SQL Commands

Presented by:
Virendra Kumar Meghwal
Research Scholar
CSE, MNIT, Jaipur

SELECT

- SELECT statement is used to select (show or get) data from a database

- SELECT *column1*, *column2*, …
  FROM *table_name*;
- SELECT * FROM *table_name*;

DISTINCT

- The SELECT DISTINCT statement is used to return only distinct (different) values.

- SELECT DISTINCT *column1*, *column2*, ...
  FROM *table_name*;

WHERE clause

- WHERE clause is used to filter records.

- SELECT *column1*, *column2*, …
  FROM *table_name*
  WHERE *condition*;

AND, OR and NOT operators

- WHERE clause can be combined with AND, OR, and NOT operators.
- ND and OR operators are used to filter records based on more than one condition:
  - The AND operator displays a record if all the conditions separated by AND are TRUE.
  - The OR operator displays a record if any of the conditions separated by OR is TRUE.
- The NOT operator displays a record if the condition(s) is NOT TRUE.

AND, OR and NOT operators

- SELECT *column1*, *column2*, …
  FROM *table_name*
  WHERE *condition1* **AND** *condition2* **AND** *condition3* …;
- SELECT *column1*, *column2*, …
  FROM *table_name*
  WHERE *condition1* **OR** *condition2* **OR** *condition3* …;
- SELECT *column1*, *column2*, …
  FROM *table_name*
  WHERE **NOT** *condition*;

ORDER BY keyword

- The ORDER BY keyword is used to sort the result-set in ascending or descending order.
- The ORDER BY keyword sorts the records in **ascending order by default**.
- To sort the records in descending order, use the DESC keyword.
- To sort the records in ascending order, use the ASC keyword.
- SELECT *column1*, *column2*, ...
  FROM *table_name*
  ORDER BY *column1*, *column2*, ... **ASC**|**DESC**;

INSERT INTO statement

- INSERT INTO statement is used to insert new records in a table.
- Two ways to use:
  - INSERT INTO *table_name* (*column1*, *column2*, *column3*, …)
    VALUES (*value1*, *value2*, *value3*, …);
  - INSERT INTO *table_name*
    VALUES (*value1*, *value2*, *value3*, …);

UPDATE statement

- UPDATE statement is used to modify the existing records in a table.
- UPDATE *table_name*
  SET *column1* = *value1*, *column2* = *value2*, …
  WHERE *condition*;

DELETE statement

- DELETE statement is used to delete existing records in a table.
- DELETE FROM *table_name* WHERE *condition*;

- Delete all records from table:
  - DELETE FROM *table_name*;

MIN() and MAX() functions

- MIN() function returns the smallest value of the selected column.
  - SELECT **MIN(*column_name*)** FROM *table_name* WHERE *condition*;

- MAX() function returns the largest value of the selected column.
  - SELECT **MAX(*column_name*)** FROM *table_name* WHERE *condition*;

COUNT(), AVG() and SUM() functions

- COUNT() function returns the number of rows that matches a specified criterion.
  - SELECT **COUNT(*column_name*)** FROM *table_name* WHERE *condition*;
- AVG() function returns the average value of a numeric column.
  - SELECT **AVG(*column_name*)** FROM *table_name* WHERE *condition*;
- SUM() function returns the total sum of a numeric column.
  - SELECT **SUM(*column_name*)** FROM *table_name* WHERE *condition*;

LIKE operator

- The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.
  - There are two wildcards often used in conjunction with the LIKE operator:
    - % - The percent sign represents zero, one, or multiple characters
    - _ - The underscore represents a single character
- SELECT *column1*, *column2*, ...
  FROM *table_name*
  WHERE *columnN* LIKE *pattern*;

IN operator

- The IN operator allows you to specify multiple values in a WHERE clause.
- The IN operator is a shorthand for multiple OR conditions.
- SELECT *column_name(s)* FROM *table_name*
  WHERE *column_name* IN (*value1*, *value2*, …);
- SELECT *column_name(s)* FROM *table_name*
  WHERE *column_name* IN (*SELECT STATEMENT*);

BETWEEN operator

- The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.
- The BETWEEN operator is inclusive: **begin and end values are included.**
- SELECT *column_name(s)* FROM *table_name*
  WHERE *column_name* BETWEEN *value1* AND *value2;*

Aliases

- SQL aliases are used to give a table, or a column in a table, a temporary name.
- Aliases are often used to make column names more readable.
- An alias only exists for the duration of the query.
- Alias Column:
  - SELECT ***column_name*** **AS** ***alias_name***
    FROM *table_name;*
- Alias Table:
  - SELECT *column_name(s)*
    FROM ***table_name*** **AS** ***alias_name****;*