

Lesson 5

Reporting Aggregated Data Using the Group Functions

What You will learn at the end of the Session?



1. Identify the available group functions
2. Describe the use of group functions.
3. Group data by using the GROUP BY clause
4. Include or exclude grouped rows by using the HAVING clause

5-2

What you will learn at the end of this Session?

This lesson further addresses functions. It focuses on obtaining summary information (such as averages) for groups of rows. It discusses how to group rows in a table into smaller sets and how to specify search criteria for groups of rows.



1. Group functions:

- Types and syntax
- Use AVG, SUM, MIN, MAX, COUNT
- Use the DISTINCT keyword within group functions
- NULL values in a group function

2. Grouping rows:

GROUP BY clause
HAVING clause

3. Nesting group functions

What Are Group Functions?

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

EMPLOYEES

	DEPARTMENT_ID	SALARY
1	10	4400
2	20	15000
3	20	6000
4	110	12000
5	110	8300
6	90	24000
7	90	17000
8	90	17000
9	60	9000
10	60	6000

18	80	11000
19	80	8600
20	(null)	7000

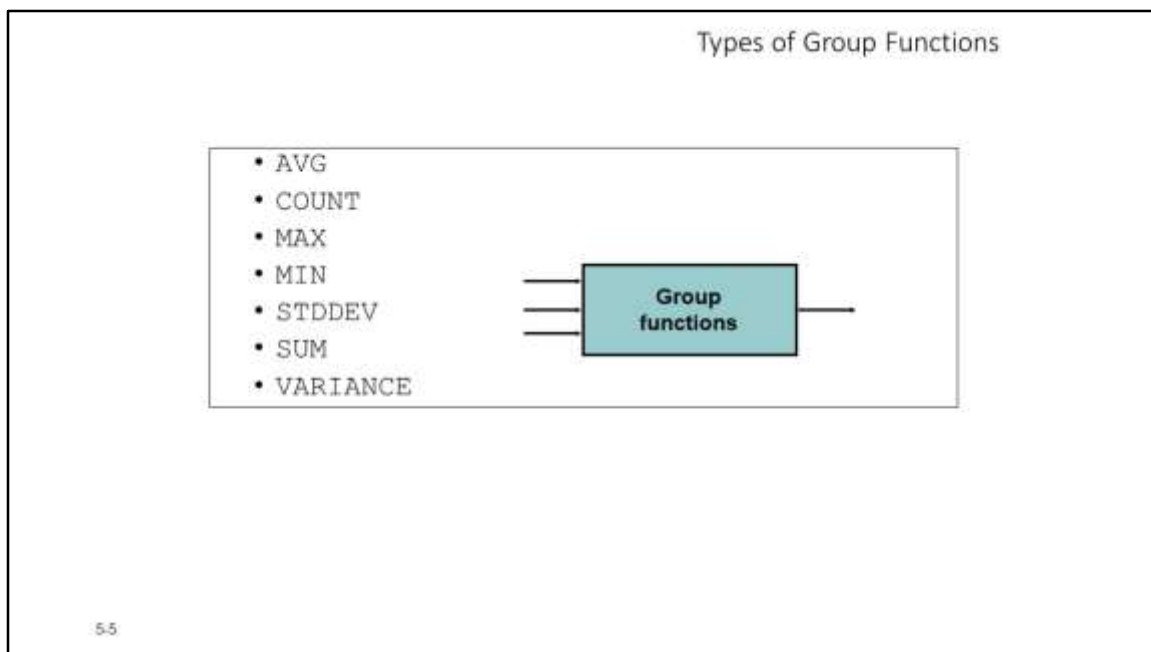
Maximum salary in
EMPLOYEES table

MAX(SALARY)
24000

5-4

What Are Group Functions?

Unlike single-row functions, group functions operate on sets of rows to give one result per group. These sets may comprise the entire table or the table split into groups.



Types of Group Functions

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

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

Group Functions: Syntax

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

5.6

Group Functions: Syntax

The group function is placed after the `SELECT` keyword. You may have multiple group functions separated by commas.

Guidelines for using the 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.

The data types for the functions with an `expr` argument may be `CHAR`, `VARCHAR2`, `NUMBER`, or `DATE`.

All group functions ignore null values. To substitute a value for null values, use the `NVL`, `NVL2`, `COALESCE`, `CASE`, or `DECODE` functions.

