

Name: Devkumar Navnitkumar Patel

Student_ID : W0787429

DAB-203

DAB203#Lab2 [Marks: 20]

1. Write a query to display the first name, last name, salary, and job title, for each employee. Order them by their last name.

```
SELECT E.first_name, E.last_name, E.salary, J.job_title
FROM employees E
LEFT JOIN jobs J
ON J.job_id = E.job_id
ORDER BY E.last_name
```

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'HRDatabase' selected. The right pane shows a query window with the following SQL code:

```
/*1. Write a query to display the first name, last name, salary, and job title, for each employee. Order them by their last name.*/
SELECT E.first_name, E.last_name, E.salary, J.job_title
FROM employees E
LEFT JOIN jobs J
ON J.job_id = E.job_id
ORDER BY E.last_name
```

Below the query window, the 'Results' pane displays the output of the query as a table with 16 rows and 4 columns: first_name, last_name, salary, and job_title. The data is sorted by last_name in ascending order.

first_name	last_name	salary	job_title
David	Austin	4800.00	Programmer
Hermann	Beer	10000.00	Public Relations Representative
Shelli	Baida	2900.00	Purchasing Clerk
Sarah	Bell	4000.00	Shipping Clerk
John	Chen	8200.00	Accountant
Karen	Colmenares	2500.00	Purchasing Clerk
Lex	De Haan	17000.00	Administration Vice President
Bruce	Ernst	6000.00	Programmer
Britney	Everett	3900.00	Shipping Clerk
Daniel	Faviet	9000.00	Accountant
Pat	Fay	6000.00	Marketing Representative
Adam	Fripp	8200.00	Stock Manager
William	Gietz	8300.00	Public Accountant
Kimberely	Grant	7000.00	Sales Representative
Nancy	Greenberg	12000.00	Finance Manager
Michael	Hartstein	13000.00	Marketing Manager

The status bar at the bottom indicates 'Query executed successfully.' and '40 rows'.

Name: Devkumar Navnitkumar Patel

Student_ID : W0787429

DAB-203

2. Write a query to display the first name, last name, department, city, and state province for each employee.

```
SELECT E.first_name, E.last_name,  
       D.department_name, L.city, L.state_province  
FROM employees E  
LEFT JOIN departments D  
ON D.department_id = E.department_id  
JOIN locations L  
ON L.location_id = D.location_id  
ORDER BY E.first_name
```

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'HRDatabase' selected. The right pane shows a query window with the following SQL query:

```
/** 2. Write a query to display the first name, last name, department, city, and state province for each employee.*/  
SELECT E.first_name, E.last_name,  
       D.department_name, L.city, L.state_province  
FROM employees E  
LEFT JOIN departments D  
ON D.department_id = E.department_id  
JOIN locations L  
ON L.location_id = D.location_id  
ORDER BY E.first_name
```

Below the query window, the 'Results' pane displays the output of the query as a table with 16 rows and 5 columns: first_name, last_name, department_name, city, and state_province. The status bar at the bottom indicates 'Query executed successfully' and '40 rows'.

	first_name	last_name	department_name	city	state_province
1	Adam	Fripp	Shipping	South San Francisco	California
2	Alexander	Hunold	IT	Southlake	Texas
3	Alexander	Khoo	Purchasing	Seattle	Washington
4	Britney	Everett	Shipping	South San Francisco	California
5	Bruce	Ernst	IT	Southlake	Texas
6	Charles	Johnson	Sales	Oxford	Oxford
7	Daniel	Farrel	Finance	Seattle	Washington
8	David	Austin	IT	Southlake	Texas
9	Den	Raphaely	Purchasing	Seattle	Washington
10	Diana	Lorentz	IT	Southlake	Texas
11	Guy	Himuro	Purchasing	Seattle	Washington
12	Hermann	Baer	Public Relations	Munich	Bavaria
13	Irene	Mikkilineni	Shipping	South San Francisco	California
14	Ismael	Sciarra	Finance	Seattle	Washington
15	Jack	Livingston	Sales	Oxford	Oxford
16	Jennifer	Whalen	Administration	Seattle	Washington

Name: Devkumar Navnitkumar Patel

Student_ID : W0787429

DAB-203

3. Write a query to display the first name, last name, department id and department name for all employees who work in the department having department id 5 or 7.

```
SELECT E.first_name, E.last_name, D.department_id, D.department_name
FROM employees E
LEFT JOIN departments D
ON D.department_id = E.department_id
WHERE D.department_id IN (5, 7)
ORDER BY E.first_name
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

