

초보자를 위한  
**Oracle SQL / Database**



내부 / 외부 조인



한국기술교육대학교  
온라인평생교육원

## 학습내용

- 내부 조인
- 외부 조인

## 학습목표

- **내부 조인**으로 원하는 데이터를 검색할 수 있다.
- **외부 조인**을 지정할 수 있다.

# 내부 조인

내부 / 외부 조인 / 내부 조인

## 조인이란

### 조인이란

- 둘 이상의 테이블들에 데이터가 중첩되지 않게 분산되어 저장되어 있음
- 테이블에 저장되어 있는 데이터와 다른 테이블에 저장되어 있는 데이터의 관계를 이용하여 내게 필요한 데이터를 출력하는 것을 조인이라고 함
- 즉, 조인은 학생 테이블과 전공테이블에 공통으로 갖고 있는 전공코드를 이용하여 학생의 전공명을 출력할 수 있음
- 이것이 조인의 종류 중 내부(등가) 조인

내부 / 외부 조인 / 내부 조인

## 조인이란

### 조인의 종류

- 내부 조인(Inner Join) : 등가 조인이라고도 하며, 테이블의 특정 컬럼 값이 일치하는 데이터를 기준으로 데이터를 출력하는 조인방식
- 외부 조인(Outer Join) : 포괄 조인이라고도 하며, 테이블의 값이 NULL일 경우에도 데이터를 출력해야 할 때 사용하고, 왼쪽 외부 조인, 오른쪽 외부 조인, 전체 외부 조인이 있음
- 자체 조인(Self Join) : 셀프 조인(Self Join)이라고도 하며, 자신의 테이블을 조인함
- 크로스 조인(Cross Join) : 두 테이블 조인 시 가능한 모든 경우의 레코드를 SELECT하는 조인으로 별도의 조건을 기술하지 않는 조인방식

# 내부 조인

내부 / 외부 조인 / 내부 조인

## 내부 조인(Inner Join)

### 내부 조인(Inner Join)

**SELECT** 테이블1.컬럼, 테이블2.컬럼, ...  
**FROM** 테이블1, 테이블2, ...

#### WHERE 조인 조건

[AND 조인 조건]  
[AND 일반 조건];



내부 / 외부 조인 / 내부 조인

## 내부 조인(Inner Join)

### 내부 조인(Inner Join) 작성 시 주의사항

- 조인 조건 : 테이블 간의 관계를 표현
- 조인에 사용되는 테이블 개수가 n이라면  
조인 조건의 개수는  $(n-1)$ 개 이상이어야 함
- FROM 절에서 테이블 별명을 지정할 수 있음
- 동일한 이름의 컬럼이 여러 테이블에 있을 경우  
테이블 이름(또는 별명)을 컬럼 앞에 지정해야 함
- 조인 조건에 “=”가 있는 것이 내부 조인



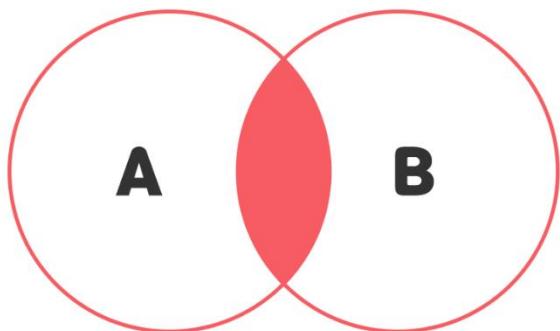
# 내부 조인

내부 / 외부 조인 / 내부 조인

## 내부 조인(Inner Join)

### 내부(등가) 조인

- 내부 조인을 그림으로 나타내면 다음과 같이 두 테이블 A, B의 교집합에 해당하는 데이터를 추출해 낼 수 있음



내부 / 외부 조인 / 내부 조인

## 내부 조인(Inner Join) 예제

### CREATE TABLE 김밥천국

```
(  
    메뉴명 NVARCHAR2(10) NOT NULL,  
    가격 NUMBER(4,0)  
)
```

```
INSERT INTO 김밥천국 VALUES ('김밥',2500);  
INSERT INTO 김밥천국 VALUES ('쫄면', 6000);  
INSERT INTO 김밥천국 VALUES ('돈가스', 7000);  
INSERT INTO 김밥천국 VALUES ('오징어덮밥', 7000);  
INSERT INTO 김밥천국 VALUES ('돌솥비빔밥', 7000);  
INSERT INTO 김밥천국 VALUES ('오므라이스', 6500);  
INSERT INTO 김밥천국 VALUES ('잔치국수', 6000);
```

