2018100936 박미란

감상문 : 저번주에 사용한 sql문법들을 더 확실하게 정의할 수 있었고 nvl과 inner join on을 알게되었고 저번에 개념을 확실히 잡지못한 group by를 이제 잘 활용할 수 있게 되었다. 오류를 고치는 과정에서 점점 발전해가는 나자신을 발견할 수 있어서 뿌듯했다.

<1번>  
1)

-> insert into emp (empno, ename, job, mgr, hiredate, sal, comm, deptno)

values(0936,'miran','student',5678,'23-03-01',60000,NULL,10);

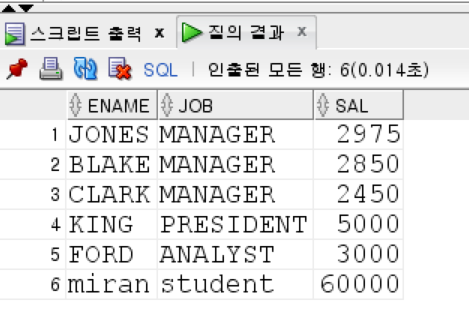
select \* from emp where ename like 'miran';



Insert를 이용해서 컬럼과 값을 짝을 맞춘다

2)

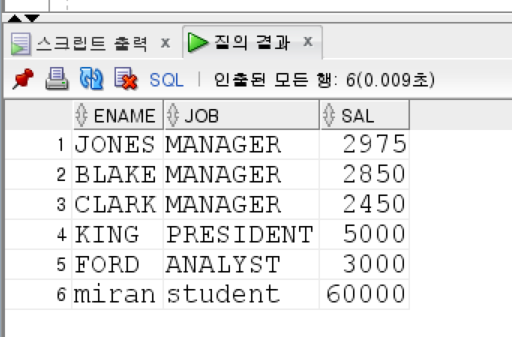
-> select ename, job, sal from emp where job = 'manager' or sal >=2000;



조건문 or을 사용하여 출력

3)

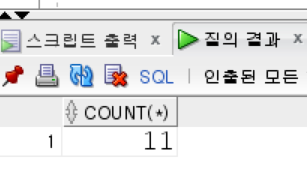
-> select ename, job, sal from emp where job = 'manager' or sal >=2000;



위에 문제와 똑같이 풀이

4)

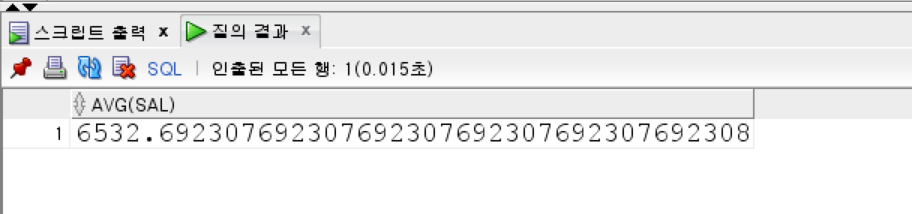
-> select count(\*)from emp where sal >=1000;



Count를 사용하여 직원수 구하기

5)

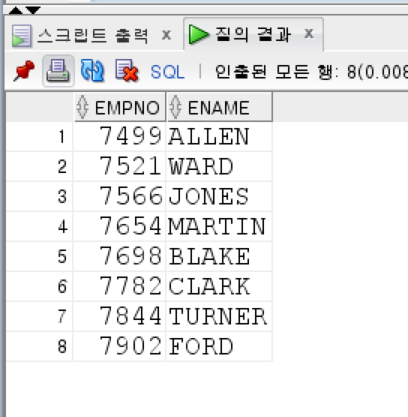
-> select avg(sal) from emp;



평균값을 구할 수 있는 avg사용

6)

-> select empno,ename from emp where job like '%A%';

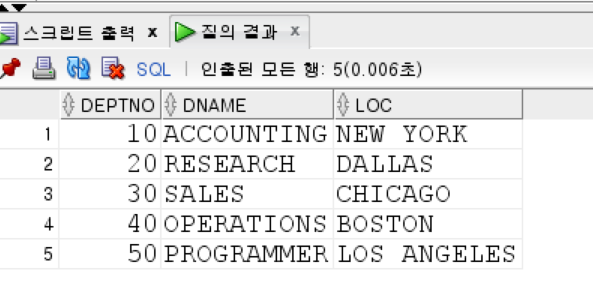


Where like를 사용해서 a가포함된 이름 출력

7)

