

## 1. 조인(Join)

어떤 값의 공통된 집합을 공유하는 둘 이상의 테이블로부터 행을 검색하는 것으로, 관계형 데이터베이스에서 매우 중요한 연산이다. 조인을 하기 위해서는 SELECT문의

- FROM절에 적어도 두 개 이상의 테이블이 있고
- 하나 이상의 조인 조건이 있을 때

조인이라 부른다.

SELECT문의 FROM절에 테이블이나 뷰 등이 두 개 이상 기술되고, 조인조건이나 출력할 칼럼 등을 기술하는 WHERE절이나 SELECT절 등에서 각 테이블에 사용된 동일한 칼럼명이 구분되어야 한다.

문 법	SELECT	테이블1.칼럼명1, 칼럼명2, 리터럴, 함수, 수식, ....
	FROM	테이블명1, 테이블명2, 뷰명1, ....
	WHERE	테이블명1.칼럼명 연산자 테이블명2.칼럼명 ....
	GROUP BY	테이블1.칼럼명1, 칼럼명2, ....
	HAVING	검색조건2
	ORDER BY	테이블명1.칼럼명1, ....

- 칼럼을 구분하기 위하여 "테이블명.칼럼명"으로 기술하며, 테이블명에 별명을 사용하면 단순화할 수 있다. 테이블명에 대한 별명은 FROM절에서 "테이블명 한 칸 이상의 공백을 띄운 후 별명"을 기술한다.
- 각 테이블의 공통된 행일 때 상호 관련 부속 질의어는 서브 쿼리에 사용하는 테이블의 칼럼을 출력하지 못하나 조인은 FROM절에 기술한 테이블의 칼럼들을 언제든지 출력할 수가 있다.

### 1.1.조인의 종류

카티션 프로덕트(Cartesian Product), 등가조인(equijoin) 또는 내부조인(Inner Join), 외부조인(outer join), 자기조인(self join), 비등가조인(non-equi join), 안티조인(anti join), 세미조인(semi join)으로 구분한다.

### 1.2.조인에서 발생할 수 있는 문제

SELECT문의 FROM절에 두 개 이상의 테이블이나 뷰 등을 기술하고, WHERE절에 조인조건이 기술되더라도 테이블명에 동일한 칼럼명이 존재할 경우 "테이블명.칼럼명"으로 구분해야 한다. 테이블

명을 생략하여 칼럼명을 기술한 경우 칼럼을 구분할 수 없기 때문에 "ORA-00918: 열의 정의가 애매합니다."라는 오류 메시지가 출력되고 실행되지 않는다.

**예) 조인 시 각 테이블의 동일한 칼럼 구분이 명확하지 않을 경우**

## 2. 카티션 프로덕트

크로스 조인(cross join) 또는 교차조인이라고도 한다. SELECT문의 WHERE절에 조인조건이 생략 또는 누락한 것이다. 카티션 프로덕트란 SELECT문의 FROM절에 기술한 테이블들의 행의 수를 곱한 것(테이블1의 행의 수 \* 테이블2의 행의 수 \* ...)으로 행이 반환된다.

문 법	SELECT	테이블명.칼럼명, 칼럼명, ...
	FROM	테이블명1 CROSS JOIN 테이블명2
	...	

**예) 1. Professor 테이블과 Course 테이블을 이용하여 교수기 담당하고 있는 과목들을 출력하시오.**

**2. Professor 테이블과 Course 테이블을 이용하여 교수가 담당하고 있는 과목들을 크로스 조인하여 출력하시오.**

## 3. 등가조인 또는 내부조인

두 개 이상의 테이블로부터 행을 검색할 때, 한 테이블에 있는 하나 이상의 칼럼 값이 다른 테이블에 있는 하나 이상의 칼럼 값과 같은 때 참이 되고, 참이 될 때 행의 칼럼 값을 조합하여 출력하는 조인을 말한다.

