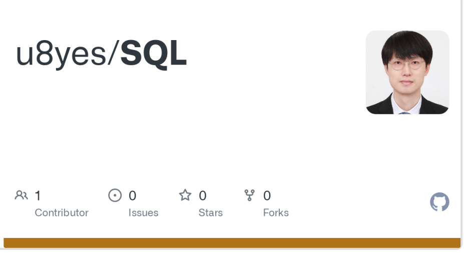
day71; 20221216

날짜	@2022년 12월 16일
유형	@2022년 12월 16일
태그	

GitHub - u8yes/SQL

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to refresh your session.

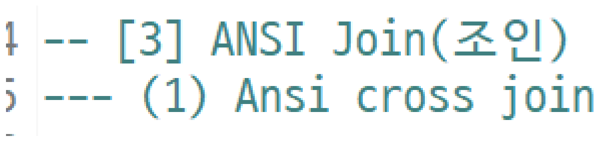
https://github.com/u8yes/SQL



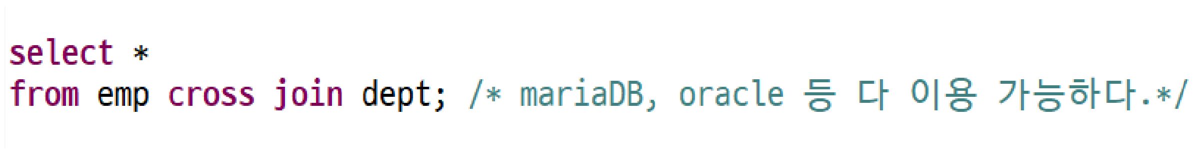
https://s3-us-west-2.amazonaws.com/secure.notion-static.com/e657a294-244b-4ecc-b757-dafa9eff9f4a/%EC%84%9C%EB%B8%8C%EC%BF%BC%EB%A6%AC.pdf

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/24aecb89-c3b6-452f-966f-720ea698286a/dbproj.sql

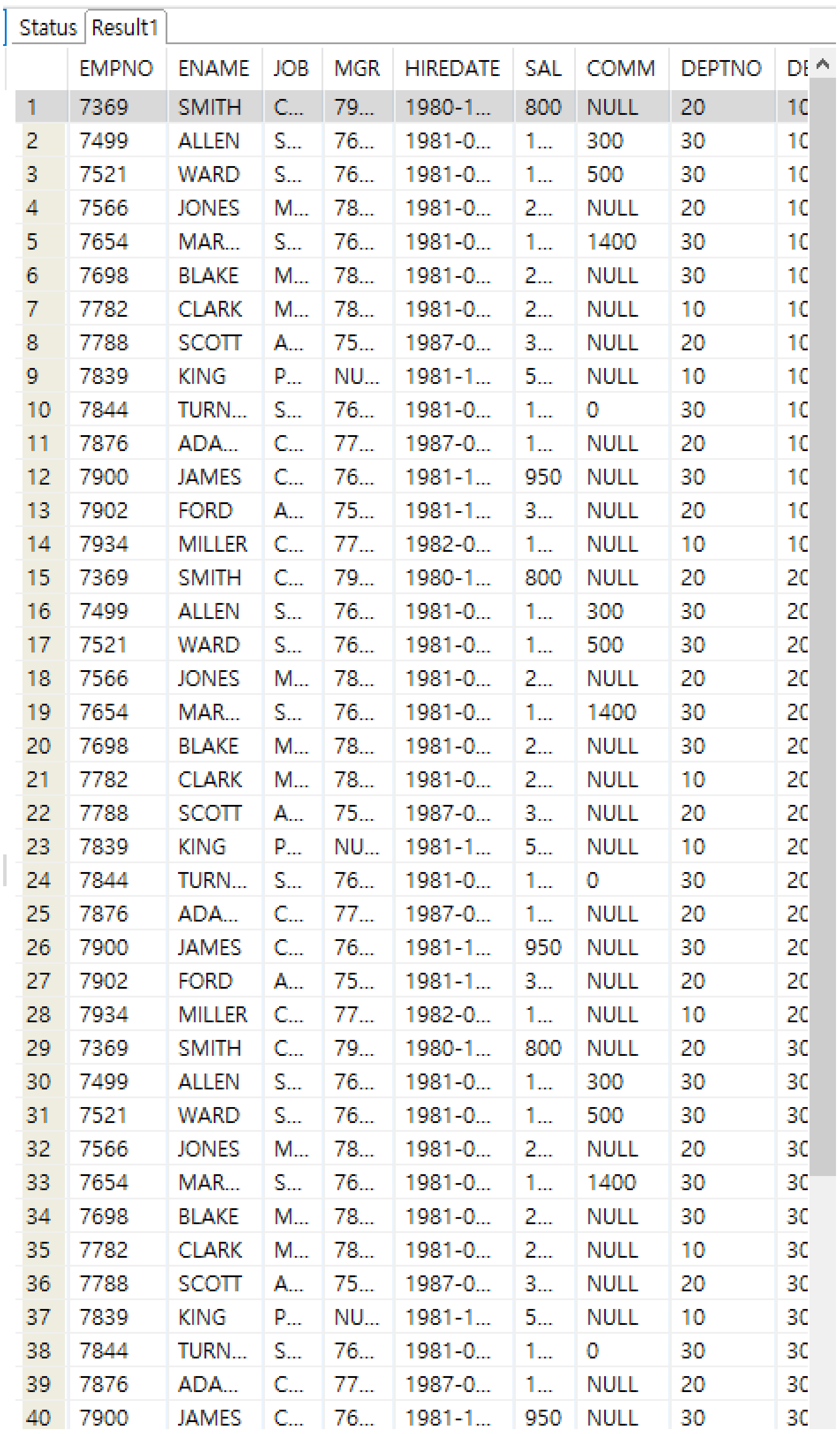
SQL



```
-- [3] ANSI Join(조인)
--- (1) Ansi cross join
```

