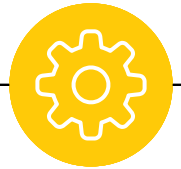
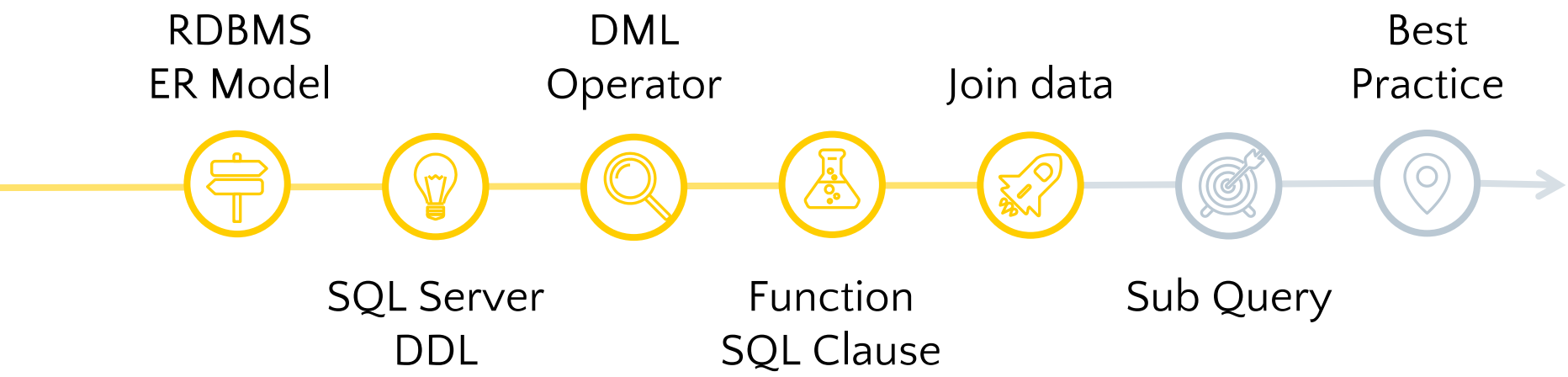
Welcome back



SQL Essentials

Roadmap


RDBMS
ER Model
SQL Server
DDL
DML
Operator
Function
SQL Clause
Join data
Sub Query
Best
Practice

Previous lecture

SQL Clause

- WHERE
- ORDER BY
- GROUP BY
- HAVING

SQL built-in Function

- String funtions
- Datetime functions
- Numeric functions
- Others

What we will explore today?

- INNER JOIN
- OUTER JOIN
- EXCLUDING JOIN
- SELF JOIN
- CROSS JOIN
- UNION & UNION ALL

Why do we need JOIN?

- Combine data from many tables with a matching condition

Prepair database

```
CREATE TABLE Customer(
            CustomerID int PRIMARY KEY AUTO_INCREMENT,
            FullName varchar(20) NOT NULL
);
CREATE TABLE DeliveryAddress(
            ID int PRIMARY KEY AUTO_INCREMENT,
            FullAddress varchar(20) NOT NULL
);

CREATE TABLE CustomerOrder(
            OrderID int PRIMARY KEY AUTO_INCREMENT,
            CustomerID int,
            FoodName varchar(20) NOT NULL,
            DeliveryAddressID int
);
```

Prepair database

- open file “LECTURE5_JOIN_DEMO.sql”

Customer

CustomerID	FullName
1	Châu Tinh Trì
2	Châu Nhuận Phát
3	Lý Tiểu Long
4	Thành Long

CustomerOrder

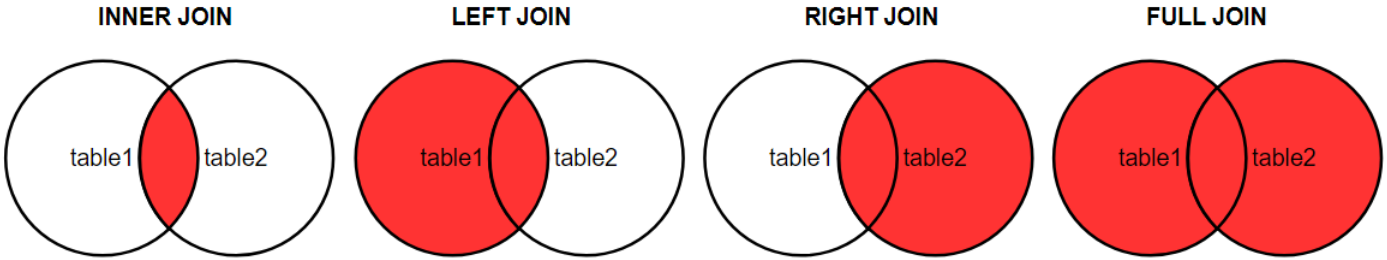
OrderID	CustomerID	FoodName	DeliveryAddressID
1	1	Heo Quay	1
2	1	Gà Luộc	1
3	3	Bò Lá Lốt	2
4	100	Dê Nướng	2

DeliveryAddress

ID	FullAddress
1	TP. HCM
2	TP. HA NOI

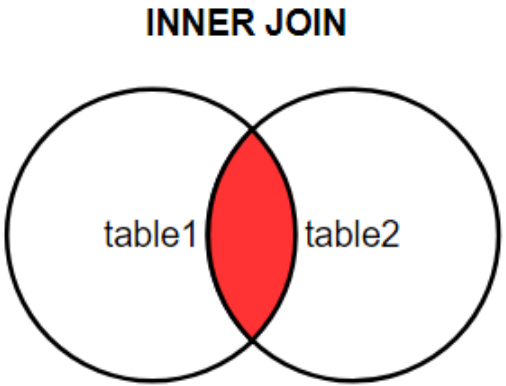
Base of syntax

```
SELECT T1.column_name, T2.column_name
FROM table1 T1
    TYPE_OF_JOIN JOIN table2 T2 ON T1.column_name = T2.column_name;
```



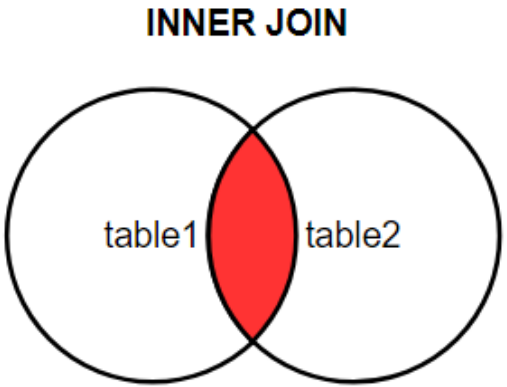
INNER JOIN

- Selects records that have matching values in both tables.



