

## **PRACTICAL 5**

### **PREREQUISITE STEPS**

**Step 1 :** Use tee PATH\TO\FOLDER\FILENAME.txt in MySQL CLI to start saving all commands and outputs of the current session into a file until stopped with note or else session is closed.

```
[24012011035_ved_joshi [(none)] Mon Oct 13 23:17:15 2025 >tee /tmp/PRACTICAL_5_LOG.txt;
Logging to file '/tmp/PRACTICAL_5_LOG.txt'
```

- **SYNTAX : tee PATH\TO\FOLDER\FILENAME.txt**
- tee in MySQL Command-Line Client enables logging of all commands and their results into a file during the same session. From this point onward, all entered commands and their outputs will be written to the specified file.
- This is useful for: Record keeping, Creating reports of SQL execution, Debugging query results later. The log continues until you turn it off with “notee” Command

**Step 2 : Changing the MySQL Prompt**

```
PROMPT set to '24012011035_ved_joshi [\d] \D \n>'
24012011035_ved_joshi [(none)] Mon Oct 13 23:12:44 2025 >
```

PROMPT changes how the MySQL CLI prompt looks.

Components:

- FULLENROLLMENT\_FULLNAME → Custom label or project name
- [\d] → Shows the current database in square brackets
- \D → Shows the current date
- \n → Inserts a newline before
- > → The actual command prompt symbol

**Step 3 : Either Create a new Database or Use Existing Database**

```
[24012011035_ved_joshi [(none)] Mon Oct 13 23:17:49 2025 >create database practical_5;
Query OK, 1 row affected (0.013 sec)
```

**Step 4 : Select a Existing Database**

```
[24012011035_ved_joshi [(none)] Mon Oct 13 23:23:01 2025 >use practical_5;
Database changed
[24012011035_ved_joshi [practical_5] Mon Oct 13 23:25:27 2025 >set sql_safe_updates = 0;
Query OK, 0 rows affected (0.002 sec)
```

- Purpose:** Disables safe update mode in MySQL for the current session.
  - Safe Mode Function:** Normally blocks UPDATE or DELETE without WHERE, indexed column, or LIMIT.
  - Effect of Disabling:** Allows unrestricted updates or deletions, even on all rows.
  - Common Use:** Bulk updates/deletions without strict conditions.
  - Scope:** Applies only to the current session; resets after disconnecting.
  - Caution:** Increases risk of accidental data loss—use carefully.
- Re-enable:** Use SET SQL\_SAFE\_UPDATES = 1; when finished.

**AIM:** Study the concepts of different operators, aggregate functions and various joins.

Create TABLE: **salesmen**

column name	data type	size	attributes
snum	varchar2	6	primary key/first letter must start with 's'
sname	varchar2	20	not null
city	varchar2	15	
comm	number	5,2	

```
24012011035_ved_joshi [practical_5] Mon Oct 13 23:29:37 2025 >CREATE TABLE salesmen (
    ->     snum VARCHAR(6) PRIMARY KEY,
    ->     sname VARCHAR(20) NOT NULL,
    ->     city VARCHAR(15),
    ->     comm DECIMAL(5,2)
    -> )
    -> ;
Query OK, 0 rows affected (0.018 sec)
```

```
[24012011035_ved_joshi [practical_5] Mon Oct 13 23:31:40 2025 >desc salesmen;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| snum  | varchar(6) | NO   | PRI | NULL    |       |
| sname | varchar(20) | NO   |     | NULL    |       |
| city  | varchar(15) | YES  |     | NULL    |       |
| comm  | decimal(5,2)| YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.009 sec)
```

