

# SELECT로 특정 데이터 추출하기

## WHERE 조건과 비교 연산자

---

### ❖ 사원 테이블의 모든 정보를 출력하는 쿼리문

```
SELECT *  
FROM employees;
```

### ❖ 사원 테이블에서 사원번호, 사원명, 급여 컬럼만 표시하는 쿼리문

```
SELECT employee_id, first_name || ' ' || last_name, salary  
FROM employees;
```

### ❖ 급여를 100000이상 받는 사원에 대해서만 출력

```
SELECT employee_id, first_name || ' ' || last_name, salary  
FROM employees  
WHERE salary >=100000;
```

## WHERE 조건과 비교 연산자

### ❖ 비교 연산자

연산자	의미	예제
=	같다	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL=3000;
>	보다 크다	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL>3000;
<	보다 작다	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL<3000;
>=	보다 크거나 같다	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL>=3000;
<=	보다 작거나 같다	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL<=3000;
<>, !=, ^=	다르다	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL<>3000;

# WHERE 조건과 비교 연산자

---

## ❖ 비교 연산자

- 부서번호가 10인 사원 정보 추출

```
SELECT *  
FROM employees  
WHERE department_id = 10;
```

## ❖ 실습

- 급여가 3000이하인 사원의 사원번호, 사원명, 급여를 출력하는 SQL문을 작성하시오.

# WHERE 조건과 비교 연산자

---

## ❖ 문자 데이터 조회

- 문자열과 날짜는 반드시 따옴표로 묶어서 표기

```
SELECT employee_id, first_name, last_name, salary
FROM employees
WHERE last_name='King';
```

- 문자 데이터는 대소문자를 구분

## ❖ 날짜 데이터 조회

- 2003년 1월 1일 이전에 입사한 사원 조회

```
SELECT *
FROM employees
WHERE hire_date <= '2003/01/01';
```

## 2. 논리 연산자

---

### ❖ 논리 연산자

- AND
- OR
- NOT

## 2. 논리 연산자

---

### ❖ AND 연산자

- 두 가지 조건을 모두 만족할 경우에만 검색
- 예
  - 110번 부서 소속 사원
  - 직급 ID가 AC\_MGR인 사원

```
SELECT *  
FROM employees  
WHERE department_id = 110 AND job_id = 'AC_MGR';
```

## 2. 논리 연산자

---

### ❖ OR 연산자

- 두가지 조건 중에서 한 가지만 만족하더라도 출력
- 예
  - 110번 부서 소속 사원
  - 직급 ID가 AC\_MGR인 사원

```
SELECT *  
FROM employees  
WHERE department_id = 110 OR job_id = 'AC_MGR';
```



## 2. 논리 연산자

---

### ❖ NOT 연산자

- 참을 거짓으로, 거짓을 참으로 구하는 연산자

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

```
SELECT *  
FROM employees  
WHERE NOT department_id = 100;
```

- NOT 대신에 <>, != 사용 가능

```
SELECT *  
FROM employees  
WHERE department_id <> 100;
```

## 2. 논리 연산자

---

### ❖ NOT 연산자

- 5000에서 7000 사이의 급여를 받는 사원 조회

```
SELECT *  
FROM employees  
WHERE salary >= 5000 AND salary <= 7000;
```

- 급여가 5000이하거나 7000 이상인 사원 조회

```
SELECT *  
FROM employees  
WHERE salary <= 5000 OR salary >= 7000;
```

- 커미션 비율이 0.1이거나 0.2이거나 0.3인 사원 조회

```
SELECT *  
FROM employees  
WHERE commission_pct=0.1 OR commission_pct=0.2 OR commission_pct=0.3;
```

## 2. 논리 연산자

---

### ❖ 실습

- 사원 번호가 116이거나 186이거나 204인 사원의 급여를 검색하는 쿼리문을 작성하세요.

# BETWEEN AND 연산자

---

## ❖ 특정 범위 내에 속하는지 검사

### ○ 형식

컬럼 이름 BETWEEN A AND B

```
SELECT *  
FROM employees  
WHERE salary BETWEEN 2000 AND 3000;
```

```
SELECT *  
FROM employees  
WHERE salary NOT BETWEEN 2000 AND 3000;
```

```
SELECT *  
FROM employees  
WHERE hire_date BETWEEN '2000/01/01' AND '2003/01/01';
```

# IN 연산자

---

## ❖ 특정 컬럼의 값이 여러 개의 값 중에서 하나인지 검사

```
SELECT *  
FROM employees  
WHERE commission_pct=0.1  
       OR commission_pct=0.2  
       OR commission_pct=0.3;
```

### ○ IN 연산자 형식

컬럼이름 IN(A, B, C)

```
SELECT *  
FROM employees  
WHERE commission_pct IN (0.1, 0.2, 0.3);
```

```
SELECT *  
FROM employees  
WHERE commission_pct NOT IN (0.1, 0.2, 0.3);
```

# IN 연산자

---

## ❖ 실습

- 사원 번호가 116이거나 186이거나 204인 사원의 이름과 급여를 검색하는 쿼리문을 IN 연산자를 사용하여 작성하세요.
- 사원 번호가 116도 아니고 186도 아니고 204도 아닌 사원의 이름과 급여를 검색하는 쿼리문을 작성하세요.
  - 비교 연산자와 논리 연산자 AND를 사용하여 작성하기
  - NOT IN 연산자를 사용하여 작성하기

## LIKE 연산자와 와일드 카드

---

### ❖ 검색하려는 값을 정확히 모를 경우 와일드카드와 함께 검색

- 형식

- 컬럼이름 LIKE 패턴

- 와일드 카드

- %: 문자가 없거나, 하나 이상의 문자에 어떤 값이 와도 상관없다
    - \_ : 하나의 문자에 어떤 값이 와도 상관없다.