->insert into dept(deptno,dname,loc) values(50,'PROGRAMMER','LOS ANGELES');

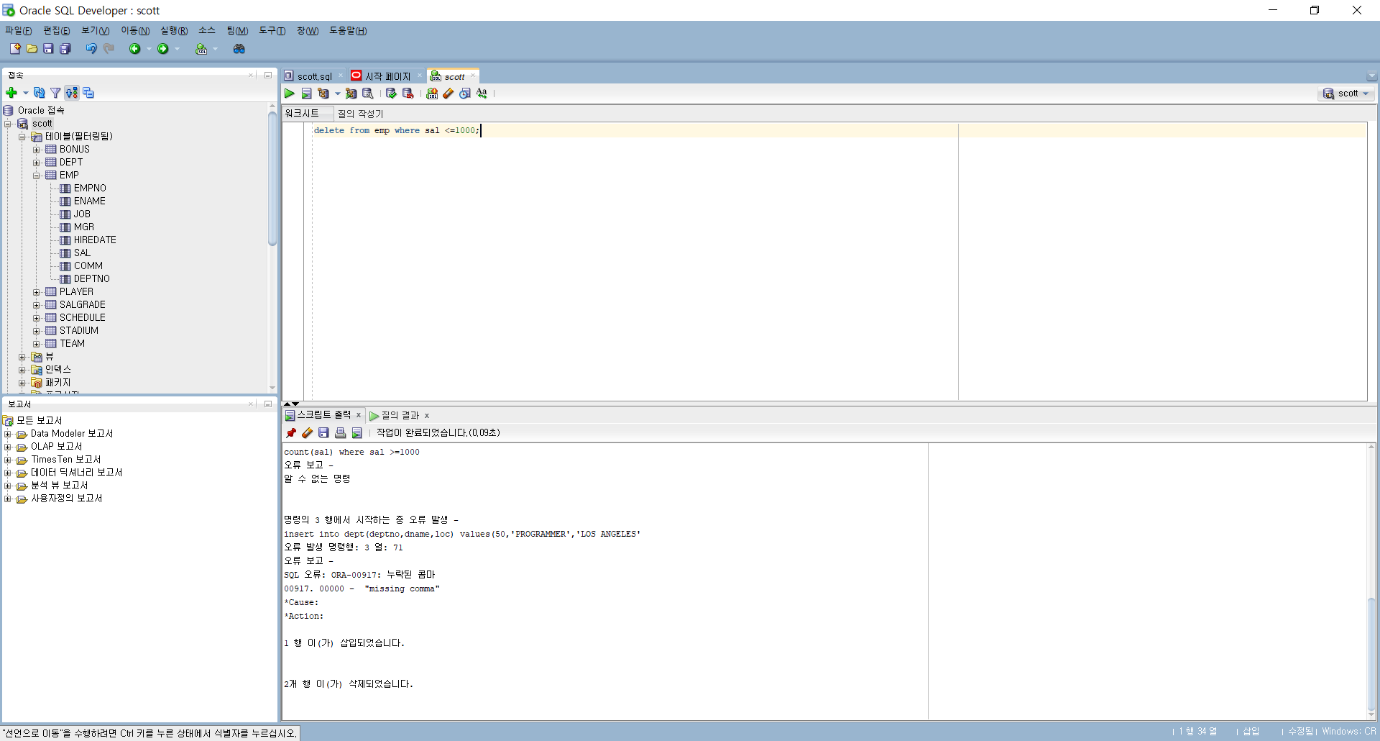
select \* from dept;



1번과 동일하게 풀이

8)

-> delete from emp where sal <=1000;

