```
SELECT TABLE_NAME FROM USER_TABLES;

--INSERT?

--INSERT INTO T/N(C/N)

--VALUES(VALUES)

SELECT * FROM ENROL;

DESC ENROL;

DROP TABLE A_ENROL;

CREATE TABLE A_ENROL

AS

SELECT *

FROM ENROL

WHERE STU_NO <'20150000';

DESC A_ENROL;

SELECT * FROM A_ENROL;


INSERT INTO A_ENROL(SUB_NO,STU_NO,ENR_GRADE)

VALUES(108,20151062,92);

INSERT INTO A_ENROL

VALUES(109,20152088,85);

SELECT * FROM A_ENROL;


INSERT INTO A_ENROL(SUB_NO,STU_NO)

VALUES(110,20152088);

SELECT * FROM A_ENROL;

INSERT INTO A_ENROL

VALUES(111,20153075,NULL);
```

```
SELECT * FROM A_ENROL;
--복수행 삽입?
SELECT * FROM ENROL;
SELECT * FROM ENROL
WHERE STU_NO LIKE '2015%';

INSERT INTO A_ENROL
SELECT * FROM ENROL
WHERE STU_NO LIKE '2015%';
SELECT * FROM A_ENROL;

--UPDATE?
--UPDATE <T/N>
--SET CO--NAME=??
--WHERE ??
SELECT * FROM A_ENROL;
UPDATE A_ENROL
SET ENR_GRADE=ENR_GRADE+5;
UPDATE A_ENROL
SET ENR_GRADE=ENR_GRADE+5
WHERE SUB_NO=101;
--과목이름이 '시스템분석설계'인 그 과목만 점수를 10점 업데이트하라
UPDATE A_ENROL
SET ENR_GRADE=ENR_GRADE+10
WHERE SUB_NO=(SELECT SUB_NO
FROM SUBJECT
```

```
WHERE SUB_NAME='시스템분석설계');

SELECT SUB_NO
FROM SUBJECT
WHERE SUB_NAME='시스템분석설계';
SELECT * FROM A_ENROL;

--DELETE?
--DELETE FROM <T/N>
--WHERE ???
DELETE FROM A_ENROL
WHERE STU_NO=20131001;
SELECT * FROM A_ENROL;
--A_ENROL테이블에서 과목이름이 기계요소설계인 과목번호를 가진 내용을 삭제하라
DELETE FROM A_ENROL
WHERE SUB_NO=(SELECT SUB_NO
FROM SUBJECT
WHERE SUB_NAME='기계요소설계');
SELECT SUB_NO
FROM SUBJECT
WHERE SUB_NAME='기계요소설계';

--다중튜플삭제?
DELETE FROM A_ENROL;

--TCL?
```

```
SELECT * FROM B_STUDENT;

DELETE FROM B_STUDENT;

SELECT * FROM B_STUDENT;

ROLLBACK;

SELECT * FROM B_STUDENT;

DELETE FROM B_STUDENT;

SELECT * FROM B_STUDENT;

CREATE TABLE C_STUDENT(STU_NO NUMBER,

STU_NAME CHAR(10));

ROLLBACK;

SELECT * FROM B_STUDENT;


SELECT * FROM A_STUDENT;

DELETE FROM A_STUDENT;

SELECT * FROM A_STUDENT;

ROLLBACK;

--병행처리?

SELECT *

FROM A_STUDENT;

INSERT INTO A_STUDENT(STU_NO,STU_NAME)

VALUES(10,'홍');

SELECT * FROM A_STUDENT;

COMMIT;


--DML실전문제

--배경환경구축테이블 생성
```

```
CREATE TABLE EMP1
AS
SELECT * FROM EMP WHERE DEPTNO IN(20,30);

CREATE TABLE DEPT1
AS
SELECT * FROM DEPT;
CREATE TABLE SALGRADE1
AS
SELECT * FROM SALGRADE;
SELECT * FROM EMP1;
SELECT * FROM DEPT1;
SELECT * FROM SALGRADE1;
--1.사원번호 7703,사원이름 JOSH,사원직무 SALESMAN,상급자사원번호가
--7566,급여1400,커미션0,부서번호 20인 사원이 오늘 입사하였다.
INSERT INTO EMP1
VALUES(7703,'JOSH','SALESMAN',7566,SYSDATE,1400,0,20);
SELECT * FROM EMP1;
--2. 사원번호 7401,사원이름 HOMER, 급여 1300,부서번호10인사원이
--입사하였다.
INSERT INTO EMP1(EMPNO,ENAME,SAL,DEPTNO)
VALUES(7401,'HOMER',1300,10);
SELECT * FROM EMP1;
--3.사원번호7323,사원이름 'BRENDA'부서번호30,사원번호7499와
--동일한 급여를 받는 사원이 입사하였다.