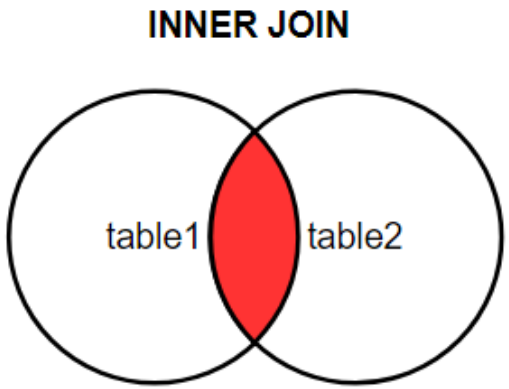
Explain

```
A = {1, 2, 3}
B = {2, 3, 4}
C = A INNER JOIN B
=> C = {2, 3}
```



INNER JOIN syntax

```
SELECT T1.column_name, T2.column_name
FROM table1 T1
    INNER JOIN table2 T2 ON T1.column_name = T2.column_name;
```



Customer

CustomerID	FullName
1	Châu Tinh Trì
2	Châu Nhuận Phát
3	Lý Tiểu Long
4	Thành Long

CustomerOrder

OrderID	CustomerID	FoodName	DeliveryAddress ID
1	1	Heo Quay	1
2	1	Gà Luộc	1
3	3	Bò Lá Lốt	2
4	100	Dê Nướng	2

```
SELECT c.CustomerID, c.FullName, o.FoodName
FROM Customer c
    INNER JOIN CustomerOrder o ON c.CustomerID = o.CustomerID
```

	CustomerID	FullName	FoodName
▶	1	Châu Tinh Trì	Heo Quay
	1	Châu Tinh Trì	Gà Luộc
	3	Lý Tiểu Long	Bò Lá Lốt

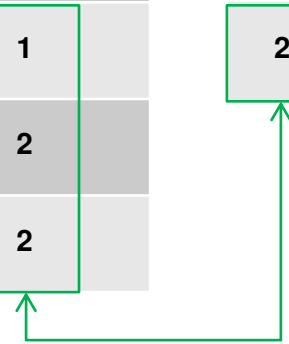
Practice

CustomerOrder

OrderID	CustomerID	FoodName	DeliveryAddressID
1	1	Heo Quay	1
2	1	Gà Luộc	1
3	3	Bò Lá Lốt	2
4	100	Dê Nướng	2

DeliveryAddress

ID	FullAddress
1	TP. HCM
2	TP. HA NOI



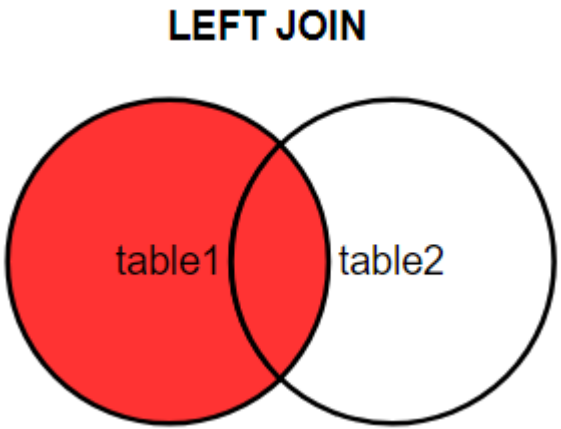
JOIN more than 2 tables

```
SELECT c.CustomerID, c.FullName, o.FoodName, d.FullAddress
FROM Customer c
    INNER JOIN CustomerOrder o ON c.CustomerID = o.CustomerID
    INNER JOIN DeliveryAddress d ON d.ID = o.DeliveryAddressID
```

CustomerID	FullName	FoodName	FullAddress
1	Châu Tinh Trì	Heo Quay	TP. HCM
1	Châu Tinh Trì	Gà Luộc	TP. HCM
3	Lý Tiểu Long	Bò Lá Lốt	TP. HA NOI

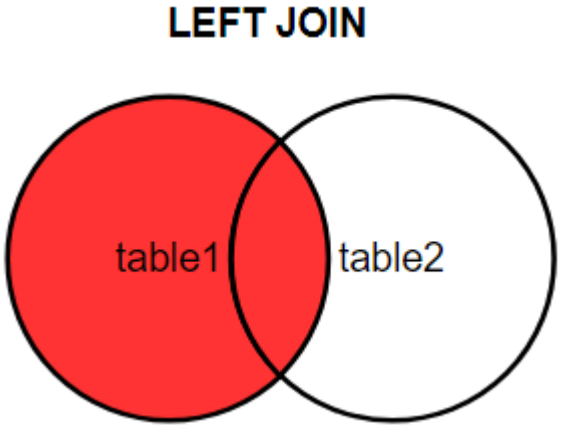
LEFT JOIN

- Returns all records from the left table (table1), and the matching records from the right table (table2)



