

① 오라클 설치 및 셋팅사항

1. 오라클 설치

- 오라클은 Edition별로 Oracle Database 10g Express Edition , Enterprise/Standard Edition 으로 나뉘며 먼저 express edition 으로 실습한 이후 Admin 부분에서 Enterprise/Standard Edition 으로 실습할 계획임.
- <http://www.oracle.com/kr/> 또는 <ftp://211.242.212.12/네트워크> id : kbst , pass : kbst2008 에서 오라클 express edition을 다운로드 하여 설치한다.
- 오라클 원격관리 tool인 SQL developer도 다운받는다.

2. SCOTT계정 생성

- express edition은 예제를 실습할 수 있는 SCOTT계정이 없으므로 생성시킨다.(SQL developer 이용)
- orther users > user name : scott > Roles : connect , resource 체크

3. emp,dept, bonus 테이블 복원

- ftp에서 다운로드 받은 후 SQL developer에서 파일을 열어 Run Script.

※ Q) system계정 암호를 잊어버린 경우.

A) 명령프롬프트 > cd.. 으로 root까지 이동.

1) 명령어 : sqlplus /nolog

2) 명령어 : conn /as sysdba

3) 명령어 : alter user system identified by 변경할비번 ;

4) 바뀐 비밀번호로 재접속한다.

② database를 하기 전 꼭 알아야 할 사항.

☞ 핵심적으로 알아야 할 데이터베이스의 학습의 범위

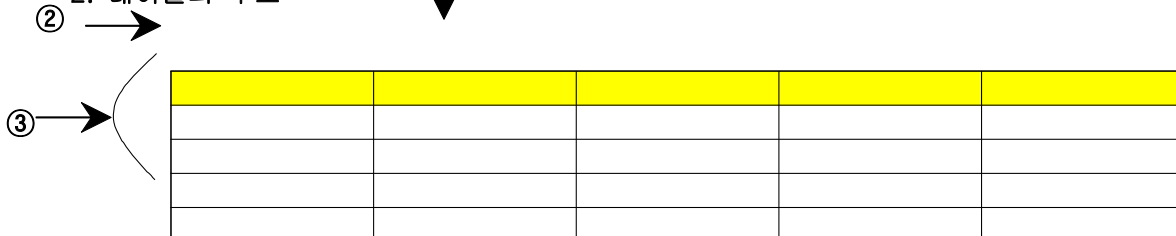
SQL	가장 기본이 되는 부분으로 기본SQL문 및 응용SQL문 반드시 숙지하여야 함.
데이터베이스관리	백업 및 복원, 사용자 관리등
데이터베이스모델링	데이터베이스설계,정규화, 모델링용어 및 ER-WIN Tool 등

☞ SQL문을 하기전 알아야 할 사항.

1. Table

데이터베이스의 여러개체중 **①**에 대해서 데이터가 저장되는 곳을 말한다.

2. 테이블의 구조



① 필드(field) : 열, 컬럼(column) - 속성(attribute),테이블에 포함된 필드의 개수를 차수(degree)라 함.

② 필드명 : 열이름, 컬럼명

③ 레코드(record) : 행, Row - 튜플(tuple),테이블에 포함된 레코드의 개수를

카디널리티(cardinality)라 함.

3. 기본 오라클 관리자 계정

- 1) SYS : 오라클 슈퍼사용자명, 오라클에서 사용할수 있는 모든 권한을 가짐
- 2) SYSTEM : SYS와의 차이점은 데이터베이스를 생성할 수 있는 권한이 없다.(나머지는 동일)
- 3) SCOTT : 오라클에서 기본으로 제공하는 테이블등으로 SQL문을 연습하기 위하여 만든 계정.

4. 데이터 타입(DATA Type)

- 1) Number 데이터 형 : 숫자데이터를 저장하기 위해서 제공
- 2) DATE 데이터 형 : 날짜 및 시간 데이터를 저장하기 위한 데이터 형
- 3) VARCHAR2 데이터형 : 문자를 저장하기 위한 데이터형(CHAR와의 차이점숙지)

5. SQL(Structured Query Language)의 종류

- 1) DML(Data Manipulation Language) - 데이터 조작어

데이터베이스에 저장된 데이터를 조작하기 위해 사용하는 언어

ex) Select(검색), Insert(삽입), Update(갱신), delete(삭제)

2) DDL(Data Definition Language) - 데이터 정의어

객체 생성 및 변경시 사용하는 언어

ex) Create, Alter, Drop

3) DCL(Data Control Language) - 데이터 제어어

권한 부여등 데이터베이스 시스템을 관리하기 위해 사용하는 언어

ex) grant, revoke

③ SQL의 기본

EX) 아래와 같은 테이블을 만든 후 데이터를 삽입한다.

학생

학번	이름	학년	신장	소속
08001	홍길동	2	170	CD
07002	김길동	4	168	CD
07012	박길동	2	180	CD
07033	이길동	3	175	ID
05014	최길동	4	183	ED
06032	정길동	3	165	ED

수강

학번	번호	성적
08001	101	95
08001	102	80
08001	103	60
08001	104	80
08001	105	85
07002	101	80
07002	104	95
07012	102	70
07012	103	80
07033	102	85
05014	104	90

설강

05014	105	75
-------	-----	----

👉 Create문

문법 :

Create table 테이블명(Field명01 DataType, Field명02 DataType, ~ Field명[n] DataType);

ex1) create table 학생 (학번 char(5) not null, 이름 char(20) , 학년 char(5),
신장 number(3), 소속 char(10));

ex2) create table 수강 (학번 char(5) not null, 번호 char(3) not null, 성적 number(3));

ex3) create table 설강 (번호 char(3) not null, 교수명 char(10) not null,
과목명 char(5), 소속 char(10));

※ 현재 DB에 있는 테이블 보기 : select * from tab;

테이블의 구조 보기 : desc 학생;

번호	교수명	과목명	소속
101	김일삼	SE	CD
102	권오영	DB	CD
103	김소월	DG	MD
104	이상	FP	CD
105	이석영	CS	ED

memo

👉 Insert문 : 데이터를 삽입하기 위한 SQL문

문법 :

INSERT INTO Table명(Field01명,Field02명, ~ , Field n명)

VALUES(Field 01의 값, Field 02의 값, ~ , Field n의 값)

※ 값을 적을때 문자형 및 날짜형 datatype은 양쪽에 ‘(홀따옴표)를 붙이고 입력한다.

ex) insert into 학생(학번,이름,학년,신장,소속) values('08001','홍길동','2',170,'CD');

Test) insert문에 입각하여 학생,수강,설강테이블의 데이터를 모두 입력하시오.

※ 미리 알아야 할 SQL-plus 몇가지 환경설정 및 명령어.

