Demonstration 2

Operators [Arithmetic, Logical, Set Theory]

Restricting and Sorting Data

- Restrict the rows returned by using the WHERE clause

SELECT column_name(s)
FROM table_name
WHERE column_name operator value

Operators : =, <, >, >=, <=, <>, BETWEEN, LIKE, IN

SELECT column_name(s)
FROM table_name
WHERE column_name1 = 'XYZ'

Logical Operators

Allow you to combine two or more conditions in the WHERE clause
AND, OR, NOT, IN, BETWEEN

AND & OR Operators

- The AND operator displays a record if both the first condition and the second condition is true.
- The OR operator displays a record if either the first condition or the second condition is true.

- SELECT * FROM Table WHERE attribute1=value1 AND attribute2=value2;
- SELECT * FROM Table WHERE attribute1=value1 OR attribute2=value2;
- SELECT * FROM Table WHERE attribute1=value1 AND (attribute2=value2 OR attribute3=value3);

LIKE Operator

- The LIKE operator is used to search for a specified pattern in a column.

- SELECT column_name(s)
  FROM table_name
  WHERE column_name LIKE pattern

- Pattern → '%pattern' or 'pattern%' or '%pattern%'

The IN Operator

- The IN operator allows you to specify multiple values in a WHERE clause.

- SELECT column_name(s)
  FROM table_name
  WHERE column_name IN (value1,value2,...);

The BETWEEN Operator

- The BETWEEN operator selects a range of data between two values. The values can be numbers, text, or dates.

- SELECT column_name(s)
  FROM table_name
  WHERE column_name
  BETWEEN value1 AND value2;

The UNION operator

- The UNION operator is used to combine the result-set of two or more SELECT statements.
- **Note:**
  - each SELECT statement within the UNION must have the same number of columns.
  - the columns must also have similar data types.
  - also, the columns in each SELECT statement must be in the same order.
- SELECT column_name(s) FROM table_name1
  UNION
  SELECT column_name(s) FROM table_name2

- The UNION operator selects only distinct values by default. To allow duplicate values, use UNION ALL.

The INTERSECTION operator

- SELECT column_name(s) FROM table_name1
  INTERSECTION
  SELECT column_name(s) FROM table_name2

Difference

- SELECT column_name(s) FROM table_name1
  MINUS
  SELECT column_name(s) FROM table_name2

The ORDER BY Keyword

- The ORDER BY keyword is used to sort the result-set by a specified column.

- SELECT column_name(s)
  FROM table_name
  ORDER BY column_name(s) ASC|DESC

Joining Multiple Tables

- A Join Combines data from multiple tables using foreign key references
- Syntax

```
SELECT column1, column2, …
FROM table1, table2
WHERE table1.joincolumn = table2.joincolumn
AND search_condition(s);
```

SQL Alias

- **SQL Alias Syntax for Tables**
- SELECT column_name(s) FROM table_name
  AS alias_name

- **SQL Alias Syntax for Columns**
- SELECT column_name AS alias_name
  FROM table_name

SQL Aggregate Functions

- SQL aggregate functions return a single value, calculated from values in a column.

- Useful aggregate functions:
  - AVG() - Returns the average value
  - COUNT() - Returns the number of rows
  - MAX() - Returns the largest value
  - MIN() - Returns the smallest value
  - SUM() - Returns the sum

The SUM() and AVG() Function

- The **SUM()** function returns the total sum of a numeric column.
- SELECT SUM(column_name) FROM table_name
- The **AVG()** function returns the average value of a numeric column.
- SELECT AVG(column_name) <AS avg_Col_name> FROM table_name

The COUNT() Function

- **The COUNT(column_name) function returns the number of values (NULL values will not be counted) of the specified column**
- SELECT COUNT(column_name) FROM table_name