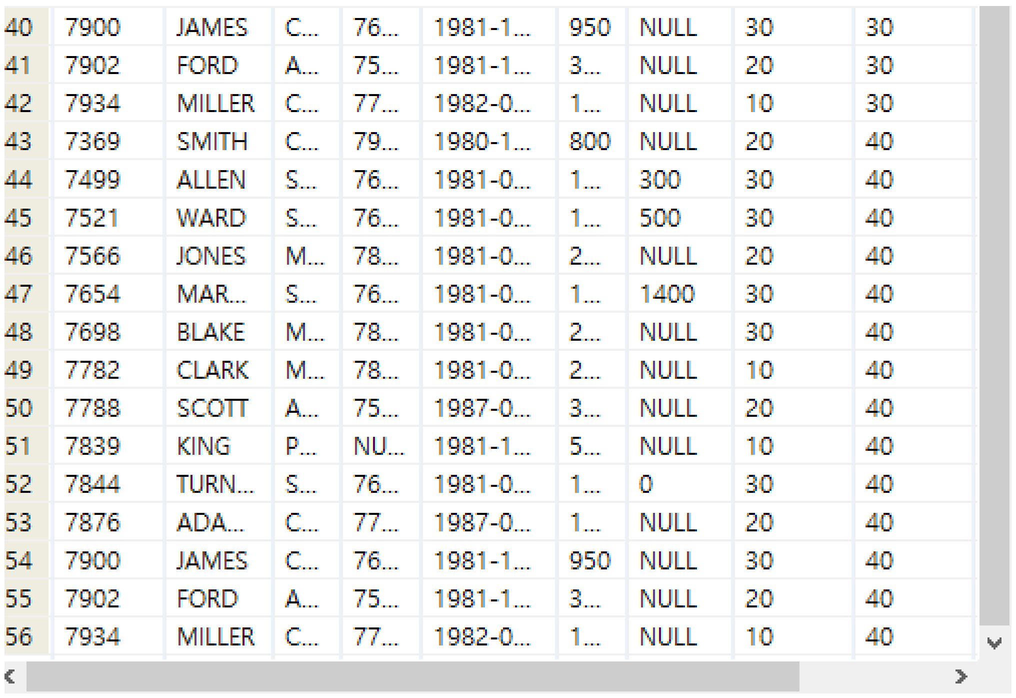


```
select *
from emp cross join dept; /* mariaDB, oracle 등 다 이용 가능하다.*/
```



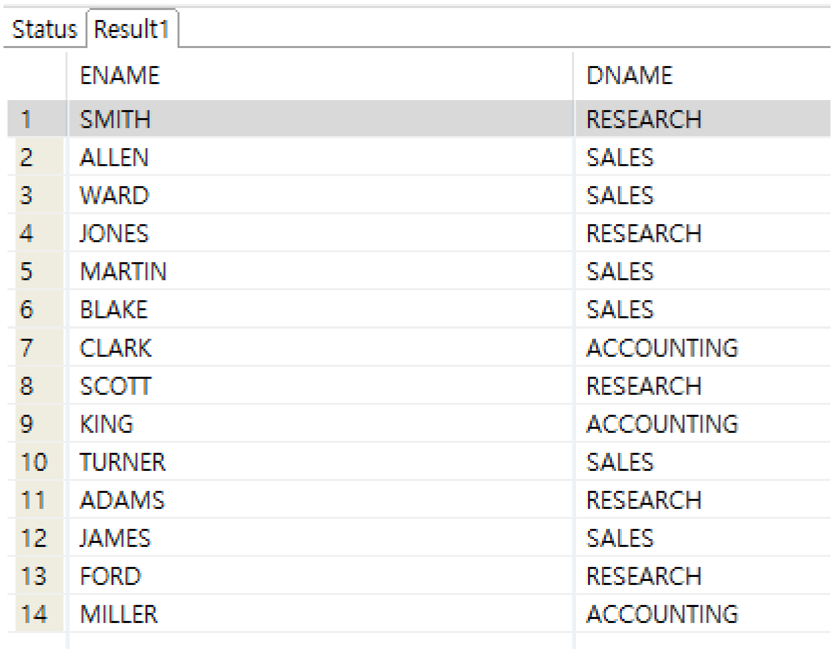
Status | Result1

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DE
1	7369	SMITH	C...	79...	1980-1...	800	NULL	20	10
2	7499	ALLEN	S...	76...	1981-0...	1...	300	30	10
3	7521	WARD	S...	76...	1981-0...	1...	500	30	10
4	7566	JONES	M...	78...	1981-0...	2...	NULL	20	10
5	7654	MAR...	S...	76...	1981-0...	1...	1400	30	10
6	7698	BLAKE	M...	78...	1981-0...	2...	NULL	30	10
7	7782	CLARK	M...	78...	1981-0...	2...	NULL	10	10
8	7788	SCOTT	A...	75...	1987-0...	3...	NULL	20	10
9	7839	KING	P...	NU...	1981-1...	5...	NULL	10	10
10	7844	TURN...	S...	76...	1981-0...	1...	0	30	10
11	7876	ADA...	C...	77...	1987-0...	1...	NULL	20	10
12	7900	JAMES	C...	76...	1981-1...	950	NULL	30	10
13	7902	FORD	A...	75...	1981-1...	3...	NULL	20	10
14	7934	MILLER	C...	77...	1982-0...	1...	NULL	10	10
15	7369	SMITH	C...	79...	1980-1...	800	NULL	20	20
16	7499	ALLEN	S...	76...	1981-0...	1...	300	30	20
17	7521	WARD	S...	76...	1981-0...	1...	500	30	20
18	7566	JONES	M...	78...	1981-0...	2...	NULL	20	20
19	7654	MAR...	S...	76...	1981-0...	1...	1400	30	20
20	7698	BLAKE	M...	78...	1981-0...	2...	NULL	30	20
21	7782	CLARK	M...	78...	1981-0...	2...	NULL	10	20
22	7788	SCOTT	A...	75...	1987-0...	3...	NULL	20	20
23	7839	KING	P...	NU...	1981-1...	5...	NULL	10	20
24	7844	TURN...	S...	76...	1981-0...	1...	0	30	20
25	7876	ADA...	C...	77...	1987-0...	1...	NULL	20	20
26	7900	JAMES	C...	76...	1981-1...	950	NULL	30	20
27	7902	FORD	A...	75...	1981-1...	3...	NULL	20	20
28	7934	MILLER	C...	77...	1982-0...	1...	NULL	10	20
29	7369	SMITH	C...	79...	1980-1...	800	NULL	20	30
30	7499	ALLEN	S...	76...	1981-0...	1...	300	30	30
31	7521	WARD	S...	76...	1981-0...	1...	500	30	30
32	7566	JONES	M...	78...	1981-0...	2...	NULL	20	30
33	7654	MAR...	S...	76...	1981-0...	1...	1400	30	30
34	7698	BLAKE	M...	78...	1981-0...	2...	NULL	30	30
35	7782	CLARK	M...	78...	1981-0...	2...	NULL	10	30
36	7788	SCOTT	A...	75...	1987-0...	3...	NULL	20	30
37	7839	KING	P...	NU...	1981-1...	5...	NULL	10	30
38	7844	TURN...	S...	76...	1981-0...	1...	0	30	30
39	7876	ADA...	C...	77...	1987-0...	1...	NULL	20	30
40	7900	JAMES	C...	76...	1981-1...	950	NULL	30	30