- **linesize 변수** : 출력되는 라인의 길이를 늘여줘야 보기에 편하다.
set linesize 120
- **pagesize 변수** : 출력되는 로우의 개수가 기본 7개이므로 50개로 늘린다.
set pazesize 50
- linesize 및 pagesize는 변수의 개념이므로 재 접속시 default로 되어버림.
고정시키는 방법 :C:\Woracle\Wproduct\10.2.0\Wdb_1\Wsqlplus\admin\Wglogin.sql 파일을 열어
하단에 명령어 입력후 저장
set pazesize 50
set linesize 120
- **LIST , L** : 버퍼에 있는 내용을 나타내기 위한 명령어.
- **/ , RUN** : 버퍼에 저장된 쿼리문을 실행시키기 위한 명령어.
- **EDIT , ED** : 버퍼에 저장된 쿼리문을 편집하기 위한 명령어.
- **SAVE** : 마지막에 실행한 쿼리문을 저장하는 명령어
저장 : sql> save d:\Wsql\W0602;
실행 : sql>@d:\Wsql\W0602;
- **GET** : save로 저장된 파일의 sql문을 불러오는 명령어

👉 **Update문** : 데이터를 수정하기 위한 SQL문

문법 :

Update Table명
SET Field 01=변경할 값, Field 02=변경할 값, ~ , Field n=변경할 값)
Where 조건지정

ex) Update 학생 set 이름='홍길남' where 학번='08001'

👉 **Delete문** : 데이터를 삭제하기 위한 SQL문

문법 : Delete from Table명 Where 조건지정
※삭제는 Record단위로 삭제됨

ex) delete from 학생 where 학번='08001'

👉 **기본 Select문** : 데이터를 삽입하기 위한 SQL문

문법 : select Field01명,Field02명, ~ , Field n명 from table명
 [where 조건절]
 [order by 필드명 asc/desc]

※ 모든 필드를 출력할때는 필드명자리에 *를 적는다.

● where절에 사용되는 SQL 연산자

연산자	설명
and	2개이상의 조건을 논리곱으로 연결
or	2개이상의 조건을 논리합으로 연결
=, >, >=, <, <=	같다, 초과, 이상, 미만, 이하
<>, !=, ^=	not equal to
BETWEEN m AND n	m과 n사이의 값(m이상,n이하)-m과 n의 값 포함
Not BETWEEN m AND n	m과 n사이의 값을 제외한 범위- m과 n의 값 포함안됨.
IN(list)	list의 값 중 어느 하나와 일치하는 데이터를 출력(or의 의미)
NOT IN(list)	list의 값과 일치하지 않는 데이터를 출력
LIKE	문자배열 비교, 주로 와일드문자와 함께 쓰인다.(%,_) % : 자리수 제한없음, _ : 자리수 제한
Not Like	Like과 반대의 의미
is null	null값을 가진 데이터를 출력
is not null	null이 아닌 데이터를 출력

ex1) 학생 table에서 학생의 이름, 신장을 출력하시오.

ex2) 학생 table에서 학번이 08001인 학생의 학번,이름,소속을 출력하시오.

ex3) 학생 table에서 학년이 3학년이면서 소속이 ID인 학생의 모든정보를 구하시오.

ex4) 학생 table에서 신장이 170이상인 학생의 이름,신장,소속을 구하시오.

ex5) 수강테이블에서 성적이 80이상 90이하인 학생의 학번과 성적을 구하시오.

1)

2)

ex6) 학생테이블에서 소속이 CD가 아닌 학생의 모든정보를 구하시오.

ex7) 설강테이블에서 교수명이 김으로 시작하는 정보의 교수명,과목명을 구하시오.

ex8) 설강테이블에서 교수명이 이로 시작하면서 두자리인 교수명과 과목명을 구하시오.

ex9) 설강테이블에서 교수명이 김씨가 아닌 모든 정보를 검색하시오.

ex10) 소속이 CD인 학생중 신장이 170이상 되는 학생의 학년과 신장을 구하시오.

ex11) 학생테이블을 신장이 큰순에서 작은순서 대로 정렬하시오.

ex12) 소속이 CD인 학생을 신장 크기 순으로 구하시오.

ex13) 학생테이블에서 소속이 ID또는 ED가 아닌 학생의 모든 정보를 구하시오.

※ 아래의 문제는 오라클에서 기본으로 제공하는 예제테이블인 emp, dept, salgrade를 이용하여 검색하시오.

ex14) 사원테이블에서 연봉 2500이상인 직원의 사원번호,이름,연봉을 구하시오.

ex15) 성과급을 받지 않는 사원의 이름과 연봉을 구하시오.

ex16) 입사일이 82년 이후인 사원의 모든정보를 구하시오.

ex17) 연봉이 2000이상 또는 job이 salesman인 사원의 모든정보를 구하시오.

ex18) 사원의 이름중 A가 들어가있는 사원의 모든정보를 구하시오.

ex19) 연봉이 높은 순서대로 정렬하시오.

ex20) 부서번호가 10또는 20인 사원의 이름과 연봉을 구하시오

1)

2)

Memo

④ 중급 SQL문

그룹(통계)함수 : count,sum,avg,max,min

- SUM : 해당열의 합계를 구하는 함수
- AVG : 해당열의 평균을 구하는 함수
- MAX : 해당열의 최대값을 구하는 함수
- MIN : 해당열의 최소값을 구하는 함수
- COUNT : row의 개수를 카운트하는 함수

- Group By 절 : 필드단위로 같은 Data끼리 그룹을 지을때 사용.
- Having : Group By 절에서의 조건절이다.(Group By 절에서는 where 대신 having을 사용)

※ Where 과 Having의 실행방법의 차이

where : 조건을 먼저 검색한 후 해당조건에 맞는 결과가 select 된다.

having : select가 먼저 된 후 그 결과에서 조건에 맞는 데이터만 남는다.

그룹함수를 적용한 컬럼만이 조건으로 올 수 있다.

<문법적 고려사항>

Select절에서의 계산함수가 아닌 컬럼의 리스트와 group by 절에서의 컬럼 리스트는 같아야함.
하지만 having절에서의 컬럼리스트는 select 리스트에 있는 모든 컬럼을 사용할수 있다.

EX) select deptno,avg(sal) from emp group by deptno; (O)
select deptno,ename,avg(sal) from emp group by deptno; (X)

ex1) 학생들의 평균 신장을 구하라.

ex2) 소속별로 학생들의 평균 신장을 구하시오.

ex3) 수강테이블에서 학생별 평균성적을 구한후 평균성적이 높은 순서대로 정렬하시오.

ex4) 교수번호 102인 교수가 설강한 과목을 수강하는 학생의 수를 계산하라.

SQL> select count(distinct 학번) from 수강 where 번호 = '102' ;

※ dintinct : 중복된 튜플을 제거하는 키워드.

ex5) 세과목 이상 수강하는 학생의 학번을 구하시오.

