

## 학습 내용

- ❖ 특정 조건의 데이터만 조회하는 WHERE 조건
- ❖ 정렬을 위한 ORDER BY 절

## 학습목표

- ❖ WHERE 절로 원하는 조건에 맞는 데이터만 검색할 수 있습니다.
- ❖ ORDER BY 절로 테이블을 일정한 순서로 나열할 수 있습니다.

## WHERE 절을 이용한 조건 검색

- ❖ WHERE 절은 테이블에 저장된 데이터 중에서 원하는 데이터만 선택적으로 추출하기 위해서 사용됩니다.

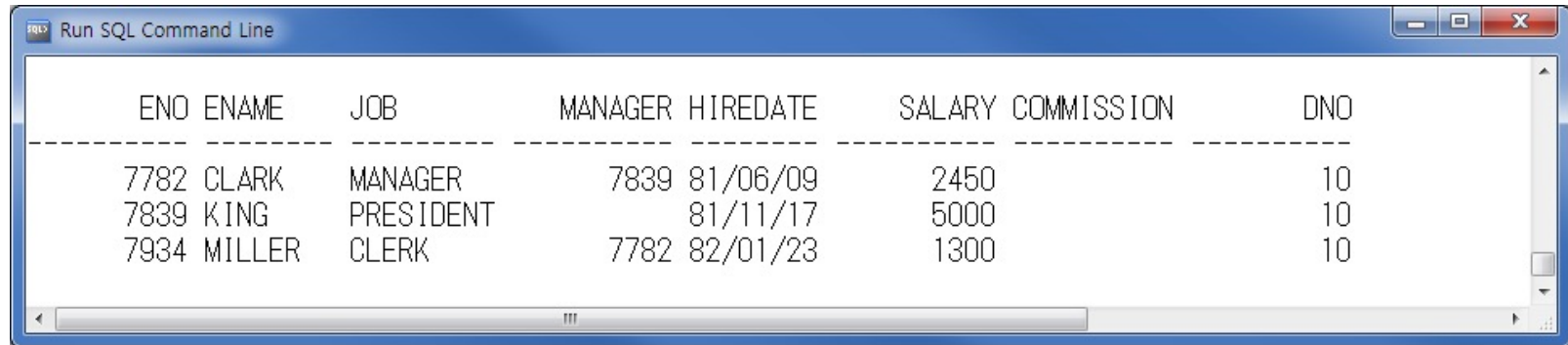
```
SELECT * [column1, column2, .. ,columnn]  
FROM table_name  
WHERE condition;
```

## 비교 연산자

연산자	의 미	예 제
=	같다.	select eno, ename, salary from employee where salary=1500;
>	보다 크다.	select eno, ename, salary from employee where salary>1500;
<	보다 작다.	select eno, ename, salary from employee where salary<1500;
>=	보다 크거나 같다.	select eno, ename, salary from employee where salary>=1500;
<=	보다 작거나 같다.	select eno, ename, salary from employee where salary<=1500;
<>, !=, ^=	다르다.	select eno, ename, salary from employee where salary<>1500;

## 비교 연산자

```
select * from employee  
where dno=10;
```



The screenshot shows a window titled "Run SQL Command Line" with a table of employee data. The table has columns: ENO, ENAME, JOB, MANAGER, HIREDATE, SALARY, COMMISSION, and DNO. The data is as follows:

ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7839	KING	PRESIDENT		81/11/17	5000		10
7934	MILLER	CLERK	7782	82/01/23	1300		10

