**SET Operators, Union, Union ALL**

-----------------------------------------------------------------------------------------------------------------------------

/\*

1. UNION (combines two data sets excluding duplicates - unique records from both the datasets)

2. UNION ALL (combines two data sets including duplicates)

3. INTERSECT (COMMON records)

4. MINUS (A-B)

\*/

-----------------------------------------------------------------------------------------------------------------------------

-- SQL UNION Operator

-----------------------------------------------------------------------------------------------------------------------------

/\*

SQL UNION Operator

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

\*/

-- UNION Syntax

SELECT column\_name(s) FROM table1

UNION

SELECT column\_name(s) FROM table2

UNION

SELECT column\_name(s) FROM table3;

-- UNION ALL Syntax

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

SELECT column\_name(s) FROM table1

UNION ALL

SELECT column\_name(s) FROM table2;

/\*

Note: The column names in the result-set are usually equal to the column names in the first SELECT statement.

Below is a selection from the "Customers" table:

CustomerID CustomerName ContactName Address City PostalCode Country

1

Alfreds Futterkiste Maria Anders Obere Str. 57 Berlin 12209 Germany

2 Ana Trujillo Emparedados y helados Ana Trujillo Avda. de la Constitución 2222 México D.F. 05021 Mexico

3 Antonio Moreno Taquería Antonio Moreno Mataderos 2312 México D.F. 05023 Mexico

And a selection from the "Suppliers" table:

SupplierID SupplierName ContactName Address City PostalCode Country

1 Exotic Liquid Charlotte Cooper 49 Gilbert St. London EC1 4SD UK

2 New Orleans Cajun Delights Shelley Burke P.O. Box 78934 New Orleans 70117 USA

3 Grandma Kelly's Homestead Regina Murphy 707 Oxford Rd. Ann Arbor 48104 USA

SQL UNION Example

The following SQL statement returns the cities (only distinct values) from both the "Customers" and the "Suppliers" table:

\*/

SELECT City FROM Customers

UNION

SELECT City FROM Suppliers

ORDER BY City;

-- Note: If some customers or suppliers have the same city, each city will only be listed once, because UNION selects only distinct values. Use UNION ALL to also select duplicate values!

-- SQL UNION ALL Example

-- The following SQL statement returns the cities (duplicate values also) from both the "Customers" and the "Suppliers" table:

SELECT City FROM Customers

UNION ALL

SELECT City FROM Suppliers

ORDER BY City;

-- SQL UNION With WHERE

-- The following SQL statement returns the German cities (only distinct values) from both the "Customers" and the "Suppliers" table:

SELECT City, Country FROM Customers

WHERE Country='Germany'

UNION

SELECT City, Country FROM Suppliers

WHERE Country='Germany'

ORDER BY City;

-- SQL UNION ALL With WHERE

-- The following SQL statement returns the German cities (duplicate values also) from both the "Customers" and the "Suppliers" table:

SELECT City, Country FROM Customers

WHERE Country='Germany'

UNION ALL

SELECT City, Country FROM Suppliers

WHERE Country='Germany'

ORDER BY City;

-- Another UNION Example

-- The following SQL statement lists all customers and suppliers:

SELECT 'Customer' AS Type, ContactName, City, Country

FROM Customers

UNION

SELECT 'Supplier', ContactName, City, Country

FROM Suppliers;

SELECT 1 as id,'Customer' AS Type, ContactName, City, Country

FROM Customers

UNION ALL

SELECT 2,'Supplier', ContactName, City, Country

FROM Suppliers;

-- INTESECT

SELECT City, Country

FROM Customers

INTERSECT

SELECT City, Country

FROM Suppliers;

-- MINUS

SELECT City

FROM Customers

MINUS

SELECT City

FROM Suppliers;

SELECT City

FROM Suppliers

MINUS

SELECT City

FROM CUSTOMERS;