Insert following records in table salesmen:

snum	sname	city	comm
s1001	Piyush	London	0.12
s1002	Niraj	san jose	0.13
s1003	Miti	London	0.11
s1004	Rajesh	Barcelona	0.15

s1005	Haresh	new York	0.10
s1006	Ram	Bombay	0.10
s1007	Nehal	Delhi	0.09

```
[24012011035_ved_joshi [practical_5] Mon Oct 13 23:35:07 2025 >INSERT INTO salesmen (snum, sname, city, comm)
-> VALUES
-> ('s1001', 'Piyush', 'London', 0.12),
-> ('s1002', 'Niraj', 'san jose', 0.13),
-> ('s1003', 'Miti', 'London', 0.11),
-> ('s1004', 'Rajesh', 'Barcelona', 0.15),
-> ('s1005', 'Haresh', 'new York', 0.10),
-> ('s1006', 'Ram', 'Bombay', 0.10),
[ -> ('s1007', 'Nehal', 'Delhi', 0.09);
Query OK, 7 rows affected (0.014 sec)
Records: 7  Duplicates: 0  Warnings: 0

[24012011035_ved_joshi [practical_5] Mon Oct 13 23:39:25 2025 >select * from salesmen;
+-----+-----+-----+-----+
| snum | sname | city   | comm  |
+-----+-----+-----+-----+
| s1001 | Piyush | London | 0.12 |
| s1002 | Niraj  | san jose | 0.13 |
| s1003 | Miti   | London | 0.11 |
| s1004 | Rajesh | Barcelona | 0.15 |
| s1005 | Haresh | new York | 0.10 |
| s1006 | Ram    | Bombay | 0.10 |
| s1007 | Nehal  | Delhi  | 0.09 |
+-----+-----+-----+-----+
7 rows in set (0.001 sec)
```

Create TABLE: **customer**

column name	data type	size	attributes
cnum	varchar2	6	primary key/first letter must start with 'c'
cname	varchar2	20	not null
city	varchar2	15	
rating	number	5	
snum	varchar2	6	

```
[24012011035_ved_joshi [practical_5] Mon Oct 13 23:44:35 2025 >CREATE TABLE customer (
->      cnum VARCHAR(6) PRIMARY KEY,
->      cname VARCHAR(20) NOT NULL,
->      city VARCHAR(15),
->      rating INT(5),
->      snum VARCHAR(6)
[ -> );
Query OK, 0 rows affected, 1 warning (0.010 sec)
```

```
[24012011035_ved_joshi [practical_5] Mon Oct 13 23:45:15 2025 >desc customer;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| cnum  | varchar(6) | NO   | PRI | NULL    |       |
| cname | varchar(20) | NO   |     | NULL    |       |
| city  | varchar(15) | YES  |     | NULL    |       |
| rating | int        | YES  |     | NULL    |       |
| snum  | varchar(6) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
5 rows in set (0.003 sec)
```

Insert following records in table customer:

cnum	cname	city	rating	snum
c2001	hardik	london	100	s1001
c2002	geeta	rome	200	s1003
c2003	kavish	san jose	200	s1002
c2004	dhruv	berlin	300	s1002
c2005	pratham	london	100	s1001
c2006	vyomesh	san jose	300	s1007
c2007	kirit	rome	100	s1004
c2008	agam		200	s1003
c2009	falgun	san jose		s1001

```
24012011035_ved_joshi [practical_5] Mon Oct 13 23:49:48 2025 >INSERT INTO customer (cnum, cname, city, rating, snum)
-> VALUES
-> ('c2001', 'hardik', 'london', 100, 's1001'),
-> ('c2002', 'geeta', 'rome', 200, 's1003'),
-> ('c2003', 'kavish', 'san jose', 200, 's1002'),
-> ('c2004', 'dhruv', 'berlin', 300, 's1002'),
-> ('c2005', 'pratham', 'london', 100, 's1001'),
-> ('c2006', 'vyomesh', 'san jose', 300, 's1007'),
-> ('c2007', 'kirit', 'rome', 100, 's1004'),
-> ('c2008', 'agam', NULL, 200, 's1003'),
-> ('c2009', 'falgun', 'san jose', NULL, 's1001');
Query OK, 9 rows affected (0.004 sec)
Records: 9  Duplicates: 0  Warnings: 0

[24012011035_ved_joshi [practical_5] Mon Oct 13 23:50:33 2025 >select * from customer;
+-----+-----+-----+-----+-----+
| cnum | cname | city | rating | snum |
+-----+-----+-----+-----+-----+
| c2001 | hardik | london | 100 | s1001 |
| c2002 | geeta | rome | 200 | s1003 |
| c2003 | kavish | san jose | 200 | s1002 |
| c2004 | dhruv | berlin | 300 | s1002 |
| c2005 | pratham | london | 100 | s1001 |
| c2006 | vyomesh | san jose | 300 | s1007 |
| c2007 | kirit | rome | 100 | s1004 |
| c2008 | agam | NULL | 200 | s1003 |
| c2009 | falgun | san jose | NULL | s1001 |
+-----+-----+-----+-----+-----+
9 rows in set (0.001 sec)
```