## 문자 데이터 조회

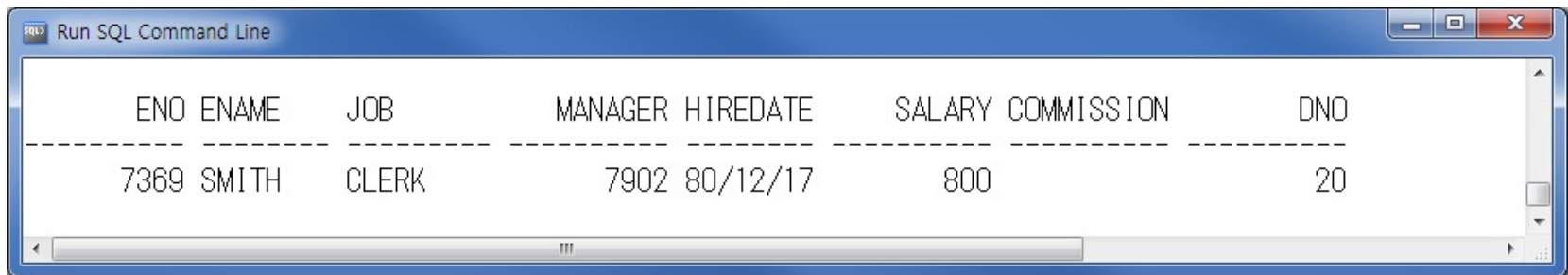
- ❖ 문자 상수임을 알리기 위해서 'SCOTT'를 작은따옴표로 묶어주면 오류가 발생하지 않고 사원 정보를 검색하는 것을 확인할 수 있습니다.

```
select * from employee  
where ename='SCOTT';
```

## 날짜 데이터 조회

- ❖ 날짜 상수 역시 문자 상수와 마찬가지로 작은따옴표로 묶어 주어야 합니다.

```
select * from employee  
where hiredate<='1981/01/01';
```



ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
7369	SMITH	CLERK	7902	80/12/17	800		20

## 논리 연산자

❖ 오라클에서 사용가능한 논리 연산자 AND나 OR나 NOT가 있습니다.

연산자	의미
AND	두 가지 조건을 모두 만족해야만 검색할 수 있습니다. <code>select * from employee where dno=10 and job='MANAGER';</code>
OR	두 가지 조건 중에서 한 가지만 만족하더라도 검색할 수 있습니다. <code>select * from employee where dno=10 or job='MANAGER';</code>
NOT	조건에 만족하지 못하는 것만 검색합니다. <code>select * from employee where not dno=10;</code>



## AND 연산자


- ❖ AND 연산자는 두 가지 조건을 모두 만족해야 할 경우에 사용됩니다. 즉, AND 연산자는 두 조건이 모두 만족할 경우에만 결과가 참이고, 조건 중 하나라도 만족하지 않으면 결과가 거짓입니다.

조건 1	조건2	AND
참	참	참
참	거짓	거짓
거짓	참	거짓
거짓	거짓	거짓

## AND 연산자

- ❖ 다음은 10번 부서 소속인 사원(dno=10)들 중에서 직급이 MANAGER인 사원(job='MANAGER')을 검색하기 위해서 AND 연산자를 사용한 예입니다.

```
select * from employee  
where dno=10 and job='MANAGER' ;
```



ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
7782	CLARK	MANAGER	7839	81/06/09	2450		10

## OR 연산자

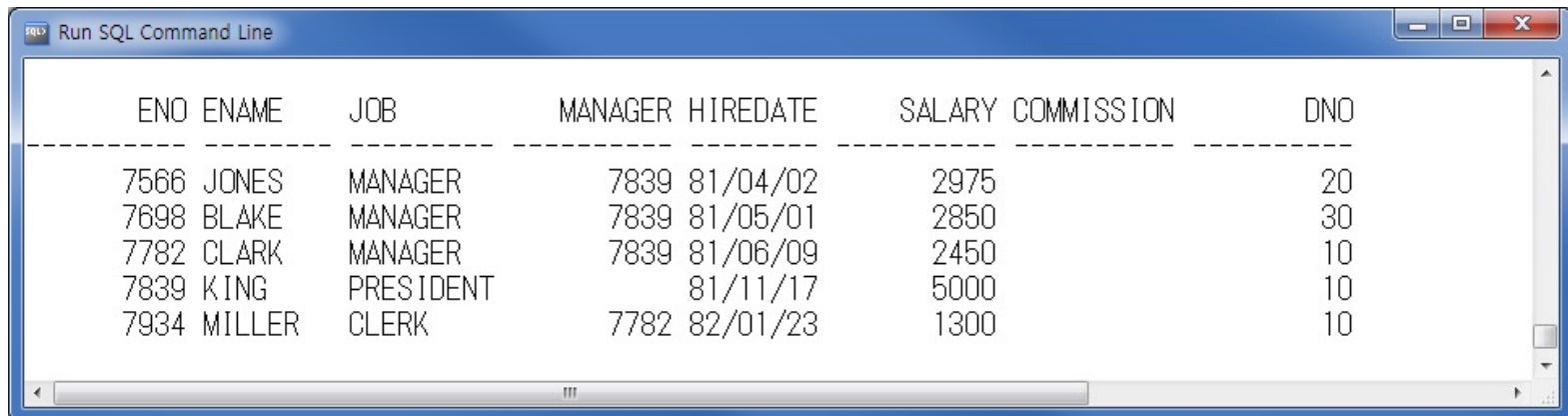
- ❖ OR 연산자는 두 조건에 모두 만족하지 않을 경우는 결과가 거짓이고 제시한 조건에 한 가지라도 만족하면 결과가 참입니다.

