

SQL

- sql은 대소문자를 가리지 않는다
 - select 와 SELECT 는 동일하다
- sql은 줄을 한줄 또는 여러줄 가리지 않는다
 - 가독성 편집의 용이성을 위해 여러줄로 사용하자
 - 명령어 한개는 여러줄로 나눌 수 없다
- 들여쓰기도 가리지 않는다
 - 가독성과 편집의 용이성을 위해 들여쓰기를 사용하자
- 명령어는 대문자로 작성하고 나머지는 전부 소문자로 작성하면 가독성이 좋아진다

1. SELECT

전부 조회

SELECT *

FROM employees; → employees테이블로 부터 모든 데이터를 조회하라

- * 은 전체 (All) 모두를 의미한다
- 줄바꿈을 하든 안하든 상관없다
- from 은 어느 테이블로부터 라는 의미로 뒤에 테이블의 이름을 적는다
- 가독성을 위해서 from 부터는 줄을 바꾼다

일부 조회

SELECT employee_id, first_name, last_name
FROM employees;

- select 명령어 뒤에 나열한 열의 이름 순으로 결과 조회된다
- 열의 이름들은 , 를 붙여 계속해서 나열할 수 있다
- * 대신 모든 열의 이름을 나열하면 같은 결과를 얻을 수 있다.

ORDER BY 출력 결과 행 정렬

```
SELECT employee_id, first_name, last_name  
FROM employees  
ORDER BY employee_id DESC;
```

- desc 내림차순이며 지정하지 않으면 default 는 오름차순이다

중복된 값 제거하기...DISTINCT

```
SELECT DISTINCT job_id  
FROM employees ;
```

- 중복된 값들이 그대로 출력되어 어떤 job_id 종류가 있는지 한눈에 알아보기 힘들다
- 중복된 값을 제거하여 고유한 값들만 출력해보자
- distinct 명령어를 사용해서 데이터 값이 종류별로 하나씩만 출력되는 것을 볼수 있다

효율적인 사용을 위한 별명 사용

- SELECT 문의 결과를 출력할때 일반적으로 열의 이름은 테이블 정의할 때 사용된 열의 제목이 출력된다
- 임의로 이 열의 이름을 바꿔서 사용할 수 있다.

AS 열이름을 변경할때는 AS 사용한다.

- 생략도 가능.
- 별명은 열의 이름을 임시로 변경할 때 사용한다.
- 물리적으로 열이름 자체를 변경하는 것은 아니다
- " " 붙여서 사용할 수도 있다

```
SELECT employee_id AS 사원번호, first_name AS 이름,  
last_name AS 성  
FROM employees
```

데이터 값 연결하기

각 열들에 따로 담겨져 있는 데이터를 하나로 붙이거나 추가 수식을 붙여 출력할 수 있다 || 연결 연산자

```
SELECT employee_id, first_name||last_name  
FROM employees  
공백없이 붙여짐
```

```
SELECT employee_id, first_name||' '||last_name
FROM employees
    공백이 붙여짐
```

이메일에 골뱅이 붙이기

```
SELECT employee_id,
       first_name||' '||last_name,
       email||'@'||'naver.com'
FROM employees
```

- 연결연산자 || 와 함께 필드들을 연결하여 표현할 수 있다
- 다양한 값들을 넣어서 다양하게 표현할 수 있다
- 보기 좋게 만들 수 있다

산술 처리, 계산

데이터를 사전에 가공하거나 보고서 작성목적으로 데이터 값들을 계산하여 나타낼 수 있다

```
SELECT employee_id, salary, salary+500, salary*1.1
FROM employees;
```

단순히 계산을 해서 출력한 결과일 뿐 새로운 열을 만들어 데이터베이스에 추가한 것은 아니다

산술 연산자는 업무용으로 많이 쓰인다

WHERE 를 통해 행에서 특정 데이터 값을 조회하거나 비교 처리한다.
WHERE 는 단어 뜻처럼 조건을 지정해서 어디에서 어떻게 가져올지 결정한다

- WHERE 절을 통해서 조회하려는 조건을 정하고
- FROM 절 다음에 위치한다

```
SELECT 열이름
FROM 테이블이름
WHERE 조건;
```

```
SELECT *
FROM employees
WHERE employee_id =100;
```

숫자는 별다른 처리 없이 연산자 뒤에 입력한다

문자는 " 를 사용한다
이 경우에는 대소 문자를 구분한다

ex) 'austin' 과 'Austin' 은 다르게 취급한다

```
SELECT *  
FROM employees  
WHERE first_name='David'; // 같은 경우
```

```
SELECT *  
FROM employees  
WHERE salary BETWEEN 10000 AND 20000; // A~B 까지
```

```
SELECT *  
FROM employees  
WHERE salary IN(10000, 17000, 24000); // 같다가 여러번
```

```
SELECT *  
FROM employees  
WHERE job_id LIKE 'AD%'; // AD로 시작하는 즉 AD를 포함하는
```