

9장 데이터 검색 명령문 3

1. 개요,
2. 관계연산자를 사용하는 조건
3. AND, OR, NOT을 사용한 다중 조건
4. BETWEEN 연산자,
5. IN 연산자
6. LIKE 연산자
7. NULL 연산자
8. 부속질의어에서 IN 연산자
9. 부속질의어에서 관계 연산자
10. ALL과 ANY 연산자
11. EXISTS 연산자,
12. 부정 조건

9.1 개요

SQL은 FROM 절의 중간 결과에 나타난 각 행을 WHERE 조건이 참이면 WHERE 절의 중간 결과 테이블로 이동

● WHERE 절에서 사용할 수 있는 조건과 연산자

- 관계 연산자
- AND, OR, NOT과 결합된 조건
- BETWEEN 연산자
- IN 연산자
- LIKE 연산자
- NULL 연산자
- 부속 질의어 함께 사용되는 IN 연산자
- 부속 질의어 함께 사용되는 관계 연산자
- ANY와 ALL 연산자
- EXISTS 연산자

9.2 관계 연산자를 사용하는 조건

관계 연산자의 좌변과 우변에 있는 값을 비교하여 조건의 결과는 참, 거짓

관계 연산자	의 미
=	같다.
<	작다.
>	크다.
<=	작거나 같다.
>=	크거나 같다.
<>	같지 않다. (NOT = 사용)

다른 SQL 제품에서 관계 연산자 중 <> 연산자는 !=, ^=, # 등으로 사용됨

- 관계 연산에서 수식은 비교할 수 있는 데이터를 가지고 있어야 한다.
- 조건의 값 24는 자동적으로 문자수치 데이터로 변환되기 때문에 비교가 허용

WHERE STU_NAME = 24

- DATE 열은 날짜(DATE) 자료형을 가지고 있지만 수치 24는 수치 값이므로 비교가 허용되지 않는다.

WHERE DATE = 24

어떤 영수치 값이 다른 문자 수치 값에 비하여 영문자의 ASCII 코드 값 보다 더 적다면 다른 값 보다 적다고 할 수 있다.

조건	값
'Jim' < 'Pete'	TRUE
'Truck' >= 'Trek'	TRUE
'Jim' = 'JIM'	FALSE
'1985/12/08' < '1985/12/09'	TRUE
'1980/05/02' > '1979/12/31'	TRUE

[예제 9-1] 성별이 여자인 학생의 학번과 이름, 성별, 생년월일을 출력하라.

```
mysql> select stu_no, stu_name, gender, birthday  
-> from student  
-> where gender = 2 or gender = 4 or gender = 6;
```

stu_no	stu_name	gender	birthday
20161001	박정인	2	19970403
20181002	정인정	2	19990315
20181004	김유미	2	19990207
20191003	고혜진	4	20000307
20191006	최에스터	6	20021003
20191007	신안나	6	20011214
20191009	유하나	4	20000921

7 rows in set (0.00 sec)

성별(gender)이 2000년 이전에 태어난 여학생은 2, 2000년 이후에 태어난 여학생은 4, 외국인여학생의 경우는 6으로 입력되었다.

[예제9-2] 야간인 학생들의 학번과 이름을 출력하라.

```
mysql> select stu_no, stu_name  
-> from student  
-> where juya='야';
```

stu_no	stu_name
20191001	김유신
20191002	홍길동
20191004	이순신
20191008	연개소문
20201001	김영호
20201002	강감찬

[예제 9-3] 학번이 20191008인 학생의 학번과 이름을 나타내어라.

```
mysql> select stu_no, stu_name  
-> from student  
-> where stu_no = '20191008';
```

stu_no	stu_name
20191008	연개소문

[예제9-4]휴대폰을 가지고 있는 학생의 학번과 이름, 휴대폰 번호를 출력하라.

```
mysql> select stu_no, stu_name, mobile  
-> from student  
-> where mobile = mobile;
```

stu_no	stu_name	mobile
20141001	박도상	010-0611-9884
20161001	박정인	010-3142-1294
20181002	정인정	010-0605-7837
20181004	김유미	010-0617-1290
20191001	김유신	010-9876-1299
20191002	홍길동	010-6425-9245
20191004	이순신	010-7141-1860
20191005	김할리	010-4624-0460
20191007	신안나	010-5897-0874
20191008	연개소문	010-0641-9304
20191009	유하나	010-0651-0707
20201001	김영호	010-4605-5598
20201002	강감찬	010-1234-4567

