

학습목표

1. JOIN의 개념, 종류
2. 별명(ALIAS)
3. 뷰(View)

학습내용

- JOIN(INNER/LEFT/RIGHT/FULL)을 이해할 수 있습니다.
- 별명(ALIAS)의 개념을 이해할 수 있습니다.
- 뷰(VIEW)의 사용법을 이해할 수 있습니다.

사전퀴즈

1. 따로 지정하지 않고 JOIN이라고 하면 INNER JOIN을 말한다.

정답 : O

해설 : 조인이라고 하면 INNER JOIN을 의미한다.

2. 별명(ALIAS)은 컬럼명을 다른 이름으로 변경하고 싶을 때 사용한다.

정답 : O

해설 : 테이블의 컬럼명을 쿼리결과에 노출하고 싶지 않거나 알기 쉽게 포기하고 싶을 때 ALIAS를 사용한다.

수업

1. JOIN의 개념, 종류

JOIN의 개념

- 서로 다른 테이블을 공통 컬럼을 기준으로 합치는(결합하는) 테이블단위연산 → 주로 외래키를 많이 사용
- 조인의 결과 테이블은 이전 테이블의 **컬럼 수의 합**과 같다.
(즉, 첫 번째 테이블 컬럼 수: 5개, 두 번째 테이블 컬럼 수: 5개 → 조인결과 테이블의 컬럼수: 10개)
- select *
- from 테이블1 join 테이블2
- on 테이블1.컬럼명 = 테이블2.컬럼명 ...
- 조인시 서로 다른 테이블에 같은 컬럼명이 존재하면 구분을 위해 테이블명, 컬럼명으로 사용해서 표시
- 조인시 NULL 값을 허용하지 않는 내부조인과 허용하는 외부조인으로 구분
- INNER JOIN
- OUTER JOIN (LEFT JOIN / RIGHT JOIN / FULL JOIN)

INNER JOIN	조인 시 NULL값을 허용하지 않음 (NULL값을 가진 레코드는 조인결과에 빠짐)
LEFT JOIN	조인 수행시 먼저 표기된 좌측 테이블에 해당하는 데이터를 먼저 읽은 후, 나중 표기된 우측 테이블에서 JOIN 대상 데이터를 읽어옴 즉, Table A와 B가 있을 때 (Table 'A'가 기준이 됨), A와 B를 비교해서 B의 JOIN 컬럼에서 같은 값이 있을 때 그 해당 데이터를 가져오고, B의 JOIN 컬럼에서 같은 값이 없는 경우에는 B 테이블에서 가져오는 컬럼들은 NULL 값으로 채움
RIGHT JOIN	조인 수행시 LEFT JOIN과 반대로 우측 테이블이 기준이 되어 결과를 생성한다. 즉, TABLE A와 B가 있을 때 (TABLE 'B'가 기준이 됨), A와 B를 비교해서 A의 JOIN 컬럼에서 같은 값이 있을 때 그 해당 데이터를 가져오고, A의 JOIN 컬럼에서 같은 값이 없는 경우에는 A 테이블에서 가져오는 컬럼들은 NULL 값으로 채움
FULL JOIN	조인 수행시, 좌측, 우측 테이블의 모든 데이터를 읽어 JOIN하여 결과를 생성한다. 즉, TABLE A와 B가 있을 때 (TABLE 'A', 'B' 모두 기준이 됨), RIGHT OUTER JOIN과 LEFT OUTER JOIN의 결과를 합집합으로 처리한 결과와 동일함. 단, UNION ALL이 아닌 UNION 기능과 같으므로 중복되는 데이터는 삭제함

예제 1-1

city 테이블과 country 테이블을 조인하시오

(city.CountryCode = country.code)

```

1 select *
2 from city join country
3 on city.CountryCode = country.Code
4 ;
5

```

결과 #1 (20×4,079)

ID	Name	CountryCode	District	Population	Code	Name	Continent	Region	SurfaceArea	IndepYear	Population
129	Oranjestad	ABW	-	29,034	ABW	Aruba	North America	Caribbean	193.00	(NULL)	103,000
1	Kabul	AFG	Kabul	1,780,000	AFG	Afghanistan	Asia	Southern and Central Asia	652,090.00	1,919	22,720,000
2	Qandahar	AFG	Qandahar	237,500	AFG	Afghanistan	Asia	Southern and Central Asia	652,090.00	1,919	22,720,000
3	Herat	AFG	Herat	186,800	AFG	Afghanistan	Asia	Southern and Central Asia	652,090.00	1,919	22,720,000
4	Mazar-e-Sharif	AFG	Balkh	127,800	AFG	Afghanistan	Asia	Southern and Central Asia	652,090.00	1,919	22,720,000
56	Luanda	AGO	Luanda	2,022,000	AGO	Angola	Africa	Central Africa	1,246,700.00	1,975	12,878,000

예제 1-2

국가코드와 해당 나라의 GDP를 표시하시오.

```

14 select city.CountryCode, country.GNP
15 from city join country
16 on city.CountryCode = country.Code
17 ;

```

결과 #1 (2×4,079)

CountryCode	GNP
ABW	828.00
AFG	5,976.00
AFG	5,976.00
AFG	5,976.00
AFG	5,976.00
AGO	6,648.00
AGO	6,648.00
AGO	6,648.00
AGO	6,648.00

예제 1-3

city 테이블에 국가코드가 없는 도시가 있는지 확인하시오.

