

4강 SELECT문의 기본 형식

04-1 실습용 테이블 살펴보기

Aa 테이블 이름	≡ 테이블에 들어 있는 정보
EMP	사원 정보
DEPT	회사 부서 정보
SALGRADE	사원들의 급여 정보

- SQL문을 작성할 때 **테이블 이름**과** 열 이름은 대문자**로 쓰기를 권장한다.
=> SQL문은 대, 소문자를 구분하지 않기 때문에 다른 프로그래밍 언어와 SQL문을 구분하고 가독성을 높이기 위함

04-2 데이터를 조회하는 3가지 방법 - 선택, 프로젝션, 조인

행 단위로 조회하는 선택

- 선택? 행 단위로 데이터를 조회하는 방식
- 테이블 전체 데이터 중 몇몇 **가로줄**의 데이터만 선택할 때 사용

행 단위로 조회하는 선택

선택(selection)은 행 단위로 원하는 데이터를 조회하는 방식입니다. 테이블 전체 데이터 중 몇몇 가로줄의 데이터만 선택할 때 사용합니다.

열 1	열 2	...	열 N
행 1			
행 2			
...		...	
행 N			

SELECT문을 사용하여 특정 행만 선별하여 조회하는 방식이 선택

열 단위로 조회하는 프로젝션

- **프로젝션?** 열 단위로 원하는 데이터를 조회하는 방식!

열 단위로 조회하는 프로젝트

프로젝션(projection)은 열 단위로 원하는 데이터를 조회하는 방식입니다.

열 1	열 2	...	열 N
행 1			
행 2			
...		...	
행 N			

SELECT문을 사용하여 특정 열만을 선별하여 조회하는 방식이 프로젝트

선택과 프로젝트 함께 사용하기

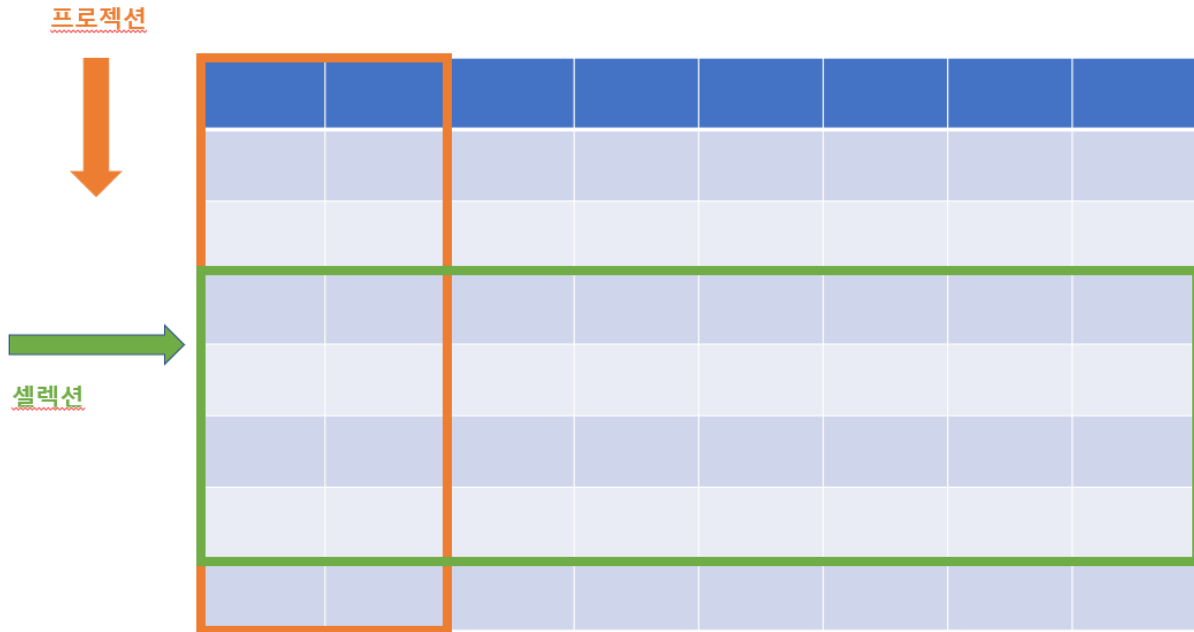
- 특정 테이블에서 조회하려는 행과 열을 모두 선별할 때 선택과 프로젝트를 함께 사용할 수 있다.
- 예; 전체 학생 중 졸업생만을 조회함을 동시에 학번, 이름, 학과 코드 항목만 출력되도록 선택

선택과 프로젝트 함께 사용하기

특정 테이블에서 조회하려는 행과 열을 모두 선별할 때 선택과 프로젝트를 함께 사용할 수 있습니다.

