

# JOIN

Course: INFO6210 Data Management and Database Design  
Week: x  
Instructor: Mutsalklisana

## Objectives

- Learn to use basic functions in SQL statement
  - Include functions that frequently appeared
  - Comparison, control flow, and cast functions
  - String, numeric, and date/time functions
  - Aggregate functions (e.g., GROUP BY)
  - System-related functions
- Explain the purpose of each of these functions, describe the results you can expect when a statement includes a function and provide examples that demonstrate how to use each function
  - Des not cover every functions but many will be use in the daily SQL statements creation

2

## JOIN

- SQL JOIN statements are used to combine rows from two or more tables
- Types of JOIN statements
  - INNER JOIN (Simple JOIN)
  - LEFT JOIN
  - RIGHT JOIN
  - FULL JOIN
  - CROSS JOIN

3

## SQL Code: Create Employee & Dept. Tables

```
CREATE TABLE department
(
  DepartmentID INT,
  DepartmentName VARCHAR(20)
);

CREATE TABLE employee
(
  LastName VARCHAR(20),
  DepartmentID INT
);

INSERT INTO department VALUES (31, 'Sales');
INSERT INTO department VALUES (33, 'Engineering');
INSERT INTO department VALUES (34, 'Clerical');
INSERT INTO department VALUES (35, 'Marketing');

INSERT INTO employee VALUES ('Rafferty', 31);
INSERT INTO employee VALUES ('Jones', 33);
```

4

## SQL Code Output: Employee & Dept Tables

Employee table		Department table	
LastName	DepartmentID	DepartmentID	DepartmentName
Rafferty	31	31	Sales
Jones	33	33	Engineering
Heisenberg	33	34	Cleaning
Robinson	34	35	Marketing
Smith	34		

Note: In the Employee table above, the employee "Williams" has not been assigned to any department yet. Also, note that no employees are assigned to the "Marketing" department.

5

## INNER JOIN

- Returns all rows when there is at least one match in BOTH tables
- INNER JOIN = JOIN

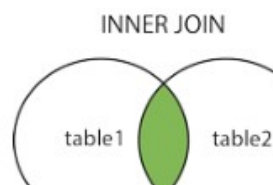
### SQL INNER JOIN Syntax

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name=table2.column_name;
```

or:

```
SELECT column_name(s)
FROM table1
```

SOURCE: <http://www.w3schools.com>



6

## Example #1 of INNER JOIN

Let's look at a selection from the "Orders" table:

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20

Then, have a look at a selection from the "Customers" table:

CustomerID	CustomerName	ContactName
1	Alfreds Futterkiste	Maria Anders
2	Ana Trujillo Emparedados y helados	Ana Trujillo

SOURCE: <http://www.w3schools.com>

7

## Example #1 of INNER JOIN – Cont'd

- SQL Code:

```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
INNER JOIN Customers
```

it will produce something like this:

OrderID	CustomerName
10308	Ana Trujillo Emparedados y helados
10365	Antonio Moreno Taquería
10383	Around the Horn
10355	Around the Horn

SOURCE: <http://www.w3schools.com>

8

## Example #2 of INNER JOIN

Below is a selection from the "Customers" table:

CustomerID	CustomerName	ContactName	Address	City	P
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	1
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	0
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	0

And a selection from the "Orders" table:

OrderID	CustomerID	EmployeeID	OrderDate
10308	2	7	1996-09-18

SOURCE: <http://www.w3schools.com>

9

## Example #2 of INNER JOIN – Cont'd

- SQL Code:

**Note:** The INNER JOIN keyword selects all rows from both tables as long as there is a match between the columns. If there are rows in the "Customers" table that do not have matches in "Orders", these customers will NOT be listed.

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
INNER JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
```

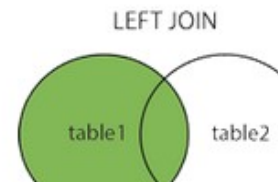
CustomerName
Ana Trujillo Emparedados y helados
Antonio Moreno Taquería
Around the Horn
Around the Horn
Berglunds snabbköp
Berglunds snabbköp
Berglunds snabbköp
Blondel père et fils

SOURCE: <http://www.w3schools.com>

10

## LEFT JOIN

- Return all rows from the left table, and the matched rows from the right table
- The result is NULL in the right side v match.
- LEFT JOIN = LEFT OUTER JOIN



### SQL LEFT JOIN Syntax

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name=table2.column_name;
```

or:

```
SELECT column_name(s)
FROM table1
```

SOURCE: <http://www.w3schools.com>

11

## Example #1 of LEFT JOIN

Below is a selection from the "Customers" table:

CustomerID	CustomerName	ContactName	Address	City	P
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	1
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	0
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	0

And a selection from the "Orders" table:

OrderID	CustomerID	EmployeeID	OrderDate
10308	2	7	1996-09-18

SOURCE: <http://www.w3schools.com>

12

## Example #1 of LEFT JOIN– Cont'd

- SQL Code: **Note:** The LEFT JOIN keyword returns all the rows from the left table (Customers), even if there are no matches in the right table (Orders).

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
LEFT JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
```

CustomerName
Alfreds Futterkiste
Ana Trujillo Emparedados y helados
Antonio Moreno Taquería
Around the Horn
Around the Horn
Berglunds snabbköp
Berglunds snabbköp
Berglunds snabbköp
Blauer See Delikatessen

SOURCE: <http://www.w3schools.com>

13

## Example #2 of LEFT JOIN using Employee & Department Tables

- SQL Code:

```
SELECT *
FROM employee LEFT OUTER JOIN department
ON employee.DepartmentID = department.DepartmentID;
```

Employee.LastName	Employee.DepartmentID	Department.DepartmentName	Depar
Jones	33	Engineering	
Rafferty	31	Sales	
Robinson	34	Clerical	
Smith	34	Clerical	

SOURCE: [http://en.wikipedia.org/wiki/Join\\_%28SQL%29](http://en.wikipedia.org/wiki/Join_%28SQL%29)

14

## Alt LEFT OUTER JOIN w/ SELECT & UNION ALL

```
SELECT employee.LastName, employee.DepartmentID, department.Dep
FROM employee
LEFT OUTER JOIN department ON employee.DepartmentID = departmen
```

can also be written as

```
SELECT employee.LastName, employee.DepartmentID, department.Dep
FROM employee
INNER JOIN department ON employee.DepartmentID = department.Dep
```

UNION ALL

```
SELECT employee.LastName, employee.DepartmentID, CAST(NULL AS V
FROM employee
```

15

## RIGHT JOIN

- Return all rows from the right table, and the matched rows from the left table
- The result is NULL in the left side when there is no match.
- RIGHT JOIN = RIGHT OUTER JOIN

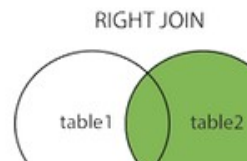
### SQL RIGHT JOIN Syntax

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name=table2.column_name;
```

or:

```
SELECT column_name(s)
FROM table1
```

SOURCE: <http://www.w3schools.com>



16



## Example #1 of RIGHT JOIN

Below is a selection from the "Orders" table:

OrderID	CustomerID	EmployeeID	OrderDate
10308	2	7	1996-09-18
10309	37	3	1996-09-19
10310	77	8	1996-09-20

And a selection from the "Employees" table:

EmployeeID	LastName	FirstName	BirthDate	Photo	Notes
1	Davolio	Nancy	12/8/1968	EmpID1.pic	Education includes a BA
2	Fuller	Andrew	2/19/1952	EmpID2.pic	Andrew received his BTS

SOURCE: <http://www.w3schools.com>

17

## Example #1 of RIGHT JOIN– Cont'd

- **SQL Code:** **Note:** The RIGHT JOIN keyword returns all the rows from the right table (Employees), even if there are no matches in the left table (Orders).

```
SELECT Orders.OrderID, Employees.FirstName
FROM Orders
RIGHT JOIN Employees
ON Orders.EmployeeID=Employees.EmployeeID
```

OrderID	FirstName
	Adam
10248	Steven
10249	Michael
10250	Margaret
10251	Janet
10252	Margaret
10253	Janet
10254	Steven

SOURCE: <http://www.w3schools.com>

18

## Example #2 of RIGHT JOIN using Employee & Department Tables

- SQL Code:

```
SELECT *
FROM employee RIGHT OUTER JOIN department
ON employee.DepartmentID = department.DepartmentID;
```

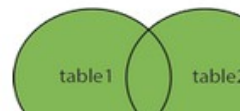
Employee.LastName	Employee.DepartmentID	Department.DepartmentName	Deepar
Smith	34	Clerical	
Jones	33	Engineering	
Robinson	34	Clerical	
Heisenberg	33	Engineering	

SOURCE: [http://en.wikipedia.org/wiki/Join\\_%28SQL%29](http://en.wikipedia.org/wiki/Join_%28SQL%29)

19

## FULL JOIN

FULL OUTER JOIN



- Returns all rows from the left table (table1) and from the right table (table2).
- Combines the result of both LEFT and RIGHT joins.

SQL FULL OUTER JOIN Syntax

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
```

