

EXERCISE-8

Aggregating Data Using Group Functions

Name: Vedhasree S

Register Number: 240701580

Department: CSE

Determine the validity of the following three statements. Circle either True or False.

1. Group functions work across many rows to produce one result per group.

True/False

Answer: True

2. Group functions include nulls in calculations.

True/False

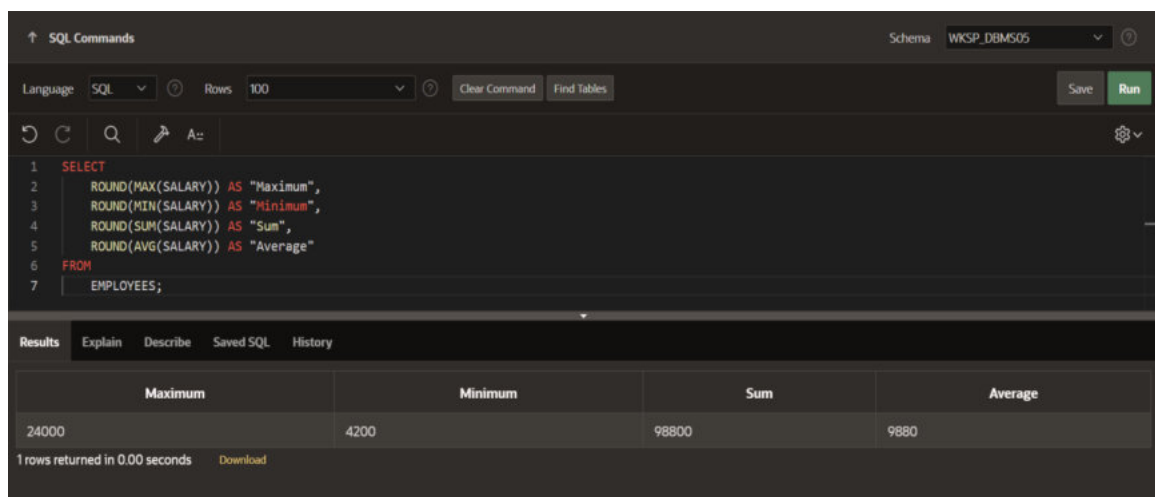
Answer: False

3. The WHERE clause restricts rows prior to inclusion in a group calculation. True/False

Answer: True

The HR department needs the following reports:

4. Find the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.



The screenshot shows a SQL IDE interface. At the top, there's a 'SQL Commands' tab. Below it, the 'Language' is set to 'SQL' and 'Rows' is set to '100'. There are buttons for 'Clear Command', 'Find Tables', 'Save', and 'Run'. The SQL query is as follows:

```
1 SELECT
2   ROUND(MAX(SALARY)) AS "Maximum",
3   ROUND(MIN(SALARY)) AS "Minimum",
4   ROUND(SUM(SALARY)) AS "Sum",
5   ROUND(AVG(SALARY)) AS "Average"
6 FROM
7   EMPLOYEES;
```

Below the query, there's a 'Results' tab. The results are displayed in a table with four columns: 'Maximum', 'Minimum', 'Sum', and 'Average'. The values are 24000, 4200, 98800, and 9880 respectively. At the bottom, it says '1 rows returned in 0.00 seconds' and there's a 'Download' button.

Maximum	Minimum	Sum	Average
24000	4200	98800	9880

5. Modify the above query to display the minimum, maximum, sum, and average salary for each job type.

The screenshot shows the SQL Workshop interface with the following SQL query:

```
1 SELECT
2   JOB_ID AS "Job",
3   ROUND(MIN(SALARY)) AS "Minimum",
4   ROUND(MAX(SALARY)) AS "Maximum",
5   ROUND(SUM(SALARY)) AS "Sum",
6   ROUND(AVG(SALARY)) AS "Average"
7 FROM
8   EMPLOYEES
9 GROUP BY
10  JOB_ID
11 ORDER BY
12  JOB_ID;
```

The results table displays the following data:

Job	Minimum	Maximum	Sum	Average
AD_PRES	24000	24000	24000	24000
AD_VP	17000	17000	34000	17000
IT_PROG	6000	9000	22000	7333
SA_REP	4200	5000	18800	4700

4 rows returned in 0.01 seconds

6. Write a query to display the number of people with the same job. Generalize the query so that the user in the HR department is prompted for a job title.