### CREATE TABLE 김밥나라

```
(  
    메뉴명 NVARCHAR2(10) NOT NULL,  
    가격 NUMBER(4,0)  
)
```

```
INSERT INTO 김밥나라 VALUES ('김밥', 2000);  
INSERT INTO 김밥나라 VALUES ('쫄면', 5500);  
INSERT INTO 김밥나라 VALUES ('오징어덮밥', 7000);  
INSERT INTO 김밥나라 VALUES ('치즈쫄볶이', 4000);  
INSERT INTO 김밥나라 VALUES ('순두부찌개', 6000);
```

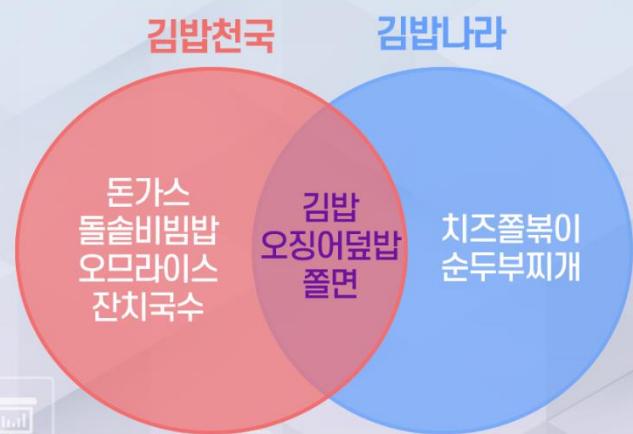
다음과 같이 두 테이블을 생성하고 데이터를 입력

# 내부 조인

내부 / 외부 조인 / 내부 조인

## 내부 조인(Inner Join) 예제

김밥천국	김밥나라
김밥	김밥
쫄면	쫄면
돈가스	
오징어덮밥	오징어덮밥
돌솥비빔밥	
오므라이스	
잔치국수	
	치즈쫄복이
	순두부찌개



두 테이블의 데이터 확인

내부 / 외부 조인 / 내부 조인

## 내부 조인(Inner Join) 예제

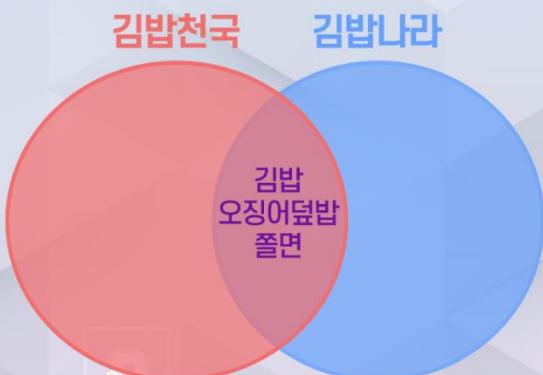
```
워크시트 | 질의 작성기
SELECT * FROM 김밥천국, 김밥나라
WHERE 김밥천국.메뉴명 = 김밥나라.메뉴명;

스크립트 출력 x | 질의 결과 x
SQL | 인출된 모든 행: 3(0.071초)


| 메뉴명     | 가격   | 메뉴명_1 | 가격_1 |
|---------|------|-------|------|
| 1 김밥    | 2500 | 김밥    | 2000 |
| 2 쫄면    | 6000 | 쫄면    | 5500 |
| 3 오징어덮밥 | 7000 | 오징어덮밥 | 7000 |


```

```
SELECT * FROM 김밥천국, 김밥나라
WHERE 김밥천국.메뉴명 = 김밥나라.메뉴명;
```



두 테이블의 내부 조인을 SQL 문으로 만들기

# 내부 조인

내부 / 외부 조인 / 내부 조인

## 내부 조인(Inner Join) 예제

```
워크시트 질의 작성기
SELECT 메뉴명 FROM 김밥천국, 김밥나라
WHERE 김밥천국.메뉴명 = 김밥나라.메뉴명;
```

SQL | 실행 중:SELECT 메뉴명 FROM 김밥천국,  
ORA-00918: 열의 정의가 애매합니다  
00918. 00000 - "column ambiguously defined"  
\*Cause:  
\*Action:  
1행, 8열에 오류 발생

테이블의 모든 데이터를 출력하는 것이 아니라 필요한 데이터만 지정하여 출력을 하려고 하면 SELECT 다음에 컬럼명을 지정해야 함

SELECT 메뉴명 FROM 김밥천국, 김밥나라  
WHERE 김밥천국.메뉴명 = 김밥나라.메뉴명;

