# PRACTICAL 2

AIM: Perform & learn following concepts:

## 1. logical -And, Or, Not 1. Logical Operators: AND, OR, NOT

### ▶ AND – All conditions must be true

```
SELECT * FROM Employees
WHERE Department = 'HR' AND Salary > 50000;
```

**Explanation**: Fetch employees who are in HR **and** earn more than 50,000.

### ▶ OR – At least one condition must be true

```
SELECT * FROM Employees
WHERE Department = 'HR' OR Salary > 50000;
```

**Explanation**: Fetch employees who are in HR **or** earn more than 50,000.

### ▶ NOT – Negates a condition

```
SELECT * FROM Employees
WHERE NOT Department = 'HR';
```

**Explanation**: Fetch employees **not** in the HR department.

## 2. BETWEEN … AND | NOT BETWEEN … AND -

### ▶ BETWEEN

```
SELECT * FROM Products
WHERE Price BETWEEN 100 AND 500;
```

**Explanation**: Fetch products with price **from 100 to 500** (inclusive).

### ▶ NOT BETWEEN

```
SELECT * FROM Products
WHERE Price NOT BETWEEN 100 AND 500;
```

**Explanation**: Fetch products with price **less than 100 or more than 500**.

## 3. LIKE Predicate

**Wildcards:**

- % → zero or more characters
-

_ → exactly one character

### ▶ Examples:

```
-- Starts with 'A'
SELECT * FROM Customers
WHERE Name LIKE ‘A%';

-- Ends with 'son'
SELECT * FROM Customers
WHERE Name LIKE ‘%son';

-- Contains 'mit'
SELECT * FROM Customers
```

```
WHERE Name LIKE '%mit%';

-- Exactly 5-letter names starting with 'A'
SELECT * FROM Customers
WHERE Name LIKE 'A____';
```

### 4. IN and NOT IN Predicates

▶ IN

```
SELECT * FROM Students
WHERE Grade IN ('A', 'B', 'C');
```

Explanation: Fetch students with grade A, B, or C.

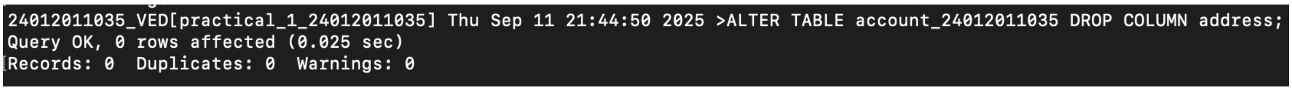
▶ NOT IN

```
SELECT * FROM Students
WHERE Grade NOT IN ('F', 'D');
```

Explanation: Fetch students whose grades are not F or D.

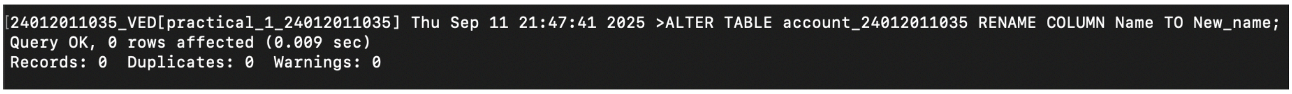
**NOTE: USE TABLES CREATED IN PRACTICAL-1. :-**

1. Drop address column from Account table (if not created, create it first).



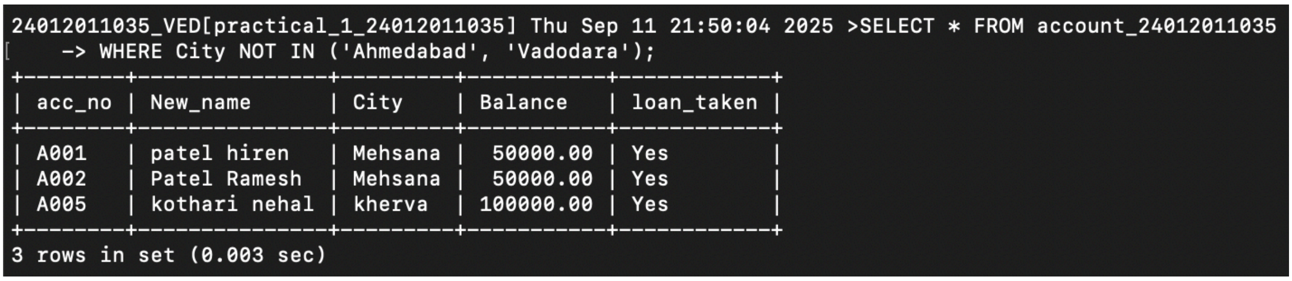
```
24012011035_VED[practical_1_24012011035] Thu Sep 11 21:44:50 2025 >ALTER TABLE account_24012011035 DROP COLUMN address;
Query OK, 0 rows affected (0.025 sec)
[Records: 0  Duplicates: 0  Warnings: 0
```