조건 1	조건2	OR
참	참	참
참	거짓	참
거짓	참	참
거짓	거짓	거짓

## OR 연산자

- ❖ 다음은 10번 부서에 소속된 사원(dno=10)이거나 직급이 MANAGER인 사원(job='MANAGER')을 검색하기 위해서 OR 연산자를 사용한 예입니다.

```
select * from employee  
where dno=10 or job='MANAGER';
```



ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
7566	JONES	MANAGER	7839	81/04/02	2975		20
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7839	KING	PRESIDENT		81/11/17	5000		10
7934	MILLER	CLERK	7782	82/01/23	1300		10

# NOT 연산자

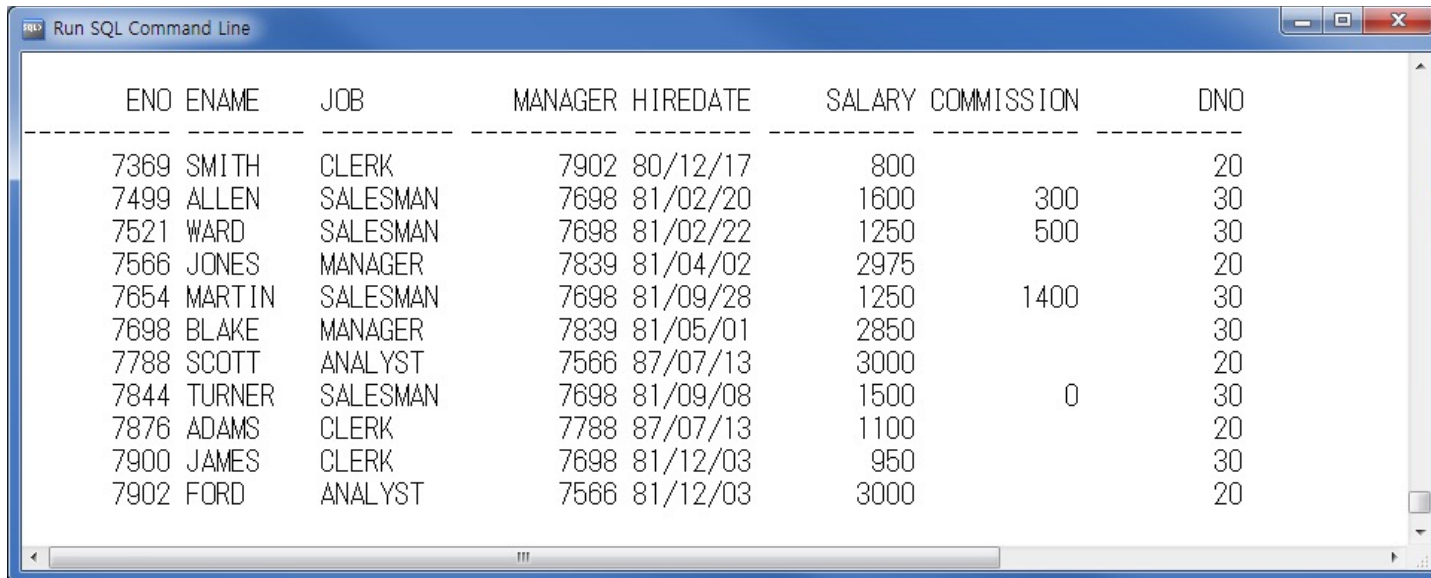
- ❖ 다음은 NOT 연산자가 논리값에 의해서 어떤 결과가 출력되는지를 나타내는 표입니다.

