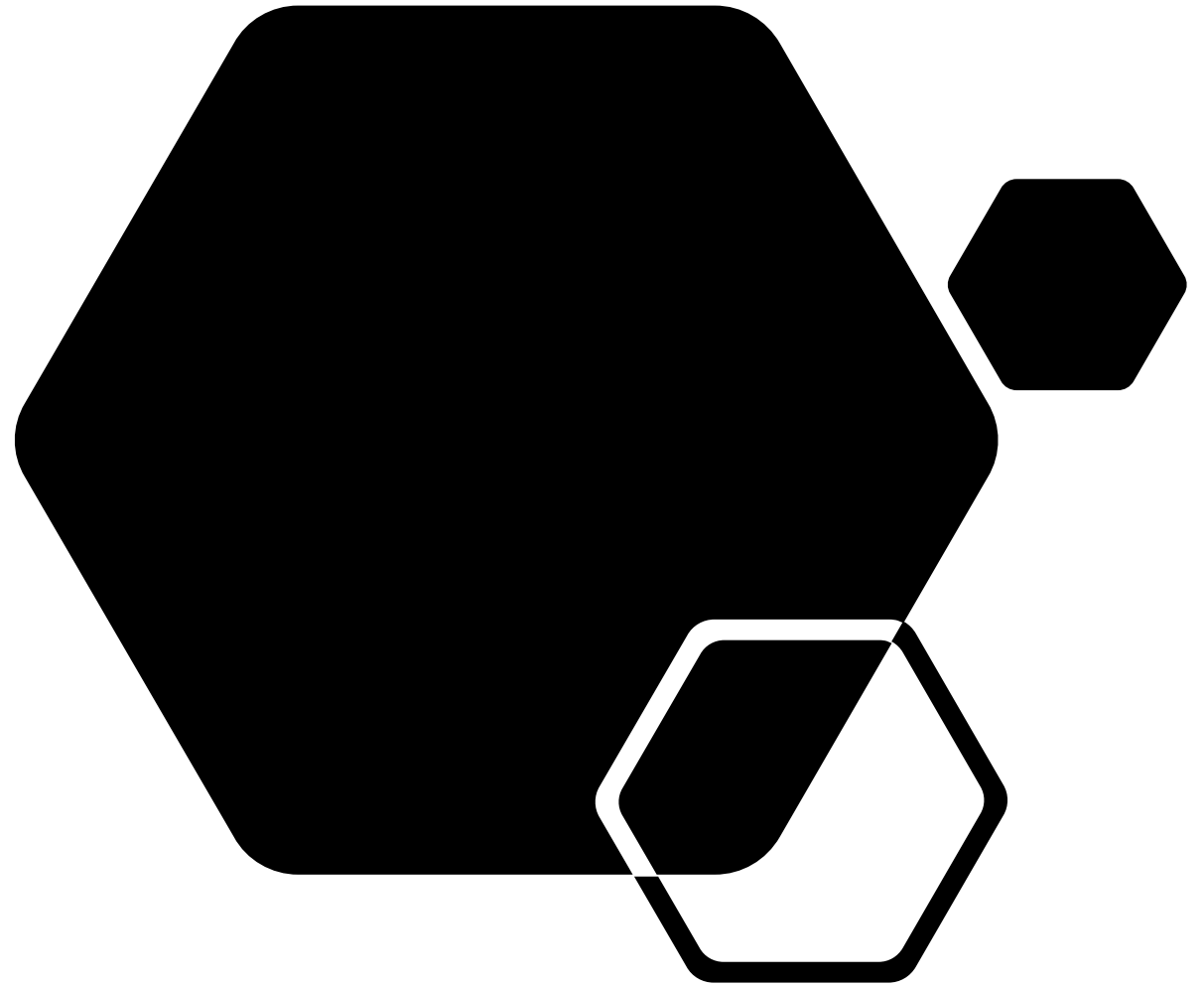


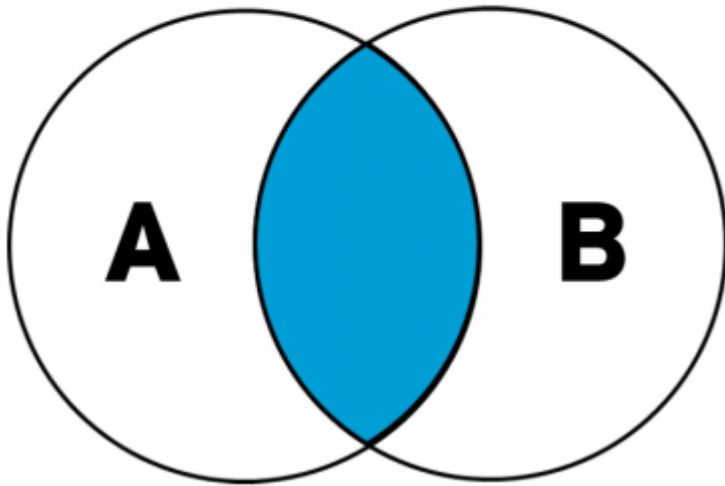
SQL Join



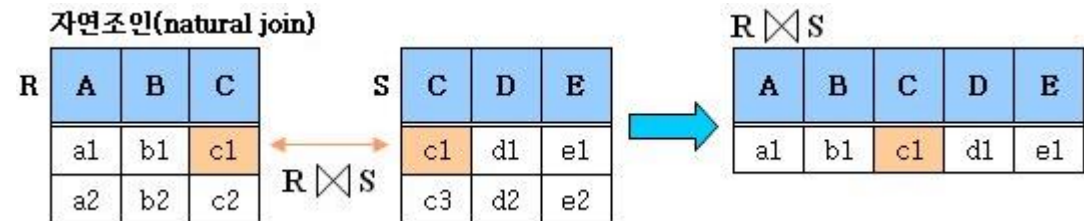
- 조인을 쓰면 두 개의 테이블을 연결하여 원하는 데이터를 추출할 수 있습니다.
 - 연결 역할을 하는 것이 조인 칼럼으로 연결 대상인 두 테이블에서 같은 값을 가진 칼럼을 말함
- 두 테이블의 조인을 위해서는 기본키(PRIMARY KEY, PK)와 외래키(FOREIGN KEY, FK) 관계로 맺어져야 하고, 이를 일대다 관계라고 합니다
- RDBMS의 특징 중 하나가 중복 데이터 저장을 최소화하는 것

