

DBMS LAB TEST 1 :-

CODE :-

```
CREATE TABLE pizza(  
    pid INT,  
    pname VARCHAR(20),  
    quality INT,  
    address VARCHAR (20),  
    PRIMARY KEY(pid)  
);
```

```
CREATE TABLE store(  
    sid INT,  
    sname VARCHAR(20),  
    phone INT,  
    qualityrating VARCHAR(20),  
    PRIMARY KEY(sid)  
);
```

```
CREATE TABLE soldby(  
    pid INT,  
    sid INT,  
    price FLOAT,  
    PRIMARY KEY(pid,sid),  
    FOREIGN KEY(pid) REFERENCES pizza(pid) ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY(sid) REFERENCES store(sid) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
INSERT INTO pizza VALUES('2', 'cheese','10','Varanasi');
```

```
INSERT INTO pizza VALUES('3', 'paneer','9','Patna');  
INSERT INTO pizza VALUES('4', 'sauce','8','Ballia');  
INSERT INTO pizza VALUES('5', 'large','7','Gujarat');  
INSERT INTO pizza VALUES('6', 'veggie','10','Ladakh');
```

```
INSERT INTO store VALUES('100','pizzahut','911','excellent');  
INSERT INTO store VALUES('101','dominoes','912','good');  
INSERT INTO store VALUES('102','gpizza','913','great');  
INSERT INTO store VALUES('103','hotpizza','914','excellent');  
INSERT INTO store VALUES('104','coolpizza','915','good');
```

```
INSERT INTO soldby VALUES('5','102','1500.99');  
INSERT INTO soldby VALUES('2','103','999.99');  
INSERT INTO soldby VALUES('3','100','50.15');  
INSERT INTO soldby VALUES('4','104','99.99');  
INSERT INTO soldby VALUES('6','101','20.32');
```

```
/*QUERY 2*/
```

```
SELECT * FROM pizza;
```

```
SELECT * FROM store;
```

```
SELECT * FROM soldby;
```

```
/*QUERY 3*/
```

```
SELECT DISTINCT s.sname FROM store s, soldby sb, pizza p WHERE sb.pid=p.pid AND sb.sid=s.sid AND sb.pid IN  
(SELECT p1.pid FROM pizza p1 WHERE p1.pname='veggie' AND p1.pname="cheese");
```

/*QUERY 4*/

Select s.sname,s.phone from store s, soldby sb where sb.sid=s.sid and sb.price<100 group by s.sname having s.qualityrating="good" or s.qualityrating="excellent";

/*QUERY 5*/

UPDATE soldby SET price=price *1.1 WHERE pid=(SELECT pid FROM pizza WHERE pname='chesse');

SELECT * FROM soldby;

/*QUERY 6*/

SELECT AVG(soldby.price) AS "Average Price",
store.sname AS "store"
FROM store, soldby
WHERE store.sid= soldby.sid
GROUP BY store.sname
HAVING AVG(soldby.price) >(SELECT AVG(price) FROM soldby);

/*QUERY 7*/

CREATE VIEW soldpizza (Pname, Totalpice, Avgprice) AS SELECT p1.pname, sum (sb.price), avg(sb.price) FROM pizza p1, soldby sb where sb.pid=p1.pid GROUP BY p.pname;

OUTPUT :-

The image displays two screenshots of the MySQL Workbench interface, showing the execution of SQL scripts and the resulting output.

Top Screenshot: The SQL script in the editor includes the following statements:

```
1 * CREATE TABLE pizzas
2   pid INT,
3   price VARCHAR(10),
4   quality INT,
5   address VARCHAR(100),
6   PRIMARY KEY(pid)
7   )
8
9 * CREATE TABLE stores
10  sid INT,
11  name VARCHAR(10),
12  phone INT,
13  qualityrating VARCHAR(10),
14  PRIMARY KEY(sid)
15  )
16
17 * CREATE TABLE supply
18  pid INT,
19  sid INT,
```

The Action Output pane shows the following results:

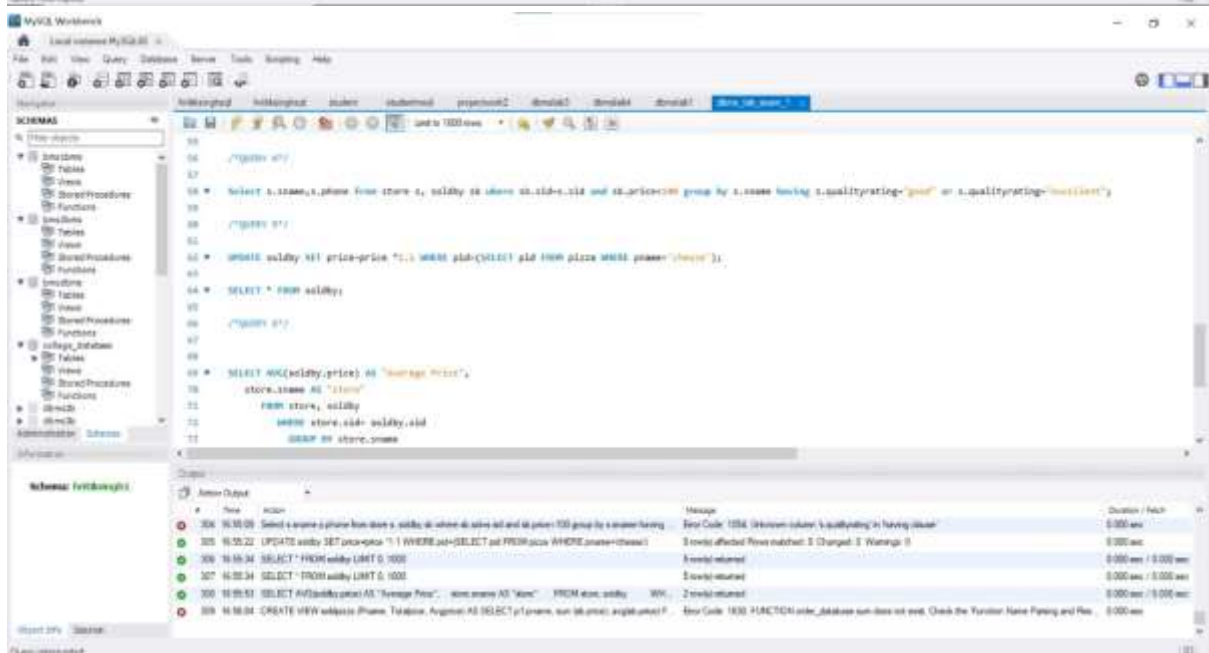
#	Time	Query	Message	Duration / Rows
1	16:55:08	SELECT * FROM pizzas	Error Code: 1054 Unknown column 'x' in field list	0:00 sec
2	16:55:22	UPDATE supply SET price=1.1 WHERE pid=SELECT pid FROM pizzas WHERE (name='Store')	3 rows affected Rows matched: 3 Changed: 3 Warnings: 0	0:00 sec
3	16:55:34	SELECT * FROM supply LIMIT 5, 1000	3 rows returned	0:00 sec / 0:00 sec
4	16:55:34	SELECT * FROM supply LIMIT 5, 1000	3 rows returned	0:00 sec / 0:00 sec
5	16:55:53	SELECT AVG(supply.price) AS 'Average Price', max(name) AS 'name' FROM supply	2 rows returned	0:00 sec / 0:00 sec
6	16:56:04	CREATE VIEW address Phone, Telephone, Address AS SELECT p.pid, s.sid, p.price, s.phone, s.name	Error Code: 1053 FUNCTION name/Address can't be used. Check the Function Name Paring and P...	0:00 sec