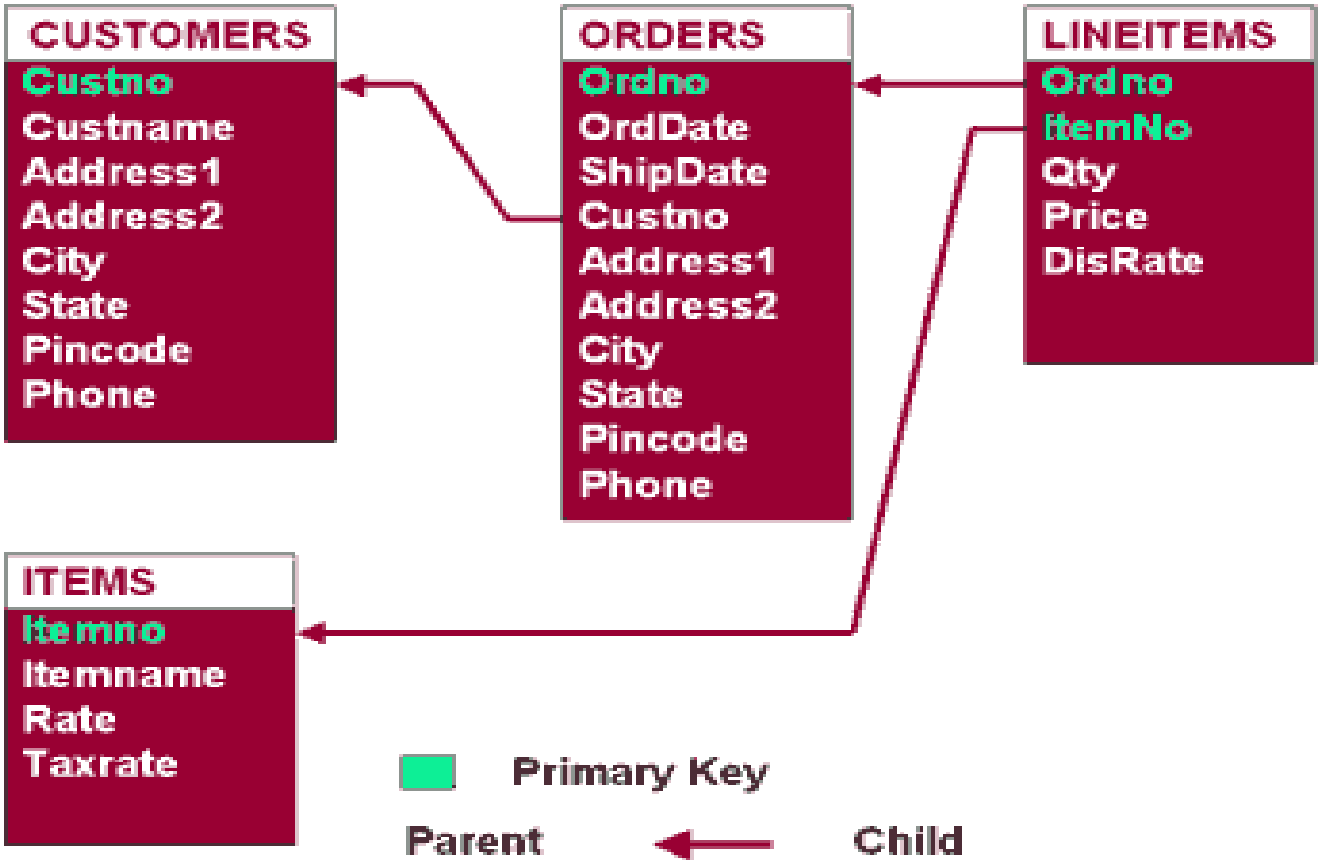
- **The COUNT(*) function returns the number of records in a table**
- SELECT COUNT(*) FROM table_name

- **The COUNT(DISTINCT column_name) function returns the number of distinct values of the specified column**
- SELECT COUNT(DISTINCT column_name) FROM table_name

The MAX() and MIN() Function

- The **MAX()** function returns the largest value of the selected column.
- SELECT MAX(column_name) FROM table_name
- The **MIN()** function returns the smallest value of the selected column.
- SELECT MIN(column_name) FROM table_name

Purchase Order Example



Queries

- Display details of orders that were placed in the month of june 2003.
- Display customer names who stays in karnataka state
- Display details of items where itemname contains letter 'o' twice
- Display all the orders that are placed in the current month
- Display orderno,orderdate,custno,name for all the orders where the order contains order for itemno 5.
- Display itemno,name,orderno,custname and amount.
- Display orderno,custname,orderdate,no.of days between shipdate and orderdate for orders that have been shipped
- Display orderno,orderdate,custno,custname for all the orders where the order contains order for itemno 5.
- Display the details of items where itemname contains letter o or m

Queries Contd..

- Display the details of orders that placed in the last 20 days and delivered.
- Change the rate of items in order 1003 so that 10% discount is given to all items.
- Display the items where itemname contains more than 10 characters.
- Display itemno,itemname in upper case for all items where the letter ‘m’ is existing in any case.
- Display the orders that are placed in the current month.
- Display how many orders are still pending.

The GROUP BY Statement

- The GROUP BY statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.

- SELECT column_name, aggregate_function(column_name)
  FROM table_name
  WHERE column_name operator value
  GROUP BY column_name

The HAVING Clause

- The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.

- SELECT column_name, aggregate_function(column_name)
  FROM table_name
  WHERE column_name operator value
  GROUP BY column_name
  HAVING aggregate_function(column_name) operator value