2. Rename Name to New_name in Account table.



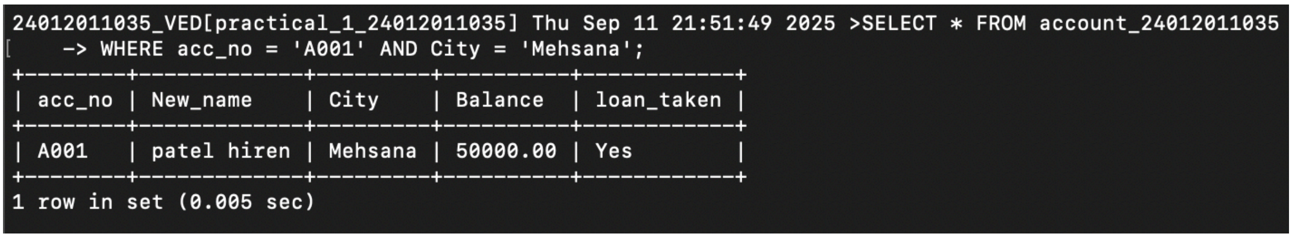
```
[24012011035_VED[practical_1_24012011035] Thu Sep 11 21:47:41 2025 >ALTER TABLE account_24012011035 RENAME COLUMN Name TO New_name;
Query OK, 0 rows affected (0.009 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

3. Retrieve specific information for the account holder who are not in 'Ahmedabad' or 'Vadodara'.



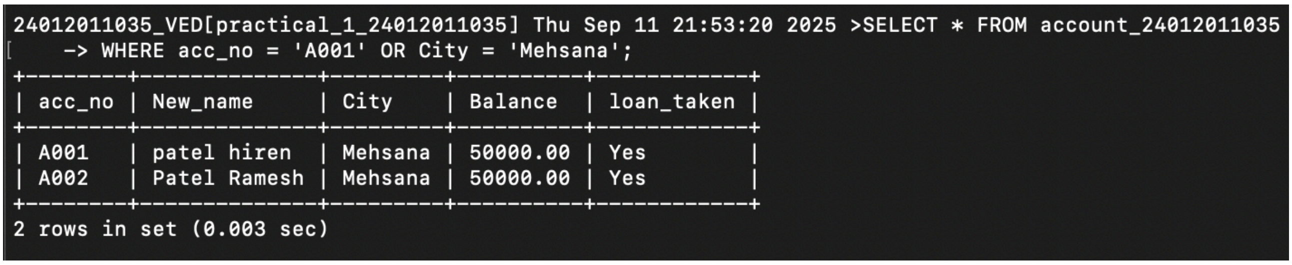
```
24012011035_VED[practical_1_24012011035] Thu Sep 11 21:50:04 2025 >SELECT * FROM account_24012011035
[    -> WHERE City NOT IN ('Ahmedabad', 'Vadodara');
+--------+---------------+---------+-----------+------------+
| acc_no | New_name      | City    | Balance   | loan_taken |
+--------+---------------+---------+-----------+------------+
| A001   | patel hiren   | Mehsana |  50000.00 | Yes        |
| A002   | Patel Ramesh  | Mehsana |  50000.00 | Yes        |
| A005   | kothari nehal | kherva  | 100000.00 | Yes        |
+--------+---------------+---------+-----------+------------+
3 rows in set (0.003 sec)
```

4. Display only those data whose account number is 'A001' and city is 'Mehsana'.



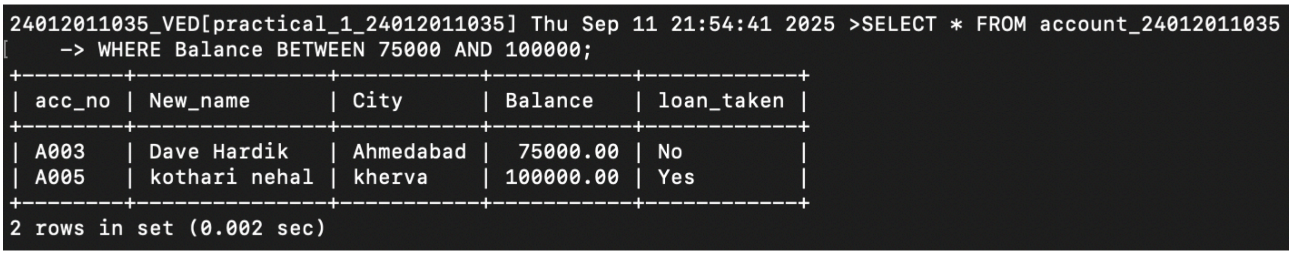
```
24012011035_VED[practical_1_24012011035] Thu Sep 11 21:51:49 2025 >SELECT * FROM account_24012011035
[    -> WHERE acc_no = 'A001' AND City = 'Mehsana';
+--------+-------------+---------+----------+------------+
| acc_no | New_name    | City    | Balance  | loan_taken |
+--------+-------------+---------+----------+------------+
| A001   | patel hiren | Mehsana | 50000.00 | Yes        |
+--------+-------------+---------+----------+------------+
1 row in set (0.005 sec)
```

5. Display only those data whose account number is ‘A001’ or city is ‘Mehsana’.



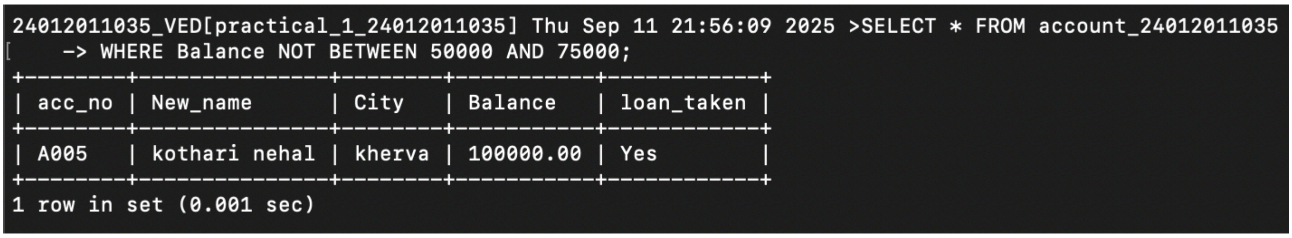
```
24012011035_VED[practical_1_24012011035] Thu Sep 11 21:53:20 2025 >SELECT * FROM account_24012011035
[    -> WHERE acc_no = 'A001' OR City = 'Mehsana';
+--------+--------------+---------+----------+------------+
| acc_no | New_name     | City    | Balance  | loan_taken |
+--------+--------------+---------+----------+------------+
| A001   | patel hiren  | Mehsana | 50000.00 | Yes        |
| A002   | Patel Ramesh | Mehsana | 50000.00 | Yes        |
+--------+--------------+---------+----------+------------+
2 rows in set (0.003 sec)
```

6. Retrieve those records of Account holders whose balance between 75000 and 100000.



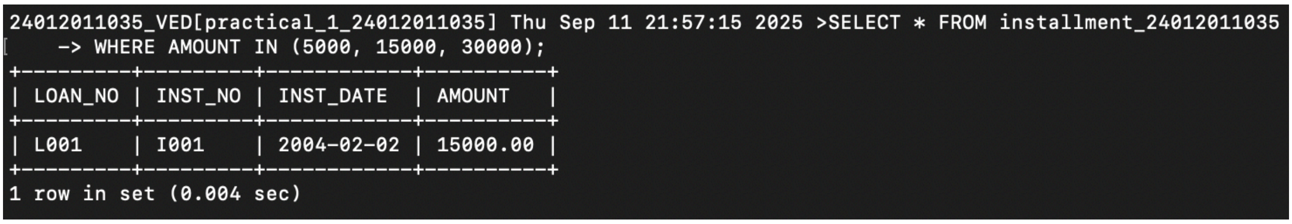
```
24012011035_VED[practical_1_24012011035] Thu Sep 11 21:54:41 2025 >SELECT * FROM account_24012011035
[    -> WHERE Balance BETWEEN 75000 AND 100000;
+--------+---------------+-----------+-----------+------------+
| acc_no | New_name      | City      | Balance   | loan_taken |
+--------+---------------+-----------+-----------+------------+
| A003   | Dave Hardik   | Ahmedabad |  75000.00 | No         |
| A005   | kothari nehal | kherva    | 100000.00 | Yes        |
+--------+---------------+-----------+-----------+------------+
2 rows in set (0.002 sec)
```

7. Retrieve those records of Account holders whose balance not between 50000 and 75000.



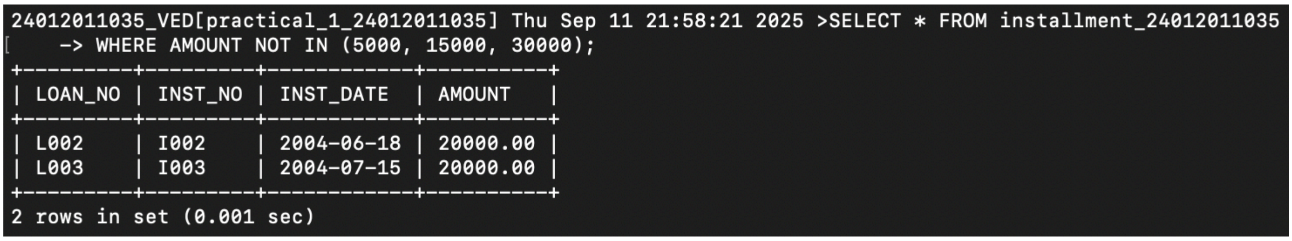
```
24012011035_VED[practical_1_24012011035] Thu Sep 11 21:56:09 2025 >SELECT * FROM account_24012011035
[    -> WHERE Balance NOT BETWEEN 50000 AND 75000;
+--------+---------------+--------+-----------+------------+
| acc_no | New_name      | City   | Balance   | loan_taken |
+--------+---------------+--------+-----------+------------+
| A005   | kothari nehal | kherva | 100000.00 | Yes        |
+--------+---------------+--------+-----------+------------+
1 row in set (0.001 sec)
```

8. Display only those records whose amount is 5000, 15000, 30000 from installment table.



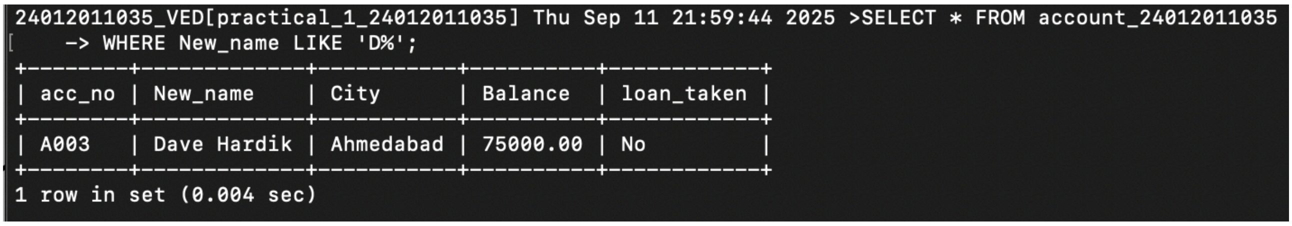
```
24012011035_VED[practical_1_24012011035] Thu Sep 11 21:57:15 2025 >SELECT * FROM installment_24012011035
[    -> WHERE AMOUNT IN (5000, 15000, 30000);
+---------+---------+------------+----------+
| LOAN_NO | INST_NO | INST_DATE  | AMOUNT   |
+---------+---------+------------+----------+
| L001    | I001    | 2004-02-02 | 15000.00 |
+---------+---------+------------+----------+
1 row in set (0.004 sec)
```

9. Display only those records whose amount is not in 5000, 15000, 30000 from installment table.

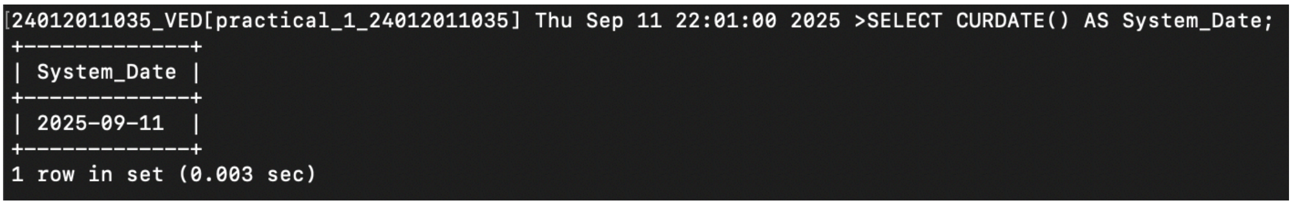


