

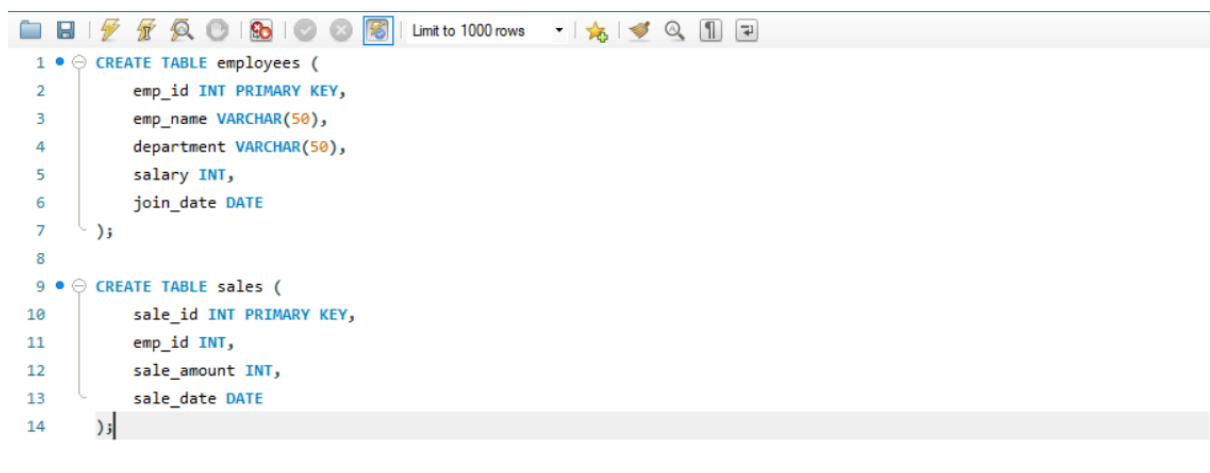
CODTECH Internship

TASK 2

Data Analysis With Complex Queries

NAME: S.Arun Ganesh

1. Table Creation



The image shows a screenshot of a SQL IDE window. The window has a toolbar at the top with various icons for file operations, execution, and navigation. Below the toolbar, the SQL code is displayed in a text editor. The code consists of two SQL statements: a 'CREATE TABLE employees' statement and a 'CREATE TABLE sales' statement. The 'employees' table has columns: emp_id (INT PRIMARY KEY), emp_name (VARCHAR(50)), department (VARCHAR(50)), salary (INT), and join_date (DATE). The 'sales' table has columns: sale_id (INT PRIMARY KEY), emp_id (INT), sale_amount (INT), and sale_date (DATE). The code is numbered from 1 to 14 on the left side of the editor.

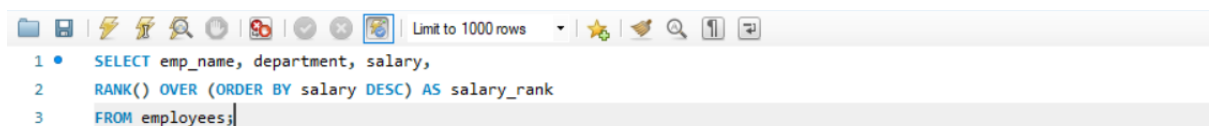
```
1 • CREATE TABLE employees (  
2     emp_id INT PRIMARY KEY,  
3     emp_name VARCHAR(50),  
4     department VARCHAR(50),  
5     salary INT,  
6     join_date DATE  
7 );  
8  
9 • CREATE TABLE sales (  
10     sale_id INT PRIMARY KEY,  
11     emp_id INT,  
12     sale_amount INT,  
13     sale_date DATE  
14 );
```

2. Insertion values

```
1 • INSERT INTO employees VALUES
2   (1, 'Arun', 'IT', 50000, '2022-01-10'),
3   (2, 'Priya', 'HR', 45000, '2021-03-15'),
4   (3, 'Rahul', 'IT', 60000, '2020-07-21'),
5   (4, 'Sneha', 'Sales', 40000, '2022-06-01');
6
7 • INSERT INTO sales VALUES
8   (101, 1, 20000, '2024-01-10'),
9   (102, 1, 15000, '2024-02-12'),
10  (103, 3, 30000, '2024-01-05'),
11  (104, 4, 18000, '2024-02-20');
```

