

초보자를 위한  
**Oracle SQL / Database**



셀프 / 기타 조인



한국기술교육대학교  
온라인평생교육원

## 학습내용

- 셀프 조인
- 기타 조인

## 학습목표

- 셀프 조인에 대하여 이해하고 활용할 수 있다.
- 기타 조인에 대하여 이해하고 활용할 수 있다.

# 셀프 조인

셀프 / 기타 조인 / 셀프 조인

## 셀프 조인(Self Join)이란

### 셀프 조인(Self Join)

- 내부 조인이나 외부 조인처럼 많이 사용되지는 않음
- 테이블을 조인할 때 자기 자신을 조인하는 것

셀프 / 기타 조인 / 셀프 조인

## 셀프 조인(Self Join)이란

### 셀프 조인 특징

- 조인 관계가 자기 자신을 상대로 함
- 동일한 테이블을 대상으로 조인을 맺는 것이기 때문에 FROM 절에서 별명을 다르게 지정하여 논리적으로 다른 테이블인 것처럼 지정
- 조회하려는 컬럼을 지정하는 SELECT 문에도 컬럼 앞에 별명을 지정하여야 함

# 셀프 조인

셀프 / 기타 조인 / 셀프 조인

## 셀프 조인(Self Join)이란

### 셀프 조인 작성

```
SELECT 별명1.컬럼명, 별명2.컬럼명 ...
FROM 테이블명 별명1
[INNER | LEFT | RIGHT | FULL] JOIN 테이블명 별명2 ...
ON 조인 조건
[AND 일반 조건];
```



셀프 / 기타 조인 / 셀프 조인

## 셀프 조인(Self Join) 예제

### Oracle에 있는 Employees 테이블

사원ID	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
이름	1 EMPLOYEE_ID	NUMBER(6,0)	Yes	(null)	1	Primary key
성	2 FIRST_NAME	VARCHAR2(20 BYTE)	Yes	(null)	2	First name
메일주소	3 LAST_NAME	VARCHAR2(25 BYTE)	Yes	(null)	3	Last name
전화번호	4 EMAIL	VARCHAR2(25 BYTE)	Yes	(null)	4	Email id
입사일	5 PHONE_NUMBER	VARCHAR2(20 BYTE)	Yes	(null)	5	Phone number
부서ID	6 HIRE_DATE	DATE	Yes	(null)	6	Date when
월급	7 JOB_ID	VARCHAR2(10 BYTE)	Yes	(null)	7	Current job
수수료	8 SALARY	NUMBER(8,2)	Yes	(null)	8	Monthly sa
매니저ID	9 COMMISSION_PCT	NUMBER(2,2)	Yes	(null)	9	Commission
부서ID	10 MANAGER_ID	NUMBER(6,0)	Yes	(null)	10	Manager id
	11 DEPARTMENT_ID	NUMBER(4,0)	Yes	(null)	11	Department

테이블의 구조를 보면, 1번에 사원ID(EMPLOYEE\_ID), 2번에 이름...  
10번에 매니저ID(MANAGER\_ID), 11번에 부서ID(DEPARTMENT\_ID) 등이 있음

# 셀프 조인

셀프 / 기타 조인 / 셀프 조인

## 셀프 조인(Self Join) 예제

COLUMN_NAME	DATATYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 EMPLOYEE_ID	NUMBER	Yes	(null)	1	Primary key
2 FIRST_NAME	VARCHAR2(20 BYTE)	Yes	(null)	2	First name
3 LAST_NAME	VARCHAR2(20 BYTE)	Yes	(null)	3	Last name
4 EMAIL	VARCHAR2(20 BYTE)	Yes	(null)	4	Email id
5 PHONE_NUMBER	VARCHAR2(20 BYTE)	Yes	(null)	5	Phone number
6 HIRE_DATE		Yes	(null)	6	Date when
7 JOB_ID	(10 BYTE)	Yes	(null)	7	Current job
8 SALARY	NUMBER(8,2)	Yes	(null)	8	Monthly sa...
9 COMMISSION_PCT	NUMBER(2,2)	Yes	(null)	9	Commission...
10 MANAGER_ID	NUMBER(6,0)	Yes	(null)	10	Manager id...
11					

