

## ❖ 데이터 조회 (SELECT 문법)

[실습 1] SELECT 예제 (booksr\_ch4\_departmentemp-ora.sql)

department			employee					
deptno	deptname	floor	empno	empname	title	manager	salary	dno
1	영업	8	1003	조민희	과장	4377	3000000	2
2	기획	10	1365	김상원	사원	3426	1500000	1
3	개발	9	2106	김창섭	대리	1003	2500000	2
4	총무	7	3011	이수민	부장	4377	4000000	3
			3426	박영권	과장	4377	3000000	1
			3427	최종철	사원	3011	1500000	3
			4377	이성래	이사	NULL	5000000	2

  

```

--department(deptno, deptname, floor)
--employee(empno, empname, title, manager, salary, dno)   외래키 : dno -> department(deptno)

--부서(부서번호, 부서명, 위치)
--사원(사번, 사원명, 직책, 관리자, 급여, 부서번호)

--1) 부서의 모든 필드를 검색하라
--2) 부서의 부서번호와 부서이름을 검색하라 (deptno는 부서번호, deptname은 부서이름 으로 검색)
--3) 사원의 직급을 검색하라
--4) 사원의 직급을 한번씩만 검색하라 (중복제거) (DISTINCT)
--5) 2번 부서에 근무하는 직원들에 관한 모든 정보를 검색하라
--6) 이씨 성을 가진 직원들의 이름, 직급, 부서번호를 검색하라

--7) 직급이 과장이면서 1번 부서에서 근무하는 직원들의 이름과 급여를 검색하라
--8) 직급이 과장이면서 1번 부서에 속하지 않은 직원들의 이름과 급여를 검색하라

--9) 급여가 3000000원 이상이고, 4500000원 이하인 직원의 이름, 직급, 급여를 검색하라
--9-1) BETWEEN ... AND 사용
--9-2) 부등호 사용

--10) 1번 부서나 3번 부서에 소속된 직원들에 관한 모든 정보를 검색하라 (IN 사용)

--11) 직급이 과장이면서 직원들에 대하여 이름, 현재급여, 급여가 10% 인상됐을 때의 값을 검색하라
hint) salary * 1.1 "급여 10% 인상액"

--12) 2번 부서에 근무하는 직원들의 급여, 직급, 이름을 검색하여 급여의 오름차순으로 정렬하라

--13) 직원들의 평균 급여와 최대급여를 검색하라
--14) 직원들이 속한 부서번호별로 그룹화하고 부서번호, 평균급여, 최대급여를 검색하라
--15) 직원들이 속한 부서번호별로 그룹화하고, 평균급여가 2500000원 이상인
--부서에 대해서 부서번호, 평균급여, 최대급여를 검색하라

--16-1) 김창섭이 속한 부서번호와 부서의 모든 부서번호를 합집합으로 검색하라 (UNION)
--16-2) 김창섭이 속한 부서번호와 부서의 모든 부서번호를 합집합으로 검색하라 (UNION ALL)

--17) 직원의 이름과 이 직원이 속한 부서이름을 검색하라 (조인)
--18) 직원에 대해서 사원이름과 직속상사의 이름을 검색하라 (Self-Join)
--19) 직원에 대해서 부서이름, 사원이름, 직급, 급여를 검색하라.
--부서이름에 대해서 오름차순, 부서이름이 같을 경우에는 salary 에 대해서 내림차순으로 정렬하라

```

## [Quiz 1] SELECT 예제

```

--Quiz 1-1) 관리자 (manager) 가 없는 직원의 이름, 직책, 급여를 검색하라 (NULL 값)
--Quiz 1-2) 개발 부서의 직원의 이름과 부서이름을 검색하라
--1) SELECT .. FROM .. WHERE 방법
--2) SELECT .. FROM .. ON 방법