3.Window Function

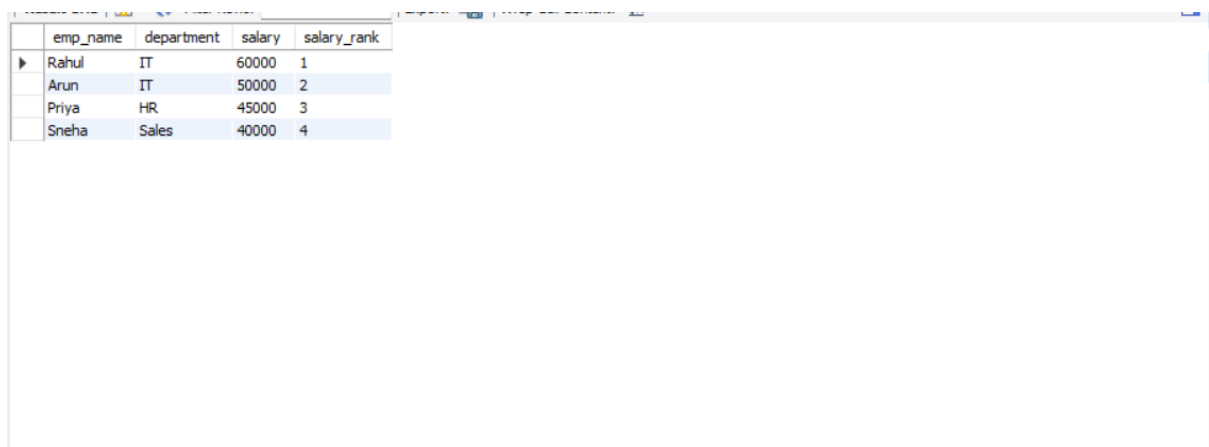
Qurey:



The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
1 • SELECT emp_name, department, salary,  
2 RANK() OVER (ORDER BY salary DESC) AS salary_rank  
3 FROM employees;
```

Output:

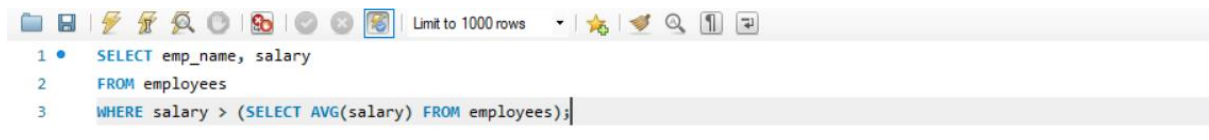


The screenshot shows the output of the SQL query in a table format. The table has four columns: emp_name, department, salary, and salary_rank. The data is as follows:

emp_name	department	salary	salary_rank
Rahul	IT	60000	1
Arun	IT	50000	2
Priya	HR	45000	3
Sneha	Sales	40000	4

4.Subquery

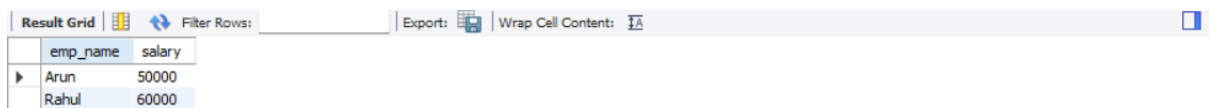
Query:



A screenshot of a SQL query editor interface. The top toolbar includes icons for file operations, execution, and a dropdown menu set to 'Limit to 1000 rows'. The query text is as follows:

```
1 • SELECT emp_name, salary
2 FROM employees
3 WHERE salary > (SELECT AVG(salary) FROM employees);
```

Output:



A screenshot of a SQL query result grid. The toolbar at the top includes 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. The table below shows the results of the query:

	emp_name	salary
▶	Arun	50000
	Rahul	60000

5.CTE (Common Table Expression)

Query:

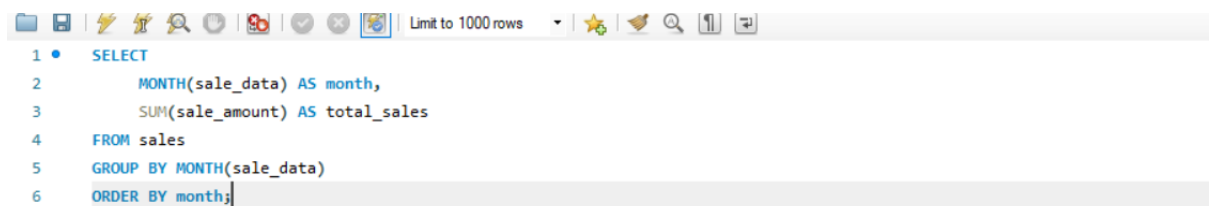
```
1 • WITH sales_summary AS (  
2     SELECT emp_id, SUM(sale_amount) AS total_sales  
3     FROM sales  
4     GROUP BY emp_id  
5 )  
6 SELECT e.emp_name, s.total_sales  
7 FROM employees e  
8 JOIN sales_summary s ON e.emp_id = s.emp_id;
```

Output:

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
emp_name	total_sales			
Arun	35000			
Rahul	30000			
Sneha	18000			

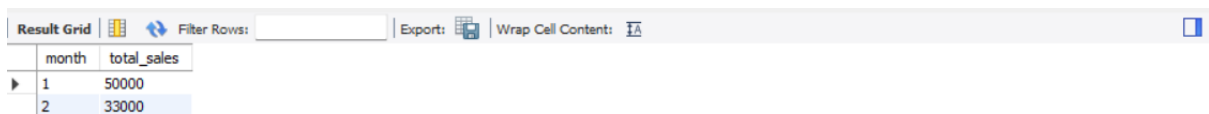
6. Trend or Pattern Analysis

Query:



```
1 • SELECT
2     MONTH(sale_data) AS month,
3     SUM(sale_amount) AS total_sales
4 FROM sales
5 GROUP BY MONTH(sale_data)
6 ORDER BY month;
```

Output:



	month	total_sales
▶	1	50000
	2	33000