13 rows in set (0.00 sec)

-MOBILE = MOBILE은 휴대폰번호가 동일하기 때문에 MOBILE 열이 채워져 있는 모든 행이 출력

-만약 MOBILE 열이 채워져 있지 않다면 조건은 알 수 없는 값으로 평가

-NULL 값이 나타나면 조건의 값은 알 수 없는 것으로 평가되어 출력 안함

9.3 AND, OR, NOT을 사용한 다중 조건

[예제9-5] 2000년 이전에 출생한 여학생의 학번,이름, 생년월일을 출력하라.

```
mysql> select stu_no, stu_name, gender, birthday  
-> from student  
-> where gender = 2 and substring(birthday , 1, 4) < 2000 ;
```

stu_no	stu_name	gender	birthday
20161001	박정인	2	19970403
20181002	정인정	2	19990315
20181004	김유미	2	19990207

3 rows in set (0.00 sec)

- 2000년 이전에 태어난 여학생의 경우는 GENDER 값이 2:
where gender = 2
- 2000년 이전에 태어난 학생의 조건 : SUBSTRING(BIRTHDAY, 1, 4) < 2000
스칼라 함수를 사용

[예제9-6] 경인지역에 거주하는 학생의 학번과이름, 집 전화번호를 나타내어라. (단 집 전화의 지역번호 02는 서울 거주자이고 031은 경기도 거주자이다. 집전화가 없는 경우는 제외한다.)

```
mysql> select stu_no, stu_name, concat(tel1, '-', tel2, '-', tel3) "집전화"  
-> from student  
-> where tel1 = '02' or tel1 = '031';
```

stu_no	stu_name	집전화
20141001	박도상	02-744-6126
20161001	박정인	02-652-2439
20181002	정인정	02-723-1078
20181003	이상진	031-691-5423
20181004	김유미	031-763-1439
20191001	김유신	02-685-7818
20191004	이순신	02-745-7667
20191005	김할리	02-746-5485
20191006	최에스터	02-945-6893
20191007	신안나	02-745-5485

10 rows in set (0.00 sec)

Concat()스칼라 함수를 사용하여 문자열을 연결시켜 출력해 주었다.

[예제 9-7] 2000년도에 태어나지 않은 학생의 학번, 이름, 생년월일을 나타내
어라.

```
mysql> select stu_no, stu_name, birthday  
-> from student  
-> where substring(birthday, 1, 4) <> 2000;
```

stu_no	stu_name	birthday
20141001	박도상	19960116
20161001	박정인	19970403
20181001	장수인	19990209
20181002	정인정	19990315
20181003	이상진	19990819
20181004	김유미	19990207
20191005	김할리	20010418
20191006	최에스터	20021003
20191007	신안나	20011214
20201001	김영호	20010811
20201002	강감찬	20010312

위의 예제를 다음과 같이 작성하여도 동일한 결과를 출력한다.

```
select stu_no, stu_name, birthday  
from student  
where not substring(birthday, 1, 4) = 2000;
```

[예제 9-8] 성별이 남자이거나 1999년에 출생한 학생의 학번, 이름, 성별, 생년월일을 나타내어라. 그러나 1988년도에 출생한 남학생은 제외한다.

```
mysql> select stu_no, stu_name, gender, birthday
-> from student
-> where (gender = 1 or substring(birthday,1,4) = 1999)
-> and not (gender = 1 and substring(birthday,1,4) = 1999);
```

stu_no	stu_name	gender	birthday
20141001	박도상	1	19960116
20181002	정인정	2	19990315
20181004	김유미	2	19990207

위의 결과처럼 1999년에 태어난 남자 20181001(장수인) 학생과 20181003(이상진) 학생은 출력되지 않는다.