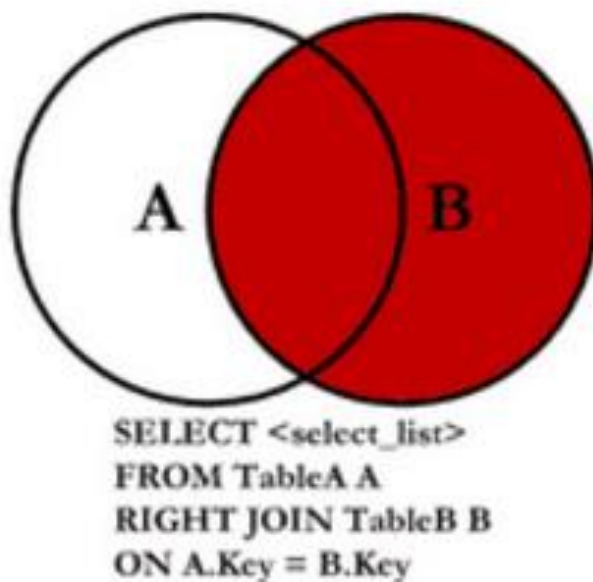
등가조인을 내부조인(inner join) 또는 단순조인(simple join)이라고도 불리며,

- WHERE절을 이용하는 방법
- NATURAL INNER JOIN 키워드를 이용한 자연조인
- INNER JOIN~USING 키워드를 이용한 내부조인
- INNER JOIN~ON 키워드를 이용한 내부조인의 다양한 방법이 있다.

### 3.1. WHERE절을 이용한 등가조인

등가조인은 SELECT문의 WHERE절에서 조인조건으로 "=" 관계연산자를 사용한다.

조인조건	테이블명1.칼럼명 = 테이블명2. 칼럼명
------	------------------------



예) 1. Professor 테이블과 Course 테이블을 이용하여 교수가 최소한 한 과목 이상을 담당하고 있는 교수의 교수번호, 교수명, 직위, 과목명, 학점수를 교수번호순으로 출력하시오.

2. SG\_Scores 테이블과 Student 테이블, Course 테이블을 이용하여 'C0901' 학번의 학년과 성명, 학점을 취득한 과목의 과목명과 학점수, 성적을 출력하시오.

### 3.2. NATURAL [INNER] JOIN 키워드의 자연조인

자연조인은 WHERE절을 생략하고 FROM절에 NATURAL [INNER] JOIN 키워드를 사용한다. FROM절에 NATURAL INNER JOIN 키워드를 사용하면 자동으로 테이블 간의 모든 칼럼을 대상으로 공통 칼럼을 조사하여 내부적으로 조인 명령문을 생성한다. 단, SELECT절에 공통 칼럼은 테이블의 별명을 붙이지 않으며, NATURAL JOIN 키워드에 사용된 칼럼은 식별자를 가질 수 없다.

문 법	SELECT 칼럼명1, 칼럼명2, ...  FROM 테이블명1 NATURAL [INNER] JOIN 테이블명2  ...
--------	--

예) 1. Professor 테이블과 Course 테이블을 이용하여 교수가 최소한 한 과목 이상을 담당하고 있는 교수의 교수번호, 교수명, 직위, 과목명, 학점수를 교수번호 순으로 출력하시오. 단, NATURAL INNER Join 키워드를 사용하여 조인하시오. INNER는 생략 가능.

2. SG\_Scores 테이블과 Student 테이블, Course 테이블을 이용하여 'C0901' 학번의 학년과 성명, 학점을 취득한 과목의 과목명과 학점수, 성적을 출력하시오. 단, NATURAL [INNER] JOIN 키워드를 사용하여 조인하시오.

### 3.3. [INNER] JOIN~USING 키워드의 내부조인

NATURAL [INNER] JOIN 키워드는 모든 칼럼을 대상으로 공통 칼럼을 조사하기 때문에 실행 속도가 느려지는 문제점이 있다. 이를 해결하기 위한 것이 JOIN~USING 키워드이다.

