

ORDER BY 절 개념서

이 문서는 SQL의 ORDER BY 절에 대한 포괄적인 개념서입니다. ORDER BY 절의 기본 개념, 주요 특징, 다양한 사용법, 성능 고려사항, 그리고 주의사항 및 제약사항을 상세히 다룹니다. 또한 SQLD 시험 대비를 위한 팁도 포함하고 있어, SQL 학습자와 데이터베이스 전문가 모두에게 유용한 자료입니다.

1. ORDER BY 절의 기본 개념

1.1 정의

- 조화된 데이터를 특정 컬럼을 기준으로 정렬하는 구문
- SELECT 문의 가장 마지막에 위치
- SQL 문장의 실행 순서에서 마지막에 수행

1.2 기본 문법

```
SELECT 컬럼1, 컬럼2, ...  
FROM 테이블명  
[WHERE 조건식]  
[GROUP BY 절]  
[HAVING 조건식]  
ORDER BY 정렬기준컬럼 [ASC|DESC][, 정렬기준컬럼2 [ASC|DESC], ...]
```

2. ORDER BY 절의 주요 특징

2.1 정렬 방식

1. ASC (Ascending)

- 오름차순 정렬 (기본값)
- 숫자: 작은 수에서 큰 수로
- 문자: 사전순서
- 날짜: 과거에서 최근 순

2. DESC (Descending)

- 내림차순 정렬
- 숫자: 큰 수에서 작은 수로
- 문자: 사전 역순
- 날짜: 최근에서 과거 순

2.2 NULL 값의 정렬

- Oracle
 - NULL 값은 오름차순(ASC)에서 마지막에 위치
 - NULL 값은 내림차순(DESC)에서 처음에 위치
- SQL Server
 - NULL 값은 오름차순(ASC)에서 처음에 위치
 - NULL 값은 내림차순(DESC)에서 마지막에 위치

```
-- NULL 값의 위치 지정  
ORDER BY 컬럼명 [NULLS FIRST | NULLS LAST]
```

3. ORDER BY 절의 다양한 사용법

3.1 단일 컬럼 정렬

```
-- 기본 사용  
SELECT 사원번호, 이름, 급여  
FROM 사원  
ORDER BY 급여 DESC;  
  
-- NULL 처리  
SELECT 사원번호, 이름, 커미션  
FROM 사원  
ORDER BY 커미션 NULLS LAST;
```

3.2 다중 컬럼 정렬

```
-- 부서별로 먼저 정렬하고, 같은 부서 내에서는 급여 순으로 정렬  
SELECT 사원번호, 이름, 부서번호, 급여  
FROM 사원  
ORDER BY 부서번호 ASC, 급여 DESC;
```

3.3 표현식을 이용한 정렬

```
-- 연봉(급여*12) 기준으로 정렬  
SELECT 사원번호, 이름, 급여  
FROM 사원  
ORDER BY 급여*12 DESC;  
  
-- 문자열 결합 결과로 정렬  
SELECT 사원번호, 성, 이름  
FROM 사원  
ORDER BY 성 || 이름;
```

3.4 컬럼 순번을 이용한 정렬

```
-- SELECT 절의 컬럼 순서를 기준으로 정렬  
SELECT 사원번호, 이름, 급여  
FROM 사원  
ORDER BY 3 DESC; -- 3번째 컬럼(급여)으로 정렬
```

3.5 함수를 이용한 정렬

```
-- 이름 길이를 기준으로 정렬  
SELECT 사원번호, 이름  
FROM 사원  
ORDER BY LENGTH(이름);  
  
-- 날짜 함수를 이용한 정렬  
SELECT 사원번호, 입사일  
FROM 사원  
ORDER BY TO_CHAR(입사일, 'YYYY') DESC;
```

1 단일 컬럼 정렬
기본적인 정렬 방식으로, 하나의 컬럼을 기준으로 정렬합니다.

2 다중 컬럼 정렬
여러 컬럼을 기준으로 순차적으로 정렬하는 방식입니다.

3 표현식 정렬
컬럼 값을 가공한 결과를 기준으로 정렬합니다.

4 함수 이용 정렬
SQL 함수를 적용한 결과를 기준으로 정렬합니다.

4. ORDER BY 절의 성능 고려사항

4.1 성능 영향 요소

- 정렬 대상 데이터 량**
 - 정렬할 데이터가 많을수록 성능 저하
- 인덱스 활용**
 - 인덱스가 있는 컬럼으로 정렬 시 성능 향상
 - 인덱스 순서와 정렬 순서가 일치할 때 효과적
- 메모리 사용**
 - 정렬을 위한 추가적인 메모리 작업공간 필요
 - 대용량 데이터 정렬 시 임시 영역 사용

4.2 성능 개선 방안

- 인덱스 활용**

```
-- 인덱스가 있는 컬럼을 정렬 기준으로 사용  
CREATE INDEX emp_salary_idx ON 사원(급여 DESC);  
SELECT * FROM 사원 ORDER BY 급여 DESC;
```

- WHERE 절을 통한 데이터 축소**

```
-- 정렬 전 데이터 양을 줄임  
SELECT * FROM 사원  
WHERE 부서번호 = 10  
ORDER BY 급여 DESC;
```

5. 주의사항 및 제약사항

5.1 주의사항

- SELECT DISTINCT와 함께 사용**
 - DISTINCT 사용 시 ORDER BY 절에 SELECT 목록의 컬럼만 사용 가능
- 컬럼 순번 사용**
 - 유지보수성이 떨어질 수 있음
 - 가급적 컬럼명을 직접 사용 권장
- 집합 연산자와 사용**
 - UNION, INTERSECT 등과 함께 사용 시 마지막에만 적용 가능

5.2 제약사항

- SELECT 문에서만 사용 가능**
 - INSERT, UPDATE, DELETE 문에서는 사용 불가
- LOB 타입 정렬 제한**
 - CLOB, BLOB 등 대용량 데이터 타입은 정렬 불가

1 DISTINCT 사용 시 주의 SELECT DISTINCT와 함께 사용할 때는 ORDER BY 절에 SELECT 목록의 컬럼만 사용해야 합니다.	2 컬럼 순번 사용 지양 컬럼 순번을 사용하면 유지보수성이 떨어질 수 있으므로, 가급적 컬럼명을 직접 사용하는 것이 좋습니다.
3 집합 연산자와 사용 제한 UNION, INTERSECT 등의 집합 연산자와 함께 사용할 때는 마지막에만 ORDER BY 절을 적용할 수 있습니다.	4 LOB 타입 정렬 불가 CLOB, BLOB 등의 대용량 데이터 타입은 ORDER BY 절을 사용하여 정렬할 수 없습니다.

SQLD 시험 대비 TIP

주요 출제 포인트

- 정렬 순서의 이해**
 - ASC, DESC의 개념
 - NULL 값의 정렬 순서
- 다중 컬럼 정렬**
 - 정렬 우선순위
 - 각 컬럼별 정렬 방향
- 특수한 정렬 상황**
 - 표현식을 이용한 정렬
 - 함수를 이용한 정렬

학습 전략

- 기본적인 정렬 구문 숙지
- NULL 값 처리방식 이해
- 다양한 정렬 방식 실습
- 성능 관련 개념 이해