- 조인의 특징
 - 조인하는 테이블에는 같은 값을 가진 칼럼이 있어야 한다
 - 각 조인 칼럼의 이름이 같을 필요는 없으나 값은 같아야 한다.
 - 2개 이상의 테이블도 조인 가능
 - 만약 원하는 정보가 3개나 4개 테이블에 흩어져 있어도 조인 컬럼이 있다면 하나의 SELECT 문장에서 여러 테이블을 조인해 원하는 정보를 조회 가능
 - 조인할 때 테이블에 대한 별칭을 사용
 - 조인 시 조인 조건이 필요

INNER JOIN

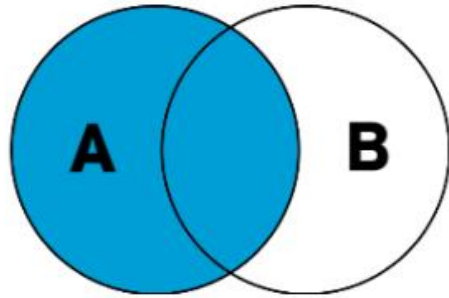


```
SELECT <열 목록>  
FROM <첫 번째 테이블> A  
    INNER JOIN <두 번째 테이블> B  
    ON A.key = B.key
```

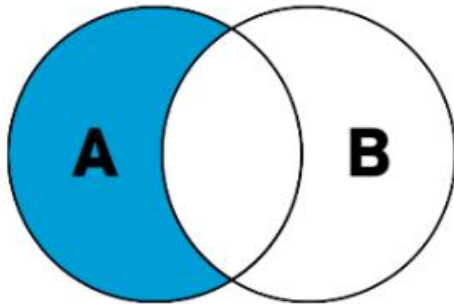


두 테이블을 연결할 때 가장 많이 사용하는 것이 내부 조인입니다. 그냥 조인이라고 부르면 내부 조인을 의미합니다.

LEFT OUTER JOIN

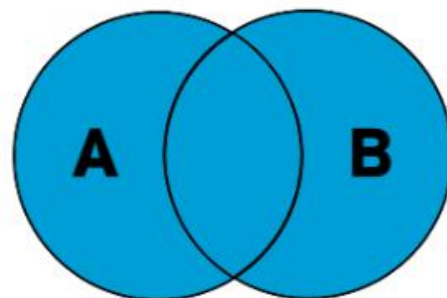


```
SELECT *  
FROM A a  
LEFT JOIN B b  
ON a.KEY = b.KEY
```

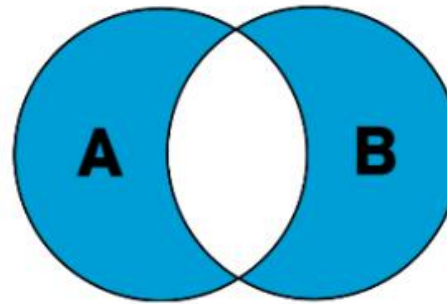


```
SELECT *  
FROM A a  
LEFT JOIN B b  
ON a.KEY = b.KEY  
WHERE b.KEY IS NULL
```

FULL OUTER JOIN

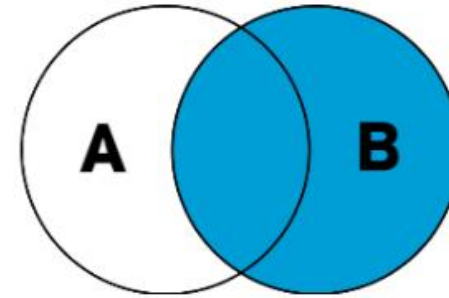


```
SELECT *  
FROM A a  
FULL OUTER JOIN B b  
ON a.KEY = b.KEY
```

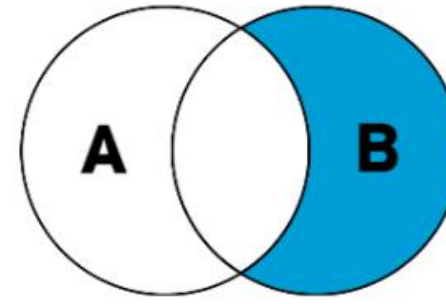


```
SELECT *  
FROM A a  
FULL OUTER JOIN B b  
ON a.KEY = b.KEY  
WHERE a.KEY IS NULL  
OR b.KEY IS NULL
```

RIGHT OUTER JOIN

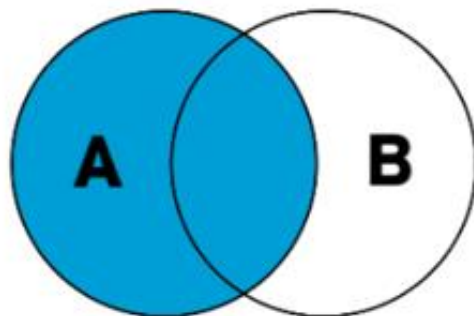


```
SELECT *  
FROM A a  
RIGHT OUTER JOIN B b  
ON a.KEY = b.KEY
```

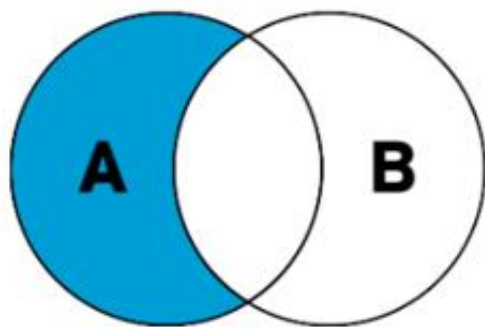


```
SELECT *  
FROM A a  
RIGHT OUTER JOIN B b  
ON a.KEY = b.KEY  
WHERE a.KEY IS NULL
```

LEFT OUTER JOIN



```
SELECT *
FROM A a
LEFT JOIN B b
ON a.KEY = b.KEY
```



```
SELECT *
FROM A a
LEFT JOIN B b
ON a.KEY = b.KEY
WHERE b.KEY IS NULL
```

왼쪽 외부조인(left outer join)