하나의 행에 있는 정보만으로는 해당 사원의 매니저ID는 알 수 있지만, 매니저 이름은 알 수 없음

사원의 매니저 이름을 알기 위해서는 매니저ID 번호로 사원 테이블에서 다시 찾아야 함

사원을 관리하는 매니저의 정보를 얻기 위해서는 같은 테이블인 사원테이블을 다시 참조하여야 하는데 이것이 셀프 조인

셀프 / 기타 조인 / 셀프 조인

## 셀프 조인(Self Join) 예제

```
워크시트 | 질의 작성기
SELECT e1.employee_id AS 사원ID, e1.first_name AS 사원명
, e2.employee_id AS 매니저ID, e2.first_name AS 매니저명
FROM employees e1
INNER JOIN employees e2
ON e1.manager_id = e2.employee_id;
```

스크립트 출력 x | 질의 결과 x

사원ID	사원명	매니저ID	매니저명
101 Neena	100 Steven		
2 102 Lex	100 Steven		
3 114 Den	100 Steven		
4 120 Matthew	100 Steven		
5 121 Adam	100 Steven		
6 122 Payam	100 Steven		
7 123 Shanta	100 Steven		
8 124 Kevin	100 Steven		
9 145 John	100 Steven		

SELECT e1.employee\_id AS 사원ID, e1.first\_name AS 사원명  
, e2.employee\_id AS 매니저ID, e2.first\_name AS 매니저명  
FROM employees e1  
INNER JOIN employees e2  
ON e1.manager\_id = e2.employee\_id;

사원ID와 사원명 그리고 그 사원을 관리하는 매니저ID와 매니저명을 찾는 셀프 조인 SQL 문을 작성

# 셀프 조인

셀프 / 기타 조인 / 셀프 조인

## 셀프 조인(Self Join) 예제

FROM과 INNER JOIN 뒤를 보면, 하나의 employees 테이블을 각각 e1과 e2로 별명을 지정

```
SELECT e1.employee_id AS 사원ID, e1.first_name AS 사원명
      , e2.employee_id AS 매니저ID, e2.first_name AS 매니저명
  FROM employees e1
 INNER JOIN employees e2
    ON e1.manager_id = e2.employee_id;
```

→ INNER JOIN을 사용하여 매니저ID(e1.manager\_id)와  
사원ID(e2.employee\_id)의 값이 서로 일치하는 데이터만 검색

따라서, 매니저ID가 없는 사원은 결과에서 제외

사원ID와 사원명 그리고 그 사원을 관리하는 매니저ID와  
매니저명을 찾는 셀프 조인 SQL 문을 작성

셀프 / 기타 조인 / 셀프 조인

## 셀프 조인(Self Join) 예제

```
SELECT e1.employee_id AS 사원ID, e1.first_name AS 사원명
      , e2.employee_id AS 매니저ID, e2.first_name AS 매니저명
  FROM employees e1
 INNER JOIN employees e2
    ON e1.manager_id = e2.employee_id;
```

SQL 문을 매니저ID가 NULL이어도 결과가  
출력되게 하려면 외부 조인을 해야 함

이때 기준을 어느 테이블로 하느냐에 따라  
LEFT, RIGHT, FULL JOIN을 선택

```
SELECT e1.employee_id AS 사원ID, e1.first_name AS 사원명
      , e2.employee_id AS 매니저ID, e2.first_name AS 매니저명
  FROM employees e1
 LEFT JOIN employees e2
    ON e1.manager_id = e2.employee_id
 ORDER BY e1.employee_id;
```

사원ID를 기준으로 하기 위해 LEFT JOIN을 하고,  
추가로 사원ID 순으로 나오게 하기 위해 ORDER BY 옵션을 사용

# 셀프 조인

셀프 / 기타 조인 | 셀프 조인

## 셀프 조인(Self Join) 예제

```

    SELECT E1.EMPLOYEE_ID AS 사원ID, E1.FIRST_NAME AS 사원명
      , E2.EMPLOYEE_ID AS 매니저ID, E2.FIRST_NAME AS 매니저명
     FROM EMPLOYEES E1
LEFT JOIN EMPLOYEES E2 ON E1.MANAGER_ID = E2.EMPLOYEE_ID
ORDER BY E1.EMPLOYEE_ID;
  
```

The screenshot shows the results of the query. The first row for employee ID 100 Steven has null values for manager ID and manager name, indicating that Steven does not have a manager.