메뉴명이라는 컬럼이 김밥천국과 김밥나라, 두 테이블에 모두 있어서, 어느 테이블에 있는 메뉴명을 출력할지를 모르기 때문에 에러 발생

출력 결과에서 메뉴명과 가격 중에 메뉴명만 출력 되도록 다음과 같이 SQL을 작성하면 에러가 나타남

내부 / 외부 조인 / 내부 조인

## 내부 조인(Inner Join) 예제

```
워크시트 질의 작성기
SELECT 김밥천국.메뉴명 FROM 김밥천국, 김밥나라
WHERE 김밥천국.메뉴명 = 김밥나라.메뉴명;
```

SQL | 인출된 모든 행: 30,012 행

메뉴명
1 김밥
2 풀면
3 오징어덮밥

SELECT 김밥천국.메뉴명 FROM 김밥천국, 김밥나라  
WHERE 김밥천국.메뉴명 = 김밥나라.메뉴명;

컬럼명 앞에 테이블명을 지정하여 정확하게 어디에 있는 메뉴를 출력할 것인지를 알려주어야 함

컬럼명 앞에 테이블명을 지정하여 김밥천국에 있는 메뉴명을 출력하도록 함

# 내부 조인

내부 / 외부 조인 / 내부 조인

## 내부 조인(Inner Join) 예제

워크시트 | 질의 작성기

```
SELECT 천.메뉴명 FROM 김밥천국 천, 김밥나라 나
    WHERE 천.메뉴명 = 나.메뉴명;
```

처음에 사용한 SQL문  
SELECT 김밥천국.메뉴명 FROM 김밥천국, 김밥나라  
WHERE 김밥천국.메뉴명 = 김밥나라.메뉴명;

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 3(0.01초)

메뉴명
1 김밥
2 쫄면
3 오징어덮밥

별명을 사용한 SQL문  
SELECT 천.메뉴명 FROM 김밥천국 천, 김밥나라 나  
WHERE 천.메뉴명 = 나.메뉴명;

테이블명과 컬럼명이 길어지고 조건이 많아지면  
별명을 사용하여 간단하고 명료하게 SQL 문을 작성할 수 있음

내부 / 외부 조인 / 내부 조인

## 내부 조인(Inner Join) 예제

워크시트 | 질의 작성기

```
SELECT 천.메뉴명 FROM 김밥천국 천
INNER JOIN 김밥나라 나
ON 천.메뉴명 = 나.메뉴명;
```

오라클 내부 조인  
SELECT 천.메뉴명 FROM 김밥천국 천, 김밥나라 나  
WHERE 천.메뉴명 = 나.메뉴명;

ANSI 내부 조인  
SELECT 천.메뉴명 FROM 김밥천국 천  
INNER JOIN 김밥나라 나  
ON 천.메뉴명 = 나.메뉴명;

오라클 9i 이후(1999년)부터 국제 표준 조인인  
ANSI JOIN을 지원하고 있음

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 3(0.012초)

메뉴명
1 김밥
2 쫄면
3 오징어덮밥

국제 표준 조인인 ANSI JOIN을 사용하면  
다른 DBMS에서도 그대로 사용할 수 있는 장점이 있음

# 외부 조인

내부 / 외부 조인 / 외부 조인

## 외부 조인이란

### 외부 조인이란

#### 내부 조인

- 내부(등가)조인은 컬럼의 공통된 데이터만을 조회하여 출력

#### 외부 조인

- 김밥천국을 기준으로 김밥나라와의 가격을 비교해 보고 싶을 때 사용

내부 / 외부 조인 / 외부 조인

## 외부 조인이란

### 외부 조인의 특징

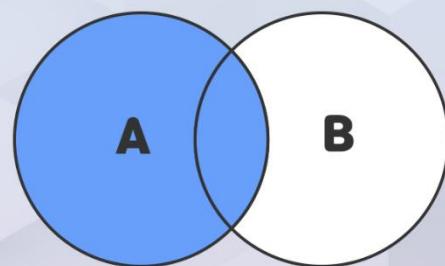
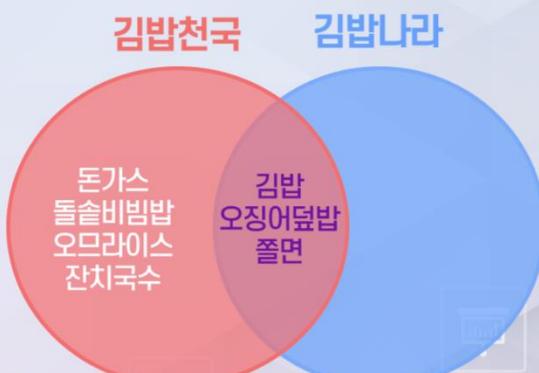
- 두 테이블간 조인에서 **조인 기준 열의 어느 한쪽이 NULL**이어도 강제로 출력
- 외부 조인은 **좌우를 나누어 지정**
  - LEFT OUTER JOIN (LEFT JOIN)  
: **왼쪽 테이블**을 기준으로 조인
  - RIGHT OUTER JOIN (RIGHT JOIN)  
: **오른쪽 테이블**을 기준으로 조인
  - FULL OUTER JOIN (FULL JOIN)  
: **모든 NULL 값까지 출력**