# LIKE 연산자와 와일드 카드

---

## ❖ 와일드 카드(%)

- last\_name이 'F'로 시작하는 사원 조회

```
SELECT *  
FROM employees  
WHERE last_name LIKE 'F%';
```

- last\_name에 'k'를 포함하는 사원 조회

```
SELECT *  
FROM employees  
WHERE last_name LIKE '%k%';
```

- last\_name이 'n'으로 끝나는 사원 조회

```
SELECT *  
FROM employees  
WHERE last_name LIKE '%n';
```



# LIKE 연산자와 와일드 카드

---

## ❖ 와일드카드(\_) 사용하기

- 지정한 위치에 어떠한 한문자가 와도 상관이 없다

```
SELECT *  
FROM employees  
WHERE last_name LIKE '_a%';
```

- 세번째 글자가 'a'인 사원 조회

```
SELECT *  
FROM employees  
WHERE last_name LIKE '__a%';
```

## LIKE 연산자와 와일드 카드

---

### ❖ 실습

- 사원들 중에서 `first_name`이 J로 시작하는 사람만을 찾는 쿼리문을 작성하세요.
- 와일드 카드를 사용하여 사원 중에서 이름이 K로 시작하는 사원의 사원번호와 이름을 출력하세요
- 와일드 카드를 사용하여 이름 중에 e를 포함하는 사원의 사원 번호와 이름을 출력하세요.
- 와일드 카드를 사용하여 이름 중에 e로 끝나는 사원의 사원 번호와 이름을 출력하세요.

# LIKE 연산자와 와일드 카드

---

## ❖ ESCAPE 옵션

- 와일드카드 문자를 와일드 카드가 아닌 문자로 인식하도록 한다

```
SELECT *  
FROM employees  
WHERE last_name LIKE '%\%%' ESCAPE '\';
```

## ❖ NOT LIKE 연산자

- last\_name 에 'k'가 포함되지 않은 사원 조회

```
SELECT *  
FROM employees  
WHERE last_name NOT LIKE '%k%';
```

# NULL을 위한 연산자

---

## ❖ NULL 조회

- 커미션 비율이 지정되지 않은 사원 조회

```
SELECT *  
FROM employees  
WHERE commission_pct IS NULL;
```

- commission\_pct =NULL로 조회하면 에러는 나지 않으나 조회결과가 없음

- 커미션 비율이 지정된 사원 조회

```
SELECT *  
FROM employees  
WHERE commission_pct IS NOT NULL;
```

# NULL을 위한 연산자

---

## ❖ 실습

- 상관이 없는 사원(CEO)를 검색하기 위한 SQL문을 작성해 보세요.

# 정렬을 위한 ORDER BY 절

---

## ❖ SELECT 결과의 정렬

- 형식

```
SELECT * [컬럼1, 컬럼2, ...]  
FROM 테이블명  
WHERE 조건절  
ORDER BY 컬럼이름 [ASC|DESC]
```

## 정렬을 위한 ORDER BY 절

---

### ❖ 오름차순 정렬을 위한 ASC

- 급여 컬럼을 기준으로 오름차순 정렬
  - ORDER BY 디폴트는 오름차순(ASC 생략 가능)

```
SELECT *  
FROM employees  
ORDER BY salary ASC;
```

### ❖ 내림차순 정렬을 위한 DESC

- 급여 컬럼을 기준으로 내림차순 정렬

```
SELECT *  
FROM employees  
ORDER BY salary DESC;
```

## 정렬을 위한 ORDER BY 절

---

### ❖ 사원의 first\_name으로 오름차순 정렬

```
SELECT *  
FROM employees  
ORDER BY last_name;
```

### ❖ 가장 최근에 입사한 사람부터 출력

```
SELECT *  
FROM employees  
ORDER BY hire_date desc;
```

### ❖ 급여를 내림차순으로 정렬하고, 급여가 같은 경우 first\_name의 철자가 빠른사람부터 출력

```
SELECT *  
FROM employees  
ORDER BY salary DESC, first_name ASC;
```



## 정렬을 위한 ORDER BY 절

---

### ❖ 실습

- Employees 테이블의 자료에서 입사일을 오름차순으로 정렬하여 출력하되 사원 번호, 사원 이름, 입사일 컬럼을 출력하는 쿼리를 출력하세요.
- Employees 테이블의 자료를 사원번호를 기준으로 내림 차순으로 정렬하여 사원 번호와 사원명 컬럼을 표시하세요.
- 부서번호가 빠른 사원부터 출력하되 같은 부서내의 사원을 출력할 경우 최근에 입사한 사원부터 출력하되 사원번호, 입사일, 사원명, 급여순으로 출력하세요.