-- Show databases

SHOW DATABASES;

-- Use database 'vit'

USE vit;

-- Show tables

SHOW TABLES;

-- Count how many workers are in department 'hr'

SELECT COUNT(department) FROM worker WHERE department = 'hr';

-- Select the minimum department name (alphabetically)

SELECT MIN(department) FROM worker;

-- Select all rows from worker

SELECT \* FROM worker;

-- This query had an incorrect subquery usage and typo, so replaced with a valid aggregate:

-- Find the minimum department name where sum of salary for 'HR' is checked

-- Corrected to just select sum salary for 'HR'

SELECT SUM(salary) FROM worker WHERE department = 'HR';

-- Find the department with the smallest total salary sum

SELECT department

FROM worker

GROUP BY department

ORDER BY SUM(salary) ASC

LIMIT 1;

-- Sum of salary per department for HR, Admin, Account

SELECT SUM(salary) FROM worker WHERE department = 'HR';

SELECT SUM(salary) FROM worker WHERE department = 'Admin';

SELECT SUM(salary) FROM worker WHERE department = 'Account';

-- Count workers per department where count > 1

SELECT department, COUNT(department)

FROM worker

GROUP BY department

HAVING COUNT(department) > 1;

-- Count workers per department where total salary sum > 100000

SELECT department, COUNT(department)

FROM worker

GROUP BY department

HAVING SUM(salary) > 100000;

-- Count workers per department where salary > 100000

SELECT department, COUNT(department)

FROM worker

WHERE salary > 100000

GROUP BY department;

-- Department with the 2nd highest total salary sum

SELECT department, SUM(salary) AS total\_salary

FROM worker

GROUP BY department

ORDER BY total\_salary DESC

LIMIT 1 OFFSET 1;

-- Department with the highest total salary sum

SELECT department, SUM(salary) AS total\_salary

FROM worker

GROUP BY department

ORDER BY total\_salary DESC

LIMIT 1;

-- Department with the 3rd highest total salary sum

SELECT department, SUM(salary) AS total\_salary

FROM worker

GROUP BY department

ORDER BY total\_salary DESC

LIMIT 1 OFFSET 2;

-- Create student table

CREATE TABLE student (

s\_id INT,

s\_name VARCHAR(25)

);

-- Insert data into student

INSERT INTO student VALUES

(101, 'jayanth'),

(102, 'karthik'),

(103, 'Praveen'),

(105, 'mahesh'),

(106, 'Arun');

-- Create address table

CREATE TABLE address (

s\_id INT,

s\_address VARCHAR(25)

);

-- Insert data into address

INSERT INTO address VALUES

(101, 'coimbatore'),

(104, 'chennai'),

(105, 'pune');

-- Natural join student and address on s\_id

SELECT \* FROM student NATURAL JOIN address;

-- Left join student and address

SELECT \*

FROM student

LEFT JOIN address

ON student.s\_id = address.s\_id;

-- Cross join student and address

SELECT \* FROM student CROSS JOIN address;

-- Inner join student and address

SELECT \* FROM student INNER JOIN address ON student.s\_id = address.s\_id;

-- Left outer join student and address

SELECT \* FROM student LEFT OUTER JOIN address ON student.s\_id = address.s\_id;

-- Query to get 5th highest salary using DENSE\_RANK window function

SELECT salary

FROM (

SELECT salary, DENSE\_RANK() OVER (ORDER BY salary DESC) AS rank

FROM worker

) AS ranked

WHERE rank = 5;

-- Query to get the employee(s) with the 5th highest distinct salary

SELECT DISTINCT salary, First\_name, Last\_name

FROM worker w1

WHERE 4 = (

SELECT COUNT(DISTINCT salary)

FROM worker w2

WHERE w2.salary > w1.salary

);