```
24012011035_VED[practical_1_24012011035] Thu Sep 11 21:58:21 2025 >SELECT * FROM installment_24012011035
[    -> WHERE AMOUNT NOT IN (5000, 15000, 30000);
+---------+---------+------------+----------+
| LOAN_NO | INST_NO | INST_DATE  | AMOUNT   |
+---------+---------+------------+----------+
| L002    | I002    | 2004-06-18 | 20000.00 |
| L003    | I003    | 2004-07-15 | 20000.00 |
+---------+---------+------------+----------+
2 rows in set (0.001 sec)
```

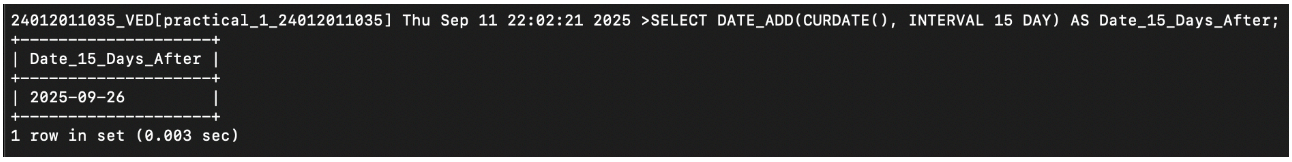
10. Display those records of account holders whose name starts with ‘D’.



```
24012011035_VED[practical_1_24012011035] Thu Sep 11 21:59:44 2025 >SELECT * FROM account_24012011035
[    -> WHERE New_name LIKE 'D%';
+--------+-------------+-----------+----------+------------+
| acc_no | New_name    | City      | Balance  | loan_taken |
+--------+-------------+-----------+----------+------------+
| A003   | Dave Hardik | Ahmedabad | 75000.00 | No         |
+--------+-------------+-----------+----------+------------+
1 row in set (0.004 sec)
```

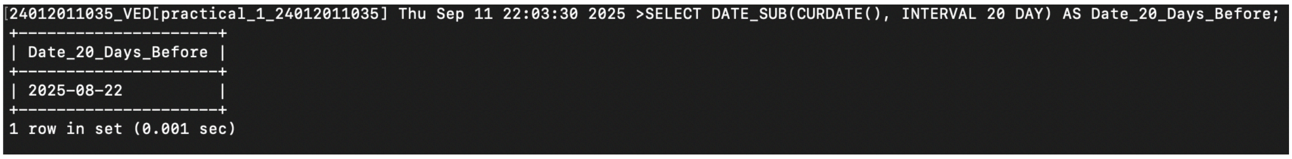


```
[24012011035_VED[practical_1_24012011035] Thu Sep 11 22:01:00 2025 >SELECT CURDATE() AS System_Date;
+-------------+
| System_Date |
+-------------+
| 2025-09-11  |
+-------------+
1 row in set (0.003 sec)
```



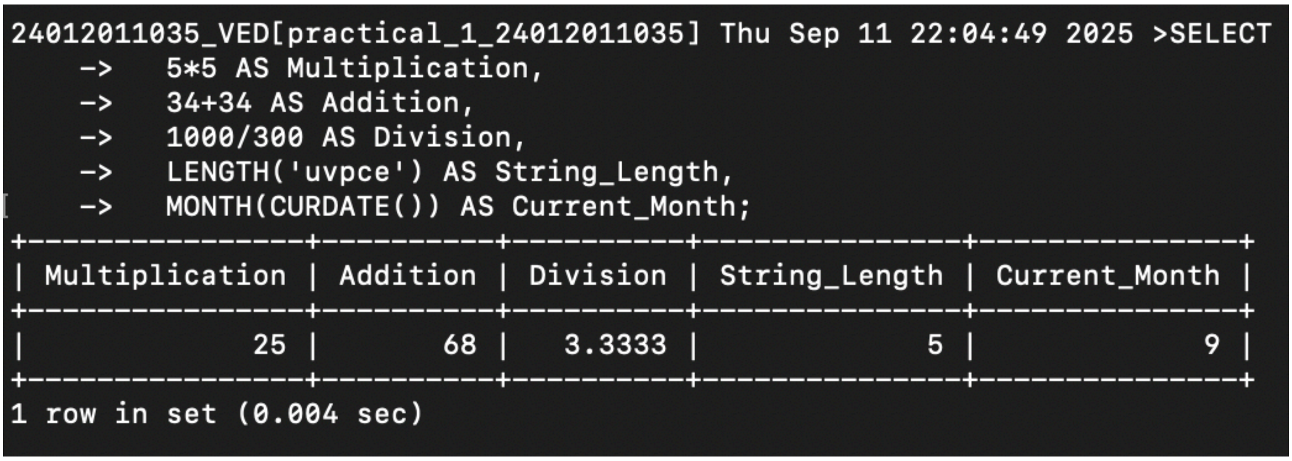
```
24012011035_VED[practical_1_24012011035] Thu Sep 11 22:02:21 2025 >SELECT DATE_ADD(CURDATE(), INTERVAL 15 DAY) AS Date_15_Days_After;
+--------------------+
| Date_15_Days_After |
+--------------------+
| 2025-09-26         |
+--------------------+
1 row in set (0.003 sec)
```

11. Display System date.