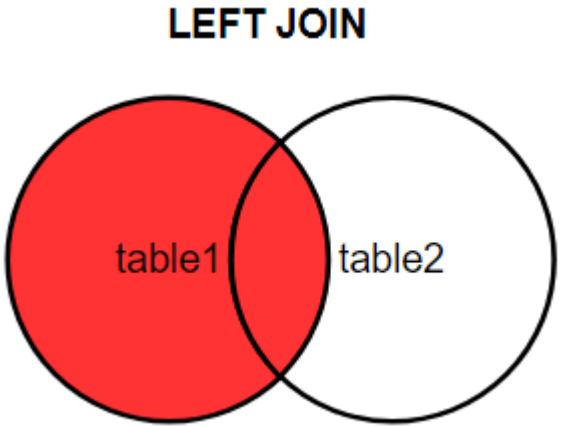
Explain

```
A = {1, 2, 3}
B = {2, 3, 4}
C = A LEFT JOIN B
=> C = {1, 2, 3}
```



LEFT JOIN syntax

```
SELECT T1.column_name, T2.column_name
FROM table1 T1
    LEFT JOIN table2 T2 ON T1.column_name = T2.column_name;
```



Customer

CustomerID	FullName
1	Châu Tinh Trì
2	Châu Nhuận Phát
3	Lý Tiểu Long
4	Thành Long

CustomerOrder

OrderID	CustomerID	FoodName	DeliveryAddressID
1	1	Heo Quay	1
2	1	Gà Luộc	1
3	3	Bò Lá Lốt	2
4	100	Dê Nướng	2

```
SELECT c.CustomerID, c.FullName, o.FoodName
FROM Customer c
    LEFT JOIN CustomerOrder o ON c.CustomerID = o.CustomerID
```

	CustomerID	FullName	FoodName
▶	1	Châu Tinh Trì	Gà Luộc
	1	Châu Tinh Trì	Heo Quay
	2	Châu Nhuận Phát	NULL
	3	Lý Tiểu Long	Bò Lá Lốt
	4	Thành Long	NULL

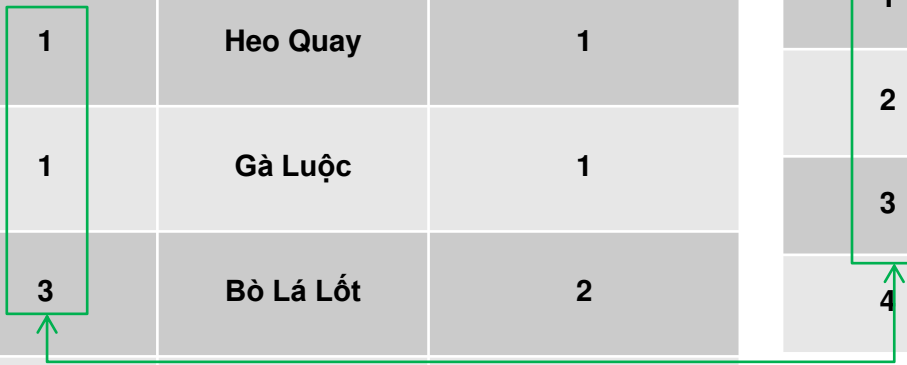
Practice

CustomerOrder

OrderID	CustomerID	FoodName	DeliveryAddressID
1	1	Heo Quay	1
2	1	Gà Luộc	1
3	3	Bò Lá Lốt	2
4	100	Dê Nướng	2

Customer

CustomerID	FullName
1	Châu Tinh Trì
2	Châu Nhuận Phát
3	Lý Tiểu Long
4	Thành Long



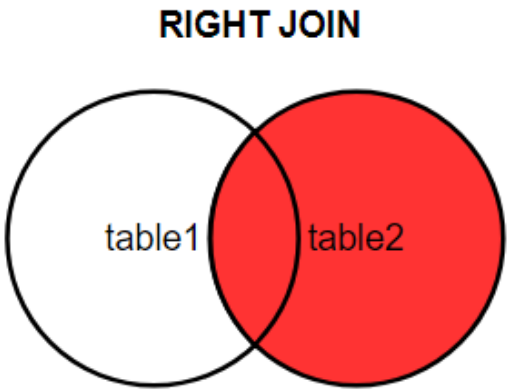
JOIN more than 2 tables

```
SELECT c.CustomerID, c.FullName, o.FoodName, d.FullAddress
FROM Customer c
    LEFT JOIN CustomerOrder o ON c.CustomerID = o.CustomerID
    LEFT JOIN DeliveryAddress d ON d.ID = o.DeliveryAddressID
```

	CustomerID	FullName	FoodName	FullAddress
▶	1	Châu Tinh Trì	Gà Luộc	TP. HCM
	1	Châu Tinh Trì	Heo Quay	TP. HCM
	2	Châu Nhuận Phát	NULL	NULL
	3	Lý Tiểu Long	Bò Lá Lốt	TP. HA NOI
	4	Thành Long	NULL	NULL

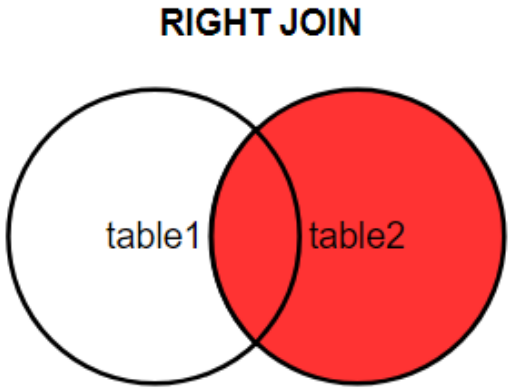
RIGHT JOIN

- Returns all records from the right table (table2), and the matching records from the left table (table1)