9.4 BETWEEN 연산자

[예제 9-9] 현재 나이가 20살부터 23살까지 학생의 학번과 이름, 나이를 출력하라. (단, AGES 뷰테이블을 사용한다.)

```
mysql> select stu_no, stu_name, age  
-> from ages  
-> where age >= 20  
-> and age <= 23;
```

stu_no	stu_name	age
20161001	박정인	23
20181001	장수인	21
20181002	정인정	21
20181003	이상진	21
20181004	김유미	21
20191001	김유신	20
20191002	홍길동	20
20191003	고혜진	20
20191004	이순신	20
20191008	연개소문	20
20191009	유하나	20

11 rows in set (0.00 sec)

BETWEEN 연산자를 사용하여 다음과 같이 작성하면 동일한 결과 출력

```
mysql> select stu_no, stu_name, age  
-> from ages  
-> where age between 20 and 23;
```

[예제 9-10] 출생연도가 1997년부터 2000년 사이에 태어난 학생의 학번과 이름, 출생년도를 출력하라. 단, 출생년도 오름차순으로 출력하라.

```
mysql> select stu_no, stu_name, birthday  
-> from student  
-> where substring(birthday, 1, 4) between 1997 and 2000  
-> order by birthday;
```

stu_no	stu_name	birthday
20161001	박정인	19970403
20181004	김유미	19990207
20181001	장수인	19990209
20181002	정인정	19990315
20181003	이상진	19990819
20191004	이순신	20000222
20191003	고혜진	20000307
20191002	홍길동	20000402
20191008	연개소문	20000615
20191009	유하나	20000921
20191001	김유신	20001007

11 rows in set (0.00 sec)

위의 결과는 출생년도 오름차순으로 정렬된 점이다.

9.5 IN 연산자

[예제 9-11] 우편번호가 01066, 01901, 06305에 해당되는 각 학생의 학번, 이름, 우편번호를 출력하라.

```
mysql> select stu_no, stu_name, post_no  
-> from student  
-> where post_no = '01066'  
-> or post_no = '01901'  
-> or post_no = '06305';
```

stu_no	stu_name	post_no
20141001	박도상	01066
20191004	이순신	01901
20191007	신안나	06305

3 rows in set (0.00 sec)

IN 연산자를 사용하여 다음과 같이 작성하면 동일한 결과 출력

```
mysql> select stu_no, stu_name, post_no  
-> from student  
-> where post_no in('01066', '01901', '06305');
```

[예제 9-12] 1996, 1997, 2001년에 출생한 각 학생의 학번과 이름, 생년월일을 출력하라.

```
mysql> select stu_no, stu_name, birthday  
-> from student  
-> where substring(birthday, 1, 4) in (1996, 1997, 2001);
```

stu_no	stu_name	birthday
20141001	박도상	19960116
20161001	박정인	19970403
20191005	김할리	20010418
20191007	신안나	20011214
20201001	김영호	20010811
20201002	강감찬	20010312

- IN 연산자 다음에 사용되는 수식에 적용되는 규칙

- 수식의 자료형을 비교할 수 있어야 한다.
- 통계 함수는 수식으로 사용할 수 없다.

9.6 LIKE 연산자

LIKE 연산자는 특별한 패턴이나 마스크를 가지는 영수치 값을 선택할 때 사용

[예제 9-13] 영문이름이 문자 K 로 시작하는 학생의 학번과 이름, 영문이름을 나타내어라.

```
mysql> select stu_no, stu_name, stu_ename  
-> from student  
-> where stu_ename like 'K%';
```

stu_no	stu_name	stu_ename
20181004	김유미	Kim Yoo-Mi
20191001	김유신	Kim Yoo-Shin
20191003	고혜진	Ko Hea-Jin
20191005	김할리	Kim Hal-Li
20201001	김영호	Kim Young-Ho

LIKE 연산자 다음에 사용되는 퍼센트 기호(%)는 특별한 의미를 가지고 있는데, 이를 패턴(pattern) 또는 마스크(mask)라 한다.