40	7900	JAMES	C...	76...	1981-1...	950	NULL	30	30
41	7902	FORD	A...	75...	1981-1...	3...	NULL	20	30
42	7934	MILLER	C...	77...	1982-0...	1...	NULL	10	30
43	7369	SMITH	C...	79...	1980-1...	800	NULL	20	40
44	7499	ALLEN	S...	76...	1981-0...	1...	300	30	40
45	7521	WARD	S...	76...	1981-0...	1...	500	30	40
46	7566	JONES	M...	78...	1981-0...	2...	NULL	20	40
47	7654	MAR...	S...	76...	1981-0...	1...	1400	30	40
48	7698	BLAKE	M...	78...	1981-0...	2...	NULL	30	40
49	7782	CLARK	M...	78...	1981-0...	2...	NULL	10	40
50	7788	SCOTT	A...	75...	1987-0...	3...	NULL	20	40
51	7839	KING	P...	NU...	1981-1...	5...	NULL	10	40
52	7844	TURN...	S...	76...	1981-0...	1...	0	30	40
53	7876	ADA...	C...	77...	1987-0...	1...	NULL	20	40
54	7900	JAMES	C...	76...	1981-1...	950	NULL	30	40
55	7902	FORD	A...	75...	1981-1...	3...	NULL	20	40
56	7934	MILLER	C...	77...	1982-0...	1...	NULL	10	40

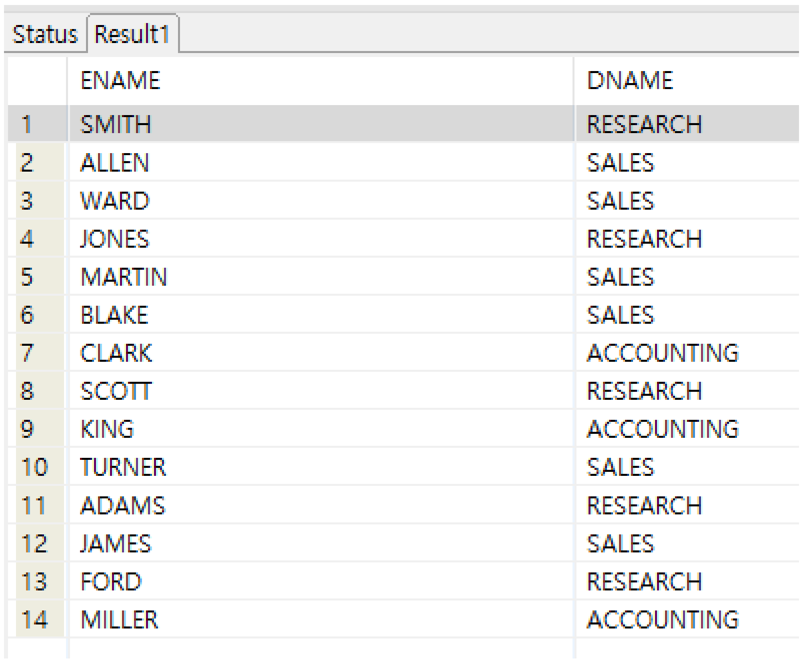
```
--- (2) Ansi inner join
/* equi join 비슷하다. */
select ename, dname
/* ename은 emp테이블, dname은 dept테이블*/
from emp inner join dept
on emp.deptno = dept.deptno;
```



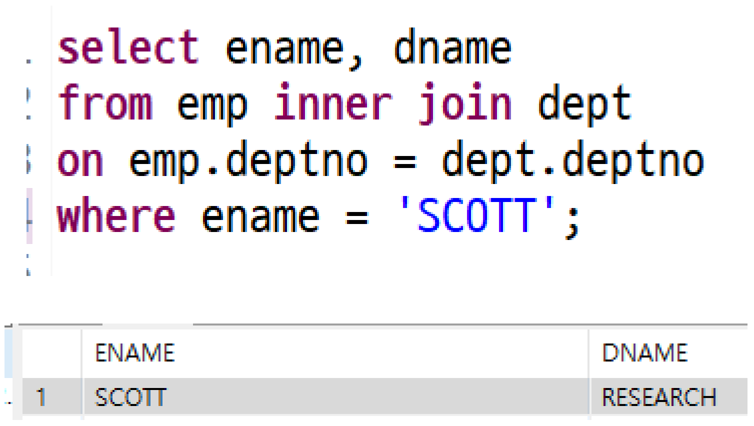
	ENAME	DNAME
1	SMITH	RESEARCH
2	ALLEN	SALES
3	WARD	SALES
4	JONES	RESEARCH
5	MARTIN	SALES
6	BLAKE	SALES
7	CLARK	ACCOUNTING
8	SCOTT	RESEARCH
9	KING	ACCOUNTING
10	TURNER	SALES
11	ADAMS	RESEARCH
12	JAMES	SALES
13	FORD	RESEARCH
14	MILLER	ACCOUNTING

dept.deptno는 primary, emp.deptno FK_DEPTNO

```sql
select ename, dname
/* ename은 emp테이블, dname은 dept테이블*/
from emp inner join dept
using (deptno); -- () 생략 불가능
```