사원ID	사원명	매니저ID	매니저명
100	Steven	(null)	(null)
101	Neena	100	Steven
102	Lex	100	Steven
103	Alexander	102	Lex
104	Bruce	103	Alexander
105	David	103	Alexander
106	Valli	103	Alexander
107	Diana	103	Alexander
108	Nancy	101	Neena
109	Daniel	108	Nancy
110	John	108	Nancy
111	Ismael	108	Nancy
112	Jose Manuel	108	Nancy

실행 결과를 보면, 사원ID가 100인 스티븐은 매니저가 없다는 것을 볼 수 있음

셀프 / 기타 조인 | 셀프 조인

## 셀프 조인(Self Join) 예제

```

    SELECT e2.employee_id AS 관리자ID, e2.first_name AS 관리자명
      , e1.employee_id AS 사원ID, e1.first_name AS 사원명
     FROM employees e1
LEFT JOIN employees e2
ON e1.manager_id = e2.employee_id
ORDER BY e2.employee_id;
  
```

The screenshot shows the results of the query. The first 10 rows all have '100 Steven' listed under '관리자ID' and '관리자명', indicating that Steven is managing everyone.

관리자ID	관리자명
100	Steven

```

SELECT e2.employee_id AS 매니저ID, e2.first_name AS 매니저명
      , e1.employee_id AS 사원ID, e1.first_name AS 사원명
     FROM employees e1
LEFT JOIN employees e2
ON e1.employee_id = e2.employee_id
ORDER BY e2.employee_id;
  
```

사원번호 순에서 매니저 순으로 검색을 하려면 ORDER BY를 e2. employee\_id로 변경하고, SELECT 문의 컬럼들도 변경

# 셀프 조인



셀프 / 기타 조인 / 셀프 조인

## 셀프 조인(Self Join) 활용

### 셀프 조인이 사용되는 경우

- 앞에서 들은 예와 같이 위계성(상급자, 족보 등)을 나타낼 때
- 작업이나 일의 순서와 같이 순차성(요리 순서, 장난감 조립 등)을 나타낼 때

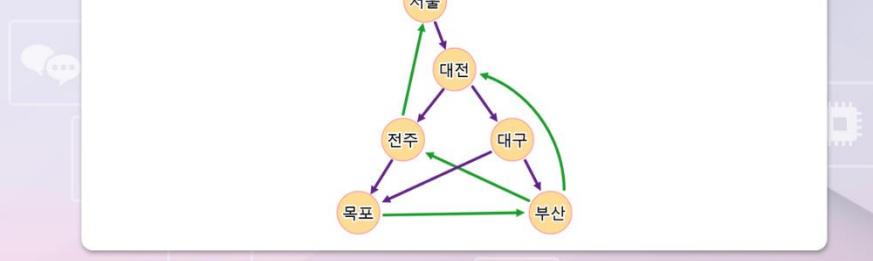
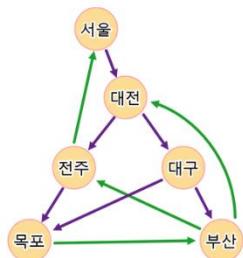


셀프 / 기타 조인 / 셀프 조인

## 셀프 조인(Self Join) 활용

### 셀프 조인이 사용되는 경우

- 데이터 사이에 여러 관계성이 있을 때
  - 기차, 비행기, 배의 항로나 항구와 같이 출발지가 되기도 하면서 목적지가 되는 경우
  - 즉, 서울 → 대전, 부산 → 대전, 목포 → 부산 → 대전, 대전 → 대구 → 부산