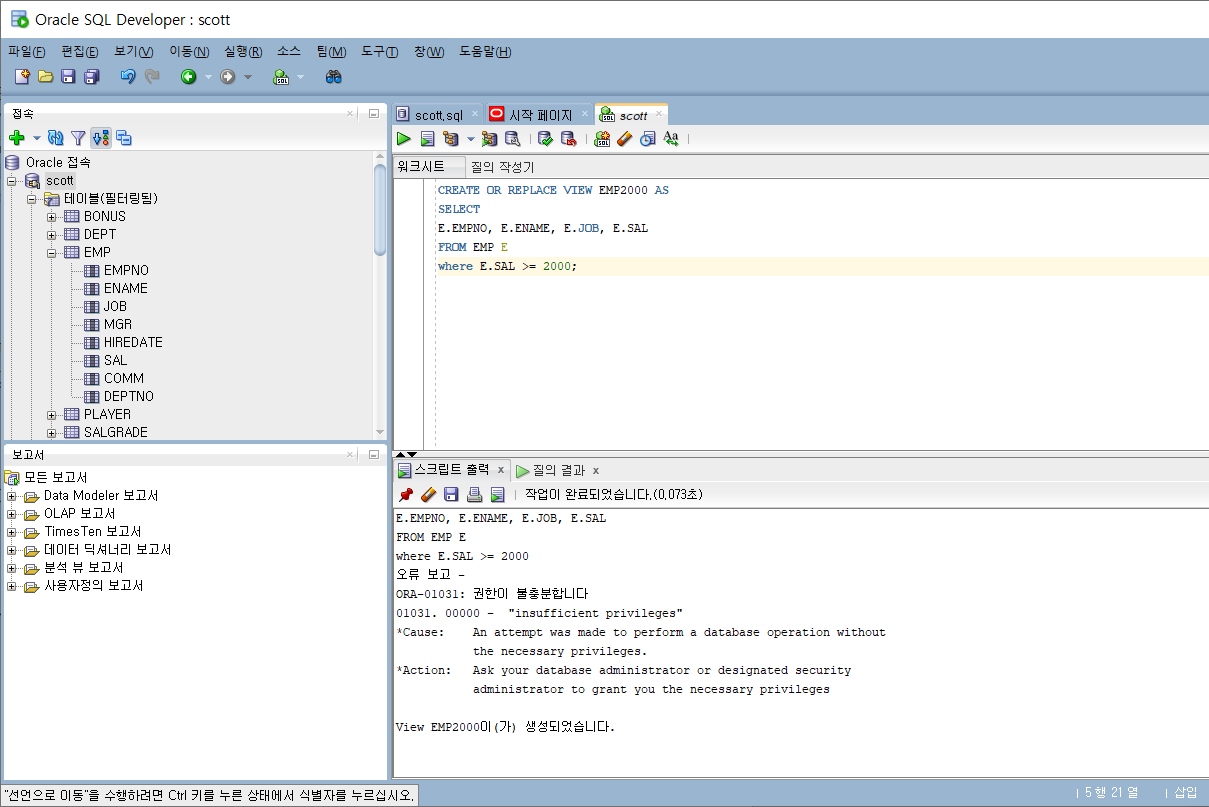


9)