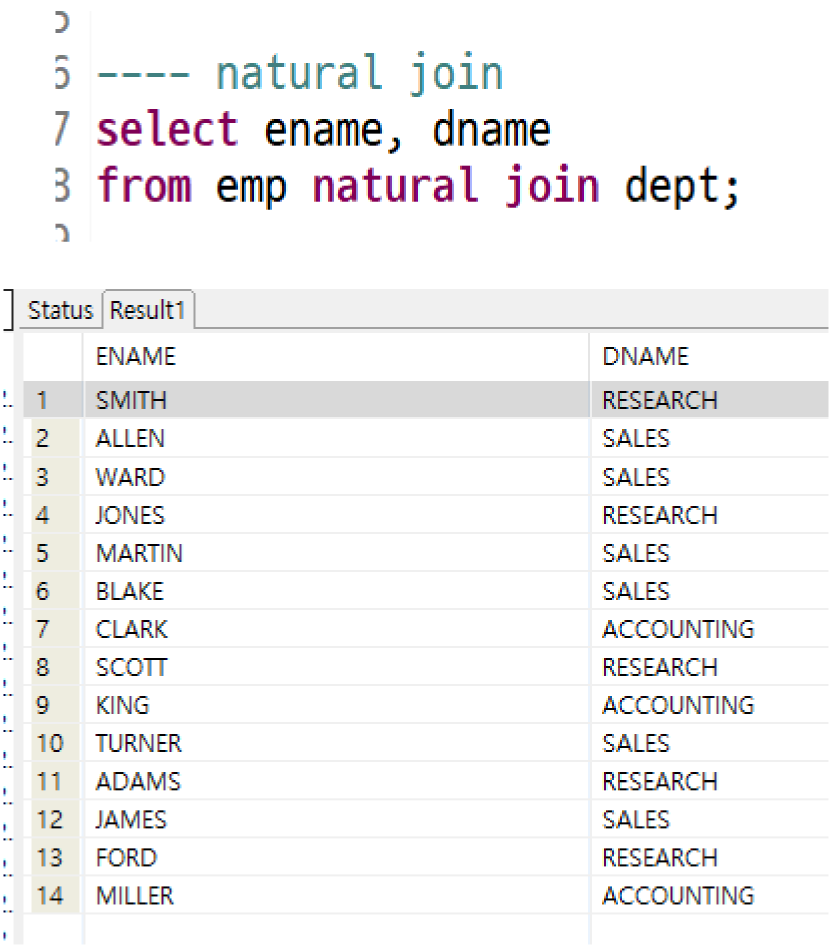
	ENAME	DNAME
1	SMITH	RESEARCH
2	ALLEN	SALES
3	WARD	SALES
4	JONES	RESEARCH
5	MARTIN	SALES
6	BLAKE	SALES
7	CLARK	ACCOUNTING
8	SCOTT	RESEARCH
9	KING	ACCOUNTING
10	TURNER	SALES
11	ADAMS	RESEARCH
12	JAMES	SALES
13	FORD	RESEARCH
14	MILLER	ACCOUNTING



```sql
select ename, dname
from emp inner join dept
on emp.deptno = dept.deptno
where ename = 'SCOTT';
```

	ENAME	DNAME
1	SCOTT	RESEARCH

두 개의 테이블에 공통된 컬럼이 반드시 있어야 됨. (on을 붙이지 않아도 됨.)



```sql
---- natural join
select ename, dname
from emp natural join dept;
```

Status | Result1

	ENAME	DNAME
1	SMITH	RESEARCH
2	ALLEN	SALES
3	WARD	SALES
4	JONES	RESEARCH
5	MARTIN	SALES
6	BLAKE	SALES
7	CLARK	ACCOUNTING
8	SCOTT	RESEARCH
9	KING	ACCOUNTING
10	TURNER	SALES
11	ADAMS	RESEARCH
12	JAMES	SALES
13	FORD	RESEARCH
14	MILLER	ACCOUNTING

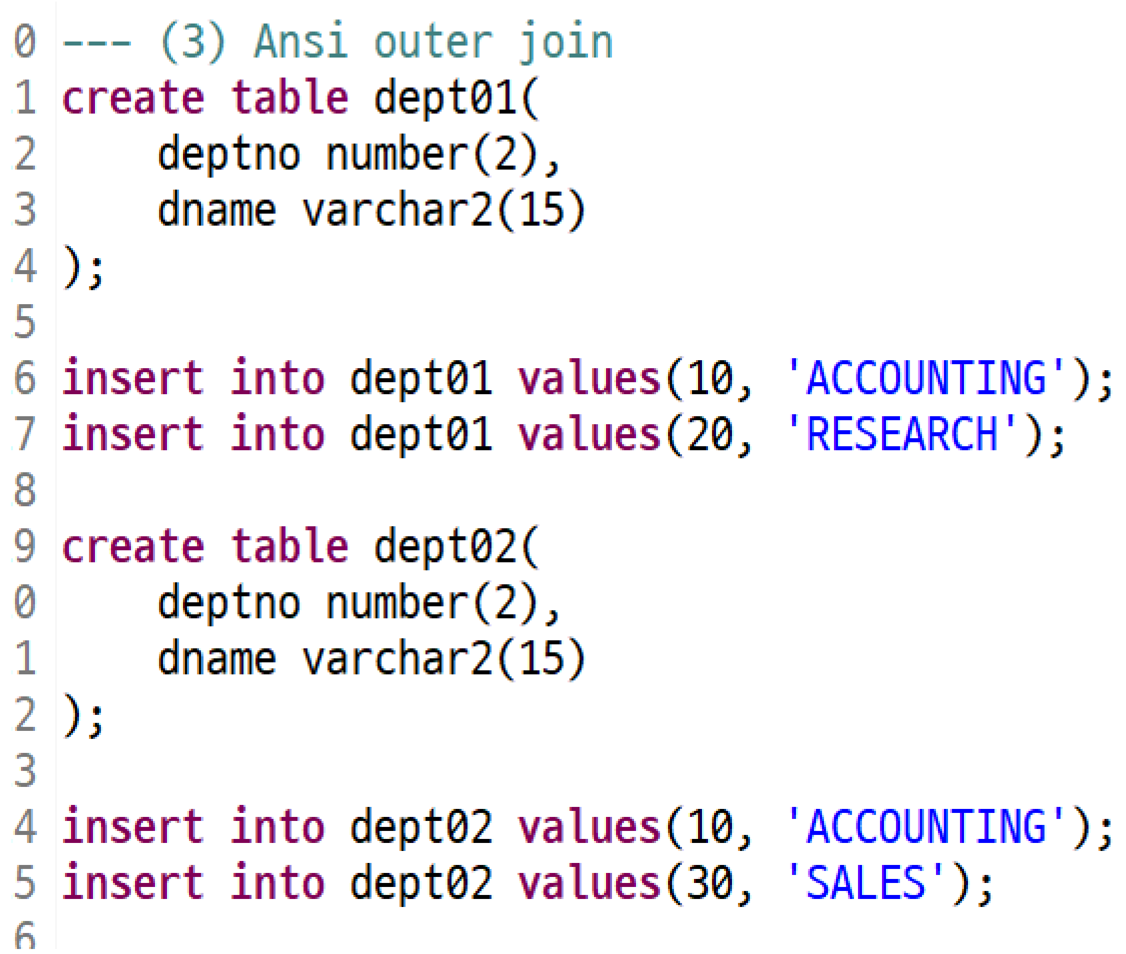
근데 ANSI가 뭐지?