열 1	열 2	...	열 N
행 1			
행 2			
...			
행 N			

선택과 프로젝트를 함께 사용하면 원하는 행의 특정 열만 조회가 가능!



두 개 이상의 테이블을 사용하여 조회하는 조인

- **조인?** 두 개 이상의 테이블을 양옆에 연결하여 마치 하나의 테이블인 것처럼 데이터를 조회하는 방식
- 외래키를 이용하여 하나의 테이블 처럼 사용할 수 있다.
- 여러 테이블의 데이터를 하나의 테이블처럼 조회할 수 있다.

두 개 이상의 테이블을 사용하여 조회하는 조인

조인(join)은 두 개 이상의 테이블을 양옆에 연결하여 마치 하나의 테이블인 것처럼 데이터를 조회하는 방식입니다. 02장에서 외래키를 알아보며 학생 정보와 학과 정보를 하나의 테이블처럼 연결하여 사용하는 예를 살펴보았습니다. 그 예와 같이 조인은 관계형 데이터베이스에서 흔히 사용하는 방식으로 여러 테이블의 데이터를 하나의 테이블처럼 조회할 수 있습니다.

열 1	열 2	...	열 N	열 1	열 2	...	열 N
행 1				행 1			
행 2				행 2			
...				...			
행 N				행 N			

사이좋게 하나의 테이블인 것처럼 붙여서 사용하면 조인

SQL의 기본 뼈대, SELECT절과 FROM절

- SELECT문은 데이터 베이스에 보관되어 있는 **데이터를 조회**하는데 사용된다.
- SELECT문의 기본 구성: SELECT절 + FROM절
- **FROM절**: 조회할 데이터가 저장된 **테이블 이름**을 명시
- **SELECT절**은 FROM절에서 명시한 테이블에서 조회할 **열**이나 여러 열에 저장된 **데이터의 조합 또는 연산식**을 지정할 수 있다.

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

Aa 키워드	필수 요소	설명
<u>SELECT</u>	조회할 열 이름 또는 출력할 데이터를 하나 이상 지정 또는 *(애스터리스크)로 전체 열을 저장	SELECT절, 조회할 열을 지정
<u>FROM</u>	조회할 테이블 이름	FROM절, 조회할 테이블을 지정

- 문장 끝에는 항상 ; 붙여야 함!

*로 테이블 전체 열 출력하기

```
SELECT * FROM EMP;
```

테이블 부분 열 출력하기

```
SELECT EMPNO, ENAME, DEPTNO FROM EMP;
```

- SELECT문은 기본적으로 출력되는 데이터의 **정렬 순서를 보장 X** => 정렬을 위해서는 ORDER BY절을 이용한다.
- 위의 SQL문 예시를 보면 각각의 열 이름을 ,(쉼표)를 이용하여 구분하고 있다.
- 데이터를 제외한 **SQL문의 띄어쓰기와 줄 바꿈은 명령 수행에 영향 X**.

04-4 중복 데이터를 삭제하는 DISTINCT

- SELECT문으로 데이터를 조회한 후 DISTINCT를 사용하여 **중복을 제거**한다.
- DISTINCT는 **SELECT절에 열 이름을 명시하기 전에** 선택적으로 사용할 수 있다.
- DISTINCT는 조회한 데이터의 내용에서 불필요한 중복을 제거하고 특정 데이터 종류만 확인하고싶을 때 유용하다.

DISTINCT로 열 중복 제거하기(열이 한 개인 경우)

- DISTINCT를 사용하면 SQL문의 출력 결과로 SELECT절에 명시한 열 중에서 같은 내용이 두 개 이상일 경우, **중복 행은 한 개만 남겨 두고 그 밖의 행은 모두 제거**한다.

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

DISTINCT로 열 중복 제거하기(열이 여러 개인 경우)

- 두 개 이상의 열 중 하나의 열만 볼 때는 중복이 발생해도, 두 개이상의 열을 보게 되면 중복이 아니게 될 경우, 중복 데이터가 아니게 되어 제거 되지 않는다.

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

ALL로 중복되는 열 제거 없이 그대로 출력하기

- ALL은 DISTINCT와 반대로 데이터 중복을 제거하지 않고 그대로 출력한다.
- SELECT절에서 중복 설정이 없을 경우, ALL을 기본으로 사용한다.

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

04-5 한눈에 보기 좋게 별칭 설정하기

- **별칭(alias)?** SQL문에서는 최종 출력되는 열 이름을 임의로 지정할 수 있는데, 본래 열 이름 대신 붙이는 이름을 별칭이라고 한다.