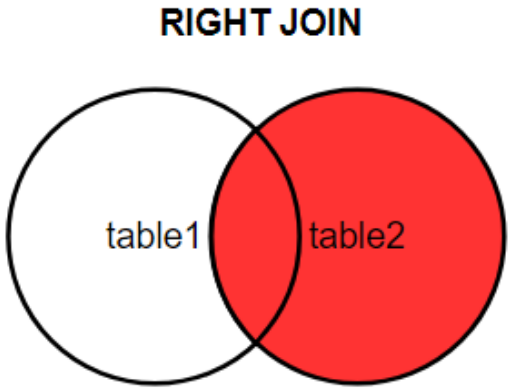
Explain

```
A = {1, 2, 3}
B = {2, 3, 4}
C = A RIGHT JOIN B
=> C = {2, 3, 4}
```



RIGHT JOIN syntax

```
SELECT * FROM T1
RIGHT JOIN T2 ON T1.id = T2.id
```



Customer

CustomerID	FullName
1	Châu Tinh Trì
2	Châu Nhuận Phát
3	Lý Tiểu Long
4	Thành Long

CustomerOrder

OrderID	CustomerID	FoodName	DeliveryAddress ID
1	1	Heo Quay	1
2	1	Gà Luộc	1
3	3	Bò Lá Lốt	2
4	100	Dê Nướng	2

```
SELECT c.CustomerID, c.FullName, o.FoodName
FROM Customer c
        RIGHT JOIN CustomerOrder o ON c.CustomerID = o.CustomerID
```

	CustomerID	FullName	FoodName
▶	1	Châu Tinh Trì	Heo Quay
	1	Châu Tinh Trì	Gà Luộc
	3	Lý Tiểu Long	Bò Lá Lốt
	NULL	NULL	Dê Nướng

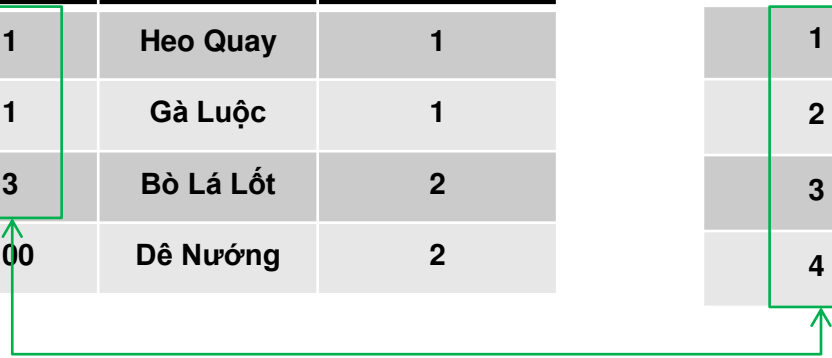
Practice

CustomerOrder

OrderID	CustomerID	FoodName	DeliveryAddress ID
1	1	Heo Quay	1
2	1	Gà Luộc	1
3	3	Bò Lá Lốt	2
4	100	Dê Nướng	2

Customer

CustomerID	FullName
1	Châu Tinh Trì
2	Châu Nhuận Phát
3	Lý Tiểu Long
4	Thành Long



JOIN more than 2 tables

```
SELECT c.CustomerID, c.FullName, o.FoodName, d.FullAddress
FROM Customer c
    RIGHT JOIN CustomerOrder o ON c.CustomerID = o.CustomerID
    RIGHT JOIN DeliveryAddress d ON d.ID = o.DeliveryAddressID
```

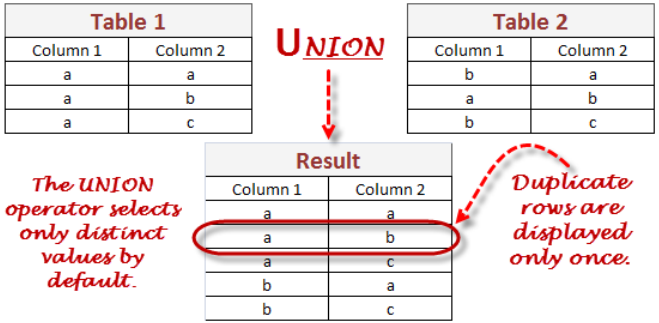
CustomerID	FullName	FoodName	FullAddress
1	Châu Tinh Trì	Heo Quay	TP. HCM
1	Châu Tinh Trì	Gà Luộc	TP. HCM
3	Lý Tiểu Long	Bò Lá Lốt	TP. HA NOI
NULL	NULL	Dê Nướng	TP. HA NOI

UNION Operator

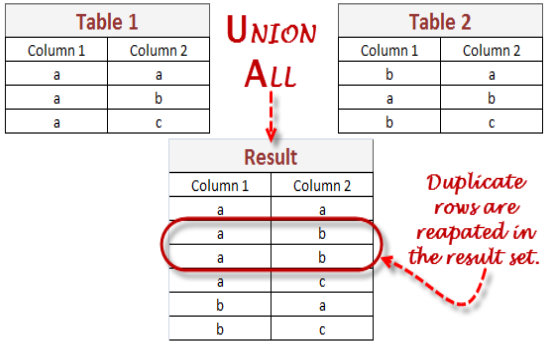
- Used to combine the result-set of two or more SELECT statements.
- Must have the same number of columns
- Columns must also have similar data types
- The columns must also be in the same order

UNION syntax

```
SELECT column_name(s) FROM table1
UNION
SELECT column_name(s) FROM table2;
```




```
SELECT column_name(s) FROM table1
UNION ALL
SELECT column_name(s) FROM table2;
```



DATA

```
DROP DATABASE IF EXISTS LECTURE5_UNION;
CREATE DATABASE LECTURE5_UNION;
USE LECTURE5_UNION;

CREATE TABLE A(
          FromColumnTableA nvarchar(20) NOT NULL
);
CREATE TABLE B(
          FromColumnTableB nvarchar(20) NOT NULL
);
INSERT INTO A(FromColumnTableA) VALUES (1),(2),(3);
INSERT INTO B(FromColumnTableB) VALUES (3),(4),(5);
SELECT * FROM A;
SELECT * FROM B;
```