INSERT INTO EMP1(EMPNO,ENAME,SAL,DEPTNO)
```

```
VALUES(7323,'BRENDA',(SELECT SAL FROM EMP1 WHERE EMPNO=7499),30);

SELECT * FROM EMP1;

--4.사원(EMP)테이블에서 부서번호가 10인 데이터를 EMP1테이블에 삽입하라?

INSERT INTO EMP1

SELECT * FROM EMP WHERE DEPTNO=10;

SELECT * FROM EMP1;

--5.사원번호 7369의 사원직부를 ANALYST로 바꾸어라

UPDATE EMP1

SET JOB='ANALYST'

WHERE EMPNO=7369;

SELECT * FROM EMP1;

--6.부서번호20인 직원들의 급여를 10%감하라

UPDATE EMP1

SET SAL=SAL-SAL*0.1

WHERE DEPTNO=20;

SELECT * FROM EMP1;

--7.모든 사원의 급여를 100증가시켜라

UPDATE EMP1

SET SAL=SAL+100;

SELECT * FROM EMP1;

--8.사원번호 7902 상급자사원번호를 7654,부서번호를 30으로 바꾸라

UPDATE EMP1

SET MGR=7654,DEPTNO=30

WHERE EMPNO=7902;

SELECT * FROM EMP1;

--9.지역이 DALLAS인사원들의 급여를 10감하라
```

```
UPDATE EMP1
SET SAL=SAL-10
WHERE DEPTNO=(SELECT DEPTNO FROM DEPT1 WHERE LOC='DALLAS');
SELECT * FROM EMP1;
SELECT DEPTNO FROM DEPT1 WHERE LOC='DALLAS';
--10.급여등급이 2인 사원들의 급여를 20감하라
UPDATE EMP1
SET SAL=SAL-20
WHERE EMPNO IN(SELECT EMPNO
FROM EMP1,SALGRADE
WHERE SAL BETWEEN LOSAL AND HISAL
AND GRADE=2);
SELECT EMPNO
FROM EMP1,SALGRADE
WHERE SAL BETWEEN LOSAL AND HISAL
AND GRADE=2;
SELECT * FROM EMP1;
--11.사원번호7499가 퇴사하였다.
DELETE FROM EMP1
WHERE EMPNO=7499;
SELECT * FROM EMP1;
--12.부서번호50, 부서이름'PLANNING',지역'MIAMI'가 추가되었다.
INSERT INTO DEPT1
VALUES('50','PLANNING','MIAMI');
SELECT * FROM DEPT1;
--13.부서번호가 40인 부서가 60으로 변경되었다.
```

```
UPDATE DEPT1
SET DEPTNO=60
WHERE DEPTNO=40;
SELECT * FROM DEPT1;
--14.부서번호가 30인 부서가 폐지되었다.
DELETE FROM DEPT1
WHERE DEPTNO=30;
SELECT * FROM DEPT1;
--15.DEPT1테이블에 없는 부서번호들을 갖고 있는 사원들의 부서번호를
--99로 변경하라
UPDATE EMP1
SET DEPTNO=99
WHERE DEPTNO NOT IN (SELECT DEPTNO FROM DEPT1);
SELECT * FROM EMP1;
--16.EMP1에서 99번 번호를 삭제하라
DELETE FROM EMP1
WHERE DEPTNO=99;
SELECT * FROM EMP1;
--17. 상급자사원번호가 없는 사원의 급여를 100올렸다.
UPDATE EMP1
SET SAL=SAL+100
WHERE MGR IS NULL;
SELECT * FROM EMP1;
--18.JONES,JOSH,CLARK가 30번부서로 옮겼다.
UPDATE EMP1
SET DEPTNO=30
```

```
WHERE ENAME IN('JONES','JOSH','CLARK');

SELECT * FROM EMP1;

--19.커미션이 NULL인데이터를 0으로 바꾸라

--NA,NULL,0

UPDATE EMP1

SET COMM=0

WHERE COMM IS NULL;

SELECT * FROM EMP1;

--20.EMP1전체테이블을 삭제하라

--EMP1의 전체튜플을 제거하라

--EMP1의 전체레코드를 제거하라

DELETE FROM EMP1;

SELECT * FROM EMP1;


--DDL?