SQL> select 학번 from 수강 group by 학번 having count(*) >= 3 ;

※ 아래의 문제는 오라클에서 기본으로 제공하는 예제테이블인 emp, dept, salgrade를 이용하여 검색하시오. 0

ex6) 가장 높은 급여와 가장 낮은 급여를 구하시오.

ex7) 커미션을 받는 사원의 수를 구하시오.

ex8) 부서번호가 10인 사원중에 커미션을 받는 사원의 수를 구하시오.

ex9) 사원테이블에서 job의 종류는 몇 개인가?

ex10) 부서별로 급여의 최고액을 구한 후 급여가 많은 순서대로 정렬하시오.

ex11) 부서별 평균 급여가 2000이상인 부서들만 부서별 평균을 구하시오.

- select deptno, avg(sal) from emp group by deptno having avg(sal) >= 2000;

ex12) 급여가 1000이상인 사원들만 부서별 평균급여를 구한 후 부서별 평균 급여가 2000이상인 부서만 부서번호와 부서별 평균급여를 구하시오.

- select deptno, avg(sal) from emp where sal>=1000
group by deptno having avg(sal) >= 2000;

※ 오라클 예제 테이블

SQL> select * from emp;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7566	JONES	MANAGER	7839	81/04/02	2975		20
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7839	KING	PRESIDENT		81/11/17	5000		10
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7876	ADAMS	CLERK	7788	87/05/23	1100		20
7900	JAMES	CLERK	7698	81/12/03	950		30
7902	FORD	ANALYST	7566	81/12/03	3000		20
7934	MILLER	CLERK	7782	82/01/23	1300		10

SQL> select * from dept;

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

SQL> select * from salgrade;

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

JOIN(조인)

데이터를 검색하다 보면 테이블 하나만을 이용해서는 원하는 정보를 다 가져오지 못하는 경우가 많습니다. 예를 들어 사원의 급여 정보를 검색하는 경우에 [급여] 테이블에는 사원 번호와 급여 정보만 있는 경우 사원 이름을 같이 검색하고자 한다면 사원 이름을 가지고 있는 [사원] 테이블을 참조해야 합니다. 이렇게 테이블을 서로 연결하여 검색을 할 때 사용되는 것이 JOIN 문입니다. JOIN의 종류는 다음과 같이 나눌 수 있습니다.

Ansi-SQL	오라클 SQL
<p>● inner join</p> <p>설명) 양쪽의 행들 중에서 서로 연관된 내용만 검색하는 조인 방법입니다. 두 테이블에서 공통되지 못한 부분은 JOIN의 대상에서 제외가 됨. ex) SELECT A.사번, A.이름, B.도서 FROM 사원 A INNER JOIN 대출 B ON A.사번 = B.사번</p>	<p>● EQUI join</p> <p>설명) 양쪽의 행들 중에서 서로 연관된 내용만 검색하는 조인 방법입니다. 두 테이블에서 공통되지 못한 부분은 JOIN의 대상에서 제외가 됨. ex) select A.사번, A.이름,B.도서 from 사원 A, 대출 B where A.사번 = B. 사번</p>
<p>● Left Join</p> <p>설명) 왼쪽 테이블은 연관된 내용이 없더라도 검색이 되는 조인방법이다.(먼저 정의된 테이블이 왼쪽 테이블임) ex) SELECT A.사번, A.이름, B.도서 FROM 사원 A LEFT OUTER JOIN 대출 B ON A.사번 = B.사번</p>	<p>● Outer join</p> <p>설명) 왼쪽,오른쪽 테이블에서 연관된 내용이 없더라도 검색이 되는 조인방법이다. “(+)”연산자의 위치에 따라 left join과 right join이 구분됨. ex) select A.사번, A.이름, B.도서 from 사원 A , 대출 B where A.사번 = B.사번(+) (left join과 같은 결과) ex) select A.사번, A.이름, B.도서 from 사원 A , 대출 B where A.사번(+) = B.사번 (right join과 같은 결과)</p>
<p>● Right Join</p> <p>설명) 오른쪽 테이블은 연관된 내용이 없더라도 검색이 되는 조인방법이다. ex) SELECT A.사번, A.이름, B.도서 FROM 사원 A Right OUTER JOIN 대출 B ON A.사번 = B.사번</p>	<p>● NON-EQUI join</p> <p>설명) 특정범위내에 있는 데이터를 조인시킬때 “=” 대신 비교 연산자를 사용하는 조인방법 ex) select e.ename, e.sal, s.grade from emp e, salgrade s where e.sal >= s.losal and e.sal <= s.hisal</p>
<p>- 해당사항없음.</p>	<p>● Self join</p> <p>설명) 같은 테이블 내에서 서로다른 필드에 공통된 데이터가 있을 때 테이블의 별명을 따로 주어서 조인시키는 방법. ex) select employee.ename '의 매니저는 ' manager.ename '입니다.' from emp employee, emp manager where employee.mgr = manager.empno;</p>

<실습1>

● 예제 테이블 작성

사원	
사번	이름
A0001	홍길동
A0002	김치국
A0003	안경태
A0004	박호준
A0005	한국인
A0006	이호빵

대출	
사번	도서
A0001	T0301
A0002	T0302
A0001	T0203
A0003	T0401
A0005	T0501
A0001	T0201
A0003	T0204
A0003	T0306
A0008	T0404

```
sql> create table 사원 (사번 varchar2(5), 이름 varchar2(6));  
sql > create table 대출 (사번 varchar2(5), 도서 varchar2(5));
```

ex) 두 테이블에 각각의 데이터를 삽입한 후 위의 표에 있는 예제들을 Ansi-SQL 및 오라클 SQL의 문법 및 조인의 종류의 의미를 생각하며 실행하여 본다.

<실습2> - 학생,수강,설강 테이블 이용.

Q1) 이길동 학생이 수강한 교수명과 성적을 구하시오.

```
select 교수명,성적 from 학생,수강,설강  
where 학생.이름 = '이길동' and 학생.학번 = 수강.학번 and 수강.번호 = 설강.번호
```

Q2) 최씨성을 갖는 학생의 모든정보를 구하라.

Q3) 최길동 학생의 학년,이름,과목번호,성적을 구하시오.

<실습3> - emp, dept, slagrade 테이블 이용

Q1) 각각의 사원의 연봉에 따른 등급을 출력하시오.(사원이름,연봉,등급순으로 출력)

Q2) 사원들의 job과 부서명을 구하시오. (이름,job, 부서명순으로 출력)

Q3) 커미션을 받는 사원의 이름, 커미션, 부서이름 및 부서위치를 출력하시오.

Q4) 사원이름,부서번호와 해당산원과 같은 부서에서 근무하는 사원을 출력하시오.

⑤ 고급 SQL문

☞ Sub Query : select,insert,update,delete에 중첩될수 있는 select문을 말한다.

