

[3-2. 그룹화 및 조건부로직]

1. GROUP BY

1-1. GROUP BY 개요

- GROUP BY 절은 SQL에서 집계함수(예: COUNT(), SUM(), AVG(), MAX(), MIN() 등)와 함께 사용되어, 하나 이상의 열(컬럼)을 기준으로 결과 세트를 그룹화하는 데 사용됩니다.
- 이를 통해 각 그룹에 대한 집계된 정보를 조회할 수 있습니다.

1-2. 기본 구조

```
SELECT 컬럼명1, 집계함수(컬럼명2)
FROM 테이블명
GROUP BY 컬럼명1;
```

- 컬럼명1 : 그룹화의 기준이 되는 열입니다. SELECT 절에 나열된 컬럼 중 집계 함수를 사용하지 않는 경우 GROUP BY 절에도 명시되어야 합니다.
- 집계함수 : 각 그룹에 대해 계산될 집계함수입니다.

1-3. 사용 예시

사용 예시 1 : 방당 평균 판매가격 계산

```
SELECT room_number, AVG(sale_price)
FROM Sales
GROUP BY room_number;
```

사용 예시 2 : 날짜별 판매개수 및 매출 합계

```
SELECT sale_date, COUNT(room_number), SUM(sale_price)
FROM Sales
GROUP BY sale_date;
```

1-4. HAVING 절과 함께 사용하기

- HAVING 절은 GROUP BY 절과 함께 사용되어, 그룹화된 결과에 대해 추가적인 필터링 조건을 적용합니다.
- WHERE 절과 달리 HAVING 절은 집계함수와 함께 사용할 수 있으며, WHERE 절이 개별 레코드에 대한 조건을 적용하는 것과 달리, HAVING 절은 그룹에 대한 조건을 적용합니다.

사용 예시 : 날짜별 판매금액이 80만원 이상인 경우에 대하여 필터링 조건 적용

```
SELECT sale_date, COUNT(room_number),  
SUM(sale_price)  
FROM Sales  
GROUP BY sale_date  
HAVING SUM(sale_price) >= 800000;
```

1-5. HAVING 절과 WHERE 절의 차이점

구분	내용
WHERE 절	데이터를 그룹화하기 전에 개별 레코드에 대한 필터링 적용. 집계 함수와 함께 사용할 수 없음.
HAVING 절	데이터를 그룹화한 후, 그 그룹에 대한 조건을 적용. 집계함수와 함께 사용 가능.

2. CASE

2-1. CASE 문 개요

- CASE 문은 SQL에서 조건에 따라 다른 값을 반환하도록 설계된 조건부 로직입니다.
- 이는 프로그래밍 언어의 "if-else" 조건문과 유사한 기능을 제공하며, 'SELECT', 'WHERE', 'ORDER BY', 'GROUP BY' 절 등 다양한 곳에서 사용할 수 있습니다.

2-2. 기본 구조

단순 CASE 문

CASE 컬럼명

WHEN 값1 THEN 결과값1

WHEN 값2 THEN 결과값2

...

ELSE 기본결과값

END

검색 CASE 문

CASE

WHEN 조건1 THEN 결과값1

WHEN 조건2 THEN 결과값2

...

ELSE 기본결과값

END

2-2. 사용 예시

방별 타입 설정

```
SELECT room_number,  
       CASE room_number  
         WHEN 101 THEN 'A Type'  
         WHEN 102 THEN 'B Type'  
         WHEN 103 THEN 'C Type'  
         WHEN 201 THEN 'A Type'  
         WHEN 202 THEN 'B Type'  
         WHEN 203 THEN 'C Type'  
         ELSE 'ETC'  
       END AS room_type,  
       sale_date, sale_price  
FROM Sales;
```

판매가격 별 등급 설정

```
SELECT room_number, sale_date, sale_price,  
       CASE  
         WHEN sale_price >= 150000 THEN 'High'  
         WHEN sale_price < 150000 AND sale_price >=  
120000 THEN 'Middle'  
         WHEN sale_price < 120000 THEN 'Low'  
       END AS grade  
FROM Sales;
```