

학습목표

1. LIKE 검색
2. NULL값
3. NULL 함수
4. GROUP BY / HAVING

학습내용

SELECT문에서 LIKE 검색(검색어의 일부만으로 조회), NULL값, NULL함수, 집합함수 (GROUP BY/ HAVING)를 활용할 수 있습니다.

사전퀴즈

1. LIKE 검색에서 사용하는 와일드 카드는 *와 ?가 있다.

정답 : X

해설

LIKE 검색에서 사용하는 와일드 카드는 %와 _를 사용한다. %는 0자 이상 여러 자에 매칭되는 와일드카드 문자이고, _는 1자에 대응하는 와일드카드 문자이다.

2. SELECT 문자에서 컬럼 값이 널(NULL)인 레코드를 찾으려면 = NULL을 사용한다.

정답 : X

해설 : 컬럼 값이 NULL인 레코드를 검색하려면 IS NULL을 사용해야 한다.

수업

1. LIKE 검색

LIKE 검색이란?

- 정확한 키워드를 모를 경우 일부만으로 검색하는 방법
- 와일드카드(% , _)를 사용하여 패턴매칭

SELECT 컬럼명 FROM 테이블명 WHERE 컬럼명 LIKE 패턴

- 와일드카드(Wildcard)

% : 0 - n 글자, _ : 1글자

- LIKE의 검색은 매칭하기 위해 DBMS에 부담이 많으므로 LIKE에 OR와 같은 논리조건자를 중복해서 사용하지 않는게 좋음

(바람직하지 않은 예)

SELECT * FROM 테이블명 WHERE 컬럼명1 LIKE ... OR 컬럼명2 LIKE ... ;

예제 1-1

city 테이블에서 국가코드가 K로 시작하는/끝나는/중간에 들어있는 국가코드를 표시하시오.

```
1 -- K로 시작하는 국가코드
2 select CountryCode
3 from city
4 where CountryCode like 'K%'
5 ;
6
7 -- K로 끝나는 국가코드
8 select CountryCode
9 from city
10 where CountryCode like '%K'
11 ;
12
13 -- K가 중간에 있는 국가코드
14 select CountryCode
15 from city
16 where CountryCode like '%K%'
17 ;
```

예제 1-2

city 테이블에서 국가코드가 K로 시작하는 3글자 국가코드를 표시하시오.

```
19 -- K로 시작하는 3글자 국가코드 |
20 select CountryCode
21 from city
22 where CountryCode like 'K__'
23 ;
```

2. NULL값

- NULL이란 해당 컬럼의 값이 없다는 의미 (해당 컬럼이 NULL값을 허용하는 경우)
- NULL값을 가지고 있는 컬럼을 검색하려면 **IS NULL**
- NULL이 아닌 값을 가지고 있는 컬럼을 검색하려면 **IS NOT NULL**

예제 2-1

country 테이블에서 기대수명(Life Expectancy)이 없는 국가 개수를 표시하시오

```
26 select count(*)
27 from country
28 where LifeExpectancy IS NULL
29 ;
30
```

결과 #1 (1x1)
count(*) 17

예제 2-2

country 테이블에서 기대수명(Life Expectancy)이 있는 국가 개수를 표시하시오

```
31 select count(*)
32 from country
33 where LifeExpectancy IS NOT null
34 ;
```

결과 #1 (1x1)
count(*) 222

3. NULL 함수

- 숫자컬럼을 연산해야 할 때 NULL을 처리해주는 함수
- NULL값이 나오면 다른 값(주로 0)으로 대체해서 계산에 문제없도록 처리
- MariaDB VS. SQL Server VS. ORACLE
 - MariaDB** : IFNULL/COALESCE
 - SQL Server** : ISNULL
 - ORACLE** : NVL
- 숫자연산(예: sum) / 집합함수(예: count)의 경우는 처리가 내장되어 있음
- 직접 함수나 쿼리에 넣는 경우는 NULL 함수를 사용해야 함

예제 3-1

country 테이블의 기대수명의 평균값을 표시하시오. (NULL값 미반영/반영)

=> 두 값이 다른이유? 기대수명의 합은 동일한데, 제수의 개수가 다르기 때문

(참고 - a 나누기 b는 c 라고 했을 때, a는 피제수, b는 제수, c는 몫이라고 함)

```
36 -- NULL값 미 반영
37 select avg(LifeExpectancy)
38 from country
39 ;
40
```

결과 #1 (1x1)
avg(LifeExpectancy) 66.48604

```
41 -- NULL값 반영
42 select avg(coalesce(LifeExpectancy, 0))
43 from country
44 ;
45
46 -- NULL값 미 반영
```

결과 #1 (1x1)
avg(coalesce(LifeExpectancy, 0)) 61.75690

예제 3-2

country 테이블의 기대수명값이 들어있는 개수를 표시하시오 (NULL값 미반영/반영)

*** 주의 ***

count(*)의 경우 NULL값을 반영한다

```

46 -- NULL 값 미 반영
47 select count(LifeExpectancy)
48 from country
49 ;
50

```

결과 #1 (1×1)

count(LifeExpectancy)	222
-----------------------	-----

```

51 -- NULL 값 반영
52 select count(ifnull(LifeExpectancy, 0))
53 from country
54 ;
55

```

결과 #1 (1×1)

count(ifnull(LifeExpectancy, 0))	239
----------------------------------	-----

```

56 select count(*)
57 from country
58 ;
59

```

결과 #1 (1×1)

count(*)	239
----------	-----

4. GROUP BY / HAVING

GROUP BY

- 집합함수와 같이 사용해 그룹별 연산을 적용한다.

select 컬럼명, 집합함수명(컬럼명) from 테이블명 group by 컬럼명;

예제 4-1

city 테이블의 국가코드별 도시숫자를 구하시오

```

68 select CountryCode, count(CountryCode)
69 from city
70 group by CountryCode
71 ;

```

city (2×232)	
CountryCode	count(CountryCode)
ABW	1
AFG	4
AGO	5
AIA	2
ALB	1
AND	1
ANT	1
ARE	5
ARG	57
ARM	3
ASM	2
ATG	1

HAVING

- HAVING은 집합연산에 WHERE 조건절 대체로 사용

예제 4-2

city 테이블의 국가코드별 도시숫자를 구하시오

(단, 70개 이상의 도시를 가지는 국가만 표시하시오)

```

68 select CountryCode, count(CountryCode)
69 from city
70 group by CountryCode
71 having count(CountryCode) >= 70
72 ;
73

```

city (2×12)	
CountryCode	count(CountryCode)
BRA	250
CHN	363
DEU	93
GBR	81
IDN	85
IND	341
JPN	248
KOR	70
MEX	173
PHL	136
RUS	189
USA	274

GROUP BY 연산과 일반 쿼리의 차이점?

전문가의견

GROUP BY는 아래 쿼리들의 합입니다.

world DB의 city 중 국가코드가 'KOR'인 도시의 숫자를 출력하시오.

world DB의 city 중 국가코드가 'JPN'인 도시의 숫자를 출력하시오.

world DB의 city 중 국가코드가 'CHN'인 도시의 숫자를 출력하시오.

world DB의 city 중 국가코드가 'USA'인 도시의 숫자를 출력하시오.

모든 국가별 도시 숫자를 한꺼번에 출력해주는 면에서 개별쿼리와 차이가 납니다.

개별국가들의 도시 숫자를 원하는 것인지, 모든 국가들의 도시숫자를 원하는 것인지 구분해서 사용해야 합니다.