Using the AVG and SUM Functions

•You can use AVG and SUM for numeric data.



```
SELECT AVG(order_total), MAX(order_total),  
       MIN(order_total), SUM( order_total)  
FROM orders;
```

	AVERAGE_ORDER_TOTAL	MAX_ORDER_TOTAL	MIN_ORDER_TOTAL	SUM_ORDER_TOTAL
1	44628.44125	295692	5451	3370275.3

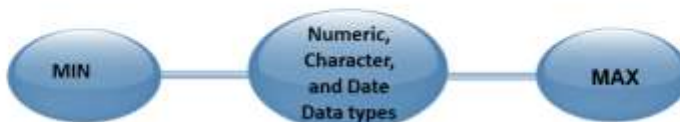
5.7

Using the AVG and SUM Functions

You can use the AVG, SUM, MIN, and MAX functions against the columns that can store numeric data. The example in the slide displays the average, highest, lowest, and sum of the total value of each order.

Using the MIN and MAX Functions

•You can use MIN and MAX for numeric, character, and date data types.



```
SELECT MIN(to_char(order_date,'fmDD Month YYYY'))
AS "Min Order Date",
MAX(to_char(order_date,'fmDD Month YYYY'))
AS "Max Order Date"
FROM orders;
```

	Min Order Date	Max Order Date
1	1 November 1999	9 January 2000

5-8

Using the MIN and MAX Functions

You can use the MAX and MIN functions for numeric, character, and date data types. The example in the slide displays the oldest and the latest orders. The following example displays the employee last name that is first and the employee last name that is last in an alphabetic list of all employees:

```
SELECT MIN(last_name), MAX(last_name)
FROM employees;
```

Note: The AVG, SUM, VARIANCE, and STDDEV functions can be used only with numeric data types. MAX and MIN cannot be used with LOB or LONG data

typ	MIN(LAST_NAME)	MAX(LAST_NAME)
1	Abel	Zlotkey

Using the COUNT Function

•COUNT(*) returns the number of rows in a table:

1

```
SELECT count(*)  
FROM inventories  
WHERE warehouse_id = 8;
```

COUNT(*)
186

•COUNT(expr) returns the number of rows with non-null values for expr:

2

```
SELECT COUNT(sales_rep_id)  
FROM orders  
WHERE order_status <= 3;
```

COUNT(SALES_REP_ID)
19

5-9

Using the COUNT Function

The COUNT function has three formats:

- COUNT (*)
- COUNT (expr)
- COUNT (DISTINCT expr)

COUNT (*) returns the number of rows in a table that satisfy the criteria of the SELECT statement, including duplicate rows and rows containing null values in any of the columns. If a WHERE clause is included in the SELECT statement, COUNT (*) returns the number of rows that satisfy the condition in the WHERE clause.

In contrast, COUNT (expr) returns the number of non-null values that are in the column identified by expr.

COUNT (DISTINCT expr) returns the number of unique, non-null values that are in the column identified by expr.

Examples:

1. The example in the slide displays the number of orders in warehouse 8.
2. The example in the slide displays the number of orders with status <= 3 and whose sales_rep_id is not null.

Using the DISTINCT Keyword

- `COUNT(DISTINCT expr)` returns the number of distinct non-null values of *expr*.
- To display the number of distinct department values in the `EMPLOYEES` table:

```
SELECT COUNT(DISTINCT department_id)  
FROM employees;
```

COUNT(DISTINCT DEPARTMENT_ID)	
1	7

5-10

Using the DISTINCT Keyword

Use the `DISTINCT` keyword to suppress the counting of any duplicate values in a column.

The example in the slide displays the number of distinct department values that are in the `EMPLOYEES` table.

Group Functions and Null Values

•Group functions ignore null values in the column:

1

```
SELECT AVG(commission_pct)
FROM employees;
```

	AVG(COMMISSION_PCT)
1	0.2125

The NVL function forces group functions to include null values:

2

```
SELECT AVG(NVL(commission_pct, 0))
FROM employees;
```

	AVG(NVL(COMMISSION_PCT,0))
1	0.0425

5-11

Group Functions and Null Values

All group functions ignore null values in the column.

However, the NVL function forces group functions to include null values.

Examples:

1. The average is calculated based on *only* those rows in the table in which a valid value is stored in the `COMMISSION_PCT` column. The average is calculated as the total commission that is paid to all employees divided by the number of employees receiving a commission (four).
2. The average is calculated based on *all* rows in the table, regardless of whether null values are stored in the `COMMISSION_PCT` column. The average is calculated as the total commission that is paid to all employees divided by the total number of employees in the company (20).