```
/* 3. Write a query to display the first name, last name, department id and department name for all employees
who work in the department having department id 5 or 7. */
SELECT E.first_name, E.last_name, D.department_id, D.department_name
FROM employees E
LEFT JOIN departments D
ON D.department_id = E.department_id
WHERE D.department_id IN (5, 7)
ORDER BY E.first_name
```

The query has been executed successfully, and the results are displayed in the Results pane. The results show 8 rows of data:

	first_name	last_name	department_id	department_name
1	Adam	Fried	5	Shipping
2	Birney	Ewert	5	Shipping
3	Hermann	Baer	7	Public Relations
4	Irene	Mikkilineni	5	Shipping
5	Matthew	Weiss	5	Shipping
6	Peyam	Kaufing	5	Shipping
7	Sarah	Ball	5	Shipping
8	Shanta	Vollman	5	Shipping

The status bar at the bottom indicates that the query was executed successfully, returning 8 rows of data.

Name: Devkumar Navnitkumar Patel

Student_ID : W0787429

DAB-203

4. Write a query to display the first name, last name and salary for those employees whose salary is less than that of the employee with id 116.

```
SELECT E.first_name, E.last_name, E.salary
FROM employees E
WHERE E.salary < 2900
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

```
/* 4. Write a query to display the first name, last name and salary for those employees
whose salary is less than that of the employee with id 116.*/

SELECT E.first_name, E.last_name, E.salary
FROM employees E
WHERE E.salary < 2900
```

The query has been executed successfully, and the results are displayed in the Results pane. The results show 4 rows of data:

	first_name	last_name	salary
1	Sigal	Tobias	2800.00
2	Guy	Himuro	2600.00
3	Karen	Colmenares	2500.00
4	Irene	Mäkilinen	2700.00

The status bar at the bottom indicates that the query was executed successfully and returned 4 rows.

Name: Devkumar Navnitkumar Patel

Student_ID : W0787429

DAB-203

5. Write a query to display the employee id, name (first name and last name with a space between), hire date and a column called 'experience' with the values 'Senior' and 'Junior' when the hire date is 1994 or before, and after 1994 respectively . [You can use CONCAT function]

```
SELECT E.employee_id, E.first_name, E.last_name, E.hire_date,  
Year(E.hire_date) AS hire_year,  
(CASE WHEN Year(E.hire_date) <= 1994 THEN 'Senior' ELSE 'Junior' END)  
AS 'experience'  
FROM employees E
```

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'HRDatabase' selected. The right pane shows a query window with the following SQL code:

```
/* 5. Write a query to display the employee id, name (first name and last name with a space between),  
hire date and a column called 'experience' with the values 'Senior' and 'Junior'  
when the hire date is 1994 or before, and after 1994 respectively . [You can use CONCAT function] */  
  
SELECT E.employee_id, E.first_name, E.last_name, E.hire_date, Year(E.hire_date) AS hire_year,  
(CASE WHEN Year(E.hire_date) <= 1994 THEN 'Senior' ELSE 'Junior' END) AS 'experience'  
FROM employees E
```

The bottom pane shows the 'Results' tab with the following data:

employee_id	first_name	last_name	hire_date	hire_year	experience
100	Steven	King	1987-06-17	1987	Senior
101	Neena	Kochhar	1989-09-21	1989	Senior
102	Lex	De Haan	1993-01-13	1993	Senior
103	Alexander	Hunold	1990-01-03	1990	Senior
104	Bruce	Ernst	1991-05-21	1991	Senior
105	David	Austin	1997-06-25	1997	Junior
106	Valli	Pataballa	1998-02-05	1998	Junior
107	Diana	Lorentz	1999-02-07	1999	Junior
108	Nancy	Greenberg	1994-08-17	1994	Senior
109	Daniel	Faviet	1994-08-16	1994	Senior
110	John	Chen	1997-09-28	1997	Junior
111	Ismail	Schumi	1997-09-30	1997	Junior
112	Jose Manuel	Ullman	1998-03-07	1998	Junior
113	Luis	Popp	1999-12-07	1999	Junior
114	Den	Raphaely	1994-12-07	1994	Senior
115	Alexander	Khoo	1995-05-18	1995	Junior
116	Shelley	Stevens	1997-12-14	1997	Junior

The status bar at the bottom indicates 'Query executed successfully.' and '40 rows'.

