

ch04. SELECT문의 기본 형식

04-1. 실습용 테이블 살펴보기

DESC 명령어 → 테이블의 구성 살펴보는 기능

바이트 크기와 문자의 관계

- 영문은 한 글자당 1바이트 차지
- 한글은 한 글자당 2바이트 차지

SQL문을 작성할 때 테이블 이름과 열 이름은 대문자로 쓰기를 권장함.

04-2. 데이터를 조회하는 3가지 방법

→ 선택전, 프로젝션, 조인

| 행 단위로 조회하는 선택전

선택전 selection :

- 행 단위로 원하는 데이터를 조회하는 방식
 - 특정 행만 선별하여 조회하는 방식 (SELECT문을 사용함)
- 테이블 전체 데이터 중 몇몇 가로줄의 데이터만 선택할 때 사용한다.

| 열 단위로 조회하는 프로젝션

프로젝션 projection :

- 열 단위로 원하는 데이터를 조회하는 방식
 - 특정 열만을 선별하여 조회하는 방식 (SELECT문을 사용함)

선택션과 프로젝션 함께 사용하기

행과 열을 모두 선별할 때 선택션과 프로젝션을 함께 사용할 수 있다.

두 개 이상의 테이블을 사용하여 조회하는 조인

조인 join :

- 두 개 이상의 테이블을 양옆에 연결하여 마치 하나의 테이블인 것처럼 하는 조회 방식

04-3. SQL의 기본 뼈대, SELECT절과 FROM절

SELECT문 ⇒ SELECT절 + FROM절 형태를 가짐

```
SELECT [조회할 열1 이름] , [열2 이름] , . . . , [열N 이름]  
FROM [조회할 테이블 이름] ;
```

SELECT → 조회할 열 이름 또는 출력할 데이터를 하나 이상 또는 에스터리스크(*)로 전체 열 지정

FROM → 조회할 테이블 이름

* 로 테이블 전체 열 출력하기

```
SELECT * FROM [테이블 이름] ;
```

테이블 부분 열 출력하기

```
SELECT [열 이름] , [열 이름] , [열 이름]  
FROM [테이블 이름] ;
```

04-4. 중복 데이터를 삭제하는 DISTINCT

DISTINCT는 SELECT문으로 데이터를 조회한 후 DISTINCT를 사용하여 데이터의 중복을 제거한다.

SELECT절에 열 이름을 명시하기 전에 선택적으로 사용할 수 있다.

| 열이 한 개인 경우

```
SELECT DISTINCT DEPTNO  
FROM EMP;
```

| 열이 여러 개인 경우

```
SELECT DISTINCT JOB, DEPTNO  
FROM EMP;
```

| ALL로 중복되는 열 제거 없이 그대로 출력하기

```
SELECT ALL JOB, DEPTNO  
FROM EMP;
```

→ 예제의 실행 결과를 보면 DISTINCT와 ALL을 사용하지 않고 SELECT만을 사용하여 조회한 결과랑 같은데, 이는 SELECT절에서 중복 설정이 없을 경우에 ALL을 기본으로 사용하기 때문이다.

04-5. 한 눈에 보기 좋게 별칭 설정하기

- SQL문에서는 최종 출력되는 열 이름을 임의로 지정할 수 있다.
- 본래 열 이름 대신 붙이는 이름을 별칭 (alias) 라고 한다.

결과 연산식

```
ex) SELECT ENAME, SAL, SAL*12+COMM, COMM  
      FROM EMP;
```

별칭을 지정하는 방식



SAL*12+COMM ANNSAL

1. 연산 및 가공된 문장 이후 한 칸 띄우고 별칭 지정



SAL*12+COMM "ANNSAL"

1. 연산 및 가공된 문장 이후 한 칸 띄우고 별칭을 큰따옴표로 묶어 지정



SAL*12+COMM AS ANNSAL

1. 연산 및 가공된 문장 이후 한 칸 띄운 후 'AS', 한 칸 뒤에 별칭 지정



SAL*12+COMM AS "ANNSAL"

1. 연산 및 가공된 문장 이후 한 칸 띄운 후 'AS', 한 칸 뒤에 별칭을 큰 따옴표로 묶어 지정

✓ 별칭은 최종적으로 출력되기를 원하는 열 이름을 직접 지정할 때 주로 사용함

✓ 보안이나 데이터 노출 문제 때문에 별칭을 사용하기도 함

04-6. 원하는 순서로 출력 데이터를 정렬하는 ORDER BY

데이터 조회를 할 때 시간이나 이름 순서 또는 어떤 다른 기준으로 데이터를 정렬해서 출력해야 하는 경우가 있다.

⇒ ORDER BY절은 SELECT문을 작성할 때 할 수있는 여러 절 중 가장 마지막 부분에 쓴다.

```
SELECT [조회할 열1 이름], [열2 이름], ... , [열N 이름]
FROM [조회할 데이터 이름]
...
... (그 밖의 절)
...
ORDER BY [정렬하려는 열 이름(여러 열 지정 가능)] [정렬 옵션] ;
```

내림차순 사용하기

```
SELECT *
FROM EMP
ORDER BY SAL DESC ;
```

VARCHAR2 같은 문자 데이터 역시 알파벳순으로 정렬할 수 있으며,

날짜 데이터를 의미하는 DATE 역시 이전 날짜, 이후 날짜로 크기를 부여하여 정렬할 수 있다.

각각의 열에 내림차순과 올림차순 동시에 사용하기

```
SELECT *
FROM EMP
ORDER BY DEPTNO ASC, SAL DESC ;
```

❌ ORDER BY 는 많은 자원(비용)을 소모하고 결과출력을 늦추기 때문에 꼭 필요한 경우가 아니라면 쓰지 않는 것이 좋다.