wbs
WARWICK BUSINESS SCHOOL
THE UNIVERSITY OF WARWICK

For the Change Makers

Programming for Data Analytics

Week 1: Data Collection SQL Extra

Information Systems and Management
Warwick Business School

DISTINCT

- Eliminate duplicate rows when reporting i.e., report the unique countries from Customers tabel.

```
SELECT DISTINCT country FROM customers
ORDER BY country
```

Country
Argentina
Austria
Belgium
Brazil
Canada
Denmark
Finland
France

Subqueries

- A query within a query i.e., report all products with a price greater than the average price.

```
SELECT ProductName, Price FROM Products
WHERE Price > (SELECT AVG(Price) FROM Products);
```

Number of Records: 25

ProductName	Price
Uncle Bob's Organic Dried Pears	30
Northwoods Cranberry Sauce	40
Mishi Kobe Niku	97
Ikura	31
Queso Manchego La Pastora	38
Alice Mutton	39
Carnarvon Tigers	62.5

The inner query returns a value (avg price) instead of a view. Therefore, it can be used in the where condition

Lists: the IN operator

- Used with a list of values i.e., report data on customers from UK, USA and France.

```
SELECT * FROM Customers WHERE Country IN
('UK','USA','France');
```

is equivalent to…

```
SELECT * FROM Customers WHERE Country = 'UK'
OR Country = 'USA' OR Country = 'France';
```

Lists: the NOT IN operator

- Not in a list of values i.e., report all customers other than those from 'UK' and 'USA'.

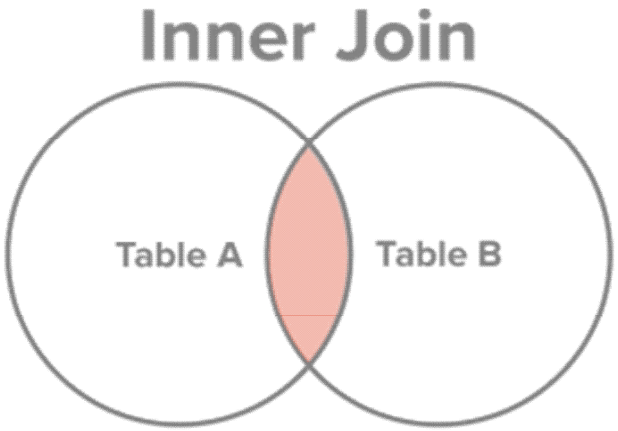
```
SELECT * FROM Customers WHERE Country NOT IN
('UK','USA');
```

is equivalent to…

```
SELECT * FROM Customers WHERE Country <> 'UK'
AND Country <> 'USA';
```

Inner Join

INNER JOIN: Returns records that have matching values in both tables.




```
SELECT columns
FROM tableA
INNER JOIN tableB
ON tableA.key1 = tableB.key2
```

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
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10248	90	5	1996-07-04	3
10249	81	6	1996-07-05	1
10250	34	4	1996-07-08	2

Example

```
SELECT CustomerName, OrderDate
FROM Customers, Orders
WHERE Customers.CustomerID =
Orders.CustomerID
```

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

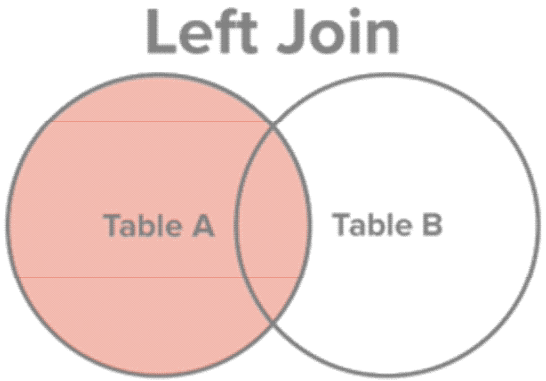
Number of Records: 196

CustomerName	OrderDate
Wilman Kala	1996-07-04
Tradição Hipermercados	1996-07-05
Hanari Carnes	1996-07-08
Victuailles en stock	1996-07-08
Suprêmes délices	1996-07-09

Joining via WHERE condition is an inner join.
JOIN by default is an inner join, so the keyword INNER is optional for some database system.