Create TABLE: **order\_info**

column name	data type	size	attributes
onum	varchar2	6	primary key/first letter must start with 'o'
amt	number	10,2	not null
odate	date		
cnum	varchar2	6	
snum	varchar2	6	

```

24012011035_ved_joshi [practical_5] Mon Oct 13 23:51:12 2025 >CREATE TABLE order_info (
->     onum VARCHAR(6) PRIMARY KEY,
->     amt DECIMAL(10,2) NOT NULL,
->     odate DATE,
->     cnum VARCHAR(6),
->     snum VARCHAR(6)
-> );
Query OK, 0 rows affected (0.007 sec)

[24012011035_ved_joshi [practical_5] Mon Oct 13 23:53:18 2025 >
[24012011035_ved_joshi [practical_5] Mon Oct 13 23:53:19 2025 >desc order_info;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| onum  | varchar(6) | NO   | PRI | NULL    |          |
| amt   | decimal(10,2)| NO  |     | NULL    |          |
| odate | date       | YES  |     | NULL    |          |
| cnum  | varchar(6) | YES  |     | NULL    |          |
| snum  | varchar(6) | YES  |     | NULL    |          |
+-----+-----+-----+-----+-----+
5 rows in set (0.003 sec)

```

Insert following records in table order\_info:

<b>onum</b>	<b>amt</b>	<b>odate</b>	<b>cnum</b>	<b>snum</b>
o3001	18.69	10-mar-20	c2008	s1007
o3003	767.19	10-mar-20	c2001	s1001
o3002	1900.10	03-oct-21	c2007	s1004
o3005	5160.45	04-oct-21	c2003	s1002
o3006	1098.16	10-mar-20	c2008	s1007
o3009	1713.23	10-april-20	c2002	s1003
o3007	75.75	10-april-20	c2004	s1002
o3008	4723.00	10-may-21	c2006	s1001
o3010	1309.95	10-may-21	c2004	s1002
o3011	9891.88	10-june-21	c2006	s1001

```

24012011035_ved_joshi [practical_5] Mon Oct 13 23:56:17 2025 >INSERT INTO order_info VALUES
-> ('o3001', 18.69, '2020-03-10', 'c2008', 's1007'),
-> ('o3003', 767.19, '2020-03-10', 'c2001', 's1001'),
-> ('o3002', 1900.10, '2021-10-03', 'c2007', 's1004'),
-> ('o3005', 5160.45, '2021-10-04', 'c2003', 's1002'),
-> ('o3006', 1098.16, '2020-03-10', 'c2008', 's1007'),
-> ('o3009', 1713.23, '2020-04-10', 'c2002', 's1003'),
-> ('o3007', 75.75, '2020-04-10', 'c2004', 's1002'),
-> ('o3008', 4723.00, '2021-05-10', 'c2006', 's1001'),
-> ('o3010', 1309.95, '2021-05-10', 'c2004', 's1002'),
[ -> ('o3011', 9891.88, '2021-06-10', 'c2006', 's1001');
Query OK, 10 rows affected (0.003 sec)
Records: 10  Duplicates: 0  Warnings: 0

[24012011035_ved_joshi [practical_5] Tue Oct 14 00:03:05 2025 >select * from order_info;
+-----+-----+-----+-----+
| onum | amt   | odate    | cnum   | snum   |
+-----+-----+-----+-----+
| o3001 | 18.69 | 2020-03-10 | c2008 | s1007 |
| o3002 | 1900.10 | 2021-10-03 | c2007 | s1004 |
| o3003 | 767.19 | 2020-03-10 | c2001 | s1001 |
| o3005 | 5160.45 | 2021-10-04 | c2003 | s1002 |
| o3006 | 1098.16 | 2020-03-10 | c2008 | s1007 |
| o3007 | 75.75 | 2020-04-10 | c2004 | s1002 |
| o3008 | 4723.00 | 2021-05-10 | c2006 | s1001 |
| o3009 | 1713.23 | 2020-04-10 | c2002 | s1003 |
| o3010 | 1309.95 | 2021-05-10 | c2004 | s1002 |
| o3011 | 9891.88 | 2021-06-10 | c2006 | s1001 |
+-----+-----+-----+-----+
10 rows in set (0.001 sec)

