

[3-3. JOIN을 활용한 여러 테이블의 데이터 결합]

1. JOIN

1-1. JOIN의 기본 개념

- JOIN이란 두 개 이상의 테이블에서 관련된 데이터를 서로 결합하여 하나의 결과 집합으로 만들어 내는 것을 말합니다.
- 각 테이블에서 공통적인 열(컬럼)을 기준으로 데이터를 결합하며, 이 과정을 통해 관련 정보를 통합하여 조회할 수 있습니다.

1-2. JOIN의 종류

- INNER JOIN : 두 테이블의 교집합을 반환합니다. 즉, 조건에 맞는 행이 양쪽 테이블에 모두 존재할 때만 결과에 포함됩니다.
- LEFT JOIN(LEFT OUTER JOIN) : 왼쪽 테이블의 모든 행과 오른쪽 테이블의 조건에 맞는 행을 반환합니다. 오른쪽 테이블에 매칭되지 않는 경우, 해당 필드는 NULL로 채워집니다.
- RIGHT JOIN(RIGHT OUTER JOIN) : LEFT JOIN의 반대로, 오른쪽 테이블의 모든 행과 왼쪽 테이블의 조건에 맞는 행을 반환합니다.
- FULL OUTER JOIN : 두 테이블의 합집합을 반환합니다. 양쪽 테이블 중 하나에라도 매칭되는 행이 있는 경우 결과에 포함됩니다. 한 쪽에만 존재하는 행은 NULL로 채워진 채로 포함됩니다.
- CROSS JOIN : 두 테이블 간의 모든 가능한 조합을 반환합니다. 명시적인 결합 조건이 없으며, 두 테이블의 각 행이 서로 결합됩니다.

2. INNER JOIN

2-1. 기본 구조

```
SELECT columns
FROM table1
INNER JOIN table2
ON table1.common_column = table2.common_column;
```

2-2. 사용 예시

사용 예시

```
SELECT *
FROM Rooms
INNER JOIN Sales
ON Rooms.room_number = Sales.room_number;
```

3. LEFT JOIN(LEFT OUTER JOIN)

3-1. 기본 구조

```
SELECT columns
FROM table1
LEFT JOIN table2
ON table1.common_column = table2.common_column;
```

3-2. 사용 예시

샘플데이터 추가 입력

```
INSERT INTO Rooms (room_number, room_type,
status_type)
VALUES
    (301, 'A', "판매가능"),
    (302, 'B', "판매가능"),
    (303, 'C', "판매가능");
```

사용 예시

```
SELECT *
FROM Rooms
LEFT JOIN Sales
ON Rooms.room_number = Sales.room_number;
```

4. RIGHT JOIN(RIGHT OUTER JOIN)

4-1. 기본 구조

```
SELECT columns  
FROM table1  
RIGHT JOIN table2  
ON table1.common_column = table2.common_column;
```

4-2. 사용 예시

사용 예시

```
SELECT *  
FROM Rooms  
RIGHT JOIN Sales  
ON Rooms.room_number = Sales.room_number;
```