● Where 절에 존재하는 서브쿼리의 유형

1. 단일값 출력 서브쿼리. (비교 연산자를 사용한다.)

ex1) 5과목을 수강하는 학생의 이름을 구하는 경우.

```
select 이름 from 학생
where 학번 = (select 학번 from 수강 group by 학번 having count(학번)=5)
```

ex2) 교수번호 102를 수강하는 학생들중 평균성적보다 높은 학생의 학번을 구하는 경우

```
select 학번 from 수강
where 성적 >= (select avg(성적) from 수강 where 번호=102) and 번호=102
```

2. 하나이상의 값을 출력하는 서브쿼리.(in,any,all,exists 연산자 사용)

● in : sub query의 결과가 하나라도 만족하면 참.

ex) 과목번호 102를 수강하는 학생의 이름을 구하라.

```
select 이름 from 학생 where 학번 in (select 학번 from 수강 where 번호 = '102');
```

● any : 주어진 상황에서 가장 자유로운 데이터.(가장작은값보다 크고 가장큰값보다 작아야 한다)

ex) 부서번호가 30번인 직원들의 최저연봉보다 많이 받는 직원의 이름,연봉을 구하라.

```
select ename,sal from emp where sal > any(select sal from emp where deptno=30);
```

● all : 주어진 상황에서 가장 제한적인 데이터.(가장작은값보다 작고 가장큰값보다 커야 한다.)

ex) 부서번호가 30번인 직원들의 최고연봉보다 많이 받는 직원의 이름,연봉을 구하라.

```
select ename,sal from emp where sal > all(select sal from emp where deptno=30);
```

● exists : 서브쿼리안에 값이 어떤것이든 상관없이 결과만 돌려주면 됨.

ex) 과목번호 101을 수강하는 학생의 이름을 구하라.

```
select 이름 from 학생 where
exists(select 학번 from 수강 where 번호='101' and 학생.학번=수강.학번)
```

<문법적 고려사항>

exists 키워드 앞에는 컬럼이나 상수, 표현식등이 올수 없다.(where 다음 바로 exists)

★ 지난시간 QnA

Q1) 자릿수 제한을 두는 경우의 와일드카드문자인 “_”사용했을때 검색이 안된 문제.

A) 문제가 발생한 기본원인은 Data Type을 char로 주어서 공백마저 문자 길이로 인식하여서 일어나는 문제였으며 length함수를 사용하여 글자길이를 반환시킬때는 한글을 제대로 인식하지 못하여 공백만이 글자길이에 포함되었던 현상으로써 아래와 같이 해결.

```
select 번호,교수명,과목명 from 설강 where trim(교수명) like '이_'
```

- 공백을 제거하는 함수인 trim(기본 SQL이 끝난후 추후 함수에 관하여 배움)을 사용하여 교수명 필드의 공백을 제거한 후 검색

Q2) drop table 명령어로 테이블로 삭제한 후 tab테이블에 잔재가 남아 있는 경우

A) purge recyclebin; 명령어 실행으로 해결(휴지통을 모두 버림)

<실습1>

학생,수강,설강 테이블 이용.

Q1) 번호가 102인 교수로부터 평균 성적이상을 받은 학생의 학번을 구하라.

Q2) 한과목도 수강하지 않는 학생의 학번과 이름, 소속을 구하라.(exists 사용)

Q3) 소속이 같은 두 학생의 이름을 모두 구하라.(sub query, group by와 hanving절을 이용하여)

Q4) 교수명 김일삼 교수와 같은 소속의 학생들의 학번과 이름을 구하시오.

Q5) 학번이 07002가 받은 성적보다 더 많은 점수를 받은 학생의 학번과 성적을 구하시오.(any 혹은 all 이용)

<실습2>

emp, dept, slagrade 테이블 이용

Q1) SCOTT의 급여와 동일하거나 더 많이 받는 사원명과 급여를 출력하시오.

Q2) 직급(job)이 사원(CLERK)인 사람의 부서의 부서번호와 부서명과 지역을 출력하시오.

1)서브 쿼리 이용 :

2) 조인 이용 :

Q3) 이름에 T를 포함하고 있는 사원들과 같은 부서에서 근무하고 있는 사원의 사원번호와 이름을 출력하시오.

Q4) SALES부서의 모든 사원의 이름과 급여를 출력하시오.

Q5) 자신의 급여가 평균 급여보다 많고 이름에 S가 들어가는 사원과 동일한 부서에서 근무하는 모든 사원의 사원번호 , 이름 및 급여를 출력하시오.

★ <실습1 모범답안 - 답안은 가리고 풀이하되 정말 모를 경우 참조할 것>

Q1) Select 학번 from 수강 where 성적>=(select avg(성적) from 수강 where 번호=102) and 번호 = 102;

Q2) select 학번, 이름, 소속 from 학생 where not exists(select 학번 from 수강 where 학생.학번=수강.학번);

Q3) select 이름, 소속 from 학생 where 소속=(select 소속 from 학생 group by 소속 hanving count(소속)=2);

● Where 절에 존재하는 서브쿼리의 유형

3. 다중 열 서브쿼리 : 서브쿼리의 결과값이 두 개 이상의 컬럼을 반환할 때.

ex) 부서번호가 30이고 커미션을 받는 직원의 이름, 연봉, job을 구하시오.

```
SELECT ename, sal, job FROM emp
WHERE (sal, deptno) IN ( SELECT sal, deptno FROM emp
                        WHERE deptno = 30
                        AND comm is NOT NULL );
```

4. 조건절에 2개이상의 서브쿼리

ex) 학년이 2학년인 학생중에 김일삼교수에게서 수강하는 학생의 학번과 성적을 구하시오.

```
select 학번, 성적 from 수강 where
        학번 in (select 학번 from 학생 where 학년='2')
        and 번호 in (select 번호 from 설강 where 교수명 = '김일삼') ;
```

● FROM절에 존재하는 서브쿼리

- 테이블 위치에 서브쿼리가 위치하며 가상의 테이블 취급을 받음.(select만 올수 있음)

- 반드시 가상의 테이블의 별명을 주어야 한다.

ex) 소속이 CD인 학생들 중에 신장이 170이상인 학생의 이름,소속,신장을 구하시오.

```
select 이름,소속,신장 from (select * from 학생 where 소속 = 'CD') AB
where 신장 >= 170;
```

● 집합쿼리

① Union : 테이블의 직렬(세로)병합(합집합의 의미) - 중복되는것은 한번만 출력.

```
ex)  SELECT deptno FROM emp
      UNION
      SELECT deptno FROM dept;
```

① Union All : 중복되는것 모두 출력.

```
ex)  SELECT deptno FROM emp
      UNION ALL
      SELECT deptno FROM dept;
```

① INTERSECT : 두 행의 집합중 공통된 행만을 반환(교집합)

```
ex)  SELECT deptno FROM emp
      INTERSECT
      SELECT deptno FROM dept;
```

① MINUS : 첫번째 SELECT문에 의해 반환되는 행중에서 두번째 SELECT문에 의해 반환되는 행에 존재하지 않는 행들을 반환

```
ex)  SELECT deptno FROM dept
      minus
      SELECT deptno FROM emp;
```

<문법적 고려사항>

두 개의 select 문의 출력 필드명의 개수가 동일 해야 한다

● 테이블 복제 및 데이터 복사와 관련한 sub-query 유형

① table 복제

1) 테이블 구조만 복사하기.

```
create table emp02
as
select * from emp where 1=0 ;
```

2) 데이터와 함께 복사하기.

```
create table emp03
as
select * from emp ;
```

② Data 복제

```
insert into emp02
select *from emp ;
```

<실습>

Q1) emp 테이블로부터 부서별 최소연봉과 부서번호를 구한후 해당사원의 모든정보를 구하시오
(다중열서브쿼리이용)

Q2) 학생 테이블에서 신장이 큰 순서대로 정렬하고 신장이 큰 상위 2명의 모든 정보를 구하시오.
(rownum함수 이용)

※ rownum 함수 : 상위의 n개를 출력할 때 사용하는 함수이며 사용방법은 where절에서 사용.