```

### Write SQL Queries for following questions:

#### Select:

- 1) Display commission values of without any repetition from Salesman Table.

```

24012011035_ved_joshi [practical_5] Tue Oct 14 00:03:46 2025 >SELECT DISTINCT comm
[ -> FROM salesmen;
+-----+
| comm |
+-----+
| 0.12 |
| 0.13 |
| 0.11 |
| 0.15 |
| 0.10 |
| 0.09 |
+-----+
6 rows in set (0.001 sec)

```

#### Special Operators:

- 2) Display all salesmen that were located in either BARCELONA or LONDON (use IN keyword).

```
24012011035_ved_joshi [practical_5] Tue Oct 14 00:06:08 2025 >SELECT *
    -> FROM salesmen
[   -> WHERE UPPER(city) IN ('BARCELONA', 'LONDON');
+-----+-----+-----+-----+
| snum | sname | city      | comm |
+-----+-----+-----+-----+
| s1001 | Piyush | London    | 0.12 |
| s1003 | Miti   | London    | 0.11 |
| s1004 | Rajesh | Barcelona | 0.15 |
+-----+-----+-----+
3 rows in set (0.003 sec)
```

- 3) Display all salesmen with commission between 0.10 and 0.12.

```
24012011035_ved_joshi [practical_5] Tue Oct 14 00:07:12 2025 >SELECT *
    -> FROM salesmen
[   -> WHERE comm BETWEEN 0.10 AND 0.12;
+-----+-----+-----+-----+
| snum | sname | city      | comm |
+-----+-----+-----+-----+
| s1001 | Piyush | London    | 0.12 |
| s1003 | Miti   | London    | 0.11 |
| s1005 | Haresh | new York | 0.10 |
| s1006 | Ram    | Bombay   | 0.10 |
+-----+-----+-----+
4 rows in set (0.002 sec)
```

### **LIKE Operator:**

- 4) List all the customers whose name's third latter is 'R'.

```
24012011035_ved_joshi [practical_5] Tue Oct 14 00:08:34 2025 >SELECT *
    -> FROM customer
[   -> WHERE cname LIKE '_ _r%';
+-----+-----+-----+-----+
| cnum | cname  | city     | rating | snum  |
+-----+-----+-----+-----+
| c2001 | hardik | london  | 100   | s1001 |
| c2004 | dhruv  | berlin  | 300   | s1002 |
| c2007 | kirit   | rome    | 100   | s1004 |
+-----+-----+-----+-----+
3 rows in set (0.003 sec)
```

- 5) List all salesmen whose name start with letter 'P' and end letter is 'h'.

```
24012011035_ved_joshi [practical_5] Tue Oct 14 00:09:28 2025 >SELECT *
-> FROM salesmen
[ -> WHERE sname LIKE 'P%h';
+-----+-----+-----+
| snum | sname | city   | comm  |
+-----+-----+-----+
| s1001 | Piyush | London | 0.12 |
+-----+-----+-----+
1 row in set (0.001 sec)
```

**NULL Operator:**

- 6) Find all records in customer table with NULL values in the city column.

```
24012011035_ved_joshi [practical_5] Tue Oct 14 00:10:39 2025 >SELECT *
-> FROM customer
[ -> WHERE city IS NULL;
+-----+-----+-----+-----+
| cnum | cname | city  | rating | snum  |
+-----+-----+-----+-----+
| c2008 | agam  | NULL  |     200 | s1003 |
+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

- 7) Write two queries that will produce all orders taken on October 3<sup>rd</sup> or 4<sup>th</sup>, 2020.

a) Use BETWEEN

