

ORACLE® 04. 데이터 추출하기

목차

1. WHERE
2. 비교 연산자
3. 논리 연산자
4. BETWEEN AND 연산자
5. IN 연산자
6. LIKE 연산자
7. 와일드 카드
8. NULL관련 연산자
9. ORDER BY

1. WHERE

- ❖ 원하는 행(Row)만 얻으려면 다음과 같이 행을 제한하는 조건을 SELECT 문에 WHERE 절을 추가형 제시해야 한다.

형식

```
SELECT * [column1, column2 ...]  
FROM 테이블명 WHERE 조건절;
```

- ❖ 조건절은 다음의 세 부분으로 구성된다.

구성

```
WHERE SAL >= 300;  
①컬럼 ②연산자 ③비교대상값
```

- ❖ 급여가 300 이하인 사원의 사원번호, 사원 이름, 급여를 출력하는 SQL문을 작성해보자.

형식

```
SELECT empno, ename, sal  
FROM emp WHERE sal <= 300;
```

	EMPNO	ENAME	SAL
1	1001	김사랑	300
2	1002	한예슬	250
3	1011	조향기	280
4	1012	강혜정	300
5	1014	조인성	250

2. 비교 연산자

연산자	의 미	예 제
=	같다.	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL=300;
>	보다 크다.	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL>300;
<	보다 작다.	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL<300;
>=	보다 크거나 같다.	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL>=300;
<=	보다 작거나 같다.	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL<=300;
<>, !=, ^=	다르다.	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL<>300;

2.1 문자 데이터 조회

- ❖ 이전 예제에서 비교 연산자를 하기 위해서 다루었던 컬럼들은 숫자 형태로 선언 되었다. 이번에는 급여가 아닌 사원 이름 같은 문자 데이터를 조회해 보자.
- ❖ 이름이 '김사랑'인 사원의 사원번호와 사원이름, 급여를 출력하는 예제이다.

형식

```
SELECT empno, ename, sal  
FROM emp WHERE ename = '김사랑';
```

- ❖ SQL에서 문자열이나 날짜는 반드시 단일 따옴표(') 안에 표시해야 한다.
- ❖ SQL문에 사용되는 키워드는 SELECT, FROM, WHERE 등은 대소문자를 구별하지 않지만 테이블 내에 저장된 데이터 값은 대소문자를 구분하기 때문에 대소문자를 정확히 입력하여 비교하여야 한다.

2.2 날짜 데이터 조회

- ❖ 날짜는 문자열과 마찬가지로 작은따옴표 안에 기술해야한다.
- ❖ 2003년 9월 1일 이후에 입사한 사원을 출력하는 예제이다.

형식

SELECT * FROM emp WHERE hiredate >= '03/09/01';

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	1004	이병헌	부장	1008	03/09/02	600	(null)	20
2	1006	장농건	부장	1008	03/10/09	480	(null)	30
3	1007	이문세	부장	1008	04/01/08	520	(null)	10
4	1008	감우성	차장	1003	04/03/08	500	0	30
5	1003	오지호	과장	1005	05/02/10	500	100	30
6	1005	신농협	과장	1005	05/04/07	450	200	30
7	1010	이병헌	과장	1003	05/04/07	500	(null)	10
8	1011	조향기	사원	1007	07/03/01	280	(null)	30
9	1001	김사랑	사원	1013	07/03/01	300	(null)	20
10	1002	한예슬	대리	1005	07/04/02	250	80	30
11	1012	강혜정	사원	1006	07/08/09	300	(null)	20
12	1014	조인성	사원	1006	07/11/09	250	(null)	10

3. 논리 연산자

연산자	의 미
AND	두 가지 조건을 모두 만족해야만 검색할 수 있다. SELECT * FROM emp WHERE deptno=10 AND job='사원';
OR	두 가지 조건 중에서 한 가지만 만족하더라도 검색할 수 있다. SELECT * FROM emp WHERE deptno=10 OR job= ' 사원';
NOT	조건에 만족하지 못하는 것만 검색한다. SELECT * FROM emp WHERE NOT deptno=10;

3.1 AND 연산자(1)

- ❖ 두 가지 조건을 모두 만족할 경우에만 검색할 수 있도록 하기 위해서는 AND 연산자를 사용해야 한다.
- ❖ 두 조건이 모두 만족할 경우에만 결과가 참이고, 조건 중 하나라도 만족하지 않으면 결과가 거짓이다.