# 기타 조인



셀프 / 기타 조인 / 기타 조인

## 기타 조인이란

### + 크로스 조인(Cross Join)

#### 크로스 조인(Cross Join)

- 크로스 조인(Cross Join)은 테이블 상호간에 연결될 수 있는 모든 경우의 수를 나타내는 조인

```
SELECT 별명1.컬럼명, 별명2.컬럼명 ...
FROM 테이블명 별명1, 별명2 ... ;
```



셀프 / 기타 조인 / 기타 조인

## 기타 조인이란

### + 크로스 조인(Cross Join) 예제

```
워크시트 | 질의 작성기
SELECT *
FROM 김밥천국, 김밥나라;
```

스크립트 출력 | 질의 결과 |

SQL | 인출된 모든 행: 35(0,003초)

번호	가격	번호	가격
1	2500	김밥	2000
2	6000	쫄면	2000
3	7000	돈가스	2000
4	7000	오징어덮밥	2000
5	7000	쫄술비빔밥	2000
6	6500	오므라이스	2000
7	6000	잔치국수	2000
8	2500	김밥	5500
9	6000	쫄면	5500
10	7000	돈가스	5500
11	7000	오징어덮밥	5500
12	7000	쫄술비빔밥	5500
13	6500	오므라이스	5500
14	6000	잔치국수	5500
15	2500	김밥	7000
16	6000	쫄면	7000
17	7000	돈가스	7000

SELECT \* FROM 김밥천국, 김밥나라;

김밥천국과 김밥나라를 크로스 조인을 하는 SQL 문과 결과



# 기타 조인

셀프 / 기타 조인 / 기타 조인

## 기타 조인이란

### + 크로스 조인(Cross Join) 예제

The screenshot shows the Oracle SQL Developer interface with a query window containing the following SQL code:

```
SELECT *
FROM 김밥천국, 김밥나라;
```

The results show a Cartesian product of all rows from the KimBapChonGu table and the KimBapNara table. Three annotations explain the connections:

- 김밥천국의 모든 메뉴가 김밥나라의 첫 번째 행인 '김밥'과 연결**
- 김밥천국의 모든 메뉴가 김밥나라의 두 번째 행인 '쫄면'과 연결**
- 같은 식으로 김밥나라의 마지막 행인 '순두부찌개'까지 연결**

결과를 보면 김밥천국의 메뉴인 첫 번째 행과 김밥나라의 메뉴 두 번째 행이 연결되어 있음

셀프 / 기타 조인 / 기타 조인

## 다중 테이블 조인

### 다중 테이블 조인

- 내부조인인 INNER JOIN에서 INNER는 생략 가능

#### [INNER] JOIN

```
SELECT X.컬럼이름a, Y.컬럼이름b, Z.컬럼이름c, ...
FROM 테이블이름x X JOIN 테이블이름y Y
ON X.컬럼이름a=Y.컬럼이름b JOIN 테이블이름z Z
ON Y.컬럼이름b=Z.컬럼이름c;
```

# 기타 조인

셀프 / 기타 조인 / 기타 조인

## 다중 테이블 조인

### 다중 테이블 조인

- 외부조인인 LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN에서 OUTER는 생략 가능

[ LEFT | RIGHT | FULL ] [OUTER] JOIN

```
SELECT X.컬럼이름a, Y.컬럼이름b, Z.컬럼이름c, ...
FROM 테이블이름x X LEFT JOIN 테이블이름y Y
ON X.컬럼이름a=Y.컬럼이름b RIGHT JOIN 테이블이름z Z
ON Y.컬럼이름b=Z.컬럼이름c;
```

CREATE TABLE 우리분식

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

```
INSERT INTO 우리분식 VALUES ('라면',3000);  
INSERT INTO 우리분식 VALUES ('쫄면', 5500);  
INSERT INTO 우리분식 VALUES ('간치국수', 4500);  
INSERT INTO 우리분식 VALUES ('순두부찌개', 6000);  
INSERT INTO 우리분식 VALUES ('햄버거', 7000);
```