- 퍼센트 기호(%) : 0 또는 임의의 모든 문자를 의미(dos명령어 *과 동일)
- 밑줄(_) : 0 또는 임의의 문자 1개를 의미(dos명령어 ?과 동일)

[예제9-14] 영문이름의 맨 끝에 문자g를 가지고 있는 학생의 학번과 이름을 출력하라.

```
mysql> select stu_no, stu_ename  
      -> from student  
      -> where stu_ename like '%g';
```

stu_no	stu_ename
20141001	Park Do-Sang
20181002	Jung In-Jung
20191002	Hong Gil-Dong

[예제 9-15] 영문이름의 끝에서 두 번째 문자가 i인 학생의 학번과 이름을 출력하라.

```
mysql> select stu_no, stu_ename  
      -> from student  
      -> where stu_ename like '%i_';
```

stu_no	stu_ename
20161001	Park Jung-In
20181001	Jang Soo-In
20181003	Lee Sang-Gin
20191001	Kim Yoo-Shin
20191003	Ko Hea-Jin
20191004	Lee Sun-Shin

[예제 9-16] 영문이름이 문자 K 로 시작하지 않는 학생의 학번과 이름을 출력하라.

```
mysql> select stu_no, stu_ename  
-> from student  
-> where not (stu_ename like 'K%');
```

stu_no	stu_ename
20141001	Park Do-Sang
20161001	Park Jung-In
20181001	Jang Soo-In
20181002	Jung In-Jung
20181003	Lee Sang-Gin
20191002	Hong Gil-Dong
20191004	Lee Sun-Shin
20191006	Choi Esther
20191007	Shin An-Na
20191008	Yean Gae-So-Moon
20191009	Yoo Ha-Na
20201002	Gang Gam-Chan

12 rows in set (0.00 sec)

9.7 NULL 연산자

NULL 연산자는 특정한 열이 값을 가지고 있지 않는 행을 선택할 때 사용

[예제9-17]휴대폰을 가지고 있는 학생의 학번과 이름, 휴대폰 번호를 출력하라

```
mysql> select stu_no, stu_name, mobile  
-> from student  
-> where mobile is not null;
```

stu_no	stu_name	mobile
20141001	박도상	010-0611-9884
20161001	박정인	010-3142-1294
20181002	정인정	010-0605-7837
20181004	김유미	010-0617-1290
20191001	김유신	010-9876-1299
20191002	홍길동	010-6425-9245
20191004	이순신	010-7141-1860
20191005	김할리	010-4624-0460
20191007	신안나	010-5897-0874
20191008	연개소문	010-0641-9304
20191009	유하나	010-0651-0707
20201001	김영호	010-4605-5598
20201002	강감찬	010-1234-4567

13 rows in set (0.00 sec)

[예제9-18] 휴대폰을 가지고 있지 않은 학생의 학번과 이름, 휴대폰 번호가 NULL인 경우에는 “휴대폰없음”을 나타내어라.

```
mysql> select stu_no, stu_name, ifnull(mobile, '휴대폰없음')  
      -> from student  
      -> where mobile is null;
```

stu_no	stu_name	ifnull(mobile, '휴대폰없음')
20181001	장수인	휴대폰없음
20181003	이상진	휴대폰없음
20191003	고혜진	휴대폰없음
20191006	최에스터	휴대폰없음

4 rows in set (0.01 sec)

[예제 9-19] 학생의 휴대폰번호(MOBILE)가 010이 아닌 모든 학생의 학번과 이름, 휴대폰번호를 출력하라. (단, 휴대폰이 없는 학생도 포함되어 출력되어야 한다.)

```
mysql> select stu_no, stu_name, mobile  
-> from student  
-> where substring(mobile, 1, 3) <> '010'  
-> or mobile is null;
```

stu_no	stu_name	mobile
20181001	장수인	NULL
20181003	이상진	NULL
20191003	고혜진	NULL
20191006	최에스터	NULL