-> CREATE OR REPLACE VIEW EMP2000 AS

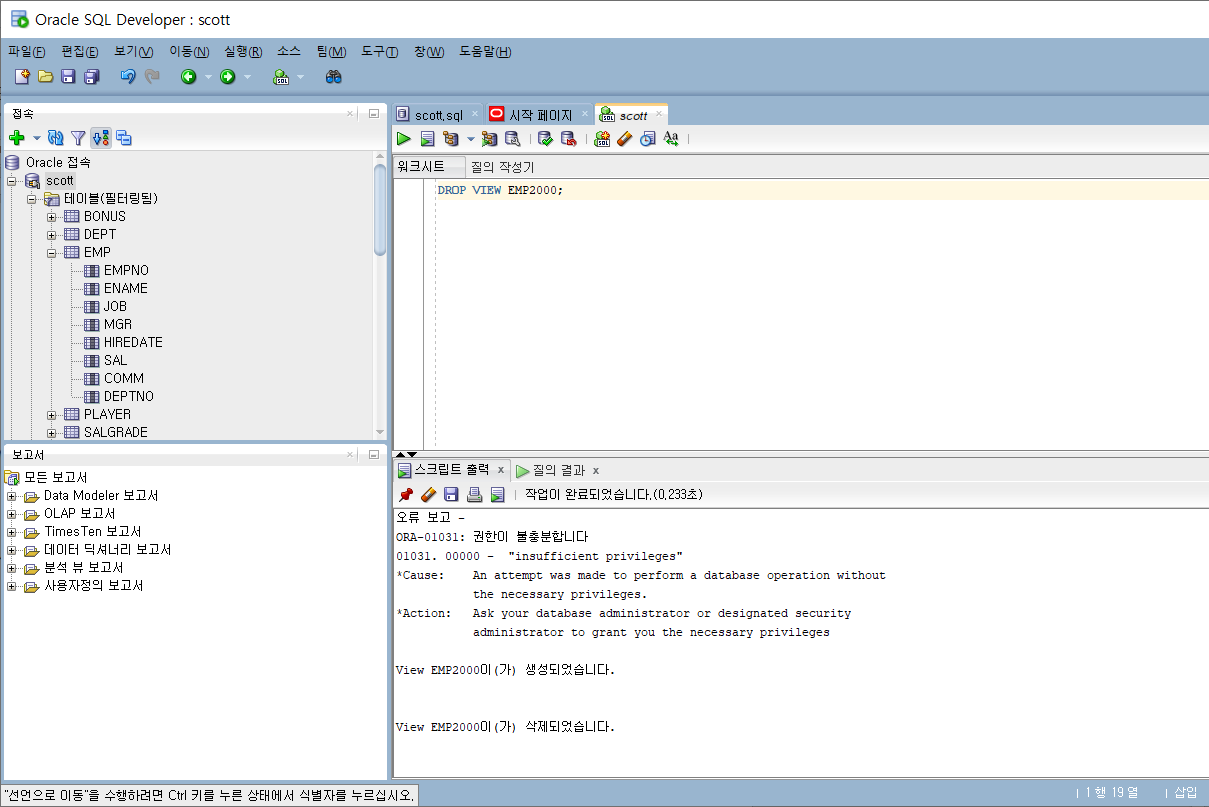
SELECT E.EMPNO, E.ENAME, E.JOB, E.SAL FROM EMP E

where E.SAL >= 2000;



10)

-> DROP VIEW EMP2000;

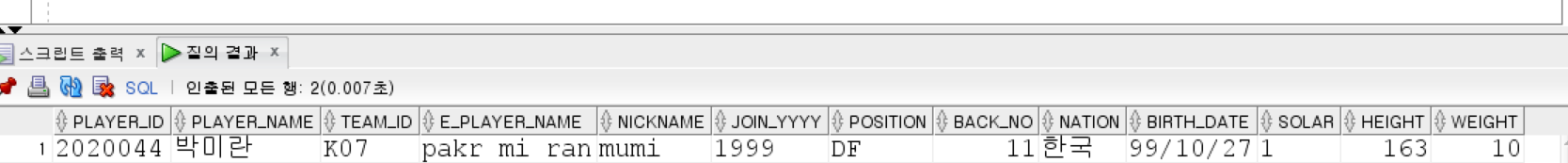


<2번>

1)

-> insert into player(player\_id,player\_name,team\_id,e\_player\_name,nickname,join\_yyyy,position,back\_no,nation,birth\_date,solar,height,weight)

VALUES(2018100,'박미란','K07','pakr mi ran','mumi',1999,'DF',11,'한국','99/10/27',1,163,10);



Insert를 사용해서 컬럼들과 값의 짝을 맞춘다.

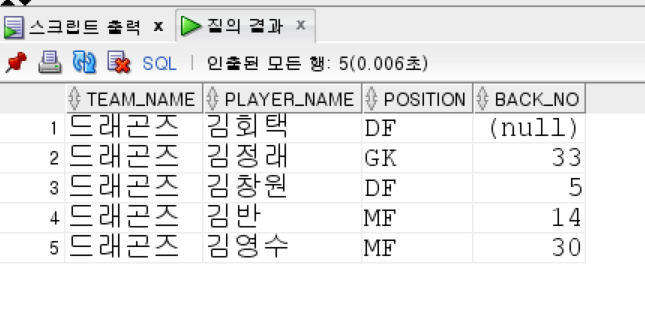
2)

select team.team\_name,player.player\_name,player.position,player.back\_no

from player

join team on team.team\_id=player.team\_id

where team\_name='드래곤즈' and player\_name like '김%';



Join on을 통해서 team과 player테이블을 출력한다.