```
[24012011035_VED[practical_1_24012011035] Thu Sep 11 22:03:30 2025 >SELECT DATE_SUB(CURDATE(), INTERVAL 20 DAY) AS Date_20_Days_Before;
+---------------------+
| Date_20_Days_Before |
+---------------------+
| 2025-08-22          |
+---------------------+
1 row in set (0.001 sec)
```

12. Find the date,15 days after today’s date.
13. Find the date,20 days before today’s date.



```
24012011035_VED[practical_1_24012011035] Thu Sep 11 22:04:49 2025 >SELECT
    ->    5*5 AS Multiplication,
    ->    34+34 AS Addition,
    ->    1000/300 AS Division,
    ->    LENGTH('uvpce') AS String_Length,
[    ->    MONTH(CURDATE()) AS Current_Month;
+----------------+----------+----------+---------------+---------------+
| Multiplication | Addition | Division | String_Length | Current_Month |
+----------------+----------+----------+---------------+---------------+
|             25 |       68 |   3.3333 |             5 |             9 |
+----------------+----------+----------+---------------+---------------+
1 row in set (0.004 sec)
```

14. Perform the following operation using DUAL table.
    5*5, 34+34, 1000/300, length of ‘uvpce’, display only month of system date

## Function Based Queries.

## 1. Group by function

The GROUP BY clause is used to group rows that have the same values in specified columns. It is commonly used with aggregate functions such as SUM(), AVG(), COUNT(), MAX(), and MIN().

**Syntax:**

```
SELECT column_name, AGGREGATE_FUNCTION(column_name)
FROM table_name
GROUP BY column_name;
```

**Example:**

```
-- Total salary by department
SELECT Department, SUM(Salary) AS Total_Salary
FROM Employees
GROUP BY Department;

-- Average marks by subject
SELECT Subject, AVG(Marks) AS Average_Marks
FROM Results
GROUP BY Subject;
```

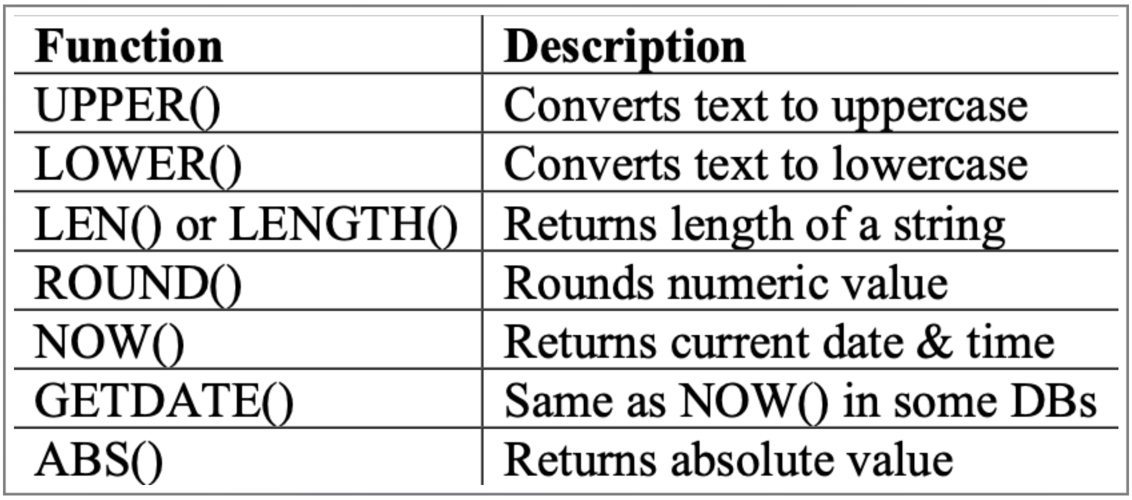
## 2. Scalar function

Scalar functions return a single value based on the input value. They operate on each row individually.

**Example:**

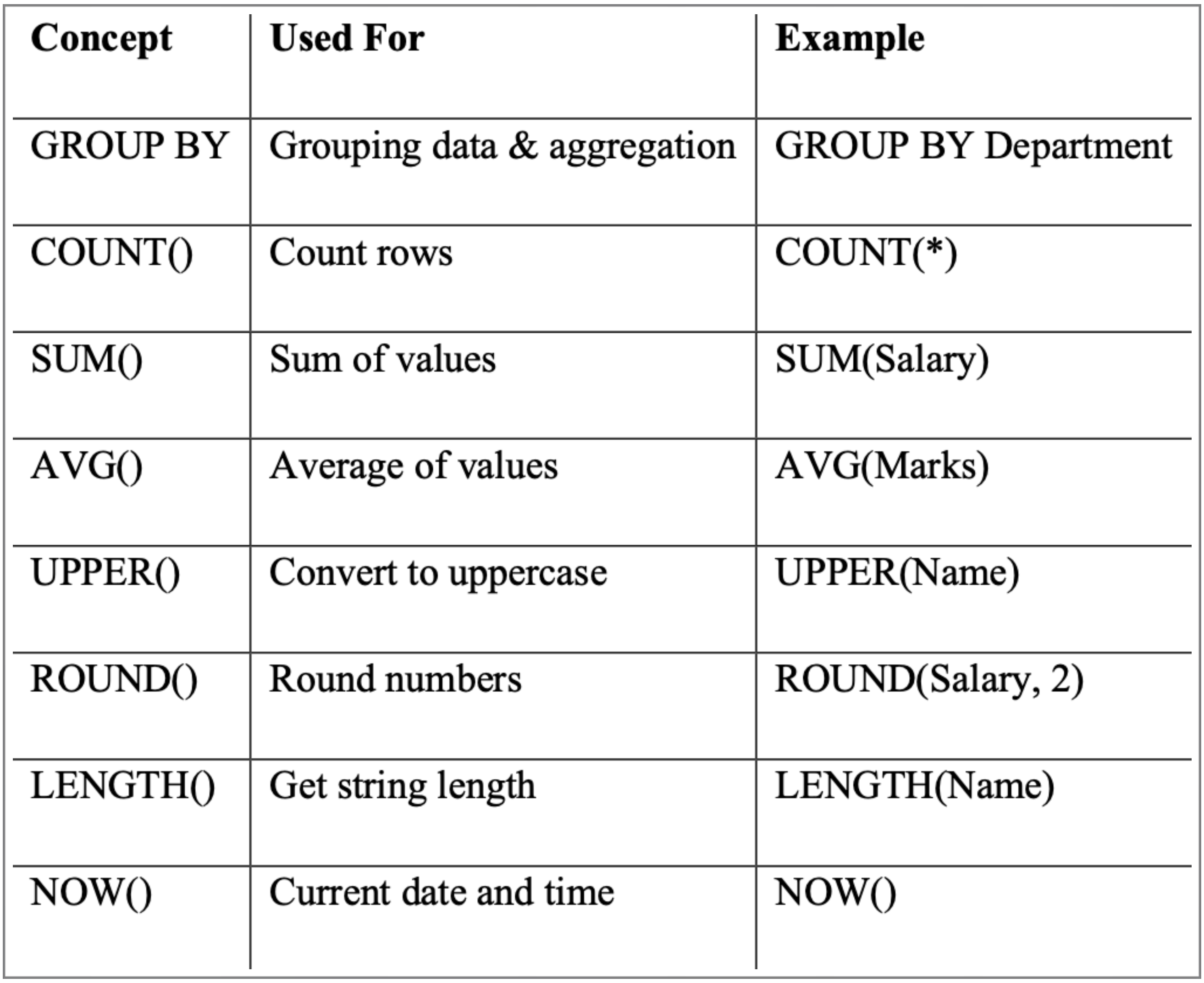
```
-- Convert name to uppercase
SELECT UPPER(Name) AS Upper_Name
FROM Customers;