4 rows in set (0.00 sec)

9.8 부속 질의어에서 IN 연산자

등록을 한 각 학생의 학번, 이름을 출력하라. 단, 중복된 값을 사용할 필요는 없으므로 distinct를 사용한다.

```
mysql> select distinct stu_no  
      -> from fee;
```

stu_no
20141001
20161001
20191004
20191005
20191006
20191007
20191008
20201002

```
8 rows in set (0.02 sec)
```

[예제 9-20] 등록을 한 각 학생의 학번, 이름을 출력하라. (in연산자를 이용)

```
mysql> select stu_no, stu_name  
-> from student  
-> where stu_no IN (20141001, 20161001, 20191004, 20191005,  
->                20191006, 20191007, 20191008, 20201002);
```

stu_no	stu_name
20141001	박도상
20161001	박정인
20191004	이순신
20191005	김할리
20191006	최에스터
20191007	신안나
20191008	연개소문
20201002	강감찬

8 rows in set (0.01 sec)

[예제 9-21] 부속질의어를 이용하여 등록을 한 각 학생의 학번, 이름을 출력하라.

```
mysql> select stu_no, stu_name  
-> from student  
-> where stu_no in  
->      (select stu_no  
->      from fee);
```

stu_no	stu_name
20141001	박도상
20161001	박정인
20191004	이순신
20191005	김할리
20191006	최에스터
20191007	신안나
20191008	연개소문
20201002	강감찬

8 rows in set (0.01 sec)

[예제 9-22] 적어도 한 번의 장학금을 받았던 학생의 학번과 이름을 출력하라.

```
mysql> select stu_no, stu_name  
-> from student  
-> where stu_no in  
-> (select stu_no  
-> from fee  
-> where jang_total > 0);
```

stu_no	stu_name
20141001	박도상
20161001	박정인
20191004	이순신
20191005	김할리
20191006	최에스터
20191007	신안나
20191008	연개소문
20201002	강감찬

8 rows in set (0.01 sec)

[예제 9-23] “20191009”인 학생이 가입한 동아리를 제외한 다른 동아리에 적어도 한 번 가입을 한 학생의 학번과 이름을 출력하라.

```
mysql> select stu_no, stu_name
      -> from student
      -> where stu_no in
      -> (select stu_no
      -> from circle
      -> where cir_name not in
      -> (select cir_name
      -> from circle
      -> where stu_no = '20191009'));
```

stu_no	stu_name
20181001	장수인
20191004	이순신
20161001	박정인
20191002	홍길동

4 rows in set (0.01 sec)

[예제 9-24] 휴대폰을 가지고 있는 학생을 출력하라.(단, 휴대폰이 있어도 야간인 학생은 제외한다.)

```
mysql> select stu_no, stu_name, mobile
-> from student
-> where mobile not in
-> (select mobile
-> from student
-> where juya = '야');
```

stu_no	stu_name	mobile
20141001	박도상	010-0611-9884
20161001	박정인	010-3142-1294
20181002	정인정	010-0605-7837
20181004	김유미	010-0617-1290
20191005	김할리	010-4624-0460
20191007	신안나	010-5897-0874
20191009	유하나	010-0651-0707

7 rows in set (0.00 sec)

학번 “**20141001**” 박도상학생은 주간 학생이므로 출력, “**20191001**”번 김유신은 야간이므로 제외되었다. 휴대폰이 없는 “**20181001**” 장수인 학생도 제외되었고, 휴대폰을 가지고 있고 주간인 학생만 최종 결과에 포함

9.9 부속 질의어에서 관계 연산자

[예제 9-25] 수강신청을 한 학생의 학번과 이름을 출력하라.

```
mmysql> select stu_no, stu_name  
-> from student  
-> where stu_no in  
-> (select stu_no  
-> from attend  
-> where att_div = 'Y');
```

stu_no	stu_name
20141001	박도상
20161001	박정인

2 rows in set (0.01 sec)

