

## 07강 조인

## 조인의 개념

- 여러 테이블에 저장된 데이터를 한꺼번에 조회할 필요가 있을때 조인을 사용한다.
- 예를 들어 특정 사원번호를 가진 사원이 소속된 부서명을 출력하고자 한다면

사원번호는 employee테이블에 정보가 있지만 부서명은department 테이블에 정보가 있기 때문에

일반적으로 사원번호로 직접 부서명을 알아 올 수 있는 방법이 없다.

## 조인의 개념

- 그래서 우선 사원번호를 가지고 사원테이블에서 부서번호를 알아 오고, 부서번호를 가지고 부서테이블에서 부서명을 알아와야 한다.
- 이렇듯 특정 정보를 가지고 어떤 정보를 조회하고자 할 때 여러 테이블을 조회해야 하는 경우가 있는데 이때 조인기술을 사용한다.

## 조인의 개념

- 조인은 기본적으로 두 테이블의 결합을 의미하는데 이때 어떤 조건없이 평면 결합을 '카테시안 곱' 이라고 한다.

```
SELECT * FROM employee, department;
```

- 카테시안 곱으로 얻어지는 컬럼의 수는 employee 컬럼 8개와 department 컬럼 3개를 합한 11개이며
- 레코드의 수는 employee의 레코드수 14개와 department레코드 4개를 곱한 56개의 레코드가 생성된다

## 조인의 종류

- 오라클에서 사용하는 조인(8i 이전부터 사용하는 방식)
  - EQUI 조인, NON-EQUI 조인, OUTER조인, SELF 조인
- ANSI 표준 SQL 조인
  - INNER 조인, CROSS조인, NATURAL 조인, JOIN USING, OUTER 조인

## EQUI 조인

- 조인 대상의 두 테이블에서 공통의 컬럼을 '='(equal)비교를 통해서 같은 값을 가지는 행을 연결하여 결과를 생성하는 조인 방법
- 예: 부서번호를 기준으로 사원(employee)테이블과 부서(department)테이블을 조인해본다

```
SELECT *  
FROM employee, department  
WHERE employee.dno= department.dno;
```

- Where 조건절에 두 테이블의 컬럼을 비교해서 일치 하는 양 테이블의 데이터를 출력함

## EQUI 조인

- EQUI 조인이후 만들어진 테이블에 추가로 데이터를 조회하기 위해서 AND 연산을 사용한다.
- 조인된 테이블에서 eno가 7788인 사원의 정보를 출력하는 예제

```
SELECT *  
FROM employee, department  
WHERE employee.dno= department.dno  
AND eno=7788;
```

- ANSI에서는 Inner JOIN또는 JOIN으로 사용할 수 있다

```
SELECT e.eno, e.ename, d.dname, e.dno  
FROM employee e INNER JOIN department d  
ON e.dno = d.dno  
WHERE e.eno = 7788;
```

## EQUI 조인

- 컬럼명이 모호한 경우 - 어느 테이블에서 온 컬럼인지 모를 때
- 예

```
SELECT eno,ename,dname, dno
FROM employee, department
WHERE employee.dno= department.dno
AND eno=7788;
```

오류 발생

- 변경사항 - 각 컬럼에 테이블명을 붙여준다

```
SELECT employee.eno,employee.ename,department.dname, department.dno
FROM employee, department
WHERE employee.dno= department.dno
AND employee.eno=7788;
```

- 오류가 발생하는 컬럼만 테이블명을 붙여 주어도 된다.

```
SELECT eno,ename,dname, department.dno
FROM employee, department
WHERE employee.dno= department.dno
AND eno=7788;
```



## EQUI 조인

- 칼럼명이 모호한 경우 - 어느 테이블에서 온 컬럼인지 모를 때
- 테이블에 별칭을 부여할 수 있다
  - 테이블 별칭은 30자까지 가능
  - FROM절에서 테이블명을 명시하고 공백 후 별칭 지정
  - 하나의 SQL 명령문에서 테이블명과 별명을 혼용할 수 없다.
  - 테이블 별칭은 해당 SQL 내부에서만 사용 가능하다

```
SELECT e.eno,e.ename,d.dname, d.dno
FROM employee e, department d
WHERE e.dno= d.dno
AND e.eno=7788;
```