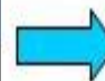
R

A	B	C
a1	b1	c1
a2	b2	c2

S

C	D	E
c1	d1	e1
c3	d2	e2

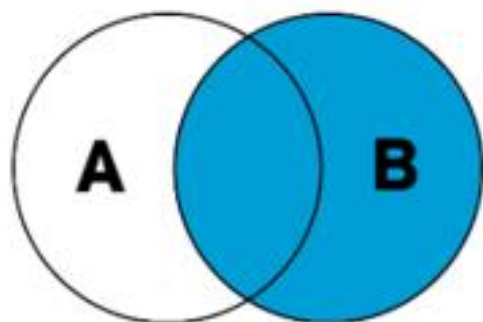
$R \bowtie S$



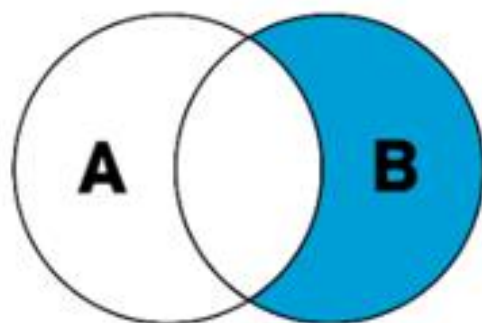
$R \bowtie S$

A	B	C	D	E
a1	b1	c1	d1	e1
a2	b2	c2	null	null

RIGHT OUTER JOIN



```
SELECT *
FROM A a
      RIGHT OUTER JOIN B b
      ON a.KEY = b.KEY
```



```
SELECT *
FROM A a
      RIGHT OUTER JOIN B b
      ON a.KEY = b.KEY
WHERE a.KEY IS NULL
```

오른쪽 외부조인(right outer join)

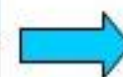
R

A	B	C
a1	b1	c1
a2	b2	c2

$R \bowtie S$

S

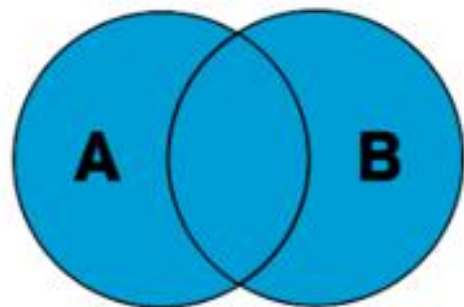
C	D	E
c1	d1	e1
c3	d2	e2



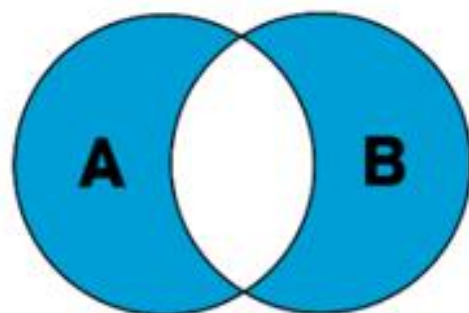
$R \bowtie S$

A	B	C	D	E
a1	b1	c1	d1	e1
null	null	c3	d2	e2

FULL OUTER JOIN

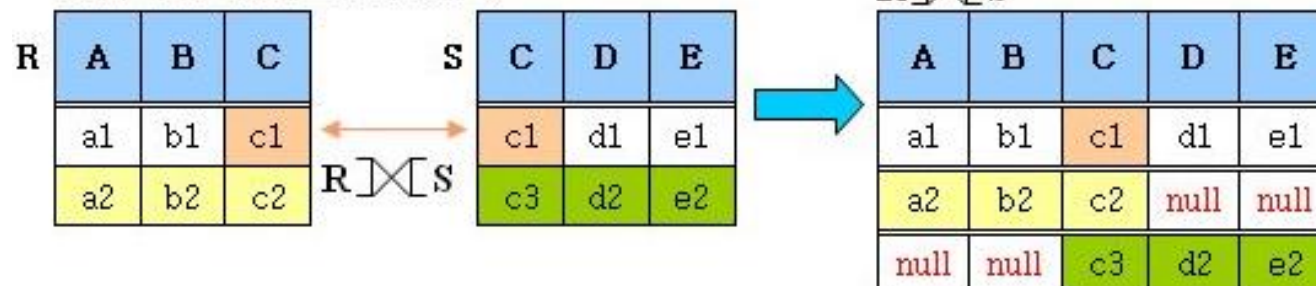


```
SELECT *
FROM A a
FULL OUTER JOIN B b
ON a.KEY = b.KEY
```



```
SELECT *
FROM A a
FULL OUTER JOIN B b
ON a.KEY = b.KEY
WHERE a.KEY IS NULL
OR b.KEY IS NULL
```

완전 외부조인(full outer join)



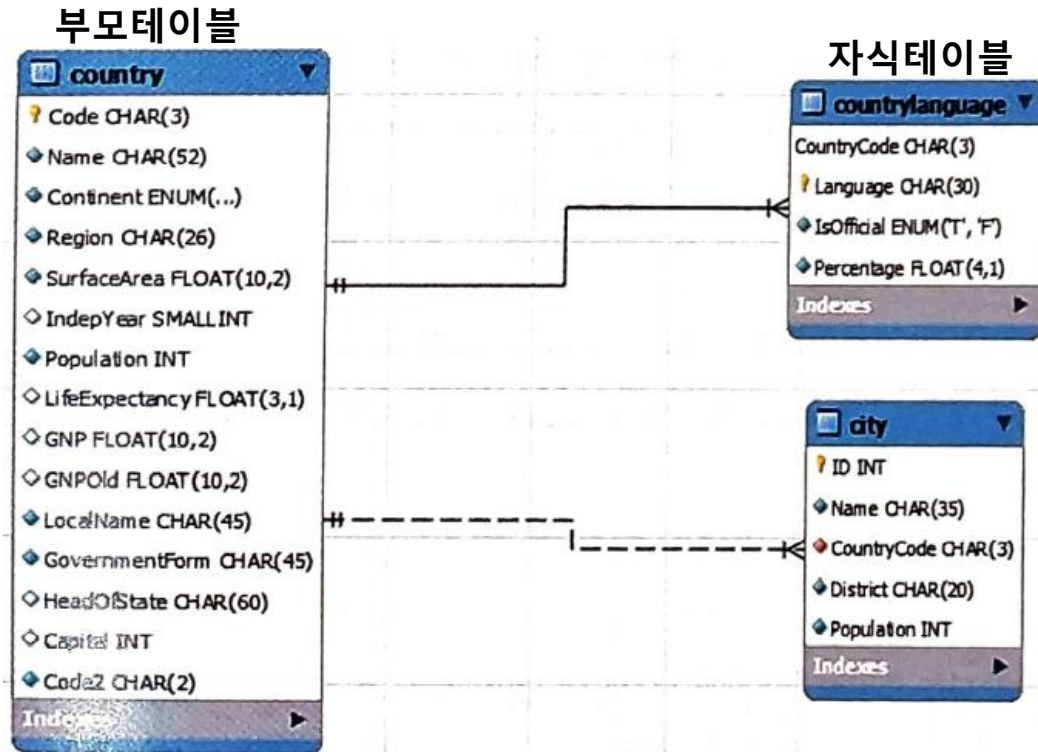
- world 데이터베이스

테이블명	테이블 설명
city	전 세계 도시 정보
country	전 세계 국가 정보
countrylanguage	각 국가에서 사용하는 언어 정보

ERD로 테이블 간 연결 관계 파악

- ERD(Entity Relationship Diagram)