```

## [실습 2] 외부조인 (ehan\_ch2\_outerjoin.sql)

ehan_ch2 p68 그림2-32 외부조인 실습		--다음 실행결과를 노트에 작성하라  1) 내부조인(동등조인 : equi join) 2) 내부조인(자연조인 : Natural join)  3) 왼쪽외부조인 4) 오른쪽외부조인 5) 완전외부조인																	
<b>freshman</b>	<table><tr><th>name</th><th>address</th></tr><tr><td>김광식</td><td>서울</td></tr><tr><td>김현정</td><td>대전</td></tr><tr><td>조영수</td><td>대전</td></tr></table>	name	address	김광식	서울	김현정	대전	조영수	대전	<b>member</b>	<table><tr><th>name</th><th>dept_name</th></tr><tr><td>김광식</td><td>컴퓨터공학과</td></tr><tr><td>김현정</td><td>산업공학과</td></tr><tr><td>이진영</td><td>전자공학과</td></tr></table>	name	dept_name	김광식	컴퓨터공학과	김현정	산업공학과	이진영	전자공학과
name	address																		
김광식	서울																		
김현정	대전																		
조영수	대전																		
name	dept_name																		
김광식	컴퓨터공학과																		
김현정	산업공학과																		
이진영	전자공학과																		
--1) 내부조인(동등조인) select * from freshman, member where freshman.name = member.name;  select * from freshman inner join member on freshman.name = member.name;		--2) 내부조인(자연조인) select F.name, address, dept_name from freshman F, member M where F.name = M.name;  select F.name, address, dept_name from freshman F INNER JOIN member M on F.name = M.name;																	
--3) 왼쪽외부조인 select freshman.name, address, member.name, dept_name from freshman left outer join member on freshman.name = member.name;  --(oracle) select * from freshman , member where freshman.name = member.name(+);		--4) 오른쪽외부조인 select freshman.name, address, member.name, dept_name from freshman right outer join member on freshman.name = member.name;  --(oracle) select * from freshman , member where freshman.name (+)= member.name;																	
--5) 완전외부조인 select freshman.name, address, member.name, dept_name from freshman full outer join member on freshman.name = member.name;		--완전외부조인 --(oracle) 주의) 테이블명.필드명 으로 사용시 오류 select name, address, name, dept_name from freshman full outer join member using (name);																	

## [Quiz 2] 외부조인

<b>emp</b>	<b>ename</b>	<b>pno</b>	<b>project</b>	<b>pno</b>	<b>pname</b>	<b>--다음 실행결과를 노트에 작성하라</b>  1) 내부조인(동등조인 : equi join) 2) 내부조인(자연조인 : Natural join)  3) 왼쪽외부조인 4) 오른쪽외부조인 5) 완전외부조인
	홍길동	101		101	작전중	
	임꺽정	102		102	특공대	
	박찬호	101		103	유레카	
	박찬호	103		104	다모여	
	신동엽	NULL				

--2-1) 사원 테이블과 프로젝트 테이블에 대해 왼쪽외부조인 질의를 수행하라						
--2-2) 사원 테이블과 프로젝트 테이블에 대해 오른쪽외부조인 질의를 수행하라						
--2-3) 사원 테이블과 프로젝트 테이블에 대해 완전외부조인 질의를 수행하라						
--2-1) 왼쪽외부조인 select \* from emp LEFT OUTER JOIN project on emp.pno = project.pno;						
--2-2) 오른쪽외부조인 select \* from emp RIGHT OUTER JOIN project on emp.pno = project.pno;						
--2-3) 완전외부조인 select \* from emp FULL OUTER JOIN project on emp.pno = project.pno;						

## [실습 3] 중첩질의

ALL : 하위질의 모든 결과에 대해 조건 수식이 참인 경우 결과를 반환  
 SOME : 하위질의 한 개 이상의 결과에 대해 조건 수식이 참인 경우 결과를 반환  
 ANY : SOME 과 동일