# 외부 조인

내부 / 외부 조인 / 외부 조인

## LEFT JOIN

- 김밥천국을 기준으로 김밥천국과 김밥나라의 메뉴와 가격을 비교하고자 할 때 사용



```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```

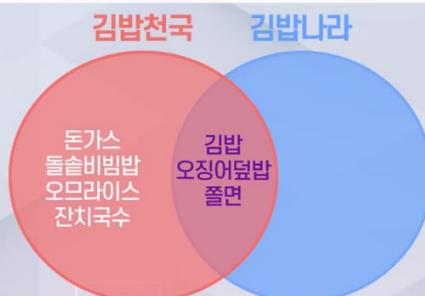
내부 / 외부 조인 / 외부 조인

## LEFT JOIN

```
워크시트 | 질의 작성기
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천, 김밥나라 나
 WHERE 천.메뉴명 = 나.메뉴명 (+);
```

스크립트 출력 x	질의 결과 x		
천메뉴	천가격	나메뉴	나가격
1 김밥	2500	김밥	2000
2 쫄면	6000	쫄면	5500
3 오징어덮밥	7000	오징어덮밥	7000
4 잔치국수	6000	(null)	(null)
5 돈가스	7000	(null)	(null)
6 오므라이스	6500	(null)	(null)
7 둘솔비빔밥	7000	(null)	(null)

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천, 김밥나라 나
 WHERE 천.메뉴명 = 나.메뉴명 (+);
```



왼쪽은 그대로 두고 오른쪽에 + 기호를 넣어 주고  
출력된 결과를 보면 김밥천국에 있는 모든 메뉴와 가격이 표시됨

# 외부 조인

내부 / 외부 조인 / 외부 조인

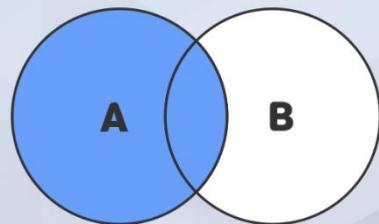
## LEFT JOIN

오라클 LEFT JOIN

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천, 김밥나라 나
 WHERE 천.메뉴명 = 나.메뉴명 (+);
```

ANSI LEFT JOIN

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천
 LEFT JOIN 김밥나라 나
    ON 천.메뉴명 = 나.메뉴명;
```



```
SELECT <select_list>
  FROM TableA A
 LEFT JOIN TableB B
    ON A.Key = B.Key
```

ANSI 조인으로 변경: WHERE 대신에 LEFT JOIN을 사용하고,  
ON 뒤에 조건을 적으면 됨

내부 / 외부 조인 / 외부 조인

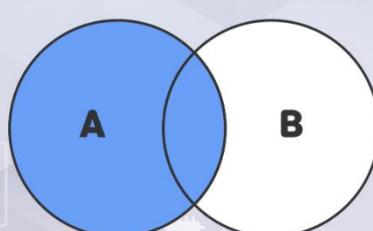
## LEFT JOIN

워크시트 질의 작성기

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천
 LEFT JOIN 김밥나라 나
    ON 천.메뉴명 = 나.메뉴명;
```

천메뉴	천가격	나메뉴	나가격
김밥	2500	김밥	2000
쫄면	6000	쫄면	5500
오징어덮밥	7000	오징어덮밥	7000
잔치국수	6000	(null)	(null)
돈가스	7000	(null)	(null)
오므라이스	6500	(null)	(null)
돌솥비빔밥	7000	(null)	(null)

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천
 LEFT JOIN 김밥나라 나
    ON 천.메뉴명 = 나.메뉴명;
```



```
SELECT <select_list>
  FROM TableA A
 LEFT JOIN TableB B
    ON A.Key = B.Key
```

ANSI LEFT JOIN 실행 결과

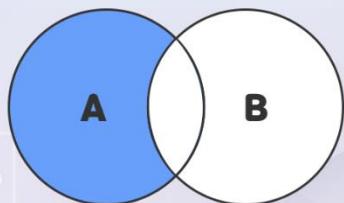
# 외부 조인



내부 / 외부 조인 / 외부 조인

## LEFT JOIN

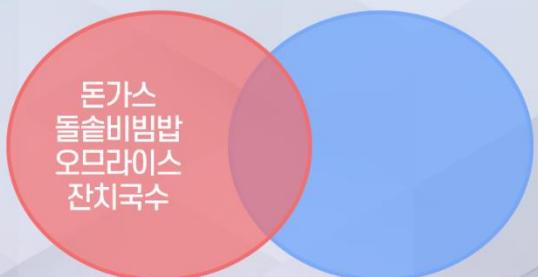
```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천
 LEFT JOIN 김밥나라나
    ON 천.메뉴명 = 나.메뉴명
 WHERE 나.메뉴명 IS NULL;
```



```
SELECT <select_list>
  FROM TableA A
 LEFT JOIN TableB B
    ON A.Key = B.Key
   WHERE B.Key IS NULL
```