Left Join

LEFT JOIN: Return all records from the left table, and the matched records from the right table.




```
SELECT columns
FROM tableA
LEFT JOIN tableB
ON tableA.key1 = tableB.key2
```

Example

```
SELECT CustomerName, OrderDate
FROM Customers
LEFT JOIN Orders
ON customers.customerID = orders.customerID
```

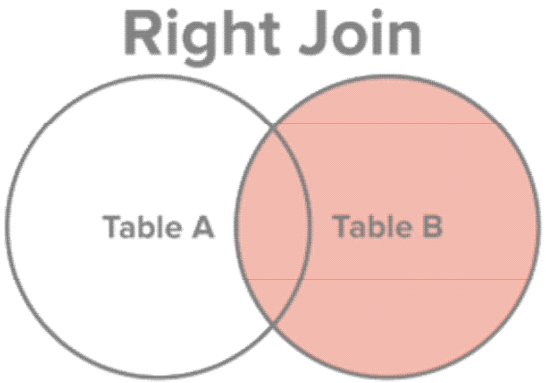
Number of Records: 213

CustomerName	OrderDate
Alfreds Futterkiste	*null*
Ana Trujillo Emparedados y helados	1996-09-18
Antonio Moreno Taquería	1996-11-27
Around the Horn	1996-11-15
Around the Horn	1996-12-16

If a record in table A do not have matching data in table b, then it will return data from table A with corresponding columns from table B showing NULL. Therefore we got 213 records from this query.

Right Join

RIGHT JOIN: Return all records from the right table, and the matched records from the left table.




```
SELECT columns
FROM tableA
RIGHT JOIN tableB
ON tableA.key1 = tableB.key2
```

Example

```
SELECT CustomerName, OrderDate
FROM Customers
RIGHT JOIN Orders
ON customers.customerID = orders.customerID;
```

Number of Records: 196

CustomerName	OrderDate
Wilman Kala	1996-07-04
Tradição Hipermercados	1996-07-05
Hanari Carnes	1996-07-08
Victuailles en stock	1996-07-08
Suprêmes délices	1996-07-09

```
SELECT CustomerName, OrderDate
FROM Orders
LEFT JOIN Customers
ON customers.customerID = orders.customerID;
```

If a record in table B do not have matching data in table b, then it will return data from table B with corresponding columns from table A showing NULL. Not all database support RIGHT JOIN. You can simply switch the order of table A and B and use LEFT JOIN instead.

The HAVING clause

- If aggregated data (such as AVG(), MAX()) needs to be used in a condition, the condition cannot be put in the WHERE clause. You will have to the HAVING clause to set such conditions.
- For example, report customers who had more than 5 orders.

```
SELECT CustomerName, count(OrderID) AS transactions
FROM Customers, Orders
WHERE Customers.CustomerID = Orders.CustomerID
GROUP BY CustomerName
HAVING transactions > 5
```

HAVING clause always comes after the GROUP BY clause

Number of Records: 6

CustomerName	transactions
Ernst Handel	10
Hungry Owl All-Night Grocers	6
QUICK-Stop	7
Rattlesnake Canyon Grocery	7
Split Rail Beer & Ale	6
Wartian Herkku	7

LIMIT

- Sometimes, you do not need to retrieve all records that your query may find, particularly when working with extremely large dataset. You can use LIMIT at the end of your query to set the number of records to return. It often used together with ORDER BY clause.
- For example, find the top 10 customers with most orders.

```
SELECT CustomerName, count(OrderID) AS transactions
FROM Customers, Orders
WHERE Customers.CustomerID = Orders.CustomerID
GROUP BY CustomerName
ORDER BY by transactions DESC
LIMIT 10
```

Number of Records: 10

CustomerName	transactions
Ernst Handel	10
Wartian Herkku	7
Rattlesnake Canyon Grocery	7
QUICK-Stop	7
Split Rail Beer & Ale	6
Hungry Owl All-Night Grocers	6
Mère Paillarde	5
La maison d'Asie	5
LILA-Supermercado	5
Tortuga Restaurante	4