조건1	조건2	&&
참	참	참
참	거짓	거짓
거짓	참	거짓
거짓	거짓	거짓

3.1 AND 연산자(2)

- ❖ 10번 부서 소속인 사원들 중에서 직급이 사원인 사람을 검색하여 사원명, 부서번호, 직급을 출력하는 예제이다.
- ❖ 조건1 : 10번 부서 소속 직원 (deptno = 10)
조건2 : 직급이 사원인 사원 (job = '사원')

예

```
SELECT ename, deptno, job  
FROM emp WHERE deptno=10 and job='사원';
```

	ENAME	DEPTNO	JOB
1	조인성	10	사원

3.2 OR 연산자(1)

- ❖ 두 가지 조건 중에서 한 가지만 만족하더라도 검색할 수 있도록 하기 위해서는 OR 연산자를 사용해야 한다.
- ❖ 두 조건에 모두 만족하지 않을 경우는 결과가 거짓이고, 제시한 조건에 한 가지라도 만족하면 결과가 참이다.

조건1	조건2	
참	참	참
참	거짓	참
거짓	참	참
거짓	거짓	거짓

3.2 OR 연산자[2]

- ❖ 10번 부서 소속인 사원이거나 직급이 사원인 사람을 검색하여 사원명, 부서번호, 직급을 출력하는 예제이다.
- ❖ 조건1 : 10번 부서 소속 직원 (deptno = 10)
조건2 : 직급이 사원인 사원 (job = '사원')

예 **SELECT** ename, deptno, job
FROM emp **WHERE** deptno=10 or job='사원';

	ENAME	DEPTNO	JOB
1	김사량	20	사원
2	이문세	10	부장
3	이병헌	10	과장
4	조향기	30	사원
5	강혜정	20	사원
6	조인성	10	사원

3.3 NOT 연산자(1)

- ❖ 10번 부서에 소속된 직원만 제외하고 나머지 직원의 정보를 출력하려고 할 때 NOT연산자를 사용할 수 있다.
- ❖ 참은 거짓으로 거짓은 참으로 반대되는 논리값을 구하는 연산자이다.

조건	NOT
참	거짓
거짓	참

3.3 NOT 연산자[2]

- ❖ 조건 앞에 NOT 을 붙이면 부서번호가 10번이 아닌 사원들에 대해서만 검색하게 된다.

예

```
SELECT *  
FROM emp  
WHERE NOT deptno = 10;
```

예

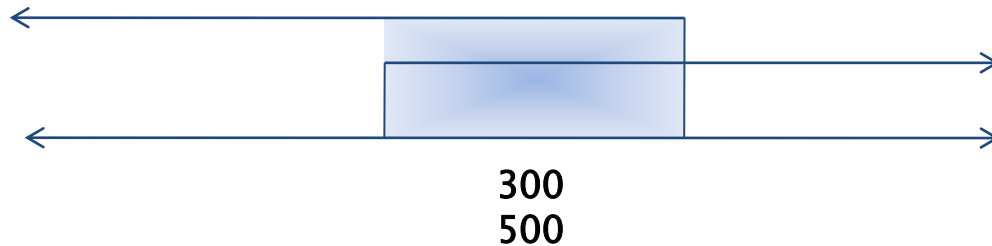
```
SELECT *  
FROM emp  
WHERE deptno <> 10;
```

예

```
SELECT *  
FROM emp  
WHERE deptno != 10;
```

3.4 논리 연산자의 다양한 활용

- ❖ 300에서 500 사이의 급여를 받는 사원을 조회해보자.
- ❖ 조건1 : 급여가 300 이상 ($sal \geq 300$)
조건2 : 급여가 500 이하 ($sal \leq 500$)



예

```
SELECT *  
FROM emp  
WHERE sal >= 300 and sal <= 500;
```

4. BETWEEN AND 연산자[1]

- ❖ 특정 범위의 값을 조회할 때 사용하는 연산자.
- ❖ ex) 월급이 200에서 300사이인 데이터 출력

형식 **SELECT * FROM emp WHERE sal >=200 and sal <=300;**

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	1001	김사랑	사원	1013	07/03/01	300	(null)	20
2	1002	한예슬	대리	1005	07/04/02	250	80	30
3	1011	조향기	사원	1007	07/03/01	280	(null)	30
4	1012	강혜정	사원	1006	07/08/09	300	(null)	20
5	1014	조인성	사원	1006	07/11/09	250	(null)	10