김밥천국

김밥나라



김밥천국에만 있는 메뉴와 가격을 출력 : 좀 전의 LEFT JOIN에서 WHERE가 추가되고 빼고자하는 테이블의 컬럼과 IS NULL을 적어 줌

내부 / 외부 조인 / 외부 조인

## LEFT JOIN

워크시트 | 질의 작성기

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천
 LEFT JOIN 김밥나라나
    ON 천.메뉴명 = 나.메뉴명
 WHERE 나.메뉴명 IS NULL;
```

스크립트 출력 x | 질의 결과 x

	나메뉴	나가격
1. 잔치국수	(null)	(null)
2. 돈가스	(null)	(null)
3. 오므라이스	(null)	(null)
4. 돌솥비빔밥	(null)	(null)

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천
 LEFT JOIN 김밥나라나
    ON 천.메뉴명 = 나.메뉴명
 WHERE 나.메뉴명 IS NULL;
```

실행 결과 : 김밥나라의 데이터는 전부 빠진 상태로 출력됨

# 외부 조인

내부 / 외부 조인

외부 조인

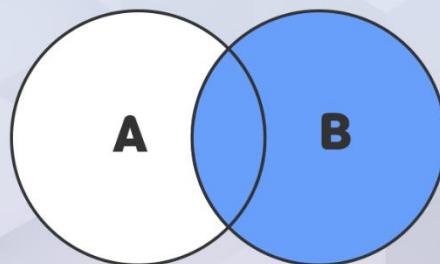
## RIGHT JOIN

- 기준을 바꾸어서 김밥나라를 기준으로 김밥천국과 김밥나라의 메뉴와 가격을 비교하려면 RIGHT JOIN을 사용

김밥천국      김밥나라

김밥  
오징어덮밥  
쫄면

치즈쫄볶이  
순두부찌개



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```

내부 / 외부 조인

외부 조인

## RIGHT JOIN

워크시트		질의 작성기	
<pre>SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,        나.메뉴명 AS 나메뉴, 나.가격 AS 나가격   FROM 김밥천국 천 RIGHT JOIN 김밥나라 나     ON 천.메뉴명 = 나.메뉴명;</pre>			
천메뉴	천가격	나메뉴	나가격
김밥	2500	김밥	2000
쫄면	6000	쫄면	5500
오징어덮밥	7000	오징어덮밥	7000
(null)	(null)	순두부찌개	6000
(null)	(null)	치즈쫄볶이	4000

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천, 김밥나라 나
 WHERE 천.메뉴명 (+) = 나.메뉴명;
```

김밥천국      김밥나라

김밥  
오징어덮밥  
쫄면

치즈쫄볶이  
순두부찌개

오른쪽은 그대로 두고 왼쪽에 + 기호를 넣어 주고  
출력된 결과를 보면 김밥나라에 있는 모든 메뉴와 가격이 표시됨

# 외부 조인

내부 / 외부 조인 / 외부 조인

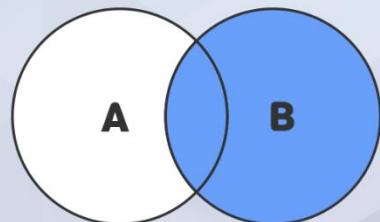
## RIGHT JOIN

오라클 LEFT JOIN

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천, 김밥나라 나
 WHERE 천.메뉴명 (+) = 나.메뉴명
```

ANSI LEFT JOIN

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천
 RIGHT JOIN 김밥나라 나
    ON 천.메뉴명 = 나.메뉴명;
```



```
SELECT <select_list>
  FROM TableA A
 RIGHT JOIN TableB B
    ON A.Key = B.Key
```

ANSI 조인으로 변경: WHERE 대신에 RIGHT JOIN을 사용하고,  
ON 뒤에 조건을 적으면 됨

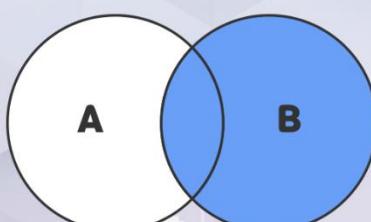
내부 / 외부 조인 / 외부 조인

## RIGHT JOIN