Name: Devkumar Navnitkumar Patel

Student_ID : W0787429

DAB-203

6. Write a query to display the first name, last name, employee id, manager id and hire date of all employees who report to 'Neena'.

```
SELECT E.first_name, E.last_name, E.employee_id, E.manager_id,
e.hire_date
FROM employees E
WHERE E.manager_id =
(SELECT E.employee_id
FROM employees E
WHERE E.first_name = 'Neena')
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

```
/* 6. Write a query to display the first name, last name, employee id and hire date of all employees who report to 'Neena'. */
SELECT E.first_name, E.last_name, E.employee_id, E.manager_id, e.hire_date
FROM employees E
WHERE E.manager_id =
(SELECT E.employee_id
FROM employees E
WHERE E.first_name = 'Neena')
```

The query has been executed successfully, and the results are displayed in the Results pane. The results show 5 rows of data:

	first_name	last_name	employee_id	manager_id	hire_date
1	Neena	Greenberg	100	101	1984-09-17
2	Jennifer	Whalen	200	101	1987-09-17
3	Susan	Mavris	203	101	1994-06-07
4	Hermann	Baer	204	101	1994-06-07
5	Shelley	Higgins	205	101	1994-06-07

The status bar at the bottom indicates "Query executed successfully." and "5 rows".

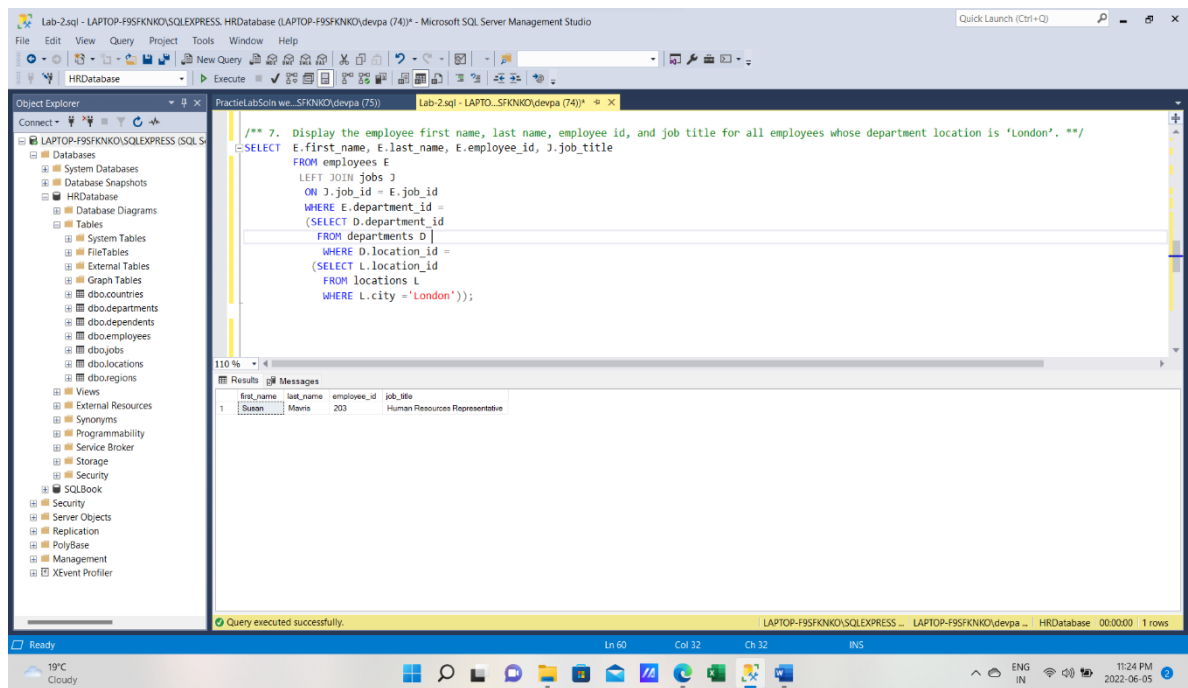
Name: Devkumar Navnitkumar Patel

Student_ID : W0787429

DAB-203

7. Display the employee first name, last name, employee id, and job title for all employees whose department location is 'London'.