American National Standards Institute의 약자인데

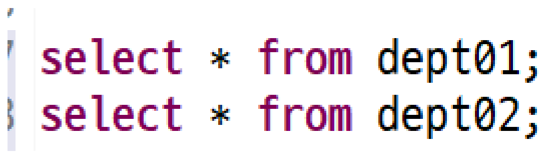
미국 국립 표준 협회의 약자가 ANSI인 것이다

미국 국립 표준 협회에서 모든 SQL에 사용할 수 있도록 만든건데

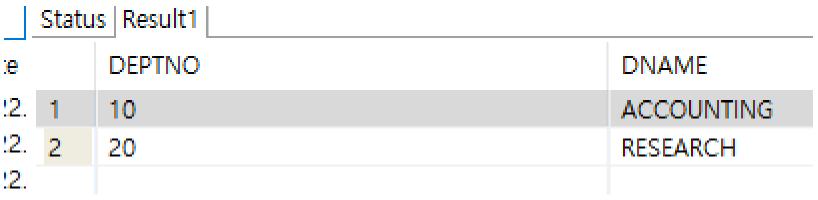
ANSI를 사용하면 어떤 SQL에서든 동일하게 사용이 가능하다



```sql
--- (3) Ansi outer join
create table dept01(
    deptno number(2),
    dname varchar2(15)
);

insert into dept01 values(10, 'ACCOUNTING');
insert into dept01 values(20, 'RESEARCH');

create table dept02(
    deptno number(2),
    dname varchar2(15)
);

insert into dept02 values(10, 'ACCOUNTING');
insert into dept02 values(30, 'SALES');
```

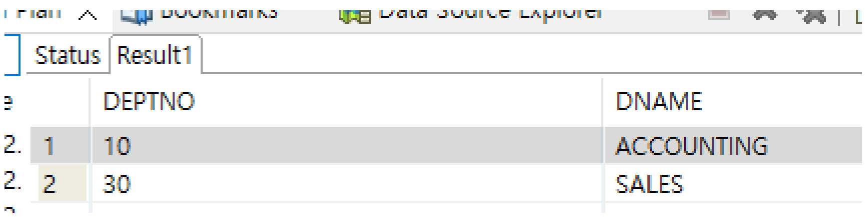


```sql
select * from dept01;
select * from dept02;
```



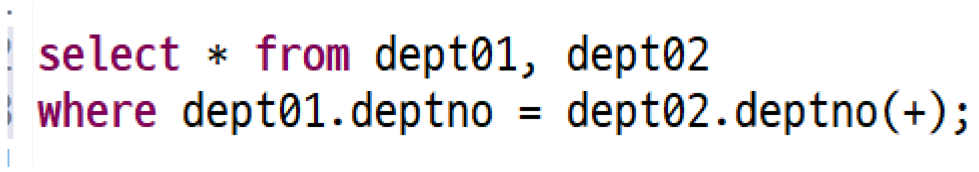
Status | Result1

	DEPTNO	DNAME
1	10	ACCOUNTING
2	20	RESEARCH

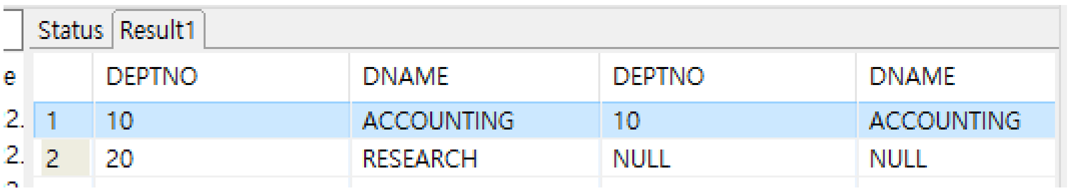


Status | Result1

	DEPTNO	DNAME
1	10	ACCOUNTING
2	30	SALES

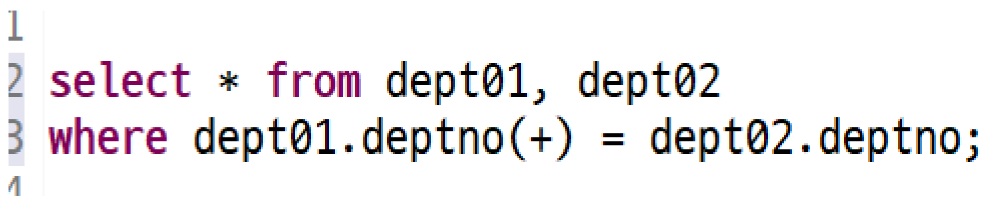


```
select * from dept01, dept02
where dept01.deptno = dept02.deptno(+);
```

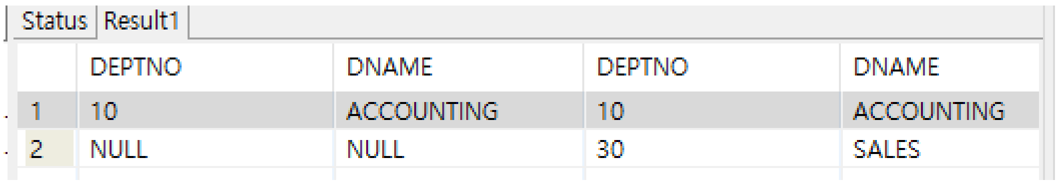


Status | Result1

	DEPTNO	DNAME	DEPTNO	DNAME
1	10	ACCOUNTING	10	ACCOUNTING
2	20	RESEARCH	NULL	NULL



```
select * from dept01, dept02
where dept01.deptno(+) = dept02.deptno;
```