Results | Messages

	FromColumnTableA
1	1
2	2
3	3

	FromColumnTableB
1	3
2	4
3	5

UNION VS UNION ALL

```
SELECT FromColumnTableA FROM A
UNION
SELECT FromColumnTableB FROM B;
```

Results | Messages

	FromColumnTableA
1	1
2	2
3	3
4	4
5	5

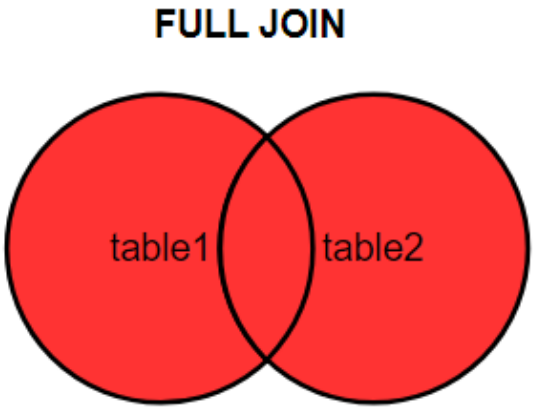
```
SELECT FromColumnTableA FROM A
UNION ALL
SELECT FromColumnTableB FROM B;
```

Results | Messages

	FromColumnTableA
1	1
2	2
3	3
4	3
5	4
6	5

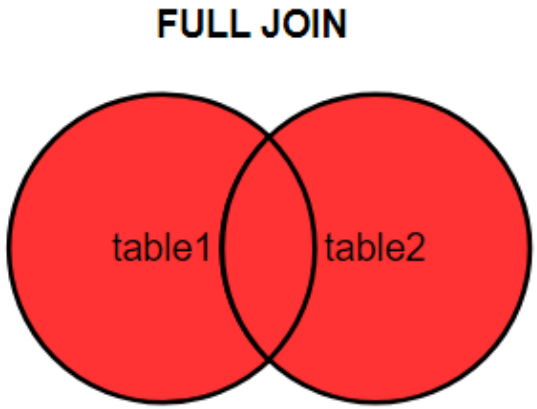
FULL JOIN

- Returns all records even they are not match.
- Not support MySQL, use UNION instead



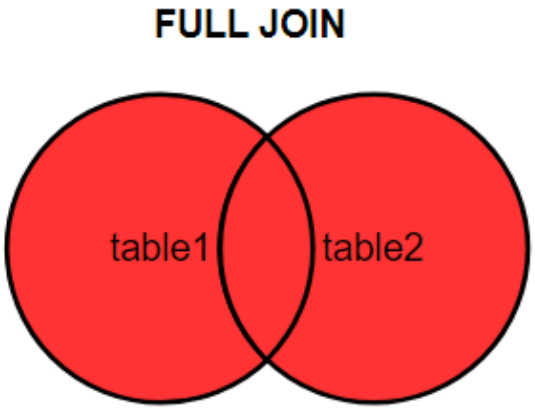
Explain

```
A = {1, 2, 3}
B = {2, 3, 4}
C = A FULL JOIN B
=> C = {1, 2, 3, 4}
```



FULL JOIN syntax

```
SELECT * FROM T1
LEFT JOIN T2 ON T1.id = T2.id
UNION
SELECT * FROM T1
RIGHT JOIN T2 ON T1.id = T2.id
```



Customer

CustomerID	FullName
1	Châu Tinh Trì
2	Châu Nhuận Phát
3	Lý Tiểu Long
4	Thành Long

CustomerOrder

OrderID	CustomerID	FoodName	DeliveryAddressID
1	1	Heo Quay	1
2	1	Gà Luộc	1
3	3	Bò Lá Lốt	2
4	100	Dê Nướng	2

```
SELECT c.CustomerID, c.FullName, o.FoodName
FROM Customer c
        LEFT JOIN CustomerOrder o ON c.CustomerID = o.CustomerID
UNION
SELECT c.CustomerID, c.FullName, o.FoodName
FROM Customer c
        RIGHT JOIN CustomerOrder o
ON c.CustomerID = o.CustomerID
```

CustomerID	FullName	FoodName
1	Châu Tinh Trì	Gà Luộc
1	Châu Tinh Trì	Heo Quay
2	Châu Nhuận Phát	NULL
3	Lý Tiểu Long	Bò Lá Lốt
4	Thành Long	NULL
NULL	NULL	Dê Nướng

Practice

CustomerOrder

OrderID	CustomerID	FoodName	DeliveryAddressID
1	1	Heo Quay	1
2	1	Gà Luộc	1
3	3	Bò Lá Lốt	2
4	100	Dê Nướng	2

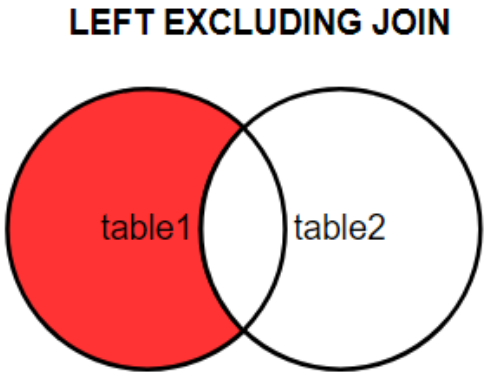
DeliveryAddress

ID	FullAddress
1	TP. HCM
2	TP. HA NOI

	OrderID	FoodName	FullAddress
▶	1	Heo Quay	TP. HCM
	2	Gà Luộc	TP. HCM
	3	Bò Lá Lốt	TP. HA NOI
	4	Dê Nướng	TP. HA NOI