[예제 9-26] 장수인(1999년생) 보다나이가더많은각학생의학번과이름, 생년월일을 출력하라.

```
mysql> select stu_no, stu_name, birthday
-> from student
-> where substring(birthday,1,4) <
-> (select substring(birthday, 1,4)
-> from student
-> where stu_name = '장수인');
```

stu_no	stu_name	birthday
20141001	박도상	19960116
20161001	박정인	19970403

2 rows in set (0.00 sec)

9.10 ALL과 ANY 연산자

[예제 9-27] 가장 나이가 많은 학생의 학번, 이름, 생년월일을 출력하라.
(단, 가장 나이가 많은 학생은 다른 학생의 생년월일보다 생년월일이 더 적거나 같은 생년월일을 가진 학생이다.)

```
mysql> select stu_no, stu_name, birthday
-> from student
-> where birthday <= all
-> (select birthday
-> from student);
```

+	-----+	-----+	-----+			
	stu_no		stu_name		birthday	
+	-----+	-----+	-----+			
	20141001		박도상		19960116	
+	-----+	-----+	-----+			

```
1 row in set (0.00 sec)
```

[예제 9-28] 가장 나이가 많은 학생(박도상)을 제외한 나머지 모든 학생의 학번, 이름, 생년월일을 출력하라.

```
mysql> select stu_no, stu_name, birthday  
-> from student  
-> where birthday > any  
-> (select birthday  
-> from student);
```

stu_no	stu_name	birthday
20161001	박정인	19970403
20181001	장수인	19990209
20181002	정인정	19990315
20181003	이상진	19990819
20181004	김유미	19990207
20191001	김유신	20001007
20191002	홍길동	20000402
20191003	고혜진	20000307
20191004	이순신	20000222
20191005	김할리	20010418
20191006	최에스터	20021003
20191007	신안나	20011214
20191008	연개소문	20000615
20191009	유하나	20000921
20201001	김영호	20010811
20201002	강감찬	20010312

16 rows in set (0.00 sec)

[예제 9-29] 학번 20191004인 학생이 등록한 등록금의 납부총액(1,000,000)보다 더 많은 등록금을 낸 학생의 학번과 납부총액을 출력하라. 이때 20191004번은 결과에서 제외한다.

```
mysql> select distinct stu_no, fee_pay
-> from fee
-> where stu_no <> '20191004'
-> and fee_pay > any
-> (select fee_pay
-> from fee
-> where stu_no = '20191004');
```

stu_no	fee_pay
20141001	3000000
20141001	2200000
20141001	2000000
20191005	3000000
20191006	3000000
20191007	3000000
20191008	3000000
20201002	3000000

8 rows in set (0.00 sec)

9.11 EXISTS 연산자

[예제 9-30] 등록을 한 학생의 학번과 이름을 출력하라.

```
mysql> select stu_no, stu_name  
-> from student  
-> where stu_no in  
-> (select stu_no  
-> from fee);
```

stu_no	stu_name
20141001	박도상
20161001	박정인
20191004	이순신
20191005	김할리
20191006	최에스터
20191007	신안나
20191008	연개소문
20201002	강감찬

8 rows in set (0.00 sec)

EXISTS 연산자를 사용하여 다음과 같이 작성하면 동일한 결과 출력

```
mysql> select stu_no, stu_name  
-> from student  
-> where exists  
-> (select * from fee  
-> where stu_no = student.stu_no);
```

[예제 9-31] 등록하지 않은 학생의 학번과 이름을 출력하라.

```
mysql> select stu_no, stu_name  
-> from student  
-> where not exists  
-> (select *  
-> from fee  
-> where stu_no = student.stu_no);
```

stu_no	stu_name
20181001	장수인
20181002	정인정
20181003	이상진
20181004	김유미
20191001	김유신
20191002	홍길동
20191003	고혜진
20191009	유하나
20201001	김영호