형식 **SELECT * FROM emp WHERE sal BETWEEN 200 and 300;**

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	1001	김사랑	사원	1013	07/03/01	300	(null)	20
2	1002	한예슬	대리	1005	07/04/02	250	80	30
3	1011	조향기	사원	1007	07/03/01	280	(null)	30
4	1012	강혜정	사원	1006	07/08/09	300	(null)	20
5	1014	조인성	사원	1006	07/11/09	250	(null)	10

4. BETWEEN AND 연산자[2]

❖ ex) 월급이 200미만이거나 300초과인 데이터 출력

형식

SELECT * FROM emp WHERE sal < 200 or sal > 300;

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	1003	오지호	과장	1005	05/02/10	500	100	30
2	1004	이병헌	부장	1008	03/09/02	600	(null)	20
3	1005	신농협	과장	1005	05/04/07	450	200	30
4	1006	장농건	부장	1008	03/10/09	480	(null)	30
5	1007	이문세	부장	1008	04/01/08	520	(null)	10
6	1008	감우성	차장	1003	04/03/08	500	0	30
7	1009	안성기	사장	(null)	96/10/04	1000	(null)	20
8	1010	이병헌	과장	1003	05/04/07	500	(null)	10
9	1013	박송훈	부장	1003	02/10/09	560	(null)	20

형식

SELECT * FROM emp WHERE sal NOT BETWEEN 200 and 300;

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	1003	오지호	과장	1005	05/02/10	500	100	30
2	1004	이병헌	부장	1008	03/09/02	600	(null)	20
3	1005	신농협	과장	1005	05/04/07	450	200	30
4	1006	장농건	부장	1008	03/10/09	480	(null)	30
5	1007	이문세	부장	1008	04/01/08	520	(null)	10
6	1008	감우성	차장	1003	04/03/08	500	0	30
7	1009	안성기	사장	(null)	96/10/04	1000	(null)	20
8	1010	이병헌	과장	1003	05/04/07	500	(null)	10
9	1013	박송훈	부장	1003	02/10/09	560	(null)	20

5. IN 연산자(1)

- ❖ 비교 연산자인 OR 연산자를 사용하기 편하게 만들어 놓은 연산자이다.
- ❖ 동일한 필드가 여러 개의 값 중 하나를 가지는지 확인 가능.
- ❖ ex)급여가 300이거나 500이거나 1000인 데이터를 출력

형식

SELECT * FROM emp WHERE sal=300 or sal=500 or sal=1000



형식

컬럼명 IN(A, B, C)



형식

SELECT * FROM emp WHERE sal IN(300, 500, 1000);

5. IN 연산자[2]

❖ ex)급여가 300, 500, 1000이 모두 아닌 데이터 출력

형식

```
SELECT * FROM emp WHERE sal!=300 and sal!=500 and  
sal!=1000
```



형식

```
SELECT * FROM emp WHERE sal not in(300, 500, 1000);
```

6. LIKE 연산자

- ❖ 검색하고자 하는 값을 정확히 알지 못하는 경우에도 검색이 가능하도록 해주는 연산자이다.
- ❖ 와일드 카드와 함께 사용한다.

형식

컬럼명 LIKE pattern(와일드카드)

- ❖ LIKE 다음에는 pattern을 기술해야 하는데, pattern에는 다음과 같은 두가지의 와일드 카드가 사용된다.

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

7. 와일드 카드(1)

❖ % 와일드 카드

- 해당 위치에 어떤 문자가, 몇 개가 올지 모르는 경우 사용
- 검색하고자 하는 값을 정확히 모르는 경우에 어떤 문자가 몇 개가 오든지 상관 없다는 뜻

❖ ex) 찾으려는 사원의 성이 '김' 씨 인건 알지만 이름이 무엇인지 모르는 경우에 사용할 수 있다.