- 테이블 간의 연결 관계를 도식화한 것으로 테이블 간의 관계를 다이어그램으로 그려 놓은 것



world 데이터베이스 ERD

country와 countrylanguage 테이블의 관계

실선은 두 테이블의 조인 칼럼인 Code 칼럼과 CountryCode 칼럼이 모두 기본 키 칼럼이라는 뜻
삼지창 모양은 country와 countrylanguage 테이블이 **일대다** 관계임을 뜻

country와 city 테이블 관계

점선은 city 테이블의 CountryCode 칼럼이 기본 키 칼럼이 아님

외래 키

countrylanguage 테이블의 CountryCode 칼럼
city 테이블 역시 CountryCode 칼럼

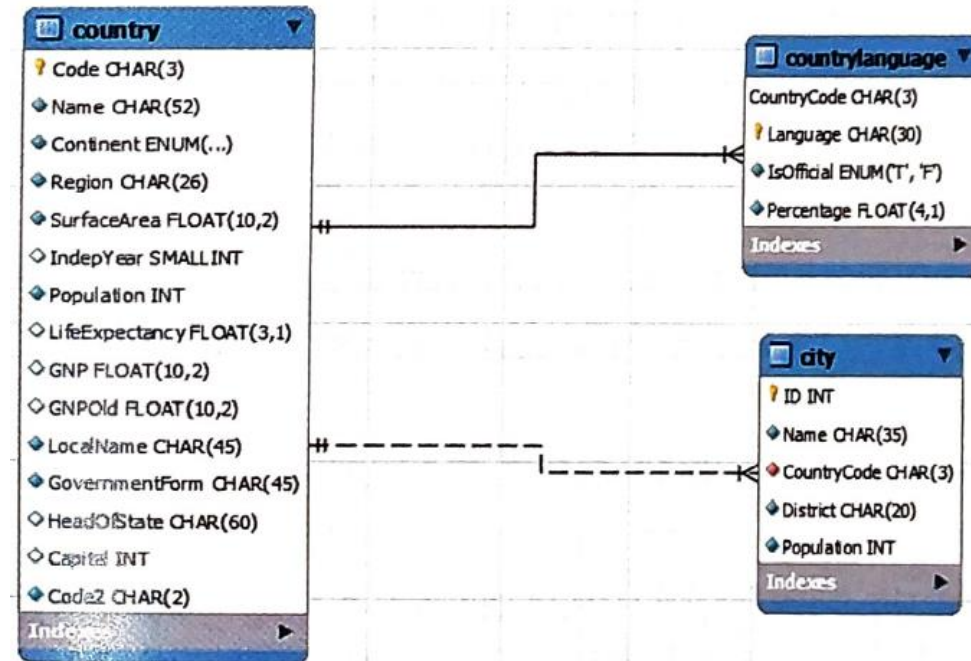
☞ 관계가 설정되며 **제약이 발생**

countrylanguage 테이블에 데이터를 넣을때, CountryCode 칼럼에는 반드시 country 테이블의 Code 칼럼에 있는 값만 넣을 수 있음

- 내부조인

```
SELECT  
FROM 테이블1 [AS] 별칭1  
[INNER] JOIN 테이블2 [AS] 별칭2  
ON 별칭1.칼럼1 = 별칭2. 칼럼2  
AND ...  
WHERE ... ;
```

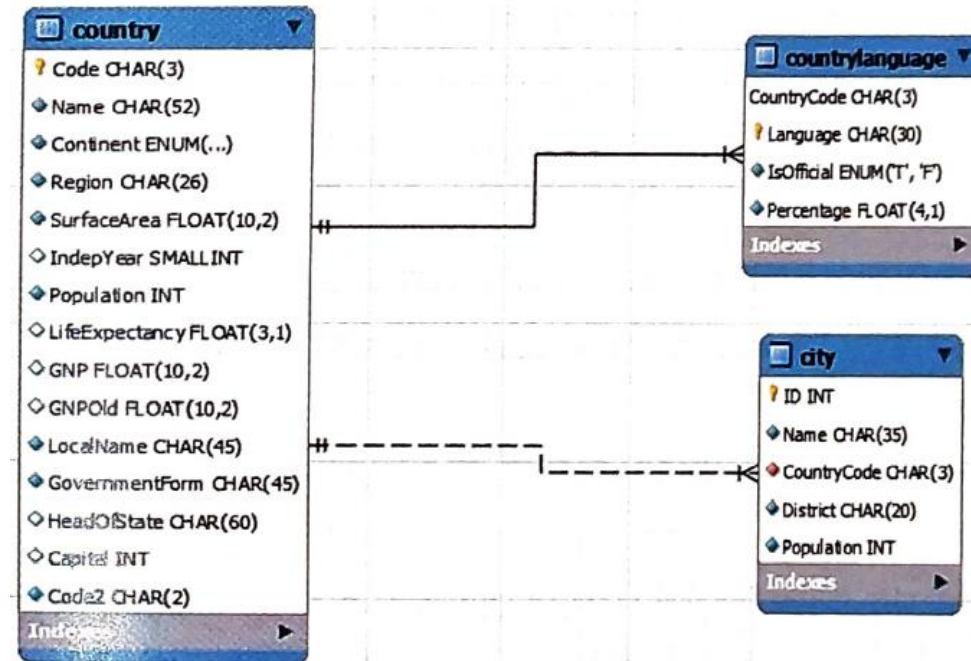
```
SELECT a.id, a.name, a.countrycode, b.code, b.country_name, a.district, a.population  
FROM city a  
INNER JOIN country b  
ON a.countrycode = b. code  
ORDER BY 1;
```