The screenshot shows the SQL Workshop interface with the following SQL query:

```
1 SELECT
2   JOB_ID AS "Job",
3   COUNT(*) AS "Number of Employees"
4 FROM
5   EMPLOYEES
6 WHERE
7   JOB_ID = 'SA_REP'
8 GROUP BY
9   JOB_ID;
```

The results table displays the following data:

Job	Number of Employees
SA_REP	4

1 rows returned in 0.01 seconds

7. Determine the number of managers without listing them. Label the column Number of Managers. Hint: Use the MANAGER_ID column to determine the number of managers.

The screenshot shows the APEX SQL Workshop interface. The SQL Commands tab is active, displaying the following query:

```
1 SELECT
2   COUNT(DISTINCT MANAGER_ID) AS "Number of Managers"
3 FROM
4   EMPLOYEES
5 WHERE
6   MANAGER_ID IS NOT NULL;
```

The Results tab is selected, showing a single row with the value 4 under the column header "Number of Managers". The status bar indicates "1 rows returned in 0.00 seconds" and provides a "Download" link.

8. Find the difference between the highest and lowest salaries. Label the column DIFFERENCE.

The screenshot shows the APEX SQL Workshop interface. The SQL Commands tab is active, displaying the following query:

```
1 SELECT
2   MAX(SALARY) - MIN(SALARY) AS "DIFFERENCE"
3 FROM
4   EMPLOYEES;
```

The Results tab is selected, showing a single row with the value 19800 under the column header "DIFFERENCE". The status bar indicates "1 rows returned in 0.00 seconds" and provides a "Download" link.

9. Create a report to display the manager number and the salary of the lowest-paid employee for that manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is \$6,000 or less. Sort the output in descending order of salary.

The screenshot shows the APEX SQL Workshop interface. The SQL command is as follows:

```
1 SELECT
2   MANAGER_ID AS "Manager Number",
3   MIN(SALARY) AS "Lowest Salary"
4 FROM
5   EMPLOYEES
6 WHERE
7   MANAGER_ID IS NOT NULL
8 GROUP BY
9   MANAGER_ID
10 HAVING
11   MIN(SALARY) > 6000
12 ORDER BY
13   MIN(SALARY) DESC;
```

The results table shows two rows:

Manager Number	Lowest Salary
100	17000
103	7000

2 rows returned in 0.01 seconds

10. Create a query to display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998. Create appropriate column headings.

The screenshot shows the APEX SQL Workshop interface. The SQL command is as follows:

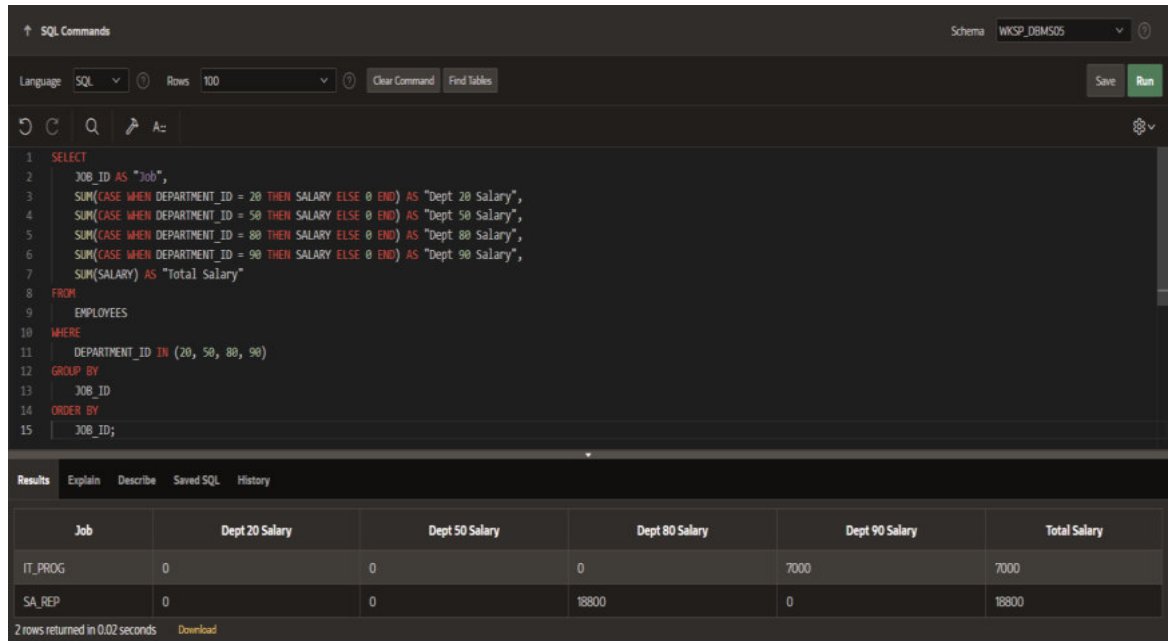
```
1 SELECT
2   COUNT(*) AS "Total Employees",
3   SUM(CASE WHEN TO_CHAR(HIRE_DATE, 'YYYY') = '1995' THEN 1 ELSE 0 END) AS "Hired in 1995",
4   SUM(CASE WHEN TO_CHAR(HIRE_DATE, 'YYYY') = '1996' THEN 1 ELSE 0 END) AS "Hired in 1996",
5   SUM(CASE WHEN TO_CHAR(HIRE_DATE, 'YYYY') = '1997' THEN 1 ELSE 0 END) AS "Hired in 1997",
6   SUM(CASE WHEN TO_CHAR(HIRE_DATE, 'YYYY') = '1998' THEN 1 ELSE 0 END) AS "Hired in 1998"
7 FROM
8   EMPLOYEES;
```

The results table shows one row:

Total Employees	Hired in 1995	Hired in 1996	Hired in 1997	Hired in 1998
10	0	0	0	0

1 rows returned in 0.02 seconds

11. Create a matrix query to display the job, the salary for that job based on department number, and the total salary for that job, for departments 20, 50, 80, and 90, giving each column an appropriate heading.