Q3) 학생테이블중 소속이 CD인 학생만을 검색하여 학생10테이블로 복제하시오.
(테이블복제이용)

Q4) 학생테이블과 학생10테이블을 병합(union all)하여 신장이 큰 순서대로 정렬하시오.

Q5) 성적이 80이상인 학생의 학번,이름,성적을 구하되 성적순으로 정렬하시오.
(가상의 테이블 이용)

★ <실습 모범답안>

Q1) select * from emp where (sal,deptno) in (select min(sal), deptno from emp group by deptno);

Q2) select * from (select * from 학생 order by 신장 desc) ab where rownum < 3

Q5) select * from (select 학생.학번,이름,성적 from 학생,수강 where 학생.학번 = 수강.학번) abc where 성적 >= 80
order by 성적 desc

⑥ 객체단위의 SQL문(DDL)

🔖 Create문 : 객체 생성

문법 :

Create table 테이블명(Field명01 DataType, Field명02 DataType, ~ Field명[n] DataType);

● Datatype(자료형)

Datatype	설명
char(N)	주어진 길이만큼 고정길이의 문자 저장(1byte~2000byte)
varchar2(N)	주어진 길이만큼 가변길이 문자저장(1byte~4000byte)
nvarchar2(N)	varchar2와 동일하며 여거 국가별 언어를 저장할수 있다.
number(p,s)	숫자를 저장할 때 사용. 소숫점이하 표현할수 있음. p: 전체숫자길이, s : 소숫점이하 숫자 길이.
date	날짜 형식을 저장.
rowid	테이블내에 레코드의 고유한 주소를 가지는 64진수 문자 (테이블이 생성될때 자동 생성됨)
CLOB	대용량의 text 데이터를 저장, 최대 4GB
Timestamp	년,월,일 시,분,초까지 나타낼수 있는 데이터 타입

🔖 Alter문 : 객체 변경

문법 :

Alter table 테이블명

[Add] : 컬럼추가

[Modify] : 컬럼변경

[Drop] : 컬럼삭제

ex1) alter table em05 add(email varchar2(10));

ex2) alter table em05 modify(email varchar2(40));

ex3) alter table em05 drop column email;

※ 컬럼 삭제시에는 사용하지 않는 컬럼으로 등록한후 삭제작업실시하는것이 좋다.

alter table em05 set unused(email);

alter table em05 drop unused columns;

🔖 drop문 : 객체 제거

문법 :

drop table 테이블명;

<실습>

Q1) 아래와 같은 테이블을 만든 후 데이터를 입력하시오.

입사지원

입사번호 char(3)	성명 char	지원부서 varchar2	지원날짜 date	면접일시 timestamp	시험 점수평균 number	지원포부 clob
01	장동건	총무부	09/04/01	09/06/01 18:00:00	80.25	긴글입력
02	이희재	영업팀	09/05/02	09/06/01 18:30:00	90.62	긴글입력
03	원빈	고객관리부	09/04/30	09/06/01 18:30:00	70.26	긴글입력

Q2) 위의 테이블을 작성 및 데이터를 입력하다가 잘못된 테이블구조를 발견한 경우에는 alter문을 이용하여 테이블 구조를 수정하시오.

※ drop table 후 휴지통 비우기
PURGE RECYCLEBIN

⑦ SQL 주요 함수(오라클)

● 문자처리 함수(string 함수)

함수	설명
Upper(문자열)	모든문자를 대문자로 변환.
Lower(문자열)	모든문자를 소문자로 변환.
Initcap(문자열)	각단어의 첫문자만 대문자로 변환.
Length(문자열)	문자열의 길이를 반환.
Instr(문자열,찾을문자, [시작위치],[N번째문자])	특정문자를 찾아 그 위치값을 반환.
Substr(문자열,시작위치,길이)	시작위치부터 문자의 추출길이만큼 문자를 반환.
Replace(문자열,문자1,문자2)	문자1을 찾아 문자2로 치환하는 함수.
Trim(문자열)	문자열의 양쪽 공백을 제거하는 함수.
Ltrim(문자열,[문자])	왼쪽공백 또는 특정문자제거
Rtrim(문자열,[문자])	오른쪽공백 또는 특정문자제거
Lpad(문자열,전체길이,특정문자)	왼쪽에 특정문자를 삽입한다.
Rpad(문자열,전체길이,특정문자)	오른쪽에 특정문자를 삽입한다.
Concat(문자1,문자2)	문자1과 문자2의 결합

ex1) select initcap('welcome to oracle 10g') from dual;

※ dual테이블은 select의 문법을 맞추기 위한 오라클에서 제공하는 허상의 테이블이다.

ex2) select ename,job from emp where upper(ename) like 'A%';

ex3) select ' welcome ' from dual;

select trim(' welcome ') from dual;

select length(' welcome ') from dual;

select length(trim(' welcome ')) from dual;

select length(ltrim(' welcome ')) from dual;

select length(rtrim(' welcome ')) from dual;

ex4) select instr('welcome to oracle 10g','o',3,2) from dual;

select instr('welcome to oracle 10g','o') from dual;

ex5) select substr('welcome to oracle 10g',4,3) from dual;

select hiredate,substr(hiredate,1,2) from emp;

ex6) select replace('webprogreem','em','am') from dual;

ex7) select substr('welcome to oracle 10g',instr('welcome to oracle 10g','o',11),6) from dual;

ex8) select Lpad('Oracle 10g', 20, '*') from dual;
select Rpad('Oracle 10g', 20, '*') from dual;

ex9) select concat('웹프로그램반',' 화이팅') from dual;
select '웹프로그램반' || ' 화이팅~' from dual

※ || : 오라클에서 사용되는 문자열 연결 연산자.

<실습>

Q1) emp테이블의 Ename 컬럼중에서 왼쪽에서 세글자중에 L자가 포함된 사원의 이름과 입사일자를 구하시오.

Q2) 학생table 학번필드를 이용해서 학생의 입학년도를 아래와 같이 출력되도록 쿼리를 작성하시오.

입학년도 이름

08년도입학 홍길동
07년도입학 김길동
07년도입학 박길동
07년도입학 이길동
05년도입학 최길동
06년도입학 정길동

Q3) dept테이블에서 지역명이 NEW로 시작하는 데이트를 찾아서 모든정보를 검색하시오.