```
워크시트 질의 작성기
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천
 RIGHT JOIN 김밥나라 나
    ON 천.메뉴명 = 나.메뉴명;
```

천메뉴	천가격	나메뉴	나가격
김밥	2500	김밥	2000
쫄면	6000	쫄면	5500
오징어덮밥	7000	오징어덮밥	7000
(null)	(null)	순두부찌개	6000
(null)	(null)	치즈쫄볶이	4000

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천
 RIGHT JOIN 김밥나라 나
    ON 천.메뉴명 = 나.메뉴명
```



```
SELECT <select_list>
  FROM TableA A
 RIGHT JOIN TableB B
    ON A.Key = B.Key
```

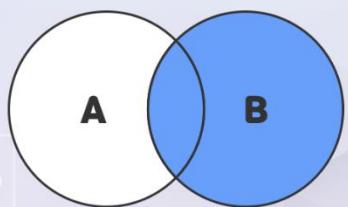
ANSI RIGHT JOIN 실행 결과

# 외부 조인

내부 / 외부 조인 / 외부 조인

## RIGHT JOIN

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천
 RIGHT JOIN 김밥나라 나
    ON 천.메뉴명 = 나.메뉴명
 WHERE 천.메뉴명 IS NULL;
```



```
SELECT <select_list>
  FROM TableA A
 RIGHT JOIN TableB B
    ON A.Key = B.Key
```

김밥천국

김밥나라

치즈쫄볶이  
순두부찌개

김밥나라에만 있는 메뉴와 가격을 출력 : 좀 전의 RIGHT JOIN에서 WHERE가 추가되고 빼고자하는 테이블의 컬럼과 IS NULL을 적어 줌

내부 / 외부 조인 / 외부 조인

## RIGHT JOIN

워크시트 질의 작성기

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천
 RIGHT JOIN 김밥나라 나
    ON 천.메뉴명 = 나.메뉴명
 WHERE 천.메뉴명 IS NULL;
```

스크립트 출력 x 질의 결과 x

SQL | 인출된 모든 행: 2(0,004초)

천메뉴	천가격	나메뉴	나가격
(null)	(null)	순두부찌개	6000
(null)	(null)	쫄볶이	4000

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천
 RIGHT JOIN 김밥나라 나
    ON 천.메뉴명 = 나.메뉴명
 WHERE 나.메뉴명 IS NULL;
```

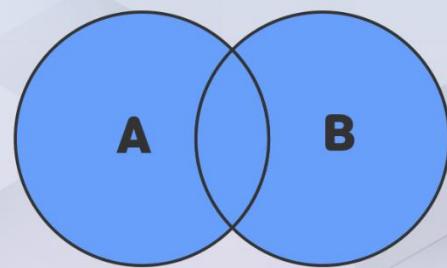
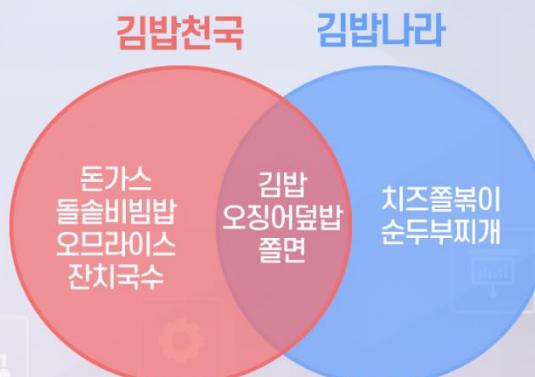
실행 결과: 김밥천국의 데이터는 전부 빠진 상태로 출력됨

# 외부 조인

내부 / 외부 조인 / 외부 조인

## FULL OUTER JOIN

- FULL OUTER JOIN은 양쪽 모두의 데이터를 가져온 뒤 ON 조건절에 일치하지 않거나 존재하지 않는 데이터들을 모두 NULL 값으로 처리하여 모든 테이블에 데이터를 출력함



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```

내부 / 외부 조인 / 외부 조인

## FULL OUTER JOIN

워크시트 / 질의 작성기

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천, 김밥나라 나
 WHERE 천.메뉴명 (+) = 나.메뉴명
 UNION
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천, 김밥나라 나
 WHERE 천.메뉴명 = 나.메뉴명 (+);
```

스크립트 출력 x | 질의 결과 x

	김밥	2500	김밥	2000
1	돈가스	7000	(null)	(null)
2	돌솥비빔밥	7000	(null)	(null)
3	오므라이스	6500	(null)	(null)
4	오징어덮밥	7000	오징어덮밥	7000
5	잔치국수	6000	(null)	(null)
6	쫄면	6000	쫄면	5500
7	(null)	(null)	순두부찌개	6000
8	(null)	(null)	치즈쫄볶이	4000
9	(null)	(null)		