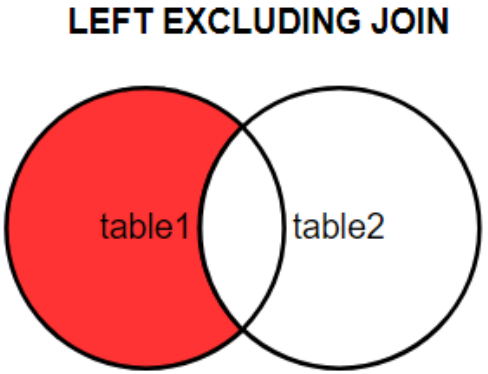
LEFT EXCLUDING JOIN

- Returns all records from the left table (table1) AND exclude matching record



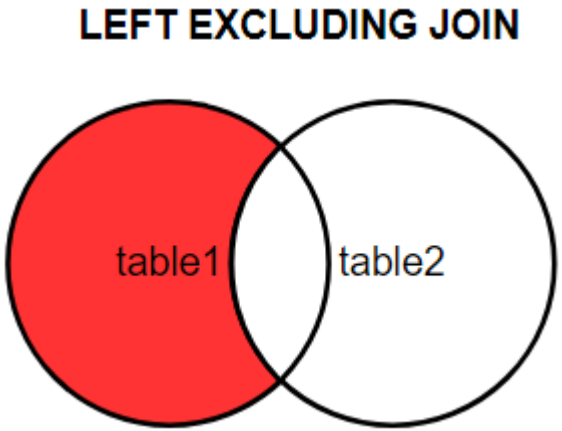
Explain

```
A = {1, 2, 3}
B = {2, 3, 4}
C = A LEFT EXCLUDING JOIN B
=> C = {1}
```



LEFT EXCLUDING JOIN syntax

```
SELECT T1.column_name, T2.column_name
FROM table1 T1
    LEFT JOIN table2 T2 ON T1.column_name = T2.column_name;
WHERE T2.column_name IS NULL
```



Customer

CustomerID	FullName
1	Châu Tinh Trì
2	Châu Nhuận Phát
3	Lý Tiểu Long
4	Thành Long

CustomerOrder

OrderID	CustomerID	FoodName	DeliveryAddressID
1	1	Heo Quay	1
2	1	Gà Luộc	1
3	3	Bò Lá Lốt	2
4	100	Dê Nướng	2

```
SELECT c.CustomerID, c.FullName, o.FoodName
FROM Customer c
    LEFT JOIN CustomerOrder o ON c.CustomerID = o.CustomerID
WHERE o.CustomerID IS NULL
```

	CustomerID	FullName	FoodName
▶	2	Châu Nhuận Phát	NULL
	4	Thành Long	NULL

Practice

CustomerOrder

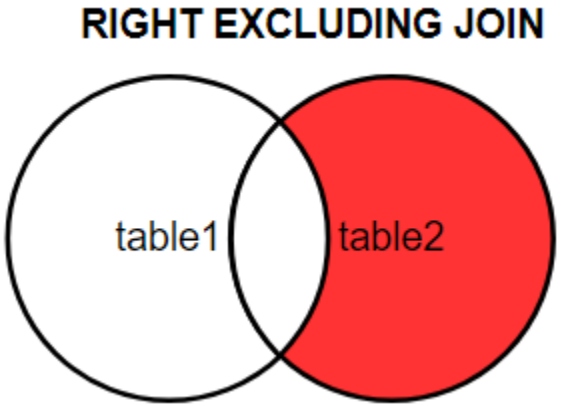
OrderID	CustomerID	FoodName	DeliveryAddressID
1	1	Heo Quay	1
2	1	Gà Luộc	1
3	3	Bò Lá Lốt	2
4	100	Dê Nướng	2

Customer

CustomerID	FullName
1	Châu Tinh Trì
2	Châu Nhuận Phát
3	Lý Tiểu Long
4	Thành Long

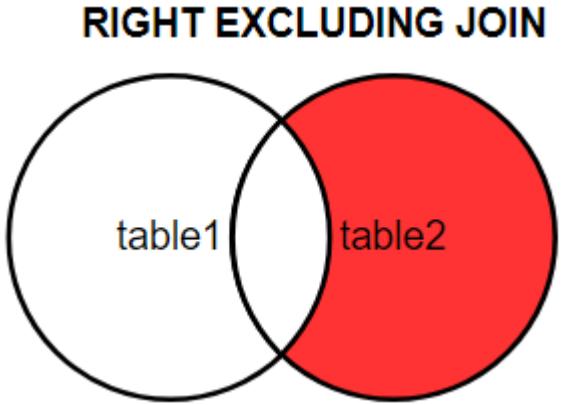
RIGHT EXCLUDING JOIN

- Returns all records from the right table (table1) AND exclude matching record



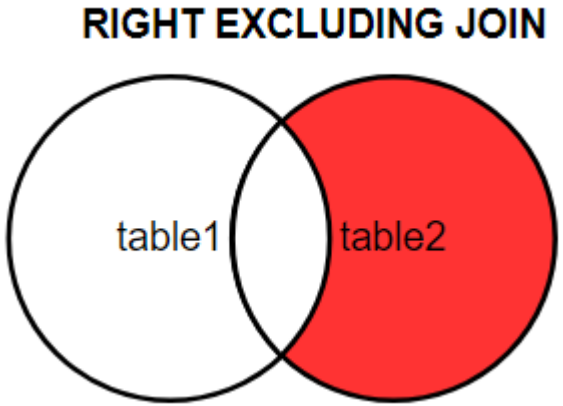
Explain

```
A = {1, 2, 3}
B = {2, 3, 4}
C = A RIGHT EXCLUDING JOIN B
=> C = {4}
```



RIGHT EXCLUDING JOIN syntax

```
SELECT T1.column_name, T2.column_name
FROM table1 T1
    RIGHT JOIN table2 T2 ON T1.column_name = T2.column_name;
WHERE T1.column_name IS NULL
```



Customer

CustomerID	FullName
1	Châu Tinh Trì
2	Châu Nhuận Phát
3	Lý Tiểu Long
4	Thành Long

