Lesson 8
Complex Queries
Group Functions

OUTLINE

- Differentiate between single-row and multiple-row functions
- Use the SUM and AVG functions for numeric calculations
- Use the COUNT function to return the number of records containing non-NULL values
- Use COUNT(*) to include records containing NULL values
- Use the MIN and MAX functions with nonnumeric fields

OUTLINE (continued)

- for nesting groupDetermine when to use the GROUP BY clause to group data
- Identify when the HAVING clause should be used
- List the order of precedence for evaluating WHERE, GROUP BY, and HAVING clauses
- State the maximum depth functions
- Nest a group function inside of a single-row function

Group Functions

- Return one result per group of rows processed
- Are also called multiple-row and aggregate functions
- All group functions ignore NULL values except COUNT(*)
- Use DISTINCT to suppress duplicate values
- Built-in aggregate functions
  - **COUNT**, **SUM**, **MAX**, **MIN**, and **AVG**
- Functions can be used in the `SELECT` clause or in a `HAVING` clause.

Group Functions (continued)

- Aggregate functions are functions that take a collection (a set or multiset) of values as input and return a single value.
- SQL offers five built-in aggregate functions:
- These functions operate on the multiset of values of a column of a relation, and return a value.

COUNT	The number of rows containing non-null values
MIN	The minimum attribute value encountered in a given column
MAX	The maximum attribute value encountered in a given column
SUM	The sum of all values for a given column
AVG	The arithmetic mean (average) for a specified column

Group Functions (continued)

- The syntax of the aggregate functions

  AVG([ALL|DISTINCT] expression)

  SUM([ALL|DISTINCT] expression)

  MIN([ALL|DISTINCT] expression)

  MAX([ALL|DISTINCT] expression)

  COUNT([ALL|DISTINCT] expression)

  COUNT(*)

- In general, NULL values are discarded when aggregate functions are applied to a particular column.

Group Functions (continued)

- Return one result per group of rows processed Are also called multiple-row and Group functions
- All group functions ignore NULL values except COUNT(*)
- Use DISTINCT to suppress duplicate values The COUNT, MIN, and MAX functions can be used on values with character, numeric, and date data types

Added Clauses

SELECT <attribute and functionlist>

FROM <table list>

[WHERE <condition>]

[GROUP BY <grouping attribute(s)>]

[HAVING <group condition>]

[ORDER BY <attribute list>];

COUNT Function

- Two purposes
  - Count non-NULL values
  - Count total records, including those with NULL values
- Retrieve the total number of employees in the company.

```
SELECT count(*) AS  no_of_employees
FROM employee
```

AVG Function

- Calculates the average of numeric values in a specified column
- The following query displays the average salary.

```
SELECT AVG(SALARY)
FROM employee;
```

SUM Function

- Calculates total amount stored in a numeric column for a group of rows
- Display the sum of the salaries of all employees.

```
SELECT SUM(SALARY)
FROM employee;
```

COUNT Function (continued)

- Two purposes
  - Count non-NULL values
  - Count total records, including those with NULL values
- Retrieve the total number of employees in the company.

```
SELECT count(*) AS  no_of_employees
FROM employee
```

COUNT Function (continued)

- Include column name in argument to count number of occurrences.
- Count the number of distinct salary values in the database.

```
SELECT count(DISTINCT salary) AS no_of_distinct_salary
FROM   employee;
```

MAX Function

- Returns largest value
- The following query illustrates the use of the Max function.

```
SELECT MAX(SALARY)
FROM employee;
```

MIN Function

- Returns the smallest value
- The following query illustrates the use of the MIN function.

```
SELECT MIN(SALARY)
FROM employee;
```

Group Functions (continued)

- Find the sum of the salaries of all employees, the maximum salary, the minimum salary, and the average salary.

```
SELECT SUM(salary)AS sum_salary
, MAX(salary) AS max_salary
, MIN(salary) AS min_salary
, AVG(salary) AS avg_salary
FROM employee;
```

Data Types

- The COUNT, MIN, and MAX functions can be used on values with character, numeric, and date datatypes

Grouping Data

- GROUP BY clause
  - Used to group data
  - Must be used for any individual column in the SELECT clause with a group function
  - Cannot reference column aliases

GROUP BY Example

- The following query displays the minimum and maximum salary of all departments.

```
SELECT MIN( salary ) , MAX( salary )
FROM employee
GROUP BY dno;
```

GROUP BY Example (continued)

- For each project, retrieve the project number, the project name, and the number of employees from department 5 who work on the project.

```
SELECT p.pnumber, p.pname, COUNT(*)
FROM   project p, works_on a, employee e
WHERE  p.pnumber = a.pno
AND e.ssn = a.essn AND p.dnum=5
GROUP BY  p.pnumber, p.pname;
```

GROUP BY Example (continued)

- For each project, list the project name and the total hours per week (by all employees) spent on that project.

```
SELECT p.pname, SUM(hours)
FROM project JOIN works_on a
ON (p.pnumber = a.pno)
GROUP BY p.pname;
```

Restricting Aggregated Output

- HAVING clause serves as the WHERE clause for grouped data

```
SELECT p.pnumber, p.pname, count(*) AS
no_of_employees
FROM   project p
 JOIN works_on a ON  (p.pnumber = a.pno)
GROUP BY p.pnumber, p.pname
HAVING count(*)>2
```

Restricting Aggregated Output (continued)

- When included in the same SELECT statement, the clauses are evaluated in the order of:
  - WHERE
  - GROUP BY
  - HAVING

Summary

- The AVG and SUM functions are used only with numeric fields
- The COUNT, MAX, and MIN functions can be applied to any datatype
- The AVG, SUM, MAX, and MIN functions all ignore NULL values
- By default, the AVG, SUM, MAX, MIN, and COUNT functions include duplicate values
- The GROUP BY clause is used to divide table data into groups

Summary (continued)

- If a SELECT clause contains both an individual field name and a group function, the field name must also be included in a GROUP BY clause
- The HAVING clause is used to restrict groups in a group function
- Group functions can be nested to a depth of only two. The inner function is always performed first, using the specified grouping. The results of the inner function are used as input for the outer function.