with values 15, 12, and 19. The bottom right corner of the interface displays a message: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.'

```
-- iv.calculate finalia marks and update the respective column for all student;
```

```
select ia.usn,(case when test1>test2 and test2>test3 then (test1+test2)/2
```

```
when test2>test3 and test3>test1 then (test2+test3)/2
```

```
else (test1+test3)/2 end) as finalia from iamarks ia;
```

```
update iamarks set finalia= case finalia when test1>test2 and test2>test3 then (test1+test2)/2
```

```
when test2>test3 and test3>test1 then (test2+test3)/2
```

```
else (test1+test3)/2 end;
```

The screenshot shows the MySQL Workbench interface. In the Navigator pane, under the Schemas section, the 'lab10' schema is selected. The 'Tables' section contains 'class', 'iamarks', 'semsec', 'student', and 'subjects'. The 'Views' section contains 'marks'. The 'Stored Procedures' and 'Functions' sections are empty. In the Query Editor pane, there are two statements:

```

1 • use lab10;
2 • select * from iamarks;

```

The Result Grid displays the following data from the 'iamarks' table:

usn	subcode	ssid	test1	test2	test3	finalia
IBM19CS091	10CS81	CSEBC	15	16	18	17
IBM19CS091	10CS82	CSEBC	12	19	14	13
IBM19CS091	10CS83	CSEBC	19	15	20	20
			NULL	NULL	NULL	NULL

In the Output pane, the execution history is shown:

#	Time	Action	Message	Duration / Fetch
3	12:53:55	use lab10	0 row(s) affected	0.000 sec
4	12:53:55	select * from iamarks LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

-- v. 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.

```
select s.usn,s.sname,s.address,s.phone,s.gender,(case when ia.finalia between 17 and 20 then
'outstanding'when ia.finalia between 12 and 16 then 'average'else 'weak' end) as cat from iamarks
ia,student s where s.usn=ia.usn and ia ssid in (select ss ssid from semsec ss where sem='8');
```

MySQL Workbench

Local instance MySQL8.0

File Edit View Query Database Server Tools Scripting Help

Schemas

145

lab1

lab10

- Tables
- class
- iamarks
- semsec
- student
- subjects

VIEWS

- marks

Stored Procedures

Functions

lab2

lab3

lab4

lab5

lab6

lab7

lab8

lab9

lt1

No object selected

Administration Schemas Information

Query 1

use lab10;

select s.usn,s.sname,s.address,s.phone,s.gender,(case when ia.finalia between 17 and 20 then 'outstanding' when ia.finalia between 12 and 16

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

usn	sname	address	phone	gender	cat
1BM19C5091	TEESHA	BENGALURU	7712312312	F	outstanding
1BM19C5091	TEESHA	BENGALURU	7712312312	F	average
1BM19C5091	TEESHA	BENGALURU	7712312312	F	outstanding

Form Editor

Field Types

Result 2

Action Output

#	Time	Action	Message	Duration / Fetch
5	12:55:08	use lab10	0 row(s) affected	0.000 sec
6	12:55:08	select s.usn,s.sname,s.address,s.phone,s.gender,(case when ia.finalia between 17 and 20 then 'outstanding' ...	3 row(s) returned	0.000 sec / 0.000 sec

Context Help Snippets

Object Info Session