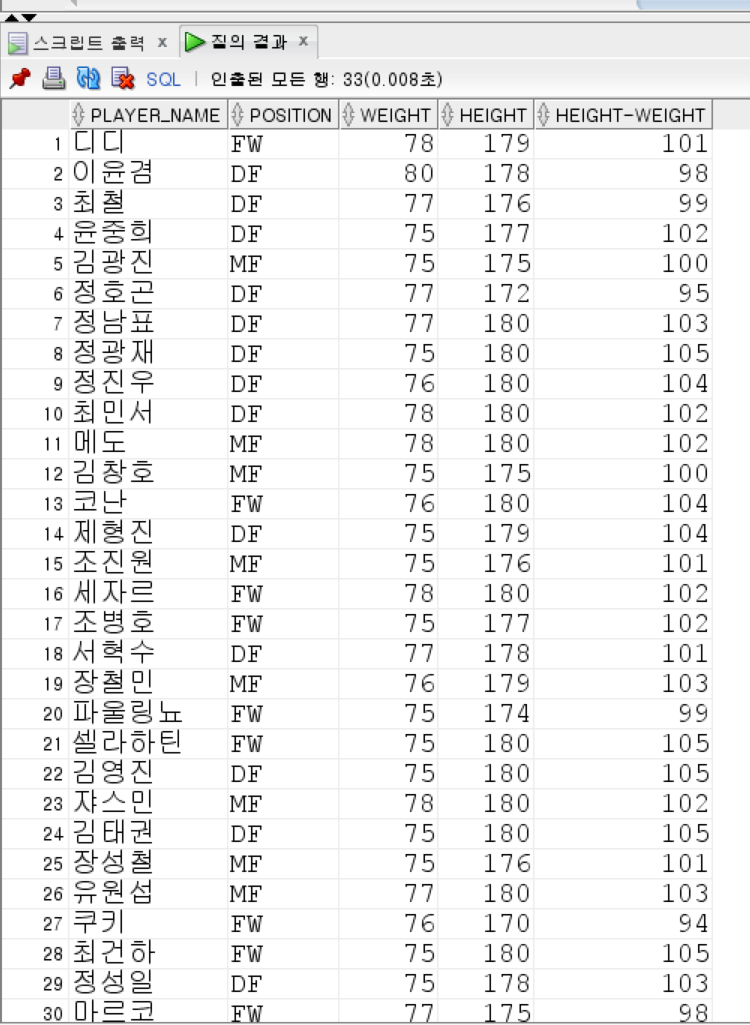
Where like를 사용하여 김씨 성을 가진 선수들을 출력한다.

3)

-> select player\_name, position, weight, height, height-weight

from player

where (weight BETWEEN 75 and 80) and (height BETWEEN 170 and 180);



사이 값의 범위를 지정하고 싶을 때

Where (칼러명 값1 and 값2)를 통해서 지정해준다.

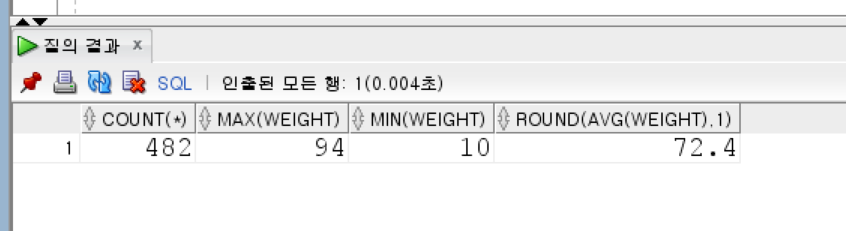
4)

-> select nation,count(\*) from player where nation is not null group by nation;



Group by를 사용할 때 group을 만드는 칼럼을 제외하고는 전부 명시해주어야한다.

5) select count(\*),max(weight),min(weight),round(avg(weight),1) from player;



Round(avg(\*),반올림원하는자릿수)를 사용해서 반올림을 해준다.