조건	NOT
참	거짓
거짓	참

# NOT 연산자

- ❖ 10번 부서에 소속된 직원만 제외하고 나머지 직원의 정보를 출력하려면 부서번호가 10번인 직원을 조회하기 위한 조건 앞에 NOT을 붙이면 됩니다.

```
select * from employee  
where not dno=10;
```



ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7566	JONES	MANAGER	7839	81/04/02	2975		20
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7788	SCOTT	ANALYST	7566	87/07/13	3000		20
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7876	ADAMS	CLERK	7788	87/07/13	1100		20
7900	JAMES	CLERK	7698	81/12/03	950		30
7902	FORD	ANALYST	7566	81/12/03	3000		20

## BETWEEN AND 연산자

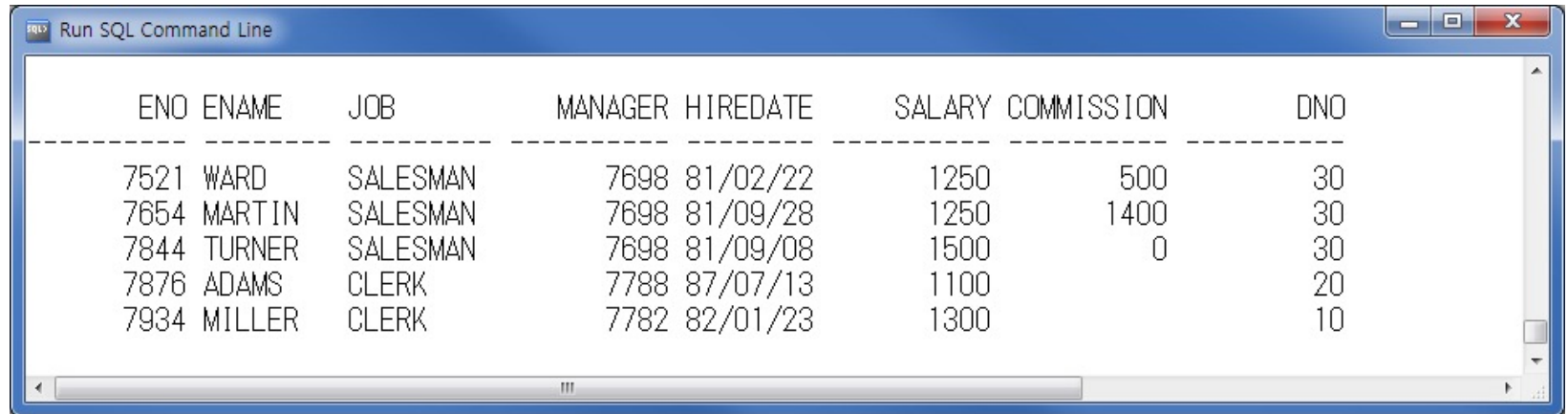
- ❖ BETWEEN AND 연산자는 특정 칼럼의 데이터 값이 하한값(A)와 상한값(B) 사이에 포함되는 로우를 검색하기 위한 연산자입니다.

형식	<i>column_name</i> BETWEEN A AND B
----	------------------------------------

## BETWEEN AND 연산자

- ❖ 다음은 BETWEEN AND 연산자를 사용하여 급여가 1000에서 1500 사이인 사원을 출력한 예입니다.

```
select * from employee  
where salary between 1000 and 1500;
```



ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7876	ADAMS	CLERK	7788	87/07/13	1100		20
7934	MILLER	CLERK	7782	82/01/23	1300		10