- 내부조인

```
SELECT  
FROM 테이블1 [AS] 별칭1  
[INNER] JOIN 테이블2 [AS] 별칭2  
ON 별칭1.칼럼1 = 별칭2. 칼럼2  
AND ...  
WHERE ... ;
```

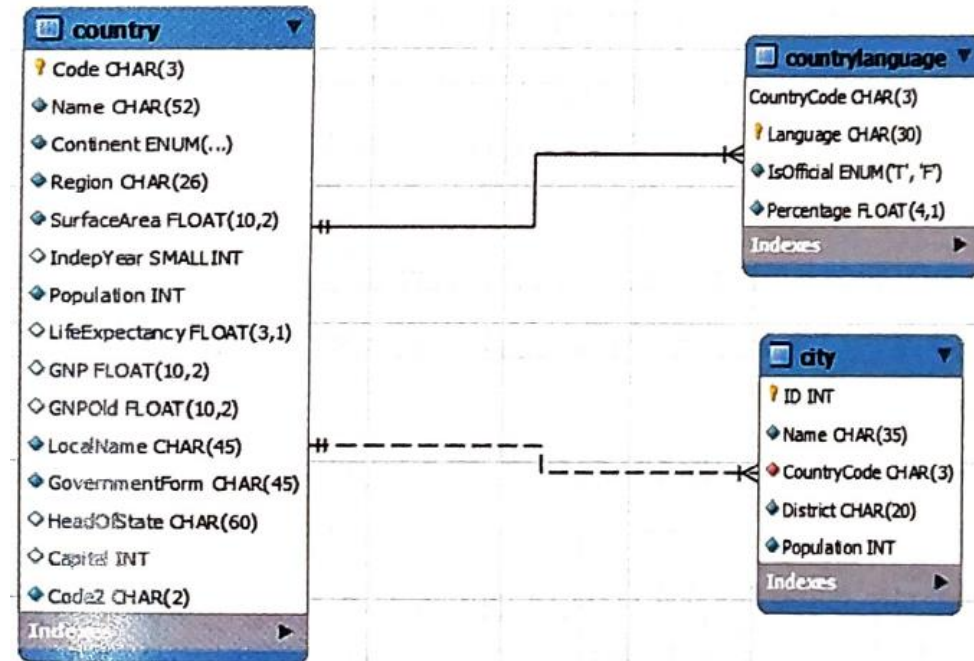
```
SELECT a.id, a.name, a.countrycode, b.code, b.country_name, a.district, a.population  
FROM city a  
INNER JOIN country b  
ON a.countrycode = b. code  
ORDER BY 1;
```



- 내부조인

```
SELECT  
FROM 테이블1 [AS] 별칭1  
[INNER] JOIN 테이블2 [AS] 별칭2  
ON 별칭1.칼럼1 = 별칭2. 칼럼2  
AND ...  
WHERE ... ;
```

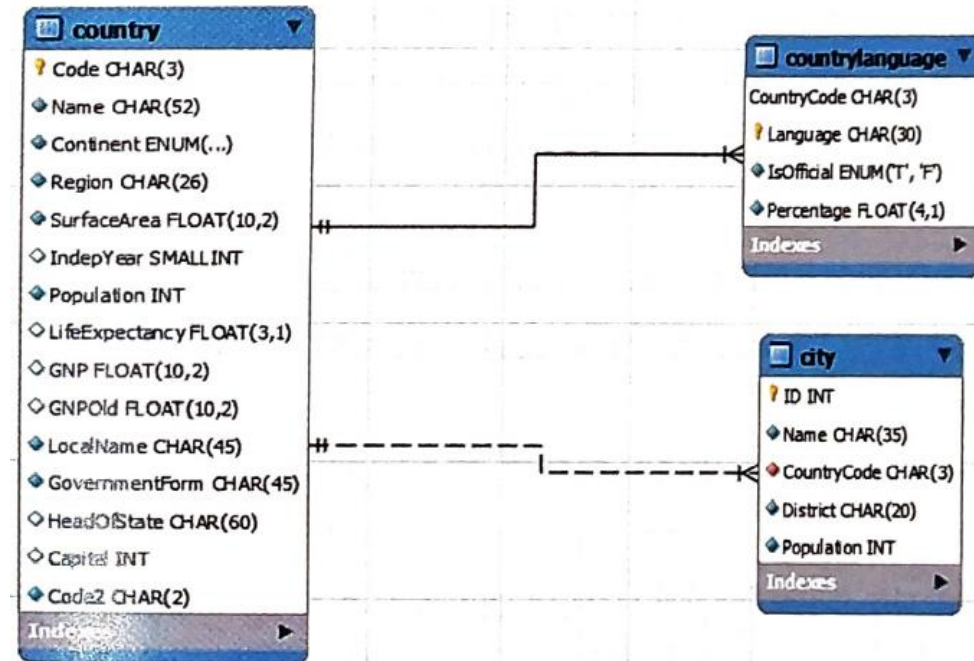
```
SELECT b.name country_name, a.language, a.isofficial, a.percentage  
FROM countrylanguage a  
INNER JOIN country b  
ON a.countrycode = b.code  
ORDER BY 1;
```



- 내부조인

```
SELECT  
FROM 테이블1 [AS] 별칭1  
[INNER] JOIN 테이블2 [AS] 별칭2  
ON 별칭1.칼럼1 = 별칭2. 칼럼2  
AND ...  
WHERE ... ;
```