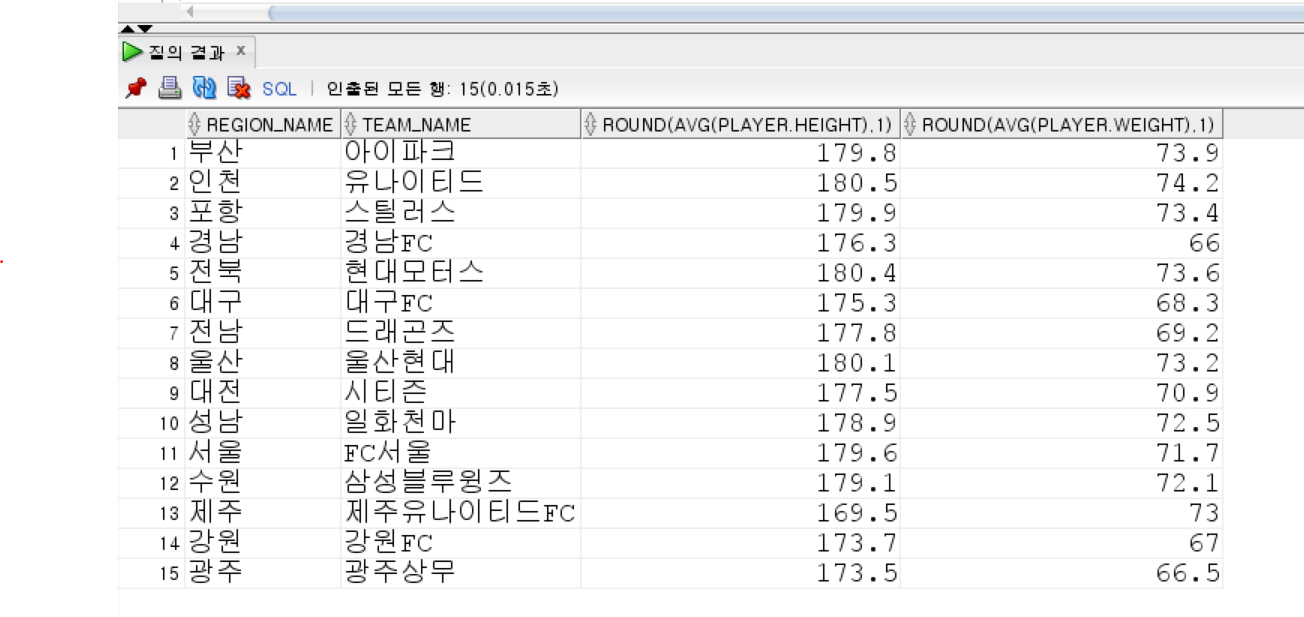
6)

-> select team.region\_name,team.team\_name,round(avg(player.height),1),round(avg(player.weight),1)

from player

join team on team.team\_id=player.team\_id

group by team.region\_name,team.team\_name;



6)과 동일한 방법으로 진행

7)

-> select player.player\_name,player.position, player.back\_no

from player join team on player.team\_id=team.team\_id

where team.team\_name='FC서울'

order by player.player\_name;



Order by를 사용해서 내림차순,오름차순 설정

8)

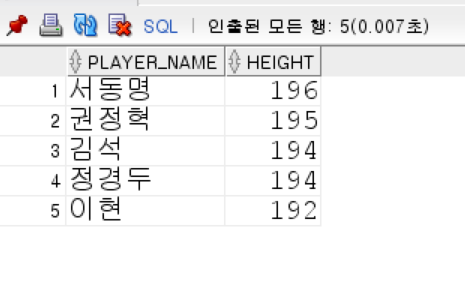
select player\_name,height

from (SELECT \* from player where height is not null order by height desc)

where rownum <=5

order by height desc;