```
24012011035_ved_joshi [practical_5] Tue Oct 14 00:11:40 2025 >SELECT *
-> FROM order_info
[ -> WHERE odate BETWEEN '03-oct-20' AND '04-oct-20';
Empty set, 20 warnings (0.003 sec)
```

b) Use OR

```
24012011035_ved_joshi [practical_5] Tue Oct 14 00:19:29 2025 >SELECT *
-> FROM order_info
[ -> WHERE odate = '2020-10-03' OR odate = '2020-10-04';
Empty set (0.002 sec)
```

- 8) Write a query that selects all orders without ZEROS or NULLS in amount field.

```
[24012011035_ved_joshi [practical_5] Tue Oct 14 00:20:01 2025 >SELECT * FROM order_info WHERE amt IS NOT NULL AND amt != 0;
+-----+-----+-----+-----+-----+
| onum | amt   | odate   | cnum   | snum   |
+-----+-----+-----+-----+-----+
| o3001 | 18.69 | 2020-03-10 | c2008 | s1007 |
| o3002 | 1900.10 | 2021-10-03 | c2007 | s1004 |
| o3003 | 767.19 | 2020-03-10 | c2001 | s1001 |
| o3005 | 5160.45 | 2021-10-04 | c2003 | s1002 |
| o3006 | 1098.16 | 2020-03-10 | c2008 | s1007 |
| o3007 | 75.75 | 2020-04-10 | c2004 | s1002 |
| o3008 | 4723.00 | 2021-05-10 | c2006 | s1001 |
| o3009 | 1713.23 | 2020-04-10 | c2002 | s1003 |
| o3010 | 1309.95 | 2021-05-10 | c2004 | s1002 |
| o3011 | 9891.88 | 2021-06-10 | c2006 | s1001 |
+-----+-----+-----+-----+
10 rows in set (0.001 sec)
```

**Functions:**

- 9) Count the numbers of salesmen without duplication from the order\_info table.

```
[24012011035_ved_joshi [practical_5] Tue Oct 14 00:21:19 2025 >SELECT COUNT(DISTINCT snum) FROM order_info;
+-----+
| COUNT(DISTINCT snum) |
+-----+
|      5 |
+-----+
1 row in set (0.005 sec)
```

- 10) Count the rating of customers (with NULL and without NULL).

With null:-

```
[24012011035_ved_joshi [practical_5] Tue Oct 14 00:23:47 2025 >SELECT COUNT(*) FROM customer;
+-----+
| COUNT(*) |
+-----+
|      9 |
+-----+
1 row in set (0.005 sec)
```

Without null:-

```
[24012011035_ved_joshi [practical_5] Tue Oct 14 00:24:53 2025 >SELECT COUNT(rating) FROM customer;
+-----+
| COUNT(rating) |
+-----+
|      8 |
+-----+
1 row in set (0.001 sec)
```

- 11) Find the largest order taken by each salesperson.

```
[24012011035_ved_joshi [practical_5] Tue Oct 14 00:27:39 2025 >SELECT snum, MAX(amt) as largest_order FROM order_info GROUP BY snum;
+-----+-----+
| snum | largest_order |
+-----+-----+
| s1007 |    1098.16 |
| s1004 |    1900.10 |
| s1001 |    9891.88 |
| s1002 |    5160.45 |
| s1003 |    1713.23 |
+-----+-----+
5 rows in set (0.003 sec)
```

- 12) Find the largest order taken by each salesperson on each date.

