

LAB # 9 & 10:

SQL JOINS (A and B) 2 Labs

Objective (aim) of the experiment

To practice and implement SQL Joins.

Scoring Rubrics for Lab 9:

S#	Task	Weightage	Obtained	Signature and Date
1	Using Join (OR Inner Join) successfully	40 %		
2	Using Left Join (OR Left Outer Join) successfully	30 %		
3	Using Right Join (OR Right Outer Join) successfully	30 %		
Total marks obtained in this lab		100%	%	

Scoring Rubrics for Lab

10 S#	Task	Weightage	Obtained	Signature and Date
1	Using Inner Join and Outer Join (OR Full Outer Join) successfully	50 %		
2	Using Union successfully	25 %		
3	Using Union ALL successfully	25 %		
Total marks obtained in this lab		100%	%	

Equipment used

Sl. No.	Facilities Required	Quantity
1	System	1
2	Operating System	Windows 7
3	DBMS	Sql Server Management Studio 2012

TASKS

SQL joins are used to combine rows from two or more tables. SQL JOIN

Lab Manual COMP 213 Database Systems

An SQL JOIN clause is used to combine rows from two or more tables, based on a common field between them.

The most common type of join is: **SQL INNER JOIN (simple join)**. An SQL INNER JOIN return all rows from multiple tables where the join condition is met.

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	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

Notice that the "CustomerID" column in the "Orders" table refers to the customer in the "Customers" table. The relationship between the two tables above is the "CustomerID" column.

Then, if we run the following SQL statement (that contains an INNER JOIN): Example

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

INNER JOIN Customers

ON

Orders.CustomerID=Customers.CustomerID

it will produce something like this:

OrderID	CustomerName	OrderDate
10308	Ana Trujillo Emparedados y helados	9/18/1996
10365	Antonio Moreno Taquería	11/27/1996
10383	Around the Horn	12/16/1996
10355	Around the Horn	11/15/1996
10278	Berglunds snabbköp	8/12/1996

Different SQL JOINS

Before we continue with examples, we will list the types the different SQL JOINS you can use:

- **INNER JOIN:** Returns all rows when there is at least one match in BOTH tables
- **LEFT JOIN:** Return all rows from the left table, and the matched rows from the right table
- **RIGHT JOIN:** Return all rows from the right table, and the matched rows from the left table
- **FULL JOIN:** Return all rows when there is a match in ONE of the tables

SQL INNER JOIN Keyword

The INNER JOIN keyword selects all rows from both tables as long as there is a match between the columns in both tables.

SQL INNER JOIN Syntax

```
SELECT column_name(s)
```

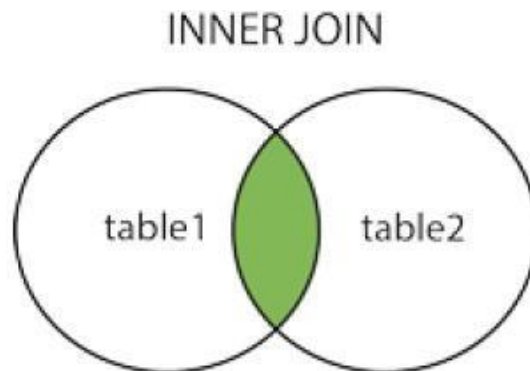
Lab Manual COMP 213 Database Systems

```
FROM table1
INNER JOIN table2
ON table1.column_name=table2.column_name;
```

or:

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

PS! INNER JOIN is the same as JOIN.



Demo Database

In this tutorial we will use the well-known Northwind sample database. Below is a selection from the "Customers" table:

Custo merID	CustomerName	ContactN ame	Address	City	PostalCod e	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y	Ana Trujillo	Avda. de la Constitución	México D.F.	05021	Mexico

Lab Manual COMP 213 Database Systems

	helados		2222			
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

And a selection from the "Orders" table:

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

SQL INNER JOIN Example

The following SQL statement will return all customers with orders: Example

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

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.