1)

2)

Q4) dept 테이블에서 sales데이터는 아래와 같이 영업으로 나오도록 쿼리를 완성하시오.

부서명

ACCOUNTING
RESEARCH
영업
OPERATIONS

Q5) 설강테이블에서 교수명의 이름중에 가운데 글자가 “소”가 들어가는 세자리 이름을 가진 교수명의 모든정보를 구하시오.

Memo

● 숫자함수(Number 함수)

함수	설명
ROUND(n, [m])	지정 한 자리 수 이하에서 반올림
Trunc(n,[m])	지정 한 자리 수에서 버림
Ceil(n)	소숫점 이하 자리에서 올림.(무조건정수반환)
FLOOR(n)	주어진 값보다 작거나 같은 최대 정수값을 구하는 함수
MOD(m, n)	m을 n으로 나누어 남은 값을 반환한다
ABS(n)	숫자의 절대값을 반환한다.
POWER(m, n)	m의 n승을 반환한다.

ex1) select round(45.293,2) from dual; 45.29
 select round(45.293) from dual; 45
 select round(45.293,-1) from dual; 50

ex2) select round(-45.293,2) from dual;
 select round(-45.293) from dual;
 select round(-45.293,-1) from dual;

ex3) select Trunc(45.296,2) from dual;
 select Trunc(45.296) from dual;
 select Trunc(45.296,-1) from dual;

ex4) select Ceil(45.293) from dual; 46
 select Ceil(-45.293) from dual; -45

ex5) select Floor(45.293) from dual; 45
 select Floor(-45.293) from dual; -46

ex6) select Mod(10,2) from dual;
 select sal, mod(sal,100) from emp;

ex7) select abs(-45) from dual;

ex8) select power(2,4) from dual;
 select power(3,2) from dual;

● 날짜관련 함수(Date 함수)

함수	설명
sysdate	시스템의 현재 날짜를 구하는 함수.
MONTHS_BETWEEN(a1, a2)	날짜와 날짜사이의 개월 수를 구함.
ADD_MONTHS(날짜형, 숫자)	특정날짜에 특정개월수를 더한 날짜를 반환
LAST_DAY(날짜형)	해당날짜가 속한 달의 마지막 날짜를 반환.
NEXT_DAY(날짜형,요일)	해당날짜 후의 첫요일의 날짜를 반환.
round(날짜형,[year,month,day])	해당년,월,일기준으로 날짜를 반올림
trunc(날짜형,[year,month,day])	해당년,월 기준으로 날짜를 절삭.

※ 날짜 연산

날짜 + 숫자 : 특정한 날로부터 며칠 후의 날짜 계산

날짜 - 숫자 : 특정한 날로부터 며칠 전의 날짜 계산

날짜 - 날짜 : 두 날짜 사이의 차이를 숫자로 계산

```

ex1) select sysdate from dual;
ex2) select sysdate,hiredate,months_between(sysdate,hiredate) from emp;
ex3) select hiredate,add_months(hiredate,6) from emp;
ex4) select hiredate,last_day(hiredate) from emp;
ex5) select hiredate, next_day(hiredate,'금') from emp;
ex6) select round('09/10/11','month') from dual;
ex7) select round(TO_DATE('09/10/11'),'MONTH') from dual;
      select round(TO_DATE('09/10/11'),'YEAR') from dual;
      select round(TO_DATE('09/10/11'),'DAY') from dual;
ex8) select trunc(TO_DATE('09/10/11'),'MONTH') from dual;
ex9) select to_date('09/10/11') - to_date('09/10/01') from dual;
      select to_date('09/10/11') -10 from dual
  
```

<실습>

Q1) 본인이 태어난 날로부터 오늘까지 며칠이 지났는지 계산하시오.

Q2) 본인이 태어난 날로부터 오늘까지 몇달이 지났는지 계산하시오.

Q3) A회사의 퇴직금 지급날짜는 입사후 12개월이 지난 후 그달의 마지막 날이다. emp테이블을 이용하여 사원들의 이름과 퇴직금 지급일자를 출력하시오.

Q4) 연봉이 3000이상인 사원들은 입사후 몇 달이 지났는지 계산하시오(emp테이블 이용,이름,job출력, 소숫점이하는 버림)

● 형변환 함수

함수	설명
TO_DATE('문자','format')	문자형날짜를 완전한 날짜형으로 변환하는 함수.
TO_CHAR(number/date,'format')	숫자 또는 날짜형을 완전한 문자형으로 변환.
TO_NUMBER('문자')	문자형을 완전한 숫자형으로 변환.

※ 날짜출력형식(To_char,To_date함수 사용시 이용)

날짜 출력형식		시간출력형식	
종류	의미	종류	의미
YYYY	년도 표현(4자리)	AM 또는 PM	오전(AM),오후(PM) 시각표시
YY	년도 표현(2자리)	HH 또는 HH12	시간(1~12)
MM	월을 숫자로 표현	HH24	24시간으로 표현(0~23)
MON,MONTH	월을 글자로 표현	MI	분
DAY	요일 표현	SS	초
DY	요일을 약어로 표현		
dd	날짜 표현(2자리)		

※ 숫자출력형식(To_char함수 사용시 이용)

종류	의미	예	결과
9	한 자리의 숫자 표현	(1111,'999999')	1111
0	앞부분을 0으로 표현	(1111,'099999')	001111
\$	달러기호를 앞에 표현	(1111,'\$99999')	\$1111
L	지역통화(W표시)	(1111,'L99999')	W1111
.	소수점을 표시	(1111,'99999.99')	1111.00
,	특정위치에 , 표시	(1111,'99,999')	1,111
MI	오른쪽에 - 기호 표시	(-1111,'99999MI')	1111-
PR	음수값을 <>로 표시	(-1111,'99999PR')	<1111>
B	공백을 0으로 표현	(1111,'B9999.99')	1111.00

● NULL 변환 함수

함수	설명
NVL(필드명,값)	Null을 0또는 다른값으로 변환하기 위해서 사용

ex1) select to_date(hiredate,'YY/mm/dd') from emp;

ex2) select to_char(sysdate,'YYYY/MM/DD, HH24:MI:SS') from dual;
select to_char(sysdate, 'yyyy"년" mm"월" dd"일" hh"시" mi"분"') from dual;

ex3) select ename,to_char(sal,'L999,999') from emp;

ex4) select ename,sal,comm,sal*12+nvl(comm,0) from emp

<실습>

Q1) 학생테이블에서 학년의 평균을 구하시오.(형변환함수 사용할것)

Q2) 수강테이블에서 성적을 세자리 단위로 나타내어서 검색하시오.

ex) 95 -> 095

Q3) 학생테이블과 수강테이블을 이용하여 학번의 앞두자리와 성적을 더한 값을 계산하시오.

(문자함수 및 형변환함수 이용할것)

Q4) emp테이블에서 comm의 평균을 계산하시오.