다중 테이블에서 예로 사용하기 위해 “우리분식” 테이블을 생성

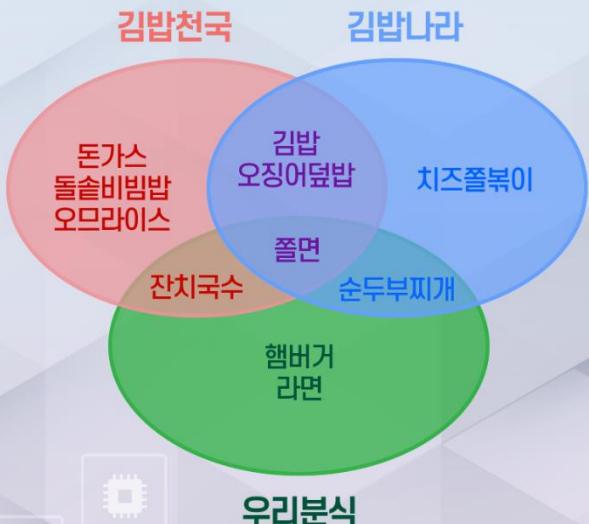
# 기타 조인

셀프 / 기타 조인 / 기타 조인

## 다중 테이블 조인

### + 내부 조인 예제

김밥천국	김밥나라	우리분식
김밥	김밥	
쫄면	쫄면	쫄면
돈가스		
오징어덮밥	오징어덮밥	
돌솥비빔밥		
오므라이스		
잔치국수		잔치국수
	치즈쫄복이	
	순두부찌개	순두부찌개
		햄버거
		라면



세 테이블의 데이터

셀프 / 기타 조인 / 기타 조인

## 다중 테이블 조인

### + 내부 조인 예제

```
워크시트 질의 작성기
SELECT 천.메뉴명, 나.메뉴명, 우.메뉴명
FROM 김밥천국 천 JOIN 김밥나라 나
ON 천.메뉴명 = 나.메뉴명 JOIN 우리분식 우
ON 나.메뉴명 = 우.메뉴명;
```

스크립트 출력 x | 질의 결과 x  
SQL | 인출된 모든 행: 1(0.003초)  
메뉴명 메뉴명\_1 메뉴명\_2  
1 쫄면 쫄면 쫄면

```
SELECT 천.메뉴명, 나.메뉴명, 우.메뉴명
FROM 김밥천국 천 JOIN 김밥나라 나
ON 천.메뉴명 = 나.메뉴명 JOIN 우리분식 우
ON 나.메뉴명 = 우.메뉴명;
```



세 테이블의 내부 조인을 SQL 문으로 만들기

# 기타 조인

셀프 / 기타 조인 / 기타 조인

## ❖ 다중 테이블 조인

### + 외부 조인 예제

워크시트 | 질의 작성기

```

1 SELECT 천.메뉴명, 나.메뉴명, 우.메뉴명
2 FROM 김밥천국 천 LEFT JOIN 김밥나라 나
3 ON 천.메뉴명 = 나.메뉴명 LEFT JOIN 우리분식 우
4 ON 나.메뉴명 = 우.메뉴명;

```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 7(0.018초)

메뉴명	메뉴명_1	메뉴명_2
쫄면	쫄면	쫄면
2 카치국수	(null)	(null)
3 돈가스	(null)	(null)
4 오므라이스	(null)	(null)
5 풀술비빔밥	(null)	(null)
6 김밥	김밥	(null)
7 오징어덮밥	오징어덮밥	(null)

**SELECT** 천.메뉴명, 나.메뉴명, 우.메뉴명  
**FROM** 김밥천국 천 **LEFT JOIN** 김밥나라 나  
**ON** 천.메뉴명 = 나.메뉴명 **LEFT JOIN** 우리분식 우  
**ON** 나.메뉴명 = 우.메뉴명;

