# SQL

Made of Tables

Row – Records and columns – fields

SQL Statements

* SELECT *column1*,*column2, ...*  
  FROM *table\_name*;
* SELECT DISTINCT column1, column2, ...

FROM table\_name;

* MS Access specific: SELECT Count(\*) AS DistinctCountries  
  FROM (SELECT DISTINCT Country FROM Customers);
* SELECT COUNT(DISTINCT Country) FROM Customers;
* SELECT DISTINCT Country FROM Customers;
* SELECT column1, column2, ...  
  FROM table\_name  
  WHERE condition;
* SELECT column1, column2, ...  
  FROM table\_name  
  ORDER BY column1, column2, ... ASC|DESC;
* SELECT column1, column2, ...  
  FROM table\_name  
  WHERE condition1 OR condition2 OR condition3 ...;
* SELECT column1, column2, ...  
  FROM table\_name  
  WHERE NOT condition;
* INSERT INTO table\_name (column1, column2, column3, ...)  
  VALUES (value1, value2, value3, ...);
* SELECT column\_namesFROM table\_name  
  WHERE column\_name IS NOT NULL;
* UPDATE table\_name  
  SET column1 = value1, column2 = value2, ...  
  WHERE condition;
* DELETE FROM table\_name WHERE condition;
* DROP TABLE table\_name;
* **SQL Server / MS Access Syntax:**
* SELECT TOP number|*percent* column\_name(s)  
  FROM table\_nameWHERE condition;
* **MySQL Syntax:**
* SELECT column\_name(s)  
  FROM table\_nameWHERE condition  
  LIMIT number;
* SELECT MAX(column\_name)  
  FROM table\_name  
  WHERE condition;
* SELECT COUNT(column\_name)  
  FROM table\_name  
  WHERE condition;
* SELECT COUNT(\*) AS [Number of records], CategoryID  
  FROM Products  
  GROUP BY CategoryID;
* SELECT SUM(column\_name)  
  FROM table\_name  
  WHERE condition;
* SELECT column1, column2, ...  
  FROM table\_name  
  WHERE columnN LIKE pattern;

There are two wildcards often used in conjunction with the LIKE operator:

* The percent sign % represents zero, one, or multiple characters
* The underscore sign \_ represents one, single character
* A wildcard character is used to substitute one or more characters in a string.
* SELECT column\_name(s)  
  FROM table\_name  
  WHERE column\_name IN (value1, value2, ...);
* SELECT \* FROM Orders  
  WHERE OrderDate BETWEEN #07/01/1996# AND #07/31/1996#;
* SELECT column\_name AS alias\_name  
  FROM table\_name;
* SELECT column\_name(s)  
  FROM table\_name AS alias\_name;
* If you want your alias to contain one or more spaces, like "My Great Products", surround your alias with square brackets or double quotes.
* SELECT CustomerName, Address + ', ' + PostalCode + ' ' + City + ', ' + Country AS Address  
  FROM Customers;

### **MySQL Example**

* SELECT CustomerName, CONCAT(Address,', ',PostalCode,', ',City,', ',Country) AS Address  
  FROM Customers;
* SELECT column\_name(s)  
  FROM table1  
  INNER JOIN table2ON table1.column\_name = table2.column\_name;
* INNER JOIN OR JOIN
* LEFT JOIN, RIGHT JOIN, FULL OUTER JOIN OR FULL JOIN
* SELF JOIN
* SELECT column\_name(s)  
  FROM table1 T1, table1 T2  
  WHERE condition;

## **The SQL UNION Operator**

The UNION operator is used to combine the result-set of two or more SELECT statements.

* Every SELECT statement within UNION must have the same number of columns
* The columns must also have similar data types
* The columns in every SELECT statement must also be in the same order
* SELECT column\_name(s) FROM table1  
  UNION  
  SELECT column\_name(s) FROM table2;
* UNION ALL – ALLOWS DUPLICATES
* UNION – DISTINCT VALUE
* SELECT column\_name(s)  
  FROM table\_name  
  WHERE condition  
  GROUP BY column\_name(s)ORDER BY column\_name(s);
* SELECT column\_name(s)  
  FROM table\_name  
  WHERE condition  
  GROUP BY column\_name(s)HAVING conditionORDER BY column\_name(s);
* SELECT column\_name(s)  
  FROM table\_name  
  WHERE EXISTS  
  (SELECT column\_name FROM table\_name WHERE condition);
* The ANY and ALL operators allow you to perform a comparison between a single column value and a range of other values.
* SELECT column\_name(s)  
  FROM table\_name  
  WHERE column\_name operator ANY/ALL  
    (SELECT column\_name  FROM table\_name  WHERE condition);
* SELECT ALL column\_name(s)  
  FROM table\_name  
  WHERE condition;
* SELECT column1, column2, column3, ...  
  INTO newtable [IN externaldb]  
  FROM oldtableWHERE condition;
* INSERT INTO table2 (column1, column2, column3, ...)  
  SELECT column1, column2, column3, ...  
  FROM table1  
  WHERE condition;
* CASE  
      WHEN condition1 THEN result1  
      WHEN condition2 THEN result2  
      WHEN conditionN THEN resultN  
      ELSE result  
  END;
* CREATE PROCEDURE procedure\_name  
  AS  
  sql\_statement  
  GO;
* EXEC procedure\_name;
* Comments --SINGLE LINE COMMENTS
* /\*Select all the columns  
  of all the records  
  in the Customers table:\*/