

# JOIN

## JOIN

- 테이블 두개를 서로 연결해서 더 많은 열 정보를 갖는 하나의 테이블로 만들 때 사용

### INNER JOIN

**SELECT** 테이블1.열이름, 테이블2.열이름

**FROM** 테이블 1

**INNER JOIN** 테이블 2

**ON** 테이블1.열이름 = 테이블2.열이름

### INNER JOIN 사용법

- 테이블1과 테이블2 공통으로 있는 행만 출력



- 테이블이 여러개 나오므로 AS를 이용해서 별칭을 사용

```
SELECT 표1.고객이름, 표2.세일
FROM 고객표 AS 표1
INNER JOIN 주문내역 AS 표2
ON 표1.고객번호 = 표2.고객번호
```

- SELECT 뒤에 열이름 사용 시 같은 열이름이 두개 이상의 테이블에 있을 때 테이블 출처를 표시해야 함  
(같은 열이름이 없을 경우 테이블의 출처를 생략해도 무관)

```
SELECT 표1.고객이름, 표2.세일
FROM 고객표 AS 표1
INNER JOIN 주문내역 AS 표2
ON 표1.고객번호 = 표2.고객번호
```

```
SELECT 고객이름, 세일
FROM 고객표 AS 표1
INNER JOIN 주문내역 AS 표2
ON 표1.고객번호 = 표2.고객번호
```

- INNER를 생략하고 JOIN 이라고만 사용해도 가능

ex)

SupplierID 기준으로 INNER JOIN해.

SQL Statement:

Get your own SQL server

```

SELECT p.*, s.*
FROM Products as p
INNER JOIN Suppliers as s
on p.supplierid = s.supplierid

```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 77

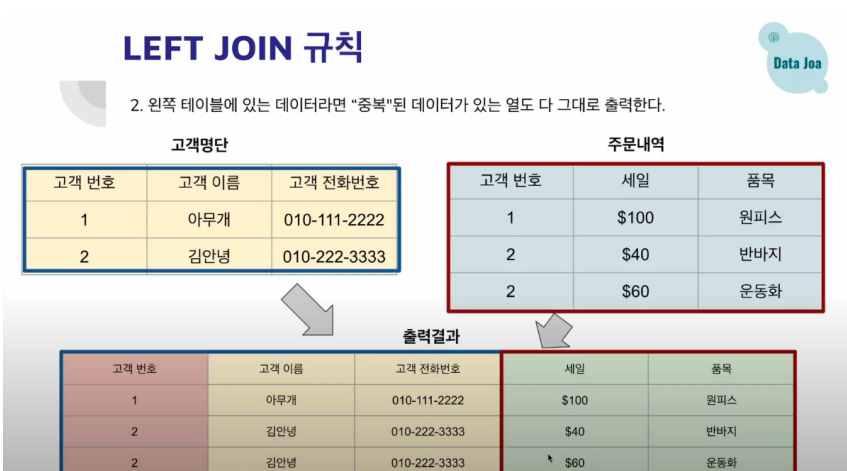
ProductID	ProductName	p.SupplierID	CategoryID	Unit	Price	s.SupplierID	SupplierName	ContactNa
2	Chang	1	1	24 - 12 oz bottles	19	1	Exotic Liquid	Charlotte Cooper
1	Chais	1	1	10 boxes x 20 bags	18	1	Exotic Liquid	Charlotte Cooper

LEFT JOIN

SELECT 테이블1.열이름, 테이블2.열이름  
FROM 테이블 1  
LEFT JOIN 테이블 2  
ON 테이블1.열이름 = 테이블2.열이름

LEFT JOIN 사용법

1. 테이블 1에 있는 모든 행을 출력



ex)  
customerid기준으로 LEFT JOIN해.

SQL Statement:

Get your own SQL server

```

SELECT C.CustomerName, O.OrderID
FROM Customers AS C
LEFT JOIN Orders AS O
ON C.CustomerID=O.CustomerID
ORDER BY C.CustomerName;