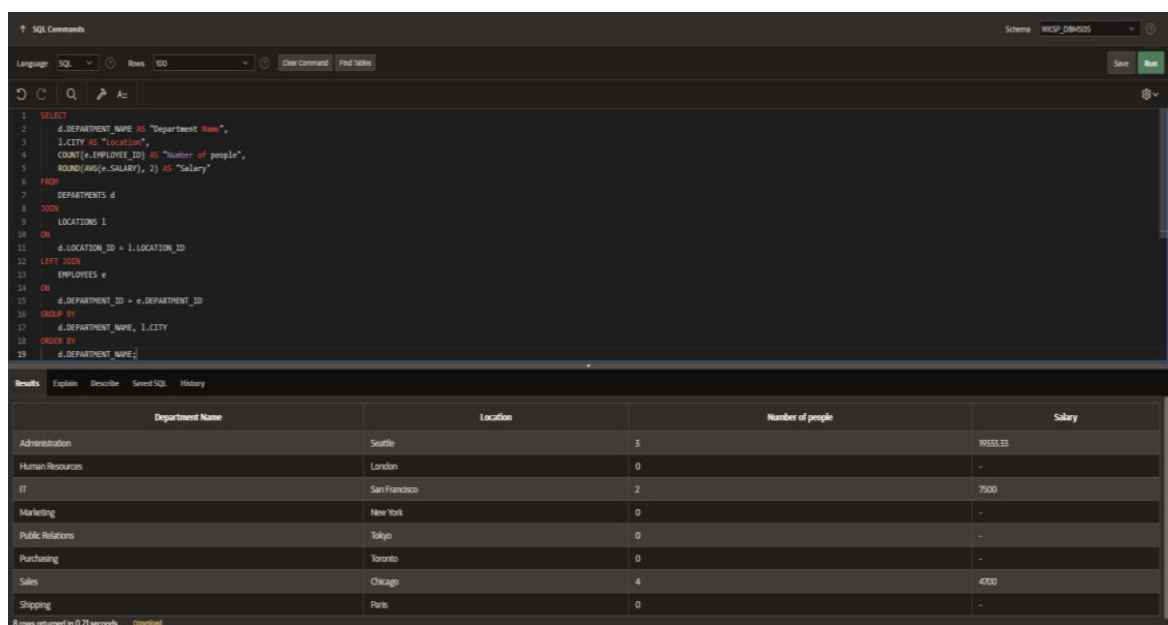
The screenshot shows the SQL Developer interface with a query window. The query is a matrix query that calculates the total salary for each job across four departments (20, 50, 80, 90). The results are displayed in a table with columns for Job, Dept 20 Salary, Dept 50 Salary, Dept 80 Salary, Dept 90 Salary, and Total Salary. The data shows that the IT_PROG job has a total salary of 7000, and the SA_REP job has a total salary of 18800.

```
1 SELECT
2   JOB_ID AS "Job",
3   SUM(CASE WHEN DEPARTMENT_ID = 20 THEN SALARY ELSE 0 END) AS "Dept 20 Salary",
4   SUM(CASE WHEN DEPARTMENT_ID = 50 THEN SALARY ELSE 0 END) AS "Dept 50 Salary",
5   SUM(CASE WHEN DEPARTMENT_ID = 80 THEN SALARY ELSE 0 END) AS "Dept 80 Salary",
6   SUM(CASE WHEN DEPARTMENT_ID = 90 THEN SALARY ELSE 0 END) AS "Dept 90 Salary",
7   SUM(SALARY) AS "Total Salary"
8 FROM
9   EMPLOYEES
10 WHERE
11   DEPARTMENT_ID IN (20, 50, 80, 90)
12 GROUP BY
13   JOB_ID
14 ORDER BY
15   JOB_ID;
```

Job	Dept 20 Salary	Dept 50 Salary	Dept 80 Salary	Dept 90 Salary	Total Salary
IT_PROG	0	0	0	7000	7000
SA_REP	0	0	18800	0	18800

2 rows returned in 0.02 seconds Download

12. Write a query to display each department's name, location, number of employees, and the average salary for all the employees in that department. Label the column name- Location, Number of people, and salary respectively. Round the average salary to two decimal places.



The screenshot shows the SQL Developer interface with a query window. The query is a standard SQL query that retrieves department information, including department name, location, number of employees, and average salary. The results are displayed in a table with columns for Department Name, Location, Number of people, and Salary. The data shows that the Administration department has 3 employees and an average salary of 9331.55, while the Sales department has 4 employees and an average salary of 4500.

```
1 SELECT
2   d.DEPARTMENT_NAME AS "Department Name",
3   l.CITY AS "Location",
4   COUNT(e.EMPLOYEE_ID) AS "Number of people",
5   ROUND(AVG(e.SALARY), 2) AS "Salary"
6 FROM
7   DEPARTMENTS d
8 JOIN
9   LOCATIONS l
10 ON d.LOCATION_ID = l.LOCATION_ID
11 LEFT JOIN
12   EMPLOYEES e
13 ON d.DEPARTMENT_ID = e.DEPARTMENT_ID
14 GROUP BY
15   d.DEPARTMENT_NAME, l.CITY
16 ORDER BY
17   d.DEPARTMENT_NAME;
```

Department Name	Location	Number of people	Salary
Administration	Seattle	3	9331.55
Human Resources	London	0	-
IT	San Francisco	2	7500
Marketing	New York	0	-
Public Relations	Tokyo	0	-
Purchasing	Toronto	0	-
Sales	Chicago	4	4500
Shipping	Paris	0	-

8 rows returned in 0.21 seconds Download