열과 연산식

- SELECT 다음에 열 이름 대신, 연산식이 들어갈 수 있다.

```
SELECT ENAME, SAL, SAL*12+COMM, COMM FROM EMP; //예시의 연산식의 의미는 연봉
```

- SELECT절에 *을 하나만 사용했을 경우, 모든 열을 조회하라는 의미이지만, 숫자나 다른 열과 함께 사용할 경우 곱하라는 뜻이 된다.

별칭을 지정하는 방식

- 오라클에서 별칭을 지정하는 방식은 4가지 방식이 존재한다.
(SAL*12+COMM의 별칭을 annual salary(연봉)의 약어인 ANNSAL로)

# 번 호	≡ 사용 방법	Aa 설명
1	SAL*12+COMM ANNUAL	<u>연산 및 가공된 문장 이후 한 칸 띄우고 별칭 지정</u>
2	SAL*12+COMM "ANNUAL"	<u>연산 및 가공된 문장 이후 한 칸 띄우고 별칭을 큰 따옴표(" ")로 묶어 지정</u>

# 번 호	≡ 사용 방법	Aa 설명
3	SAL*12+COMM AS ANNUAL	<u>연산 및 가공된 문장 이후 한 칸 띄우고 'AS', 한 칸 뒤에 별칭 지정</u>
4	SAL*12+COMM AS "ANNUAL"	<u>연산 및 가공된 문장 이후 한 칸 띄우고 'AS', 한 칸 뒤에 큰따옴표(""). 로 묶어 별칭 지정</u>

```
SELECT ENAME, SAL, SAL*12+COMM AS ANNSAL, COMM FROM EMP;
```

- 별칭은 최종적으로 출력되기를 원하는 열 이름을 직접 지정할 때 주로 사용한다.
- 단순히 긴 열 이름의 불편함뿐만 아니라, 보안이나 데이터 노출, 데이터 처리 진행 과정을 숨기는 용도로도 별칭을 사용한다.

실무에서의 별칭 지정

- 실무에서는 별칭을 지정하는 4가지 방식 중, 세 번째 방식(AS 사용)을 선호하는 경향이 있다.
 1. 어떤 단어가 별칭인지 알아보기 편하다.
 2. 큰 따옴표를 사용하지 않으려는 프로그래머들의 습관
 - > 프로그래밍 언어에서는 큰 따옴표는 문자열 데이터의 시작과 끝을 알리는 기호로 이미 사용되고 있기 때문
 - ==> 따라서, 코드에 사용한 큰 따옴표가 SQL문에서 사용한건지, 아니면 기존 프로그래밍 코드에서 문법으로 사용한 건지를 구별하는 추가 작업이 필요하다.

(밑의 코드는 자바에서 SQL문을 가져올 때의 코드이다.)

```
String sql = "SELECT ENAME, SAL, SAL*12+COMM AS ANNSAL, COMM FROM EMP";
```

하지만 위에 ANNSAL을 큰 따옴표로 감싸게 되면, 원래 있던 큰 따옴표와 헷갈리게 되어 오류나 예외 상황이 발생할 수 있다.

04-6 원하는 순서로 출력 데이터를 정렬하는 ORDER BY

- SELECT문을 사용하여 데이터를 조회할 때 시간이나 이름 순서 또는 어떤 다른 기준으로 데이터를 정렬해서 출력해야 하는 경우가 발생하게 되는데, 이때 데이터를 정렬된 상태로

출력하기 위해 **ORDER BY절**을 사용한다.

- ORDER BY절은 **SELECT문을 작성할 때** 사용할 수 있는 여러 절 중 **가장 마지막 부분**에 사용한다.

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

오름차순 사용하기

- ORDER BY절에는 **정렬기준**이 되는 **열** 이름을 지정한다.
- 열 이름은 **하나 또는 여러 개의 열**을 지정할 수 있다.

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

내림차순 사용하기

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

- VARCHAR2와 같은 문자 데이터 역시 알파벳순(사전 순서)으로 정렬할 수 있으며, 날짜 데이터를 의미하는 DATE 역시 이전 날짜, 이후 날짜로 크기를 부여하여 정렬할 수 있다.

각각의 열에 내림차순과 올림차순 동시에 사용하기

- ORDER BY절에는 우선순위를 고려하여 여러 개의 정렬 기준을 지정할 수 있다.
- 예; 부서 번호(DEPTNO)를 오름차순으로 정렬하고, 부서 번호가 같은 사원일 경우 급여(SAL)를 기준으로 내림차순으로 정렬할 수 있다.

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

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


ORDER BY절을 사용할 때 주의 사항

- ORDER BY절을 사용한 정렬은 꼭 필요한 경우가 아니면 사용하지 않는 것이 좋다.
==> 많은 자원, 비용을 소모하기 때문!