```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 213

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

## RIGHT JOIN

SELECT 테이블1.열이름, 테이블2.열이름

FROM 테이블 1

RIGHT JOIN 테이블 2

ON 테이블1.열이름 = 테이블2.열이름

실전에서 전혀 사용하지 않는다고 봐도 무방

- 왜냐하면 방향만 다른거고 왼쪽으로 합치는 것이 직관적으로 해석하기 편하므



## FULL JOIN

SELECT 테이블1.열이름, 테이블2.열이름

FROM 테이블 1

FULL JOIN 테이블 2

ON 테이블1.열이름 = 테이블2.열이름



# FULL JOIN 언제 쓰일까?

가게주문

고객 번호	세일	품목
1	\$10	양말
2	\$400	드레스

온라인주문

고객 번호	세일	품목
2	\$100	원피스
3	\$40	반바지

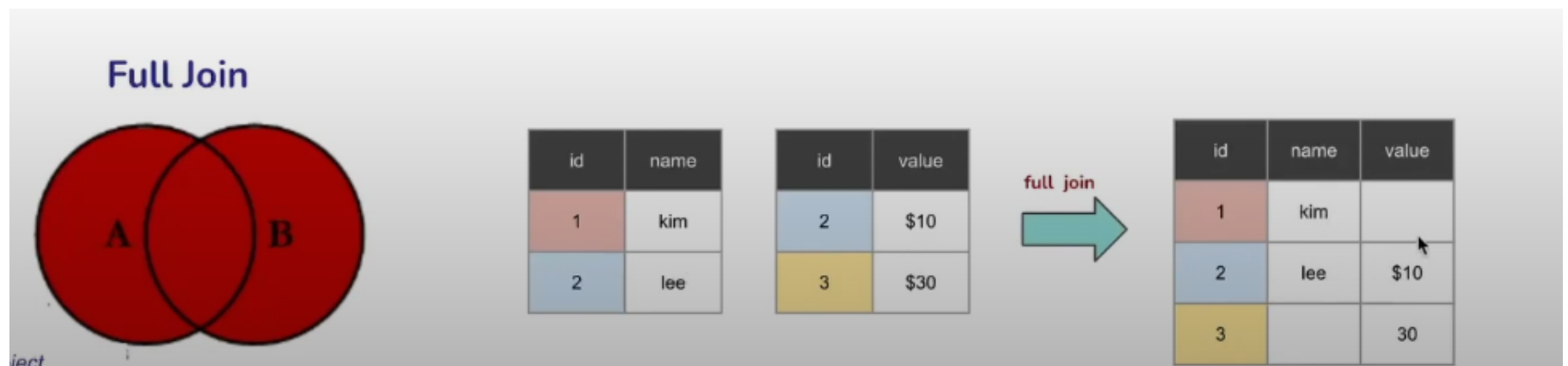
FULL JOIN 결과

고객 번호	가게 세일	가게 세일 품목	온라인 세일	온라인 세일 품목
1	\$10	양말		
2	\$400	드레스	\$100	원피스
3			\$40	반바지

- 어디서 구매했는지만 다를 뿐 데이터는 동일
- 다른 테이블로 저장되는 경우 하나로 병합하여 표기하기 위해 사용

## FULL JOIN 사용법

- 테이블1, 2에 있는 모든 행 출력



- 주의할 점이 있음  
(모든 행을 출력하기 때문에 출력 테이블이 커지므로 꼭 필요할 때만 사용)

## UNION

SELECT 열이름

FROM 테이블 1

UNION

SELECT 열이름

FROM 테이블 2

## 이번엔 UNION



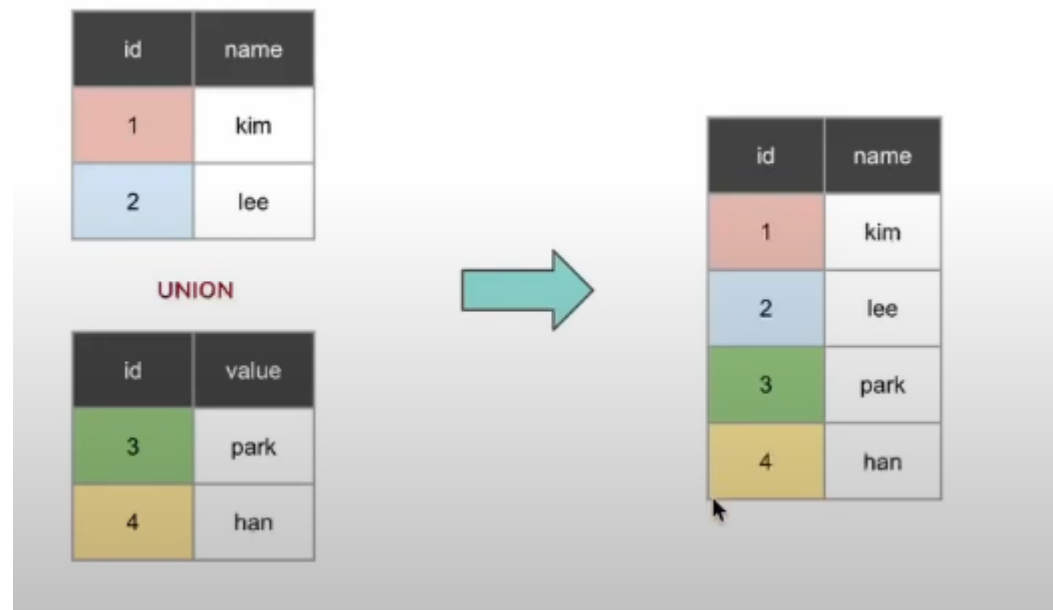
가게주문			온라인주문		
고객 번호	세일	품목	고객 번호	세일	품목
1	\$10	양말	2	\$100	원피스
2	\$400	드레스	3	\$40	반바지

UNION 결과		
고객 번호	세일	품목
1	\$10	양말
2	\$400	드레스
2	\$100	원피스
3	\$40	반바지

### UNION 사용법

1. UNION 앞에 써있는 쿼리 결과와 UNION 뒤에 써있는 쿼리 결과를 합침 (CONCAT)



2. 이어주는 테이블의 열의 갯수가 같고, 열의 순서도 같아야 함
3. 열의 이름은 같을 필요는 없지만 그 열의 데이터 타입은 같아야 함  
(열의 이름은 UNION 전에 오는 테이블의 이름을 따름)
4. 중복되는 항목이 존재 시 자동으로 하나만 출력  
(중복되는 항목을 모두 출력하고 싶다면 UNION ALL을 사용)

ex)

UNION ALL 차이를 비교해.

SQL Statement:

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

Edit the SQL Statement, and click "Run SQL" to see the result.

[Run SQL »](#)

Result:

Number of Records: 94

City
Aachen
Albuquerque

SQL Statement:

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

Edit the SQL Statement, and click "Run SQL" to see the result.

[Run SQL »](#)

Result:

Number of Records: 120

City
Aachen
Albuquerque