1) 커미션을 받는 사람의 평균

2) 전체 평균

Q5) 연봉이 3000이상인 직원들의 입사년월을 아래와 같이 나타내고 연봉이 높은순서 대로 정렬하시오.

(형변환함수 사용)

ENAME	SAL	입사년월
KING	5000	81년 11월입사
FORD	3000	81년 12월입사
SCOTT	3000	87년 04월입사

Q6) emp 테이블의 mgr컬럼에서 null값을 "CEO"라는 문자로 바꾸어서 출력하시오.

Q7) emp테이블에서 입사일을 아래와 같이 나타내시오.

입사년월일

1980-12월/17	수요일
1981-2월 /20	금요일
1981-2월 /22	일요일
1981-4월 /02	목요일
1981-9월 /28	월요일
1981-5월 /01	금요일
1981-6월 /09	화요일
1987-4월 /19	일요일
1981-11월/17	화요일
1981-9월 /08	화요일
1987-5월 /23	토요일
1981-12월/03	목요일
1981-12월/03	목요일
1982-1월 /23	토요일

★ 지난시간 QnA

Q) select round(TO_DATE('09/10/11'),'DAY') from dual

round함수에서 구성인수로 DAY가 쓰였을때?

A) 하루 24시간에서 반올림이 아니라 일주일 단위로 반올림되어 일요일의 날짜를 반환함.

목요일부터 반올림되어 일요일의 날짜를 반환함.

● 기타 함수

함수	설명
DECODE	<p>언어의 else if문과 같은역할(다양한 조건검색)</p> <p>문법 : decode(컬럼명,조건1,결과1,조건2,결과2,...조건N,결과N)</p>
CASE	<p>decode함수와 유사함. 다양한 비교연산자 이용가능.</p> <p>문법 : case [컬럼명] when 조건1 then 결과1 when 조건2 then 결과2 when 조건3 then 결과3 else 결과 N end</p>

[illegible]

```
ex2) SELECT deptno, CASE deptno
      WHEN 10 THEN 'ACCOUNTING'
      WHEN 20 THEN 'RESEARCH'
      WHEN 30 THEN 'SALES'
      ELSE 'OPERATIONS'
      END
      FROM emp
```

```
select  job, sal, case when job='CLERK' then sal * 1.20
                        when job='ANALYST' then sal * 1.15
                        when job='MANAGER' then sal * 1.10
                        else sal * 1
                        end as 연봉인상
from emp;
```

<실습>

Q1) 학생테이블에서 신장이 170미만이면 “소”, 170~178이면 “중”, 179이상이면 “대”를 출력하시오.
(이름,신장도 함께 출력)

Memo

⑧ 무결성 제약 조건

☞ 정의 : 테이블에 부적절한 자료가 입력되는 것을 방지하기 위해서 여러 가지 규칙을 정의하는 것을 말함.

● 무결성제약조건의 종류

무결성제약조건	설명
NOT NULL	해당 컬럼값으로 NULL을 허용하지 않음.
UNIQUE	해당 컬럼값은 중복된 값이 없어야 한다.
PRIMARY KEY	NOT NULL And UNIQUE
FOREIGN KEY	참조되는 컬럼에 없는 값은 입력이 불가함.
CHECK	해당컬럼에 저장가능한 데이터 값의 범위나 조건을 지정.

● 제약조건의 정의 방법

1. 컬럼레벨 정의 방법 : 테이블을 생성 또는 수정할 때 각각의 컬럼마다 지정하는 방법이다.

문법 : 컬럼명 data_type [constraint] [제약조건명] constraint_type

ex)

```
create table dept03
```

```
(deptno number(2) constraint dept03_deptno_pk primary key,
```

```
  dname varchar(15),
```

```
  loc varchar2(15));
```

2. 테이블레벨정의 방법 : 컬럼이 모두 정의 된후 마지막에 제약조건의 정의하는 방법.

동일한 제약조건이 여러컬럼에 적용될때 유용하며

Not null제약조건은 이 방법으로 정의할 수 없다.

문법 : [constraint] [제약조건명] constraint_type (column_name1, column_name2)

ex)

```
create table dept05
```

```
(deptno number(2),
```

```
  dname varchar(15),
```

```
  loc varchar2(15),
```

```
  constraint dept06_deptno_pk primary key(deptno,dname));
```

※ 제약조건 확인

- 테이블에 정의되어 있는 제약조건을 확인하기 위해서는 오라클에서 제공하는 데이터 디크셔너리 중 user_constraints 를 이용하여 확인할 수 있다.

P : Primary key, R: foreign key, U: unique, C : check 또는 not null

ex) select constraint_name,constraint_type,table_name from user_constraints;

1. 기본키(PK,Primary Key) 제약조건

관계형 데이터베이스에서는 모든 테이블은 하나이상의 주키를 가져야 하며 주키는 아래 두가지 사항을 항상 만족하는 필드만이 될수 있다.

1) UNIQUE하여야 한다.(중복된 데이터가 없어야 한다)

2) Not Null이어야 한다.(항상 데이터가 채워져야 한다)

<실습>

```
create table dept01
(deptno number(2) primary key,
  dname varchar2(15),
  loc varchar2(15));

insert into dept01 values(1,'A','AA');
insert into dept01 values(1,'B','BB');
insert into dept01(dname,loc) values('C','CC'); V
```

2. 외래키(FK,Foreign Key) 제약조건

관계형 데이터베이스에서는 테이블과 테이블이 하나이상의 관계를 맺고 있다. 이 관계를 맺는 역할을 담당하는 것이 주키와 외래키이며 외래키란 다른테이블의 주키를 참조하는 키를 말한다.

- 참조하고 있는 PK와 data_type 과 길이가 일치하여야 한다.
- 자식테이블에 공통된 data가 있을때는 부모테이블의 data를 삭제할 수 없다.

<실습>

```
create table dept07
(deptno number(2),
  dname varchar2(15),
  loc varchar2(15),
  constraint dept07_deptno_pk primary key(deptno));

create table emp07
(empno number(4) primary key,
  ename varchar2(15),
  deptno number(2),
  constraint emp07_deptno_fk foreign key(deptno) references dept07 (deptno));

select table_name,constraint_type,constraint_name,r_constraint_name
  from user_constraints
 where table_name in('DEPT07','EMP07');

insert into dept07 values(1,'A','AA');
insert into dept07 values(2,'B','BB');

insert into emp07 values(1111,'EA',1);
insert into emp07 values(2222,'EB',2);
insert into emp07 values(3333,'EA',3); V
```

```
delete from dept07 where deptno=1; V
```

```
alter table dept07
```

```
drop constraint dept07_deptno_pk; V
```

```
alter table dept07
```

```
drop constraint dept07_deptno_pk cascade;
```

※ cascade : 제약조건을 참조하는 모든 제약 조건들을 먼저 제거한 후 본 제약을 삭제하라.

```
select table_name,constraint_type,constraint_name,r_constraint_name
```

```
from user_constraints
```

```
where table_name in('DEPT07','EMP07');
```