Status | Result1

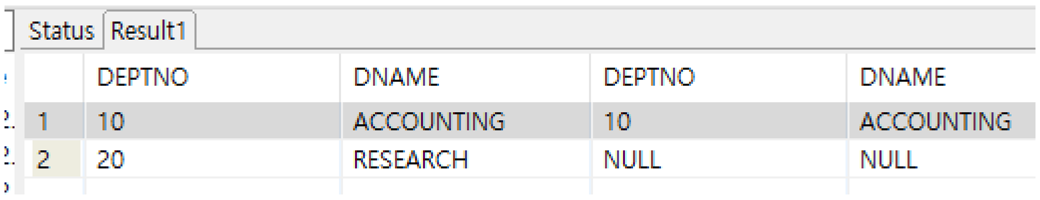
	DEPTNO	DNAME	DEPTNO	DNAME
1	10	ACCOUNTING	10	ACCOUNTING
2	NULL	NULL	30	SALES

```sql
select * from dept01, dept02
where dept01.deptno(+) = dept02.deptno(+);
/* ORA-01468: a predicate may reference only one outer-joined table */
```

에러가 나기에 차라리 cross 조인을 하라

left outer join

```sql
---- Ansi
select *
from dept01 left outer join dept02
on dept01.deptno = dept02.deptno;
/* Ansi 표준은 무조건 on으로 표시하고
 * 더 구체적으로 하고 싶을 경우에 where를 더 붙여준다.*/
```

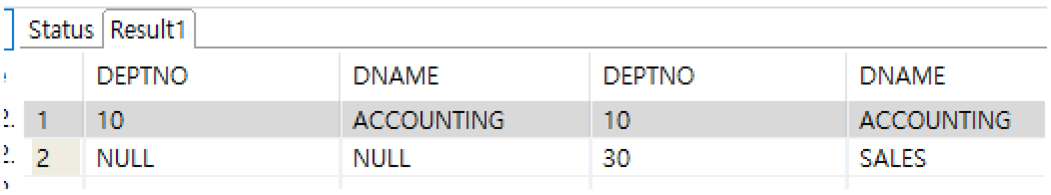


Status | Result1

	DEPTNO	DNAME	DEPTNO	DNAME
1	10	ACCOUNTING	10	ACCOUNTING
2	20	RESEARCH	NULL	NULL

right outer join

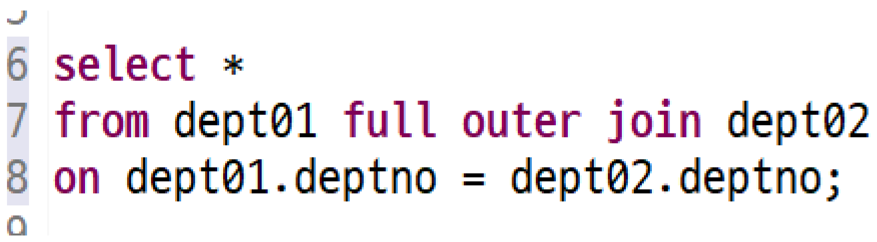
```sql
select *
from dept01 right outer join dept02
on dept01.deptno = dept02.deptno;
```



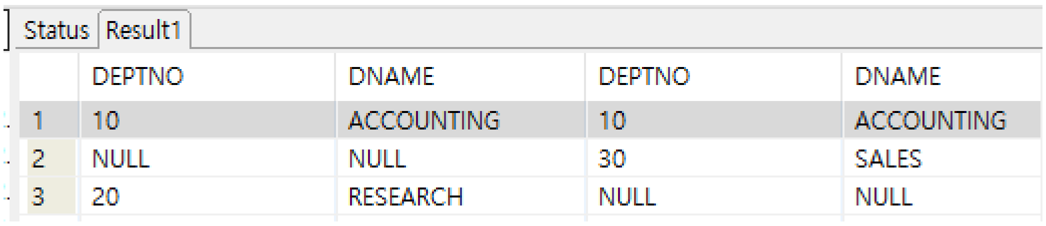
Status | Result1

	DEPTNO	DNAME	DEPTNO	DNAME
1	10	ACCOUNTING	10	ACCOUNTING
2	NULL	NULL	30	SALES

양쪽다(full outer join)



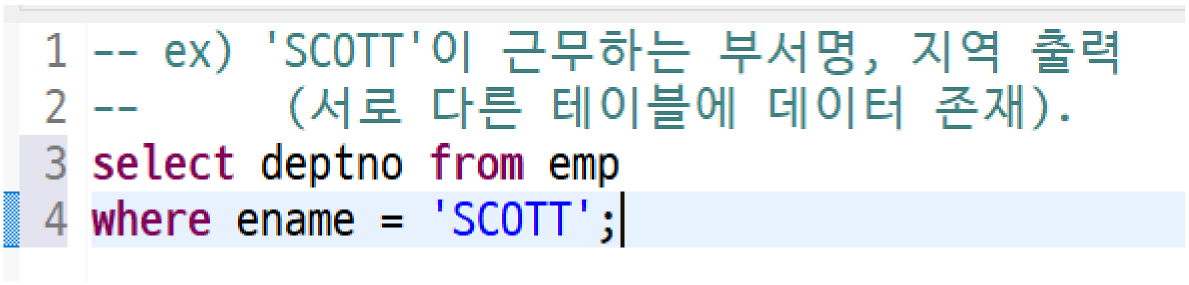
```sql
select *
from dept01 full outer join dept02
on dept01.deptno = dept02.deptno;
```



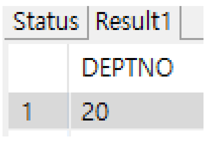
	DEPTNO	DNAME	DEPTNO	DNAME
1	10	ACCOUNTING	10	ACCOUNTING
2	NULL	NULL	30	SALES
3	20	RESEARCH	NULL	NULL

서브 쿼리

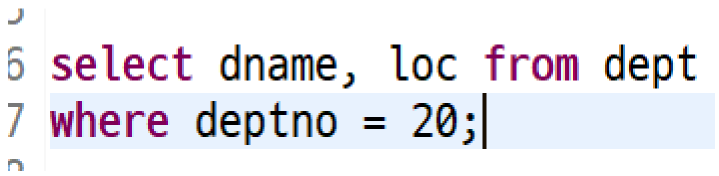
메인 쿼리가 실행되기 이전에 ( )안에 있는 것이 먼저 실행된다.