```
SELECT E.first_name, E.last_name, E.employee_id, J.job_title
FROM employees E
LEFT JOIN jobs J
ON J.job_id = E.job_id
WHERE E.department_id =
(SELECT D.department_id
FROM departments D
WHERE D.location_id =
(SELECT L.location_id
FROM locations L
WHERE L.city = 'London'));
```



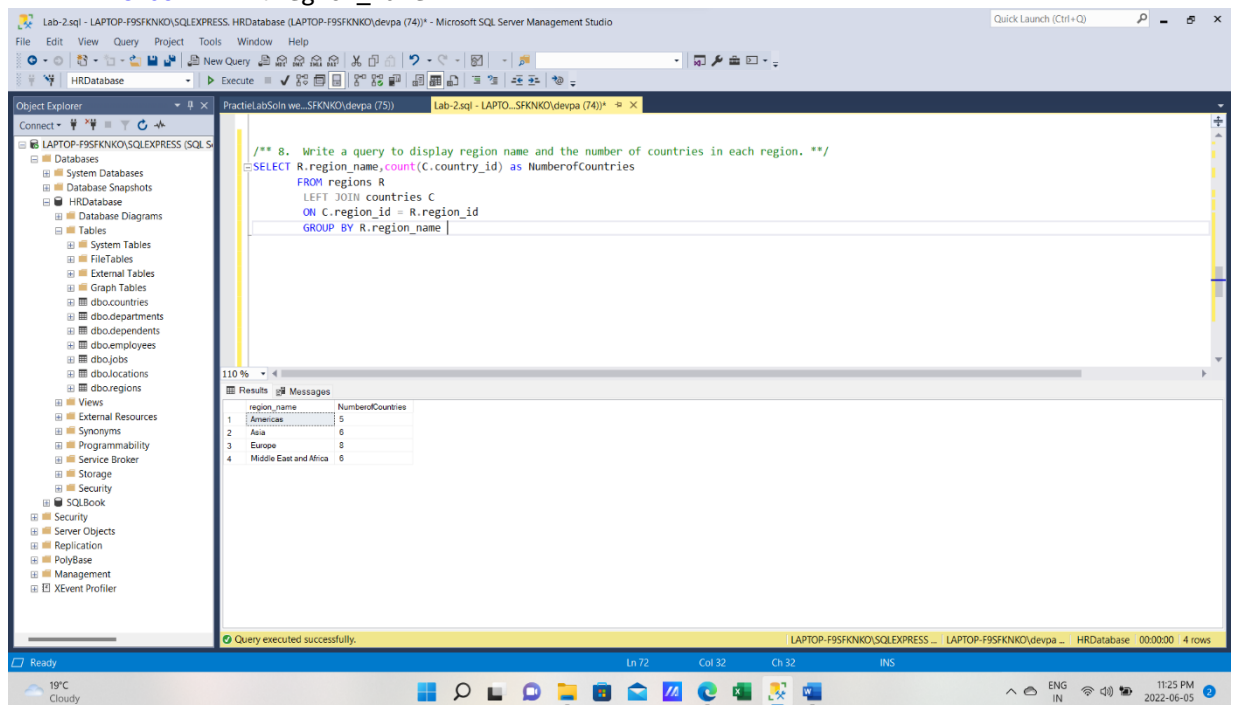
Name: Devkumar Navnitkumar Patel

Student_ID : W0787429

DAB-203

8. Write a query to display region name and the number of countries in each region.

```
SELECT R.region_name, count(C.country_id) as NumberOfCountries
FROM regions R
LEFT JOIN countries C
ON C.region_id = R.region_id
GROUP BY R.region_name
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

```
/** 8. Write a query to display region name and the number of countries in each region. */
SELECT R.region_name, count(C.country_id) as NumberOfCountries
FROM regions R
LEFT JOIN countries C
ON C.region_id = R.region_id
GROUP BY R.region_name
```

The query has been executed successfully, and the results are displayed in the Results pane. The results show the region name and the number of countries for each region:

region_name	NumberOfCountries
Americas	5
Asia	6
Europe	6
Middle East and Africa	6

The status bar at the bottom indicates that the query was executed successfully and returned 4 rows.

Name: Devkumar Navnitkumar Patel

Student_ID : W0787429

DAB-203

9. Write a query to display the department id, sum of salary as 'total_amt', number of employees in the department as 'number_of_employees', and a column called 'size' with the values 'Big' and 'Small' when the number of employees in the department is equal or more than 4, and less than 4 respectively.

```
SELECT E.department_id, SUM(E.salary) AS total_amt
, Count(E.employee_id) AS number_of_employees,
(CASE WHEN E.department_id >= 4 THEN 'Big' ELSE 'Small' END) AS 'size'
FROM employees E
GROUP BY E.department_id
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

```
/*9. Write a query to display the department id, sum of salary as 'total_amt', number of employees in the department as 'number_of_employees',
and a column called 'size' with the values 'Big' and 'Small' when the number of employees
in the department is equal or more than 4, and less than 4 respectively. */

SELECT E.department_id, SUM(E.salary) AS total_amt ,Count(E.employee_id)
AS number_of_employees,
(CASE WHEN E.department_id >= 4 THEN 'Big' ELSE 'Small' END) AS 'size'
FROM employees E
GROUP BY E.department_id
```

The Results pane shows the following data:

department_id	total_amt	number_of_employees	size
1	4400.00	1	Small
2	19000.00	2	Small
3	24900.00	6	Small
4	6050.00	1	Big
5	41200.00	7	Big
6	28800.00	5	Big
7	10000.00	1	Big
8	57700.00	6	Big
9	56000.00	3	Big
10	51600.00	6	Big
11	20300.00	2	Big

The status bar at the bottom indicates "Query executed successfully." and "11 rows".