-- Round salary to nearest thousand
SELECT Name, ROUND(Salary, -3) AS Rounded_Salary
FROM Employees;

-- Length of name
SELECT Name, LENGTH(Name) AS Name_Length
FROM Customers;

-- Current date and time
SELECT NOW() AS Current_Timestamp;
```

### 🔷 Common Scalar Functions:



| Function | Description |
|---|---|
| UPPER() | Converts text to uppercase |
| LOWER() | Converts text to lowercase |
| LEN() or LENGTH() | Returns length of a string |
| ROUND() | Rounds numeric value |
| NOW() | Returns current date & time |
| GETDATE() | Same as NOW() in some DBs |
| ABS() | Returns absolute value |

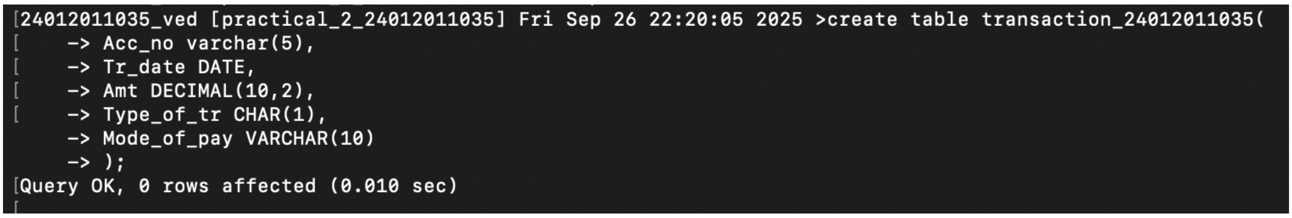
**Summary Table**



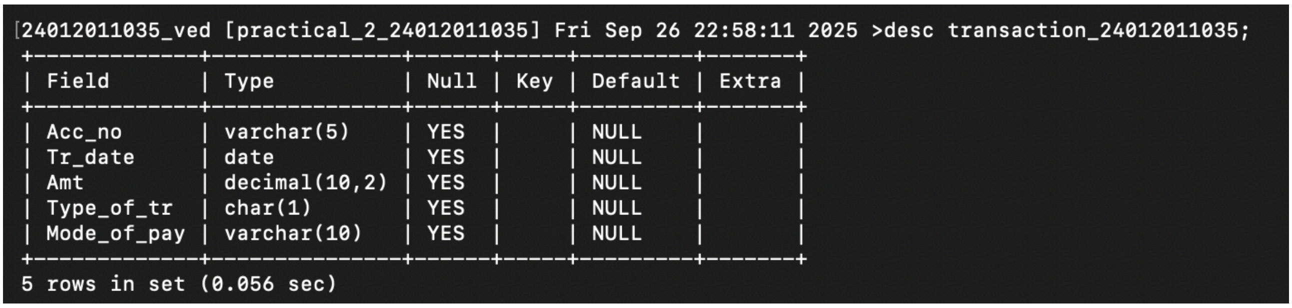
| Concept | Used For | Example |
|---|---|---|
| GROUP BY | Grouping data & aggregation | GROUP BY Department |
| COUNT() | Count rows | COUNT(*) |
| SUM() | Sum of values | SUM(Salary) |
| AVG() | Average of values | AVG(Marks) |
| UPPER() | Convert to uppercase | UPPER(Name) |
| ROUND() | Round numbers | ROUND(Salary, 2) |
| LENGTH() | Get string length | LENGTH(Name) |
| NOW() | Current date and time | NOW() |

**Create TABLE “TRANSACTION_yorEnNo” as given below.**

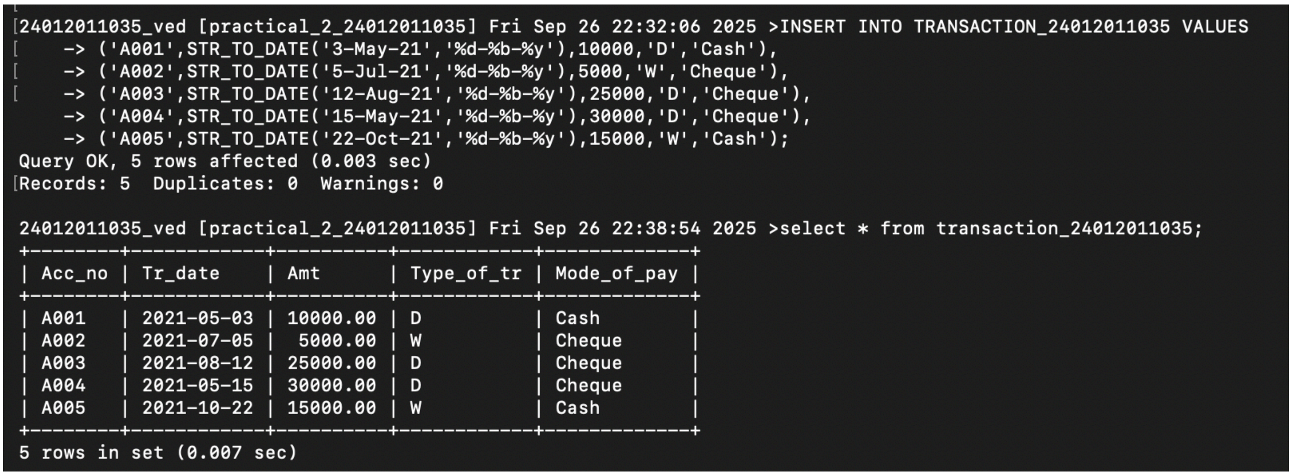
| Acc_no | Tr_date | Amt | Type_of_tr | Mode_of_pay |
|---|---|---|---|---|
| A001 | 3-may-21 | 10000 | D | Cash |
| A002 | 5-july-21 | 5000 | W | Cheque |
| A003 | 12-Aug-21 | 25000 | D | Cheque |
| A004 | 15-may-21 | 30000 | D | Cheque |
| A005 | 22-oct-21 | 15000 | W | Cash |



```
24012011035_ved [practical_2_24012011035] Fri Sep 26 22:20:05 2025 >create table transaction_24012011035(
    -> Acc_no varchar(5),
    -> Tr_date DATE,
    -> Amt DECIMAL(10,2),
    -> Type_of_tr CHAR(1),
    -> Mode_of_pay VARCHAR(10)
    -> );
Query OK, 0 rows affected (0.010 sec)
```



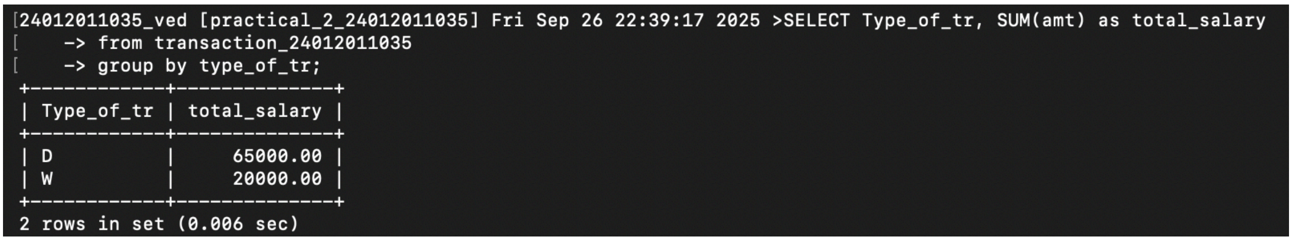
```
24012011035_ved [practical_2_24012011035] Fri Sep 26 22:58:11 2025 >desc transaction_24012011035;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| Acc_no      | varchar(5)    | YES  |     | NULL    |       |
| Tr_date     | date          | YES  |     | NULL    |       |
| Amt         | decimal(10,2) | YES  |     | NULL    |       |
| Type_of_tr  | char(1)       | YES  |     | NULL    |       |
| Mode_of_pay | varchar(10)   | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
5 rows in set (0.056 sec)
```



