

3장 정렬과 연산

Ybigta 14기 안주영

정렬

- SELECT 열명
FROM 테이블명
(WHERE 조건식)
ORDER BY (DESC | ASC)열명
- 검색 결과의 행 순서를 바꿀 수 있다.
- DESC로 내림차순 정렬 // ASC로 오름차순 정렬
- ORDER BY 구문의 경우는 데이터 출력을 지정해주는 것으로 저장된 데이터의 행 순서를 바꿔주지는 않는다

◦ ORDER BY 단일컬럼 - 기본은 오름차순



TABLE : 고객정보

고객번호	고객명	전화번호	도시명	나이
1000001	유시진	010-0112-0001	서울	33
1000002	강모연	010-0090-1004	서울	33
1000003	서대영	010-0110-0110	정선	36
1000004	윤명주	010-0220-0220	서울	30

SELECT * FROM 고객정보 WHERE 도시명 = '서울' ORDER BY 나이

고객번호	고객명	전화번호	도시명	나이
1000001	유시진	010-0112-0001	서울	33
1000002	강모연	010-0090-1004	서울	33
1000004	윤명주	010-0220-0220	서울	30

도시명 = '서울' 데이터

고객번호	고객명	전화번호	도시명	나이
1000004	윤명주	010-0220-0220	서울	30
1000001	유시진	010-0112-0001	서울	33
1000002	강모연	010-0090-1004	서울	33

나이 순(오름차순)으로 정렬

대소관계

- 수치형 데이터의 경우는 숫자 크기로 대소관계가 정해짐

Ex) $1 < 2 < 10 < 100$

- 날짜 시간형의 경우에도 마찬가지로 숫자의 크기로 대소관계가 정해짐

Ex) 1999년 < 2013년 < 2014년

- 문자열 데이터의 경우 사전순서로 대소관계가 정해짐

Ex) 가방 < 가족 < 나비

*** 문자열 데이터(varchar 등)로 숫자가 들어가 있을 경우 유의 할 것

$1 < 10 < 11 < 2$

복수의 열을 지정해 정렬하기

- SELECT 열명
FROM 테이블명
(WHERE 조건식)
ORDER BY (DESC | ASC)열명1 , (DESC | ASC)열명1
- 각 열명마다 (DESC | ASC)를 지정해 주는 것에 유의

◦ ORDER BY 다중컬럼 + DESC



TABLE : 고객정보

고객번호	고객명	전화번호	도시명	나이
1000001	유시진	010-0112-0001	서울	31
1000002	강모연	010-0090-1004	서울	33
1000003	서대영	010-0110-0110	정선	36
1000004	윤명주	010-0220-0220	서울	30

SELECT * FROM 고객정보 WHERE 고객번호 < 1000005 ORDER BY 도시명, 나이 DESC

• 컬럼명 뒤에 DESC 기술시 해당 컬럼은 내림차순 정렬

고객번호	고객명	전화번호	도시명	나이
1000001	유시진	010-0112-0001	서울	31
1000002	강모연	010-0090-1004	서울	33
1000004	윤명주	010-0220-0220	서울	30
1000003	서대영	010-0110-0110	정선	36

• 도시명으로 오름차순

고객번호	고객명	전화번호	도시명	나이
1000002	강모연	010-0090-1004	서울	33
1000001	유시진	010-0112-0001	서울	31
1000004	윤명주	010-0220-0220	서울	30
1000003	서대영	010-0110-0110	정선	36

• 도시명 정렬 상태에서 동일한 도시명 컬럼에 대해 나이순으로 내림차순 정렬

NULL 값의 경우

- NULL 값의 경우는 가장 먼저 표시되거나 가장 나중에 표시된다. 표준 SQL 규정으로 정의되어 있지 않아서 데이터 베이스 제품에 따라 기준이 다르다.
- MySQL (MariaDB)의 경우 NULL 값을 가장 작은 값으로 취급하여 ASC에서는 가장 먼저, DESC에서는 가장 나중에