[INNER] JOIN~USING 키워드는 두 테이블 간의 조인조건에 해당하는 "공통칼럼명"을 괄호( )속에 기술한다. INNER는 생략할 수 있으며, 복수 개의 공통칼럼명을 기술할 수도 있다.

문 법	SELECT 칼럼명1, 칼럼명2, ...  FROM 테이블명1 [INNER] JOIN 테이블명2 USING (공통칼럼명, ...)  ...
--------	---

예) 1번, 2번. NATURAL [INNER] JOIN으로 검색한 것을 다시 JOIN~USING 키워드를 이용하여 조인해보자.

### 3.4. [INNER] JOIN~ON 키워드의 내부조인

테이블 간의 조인조건을 기술한다. 조인조건은 "테이블명1.칼럼명 = 테이블명2.칼럼명"으로 기술하고, 모호한 칼럼은 "테이블명.칼럼명"으로 구분해야 한다. INNER는 생략할 수 있다.

문 법	SELECT 테이블명.칼럼명1, 칼럼명2, ...  FROM 테이블명1 [INNER] JOIN 테이블명2 ON (조인조건)  ...
--------	---

예) 1번, 2번. [INNER] JOIN ON 키워드를 이용하여 조인해보자.

## 4. 외부조인

테이블 간의 공통된 행과 공통되지 않은 테이블의 행들도 모두 출력한다. 외부조인은 출력 결과에 따라 좌 외부조인(left outer join), 우 외부조인(right outer join), 전체 외부조인(full outer join)으로 구분하고, 기술하는 방법은

☞ WHERE절의 등가조인 조건에 (+)를 기술한다.

- 좌 외부조인은 조인조건 오른쪽에 (+)를 기술한다.
- 우 외부조인은 조인조건 왼쪽에 (+)를 기술한다.

#### ☞ OUTER JOIN 키워드로 기술한다.

- 좌 외부조인은 LEFT [OUTER] JOIN 키워드로 사용한다.
- 우 외부조인은 RIGHT [OUTER] JOIN 키워드를 사용한다.
- 전체 외부조인은 FULL [OUTER] JOIN 키워드를 사용한다.

문	SELECT 칼럼명1, 칼럼명2, ...
법	FROM 테이블명1 NATURAL {LEFT   RIGHT   FULL} [OUTER] JOIN 테이블명2

문	SELECT 칼럼명1, 칼럼명2, ...
법	FROM 테이블명1 {LEFT   RIGHT   FULL} [OUTER] JOIN 테이블명2 USING (공통칼럼명, ...)

문	SELECT 테이블명.칼럼명1, 칼럼명2, ...
법	FROM 테이블명1 {LEFT   RIGHT   FULL} [OUTER] JOIN 테이블명2 ON 조인조건

#### 4.1. WHERE절에 좌 외부조인 기술

좌 외부조인(left outer join)은 등가조인의 WHERE절 조인조건에서 오른쪽 위치에 (+)를 기술하며, 공통된 행과 테이블1의 공통되지 않은 행들을 출력한다.

예) 1. Professor 테이블의 모든 교수에 대한 교수번호, 교수명, 직위를 출력하되, Course 테이블을 참조하여 교수가 개설하고 있는 과목에 대하여 과목명, 학점수를 출력하시오.

#### 4.2. WHERE절에 우 외부조인 기술

등가조인의 WHERE절 조인조건에서 왼쪽 위치에 (+)를 기술하며, 공통된 행과 테이블2의 공통되지 않은 행들을 출력한다.

예) 1. Course 테이블의 모든 개설과목에 대하여 과목명, 학점수를 출력하되, Professor 테이블을 참조하여 담당하고 있는 과목의 교수번호, 교수명, 직위도 출력하시오.

#### 4.3. LEFT JOIN 키워드를 사용한 좌 외부조인

좌 외부조인은 FROM절에 "테이블명1 LEFT [OUTER] JOIN 테이블명2 ON 조인조건" 키워드의 3가지 방법으로 기술할 수 있다. 조인 조건에 만족하는 공통된 행과 만족하지 않는 테이블2의 칼럼 값이 널(null)일 경우에도 출력한다. OUTER는 생략할 수 있다.