```
24012011035_ved [practical_2_24012011035] Fri Sep 26 22:32:06 2025 >INSERT INTO TRANSACTION_24012011035 VALUES
    -> ('A001',STR_TO_DATE('3-May-21','%d-%b-%y'),10000,'D','Cash'),
    -> ('A002',STR_TO_DATE('5-Jul-21','%d-%b-%y'),5000,'W','Cheque'),
    -> ('A003',STR_TO_DATE('12-Aug-21','%d-%b-%y'),25000,'D','Cheque'),
    -> ('A004',STR_TO_DATE('15-May-21','%d-%b-%y'),30000,'D','Cheque'),
    -> ('A005',STR_TO_DATE('22-Oct-21','%d-%b-%y'),15000,'W','Cash');
Query OK, 5 rows affected (0.003 sec)
Records: 5  Duplicates: 0  Warnings: 0

24012011035_ved [practical_2_24012011035] Fri Sep 26 22:38:54 2025 >select * from transaction_24012011035;
+--------+------------+----------+------------+-------------+
| Acc_no | Tr_date    | Amt      | Type_of_tr | Mode_of_pay |
+--------+------------+----------+------------+-------------+
| A001   | 2021-05-03 | 10000.00 | D          | Cash        |
| A002   | 2021-07-05 |  5000.00 | W          | Cheque      |
| A003   | 2021-08-12 | 25000.00 | D          | Cheque      |
| A004   | 2021-05-15 | 30000.00 | D          | Cheque      |
| A005   | 2021-10-22 | 15000.00 | W          | Cash        |
+--------+------------+----------+------------+-------------+
5 rows in set (0.007 sec)
```

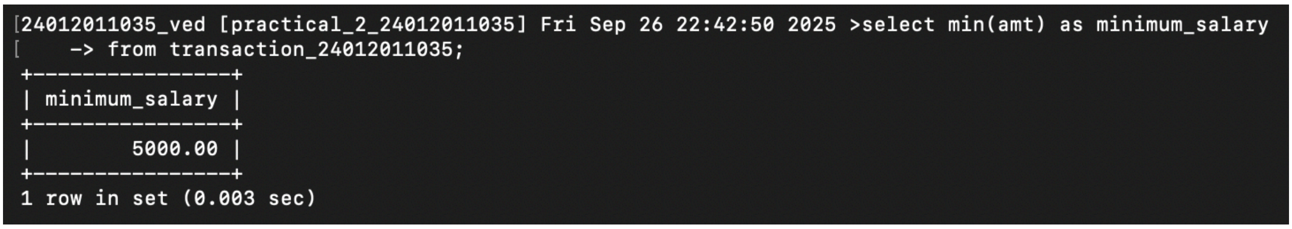
**Perform given queries:**

1. Find the total transaction amount of account holder from transaction table.



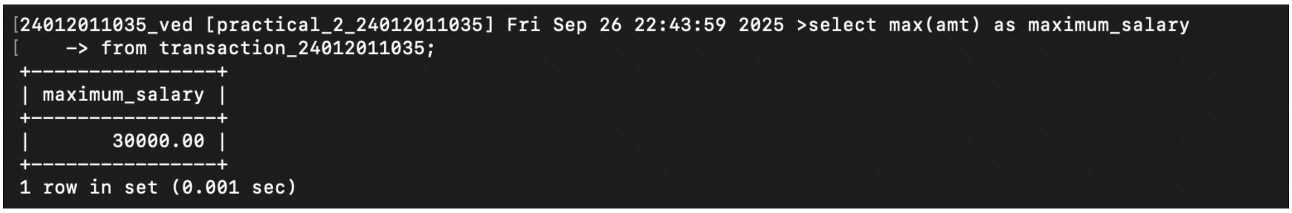
```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 22:39:17 2025 >SELECT Type_of_tr, SUM(amt) as total_salary
[    -> from transaction_24012011035
[    -> group by type_of_tr;
+------------+--------------+
| Type_of_tr | total_salary |
+------------+--------------+
| D          |     65000.00 |
| W          |     20000.00 |
+------------+--------------+
2 rows in set (0.006 sec)
```

2. Find minimum amount of transaction table.



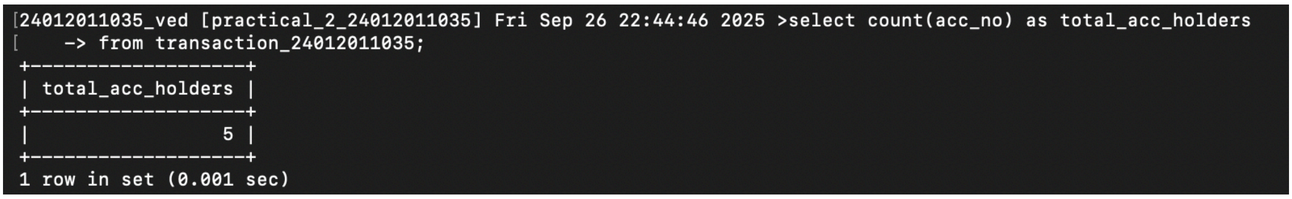
```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 22:42:50 2025 >select min(amt) as minimum_salary
[    -> from transaction_24012011035;
+----------------+
| minimum_salary |
+----------------+
|        5000.00 |
+----------------+
1 row in set (0.003 sec)
```

3. Find maximum amount of transaction table.



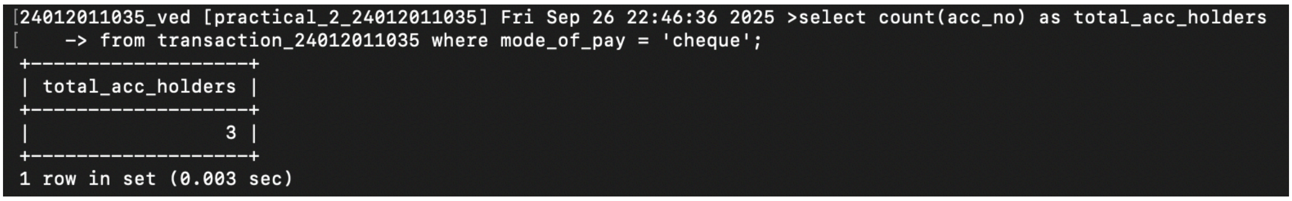
```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 22:43:59 2025 >select max(amt) as maximum_salary
[    -> from transaction_24012011035;
+----------------+
| maximum_salary |
+----------------+