형식

SELECT * FROM emp WHERE ename LIKE '김%';

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	1001	김사량	사원	1013	07/03/01	300	(null)	20

7. 와일드 카드[2]

- ❖ '%' 을 문자 앞뒤에 기술하면 문자 앞뒤에 어떤 문자가 오든지 상관없이 검색할 수 있다.
- ❖ ex) 이름 중 '기'을 포함하고 있는 사원 테이블의 데이터를 출력

형식

SELECT * FROM emp WHERE ename LIKE '%기%';

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	1009	안성기	사장	(null)	96/10/04	1000	(null)	20
2	1011	조향기	사원	1007	07/03/01	280	(null)	30

7. 와일드 카드(3)

❖ '_' 와일드 카드

- '%'와 마찬가지로 어떤 문자라도 검색 가능
- '%'은 문자의 개수에 제한이 없지만, '_'은 한문자에 대해서만 와일드 카드 역할

❖ ex) 이름이 3글자이고 그 중 두번째 이름에 '동'인 사원의 정보를 출력

형식 **SELECT * FROM emp WHERE ename LIKE '_동_';**

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	1005	신동협	과장	1005	05/04/07	450	200	30
2	1006	장동건	부장	1008	03/10/09	480	(null)	30

❖ 앞, 뒤에 오는 문자의 개수를 확인을 한 상태로 작업을 진행해야 한다.

8. NULL관련 연산자(1)

- ❖ NULL을 가진 데이터를 검색하기 위한 기능.
- ❖ '=' 연산자로 comm 컬럼의 값이 NULL인지 확인

형식 **SELECT * FROM emp WHERE comm=NULL**

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO

- ❖ NULL을 가진 데이터와 비교 연산을 하게 되면 위 화면처럼 결과를 찾을 수 없다. NULL은 = 연산자로 검색이 불가능하다.

8. NULL관련 연산자(2)

❖ IS NULL

- NULL인 데이터를 검색할 때 사용하는 연산자
- 일반적인 비교 연산자인(=)와 같은 기능이지만 NULL은 (=) 연산자 사용 불가

형식 **컬럼명 IS NULL**

❖ ex) 커미션(comm)이 NULL인 직원들의 데이터를 출력

형식 **SELECT * FROM emp WHERE comm IS NULL;**

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	1001	김사량	사원	1013	07/03/01	300	(null)	20
2	1004	이병헌	부장	1008	03/09/02	600	(null)	20
3	1006	장동건	부장	1008	03/10/09	480	(null)	30
4	1007	이문세	부장	1008	04/01/08	520	(null)	10
5	1009	안성기	사장	(null)	96/10/04	1000	(null)	20
6	1010	이병헌	과장	1003	05/04/07	500	(null)	10
7	1011	조항기	사원	1007	07/03/01	280	(null)	30
8	1012	강혜정	사원	1006	07/08/09	300	(null)	20
9	1013	박송훈	부장	1003	02/10/09	560	(null)	20
10	1014	조인성	사원	1006	07/11/09	250	(null)	10

8. NULL관련 연산자(3)

❖ IS NOT NULL

- NULL이 아닌 데이터를 검색할 때 사용하는 연산자
- 일반적인 검색 연산자인(!=, <>)와 같은 기능이지만 NULL은 사용 불가

형식 컬럼명 IS NOT NULL

❖ ex) 커미션(comm)이 NULL이 아닌 직원들의 데이터를 출력

형식 **SELECT * FROM emp WHERE comm IS NOT NULL;**

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	1002	한예슬	대리	1005	07/04/02	250	80	30
2	1003	오지호	과장	1005	05/02/10	500	100	30
3	1005	신농협	과장	1005	05/04/07	450	200	30
4	1008	감우성	차장	1003	04/03/08	500	0	30

9. ORDER BY

- ❖ 순서에 맞게 정렬하는 조건을 말한다.
- ❖ 오름차순(Ascending)
 - 작은 값이 위에 출력되고 아래로 갈수록 큰 값이 출력
- ❖ 내림차순(Descending)
 - 큰 값이 위에 출력되고 아래로 갈수록 작은 값이 출력
- ❖ 정렬을 하기 위해서 SELECT문에 ORDER BY 절을 추가하고 어떤 컬럼을 기준으로 정렬할 것인지 결정해야 한다.

	ASC(오름차순)	DESC(내림차순)
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.

9.1 오름차순

❖ 오름차순

- 작은 값부터 큰 값으로 정렬하는 것을 의미(1~9, 'A'~'Z')
- 정렬 구문에 오름차순, 내림차순을 따로 설정하지 않으면 기본으로 오름차순 정렬 된다.

형식 **ORDER BY 컬럼명 ASC;**

❖ ex) 급여를 적게 받는 사원순으로 출력한 예제이다.