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천, 김밥나라 나
 WHERE 천.메뉴명 (+) = 나.메뉴명
 UNION
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천, 김밥나라 나
 WHERE 천.메뉴명 = 나.메뉴명 (+);
```



오라클 9i 이전에서는 왼쪽과 오른쪽 조인을 하고 UNION으로 묶어 FULL OUTER JOIN을 해결

# 외부 조인

내부 / 외부 조인 / 외부 조인

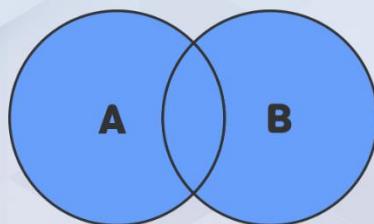
## FULL OUTER JOIN

오라클 FULL OUTER JOIN

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천, 김밥나라 나
 WHERE 천.메뉴명 (+) = 나.메뉴명
UNION
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천, 김밥나라 나
 WHERE 천.메뉴명 = 나.메뉴명 (+);
```

ANSI FULL OUTER JOIN

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천
 FULL OUTER JOIN 김밥나라 나
    ON 천.메뉴명 = 나.메뉴명;
```



```
SELECT <select_list>
  FROM TableA A
 FULL OUTER JOIN TableB B
    ON A.Key = B.Key
```

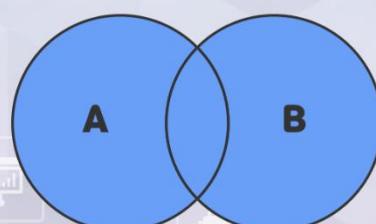
**ANSI 조인으로 변환: WHERE 대신에 FULL OUTER JOIN을 사용하고, ON 뒤에 조건을 적고 UNION 이후는 삭제**

내부 / 외부 조인 / 외부 조인

## FULL OUTER JOIN

워크시트   질의 작성기			
<pre>④ SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,           나.메뉴명 AS 나메뉴, 나.가격 AS 나가격      FROM 김밥천국 천  FULL OUTER JOIN 김밥나라 나     ON 천.메뉴명 = 나.메뉴명;</pre>			
천메뉴	천가격	나메뉴	나가격
김밥	2500	김밥	2000
쫄면	6000	쫄면	5500
돈가스	7000	(null)	(null)
오징어덮밥	7000	오징어덮밥	7000
돌솥비빔밥	7000	(null)	(null)
오므라이스	6500	(null)	(null)
잔치국수	6000	(null)	(null)
(null)	(null)	순두부찌개	6000
(null)	(null)	치즈쫄볶이	4000

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천
 FULL OUTER JOIN 김밥나라 나
    ON 천.메뉴명 = 나.메뉴명;
```



```
SELECT <select_list>
  FROM TableA A
 FULL OUTER JOIN TableB B
    ON A.Key = B.Key
```

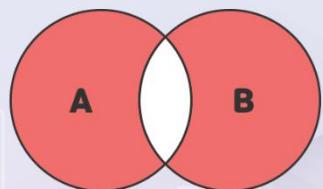
**ANSI FULL OUTER JOIN 실행 결과**

# 외부 조인

내부 / 외부 조인 / 외부 조인

## FULL OUTER JOIN

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천
 FULL OUTER JOIN 김밥나라 나
    ON 천.메뉴명 = 나.메뉴명
   WHERE 천.메뉴명 IS NULL
      OR 나.메뉴명 IS NULL;
```



```
SELECT <select_list>
  FROM TableA A
 FULL OUTER JOIN TableB B
    ON A.Key = B.Key
   WHERE A.Key IS NULL
      OR B.Key IS NULL;
```

김밥천국

김밥나라

돈가스  
돌솥비빔밥  
오므라이스  
잔치국수

치즈쫄볶이  
순두부찌개

김밥천국과 김밥나라에 공통으로 있는 메뉴를 빼고 출력 : 좀 전의 FULL OUTER JOIN에 WHERE 추가 후 LEFT, RIGHT JOIN에서 적었었던 테이블의 컬럼과 IS NULL 적기

내부 / 외부 조인 / 외부 조인

## FULL OUTER JOIN

워크시트		질의 작성기		
<pre>SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,        나.메뉴명 AS 나메뉴, 나.가격 AS 나가격   FROM 김밥천국 천  FULL OUTER JOIN 김밥나라 나     ON 천.메뉴명 = 나.메뉴명    WHERE 천.메뉴명 IS NULL       OR 나.메뉴명 IS NULL;</pre>				
<span>스크립트 출력 x</span> <span>▶ 질의 결과 x</span>				
	천메뉴	천가격	나메뉴	나가격
	돈가스	7000	(null)	(null)
	돌솥비빔밥	7000	(null)	(null)
	오므라이스	6500	(null)	(null)
	잔치국수	6000	(null)	(null)
	(null)	(null)	순두부찌개	6000
	(null)	(null)	치즈쫄볶이	4000