예) 위의 "WHERE절에 좌 외부조인 기술"에서 실습했던 예제를 LEFT OUTER JOIN 키워드로 좌 외부조인 하시오.

#### 4.4. RIGHT JOIN 키워드를 사용한 우 외부조인

우 외부조인은 FROM절에 "테이블명1 RIGHT [OUTER] JOIN 테이블명2 ON 조인조건" 키워드의 3가지 방법으로 기술할 수 있다. 조인 조건에 만족하는 공통된 행과 만족하지 않는 테이블1의 칼럼 값이 널(NULL)경우에도 출력한다. OUTER는 생략할 수 있다.

예) 위의 예제 중 "WHERE절에 우 외부조인 기술" 관련 예제를 RIGHT OUTER JOIN 키워드로 우 외부조인 하시오.

#### 4.5. FULL JOIN 키워드를 사용한 전체 외부조인

전체 외부조인은 FROM절에 "테이블명1 FULL [OUTER] JOIN 테이블명2 ON 조인조건" 키워드로 기술하며, 조인조건에 만족하는 공통된 행과 조인조건에 만족하지 않은 테이블1과 테이블2의 칼럼 값이 널(NULL) 경우에도 출력한다. 전체 외부조인은 (+) 기호로 기술할 수 없으며, OUTER는 생략할 수 있다.

예) Professor 테이블의 모든 교수에 대한 교수번호, 교수명, 직위를 출력하고, Course 테이블의 교수가 개설하고 있는 모든 과목에 대하여 과목명, 학점수를 출력하시오.

### 5. 자기 조인

자기조인(self join)이란 순환적 관계 모델링에 의해 표현되는 테이블, 즉 테이블이 자신의 기본 키를 참조하는 외부 키를 갖고 있을 때 사용된다. 자기조인은 FROM절에 동일한 테이블을 두 번 기술하고 별명으로 구분하여 조인한다.

☞ 자기조인은 세 가지 기술 방법이 있다.

- FROM절에 "테이블명 별명1, 테이블명 별명2" WHERE절에 "조인조건" 기술한다.
- FROM절에 "테이블명 별명1 JOIN 테이블명 별명2 ON 조인조건" 기술한다.
- FROM절에 "테이블명 별명1 INNER JOIN 테이블명 별명2 ON 조인조건" 기술한다.

예) Department 테이블에 "대학" 학과코드와 Professor 테이블에 "P00" 총장 데이터를 추가하시오.

### 5.1. WHERE절을 이용한 자기조인

자기조인은 FROM절에 동일 Professor 테이블에 대하여 FROM절에 별명 T1은 교수의 정보로, 별명 T2는 관리자의 정보로 지정한다. WHERE절에 "별명1.칼럼명 = 별명2.칼럼명"으로 조인조건을 기술한다.

예) Professor 테이블을 이용하여 각 교수의 관리자명을 출력하시오.

### 5.2. INNER JOIN 키워드를 이용한 자기조인

INNER JOIN 키워드를 이용하여 FROM절에 "테이블명 별명1 JOIN 테이블명 별명2 ON 조인조건"을 기술하고, 조인조건은 "별명1.칼럼명 = 별명2.칼럼명"으로 기술한다.

예) 위의 "WHERE절을 이용한 자기조인" 예제를 INNER JOIN 키워드로 자기조인하시오.

## 6. 기타조인

기타 조인의 종류로 비등가조인(NON-EQUI JOIN), 세미조인(SEMI JOIN), 안티조인(ANTI JOIN)이 있다.

### 6.1. 비등가조인

조인조건에서 '='조건이 아닌 '<' 관계연산자나 BETWEEN~AND~연산자와 같은 연산자를 사용하는 조인 방법을 말하며, 대부분의 조인은 외부 키와 외부 키로 연결되어 등가조인으로 처리된다.