다중 테이블에 대한 **LEFT JOIN** 실행 결과

셀프 / 기타 조인 / 기타 조인

## ❖ 다중 테이블 조인

### + 외부 조인 예제

**SELECT** 천.메뉴명, 나.메뉴명, 우.메뉴명

**FROM** 김밥천국 천 **LEFT JOIN** 김밥나라 나

1번째 **LEFT JOIN**

**ON** 천.메뉴명 = 나.메뉴명

1번째 조인 조건

**LEFT JOIN** 우리분식 우

2번째 **LEFT JOIN**

**ON** 나.메뉴명 = 우.메뉴명;

2번째 조인 조건

조인의 실행 순서를 주의 깊게 봐야 함

# 기타 조인

셀프 / 기타 조인 / 기타 조인

## 다중 테이블 조인

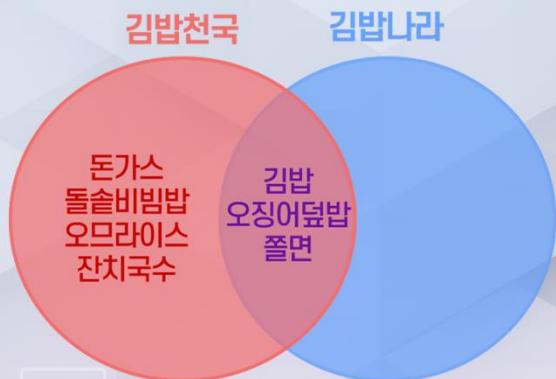
### 외부 조인 예제

```
SELECT 천.메뉴명, 나.메뉴명, 우.메뉴명
FROM 김밥천국 천 LEFT JOIN 김밥나라 나
ON 천.메뉴명 = 나.메뉴명
```

1번째 조인 조건

1번째 LEFT JOIN

첫 번째 조인을 보면 김밥천국과 김밥나라의 왼쪽 조인



셀프 / 기타 조인 / 기타 조인

## 다중 테이블 조인

### 외부 조인 예제

```
LEFT JOIN 우리분식 우
ON 나.메뉴명 = 우.메뉴명;
```

2번째 조인 조건

2번째 LEFT JOIN

두 번째 조인으로 김밥나라와 우리분식의 왼쪽 조인  
여기서 조인의 실행 순서를 주의 깊게 봐야 함



# 기타 조인

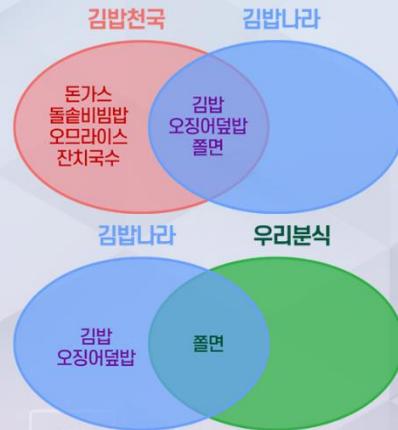
셀프 / 기타 조인 / 기타 조인

## 다중 테이블 조인

### 외부 조인 예제

```
워크시트 절의 작성기
1 SELECT 천.메뉴명, 나.메뉴명, 우.메뉴명
2 FROM 김밥천국 천 LEFT JOIN 김밥나라 나
3 ON 천.메뉴명 = 나.메뉴명 LEFT JOIN 우리분식 우
4 ON 나.메뉴명 = 우.메뉴명;

스크립트 출력 x 절의 결과 x
SQL | 인출된 모든 행: 7(0.018초)
메뉴명 메뉴명_1 메뉴명_2
1 쫄면 쫄면 쫄면
2 잔치국수 (null) (null)
3 돈가스 (null) (null)
4 오므라이스 (null) (null)
5 둘솔비빔밥 (null) (null)
6 김밥 김밥 (null)
7 오징어덮밥 오징어덮밥 (null)
```



첫 번째와 두 번째 조인 그림을 보면 전부 LEFT JOIN  
먼저 김밥천국과 김밥나라의 LEFT JOIN 후, 김밥나라와 우리분식의 LEFT JOIN