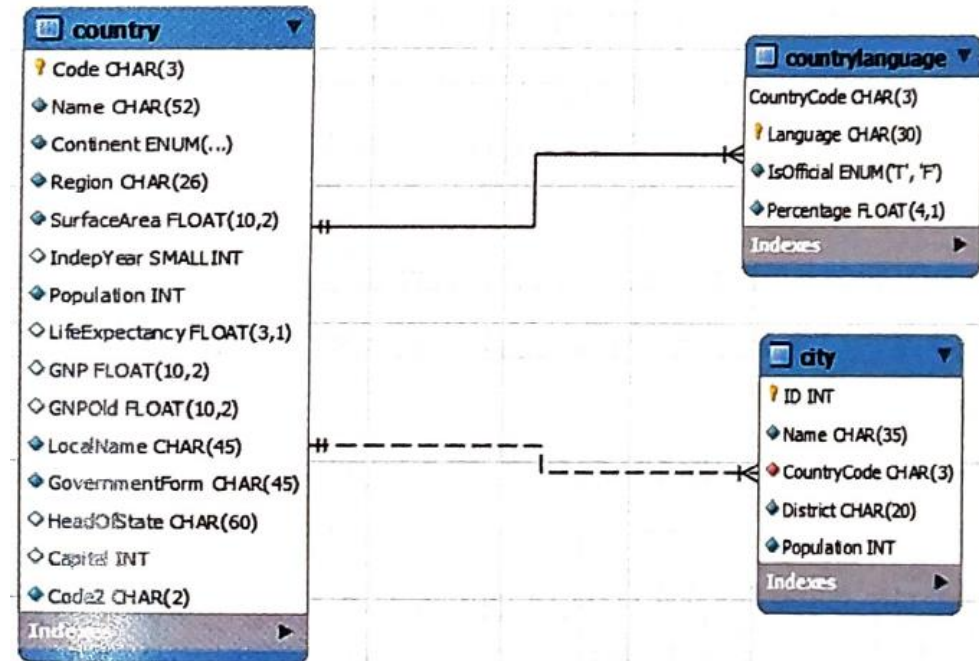
```
SELECT b.name country_name, a.language, a.isofficial, a.percentage  
FROM countrylanguage a  
INNER JOIN country b  
ON a.countrycode = b.code  
WHERE a.countrycode = 'KOR'  
ORDER BY 1;
```



- 내부조인

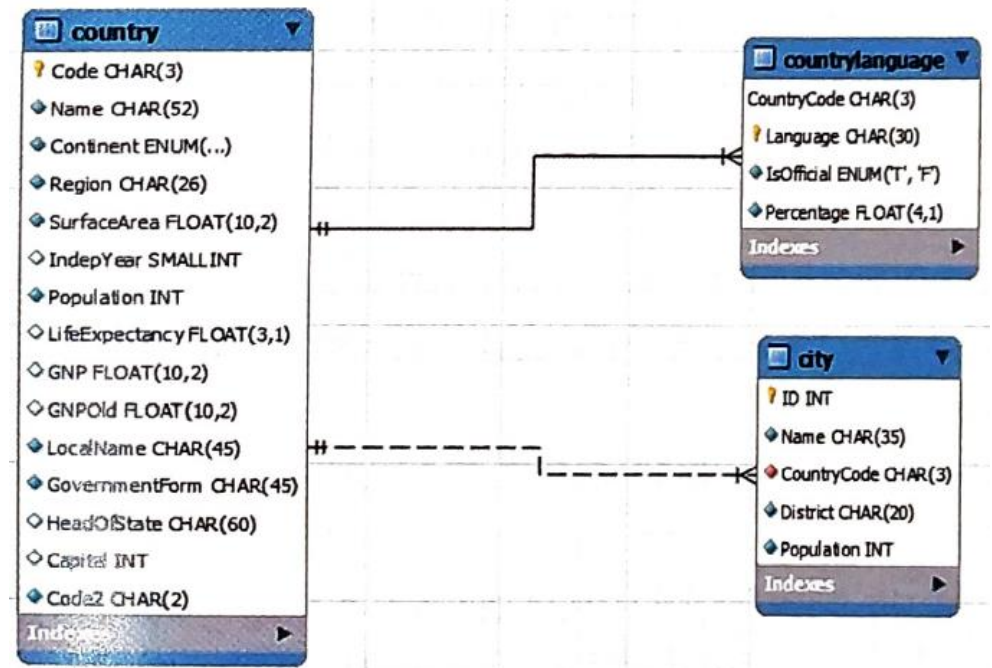
```
SELECT  
FROM 테이블1 [AS] 별칭1  
[INNER] JOIN 테이블2 [AS] 별칭2  
ON 별칭1.칼럼1 = 별칭2. 칼럼2  
AND ...  
WHERE ... ;
```

```
SELECT a.code, a.name, a.continent, a.region, a.population, b.language,  
c.name, c.district, c.population  
FROM country a  
INNER JOIN countrylanguage b  
ON a.code = b.countrycode  
INNER JOIN city c  
ON a.code = c.countrycode  
WHERE a.code = 'KOR'  
ORDER BY 1;
```



오류의 이유?

```
SELECT a.code, a.name, a.continent, a.region,  
a.population, b.language,  
name, c.district, c.population  
FROM country a  
INNER JOIN countrylanguage b  
ON a.code = b.countrycode  
INNER JOIN city c  
ON a.code = c.countrycode  
WHERE a.code = 'KOR'  
ORDER BY 1;
```



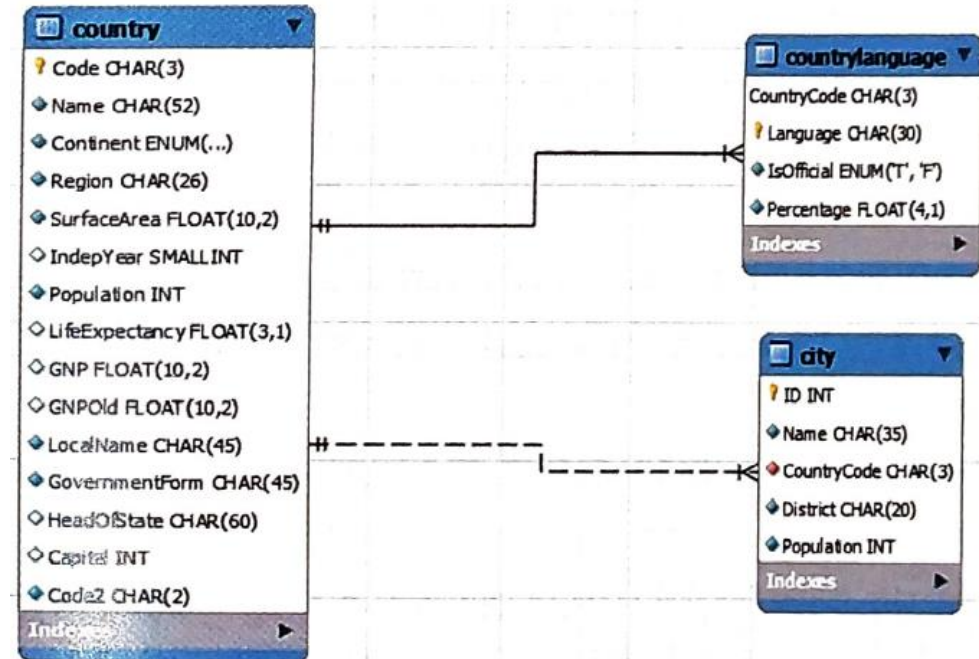
■ 내부조인

```
SELECT  
FROM 테이블1 [AS] 별칭1, 테이블2 [AS] 별칭2  
WHERE 별칭1.칼럼1 = 별칭2.칼럼2  
AND ;
```