9 rows in set (0.00 sec)

EXISTS 연산자를 사용하여 다음과 같이 작성하면 동일한 결과 출력

```
mysql> select stu_no, stu_name  
-> from student  
-> where not exists  
-> (select 'Nothing'  
-> from fee  
-> where stu_no = student.stu_no);
```

[예제 9-32] 학적 테이블에서 학번, 이름, 휴대폰번호, 우편번호를 출력하라.

```
mysql> select stu_no, stu_name, mobile, post_no from student;
```

stu_no	stu_name	mobile	post_no
20141001	박도상	010-0611-9884	01066
20161001	박정인	010-3142-1294	04957
20181001	장수인	NULL	57991
20181002	정인정	010-0605-7837	05270
20181003	이상진	NULL	17826
20181004	김유미	010-0617-1290	15348
20191001	김유신	010-9876-1299	06034
20191002	홍길동	010-6425-9245	59635
20191003	고혜진	NULL	47783
20191004	이순신	010-7141-1860	01901
20191005	김할리	010-4624-0460	02463
20191006	최에스터	NULL	03975
20191007	신안나	010-5897-0874	06305
20191008	연개소문	010-0641-9304	48020
20191009	유하나	010-0651-0707	61053
20201001	김영호	010-4605-5598	61689
20201002	강감찬	010-1234-4567	34331

17 rows in set (0.00 sec)

[예제 9-33] 각각의 도시에 거주하는 모든 학생에 대하여 휴대폰을 가지고 있는 학생의 학번과 이름 우편번호, 휴대폰 번호를 나타내어라.(단, 휴대폰이 있는 학생과 휴대폰이 없는 학생의 우편번호 앞 1자리가 동일한 학생은 제외시킨다)

```
mysql> select stu_no, stu_name, post_no, mobile
-> from student s1
-> where not mobile in
-> (select mobile
-> from student s2
-> where substring(s1.post_no, 1, 1) = substring(s2.post_no, 1, 1)
-> and mobile is null);
```

stu_no	stu_name	post_no	mobile
20191009	유하나	61053	010-0651-0707
20201001	김영호	61689	010-4605-5598
20201002	강감찬	34331	010-1234-4567

3 rows in set (0.00 sec)

위의 실행결과는 휴대폰이 없는 학생과 휴대폰이 있어도 휴대폰이 없는 학생의 우편번호 앞 1자리가 같으면 제외되었다. 장수인 학생의 우편번호(57991)과 이상진 학생의 우편번호(17826), 고혜진 학생의 우편번호(47783), 최예스터 학생의 우편번호(03975)의 앞 1번째 자리는 (0, 1, 4, 5)이므로, 학생들이 휴대폰을 가지고 있더라도 우편번호 앞 1자리가 (0, 1, 4, 5)로 시작하는 학생들은 제외된다. 여기에 해당되는 학생들 박도상, 박정인, 정인정, 김유미, 김유신, 홍길동, 이순신, 김할리, 신안나, 연개소문 학생은 휴대폰이 있지만 제외되었다.

9.11 부정 조건

- SQL에서 자주 발생하는 오류는 부정 조건을 가지고 있는 오류
- 부정 조건은 긍정 조건 앞에 NOT을 위치시킴으로써 만듦

[예제 9-34] “Java길라잡이” 동아리에 가입한 학생의 학번과 이름을 출력하라.

```
mysql> select stu_no, stu_name  
-> from circle  
-> where cir_name = 'Java길라잡이';
```

stu_no	stu_name
20181001	장수인
20191004	이순신
20161001	박정인

3 rows in set (0.00 sec)

[예제 9-35] “Java길라잡이” 동아리에 가입하지 않은 학생의 학번과 이름을 출력하라.

```
mysql> select stu_no, stu_name, cir_name  
-> from circle  
-> where cir_name <> 'Java길라잡이';
```

stu_no	stu_name	cir_name
20141001	박도상	컴맹탈출
20191009	유하나	컴맹탈출
20191001	김유신	컴맹탈출
20191002	홍길동	PHP길라잡이