예) SG\_Scores 테이블로부터 'C0902'학번의 성적을 이용하여 Score\_Grade 테이블의 등급을 출력하시오.

## 6.2. 서브 쿼리를 이용한 조인

서브 쿼리를 이용한 조인 방법으로 세미조인과 안티조인이 있다.

- 세미조인(semi join)은 EXIST와 IN 연산자의 효율적 처리를 위한 조인이다.
- 안티조인(anti join)이란 메인 쿼리의 칼럼 값이 서브 쿼리의 반환 값을 만족하지 않는 행들을 반환하며, NOT IN 연산자를 사용한 조인을 말한다.

**예) Course 테이블의 교수번호가 Professor 테이블의 교수번호와 일치하지 않는 모든 행에 대하여 출력하시오.**

## 6.3. 기타 조인 응용

외부조인에서 조인되지 않은 결과를 출력할 수 있다. 하나의 테이블에서 다른 테이블의 값을 뺀 행들에 대하여 출력하는 것과 같다. 조인조건에서 "테이블명.칼럼명 IS NULL"을 추가한다.

**예) Professor 테이블과 Course 테이블을 참조하여 과목을 개설하지 않은 교수를 출력하시오.**

## 7. 집합연산자

각 쿼리 결과 행에 대한 집합 연산을 할 수 있다. 집합연산자는 UNION, UNION ALL, INTERSECT, MINUS가 있다.

### 7.1. 집합 연산자 종류

UNION 연산자는 각 집합의 합을 반환하고, INTERSECT 연산자는 각 집합의 공통된 행을 반환하고, MINUS 연산자는 한 집합에서 다른 집합을 뺀 행을 반환한다.

집합연산자	설 명
UNION UNION ALL	• 각 집합의 합을 반환하는 합 집합 - UNION ALL은 모든 원소를 포함한 합집합
INTERSECT	• 각 집합의 공통된 행을 반환하는 교집합
MINUS	• 한 집합에서 다른 집합을 뺀 행을 반환하는 차집합

### ☞ 집합 연산자 사용시 주의사항

① 각 SELECT문은 ORDER BY절을 포함하지 않는다. 그러나 전체 결과 행에 대한 ORDER BY절을 포함할 수 있다.



- ② 첫 번째 SELECT문의 칼럼 수와 두 번째 SELECT문의 칼럼 수가 반드시 같아야 한다.
- ③ 첫 번째 SELECT문의 칼럼과 두 번째 SELECT문의 칼럼의 데이터타입이 반드시 같아야 한다.
- ④ 전체 결과 행에 관한 ORDER BY절은 칼럼명보다 순서번호를 사용한다.

#### ☞ 집합 연산자의 예

각 SELECT문의 결과에 대한 두 집합  $a = \{1, 3, 5, 7, 9\}$ ,  $b = \{1, 2, 3, 4\}$ 가 있을 때,

✓ **UNION** 집합연산자는 각 집합의 합을 반환하는 합집합과 같다.

- $a \text{ UNION } b = \{1, 2, 3, 4, 5, 7, 9\}$
- $a \text{ UNION ALL } b = \{1, 1, 2, 3, 3, 4, 5, 7, 9\}$

✓ **INTERSECT** 집합연산자는 각 집합의 공통된 행을 반환하는 교집합과 같다.

- $a \text{ INTERSECT } b = \{1, 3\}$

✓ **MINUS** 집합연산자는 한 집합에서 다른 집합을 뺀 행을 반환하는 차집합과 같다.

- $a \text{ MINUS } b = \{5, 7\}$

예) 1. **Course** 테이블과 **T\_Course** 테이블을 모두 합하여 과목코드 순으로 출력하시오.

2. **Professor** 테이블과 **Course** 테이블을 이용하여 과목을 개설한 교수의 교수번호를 교수번호 순으로 출력하시오.

3. **Student** 테이블과 **SG\_Scores** 테이블을 참조하여 학점을 취득하지 못한 학생의 학번을 출력하시오.