SOURCE: <http://www.w3schools.com>

20

## Example #1 of FULL OUTER JOIN

Below is a selection from the "Customers" table:

CustomerID	CustomerName	ContactName	Address	City	P
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	1
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	0
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	0

And a selection from the "Orders" table:

OrderID	CustomerID	EmployeeID	OrderDate
10308	2	7	1996-09-18

SOURCE: <http://www.w3schools.com>

21

## Example #1 of FULL OUTER JOIN– Cont'd

- SQL Code:

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
FULL OUTER JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
```

### CustomerName

Alfreds Futterkiste
Ana Trujillo Emparedados y helados
Antonio Moreno Taquería

**Note:** The FULL OUTER JOIN keyword returns all the rows from the left table (Customers), and all the rows from the right table (Orders). If there are rows in "Customers" that do not have matches in "Orders", or if there are rows in "Orders" that do not have matches in "Customers", those rows will be listed as well.

SOURCE: <http://www.w3schools.com>

22

## Example #2 of FULL OUTER JOIN using Employee & Department Tables

- SQL Code:

```
SELECT *
FROM employee FULL OUTER JOIN department
ON employee.DepartmentID = department.DepartmentID;
```

Employee.LastName	Employee.DepartmentID	Department.DepartmentName	Depar
Smith	34	Clerical	
Jones	33	Engineering	
Robinson	34	Clerical	
Williams	NULL	NULL	
Heisenberg	33	Engineering	

SOURCE: [http://en.wikipedia.org/wiki/Join\\_%28SQL%29](http://en.wikipedia.org/wiki/Join_%28SQL%29)

23

## Using UNION ALL substituting for FULL OUTER JOIN

- Some database systems do not support the full outer join functionality directly, but they can emulate it through the use of an inner join and UNION ALL selects of the "single table rows" from left and right tables respectively.

24

### SQL Code: INNER JOIN + UNION ALL = FULL OUTER JOIN

```

SELECT employee.LastName, employee.DepartmentID,
       department.DepartmentName, department.DepartmentID
FROM employee
INNER JOIN department ON employee.DepartmentID = department.DepartmentID

UNION ALL

SELECT employee.LastName, employee.DepartmentID,
       CAST(NULL AS VARCHAR(20)), CAST(NULL AS INTEGER)
FROM employee
WHERE NOT EXISTS (
    SELECT * FROM department
    WHERE employee.DepartmentID = department.DepartmentID)

UNION ALL

SELECT CAST(NULL AS VARCHAR(20)), CAST(NULL AS INTEGER),
       department.DepartmentName, department.DepartmentID

```

25

### Result: INNER JOIN + UNION ALL = FULL OUTER JOIN

Employee.LastName	Employee.DepartmentID	Department.DepartmentName	Depar
Smith	34	Clerical	
Jones	33	Engineering	
Robinson	34	Clerical	
Williams	NULL	NULL	
Heisenberg	33	Engineering	
Defferre	34	Sales	

26

## CROSS JOIN

- Returns the Cartesian product of rows from tables in the join.
- It will produce rows which combine each row from the first table with each row from the second table
- Within SELECT statement, use CROSS JOIN explicitly or implicitly

Example of an explicit cross join:

```
SELECT *
FROM employee CROSS JOIN department;
```

Example of an implicit cross join:

```
SELECT *
```

27

## CROSS JOIN

Employee.LastName	Employee.DepartmentID	Department.DepartmentName	Depar
Rafferty	31	Sales	
Jones	33	Sales	
Heisenberg	33	Sales	
Smith	34	Sales	
Robinson	34	Sales	
Williams	NULL	Sales	
Rafferty	31	Engineering	
Jones	33	Engineering	
Heisenberg	33	Engineering	
Smith	34	Engineering	
Robinson	34	Engineering	
Williams	NULL	Engineering	
Rafferty	31	Clerical	

28

## Summary

- ✓ *Covered Basic/Useful SQL Functions*
  - ✓ *Comparison, Control flow, Cast Functions*
  - ✓ *String, Numeric Functions*
  - ✓ *Date/Time Functions*
  - ✓ *Aggregate Functions*
  - ✓ *Summary Functions*
  - ✓ *System-Related Functions*

29