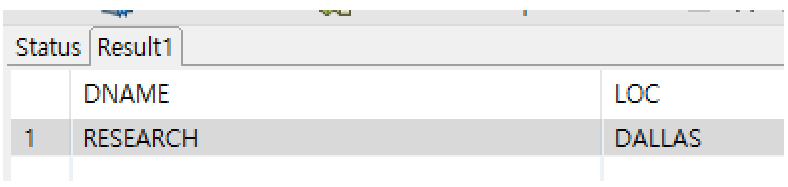
```sql
-- ex) 'SCOTT'이 근무하는 부서명, 지역 출력
--      (서로 다른 테이블에 데이터 존재).
select deptno from emp
where ename = 'SCOTT';
```



	DEPTNO
1	20

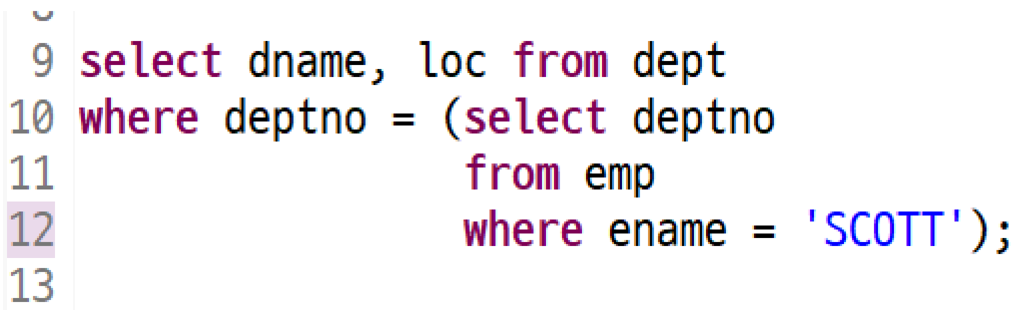


```sql
select dname, loc from dept
where deptno = 20;
```

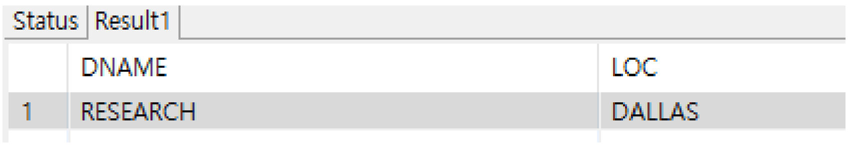


	DNAME	LOC
1	RESEARCH	DALLAS

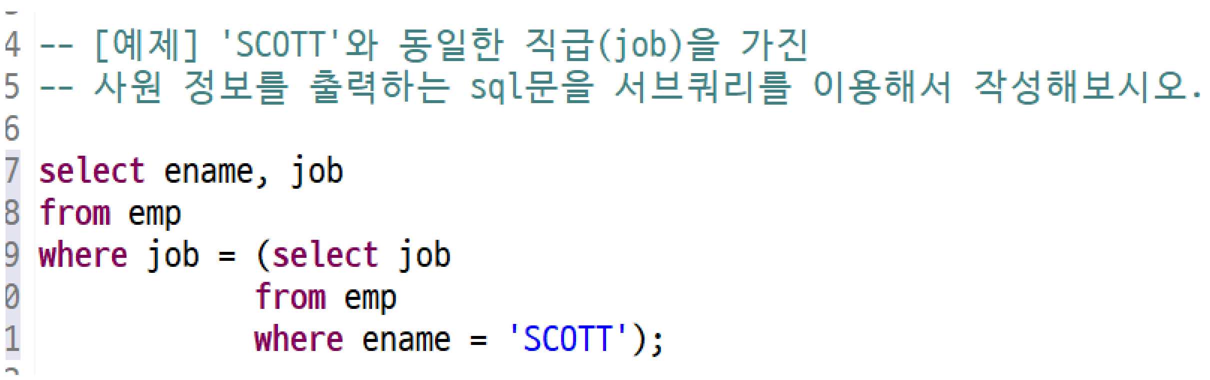
어차피 20을 부르기 보다는 서브쿼리를 작성해서 다른 테이블의 데이터를 하나의 쿼리문으로 작업



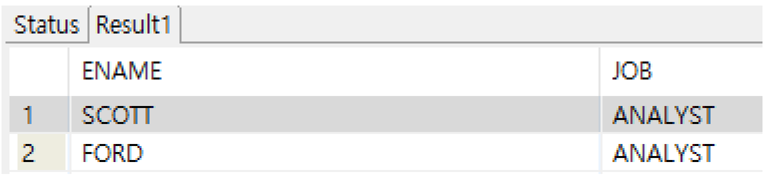
```
select dname, loc from dept
where deptno = (select deptno
                from emp
                where ename = 'SCOTT');
```



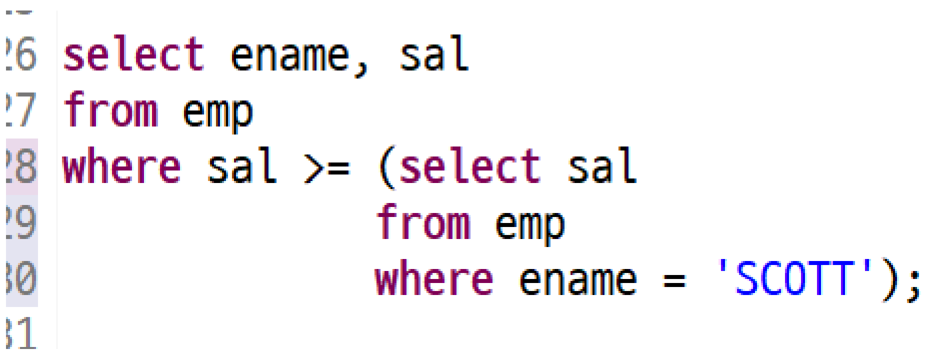
	DNAME	LOC
1	RESEARCH	DALLAS



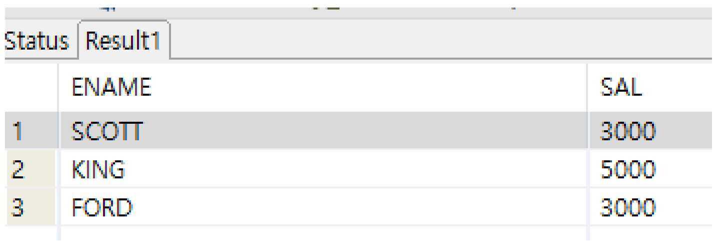
```
-- [예제] 'SCOTT'와 동일한 직급(job)을 가진
-- 사원 정보를 출력하는 sql문을 서브쿼리를 이용해서 작성해보시오.

select ename, job
from emp
where job = (select job
             from emp
             where ename = 'SCOTT');
```