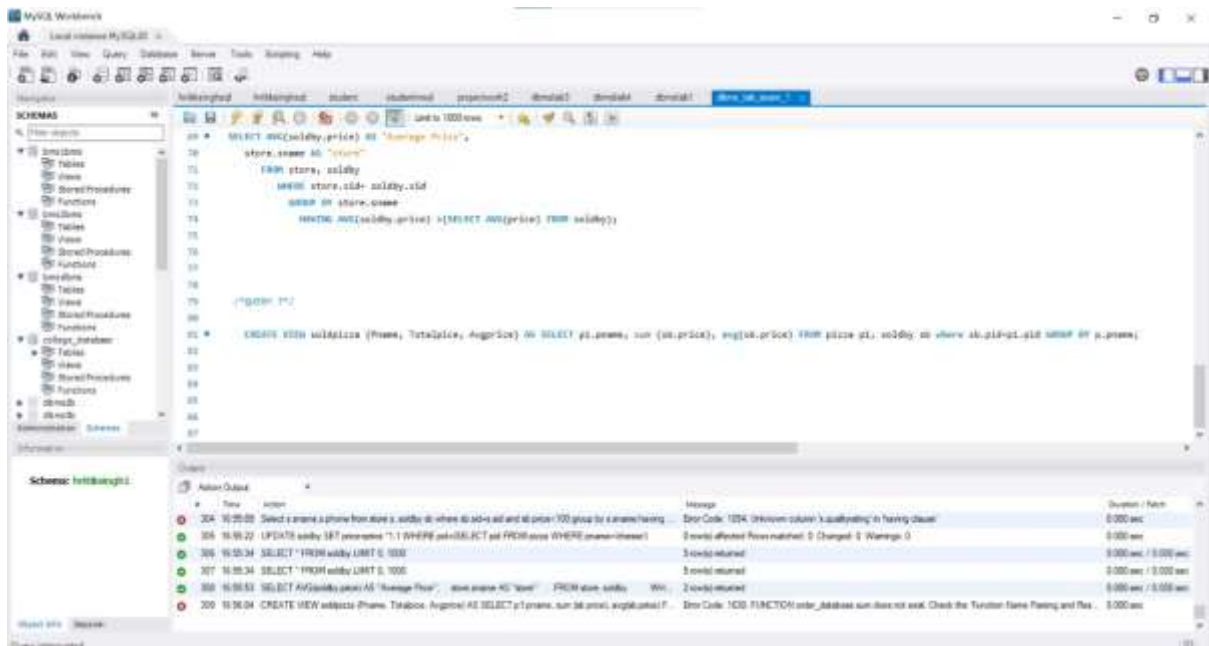
Bottom Screenshot: The SQL script in the editor includes the following statements:

```
19  sid INT,
20  price FLOAT,
21  PRIMARY KEY(pid,sid),
22  FOREIGN KEY(pid) REFERENCES pizzas(pid) ON DELETE CASCADE ON UPDATE CASCADE,
23  FOREIGN KEY(sid) REFERENCES stores(sid) ON DELETE CASCADE ON UPDATE CASCADE
24  )
25
26 * INSERT INTO pizzas VALUES('1','cheese','10','New York')
27 * INSERT INTO pizzas VALUES('2','pepperoni','12','New York')
28 * INSERT INTO pizzas VALUES('3','sausage','15','New York')
29 * INSERT INTO pizzas VALUES('4','large','18','New York')
30 * INSERT INTO pizzas VALUES('5','hawaii','10','New York')
31
32 * INSERT INTO stores VALUES('100','pizza hut','10','New York')
33 * INSERT INTO stores VALUES('101','dominos','12','New York')
34 * INSERT INTO stores VALUES('102','garden','15','New York')
35 * INSERT INTO stores VALUES('103','hempstone','18','New York')
36 * INSERT INTO stores VALUES('104','california','10','New York')
37
```

The Action Output pane shows the following results:

#	Time	Query	Message	Duration / Rows
1	16:55:08	SELECT * FROM pizzas	Error Code: 1054 Unknown column 'x' in field list	0:00 sec
2	16:55:22	UPDATE supply SET price=1.1 WHERE pid=SELECT pid FROM pizzas WHERE (name='Store')	3 rows affected Rows matched: 3 Changed: 3 Warnings: 0	0:00 sec
3	16:55:34	SELECT * FROM supply LIMIT 5, 1000	3 rows returned	0:00 sec / 0:00 sec
4	16:55:34	SELECT * FROM supply LIMIT 5, 1000	3 rows returned	0:00 sec / 0:00 sec
5	16:55:53	SELECT AVG(supply.price) AS 'Average Price', max(name) AS 'name' FROM supply	2 rows returned	0:00 sec / 0:00 sec
6	16:56:04	CREATE VIEW address Phone, Telephone, Address AS SELECT p.pid, s.sid, p.price, s.phone, s.name	Error Code: 1053 FUNCTION name/Address can't be used. Check the Function Name Paring and P...	0:00 sec





QUERY 2 TABLES :-

	pid	pname	quality	address
▶	2	cheese	10	Varanasi
	3	paneer	9	Patna
	4	sauce	8	Ballia
	5	large	7	Gujarat
	6	veggie	10	Ladakh
•	NULL	NULL	NULL	NULL

	sid	sname	phone	qualityrating
▶	100	pizzahut	911	excellent
	101	dominoes	912	good
	102	gpizza	913	great
	103	hotpizza	914	excellent
	104	coolpizza	915	good
•	NULL	NULL	NULL	NULL

	pid	sid	price
▶	2	103	999.99
	3	100	50.15
	4	104	99.99
	5	102	1500.99
	6	101	20.32
•	NULL	NULL	NULL