## EQUI 조인 ~ NATURAL 조인

- 과거에 오라클에서 EQUI방식으로 사용하던 방식의 조인
- WHERE절을 사용하지 않고 자동적으로 모든 컬럼을 대상으로 데이터 유형 및 이름을 가지고 일치여부를 조사한 후 해당 컬럼으로 조인을 수행한다.

```
SELECT e.eno,e.ename,d.dname, d.dno  
FROM employee e NATURAL JOIN department d  
WHERE e.eno=7788;
```

오류 발생

```
SELECT e.eno,e.ename,d.dname, dno  
FROM employee e NATURAL JOIN department d  
WHERE e.eno=7788;
```

## EQUI 조인 ~ JOIN USING

- USING 절에 조인 대상이 되는 컬럼을 지정하는 방식

```
SELECT e.eno,e.ename,d.dname, dno  
FROM employee e JOIN department d  
USING(dno)  
WHERE e.eno=7788;
```

## EQUI 조인 ~ JOIN ON

- NATURAL 조인은 기본적으로 조인할 칼럼의 이름의 일치여부가 확인하는데 JOIN ON 사용하면 조인할 칼럼을 지정할 수 있다.

```
SELECT e.eno, e.ename, d.dname, e.dno  
FROM employee e JOIN department d  
ON e.dno = d.dno  
WHERE e.eno = 7788;
```

## NON - EQUI 조인

- EQUI 조인은 조인의 대상이 되는 컬럼의 일치를 가지고 조인을 한다.
- NON-EQUI 조인은 조인의 조건이 일치여부가 아닌 특정 범위 내에 있는지 조사하기 위해 사용한다
- WHERE 조건절에 <, > 나 BETWEEN a AND b 와 같은 '='이 아닌 연산자 사용

## NON - EQUI 조인

- 급여 등급 테이블의 정보를 확인

```
SELECT * FROM salgrade;
```

- 사원 정보에서 급여를 기준으로 급여등급 테이블의 급여범위로 등급을 결정하는 조인을 진행하는 예제

```
SELECT ename, salary, grade  
FROM employee, salgrade  
WHERE salary BETWEEN losal AND hisal;
```

## NON - EQUI 조인

- 급여 등급 테이블의 정보를 확인

```
SELECT * FROM salgrade;
```

- 사원 정보에서 급여를 기준으로 급여등급 테이블의 급여범위로 등급을 결정하는 조인을 진행하는 예제

```
SELECT ename, salary, grade  
FROM employee, salgrade  
WHERE salary BETWEEN losal AND hisal;
```

```
SELECT ename, salary, grade  
FROM employee, salgrade  
WHERE salary >= losal AND salary <= hisal;
```

### 3중 조인

- 사원의 이름과 소속된 부서이름, 그리고 급여와 급여 등급을 출력하는 쿼리문 예제

```
SELECT e.ename,d.dname ,e.salary, s.grade
FROM employee e,department d, salgrade s
WHERE e.dno = d.dno
AND e.salary BETWEEN s.losal AND s.hisal;
```



## SELF 조인

- SELF라는 말 그대로 자기 자신과 조인을 맺는 것을 말한다.
- 조인 대상이 자기 자신이라는 것 외는 EQUI와 같다.
- 예 사원이름과 담당 상관이름을 출력해보자(Manager)  
담당상관번호는 사원번호와 일치한다.

```
SELECT employee.ename as "사원이름", manager.ename as "직속 상관이름"  
FROM employee employee, employee manager  
WHERE employee.manager = manager.eno;
```

## SELF 조인 ~ JOIN ON

- 예 사원이름과 담당 상관이름을 출력해보자(Manager)  
담당상관번호는 사원번호와 일치한다.

```
SELECT employee.ename || '의 직속상관은 ' || manager.ename  
FROM employee employee JOIN employee manager  
ON employee.manager = manager.eno;
```

## OUTER JOIN

- EQUI 조인에서 조인 조건의 칼럼에서 한쪽의 데이터가 없는 경우  
= 비교 결과는 거짓이라 조인되지 않는다.
- 양측 칼럼의 값 중 하나가 NULL인 상태로 조인 결과를 출력할 필요가 있을 때 OUTER JOIN을 사용한다.
- OUTER 조인을 할 때는 테이블의 기준점을 잡는 것이 중요하다.