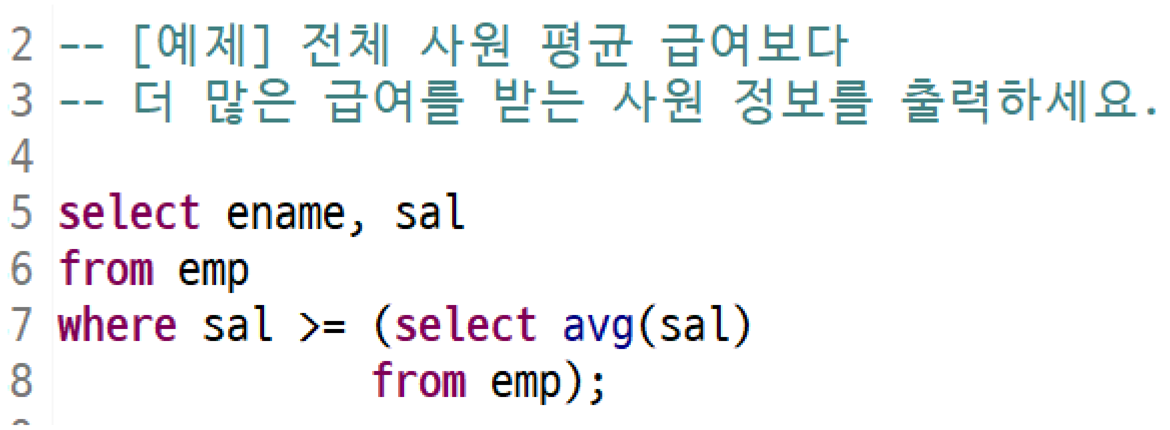
	ENAME	JOB
1	SCOTT	ANALYST
2	FORD	ANALYST



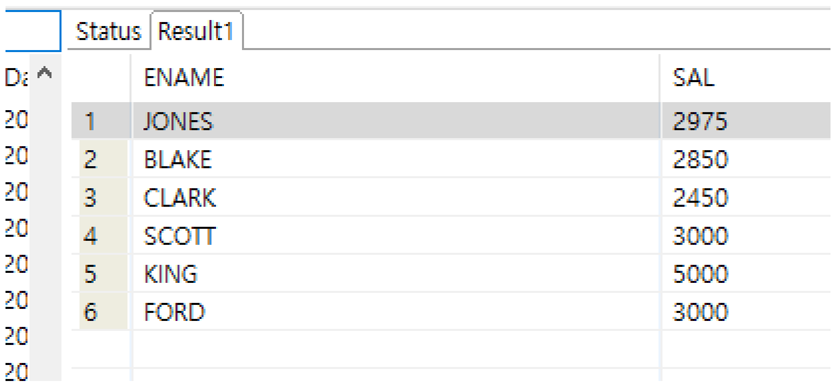
```sql
select ename, sal
from emp
where sal >= (select sal
              from emp
              where ename = 'SCOTT');
```



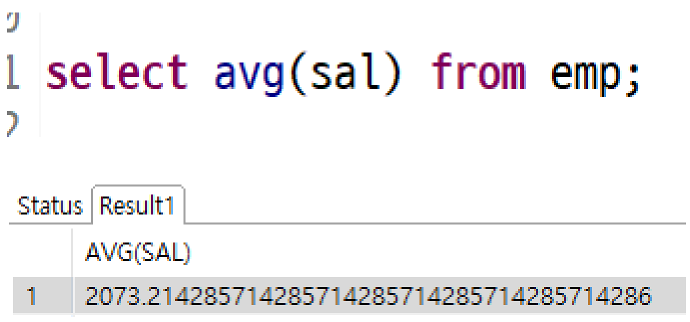
	ENAME	SAL
1	SCOTT	3000
2	KING	5000
3	FORD	3000



```sql
-- [예제] 전체 사원 평균 급여보다
-- 더 많은 급여를 받는 사원 정보를 출력하세요.

select ename, sal
from emp
where sal >= (select avg(sal)
              from emp);
```



	ENAME	SAL
1	JONES	2975
2	BLAKE	2850
3	CLARK	2450
4	SCOTT	3000
5	KING	5000
6	FORD	3000



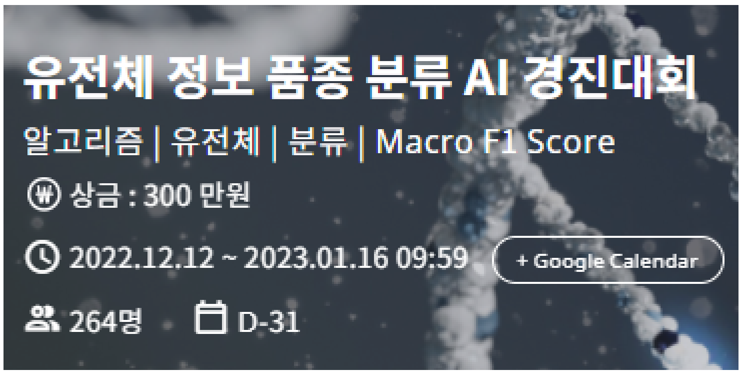
```
select avg(sal) from emp;
```

```
Status  Result1
   AVG(SAL)
1  2073.2142857142857142857142857142857142857142857142857
```

유전체 정보 품종 분류 AI 경진대회

상금 : 300 만원 264명 D-31 유전체 염기서열에서 획득한 유전체 변이 정보인 Single Nucleotide Polymorphism 정보는 특정 개체 및 특정 품종에 따라 다른 변이 양상을 나타낼 수 있기 때문에 동일개체를

https://dacon.io/competitions/official/236035/overview/description


유전체 정보 품종 분류 AI 경진대회
알고리즘 | 유전체 | 분류 | Macro F1 Score
상금 : 300 만원
2022.12.12 ~ 2023.01.16 09:59
+ Google Calendar
264명
D-31