## IN 연산자

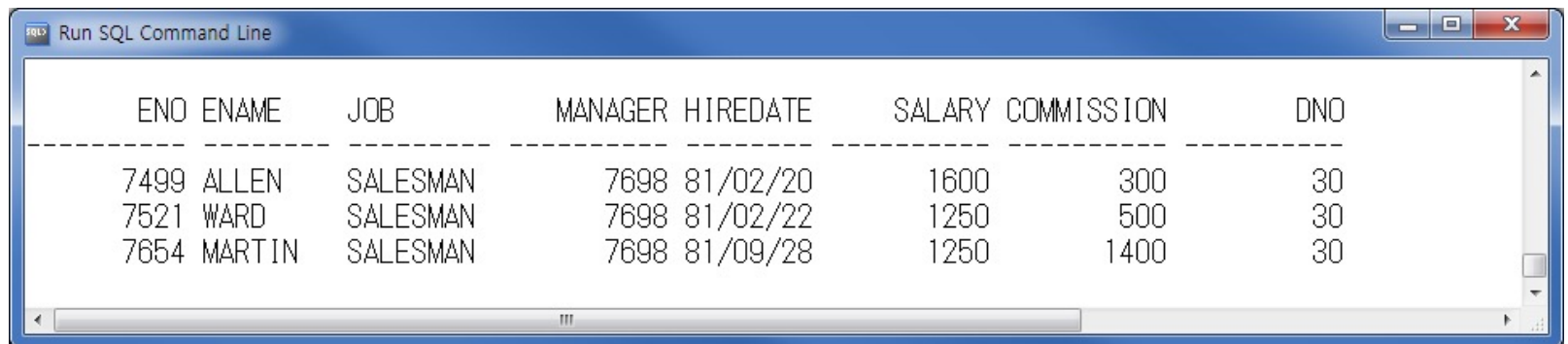
- ❖ IN 연산자는 특정 칼럼의 값이 A, B, C 중에 하나라도 일치하면 참이 되는 연산자입니다.

형식	<i>column_name</i> IN(A, <b>B</b> , C)
----	--

# IN 연산자

- ❖ 다음은 커미션이 300이거나 500이거나 1400인 사원을 검색하기 위해서 IN 연산자를 사용한 예입니다.

```
select * from employee  
where commission in(300, 500, 1400);
```



ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30

## LIKE 연산자와 와일드카드

- ❖ LIKE 연산자는 칼럼에 저장된 문자상수 중에서 LIKE 연산자에서 지정한 문자패턴과 부분적으로 일치하면 참이 되는 연산자로 이 씨성을 갖는 사람을 찾거나 거주지가 서울인 사람을 찾는 것과 같이 칼럼에 저장된 데이터의 일부만 일치하더라도 조회가 가능하도록 하기 위해서 사용합니다.

형식

*column\_name* LIKE *pattern*

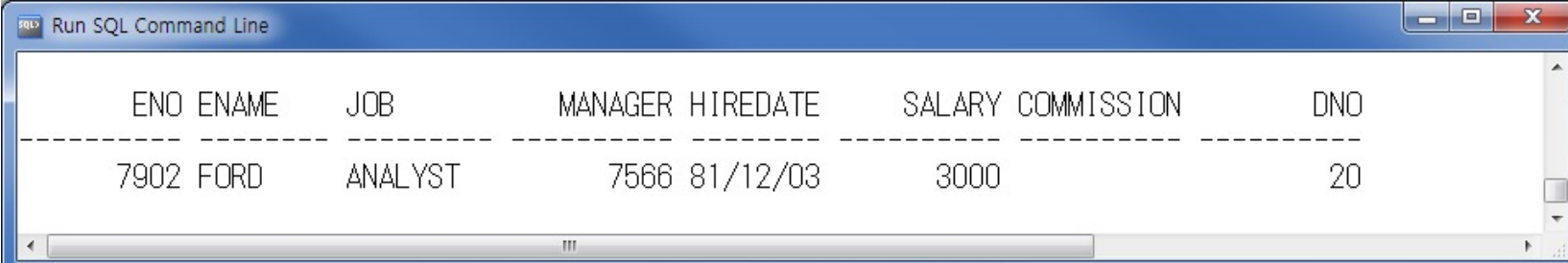
*pattern*에는 두 가지 와일드카드가 사용되며 이 와일드카드 덕분에 임의의 문자 또는 문자열을 포함한 값을 찾을 수 있습니다.

와일드카드	의미
%	문자가 없거나, 하나 이상의 문자가 어떤 값이 와도 상관 없다.
_	하나의 문자가 어떤 값이 와도 상관 없다.

## 와일드카드(%) 사용하기

- ❖ 사원 테이블 중 F로 시작 하는 이름을 갖는 사원을 검색하기 해서는 검색하고자 하는 값을 정확히 모를 경우 즉, 특정 문자 포함되기만 하고 그 이전이나 이후에 어떤 문자가 몇 개가 오든지 상관없다는 의미를 표현하기 위해서는 LIKE 연산자와 함께 '%'를 사용해야 합니다.

```
select * from employee  
where ename like 'F%';
```



The screenshot shows a window titled "Run SQL Command Line" with a table of employee data. The table has columns: ENO, ENAME, JOB, MANAGER, HIREDATE, SALARY, COMMISSION, and DNO. The data row shows ENO 7902, ENAME FORD, JOB ANALYST, MANAGER 7566, HIREDATE 81/12/03, SALARY 3000, and DNO 20.

ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
7902	FORD	ANALYST	7566	81/12/03	3000		20

## 와일드카드(\_) 사용하기

- ❖ '\_' 기호는 단 한 문자에 대해서만 와일드카드 역할을 하기 때문에 '\_'를 기술한 위치에 어떤 문자가 오든 상관없다는 의미로 사용됩니다. 그렇기 때문에 '\_' 기호는 사용되는 위치와 순서에 중요합니다.

```
select * from employee  
where ename like '_A%';
```

ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7900	JAMES	CLERK	7698	81/12/03	950		30

## IS NULL과 IS NOT NULL

- ❖ 어떤 컬럼을 NULL 즉, 모르는 값과 같다(=)라는 것은 의미상으로 말이 되지 않기 때문에 = 대신 IS NULL 연산자를 사용해야 합니다.
- ❖ IS NULL 연산자 역시 조건절에 사용되면 대상 컬럼과 연산자, 비교할 값 세부분으로 구성되어야 합니다.

형식 대상컬럼 IS (연산자) NULL(비교값)

- ❖ 컬럼 값이 NULL인지 조회합니다.

```
select * from employee  
where commission is null;
```

## IS NULL과 IS NOT NULL

- ❖ 칼럼이 NULL 아닌 자료만 추출하면 되므로 IS NOT NULL 연산자를 사용하면 됩니다.

```
select * from employee  
where commission is not null;
```

## 정렬을 위한 ORDER BY 절

- ❖ 정렬이란 크기 순서대로 나열하는 것을 의미합니다.
- ❖ 오름차순(ascending) 정렬 방식
  - 작은 것이 위에 출력되고 아래로 갈수록 큰 값이 출력
- ❖ 내림차순(descending) 정렬 방식이
  - 큰 값이 위에 출력되고 아래로 갈수록 작은 값이 출력

❖	ASC(오름차순)	DESC(내림차순)	하고 니다.
숫자	작은 값부터 정렬	큰 값부터 정렬	
문자	사전 순서로 정렬	사전 반대 순서로 정렬	
날짜	빠른 날짜 순서로 정렬	늦은 날짜 순서로 정렬	
NULL	가장 마지막에 나온다.	가장 먼저 나온다.	



## 오름차순 정렬을 위한 ASC

- ❖ 오름차순 정렬은 작은 값부터 큰 값으로 정렬하는 것을 의미합니다.(예:1~9, 'A'~'Z') 이를 위해서는 ASC를 컬럼 다음에 기술해야 하는데 만일 생략하게 되면 디폴트로 ASC로 지정되어 있기 때문에 오름차순으로 출력됩니다.
- ❖ 다음은 급여 컬럼을 기준으로 오름차순으로 정렬한 예입니다.

```
select * from employee  
order by salary asc;
```

- ❖ 정렬방식을 지정하지 않은 경우에는 디폴트로 오름차순으로 정렬합니다.

```
select * from employee  
order by salary;
```

## 내림차순 정렬을 위한 DESC

- ❖ 내림차순 정렬은 큰 값부터 작은 값으로 정렬을 하는 것이다.(예:9~1, Z~A)
- ❖ 이번에는 급여를 많이 받는 사람부터 적게 받는 사람 순으로 순차적으로 출력해 봅시다.

```
select * from employee  
order by salary desc;
```

- ❖ 큰 값이 위에 출력되고 아래로 갈수록 작은 값이 출력되도록 하려면 내림차순(descending) 으로 정렬해야 하기 때문에 칼럼 다음에 DESC를 기술해야 합니다.

## 정렬 방식에 여러 가지 조건 제시

- ❖ 급여를 많이 받는 사람부터 적게 받는 사람 순으로 순차적으로 출력하는 결과 화면을 살펴보면 동일한 급여를 받는 사람이 존재합니다.
- ❖ 급여가 같은 사람이 존재할 경우 이름의 철자가 빠른 사람부터 출력되도록 하려면 정렬 방식을 여러 가지로 지정해야 합니다.

```
select * from employee  
order by salary desc, ename asc;
```