```
SELECT 천.메뉴명 AS 천메뉴, 천.가격 AS 천가격,
       나.메뉴명 AS 나메뉴, 나.가격 AS 나가격
  FROM 김밥천국 천
 FULL OUTER JOIN 김밥나라 나
    ON 천.메뉴명 = 나.메뉴명
   WHERE 천.메뉴명 IS NULL
      OR 나.메뉴명 IS NULL;
```

실행 결과 : 김밥천국과 김밥나라에 공통으로 있는 메뉴가 빠진 상태로 출력됨

# 실습하기

## 내부 / 외부 조인 실습하기

- 다음과 테이블의 관계를 고려하여 내부 조인을 이용하여 다음과 같은 결과를 출력해 보세요.

The diagram illustrates the relationship between two tables, STUDY01.Grade and STUDY01.Scholarship, for performing an inner join. The Grade table has columns: 학번 (Grade Number), 수강년도 (Semester Year), 수강학기 (Semester), and 취득점수 (Score). The Scholarship table has columns: 학번 (Grade Number), 수상구분 (Award Category), 수상학기 (Semester), and 수상여부 (Award Status). An arrow points from the Grade table to the Scholarship table, indicating the join condition.

STUDY01.Grade			
P	교과목코드	CHAR (6 BYTE)	
	교과목명	NVARCHAR2 (30)	
	이수구분	CHAR (1 BYTE)	
	학점	NUMBER (2)	
F	학부코드	NUMBER (2)	
	교과목_PK (교과목코드)		
	교과목_FK (학부코드)		
	교과목_PK (교과목코드)		

STUDY01.Scholarship			
*	학번	NUMBER (10)	
	수강년도	NUMBER (4)	
	수강학기	NUMBER (1)	
	이수구분	CHAR (1 BYTE)	
	취득점수	NUMBER (3)	
	제수강여부	CHAR (1 BYTE)	
F	교과목코드	CHAR (6 BYTE)	
F	교수사번	NUMBER (8)	
	수강_FK1 (교과목코드)		
	수강_FK2 (교수사번)		

학번	수강년도	수강학기	취득점수	수상구분	수상학기	수상여부
1	2018120034	2021	1	94	CSE120	컴퓨터시스템기초설계
2	2018120034	2021	1	94	CSE121	디지털공학 및 실습
3	2018120034	2021	1	88	CSE122	C++프로그래밍
4	2018120034	2021	1	83	CSE130	알고리즘및실습
5	2018120034	2021	1	91	CSE131	컴퓨터구조
6	2018120034	2021	1	96	CSE132	C++프로그래밍
7	2018120034	2021	1	72	CSE130	알고리즘및실습
	18120034		1	95	CSE131	컴퓨터구조
36	2016174067	2021	1	94	CSE130	알고리즘및실습
37	2016174067	2021	1	73	CSE131	컴퓨터구조
38	2016174067	2021	1	65	CSE132	운영체제
39	2016174067	2021	1			

- Oracle Database 19c 버전 사용
- 최신 버전인 Oracle Database 21c는 현재 cloud에서만 사용 가능하며, 지원 기간은 2년

## 적용하기 예1

### 실습단계

적용하기 예1

두 테이블의 내부 조인 관계

교과목코드는 “수강”, “교과목” 테이블 양쪽에 있기 때문에 명확한 지정이 필요

적용하기 예2

적용하기 예3

3개의 테이블을 내부 조인하는 예

적용하기 예4

Oracle SQL 문을 ANSI JOIN인 INNER JOIN으로 변경하기

적용하기 예5

3개의 테이블을 사용한 Oracle SQL 문을 ANSI JOIN인 INNER JOIN으로 변경하기

적용하기 예6

내부 조인에서 테이블의 순서를 변경했을 때 결과를 비교

INNER JOIN으로 변경한 결과 동일

# 실습하기

## 실습단계

INNER JOIN 순서가 변경되어도 결과는 동일

FROM 뒤에 있는 테이블 변경

FROM 뒤 테이블이 변경되어도 결과는 동일

적용하기 예7

INNER JOIN을 LEFT JOIN으로 변경했을 때 결과를 비교

LEFT JOIN 순서를 변경

LEFT JOIN의 FROM 뒤 테이블 변경