결과 행 제한

- SELECT 열명
FROM 테이블명
(ORDER BY 열명)
LIMIT 행수 (OFFSET 시작행)
- (시작행) + 1 번째 행부터 (행수) 개만큼 출력해준다
- OFFSET 절에서 PYTHON 등의 SLICING처럼 첫번째 ROW가 0번 ROW인건 유의

결과 행 제한 기타 DB

- SQL SERVER

```
SELECT TOP 3 *  
FROM sample33;
```

- Oracle

```
SELECT *  
FROM sample 33  
WHERE ROWNUM <=3;
```

** 오라클의 경우 WHERE절에서 지정하기 때문에 정렬하기 전에 처리되어 limit행을 제한한 경우와 결과값이 다를 수 있다.

** 오라클은 12버전부터 OFFSET과 LIMIT가 추가되었다.

수치연산

[표 II-1-13] 산술 연산자의 종류

산술 연산자	설 명
()	연산자 우선순위를 변경하기 위한 괄호 (괄호 안의 연산이 우선된다)
*	곱하기
/	나누기
+	더하기
-	빼기

- 곱셈, 나눗셈, 뺄셈(*,/,%)등이 우선순위 덧셈,뺄셈(+,-)가 후순위
- 함수(MOD, ROUND)등도 사용 할 수 있다

수치연산

- SELECT 구문
 - SELECT 식1, 식2 FROM 테이블명
- ALIAS(별칭)
 - SELECT *, price*quantity AS amount FROM sample34;
 - price*quantity 를 계산한 컬럼에 amount라는 별칭을 줌
- WHERE 구문
 - SELECT *, price*quantity AS amount FROM sample34
WHERE price*quantity >=2000;
 - WHERE 절에서는 ALIAS를 사용 할 수 없다.

수치연산

- ORDER BY 구문
 - SELECT *, price*quantity AS amount FROM sample34
ORDER BY price*quantity;
 - ORDER BY 절에서는 ALIAS를 사용 할 수 없다.

** SQL 실행 순서

WHERE -> SELECT(별명을 생성) -> ORDER BY 이므로 SELECT 절에서 생성한 ALIAS를
WHERE절에서는 사용 불가

- NULL 값이 들어간 연산은 모두 NULL 값을 반환

문자열 연산

- || , + , CONCAT
 - 문자열을 결합해 주는 함수 (PYTHON의 문자열+문자열과 유사)
 - SQL Server : + , Oracle : || , MySQL : CONCAT
- SUBSTRING(문자열, 시작, 끝)
 - 문자열을 시작부터 끝까지 잘라준다
 - SUBSTRING('20140125001', 1, 4) -> '2014'
- TRIM
 - 문자열의 스페이스를 제거해준다
- CHARACTER_LENGTH 함수
 - VARCHAR 등의 문자열 길이를 반환해준다
 - OCTET_LENGTH는 문자열의 길이를 바이트로 계산해준다
 - EUC-KR 에서 ASCII는 1바이트 한글은 2바이트
 - UTF-8 에서 ASCII는 1바이트 한글은 3바이트

날짜 연산

- SELECT CURRENT_TIMESTAMP;
 - 함수를 실행했을 때를 기준으로 시간을 표시
- 날짜 서식
 - TO_DATE('2014/01/25','YYYY/MM/DD')
- SELECT CURRENT_TIMESTAMP + INTERVAL 1 DAY;
 - 함수를 실행했을 때 기준 하루 뒤
- DATEDIFF
 - 날짜 사이의 차이를 계산 가능

CASE 문

- COALESCE(NULL이 아닐때, NULL 일때)
 - SELECT a, COALESCE(a,0) FROM sample37;
 - a가 NULL 이 아닐때는 a, NULL일 경우 0
- CASE WHEN 조건식1 THEN 식1
 [WHEN 조건식2 THEN 식2]
 [ELSE 식3]
END
- CASE 식1 WHEN 식2 THEN 식3
 [WHEN 식4 THEN 식5]
 [ELSE 식6]
END

CASE 문

- NULL의 경우는 '='연산자로 비교 불가능하므로
 - CASE
 - WHEN a = 1 THEN '남자'
 - WHEN a = 2 THEN '여자'
 - WHEN a IS NULL THEN '데이터 없음'
- END;