셀프 / 기타 조인 / 기타 조인

## 다중 테이블 조인

### 외부 조인 예제

```
워크시트 절의 작성기
1 SELECT 천.메뉴명, 나.메뉴명, 우.메뉴명
2 FROM 김밥천국 천 LEFT JOIN 김밥나라 나
3 ON 천.메뉴명 = 나.메뉴명 LEFT JOIN 우리분식 우
4 ON 천.메뉴명 = 우.메뉴명;

절의 결과 x
SQL | 인출된 모든 행: 7(0.004초)
메뉴명 메뉴명_1 메뉴명_2
1 김밥 김밥 (null)
2 쫄면 쫄면 쫄면
3 오징어덮밥 오징어덮밥 (null)
4 잔치국수 (null) 잔치국수
5 돈가스 (null) (null)
6 오므라이스 (null) (null)
7 둘솔비빔밥 (null) (null)
```

**SELECT 천.메뉴명, 나.메뉴명, 우.메뉴명  
FROM 김밥천국 천 LEFT JOIN 김밥나라 나  
ON 천.메뉴명 = 나.메뉴명 LEFT JOIN 우리분식 우  
ON 천.메뉴명 = 우.메뉴명;**

두 번째 조인 조건을 김밥나라에서 김밥천국으로 바꾸어 보기

# 기타 조인

셀프 / 기타 조인 / 기타 조인

## ❖ 다중 테이블 조인

### + 외부 조인 예제

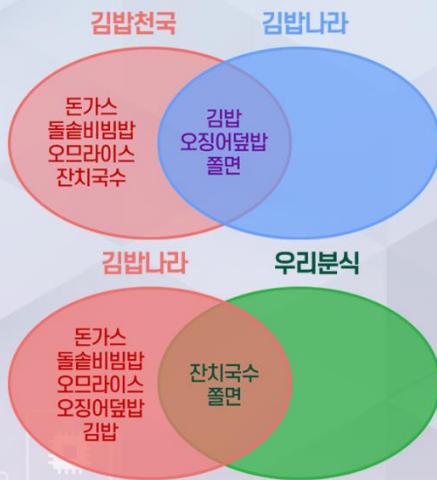
워크시트 | 질의 작성기

```

1 SELECT 천.메뉴명, 나.메뉴명, 우.메뉴명
2 FROM 김밥천국 천 LEFT JOIN 김밥나라 나
3 ON 천.메뉴명 = 나.메뉴명 LEFT JOIN 우리분식 우
4 ON 천.메뉴명 = 우.메뉴명;
  
```

질의 결과 x | SQL | 인출된 모든 행: 7(0.004초)

메뉴명	메뉴명_1	메뉴명_2
1 김밥	김밥	(null)
2 쫄면	쫄면	쫄면
3 오징어덮밥	오징어덮밥	(null)
4 잔치국수	(null)	잔치국수
5 돈가스	(null)	(null)
6 오므라이스	(null)	(null)
7 돌솥비빔밥	(null)	(null)



첫 번째 조인은 변경이 없고, 두 번째 조인은 김밥천국과 우리분식의 LEFT JOIN

셀프 / 기타 조인 / 기타 조인

## ❖ 다중 테이블 조인

### + 외부 조인 예제

#### LEFT JOIN을 다중 조인할 때 주의할 점

- 조인을 2개가 아닌 3개 이상의 테이블을 조인하는 것은 어렵고 까다로운 작업
- INNER JOIN과는 달리 LEFT JOIN은 조인하는 테이블의 순서가 상당히 중요
- 어떤 순서로 테이블을 조인하는지에 따라 결과 테이블에 조회되는 행의 개수와 구성 등이 달라짐
- 3개 이상의 테이블을 LEFT JOIN 할 경우에는 다음 2가지를 꼭 기억
  - **가장 첫 번째의 테이블은 SELECT 문에 가장 많은 열을 가져와야 할 테이블을 먼저 적을 것**
  - **시작을 LEFT JOIN으로 했다면 나머지 조인도 LEFT JOIN을 할 것**  
(LEFT JOIN을 쓰다가 갑자기 RIGHT JOIN이나 다른 조인을 사용하면 안됨)

# 기타 조인

셀프 / 기타 조인 / 기타 조인

## 다중 테이블 조인

### 외부 조인 예제

워크시트 질의 작성기

```

1 SELECT 천.메뉴명, 나.메뉴명, 우.메뉴명
2 FROM 김밥천국 천 LEFT JOIN 김밥나라 나
3 ON 천.메뉴명 = 나.메뉴명 RIGHT JOIN 우리분식 우
4 ON 천.메뉴명 = 우.메뉴명;
    
```