```
[24012011035_ved_joshi [practical_5] Tue Oct 14 00:28:29 2025 >SELECT snum, odate, MAX(amt) as largest_order FROM order_info GROUP BY snum, odate;
+-----+-----+-----+
| snum | odate      | largest_order |
+-----+-----+-----+
| s1007 | 2020-03-10 |    1098.16 |
| s1004 | 2021-10-03 |    1900.10 |
| s1001 | 2020-03-10 |     767.19 |
| s1002 | 2021-10-04 |    5160.45 |
| s1002 | 2020-04-10 |     75.75 |
| s1001 | 2021-05-10 |    4723.00 |
| s1003 | 2020-04-10 |    1713.23 |
| s1002 | 2021-05-10 |    1369.95 |
| s1001 | 2021-06-10 |    9891.88 |
+-----+-----+-----+
9 rows in set (0.001 sec)
```

- 13) Find total amount ordered by each customer, only display records whose total amount is greater than 800.

```
[24012011035_ved_joshi [practical_5] Tue Oct 14 00:30:01 2025 >SELECT cnum, SUM(amt) as total_amount FROM order_info GROUP BY cnum HAVING SUM(amt) > 800;
+-----+-----+
| cnum | total_amount |
+-----+-----+
| c2008 |    1116.85 |
| c2007 |    1900.10 |
| c2003 |    5160.45 |
| c2004 |    1385.70 |
| c2006 |    14614.88 |
| c2002 |    1713.23 |
+-----+-----+
6 rows in set (0.002 sec)
```

- 14) Write a query that counts the number of different non-NULL city in the customer table.

```
[24012011035_ved_joshi [practical_5] Tue Oct 14 00:31:04 2025 >SELECT COUNT(DISTINCT city) FROM customer WHERE city IS NOT NULL;
+-----+
| COUNT(DISTINCT city) |
+-----+
|                  4 |
+-----+
1 row in set (0.001 sec)
```

- 15) Display sname and commission from salesmen in descending order of salesman number.

```
[24012011035_ved_joshi [practical_5] Tue Oct 14 00:31:56 2025 >SELECT sname, comm FROM salesmen ORDER BY snum DESC;
+-----+-----+
| sname | comm  |
+-----+-----+
| Nehal | 0.09 |
| Ram   | 0.10 |
| Haresh| 0.10 |
| Rajesh| 0.15 |
| Miti  | 0.11 |
| Niraj | 0.13 |
| Piyush| 0.12 |
+-----+-----+
7 rows in set (0.001 sec)
```

- 16) Assume each salesperson has a 0.12 commission. Write a query on the order\_info table that will produce the order number, the salesperson number and the amount of the salesperson's commission for that order.

```
[24012011035_ved_joshi [practical_5] Tue Oct 14 00:32:51 2025 >SELECT onum, snum, amt * 0.12 as commission FROM order_info;
+-----+-----+-----+
| onum | snum | commission |
+-----+-----+-----+
| o3001 | s1007 |    2.2428 |
| o3002 | s1004 |    228.0120 |
| o3003 | s1001 |    92.0628 |
| o3005 | s1002 |    619.2540 |
| o3006 | s1007 |    131.7792 |
| o3007 | s1002 |     9.0900 |
| o3008 | s1001 |    566.7600 |
| o3009 | s1003 |    205.5876 |
| o3010 | s1002 |    157.1940 |
| o3011 | s1001 |    1187.0256 |
+-----+-----+-----+
10 rows in set (0.002 sec)
```

- 17) Write a query on the customers table that will find the highest rating in each city. Put the output in this form.

```
[24012011035_ved_joshi [practical_5] Tue Oct 14 00:33:38 2025 >SELECT CONCAT('For the city ', city, ', the highest rating is: ', MAX(rating)) as result
[   -> FROM customer WHERE city IS NOT NULL GROUP BY city;
+-----+
| result
+-----+
| For the city london, the highest rating is: 100
| For the city rome, the highest rating is: 200
| For the city san jose, the highest rating is: 300
| For the city berlin, the highest rating is: 300
+-----+
4 rows in set (0.003 sec)
```

*For the city (city column), the highest rating is: (rating column).*

- 18) Write a query that totals the orders for each day and places the results in descending order.