- 조인에 참여할 테이블을 FROM 절에 모두 명시
- 조인 조건은 다른 조건처럼 WHERE 절에 기술

```
SELECT b.name country_name, a.language, a.isofficial, a.percentage  
FROM country language a, country b  
WHERE a.countrycode = b.code  
AND a.countrycode = 'KOR'  
ORDER BY 1;
```

```
SELECT b.name country_name, a.language, a.isofficial, a.percentage  
FROM countrylanguage a  
INNER JOIN country b  
ON a.countrycode = b.code  
WHERE a.countrycode = 'KOR'  
ORDER BY 1;
```

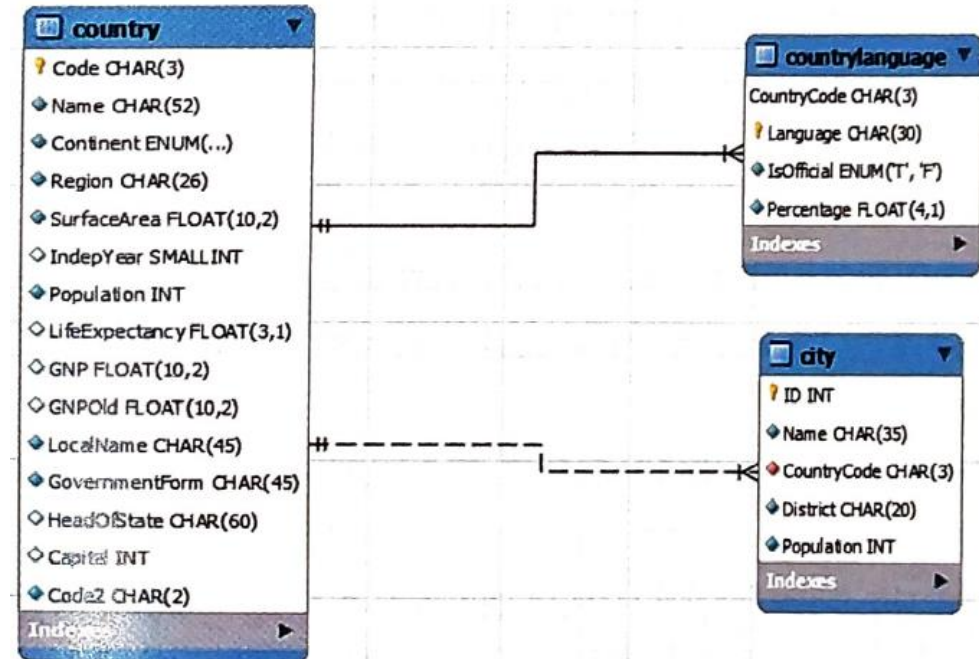


■ 내부조인

```
SELECT  
FROM 테이블1 [AS] 별칭1, 테이블2 [AS] 별칭2  
WHERE 별칭1.칼럼1 = 별칭2.칼럼2  
AND ;
```

```
SELECT a.code, a.name, a.continent, a.region, a.population, b.language, c.name, c.district, c.population  
FROM country a, country language b, city c  
WHERE a.code = b.countrycode  
AND a.code = c.countrycode  
AND a.code = 'KOR'  
ORDER BY 1;
```

```
SELECT a.code, a.name, a.continent, a.region, a.population, b.language,  
c.name, c.district, c.population  
FROM country a  
INNER JOIN countrylanguage b  
ON a.code = b.countrycode  
INNER JOIN city c  
ON a.code = c.countrycode  
WHERE a.code = 'KOR'  
ORDER BY 1;
```



- Quiz

world 데이터베이스에서 country와 city 테이블을 내부 조인해 국가명과 해당 국가에 속한 도시 개수를 구하고, 마지막에는 전체 도시 수를 구하는 쿼리를 작성하세요.

- Quiz

world 데이터베이스의 country 테이블에서 대륙별로 몇 개 국가가 있는지를 구하는 쿼리를 작성하시오.

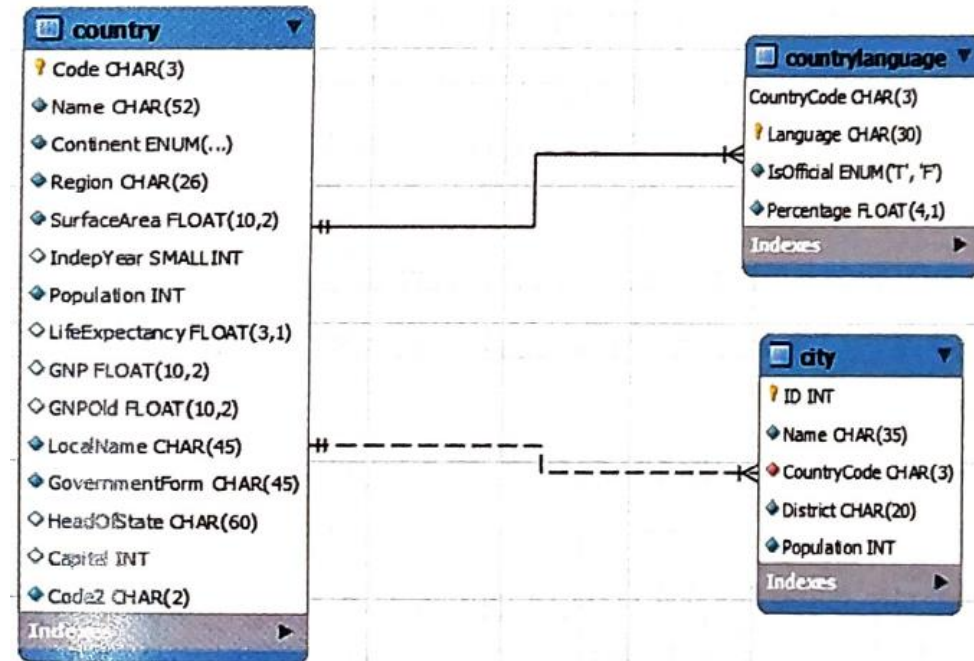
```
SELECT a.continent, ?  
FROM country  
GROUP BY ? ;
```