SQL LEFT JOIN Keyword

Lab Manual COMP 213 Database Systems

The LEFT JOIN keyword returns all rows from the left table (table1), with the matching rows in the right table (table2). The result is NULL in the right side when there is no match.

```
SQL LEFT JOIN Syntax
SELECT column_name(s)

FROM table1

LEFT JOIN table2

ON table1.column_name=table2.column_name;
```

or:

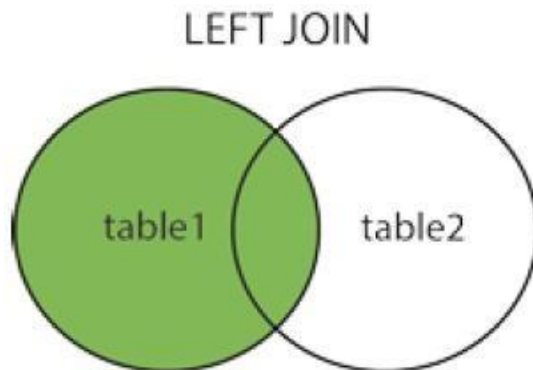
```
SELECT column_name(s)

FROM table1

LEFT OUTER JOIN table2

ON table1.column_name=table2.column_name;
```

PS! In some databases LEFT JOIN is called LEFT OUTER JOIN.



Demo Database

In this tutorial we will use the well-known Northwind sample database. 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	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

And a selection from the "Orders" table:

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

SQL LEFT JOIN Example

The following SQL statement will return all customers, and any orders they might have: Example

```
SELECT Customers.CustomerName,
Orders.OrderID FROM Customers
```

```
LEFT JOIN Orders
```

```
ON
```

```
Customers.CustomerID=Orders.CustomerID
```

```
ORDER BY Customers.CustomerName;
```

Lab Manual COMP 213 Database Systems

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).

SQL RIGHT JOIN Keyword

The RIGHT JOIN keyword returns all rows from the right table (table2), with the matching rows in the left table (table1). The result is NULL in the left side when there is no match.

SQL RIGHT JOIN Syntax
SELECT *column_name(s)*

FROM *table1*

RIGHT JOIN *table2*

ON *table1.column_name=table2.column_name;*

or:

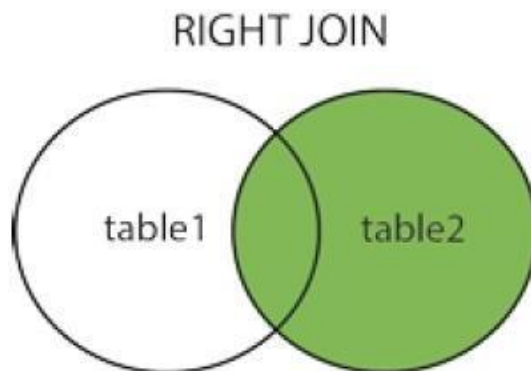
SELECT *column_name(s)*

FROM *table1*

RIGHT OUTER JOIN *table2*

ON *table1.column_name=table2.column_name;*

PS! In some databases RIGHT JOIN is called RIGHT OUTER JOIN.



Lab Manual COMP 213 Database Systems

In this tutorial we will use the well-known Northwind sample database. Below is a selection from the "Orders" table:

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

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 in psychology.. ...
2	Fuller	Andrew	2/19/1952	EmpID2.pic	Andrew received his BTS commercial and....
3	Leverling	Janet	8/30/1963	EmpID3.pic	Janet has a BS degree in chemistry. ...

SQL RIGHT JOIN Example

The following SQL statement will return all employees, and any orders they have placed: Example

```
SELECT Orders.OrderID,  
Employees.FirstName
```

Lab Manual COMP 213 Database Systems

FROM Orders

RIGHT JOIN Employees

ON

Orders.EmployeeID=Employees.EmployeeID

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).