```
[24012011035_ved_joshi [practical_5] Tue Oct 14 00:34:40 2025 >SELECT odate, SUM(amt) as total_orders FROM order_info GROUP BY odate ORDER BY total_orders DESC;
+-----+
| odate    | total_orders |
+-----+
| 2021-06-10 |      9891.88 |
| 2021-05-10 |      6032.95 |
| 2021-10-04 |      5168.45 |
| 2021-10-03 |      1900.10 |
| 2020-03-10 |      1884.04 |
| 2020-04-10 |      1788.98 |
+-----+
6 rows in set (0.002 sec)
```

### JOIN Operation:

- 19) Show the names of all customers matched with the salesmen serving them.

```
[24012011035_ved_joshi [practical_5] Tue Oct 14 00:35:58 2025 >SELECT c.cname as customer_name, s.sname as salesman_name
[   -> FROM customer c JOIN salesmen s ON c.snum = s.snum;
+-----+
| customer_name | salesman_name |
+-----+
| hardik        | Piyush          |
| geeta         | Miti            |
| kavish        | Niraj           |
| dhruv         | Niraj           |
| pratham       | Piyush          |
| vyomesh       | Nehal           |
| kirit          | Rajesh          |
| agam          | Miti            |
| falgun         | Piyush          |
+-----+
9 rows in set (0.003 sec)
```

- 20) Write a query that lists each order number followed by customer name who made the order.

```
[24012011035_ved_joshi [practical_5] Tue Oct 14 00:36:59 2025 >SELECT o.onum, c.cname FROM order_info o JOIN customer c ON o.cnum = c.cnum;
+-----+
| onum | cname  |
+-----+
| 03001 | agam   |
| 03002 | kirit  |
| 03003 | hardik |
| 03005 | kavish |
| 03006 | agam   |
| 03007 | dhruv  |
| 03008 | vyomesh|
| 03009 | geeta  |
| 03010 | dhruv  |
| 03011 | vyomesh|
+-----+
10 rows in set (0.001 sec)
```

- 21) Write a query that gives the names of both the salesperson and the customer for each order after the order number.

```
24012011035_ved_joshi [practical_5] Tue Oct 14 00:37:50 2025 >SELECT o.onum, s.sname as salesman_name, c.cname as customer_name
  -> FROM order_info o
  -> JOIN salesmen s ON o.snum = s.snum
  -> JOIN customer c ON o.cnum = c.cnum;
+-----+-----+-----+
| onum | salesman_name | customer_name |
+-----+-----+-----+
| 03001 | Nehal        | agam          |
| 03002 | Rajesh       | kirit          |
| 03003 | Piyush        | hardik         |
| 03005 | Niraj        | kavish         |
| 03006 | Nehal        | agam          |
| 03007 | Niraj        | dhruv          |
| 03008 | Piyush        | vyomesh        |
| 03009 | Miti          | geeta          |
| 03010 | Niraj        | dhruv          |
| 03011 | Piyush        | vyomesh        |
+-----+-----+-----+
10 rows in set (0.001 sec)
```

- 22) Write a query that produces all customers serviced by salesmen with a commission above 0.12. Output shows the customer's name, the salesperson's name and the salesperson's rate of commission.

```
24012011035_ved_joshi [practical_5] Tue Oct 14 00:38:46 2025 >SELECT c.cname as customer_name, s.sname as salesman_name, s.comm
  -> FROM customer c JOIN salesmen s ON c.snum = s.snum WHERE s.comm > 0.12;
+-----+-----+-----+
| customer_name | salesman_name | comm |
+-----+-----+-----+
| kavish        | Niraj        | 0.13 |
| dhruv         | Niraj        | 0.13 |
| kirit          | Rajesh       | 0.15 |
+-----+-----+-----+
3 rows in set (0.002 sec)
```

- 23) Write a query that calculates the amount of the salesperson's commission on each order by a customer with a rating above 100.

