Date: 29/9/25

EXERCISE-8

Aggregating Data Using Group Functions

AIM: To Aggregate data using group functions

Objectives

After the completion of this exercise, the students be will be able to do the following:

- Identify the available group functions
- Describe the use of group functions
- Group data by using the GROUP BY clause
- Include or exclude grouped rows by using the HAVING clause

What Are Group Functions?

Group functions operate on sets of rows to give one result per group

Types of Group Functions

- AVG
- COUNT
- MAX
- MIN
- STDDEV
- SUM
- VARIANCE

Each of the functions accepts an argument. The following table identifies the options that you can use in the syntax:

Function	Description
AVG([DISTINCT|ALL] *n*)	Average value of *n*, ignoring null values
COUNT({*| [DISTINCT|ALL] *expr*})	Number of rows, where *expr* evaluates to something other than null (count all selected rows using *, including duplicates and rows with nulls)
MAX([DISTINCT|ALL] *expr*)	Maximum value of *expr*, ignoring null values
MIN([DISTINCT|ALL] *expr*)	Minimum value of *expr*, ignoring null values
STDDEV([DISTINCT|ALL] *x*)	Standard deviation of *n*, ignoring null values
SUM([DISTINCT|ALL] *n*)	Sum values of *n*, ignoring null values
VARIANCE([DISTINCT|ALL] *x*)	Variance of *n*, ignoring null values

Group Functions: Syntax

SELECT [*column*,] *group_function(column)*, ...
FROM *table*
[WHERE *condition*]
[GROUP BY *column*]
[ORDER BY *column*];

Guidelines for Using Group Functions

- DISTINCT makes the function consider only nonduplicate values; ALL makes it consider every value, including duplicates. The default is ALL and therefore does not need to be specified.

SELECT department_id, AVG(salary) FROM employees GROUP BY department_id
HAVING max(salary)>10000;

Example displays the job ID and total monthly salary for each job that has a total payroll exceeding $13,000. The example excludes sales representatives and sorts the list by the total monthly salary.

SELECT job_id, SUM(salary) PAYROLL FROM employees WHERE job_id NOT LIKE '%REP%'
GROUP BY job_id HAVING SUM(salary) > 13000 ORDER BY SUM(salary);

Nesting Group Functions

Display the maximum average salary:
Group functions can be nested to a depth of two. The slide example displays the maximum average salary.

SELECT MAX(AVG(salary)) FROM employees GROUP BY department_id;

Summary

In this exercise, students should have learned how to:

- Use the group functions COUNT, MAX, MIN, and AVG
- Write queries that use the GROUP BY clause
- Write queries that use the HAVING clause

SELECT *column, group_function*
FROM *table*
[WHERE *condition*]
[GROUP BY *group_by_expression*]
[HAVING *group_condition*]
[ORDER BY *column*];

Find the Solution for the following:

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

2. Group functions include nulls in calculations.
True/False

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

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

select ROUND (MAX(salary)) AS Maximum, ROUND(MIN (salary)) AS Minimum, ROUND (SUM(salary)) AS Sum, ROUND (AVG(salary)) AS Average FROM employees;

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

```
Select job_id, ROUND(MIN(salary)) AS Minimum, ROUND(MAX(salary)) AS Maximum, ROUND(SUM(salary)) AS Sum, ROUND(AVG(salary)) AS Average FROM employees group by job_id;
```

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.

```
Select Count(*) AS Number_of_Employees FROM employees WHERE job_id = (SELECT job_id FROM jobs WHERE job_title = '&job_title');
```

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.*

```
Select Count(DISTINCT manager_id) AS Number_of_Managers FROM employees;
```

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

```
Select (MAX(salary) - MIN(salary)) AS Difference FROM employees;
```

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.

```
Select manager_id, MIN(salary) AS lowest_salary FROM employees WHERE manager_id IS NOT NULL GROUP BY manager_id HAVING MIN(salary) > 6000 ORDER BY lowest_salary DESC;
```

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.

```
Select Count(*) FROM employees WHERE YEAR(hire_date) BETWEEN 1995 AND 1998;
```

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.

```
Select job_id, dept_id, SUM(salary) AS total_salary
FROM employees WHERE dept_id IN(20,50,80,90)
GROUP BY job_id, dept_id ORDER BY job_id, dept_id;
```

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.

```
Select d.dept_name, d.location_id, Count(e.emp_id) AS
'Number of People', ROUND(AVG(e.salary),2) AS 'Average salary'
FROM employees e JOIN departments d ON e.dept_id =
d.dept_id GROUP BY d.dept_name, d.location_id;
```

Evaluation Procedure	Marks awarded
Query(5)	5
Execution (5)	5
Viva(5)	3
Total (15)	13
Faculty Signature	

RESULT:

Thus the data is aggregated using group functions.