IN 는 =ANY 와 동일, NOT IN 은 <>ALL 과 동일하다

stu	sno	sname	dept	sage	prof	pno	pname	dept	page	--다음 실행결과를 노트에 작성하라	
	s1	유준호	컴퓨터	23		p1	이정무	컴퓨터	36		
	s2	오정민	컴퓨터	34		p2	우태하	컴퓨터	32		
	s3	이태현	건축	22		p3	이성민	건축	45		
	s4	신현주	건축	21							

--stu(sno, sname, dept, sage)

--pro(pno, pname, dept, page)

--> “다음의 빨간색 과 같이 문제를 수정하세요”

--3-1) 교수 테이블에서 이정무 의 학과 와 같은 학생이름, 학과, 나이를 검색하라

--3-2) 교수 테이블에서 pno 가 p1 이거나 p2 인 학과 와 같은 학생이름, 학과, 나이를 검색하라

--3-3) 학생 테이블과 교수 테이블에서 모든 학생들보다 나이가 많은 교수의

--교번, 이름, 나이를 ALL 구문을 이용하여 검색하라

--3-4) 학생 테이블과 교수 테이블에서 한 명 이상 교수보다

--나이가 많은 학생이 있을 경우 학생의 학번, 이름, 나이를 SOME 구문으로 검색하라

--3-5) 가장 나이가 많은 학생의 이름, 나이를 검색하라

--3-6) 각 학과마다 가장 나이가 많은 학생의 이름, 학과, 나이를 검색하라

<답안>

--3-1) 교수 테이블에서 이정무 의 학과 와 같은 학생이름, 학과, 나이를 검색하라

```
select sname, dept, sage
from stu
where dept = (select dept
              from prof
              where pname = '이정무');
```

--3-2) 교수 테이블에서 pno 가 p1 이거나 p2 인 학과 와 같은 학생이름, 학과, 나이를 검색하라

```
select sname, dept, sage
from stu
where dept IN (select dept
              from prof
              where pno IN ('p1', 'p2'));
```

--3-3) 학생 테이블과 교수 테이블에서 모든 학생들보다 나이가 많은 교수의

--교번, 이름, 나이를 ALL 구문을 이용하여 검색하라

```
select pno, pname, page
from prof
where page > ALL (select sage
                 from stu);
```

--3-4) 학생 테이블과 교수 테이블에서 한 명 이상 교수보다

--나이가 많은 학생이 있을 경우 학생의 학번, 이름, 나이를 SOME 구문으로 검색하라

```
select sno, sname, sage
from stu
where sage > SOME (select page
                  from prof);
```

--3-5) 가장 나이가 많은 학생 이름, 나이를 검색하라

```
select sname, dept, sage
from stu
where sage = (select MAX(sage)
             from stu);
```

--3-6) 각 학과마다 가장 나이가 많은 학생 이름, 학과, 나이를 검색하라

```
select sname, dept, sage
from stu S1
where sage IN (select MAX(sage)
              from stu S2
              where S1.dept = S2.dept);
```

## [Quiz 3] 중첩질의

--department( <u>deptno</u> , deptname, floor)
--employee( <u>empno</u> , empname, title, manager, salary, <i>dno</i> )
--중첩질의
--20) 박영권과 같은 직급을 갖는 모든 직원들의 이름과 직급을 검색하라
--21) 영업부나 개발부에 근무하는 직원들의 이름을 검색하라 (IN 사용)
--22) 영업부나 개발부에 근무하는 직원들의 이름을 검색하라 (EXISTS 사용)
--23) 영업부나 개발부에 근무하는 직원들의 이름을 검색하라 (조인 사용)
--24) 직원들이 한 명도 소속되지 않은 부서명을 검색하라 (NOT EXISTS 사용)
--25) 자신이 속한 부서의 직원들의 평균급여보다 많은 급여를 받는 직원들에 대해서
--이름, 부서번호, 급여를 검색하라 (상관중첩질의)

&lt;답안&gt;