```
24012011035_ved_joshi [practical_5] Tue Oct 14 00:40:32 2025 >SELECT o.onum, o.amt, s.comm, (o.amt * s.comm) as commission_amount
  -> FROM order_info o
  -> JOIN salesmen s ON o.snum = s.snum
  -> JOIN customer c ON o.cnum = c.cnum
  -> WHERE c.rating > 100;
+-----+-----+-----+-----+
| onum | amt   | comm | commission_amount |
+-----+-----+-----+-----+
| 03001 | 18.69 | 0.09 | 1.6821 |
| 03005 | 5160.45 | 0.13 | 670.8585 |
| 03006 | 1098.16 | 0.09 | 98.8344 |
| 03007 | 75.75 | 0.13 | 9.8475 |
| 03008 | 4723.00 | 0.12 | 566.7600 |
| 03009 | 1713.23 | 0.11 | 188.4553 |
| 03010 | 1309.95 | 0.13 | 170.2935 |
| 03011 | 9891.88 | 0.12 | 1187.0256 |
+-----+-----+-----+-----+
8 rows in set (0.001 sec)
```

## OTHERS:

- 24) Create a union of two queries that shows the names, cities and ratings of all customers. Those with rating of  $\geq 200$  should display 'HIGH RATING' and those with  $< 200$  should display 'LOW RATING'.

```
24012011035_ved_joshi [practical_5] Tue Oct 14 00:41:30 2025 >SELECT cname, city, rating, 'HIGH RATING' as rating_category
  -> FROM customer WHERE rating >= 200
  -> UNION
  -> SELECT cname, city, rating, 'LOW RATING' as rating_category
  -> FROM customer WHERE rating < 200 OR rating IS NULL;
+-----+-----+-----+-----+
| cname | city  | rating | rating_category |
+-----+-----+-----+-----+
| geeta | rome  | 200   | HIGH RATING   |
| kavish| san jose | 200   | HIGH RATING   |
| dhruv | berlin | 300   | HIGH RATING   |
| vyomesh| san jose | 300   | HIGH RATING   |
| agam  | NULL   | 200   | HIGH RATING   |
| hardik| london | 100   | LOW RATING    |
| pratham| london | 100   | LOW RATING    |
| kirit  | rome  | 100   | LOW RATING    |
| falgun | san jose | NULL  | LOW RATING    |
+-----+-----+-----+-----+
9 rows in set (0.009 sec)
```

- 25) Find all customers who made order on 10th March 2020 using sub-query.

```
24012011035_ved_joshi [practical_5] Tue Oct 14 00:42:28 2025 >SELECT * FROM customer
[   -> WHERE cnum IN (SELECT cnum FROM order_info WHERE odate = '2020-03-10');
+-----+-----+-----+-----+
| cnum | cname | city  | rating | snum |
+-----+-----+-----+-----+
| c2008 | agam  | NULL  |    200 | s1003 |
| c2001 | hardik | london |    100 | s1001 |
+-----+-----+-----+-----+
2 rows in set (0.003 sec)
```

- 26) Find all customers having rating greater than any customer in 'ROME'.

```
24012011035_ved_joshi [practical_5] Tue Oct 14 00:43:22 2025 >SELECT * FROM customer
[   -> WHERE rating > ANY (SELECT rating FROM customer WHERE city = 'rome');
+-----+-----+-----+-----+
| cnum | cname | city  | rating | snum |
+-----+-----+-----+-----+
| c2002 | geeta | rome  |    200 | s1003 |
| c2003 | kavish | san jose |    200 | s1002 |
| c2004 | dhruv  | berlin |    300 | s1002 |
| c2006 | vyomesh | san jose |    300 | s1007 |
| c2008 | agam  | NULL  |    200 | s1003 |
+-----+-----+-----+-----+
5 rows in set (0.002 sec)
```

- 27) Create another table London\_staff having same structure as salesmen table.

```
[24012011035_ved_joshi [practical_5] Tue Oct 14 00:44:15 2025 >CREATE TABLE London_staff AS SELECT * FROM salesmen WHERE 1=0;
Query OK, 0 rows affected (0.034 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

- 28) Delete all salesmen who have at least one customer with a rating of 100 from salesmen table.

```
24012011035_ved_joshi [practical_5] Tue Oct 14 00:45:15 2025 >DELETE FROM salesmen
[   -> WHERE snum IN (SELECT DISTINCT snum FROM customer WHERE rating = 100);
Query OK, 2 rows affected (0.004 sec)
```