

1. 데이터베이스

- 필요한 데이터를 중복 최소화하여 컴퓨터 기억 장치 내에 모아놓은 데이터의 집합이다.
- 기능: 정의 기능 / 조작 기능 (검색, 갱신, 삽입, 삭제) / 제어 기능

2. Select문

(1) SQL 문장 작성법

SELECT * FROM TAB; -- 계정 SCOTT이 가지고 있는 테이블들 (*: 모든 필드를 불러온다)

SELECT * FROM EMP; -- EMP 테이블의 모든 열, 모든 행

(2) SQL 문장 실행

<ex - EMP테이블에서 필드명과 별명 불러오기>

SELECT EMPNO AS "사번", ENAME AS "이름", SAL AS "급여", JOB AS "작업" FROM EMP; -- 여기서 ""와 AS는 생략해도 됨

(3) WHERE 조건과 비교연산자

<ex - 급여가 3000인 사람 / 3000이 아닌 사람들의 사번, 이름, 급여 출력하기>

SELECT EMPNO 사번, ENAME 이름, SAL 급여, JOB 작업 FROM EMP WHERE SAL = 3000; -- 급여가 3000일 때

SELECT EMPNO 사번, ENAME 이름, SAL 급여, JOB 작업 FROM EMP WHERE SAL != 3000; -- 급여가 3000이 아닐 때

-- '다르다'는 뜻의 기호: !=, <>, ^=

(4) 논리연산자 (AND, OR, NOT)

<ex - 82년에 입사한 직원의 모든 필드 출력>

SELECT * FROM EMP WHERE HIREDATE >= '82/01/01' AND HIREDATE <= '82/12/31';

(5) 연결연산자(||): 열이나 문자를 연결

<ex>

SELECT ENAME || '은 ' || JOB || '이다.' FROM EMP; => 출력 결과: 테이블에 할당된 각각의 직원명과 직업 출력

ex) SCOTT은 ANALYST이다.

(6) SQL연산자

<ex - 'BETWEEN A AND B': 82년에 입사한 직원의 모든 필드 출력>

SELECT * FROM EMP WHERE HIREDATE BETWEEN 1500 AND 3000; -- 1500과 3000도 포함되는 식이다.

<ex - 'IN': 부서번호가 10, 20인 사원의 모든 필드>

SELECT * FROM EMP WHERE DEPTNO IN (10,20);

<ex - 'LIKE': 이름에 N이 들어가는 사원의 모든 필드>

SELECT * FROM EMP WHERE ENAME LIKE '%N%'; -- cf) 'N%' N으로 시작하는 글자 / '%N' N으로 끝나는 글자

<ex - 'IS NULL' / 'IS NOT NULL': 상여금(COMM)이 없는 사원의 모든 필드>

SELECT * FROM EMP WHERE COMM=0 OR COMM IS NULL; -- NULL은 연산자가 아닌 이런 방법으로 표기한다.

(7) 정렬

1) 오름차순: (ORDER BY 오름차순으로 정렬할 필드 ASC) - ASC는 생략 가능하다.

<ex - 사원명, 급여를 급여 순으로 정렬>

SELECT ENAME, SAL FROM EMP ORDER BY SAL ASC;

SELECT ENAME, SAL FROM EMP ORDER BY SAL;

2) 내림차순: (ORDER BY 내림차순으로 정렬할 필드 DESC)

<ex - 사원명, 급여를 급여가 높은 순으로 정렬하고, 급여가 같으면 입사일 최신 순으로 정렬>

SELECT ENAME, SAL FROM EMP ORDER BY SAL DESC, HIREDATE DESC ;