SQL FULL OUTER JOIN Keyword

The FULL OUTER JOIN keyword returns all rows from the left table (table1) and from the right table (table2).

The FULL OUTER JOIN keyword combines the result of both LEFT and

RIGHT joins. SQL FULL OUTER JOIN Syntax

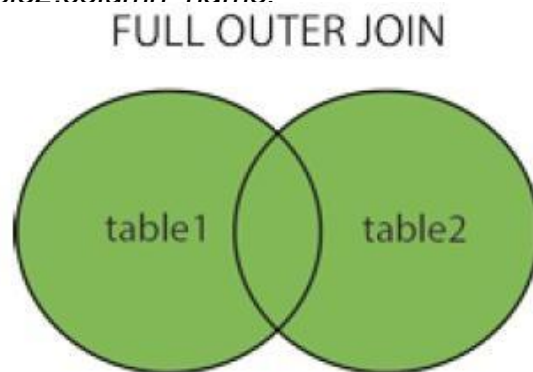
SELECT *column_name(s)*

FROM *table1*

FULL OUTER JOIN *table2*

ON

table1.column_name=table2.column_name:



Demo Database

In this tutorial we will use the well-known Northwind sample database.

Lab Manual COMP 213 Database Systems

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 "Orders" table:

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

SQL FULL OUTER JOIN Example

The following SQL statement selects all customers, and all orders:

```
SELECT Customers.CustomerName, Orders.OrderID
```

```
FROM Customers
```

```
FULL OUTER JOIN Orders
```

```
ON Customers.CustomerID=Orders.CustomerID
```

```
ORDER BY Customers.CustomerName;
```

A selection from the result set may look like this:

CustomerName	OrderID
Alfreds Futterkiste	
Ana Trujillo Emparedados y helados	10308
Antonio Moreno Taquería	10365
	10382
	10351

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.

The SQL UNION Operator

The SQL UNION operator combines the result of two or more SELECT statements.

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

Notice that 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.

SQL UNION Syntax

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

UNION

```
SELECT column_name(s) FROM table2;
```

Note: The UNION operator selects only distinct values by default. To allow duplicate values, use the ALL keyword with UNION.

SQL UNION ALL Syntax

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

```
UNION ALL
```

```
SELECT column_name(s) FROM table2;
```

PS: The column names in the result-set of a UNION are usually equal to the column names in the first SELECT statement in the UNION.

Demo Database

In this tutorial we will use the well-known Northwind sample database. Below is a selection from the "Customers" table:

Custo merID	CustomerName	ContactNa me	Address	City	PostalCod e	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitució n 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

Lab Manual COMP 213 Database Systems

And a selection from the "Suppliers" table:

Supplier ID	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 selects all the **different** cities (only distinct values) from the "Customers" and the "Suppliers" tables:

Example

```
SELECT City FROM Customers UNION SELECT City FROM Suppliers ORDER BY City;
```

Note: UNION cannot be used to list ALL cities from the two tables. If several customers and suppliers share the same city, each city will only be listed once. UNION selects only distinct values. Use UNION ALL to also select duplicate values!

SQL UNION ALL Example

The following SQL statement uses UNION ALL to select **all** (duplicate values also) cities from the "Customers" and "Suppliers" tables:

Example

```
SELECT City FROM
```

```
Customers UNION ALL
```

```
SELECT City FROM  
Suppliers ORDER BY  
City;
```

SQL UNION ALL With WHERE

The following SQL statement uses UNION ALL to select **all** (duplicate values also) **German** cities from the "Customers" and "Suppliers" tables:

Example

```
SELECT City, Country FROM  
Customers WHERE  
Country='Germany'  
  
UNION ALL  
  
SELECT City, Country FROM  
Suppliers WHERE  
Country='Germany'  
  
ORDER BY City;
```

EXPECTED

DELIVERABLE

A spool file showing all executions of the above queries.