4 rows in set (0.00 sec)

NOT = 연산자를 사용하여 다음과 같이 작성하면 동일한 결과 출력

```
mysql> select stu_no, stu_name, cir_name  
-> from circle  
-> where not cir_name = 'Java길라잡이';
```

[예제 9-36] 등록테이블에서 장학코드가 '11' 학생의 학번과 장학코드, 장학금 총액을 출력하라.

```
mysql> select stu_no, jang_code, jang_total  
-> from fee  
-> where jang_code = 11;
```

stu_no	jang_code	jang_total
20141001	11	2000000
20161001	11	2000000
20191004	11	2000000

3 rows in set (0.00 sec)

[예제 9-37] 등록테이블에서 장학코드가 '11'이 아닌 학생의 학번과 장학코드, 장학금총액을 출력하라.

```
mysql> select stu_no, jang_code, jang_total  
-> from fee  
-> where jang_code <> 11;
```

stu_no	jang_code	jang_total
20141001	1	500000
20141001	10	2500000
20141001	21	800000
20141001	2	1000000
20141001	10	2500000
20141001	10	2500000
20141001	10	2500000
20161001	10	2500000
20161001	10	2500000
20161001	10	2500000
20191004	1	500000
20191005	1	500000
20191006	1	500000
20191007	1	500000
20191008	1	500000
20201002	1	500000
20201002	10	2500000

17 rows in set (0.00 sec)

장학코드가 11이거나,
NULL인 경우가 제외
되어 원하는 결과 값이
아니다.

[예제 9-38] 등록테이블에서 장학코드가 '11'이 아닌 학생의 학번과 장학코드, 장학금총액을 출력하라.(NOT IN 이용)

```
mysql> select stu_no, jang_code, jang_total  
-> from fee  
-> where jang_code not in  
-> (select jang_code  
-> from fee  
-> where jang_code in (11));
```

stu_no	jang_code	jang_total
20141001	1	500000
20141001	10	2500000
20141001	21	800000
20141001	2	1000000
20141001	10	2500000
20141001	10	2500000
20141001	10	2500000
20161001	10	2500000
20161001	10	2500000
20161001	10	2500000
20191004	1	500000
20191005	1	500000
20191006	1	500000
20191007	1	500000
20191008	1	500000
20201002	1	500000
20201002	10	2500000

17 rows in set (0.00 sec)

```
mysql> select stu_no, jang_code, jang_total  
-> from fee  
-> where jang_code not in  
-> (select jang_code  
-> from fee  
-> where jang_code = 11);
```

장학코드가 11이거나,
NULL인 경우가 제외
되어 원하는 결과 값이
아니다.

[예제 9-39] 등록테이블에서 장학코드가 '11'이 아닌 학생의 학번과 장학코드, 장학금총액을 출력하라.(단, NOT IN 이용하고 장학코드가 NULL인 학생도 포함하여 출력하라.)

```
mysql> select stu_no, jang_code, jang_total
-> from fee
-> where jang_code not in
-> (select jang_code
-> from fee
-> where jang_code in(11))
-> or jang_code is null;
```

```
mysql> select stu_no, jang_code, jang_total
-> from fee
-> where jang_code <> 11 or
-> jang_code is null;
```

좌우 select 명령문은 동일한 결과 값 출력

stu_no	jang_code	jang_total
20081001	NULL	NULL
20081002	NULL	NULL
20141001	1	500000
20141001	10	2500000
20141001	21	800000
20141001	2	1000000
20141001	10	2500000
20141001	10	2500000
20141001	10	2500000
20161001	10	2500000
20161001	10	2500000
20161001	10	2500000
20191004	1	500000
20191005	1	500000
20191005	NULL	100000
20191006	1	500000
20191006	NULL	100000
20191007	1	500000
20191007	NULL	100000
20191008	1	500000
20191008	NULL	100000
20201002	1	500000
20201002	10	2500000

23 rows in set (0.00 sec)