메뉴명
쫄면
잔치국수
(null)
(null)
(null)

SELECT 천.메뉴명, 나.메뉴명, 우.메뉴명  
 FROM 김밥천국 천 LEFT JOIN 김밥나라 나  
 ON 천.메뉴명 = 나.메뉴명 RIGHT JOIN 우리분식 우  
 ON 천.메뉴명 = 우.메뉴명;

1번째 조인을 김밥천국 LEFT로 하여  
 김밥천국의 메뉴명이 다 나오기를 기대함

2번째 조인을 RIGHT로 변경하였기 때문에  
 원하지 않는 결과를 얻음

두 번째 조인을 LEFT에서 RIGHT로 변경한 경우

셀프 / 기타 조인 / 기타 조인

## 다중 테이블 조인

### 다중 테이블 조인을 정리하면

- INNER JOIN은 각 테이블에서 조건에 맞는 값만 보여 줌
- OUTER JOIN은 어떤 테이블을 주 테이블로 하느냐  
 즉, 모두 보여줄 테이블이 어떤 것인가에 따라  
 LEFT JOIN, RIGHT JOIN, FULL JOIN으로 분류
- 따라서 주 테이블을 무엇으로 지정하느냐에 따라  
 조인 결과가 달라지니 주의해야 함
- 조인을 할 때 기준이 되는 주 테이블과 그 관계에 대해  
 잘 계산을 하면, 테이블이 3개나 4개 이상이라고  
 하더라도 그 관계를 해석하고 설계할 수 있음

# 실습하기

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays a tree view of database objects under '19c remote'. The central workspace shows a script editor with the following code:

```
--셀프 조인 (Self Join)
--셀프 조인은 내부나 외부 조인처럼 많이 사용되지는 않습니다.
--셀프 조인은 말 그대로 테이블을 조인할 때 자기 자신을 조인하는 것입니다.

--셀프 조인 특징
--조인 관계가 자기 자신을 상대로 합니다.
--즉, 자기 자신을 대상으로 조인을 맞는 것이기 때문에 FROM 절에서
--별명을 다르게 지정하여 논리적으로 서로 다른 테이블인 것 처럼 사용합니다.
--조회하려는 컬럼을 지정하는 SELECT 문의 컬럼 앞에도 별명을 지정하여야 합니다.

--셀프 조인 예는 오라클에 있는 Employees 테이블로 하겠습니다.

--아래에 사원ID와 사원명 그리고 그 사원을 관리하는 매니저ID와 매니저명을 찾는 셀프 조인 SQL문을 작성하였습니다.

SELECT el.employee_id AS 사원ID, el.first_name AS 사원명
, e2.employee_id AS 매니저ID, e2.first_name AS 매니저명
```

The status bar at the bottom right indicates "작업이 완료되었습니다.(0.042초)". A yellow sticky note in the center says "셀프 조인".

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

## 실습단계

셀프 조인

테이블을 조인할 때 자기 자신을 조인

셀프 조인

SELECT 문에도 컬럼 앞에 별명을 지정

하나의 행에 있는 정보만으로는 매니저ID는 알 수 있지만, 매니저 이름은 알 수 없음

FROM에 있는 employees 테이블의 별명을 e1으로 지정

INNER JOIN에 있는 동일한 employees 테이블의 별명을 e2로 지정

내부 조인의 경우 NULL값은 검색에서 제외

외부(LEFT, RIGHT, FULL)조인은 NULL 값도 조회가 됨

다중 테이블 조인

INNER JOIN에서 INNER는 생략

외부(LEFT, RIGHT, FULL)조인에서 OUTER는 생략

“우리분식” 테이블 생성

# 실습하기

## 실습단계

테이블 새로 고침

외부 조인

외부 조인은 조인 순서가 중요!

2번째 조인 조건을 김밥나라에서 김밥천국으로 바꾸면 결과는?

LEFT JOIN을 다중 조인할 때 주의할 점