|       30000.00 |
+----------------+
1 row in set (0.001 sec)
```

4. Count the total account holders from transaction table.



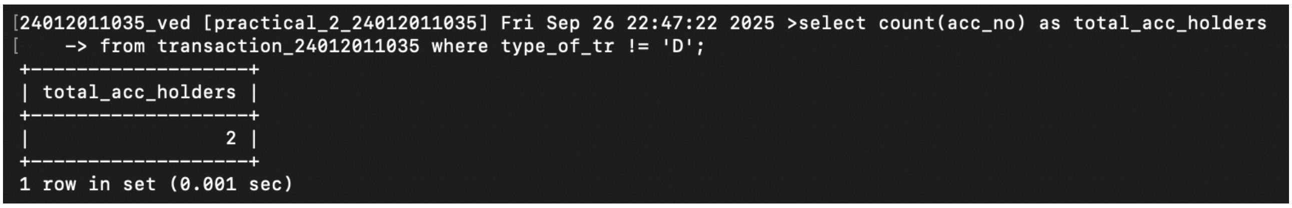
```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 22:44:46 2025 >select count(acc_no) as total_acc_holders
[    -> from transaction_24012011035;
+-------------------+
| total_acc_holders |
+-------------------+
|                 5 |
+-------------------+
1 row in set (0.001 sec)
```

5. Count only those records whose made of payment is ‘Cheque’.



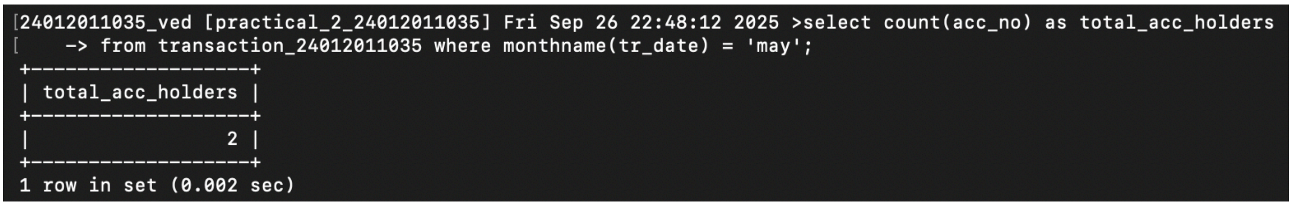
```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 22:46:36 2025 >select count(acc_no) as total_acc_holders
[    -> from transaction_24012011035 where mode_of_pay = 'cheque';
+-------------------+
| total_acc_holders |
+-------------------+
|                 3 |
+-------------------+
1 row in set (0.003 sec)
```

6. Count only those records whose Type_of_tr is not ‘D’.



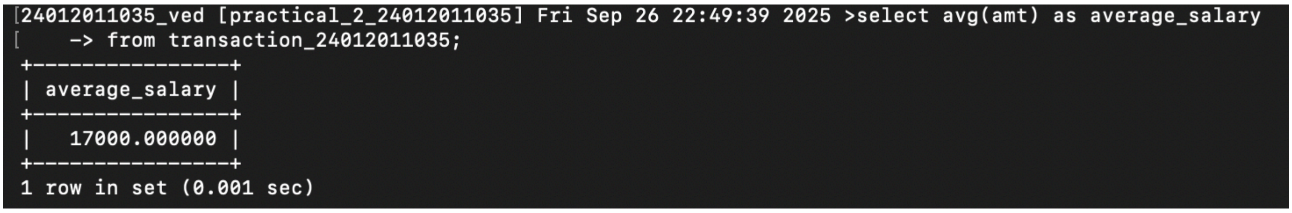
```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 22:47:22 2025 >select count(acc_no) as total_acc_holders
[    -> from transaction_24012011035 where type_of_tr != 'D';
+-------------------+
| total_acc_holders |
+-------------------+
|                 2 |
+-------------------+
1 row in set (0.001 sec)
```

7. Count only those records whose transaction made in the month of ‘may’.



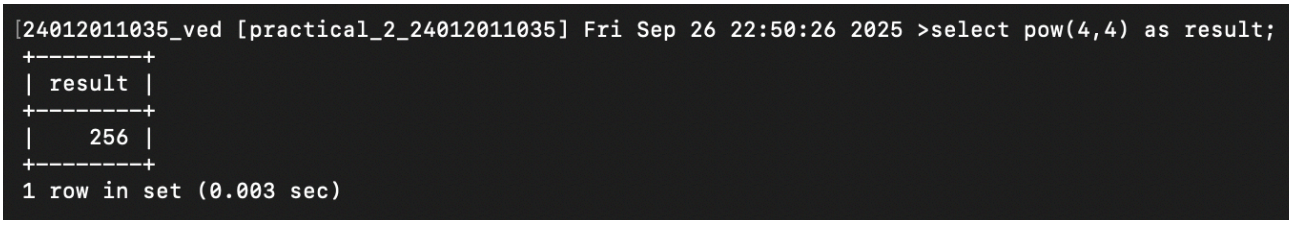
```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 22:48:12 2025 >select count(acc_no) as total_acc_holders
[    -> from transaction_24012011035 where monthname(tr_date) = 'may';
+-------------------+
| total_acc_holders |
+-------------------+
|                 2 |
+-------------------+
1 row in set (0.002 sec)
```

8. Find the average value of transaction.



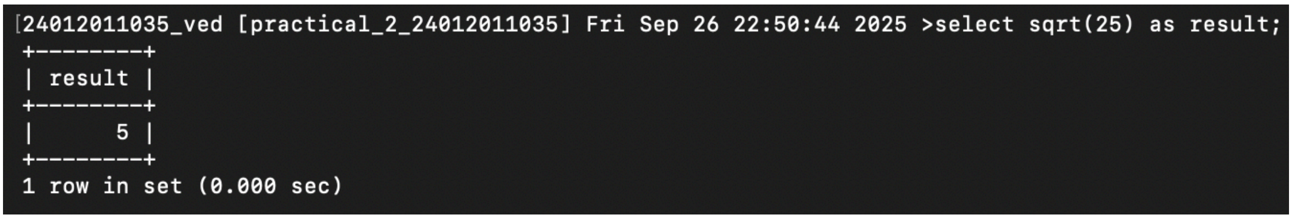
```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 22:49:39 2025 >select avg(amt) as average_salary
[    -> from transaction_24012011035;
+----------------+
| average_salary |
+----------------+
|   17000.000000 |
+----------------+
1 row in set (0.001 sec)
```

9. Display the result of 4 rest to 4 (use power function).



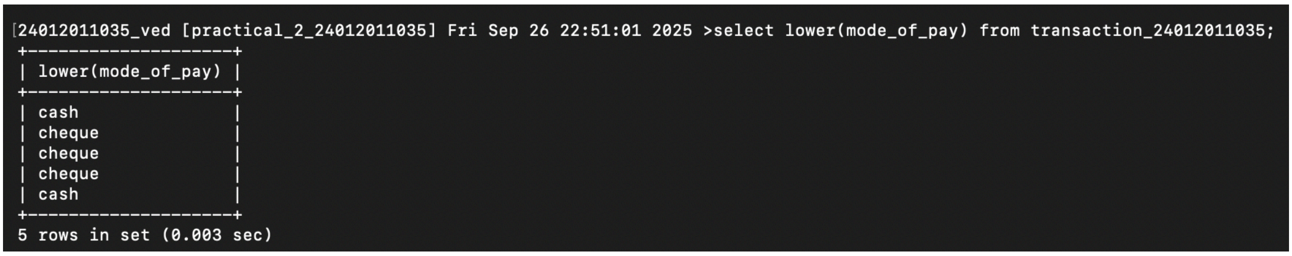
```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 22:50:26 2025 >select pow(4,4) as result;
+--------+
| result |
+--------+
|    256 |
+--------+
1 row in set (0.003 sec)
```

10. Find the square root of 25 (use sqrt function).

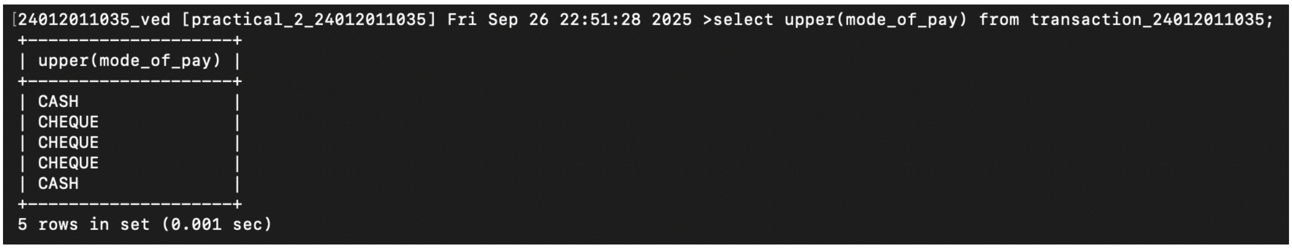