CustomerOrder

OrderID	CustomerID	FoodName	DeliveryAddress ID
1	1	Heo Quay	1
2	1	Gà Luộc	1
3	3	Bò Lá Lốt	2
4	100	Dê Nướng	2

```
SELECT c.CustomerID, c.FullName, o.FoodName
FROM Customer c
        RIGHT JOIN CustomerOrder o ON c.CustomerID = o.CustomerID
WHERE c.CustomerID IS NULL
```

	CustomerID	FullName	FoodName
▶	NULL	NULL	Dê Nướng

Practice

CustomerOrder

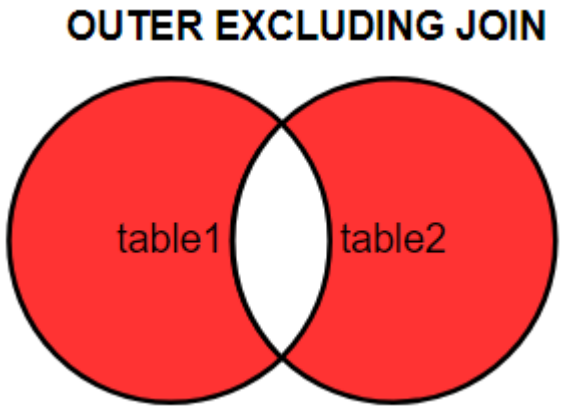
OrderID	CustomerID	FoodName	DeliveryAddressID
1	1	Heo Quay	1
2	1	Gà Luộc	1
3	3	Bò Lá Lốt	2
4	100	Dê Nướng	2

Customer

CustomerID	FullName
1	Châu Tinh Trì
2	Châu Nhuận Phát
3	Lý Tiểu Long
4	Thành Long

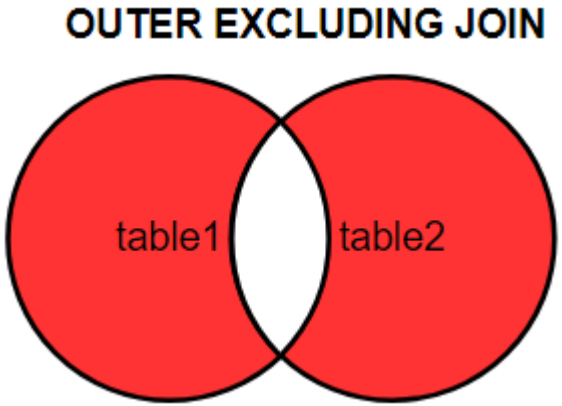
OUTER EXCLUDING JOIN

- Returns all records exclude matching record



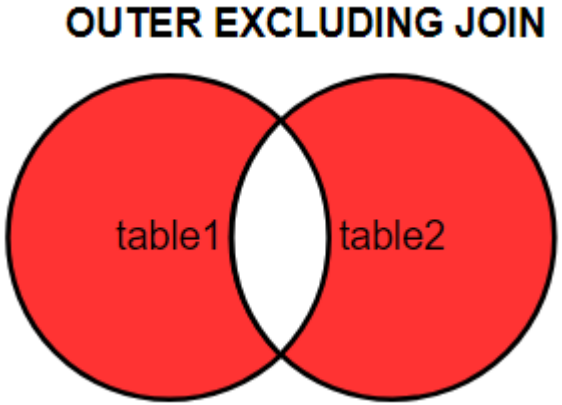
Explain

```
A = {1, 2, 3}
B = {2, 3, 4}
C = A OUTER EXCLUDING JOIN B
=> C = {1, 4}
```



OUTER EXCLUDING JOIN syntax

```sql
SELECT * FROM T1
LEFT JOIN T2 ON T1.id = T2.id
WHERE T2.column_name IS NULL
UNION
SELECT * FROM T1
RIGHT JOIN T2 ON T1.id = T2.id
WHERE T1.column_name IS NULL
```



Customer

CustomerID	FullName
1	Châu Tinh Trì
2	Châu Nhuận Phát
3	Lý Tiểu Long
4	Thành Long

CustomerOrder

OrderID	CustomerID	FoodName	DeliveryAddress ID
1	1	Heo Quay	1
2	1	Gà Luộc	1
3	3	Bò Lá Lốt	2
4	100	Dê Nướng	2

```
SELECT c.CustomerID, c.FullName, o.FoodName
FROM Customer c
        LEFT JOIN CustomerOrder o ON c.CustomerID = o.CustomerID
WHERE o.CustomerID IS NULL
UNION
SELECT c.CustomerID, c.FullName, o.FoodName
FROM Customer c
        RIGHT JOIN CustomerOrder o
ON c.CustomerID = o.CustomerID
WHERE c.CustomerID IS NULL
```

SELF JOIN data

```
CREATE TABLE Employee(
          ID int PRIMARY KEY AUTO_INCREMENT,
          FullName varchar(20) NOT NULL,
          ManagerID int
);

INSERT INTO Employee (FullName, ManagerID)
VALUES
       ('HÀO CEO', NULL),
       ('HƯỜNG DRECTOR', 1),
       ('MẪN CTO', 1),
       ('HUY CULI', 3)
```

ID	FullName	ManagerID
1	HÀO CEO	NULL
2	HƯƠNG DRECTOR	1
3	MẪN CTO	1
4	HUY CULI	3
NULL	NULL	NULL

With manager name

	ID	FullName	ManagerID
▶	1	HÀO CEO	NULL
	2	HƯƠNG DRECTOR	1
	3	MÃN CTO	1
	4	HUY CULI	3
*	NULL	NULL	NULL

```
SELECT emp.ID, emp.FullName, manager.FullName AS Manager
FROM Employee emp, Employee manager
WHERE emp.ManagerID = manager.ID
```

	ID	FullName	Manager
▶	2	HƯƠNG DRECTOR	HÀO CEO
	3	MÃN CTO	HÀO CEO
	4	HUY CULI	MÃN CTO

