[3] SELECT 문

\* 교재 51p

목차

1. SELECT문

2. WHERE절(데이터 제한)

2.1. 비교연산자와 논리 연산자

3. ORDER BY절(정렬하기)

1. SELECT 문(SELECT statement)

- 테이블의 데이터를 조회하는 명령문

- SELECT문의 기능

(1) 프로젝션(Projection) : 내가 원하는 열(column) 뽑아내기

(2) 셀렉션(Selection) : 내가 원하는 행(row) 뽑아내기

(3) 조인(Join) : 서로 다른 테이블의 데이터를 함께 가져오기

- 가장 기본적인 조회 기능들

(1) 테이블의 모든 행과 열 조회하기

SELECT \*

FROM 테이블명;

(2) 테이블의 특정 컬럼을 선택하여 조회하기

SELECT 컬럼명, 컬럼명, ...

FROM 테이블명;

(3) 산술 연산자 (+, -, \*, /) 사용하기

SELECT 컬럼명+10, 컬럼명+컬럼명, ...

FROM 테이블명;

(4) 별칭(Alias) 정의하기

- 별칭을 사용해 표시되는 컬럼의 이름을 변경할 수 있음

- 컬럼명 자체가 변하는 것은 아님

- 계산식이 포함되어 길어진 컬럼의 이름을 짧게 만들 수 있음

- 별칭에 공백이나 특수 문자가 있는 경우 " " 사용

- 별칭은 컬럼명 바로 뒤에 적거나 AS 키워드를 사용해 적는다.

SELECT 컬럼명 AS 별칭, 컬럼명 별칭, ...

FROM 테이블명;

(5) 중복 없이 조회하기

SELECT DISTINCT 컬럼명, 컬럼명, ...

FROM 테이블명

- DISTINCT 키워드는 제시된 컬럼의 값을 비교해 값이 동일할 경우

한 번만 조회하게 하는 키워드이다.

- 다수의 컬럼명이 있는 경우,

모든 값이 일치할 경우에만 중복으로 판정한다.

- 테이블 설계 실수로 중복값이 섞일 때가 아니면

평소에는 잘 사용할 기회가 없다.

\* (TIP) Oracle의 NULL

- Java의 null과 Oracle의 null은 조금 다르다.

- Java의 null : 객체를 더 이상 참조하지 않는다는 것을 의미

- Oracle의 null : 아직 결정되지 않은 값을 의미

Oracle의 null은 다른 값과 연산할 경우, 그 결과도 null이 된다.

산술식에 null이 포함되어 있으면 식의 결과는 항상 null이다.

- NULLABLE : 어떤 열에 NULL 값이 들어갈 수 있는지를 결정하는 제약조건

NULLABLE이 NO이면 반드시 구체적인 값이 들어가야만 하는 열.

2. WHERE절 (데이터 제한)

- 행의 제한

- 테이블 전체의 정보를 가져올 일은 거의 없다.

- DB에서 데이터를 가져올 때 필수적인 기능이다.

- WHERE절을 사용한 SELECT문

SELECT 컬럼명, 컬럼명, ...

FROM 테이블명

[WHERE 조건식];

\* WHERE 절은 생략할 수 있다.

\* WHERE 절 조건식의 연산 결과가 TRUE인 행만 조회한다.

\* WHERE 절에서는 SELECT 절에서 정의한 별칭을 사용할 수 없다.

\* 비교 연산자

= 같다

!=, <> 같지 않다.

>, >=, <, <= 대소 비교

\* 조건식

WHERE 컬럼명 비교연산자 값

WHERE 연산식 비교연산자 값

WHERE 연산식 비교연산자 연산식

\* WHERE절의 왼쪽은 가능하면 단순하게 작성한다.

- table 간 참조 관계

- table들 간에도 부모-자식 관계가 형성된다.

- 부모 테이블 : 참조당하는 쪽

- 자식 테이블 : 참조하는 쪽

- 부모-자식 관계는 1:n의 관계

\* 부모 테이블의 정보 1개를 자식 테이블의 여러 행에서 참조한다.

- foreign key는 다른 열의 값을 참조하는 열을 말한다.

2.1. 비교 연산자와 논리 연산자

2.1.1. 비교 연산자 더 알아보기

(1) BETWEEN 연산자

값 BETWEEN 값1 AND 값2

- 값이 값1 이상 값2 이하인 경우 TRUE를 반환한다.

(2) IN 연산자

값 IN (값1, 값2, 값3, ...)

- 값이 ( ) 안에 제시된 값들 중 하나라도 일치하면 TRUE를 반환한다.

(3) LIKE 연산자

값 LIKE 문자 패턴

- 값이 문자 패턴과 일치하면 TRUE를 반환한다.

- 문자 패턴 : 와일드카드가 사용된 표기

\* 와일드카드 : %, \_ 등

% : 0 또는 무한대

\_ : 아무 값

- 예1)

성이 김씨인 사람 조회하기

SELECT name

FROM users

WHERE name LIKE '김%';

- 김, 김구, 김유신, 김수한무거북이와두루미... 모두 허용

- 예2) 성이 '김'이고 이름이 한 글자인 사람 조회하기

SELECT name

FROM users

WHERE name LIKE '김\_';

- 김구, 김김은 조회

- 김, 김유신, 김수한무거북이...는 조회되지 않음

- 예3) 이름 중 '김'이라는 글자가 들어가는 사람 조회하기

SELECT name

FROM users

WHERE name LIKE '%김%';

- 김구, 김김, 김유신, 김수한무.., 양반김, 홍김홍 등등 모두 허용

- 예4) 이름이 4글자인 사람 찾기

SELECT name

FROM users

WHERE name LIKE '\_\_\_\_';

2.1.2. 논리 연산자

(1) AND

조건식의 모든 조건이 TRUE이면 TRUE

(2) OR

조건식의 조건 중 하나라도 TRUE이면 TRUE

(3) NOT

조건식의 연산 결과로 나온 논리값을 반전

\* (Tip) SELECT 문이 길어질 경우 다음과 같이 쓰는 경우도 많다.

SELECT 컬럼명

, 컬럼명, 컬럼명

, 컬럼명

\* (Tip) 참고할 만한 사이트

구루비(gurubee.net)

- DB, SQL 관련 한국 커뮤니티

- 질문과 답변 활용

3. ORDER BY 절(정렬하기)

ORDER BY 컬럼명 [ASC|DESC];

- ASC : 오름차순

- DESC : 내림차순

- ORDER BY 절에서는 SELECT 절에서 정의한 별칭을 사용할 수 있다.

SQL문의 실행 순서

(3) SELECT : 가져올 열을 지정

(1) FROM : 데이터를 가져올 테이블 지정

(2) WHERE : 가져올 행의 조건을 지정

(4) ORDER BY : 가져온 값을 정렬

\* SQL문 실행 순서상 ORDER BY 절이 SELECT 절 뒤에 있기 때문에 가능하다.

\* SQL은 항상 이 순서(구조)대로 실행된다.

그래서 구조화된 질의 언어(Structured Query Language)라고 하는 것.