## Inner JOIN

- 조인 결합 조건이 일치 하는 것만 출력

물품

물품코드	물품 이름	물품 수량	물품 가격
A001	연필	100	100
A002	지우개	200	500
A003	필통	300	1000
A004	노트	400	500

거래정보

거래처 정보	물품 코드	거래량
S001	A001	10
S002	A002	20
S003	A004	20
S004	A005	10

### Inner Join EQUI 조인

물품코드	물품 이름	물품 수량	물품 가격	거래처 정보	거래량
A001	연필	100	100	S001	10
A002	지우개	200	500	S002	20
A004	노트	400	500	S003	20

## LEFT OUTER JOIN

- FROM 기준으로 왼쪽 테이블이 모두 합쳐져야 함을 의미

물품

물품코드	물품 이름	물품 수량	물품 가격
A001	연필	100	100
A002	지우개	200	500
A003	필통	300	1000
A004	노트	400	500

거래정보

거래처 정보	물품 코드	거래량
S001	A001	10
S002	A002	20
S003	A004	20
S004	A005	10

LEFT OUTER 조인 : FROM 물품 left outer join 거래정보

물품코드	물품 이름	물품 수량	물품 가격	거래처 정보	거래량
A001	연필	100	100	S001	10
A002	지우개	200	500	S002	20
A003	필통	300	1000	NULL	NULL
A004	노트	400	500	S003	20

LEFT OUTER 조인 : FROM 거래정보 left outer join 물품

거래처 정보	물품 코드	거래량	물품 이름	물품 수량	물품 가격
S001	A001	10	연필	100	100
S002	A002	20	지우개	200	500
S003	A004	20	노트	400	500
S004	A005	10	NULL	NULL	NULL

## RIGHT OUTER JOIN

- FROM 기준으로 오른쪽 테이블이 모두 합쳐져야 함을 의미

물품

물품코드	물품 이름	물품 수량	물품 가격
A001	연필	100	100
A002	지우개	200	500
A003	필통	300	1000
A004	노트	400	500

거래정보

거래처 정보	물품 코드	거래량
S001	A001	10
S002	A002	20
S003	A004	20
S004	A005	10

RIGHT OUTER JOIN : FROM 물품 right outer join 거래정보

물품코드	물품 이름	물품 수량	물품 가격	거래처 정보	거래량
A001	연필	100	100	S001	10
A002	지우개	200	500	S002	20
A004	노트	400	500	S003	20
A005	NULL	NULL	NULL	S004	10

RIGHT OUTER JOIN : FROM 거래정보 right outer join 물품

거래처 정보	물품 코드	거래량	물품 이름	물품 수량	물품 가격
S001	A001	10	연필	100	100
S002	A002	20	지우개	200	500
NULL	A003	NULL	필통	300	1000
S003	A004	20	노트	400	500

## FULL OUTER JOIN

LEFT OUTER와 RIGHT OUTER 조인을 합친 형태 : FULL OUTER

물품

물품코드	물품 이름	물품 수량	물품 가격
A001	연필	100	100
A002	지우개	200	500
A003	필통	300	1000
A004	노트	400	500

거래정보

거래처 정보	물품 코드	거래량
S001	A001	10
S002	A002	20
S003	A004	20
S004	A005	10

FULL OUTER JOIN : FROM 물품 full outer join 거래정보

물품코드	물품 이름	물품 수량	물품 가격	거래처 정보	거래량
A001	연필	100	100	S001	10
A002	지우개	200	500	S002	20
A003	필통	300	1000	NULL	NULL
A004	노트	400	500	S003	20
A005	NULL	NULL	NULL	S004	10

## OUTER JOIN

- 오라클에서는 NULL이 출력 되어야 하는 쪽 조건에 (+)기호를 붙인다.
- 예제 : 사원이름과 담당 상관이름을 출력해보자(Manager)  
단 상관 번호가 없는 경우 그냥 출력한다.

```
SELECT employee.ename || '의 직속상관은 ' || manager.ename  
FROM employee employee JOIN employee manager  
ON employee.manager = manager.eno(+);
```



## OUTER JOIN

- ANSI 기준으로 LEFT, RIGHT, FULL 세가지를 지원한다.
- 예제 : 사원이름과 담당 상관이름을 출력해보자(Manager)  
단 상관 번호가 없는 경우 그냥 출력한다.

```
SELECT employee.ename || '의 직속상관은 ' || manager.ename  
FROM employee employee LEFT OUTER JOIN employee manager  
ON employee.manager = manager.eno;
```