--20) 박영권과 같은 직급을 갖는 모든 직원들의 이름과 직급을 검색하라
--21) 영업부나 개발부에 근무하는 직원들의 이름을 검색하라 (IN 사용)
--22) 영업부나 개발부에 근무하는 직원들의 이름을 검색하라 (EXISTS 사용) select empname from employee E where EXISTS (select * from department D where (deptname = '영업' or deptname = '개발') and E.dno = D.deptno);
--23) 영업부나 개발부에 근무하는 직원들의 이름을 검색하라 (조인 사용) select empname from employee E, department D where E.dno = D.deptno and deptname IN ('영업', '개발');
--24) 직원들이 한 명도 소속되지 않은 부서명을 검색하라 (NOT EXISTS 사용) select deptname from department D where NOT EXISTS (select * from employee E where D.deptno = E.dno);
--상관중첩질의 --25) 자신이 속한 부서의 직원들의 평균급여보다 많은 급여를 받는 직원들에 대해서 --이름, 부서번호, 급여를 검색하라 (상관중첩질의) select empname, dno, salary from employee E where salary > (select AVG(salary) from employee where E.dno = employee.dno);

## [실습 4] 함수

급여	직원번호	직원명	주민등록번호	소속부서	직급	성별	기본급	수당
	1111	홍길동	801211-111111	AA	과장	남	2000000	100000
	2222	임꺽정	850505-222222	AA	사원	여	1200000	300000
	3333	이병진	830301-155555	CC	사원	남	1200000	NULL
	4444	우현정	891004-266666	CC	사원	여	1150000	NULL
	5555	차두리	821004-133333	AA	사원	남	1150000	300000
	6666	송종국	730301-144444	BB	부장	남	2500000	50000

--4-1) 직원명, 기본급, 수당, 기본급+수당 을 검색하라 : NVL() 함수  
 --4-2) 수당 을 검색하는데 앞자리에 \* 를 표시하라 : LPAD() 함수  
 --4-3) 주민등록번호 뒷자리 7자리를 \* 로 표시하여 검색하라 : REPLACE(), SUBSTR() 함수

&lt;답안&gt;

<p>--4-1) 직원명, 기본급, 수당, 기본급+수당 을 검색하라 : NVL() 함수</p> <pre>select 직원명, 기본급, 수당, 기본급 + NVL(수당, 0) "급여액" from 급여;</pre> <p>select 직원명, 기본급, 수당, NVL(기본급 + 수당, 기본급) "급여액" from 급여;</p>
<p>/* 설명 : LPAD() 함수          부서명 필드값 에서 10자리를 출력하는데 해당 데이터값을 제외하고          나머지는 * 로 왼쪽기준으로 10자리수만큼 채운다          */</p> <pre>select 직원명, LPAD(수당, 10, '*') "수당" from 급여 where 수당 IS NOT NULL;</pre>
<p>/* 설명 : REPLACE() 함수, SUBSTR() 함수          주민번호 필드값 에서 8번째 부터 7자리를 * 로 치환하여 출력한다          */</p> <pre>select REPLACE(주민등록번호, SUBSTR(주민등록번호, 8, 7), '*****') "주민등록번호" from 급여;</pre>

## \* 순위함수

<p>--RANK() 함수 사용</p> <pre>select 직원명, 직급, 기본급,        RANK() over (order by 기본급 DESC) rank from 급여;</pre>
<p>--PARTITION : 그룹별 순위</p> <pre>select 직원명, 직급, 기본급,        RANK() over (PARTITION by 직급 order by 기본급 DESC) rank from 급여;</pre>
<p>--ROW_NUMBER() : 정렬된 결과에 순위를 부여하는 기능(동일한 경우에도 특별한 순서를 지정하지 않음)</p> <pre>select ROW_NUMBER() over (order by 기본급 DESC) rank,        직원명, 직급, 기본급 from 급여;</pre>

## [Quiz 4] 뷰

<p>--Quiz 4-1) 직원번호, 직원명, 직급, 성별로 이루어진 직원_뷰 를 생성하라</p>
<p>--Quiz 4-2) 직원_뷰 를 검색하라</p>