주의 - 컬럼값이 null 인 데이터 확인

CountryCode = null (X)

CountryCode is NULL (O)

```

1 -- city 테이블에 국가코드가 없는 경우
2 select count(*)
3 from city
4 where CountryCode is NULL
5 ;
6

```

결과 #1 (1×1)

count(*)
0

country 테이블에는 존재하지만 도시가 하나도 없는 나라가 있는지 확인하시오.

```
26 -- 나라는 있지만 도시가 없는 경우
27 select count(*)
28 from city right join country
29 on city.CountryCode = country.Code
30 where city.name is NULL
31 ;
32
```

결과 #1 (1x1)

count(*)
7

이 때 INNER JOIN / LEFT JOIN / RIGHT JOIN의 차이점을 확인하시오.

=> INNER JOIN 개수와 LEFT JOIN 개수가 같다는 말은 "도시는 있지만 나라가 없는 경우는 없다"는 것을 의미

=> LEFT JOIN 개수와 RIGHT JOIN 개수가 다르다는 말은 "나라는 있지만 도시가 없는 경우가 있다"는 것을 의미

```
7 select count(*)
8 from city inner join country
9 on city.CountryCode = country.Code
10 ;
11
```

결과 #1 (1x1)

count(*)
4,079

```
1 -- 도시는 있지만 나라가 없는 경우 포함 - 4,079개
2 select count(*)
3 from city left join country
4 on city.CountryCode = country.Code
5 ;
6
7 -- 도시는 있지만 나라가 없는 경우는 없음
8 select *
9 from city left join country
10 on city.CountryCode = country.Code
11 where country.Code is NULL
12 ;
13
20 -- 나라는 있지만 도시가 없는 경우 포함 - 4,086개
21 select count(*)
22 from city right join country
23 on city.CountryCode = country.Code
24 ;
25
26 -- 나라는 있지만 도시가 없는 경우 - 7개
27 select count(*)
28 from city right join country
29 on city.CountryCode = country.Code
30 where city.name is NULL
31 ;
```

2. 별명(ALIAS)

- SQL 쿼리 결과생성시 컬럼명에 대한 별명을 사용해 표시하는 기능
- SELECT 테이블명1.컬럼명1 AS 별명1, 테이블명2.컬럼명2 AS 별명2 FROM ...
- 조인할 때 많이 사용됨
- 테이블의 컬럼명을 쿼리결과에 노출하고 싶지 않거나 알기 쉽게 포기하고 싶을 때 사용

예제 2-1

city 테이블과 country 테이블을 조인해서 국가코드가 'KOR'인 나라의 축약표시명(Abbr)과 정식명(Full Name)을 표시하시오.

```
33 select city.CountryCode as Abbr, country.Name as FullName
34 from city join country
35 on city.CountryCode = country.code
36 where city.CountryCode = 'KOR'
37 ;
```

결과 #1 (2x70)	
Abbr	FullName
KOR	South Korea
KOR	South Korea
KOR	South Korea
KOR	South Korea
KOR	South Korea
KOR	South Korea
KOR	South Korea
KOR	South Korea
KOR	South Korea
KOR	South Korea
KOR	South Korea
KOR	South Korea
KOR	South Korea
KOR	South Korea
KOR	South Korea
KOR	South Korea
KOR	South Korea
KOR	South Korea
KOR	South Korea
KOR	South Korea

3. 뷰 (View)

- SQL 쿼리의 결과값을 임시테이블로 저장해서 사용할 수 있음
- 사용용도가 끝나면 명시적으로 삭제해야 함 (**DROP VIEW ...**)
- **CREATE VIEW 뷰명 AS SELECT ...**

예제 3-1

국가코드가 "KOR"인 도시의 국가코드와 국가명(Name)을 뷰로 생성하시오.

(이 때, 국가코드를 Abbr, 국가명을 FullName으로 표시하시오)

* 뷰 생성 후, 뷰를 테이블처럼 사용할 수 있는 것을 확인할 수 있음

```
39 create view sampleView as select city.CountryCode as Abbr, country.Name as FullName
40 from city join country
41 on city.CountryCode = country.code
42 where city.CountryCode = 'KOR'
43 ;
44
45 select *
46 from sampleView
47 ;
```

조인의 네 가지 경우와 차이점?

전문가 의견

조인은 크게 내부조인(INNER JOIN)과 외부조인(OUTER)으로 나뉩니다.

그리고 외부조인에는 LEFT/RIGHT/FULL JOIN이 있습니다. 기본 JOIN 값은 내부 조인입니다.

조인이란 여러 개의 테이블을 공통컬럼(주로 외래키)를 기준으로 테이블들을 합쳐서 큰 테이블을 만드는 것입니다.

이 과정에서 기본 테이블의 공통컬럼기준으로 기준컬럼 값에 해당하는 레코드가 다른 테이블에 있을 경우와 없을 경우 처리방법이 달라집니다.

내부조인은 양쪽에 다 있는 레코드만 표시합니다. 예를들어, world DB의 city와 country 테이블을 조인할 경우에 도시의 국가코드와 대응하는 국가코드가 존재하는 국가만 표시되도록 합니다. 만약 국가코드는 있으나 해당 국가코드명을 가진 도시가 없다면 국가를 표시하지 않습니다. 그리고 도시명에 국가코드가 없는 도시도 표시하지 않습니다.

외부조인은 위의 두 가지 경우에 표시를 선택적으로 하게 됩니다. 예를 들어 city 테이블을 기반으로 country 테이블을 조인하는 경우 도시레코드에 국가코드가 없는 레코드도 표시하는 것이 LEFT 조인이고, 도시가 없는 국가의 레코드는 표시하도록 하는 것이 RIGHT 조인입니다. 두 가지 경우 모두 표시하는 것이 FULL 조인입니다.