QUERY 3 :-

Result Grid

Filter Rows

Export

Wrap Cell Contents

name

Result Grid

View

Print

Print Range

Result 123

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
308	16:55:53	SELECT AVG(soldby.price) AS "Average Price" FROM store, soldby WHERE store.name AS "store"	2 row(s) returned	0.000 sec / 0.000 sec
309	16:56:04	CREATE VIEW soldpizza (Pname, Totalprice, Avgprice) AS SELECT p1.pname, sum (s1.price), avg(s1.price) F...	Error Code: 1030. FUNCTION order_database.sum does not exist. Check the Function Name Parsing and Res...	0.000 sec
310	17:02:30	SELECT * FROM pizza LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
311	17:02:47	SELECT * FROM store LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
312	17:03:01	SELECT * FROM soldby LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
313	17:03:17	SELECT DISTINCT s.name FROM store s, soldby sb, pizza p WHERE sb.pid=p.pid AND sb.name=s.name AND sb...	0 row(s) returned	0.000 sec / 0.000 sec

QUERY 4 :-

```

56 /*QUERY 4*/
57
58 * select s.name,s.phone from store s, soldby sb where sb.sname=s.sid and sb.price<100 group by s.name having s.qualityrating="good" or s.qualityrating="excellent";
59
60 /*QUERY 5*/
61
62 * UPDATE soldby SET price=price *1.1 WHERE pid=(SELECT pid FROM pizza WHERE pname='cheese');
63
64 * SELECT * FROM soldby;
65
66 /*QUERY 6*/

```

Output

Action	Time	Action	Message	Duration / Fetch
315	17:03:51	SELECT * FROM soldby LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
316	17:04:11	SELECT AVG(soldby.price) AS "Average Price" FROM store, soldby WHERE	2 row(s) returned	0.000 sec / 0.000 sec
317	17:04:30	CREATE VIEW soldpizza (Pname, Totalprice, Avgprice) AS SELECT p1.pname, sum (s1.price), avg(s1.price) F...	Error Code: 1030. FUNCTION order_database.sum does not exist. Check the Function Name Parsing and Res...	0.000 sec
318	17:05:05	CREATE VIEW soldpizza (Pname, Totalprice, Avgprice) AS SELECT p1.pname, sum (s1.price), avg(s1.price) F...	Error Code: 1030. FUNCTION order_database.sum does not exist. Check the Function Name Parsing and Res...	0.000 sec
319	17:05:08	CREATE VIEW soldpizza (Pname, Totalprice, Avgprice) AS SELECT p1.pname, sum (s1.price), avg(s1.price) F...	Error Code: 1030. FUNCTION order_database.sum does not exist. Check the Function Name Parsing and Res...	0.000 sec
320	17:09:56	Select s.name,s.phone from store s, soldby sb where sb.sname=sid and sb.price<100 group by s.name having	Error Code: 1054. Unknown column 's.qualityrating' in 'having clause'	0.000 sec

QUERY 5 :-

62	•	UPDATE soldby SET price=price *0.1 WHERE pid=(SELECT pid FROM pizza WHERE pname='cheese');
63		
64	•	SELECT * FROM soldby;
65		

pid	sid	price
2	202	999.99
3	200	90.15
4	204	99.99
5	202	1900.99
6	201	20.32

#	Time	Action	Message	Duration / Pgm
310	17:02:30	SELECT * FROM pizza LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
311	17:02:47	SELECT * FROM store LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
312	17:03:01	SELECT * FROM soldby LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
313	17:03:17	SELECT DISTINCT s.pname FROM store s, soldby sb, pizza p WHERE sb.pid=p.pid AND sb.sid=s.sid AND sb...	5 row(s) returned	0.000 sec / 0.000 sec
314	17:03:43	UPDATE soldby SET price=price *0.1 WHERE pid=(SELECT pid FROM pizza WHERE pname='cheese')	5 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
315	17:03:51	SELECT * FROM soldby LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

QUERY 6 :-

	Average Price	store
▶	1500.9899990234375	gpizza
	999.9899990234375	hotpizza

QUERY 7 :-

```

76
77
78
79  /*QUERY 7*/
80
81 • CREATE VIEW soldpizza (Pname, Totalprice, Avgprice) AS SELECT p.pname, sum(sb.price), avg(sb.price) FROM pizza p, soldby sb where sb.pid=p.pid GROUP BY p.pname;
82
83
84
85
86

```