위의 쿼리에서 city 테이블과의 내부 조인해서 결과를 조회하는 쿼리를 작성하시오

- 외부조인

```
SELECT  
FROM 테이블1 [AS] 별칭1  
LEFT [OUTER] JOIN 테이블2 [AS] 별칭2  
ON 별칭1. 칼럼1 = 별칭2. 칼럼2  
AND ...  
WHERE ...;
```

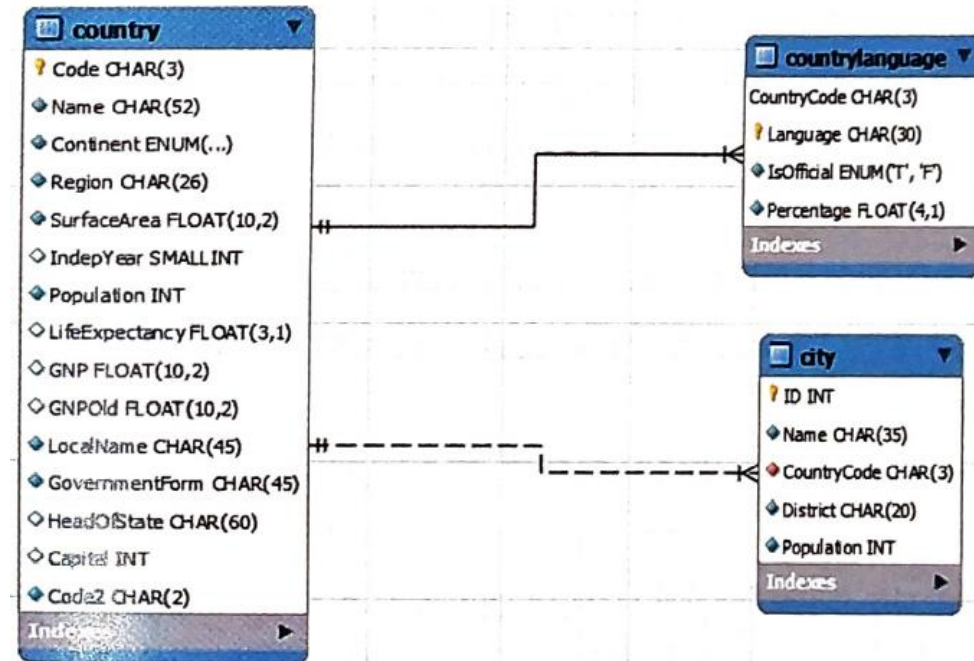
```
SELECT a.continent, COUNT(*) AS 전체건수  
FROM country a  
LEFT OUTER JOIN city b  
ON a.code = b.countrycode  
GROUP BY a.continent;
```



- 외부조인

```
SELECT  
FROM 테이블1 [AS] 별칭1  
LEFT [OUTER] JOIN 테이블2 [AS] 별칭2  
ON 별칭1. 칼럼1 = 별칭2. 칼럼2  
AND ...  
WHERE ...;
```

```
SELECT a.continent, COUNT(*) 전체건수, COUNT(b.name) 도시건수  
FROM country a  
LEFT OUTER JOIN city b  
ON a.code = b.countrycode  
GROUP BY a.continent;
```



- 외부조인

```
SELECT  
FROM 테이블1 [AS] 별칭1  
LEFT [OUTER] JOIN 테이블2 [AS] 별칭2  
ON 별칭1. 칼럼1 = 별칭2. 칼럼2  
AND ...  
WHERE ...;
```

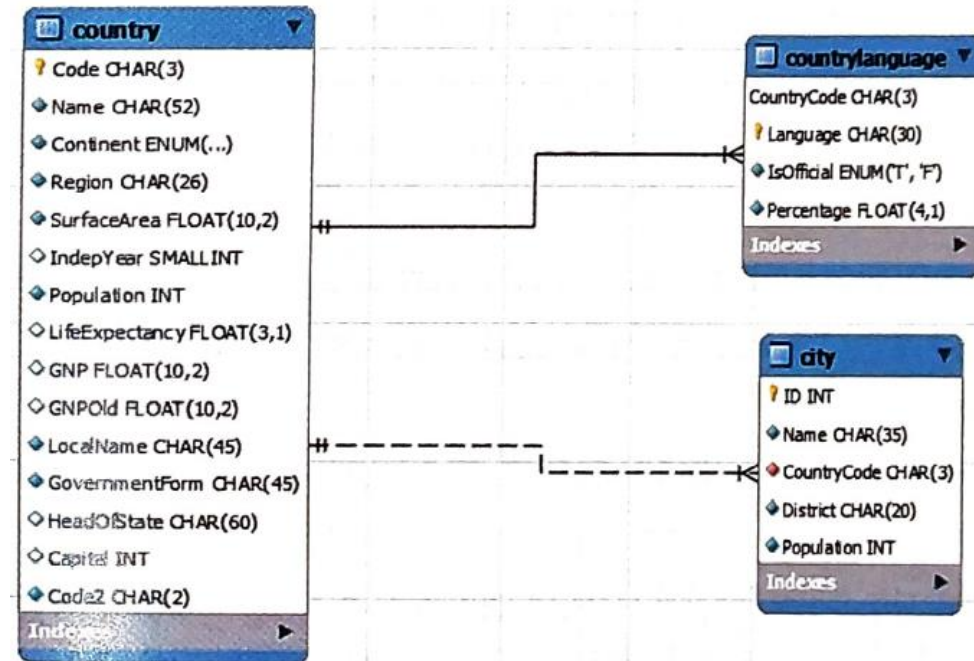
SELECT a.continent, COUNT(*) 전체건수, COUNT(b.name) 도시건수

FROM country a

LEFT OUTER JOIN city b

ON a.code = b.countrycode

GROUP BY a.continent;



- Quiz

아프리카(Africa) 대륙에 속한 국가 중 사용 언어가 없는 국가가 있습니다. Country와 countrylanguage 테이블을 외부 조인해서 이 국가의 이름이 무엇인지 찾는 쿼리를 작성하세요

```
SELECT  
FROM  
LEFT JOIN  
WHERE  
GROUP  
HAVING
```