예제 **SELECT * FROM emp ORDER BY sal ASC;**

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	1002	한예슬	대리		1005 07/04/02	250	80	30
2	1014	조인성	사원		1006 07/11/09	250	(null)	10
3	1011	조향기	사원		1007 07/03/01	280	(null)	30
4	1001	김사랑	사원		1013 07/03/01	300	(null)	20
5	1012	강혜정	사원		1006 07/08/09	300	(null)	20
6	1005	신농협	과장		1005 05/04/07	450	200	30
7	1006	장농건	부장		1008 03/10/09	480	(null)	30
8	1010	이병현	과장		1003 05/04/07	500	(null)	10
9	1008	감우성	차장		1003 04/03/08	500	0	30
10	1003	오지호	과장		1005 05/02/10	500	100	30
11	1007	이문세	부장		1008 04/01/08	520	(null)	10
12	1013	박송훈	부장		1003 02/10/09	560	(null)	20
13	1004	이병현	부장		1008 03/09/02	600	(null)	20
14	1009	안성기	사장		(null) 96/10/04	1000	(null)	20

9.2 내림차순

❖ 내림차순

- 큰 값부터 작은 값으로 정렬하는 것을 의미
- 9~1, 'Z'~'A'

❖ ex) 급여를 많이 받는 직원순으로 출력한 예제이다.

형식

SELECT * FROM emp ORDER BY sal DESC;

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	1009	안성기	사장	(null)	96/10/04	1000	(null)	20
2	1004	이병헌	부장	1008	03/09/02	600	(null)	20
3	1013	박승훈	부장	1003	02/10/09	560	(null)	20
4	1007	이문세	부장	1008	04/01/08	520	(null)	10
5	1008	감우성	차장	1003	04/03/08	500	0	30
6	1003	오지호	과장	1005	05/02/10	500	100	30
7	1010	이병헌	과장	1003	05/04/07	500	(null)	10
8	1006	장농건	부장	1008	03/10/09	480	(null)	30
9	1005	신농협	과장	1005	05/04/07	450	200	30
10	1001	김사랑	직원	1013	07/03/01	300	(null)	20
11	1012	강혜정	직원	1006	07/08/09	300	(null)	20
12	1011	조향기	직원	1007	07/03/01	280	(null)	30
13	1014	조인성	직원	1006	07/11/09	250	(null)	10
14	1002	한예슬	대리	1005	07/04/02	250	80	30

9.3 문자 순으로 출력

- ❖ 크기에 대한 비교는 수치 데이터 뿐만 아니라 문자 데이터나 날짜 데이터에 대해서도 가능.
- ❖ 문자 데이터의 경우 아스키 코드 값으로 저장되므로 아스키 코드 값을 기준으로 정렬.
- ❖ 사전적 순서를 따름.
- ❖ 오름차순인 경우에는 A, B, ... Z 순으로 출력되고 내림차순인 경우에는 Z, Y, ... A 순으로 출력된다.
- ❖ ex) 사원의 이름을 철자 순 오름차순 정렬 한 예제.

형식

SELECT * FROM emp ORDER BY ename ASC;

9.4 날짜 순으로 출력

- ❖ 날짜의 경우에도 오름차순 혹은 내림차순으로 출력할 수 있다.
- ❖ 오름차순으로 지정하면 가장 오래된 과거의 시점이 가장 위에 출력되고 아래로 갈수록 최근 시점이 출력된다.
- ❖ 내림차순인 경우에는 최근 시점부터 출력된다.
- ❖ ex) 최근 입사한 사람부터 출력하는 예제.

형식

SELECT * FROM emp ORDER BY hiredate DESC;

9.5 다중 정렬

❖ 정렬에 여러가지 조건 대입

- 해당 컬럼 기준으로 정렬을 마치고 나서 추가적인 정렬이 필요할 때 사용

❖ ex) 급여를 많이 받는 사람을 기준으로 정렬한 뒤 혹시 급여가 같을 경우 이름의 철자가 빠른 사람부터 출력되도록 하는 예제

형식

SELECT * FROM emp ORDER BY sal DESC, ename ASC;