Is it okay with join?

```
SELECT emp.ID, emp.FullName, manager.FullName
FROM Employee emp, Employee manager
WHERE emp.ManagerID = manager.ID
```

	ID	FullName	Manager
▶	2	HƯƠNG DRECTOR	HÀO CEO
	3	MẪN CTO	HÀO CEO
	4	HUY CULI	MẪN CTO

```
SELECT emp.ID, emp.FullName, manager.FullName
FROM Employee emp
    INNER JOIN Employee manager ON emp.ManagerID = manager.ID
```

Can we just use WHERE?

```
USE LECTURE5_JOIN_DEMO;
-- INNER JOIN
SELECT c.CustomerID, c.FullName, o.FoodName
FROM Customer c
   INNER JOIN CustomerOrder o ON c.CustomerID = o.CustomerID



-- USING WHERE
SELECT c.CustomerID, c.FullName, o.FoodName
FROM Customer c, CustomerOrder o
WHERE c.CustomerID = o.CustomerID
```

CustomerID	FullName	FoodName
1	Châu Tinh Trì	Heo Quay
1	Châu Tinh Trì	Gà Luộc
3	Lý Tiểu Long	Bò Lá Lốt

CustomerID	FullName	FoodName
1	Châu Tinh Trì	Heo Quay
1	Châu Tinh Trì	Gà Luộc
3	Lý Tiểu Long	Bò Lá Lốt

CROSS JOIN

- CROSS JOINs are used to combine each row of one table with each row of another table

NEED DATA?

```
DROP DATABASE IF EXISTS LECTURE5_CROSSJOIN;
CREATE DATABASE LECTURE5_CROSSJOIN;
USE LECTURE5_CROSSJOIN;

CREATE TABLE A(
    FromColumnTableA varchar(20) NOT NULL
);
CREATE TABLE B(
    FromColumnTableB varchar(20) NOT NULL
);
INSERT INTO A(FromColumnTableA) VALUES (1),(2),(3)
INSERT INTO B(FromColumnTableB) VALUES (7),(8),(9)
```

FromColumnTableA
1
2
3

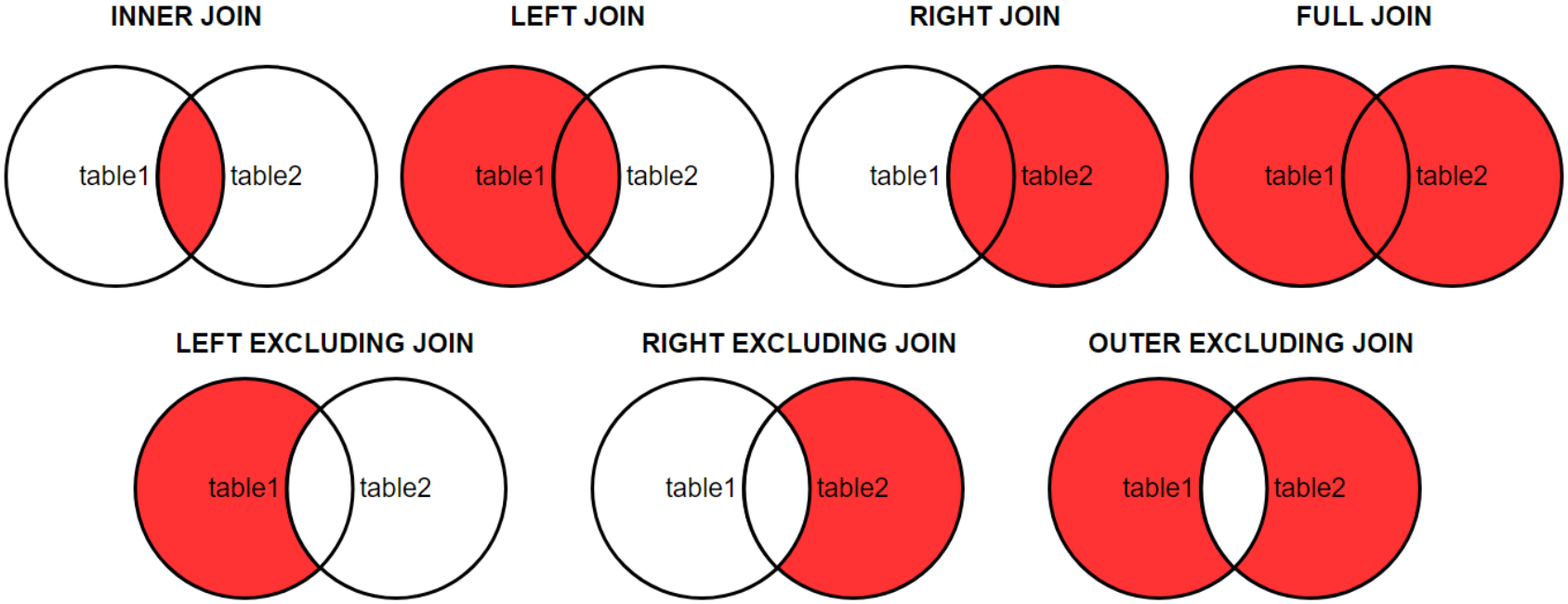
FromColumnTableB
7
8
9

HOW TO DO IT

```
SELECT *
FROM A
    CROSS JOIN B
```

	FromColumnTableA	FromColumnTableB
▶	3	7
	2	7
	1	7
	3	8
	2	8
	1	8
	3	9
	2	9
	1	9

SUMARY


INNER JOIN
table1
table2
LEFT JOIN
table1
table2
RIGHT JOIN
table1
table2
FULL JOIN
table1
table2
LEFT EXCLUDING JOIN
table1
table2
RIGHT EXCLUDING JOIN
table1
table2
OUTER EXCLUDING JOIN
table1
table2

Extra Resources

Name	Link
Became SQL god?	https://www.w3schools.com/sql/default.asp