from절에 select를 사용해서 키를 내림차순으로 정렬 후 다시 정렬



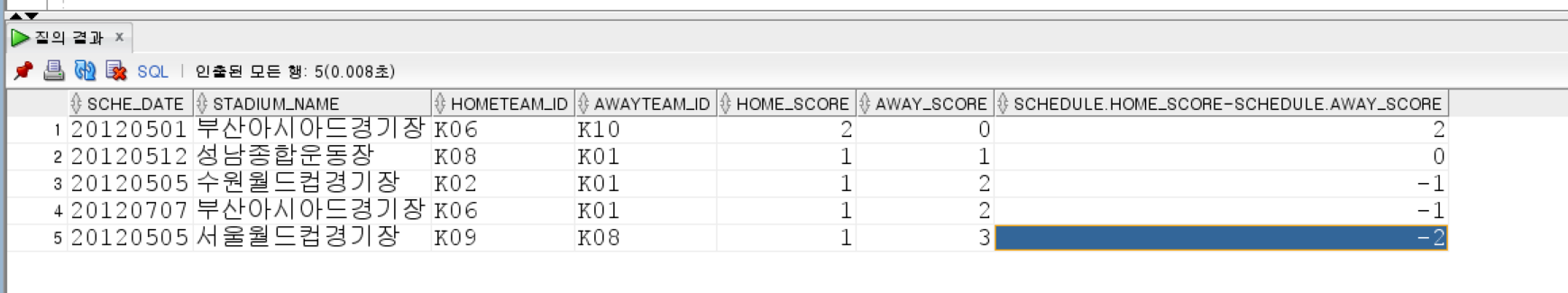
9)

->select schedule.sche\_date,stadium.stadium\_name, stadium.hometeam\_id,schedule.awayteam\_id,schedule.home\_score,schedule.away\_score

,schedule.home\_score-schedule.away\_score

from schedule

join stadium on stadium.stadium\_id=schedule.stadium\_id

where rownum <=5   
order by schedule.away\_score-schedule.home\_score;

10)

-> select nvl(player.nation,'한국'),team.team\_name, player.player\_name

from player

join team on player.team\_id=team.team\_id

where position = 'FW'

order by team\_name,player\_name desc;

왼쪽이 null값 치환했을 경우 오른쪽이 기본상태표시

Nvl(컬럼값,치환값)을 사용하여 null값을 치환

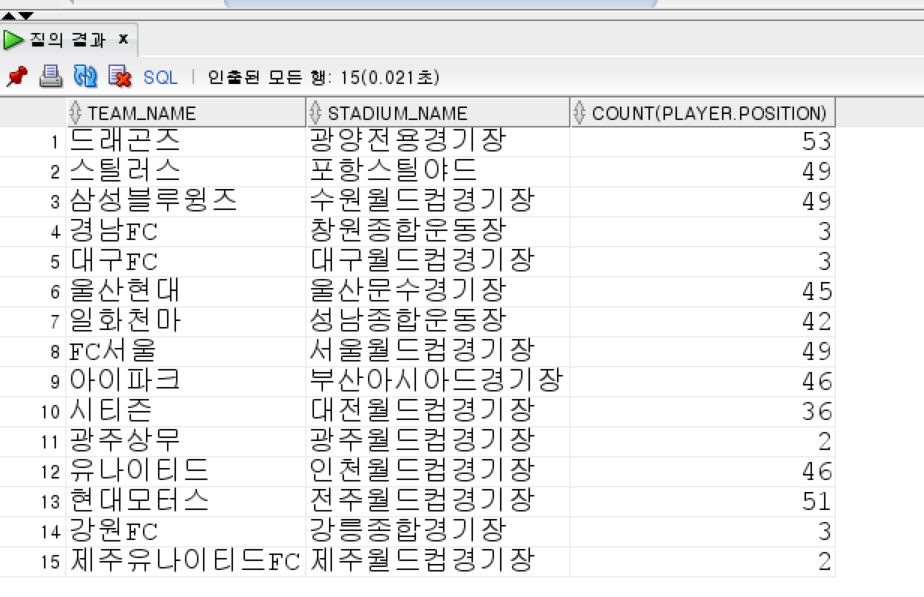
11)

-> select team.team\_name,stadium.stadium\_name,count(player.position)

from player

join team on player.team\_id= team.team\_id

inner join stadium on team.stadium\_id = stadium.stadium\_id

group by team.team\_name,stadium.stadium\_name; 

3개 이상 테이블을 join 할 때 inner join on 을 사용하여 연결.