※ 참조무결성이 정의되어 있는 table마다 위의 사항처럼 모든 제약을 삭제한후 data를 삭제한다면 데이터무결성이 보장되지 못한다, 그래서 data 꼭 삭제해야 하는 경우라면 아래처럼 disable옵션을 사용하여 참조무결성이 정의되어 있는 table의 data를 삭제할 수 있다.

```
alter table dept07
```

```
add constraint dept07_deptno_pk primary key(deptno);
```

```
alter table emp07
```

```
add constraint emp07_deptno_fk foreign key(deptno) references dept07(deptno);
```

```
select table_name,constraint_type,constraint_name,r_constraint_name
```

```
from user_constraints
```

```
where table_name in('DEPT07','EMP07');
```

```
alter table dept07
```

```
disable constraint dept07_deptno_pk cascade;
```

```
select table_name,constraint_type,constraint_name,r_constraint_name, status
```

```
from user_constraints
```

```
where table_name in('DEPT07','EMP07');
```

```
delete from dept07 where deptno=1;
```

```
alter table dept07
```

```
enable constraint dept07_deptno_pk; -활성화
```

```
alter table emp07
```

```
enable constraint EMP07_DEPTNO_FK;V
```

```
insert into dept07 values(1,'A','AA');
```

```
alter table emp07
```

```
enable constraint EMP07_DEPTNO_FK;
```

3. UNIQUE 제약

- 유일한 데이터의 집합이어야 한다.(중복된값을 가질수 없다.)
- NULL은 허용한다.

<실습>

```
alter table dept07
add constraint dept07_deptno_uq unique(dname);
```

```
insert into dept07 values(5,'E','EE');
insert into dept07 values(6,'E','EA');V
insert into dept07 values(7,null,'EA');
insert into dept07 values(8,null,'KA');
```

4. CHECK 제약

- 특정한 값이나 data의 범위를 지정하여 그 이외의 값은 들어오지 못하도록 하는 제약조건.

<실습>

```
create table emp08
(empno number(4) primary key,
ename varchar2(15),
sal number(7,2) constraint emp08_sal_ck check(sal between 500 and 5000) ,
deptno number(2));
```

```
select table_name,constraint_type,constraint_name,r_constraint_name, search_condition
from user_constraints
where table_name in('EMP08');
```

```
insert into emp08 values(1,'A',800.25,11);
insert into emp08 values(1,'A',400.15,11);V
```

5. NOT NULL 제약

- null값을 가질 수 없다는 제약
- 컬럼레벨방식으로만 정의 할 수 있다.

<실습>

```
create table dept09
(deptno number(2),
dname varchar2(15) constraint dept09_dname_nn not null,
loc varchar2(15));
```

```
alter table dept09
modify loc constraint dept09_loc_nn not null;
```

※ not null 제약조건은 add constraint 대신 modify명령문을 사용한다. 왜냐하면 not null 제약조건 없이 테이블 생성시는 null을 허용한다는 의미로 이미 생성된 것이므로 컬럼의 수정사항이라고 보면 된다.

⑨ View

☞ 정의 : DB내에서 하나의 Object로 존재하며 테이블을 근거한 가상의 테이블이라고 정의 할수 있음. 자주 select되는 Join등이 포함된 복잡한 select문장을 view의 형태로 저장을 시켜놓고 사용하면 DBMS의 성능향상과 효율적인 Query작업을 할 수 있다.

● View 생성

```
Create view view_name  
as  
select_statement
```

<문법적 고려사항>

- VIEW를 정의하는 질의어에는 ORDER BY절을 사용 할 수 없습니다.
- 함수를 사용한 컬럼은 반드시 ALIAS를 지정해야 합니다.

● View 변경

```
Create or Replace view view_name  
as  
select_statement
```

● View 삭제

```
Drop View view_name
```

● View 검색

```
select * from view_name
```

<실습1>

- dept_copy, emp_copy 생성(예제복사 테이블)

```
create table dept_copy  
as  
select * from dept;  
create table emp_copy  
as  
select * from emp;
```

- view 생성(부서번호가 30번인 사원의 사원번호,성명,부서번호가 자주 검색된다고 가정)

```
create view emp_view30  
as  
select empno,ename,deptno from emp_copy  
where deptno=30;
```

※ 현재 scott계정은 view 생성권한을 가지고 있지 않으므로 권한 부여 후 view생성.
(권한설정에 관한 부분은 DB관리 part에서 학습 할 예정)

- 1) conn system/비밀번호
- 2) grant create view to scott;
- 3) conn scott/비밀번호
- 4) View 생성

– view 검색

```
select * from emp_view30
```

– view 객체 검색

```
select view_name,text from user_views;
```

<실습2>

– 함수를 사용한 컬럼은 반드시 별명을 지정

```
create view dept_sum  
as  
select deptno, sum(sal) sum_sal  
from emp  
group by deptno;
```

– create view emp_view_join

```
as  
select e.empno,e.ename,d.dname  
from emp e, dept d  
where e.deptno = d.deptno;
```

– 뷰의 변경

```
create or replace view emp_view30  
as  
select empno,ename,deptno from emp_copy  
where deptno=10;
```

– 뷰의 삭제

```
drop view emp_view30
```

⑩ SEQUENCE

☞ 정의 : 연속적으로 자동증가값을 만들어주는 Object.

● Sequence 생성

```
Create Sequence sequence_name
[increment by n]
[start with n]
[Maxvalue n]      - 기본값 1027
[minvalue n]      - 기본값 1
```

※ Sequence는 테이블과는 독립적으로 저장되고 생성됩니다. 따라서 하나의 sequence를 여러 테이블에서 쓸 수 있다.

● Sequence 수정

```
Alter Sequence sequence_name
[increment by n]
[start with n]
[Maxvalue n]
[minvalue n]
```

● Sequence 삭제

```
Drop Sequence sequence_name
```

● Currval과 nextval

currval : 시퀀스의 현재값을 알아내기 위해서 사용

nextval : 시퀀스의 값을 증가시킬때 사용

```
ex) select dept_deptno_seq.nextval from dual;
     select dept_deptno_seq.currval from dual;
```

<실습>

- sequence 생성

```
create sequence dept_deptno_seq
increment by 1
start with 1 ;
```

- sequence에 관한 데이터 덱서너리

```
select sequence_name, min_value,max_value, increment_by from user_sequences ;
```

- sequence의 실무 적용 : 시퀀스는 거의 테이블의 컬럼에서 자동증가값을 생성할 때 사용한다.

```
create table dept15
(deptno number(4) primary key, dname varchar(15), loc varchar(15));
```

```
insert into dept15 values(dept_deptno_seq.nextval, '인사과','서울');
insert into dept15 values(dept_deptno_seq.nextval, '경리과','서울');
insert into dept15 values(dept_deptno_seq.nextval, '총무과','대전');
```

```
select * from dept15;
```