```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 22:50:44 2025 >select sqrt(25) as result;
+--------+
| result |
+--------+
|      5 |
+--------+
1 row in set (0.000 sec)
```

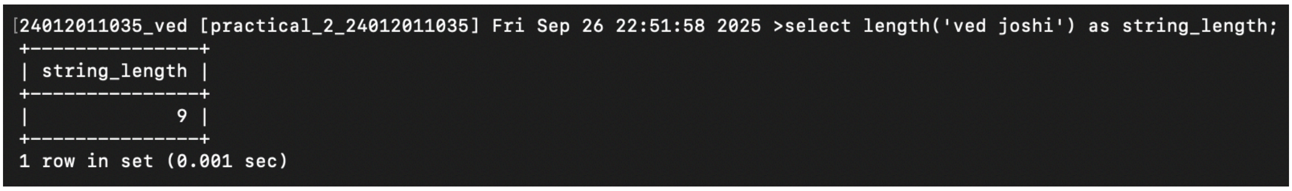
11. Write the query for the following inbuilt Function.
LOWER, INITCAP, UPPER, SUBSTR, LENGTH, LTRIM, RTRIM, LPAD,
RPAD.



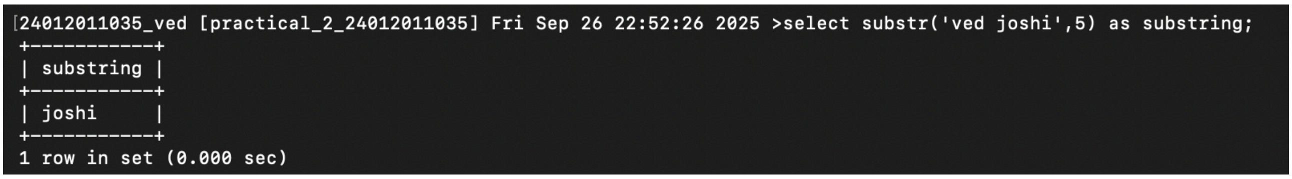
```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 22:51:01 2025 >select lower(mode_of_pay) from transaction_24012011035;
+---------------------+
| lower(mode_of_pay) |
+---------------------+
| cash               |
| cheque             |
| cheque             |
| cheque             |
| cash               |
+---------------------+
5 rows in set (0.003 sec)
```



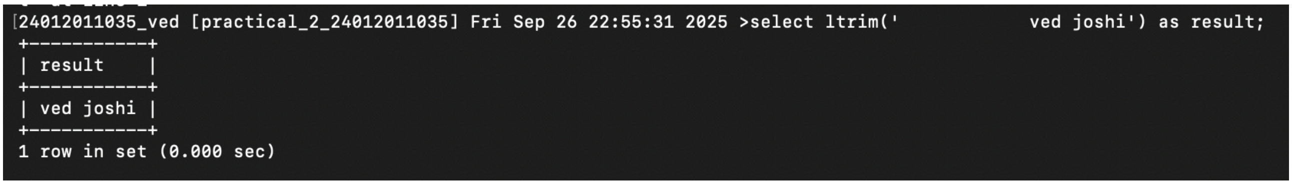
```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 22:51:28 2025 >select upper(mode_of_pay) from transaction_24012011035;
+---------------------+
| upper(mode_of_pay) |
+---------------------+
| CASH               |
| CHEQUE             |
| CHEQUE             |
| CHEQUE             |
| CASH               |
+---------------------+
5 rows in set (0.001 sec)
```



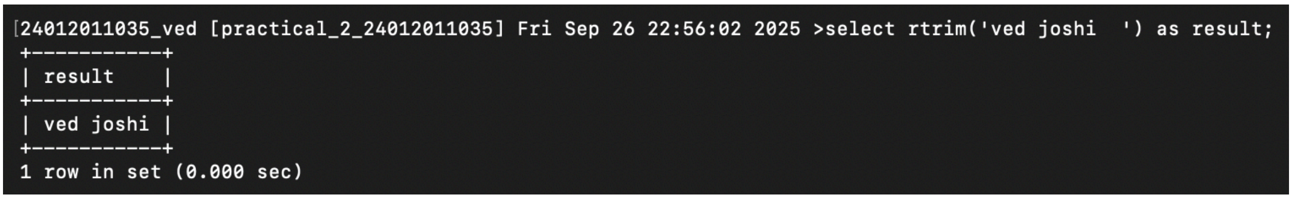
```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 22:51:58 2025 >select length('ved joshi') as string_length;
+---------------+
| string_length |
+---------------+
|             9 |
+---------------+
1 row in set (0.001 sec)
```



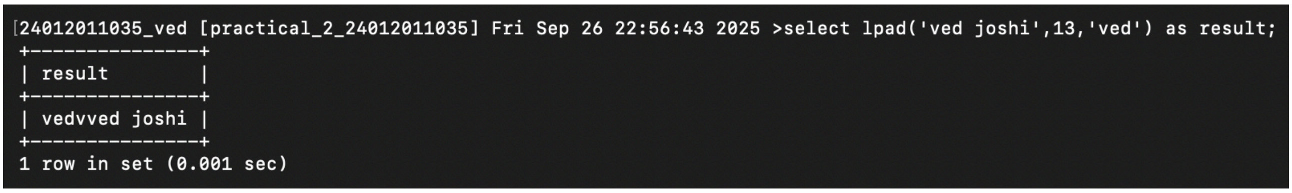
```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 22:52:26 2025 >select substr('ved joshi',5) as substring;
+-----------+
| substring |
+-----------+
| joshi     |
+-----------+
1 row in set (0.000 sec)
```



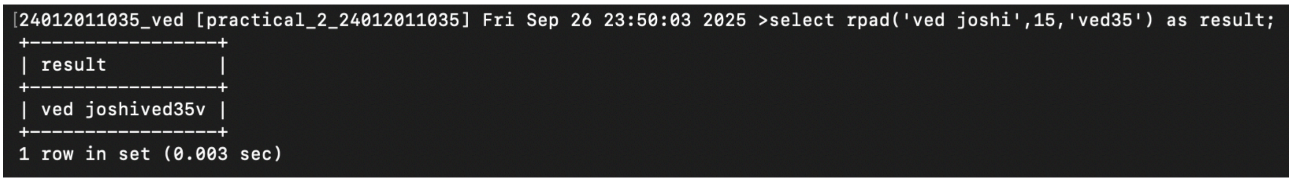
```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 22:55:31 2025 >select ltrim('            ved joshi') as result;
+-----------+
| result    |
+-----------+
| ved joshi |
+-----------+
1 row in set (0.000 sec)
```



```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 22:56:02 2025 >select rtrim('ved joshi  ') as result;
+-----------+
| result    |
+-----------+
| ved joshi |
+-----------+
1 row in set (0.000 sec)
```



```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 22:56:43 2025 >select lpad('ved joshi',13,'ved') as result;
+---------------+
| result        |
+---------------+
| vedvved joshi |
+---------------+
1 row in set (0.001 sec)
```



```
[24012011035_ved [practical_2_24012011035] Fri Sep 26 23:50:03 2025 >select rpad('ved joshi',15,'ved35') as result;
+-----------------+
| result          |
+-----------------+
| ved joshived35v |
+-----------------+
1 row in set (0.003 sec)
```