Practice Questions

Date Functions

1. For DJs on Demand, display the number of months between the event_date of the Vigil wedding and today's date. Round to the nearest month.

```
Select ROUND (MONTHS_BETWEEN (SYSDATE, event_date))
AS "Months" FROM vigils;
```

2. Display the days between the start of last summer's school vacation break and the day school started this year. Assume 30.5 days per month. Name the output "Days."

```
Select (school_start - vacation_start) AS "Days"
FROM your_table_name;
```

3. Display the days between January 1 and December 31.

```
Select TO_DATE('31-DEC-2025') - TO_DATE
('01-JAN-2025') AS "Total_Days" FROM DUAL;
```

4. Using one statement, round today's date to the nearest month and nearest year and truncate it to the nearest month and nearest year. Use an alias for each column.

```
Select ROUND (Sysdate, 'Month') AS "Rounded_Month",
ROUND (Sysdate, 'Year') AS "Rounded_Year",
TRUNC (Sysdate, 'Month') AS "Truncated_Month",
TRUNC (Sysdate, 'Year') AS "Truncated_Year"
FROM DUAL;
```

5. What is the last day of the month for June 2005? Use an alias for the output.

```
Select LAST_DAY('01-JUN-2005') AS "Last_Day_of_June"
FROM DUAL;
```

6. Display the number of years between the Global Fast Foods employee Bob Miller's birthday and

```
Select TRUNC(MONTHS_BETWEEN (SYSDATE, birth_date)/12)
AS "Age_in_years" FROM emp_employee where name = 'Bob Miller';
```

today. Round to the nearest year.

7. Your next appointment with the dentist is six months from today. On what day will you go to the dentist? Name the output, "Appointment."

```
Select ADD_MONTHS (SYSDATE, 6) AS "Appointment"
FROM DUAL;
```

This returns a date exactly six months from current date.

8. The teacher said you have until the last day of this month to turn in your research paper. What day will this be? Name the output, "Deadline."

```
Select LAST_DAY (SYSDATE) AS "Deadline" FROM DUAL;
```

9. How many months between your birthday this year and January 1 next year?

```
Select MONTHS_BETWEEN (ADD_MONTHS (birth_date,
12), birth_date) FROM DUAL;
```

10. What's the date of the next Friday after your birthday this year? Name the output, "First Friday."

```
Select NEXT_DAY ('your_birthday_date', 'FRIDAY')
AS "First_Friday" FROM DUAL;
```

11. Name a date function that will return a number.

A date function that returns a number is MONTHS_BETWEEN().

12. Name a date function that will return a date.

A date function that returns a date is ADD_MONTHS().

13. Give one example of why it is important for businesses to be able to manipulate date data?

Manipulating date data is crucial for businesses for several reasons. It allows them to track time-sensitive information, generate reports, forecast and plan, manage scheduling.

Conversion Functions

In each of the following exercises, feel free to use labels for the converted column to make the output more readable.

1. List the last names and birthdays of Global Fast Food Employees. Convert the birth dates to character data in the Month DD, YYYY format. Suppress any leading zeros.

```
Select l_name, TO_char(birth_date, 'Month DD, YYYY')
AS "Birth Date" FROM employees;
```

2. Convert January 3, 04, to the default date format 03-Jan-2004.

```
Select TO_DATE('January 3, 04', 'Month DD, YY')
FROM DUAL;
```

Output: 03-JAN-04

3. Format a query from the Global Fast Foods f_promotional_menus table to print out the start_date of promotional code 110 as: The promotion began on the tenth of February 2004.

```
select 'The promotion began on the' || TO_CHAR
(start_date, 'Month YYYY') AS "promotional_Date" FROM
promotional_menus where promotional_code=110;
```

4. Convert today's date to a format such as: "Today is the Twentieth of March, Two Thousand Four"

```
Select 'Today is the' || TO_CHAR(SYSDATE, 'fmddth') ||
'of' || TO_CHAR(SYSDATE, 'Month, fmYYYY') FROM DUAL;
```

5. List the ID, name and salary for all Global Fast Foods employees. Display salary with a $ sign and two decimal places.

```
Select emp_id, l_name, TO_char(salary, '$99,999.00')
AS "Salary" FROM employees;
```

Evaluation Procedure	Marks awarded
Practice Evaluation (5)	5
Viva(5)	3
Total (10)	8
Faculty Signature	