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	1009	안성기	사장	(null)	96/10/04	1000	(null)	20
2	1004	이병헌	부장	1008	03/09/02	600	(null)	20
3	1013	박승훈	부장	1003	02/10/09	560	(null)	20
4	1007	이눈세	부장	1008	04/01/08	520	(null)	10
5	1008	감우성	차장	1003	04/03/08	500	0	30
6	1003	오지호	과장	1005	05/02/10	500	100	30
7	1010	이병헌	과장	1003	05/04/07	500	(null)	10
8	1006	장동건	부장	1008	03/10/09	480	(null)	30
9	1005	신동엽	과장	1005	05/04/07	450	200	30
10	1012	강혜정	사원	1006	07/08/09	300	(null)	20
11	1001	김사랑	사원	1013	07/03/01	300	(null)	20
12	1011	조향기	사원	1007	07/03/01	280	(null)	30
13	1014	조인성	사원	1006	07/11/09	250	(null)	10
14	1002	한예슬	대리	1005	07/04/02	250	80	30

문제

1. 급여가 300 이하인 사원의 사원 번호, 사원명, 급여를 출력하시오.

EMPNO	ENAME	SAL
1001	김사랑	300
1002	한예슬	250
1011	조향기	280
1012	강혜정	300
1014	조인성	250

2. 이름이 "오지호"인 사원의 사원 번호, 사원명, 급여를 출력하시오.

EMPNO	ENAME	SAL
1003	오지호	500

문제

3. 급여가 250이거나, 300이거나, 500인 직원들의 직원 번호, 직원명, 급여를 출력하시오. (비교 연산자와 or 논리 연산자를 작성해보고, in 연산자로 한 번 더 작성하시오.)

EMPNO	ENAME	SAL
1001	김사랑	300
1002	한예슬	250
1003	오지호	500
1008	감우성	500
1010	이병헌	500
1012	강혜정	300
1014	조인성	250

문제

4. 급여가 250도, 300도, 500도 아닌 직원들을 검색하시오.

(비교 연산자와 and 논리 연산자를 사용하여 작성해보고, in 연산자로 한 번 더 작성하시오.)

EMPNO	ENAME	SAL
1004	이병헌	600
1005	신농협	450
1006	장농건	480
1007	이문세	520
1009	안성기	1000
1011	조향기	280
1013	박송훈	560

문제

5. 사원들 중에서 이름이 "김 " 으로 시작하는 사람과 이름 중에 "기 " 를 포함하는 사원의 사원번호와 사원명을 출력하시오. (**LIKE** 연산자와 와일드카드를 이용)

EMPNO	ENAME
1001	김사랑
1009	안성기
1011	조향기

6. 상급자가 없는 사원(사장)을 검색하기 위한 SQL문을 작성하시오. (**NULL** 이용)

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1009	안성기	사장	(null)	96/10/04	1000	(null)	20

문제

7. 사원 테이블의 자료를 입사일을 기준으로 정렬하여 최근에 입사한 직원을 가장 먼저 출력하는 쿼리문을 작성하시오(사원번호, 사원명, 직급, 입사일만 출력)

EMPNO	ENAME	JOB	TO_CHAR(HIREDATE, 'YYYY/MM/DD')
1014	조인성	사원	2007/11/09
1012	강혜정	사원	2007/08/09
1002	한예슬	대리	2007/04/02
1011	조향기	사원	2007/03/01
1001	김사랑	사원	2007/03/01
1010	이병헌	과장	2005/04/07
1005	신농협	과장	2005/04/07
1003	오지호	과장	2005/02/10
1008	감우성	차장	2004/03/08
1007	이문세	부장	2004/01/08
1006	장농건	부장	2003/10/09
1004	이병헌	부장	2003/09/02
1013	박송훈	부장	2002/10/09
1009	안성기	사장	1996/10/04

문제

8. 부서 번호가 빠른 사원부터 출력하되, 같은 부서 내의 사원을 출력할 경우
입사한지 가장 오래된 사원부터 출력되도록 하시오.

DEPTNO	EMPNO	ENAME	JOB	TO_CHAR(HIREDATE, 'YYYY/MM/DD')
10	1007	이문세	부장	2004/01/08
10	1010	이병헌	과장	2005/04/07
10	1014	조인성	사원	2007/11/09
20	1009	안성기	사장	1996/10/04
20	1013	박송훈	부장	2002/10/09
20	1004	이병헌	부장	2003/09/02
20	1001	김사랑	사원	2007/03/01
20	1012	강혜정	사원	2007/08/09
30	1006	장농건	부장	2003/10/09
30	1008	감우성	차장	2004/03/08
30	1003	오지호	과장	2005/02/10
30	1005	신농협	과장	2005/04/07
30	1011	조향기	사원	2007/03/01
30	1002	한예슬	대리	2007/04/02