--CREATE/DROP/TRUNCATE/ALTER

CREATE TABLE TEST1

(U_ID VARCHAR2(20),

U_DATE DATE);

DESC TEST1;

SELECT * FROM TEST1;

--기존의테입을 이용하여 새로운 테이블을 만드는 방법

CREATE TABLE T_STUDENT

AS

SELECT * FROM STUDENT

WHERE STU_DEPT='기계';
```

```
DESC T_STUDENT;

SELECT * FROM T_STUDENT;

--열내용을 추가하는 방법

ALTER TABLE T_STUDENT

ADD (ARMY CHAR(1));

DESC T_STUDENT;

SELECT * FROM T_STUDENT;


--열의 데이터타입을 변경하는 방법

ALTER TABLE T_STUDENT

MODIFY(ARMY NUMBER);

DESC T_STUDENT;


--열의 내용을 삭제하는 방법

ALTER TABLE T_STUDENT

DROP(ARMY);


DESC T_STUDENT;

SELECT * FROM T_STUDENT;


--열의 이름을 바꾸는 방법

ALTER TABLE T_STUDENT

RENAME COLUMN STU_NAME TO NAME;


--테이블 이름을 변경하는 방법

RENAME T_STUDENT TO TEST_STUDENT;
```

```
DESC T_STUDENT;

DESC TEST_STUDENT;


--테이블의 데이터삭제하는방법(완전삭제)

TRUNCATE TABLE TEST_STUDENT;

DESC TEST_STUDENT;

SELECT * FROM TEST_STUDENT;

ROLLBACK;


--테이블삭제?

DROP TABLE TEST_STUDENT;

DESC TEST_STUDENT;


--CONSTRAINT(제약조건)

--1.NOT NULL (C)

--2.UNIQUE KEY (UK)

--3.PRIMARY KEY (PK)

--4.FOREIGN KEY (FK)

--5.CHECK (C)

--NOT NULL CONSTRAINT CASE?

CREATE TABLE T_STUDENT(

STU_NO CHAR(9),

STU_NAME VARCHAR2(12),

STU_DEPT VARCHAR2(20)

CONSTRAINT N_STU_DEPT NOT NULL,

STU_GRADE NUMBER(1),
```

```
STU_CLASS CHAR(1),
STU_GENDER CHAR(1),
STU_HEIGHT NUMBER(5,2),
STU_WEIGHT NUMBER(5,2));

--제약조건을 확인하는 방법
SELECT *
FROM USER_CONSTRAINTS
WHERE TABLE_NAME='T_STUDENT';

DROP TABLE T_STUDENT;
CREATE TABLE T_STUDENT(
STU_NO CHAR(9),
STU_NAME VARCHAR2(12)
CONSTRAINT U_STU_NAME UNIQUE,
STU_DEPT VARCHAR2(20)
CONSTRAINT N_STU_DEPT NOT NULL,
STU_GRADE NUMBER(1),
STU_CLASS CHAR(1),
STU_GENDER CHAR(1),
STU_HEIGHT NUMBER(5,2),
STU_WEIGHT NUMBER(5,2));
SELECT *
FROM USER_CONSTRAINTS
WHERE TABLE_NAME='T_STUDENT';
```

```
DROP TABLE T_STUDENT;
CREATE TABLE T_STUDENT(
STU_NO CHAR(9),
STU_NAME VARCHAR2(12)
CONSTRAINT U_STU_NAME UNIQUE,
STU_DEPT VARCHAR2(20)
CONSTRAINT N_STU_DEPT NOT NULL,
STU_GRADE NUMBER(1),
STU_CLASS CHAR(1),
STU_GENDER CHAR(1),
STU_HEIGHT NUMBER(5,2),
STU_WEIGHT NUMBER(5,2),
CONSTRAINT P_STU_NO PRIMARY KEY(STU_NO) );

--PRIMARY KEY를 동시에 두개를 할당하는 경우의 CASE임.
CREATE TABLE T_ENROL(
SUB_NO CHAR(3),
STU_NO CHAR(9),
ENR_GRADE NUMBER(3),
CONSTRAINT P_ENROL PRIMARY KEY(SUB_NO,STU_NO));
SELECT *
FROM USER_CONSTRAINTS
WHERE TABLE_NAME='T_ENROL';

SELECT *
FROM USER_CONSTRAINTS
```

```
WHERE TABLE_NAME='SUBJECT';

SELECT *

FROM USER_CONSTRAINTS

WHERE TABLE_NAME='STUDENT';


DROP TABLE T_ENROL;

DESC STUDENT;

DESC SUBJECT;

CREATE TABLE T_SUBJECT

AS

SELECT *

FROM SUBJECT;


SELECT *

FROM USER_CONSTRAINTS

WHERE TABLE_NAME='SUBJECT';

SELECT * FROM T_SUBJECT;

SELECT * FROM SUBJECT;

DROP TABLE T_SUBJECT;

CREATE TABLE T_SUBJECT(

SUB_NO NUMBER(5),

SUB_NAME VARCHAR2(20),

SUB_PROF CHAR(10),

SUB_GRADE CHAR(5),

SUB_DEPT VARCHAR2(10),

CONSTRAINT P_SUB_NO PRIMARY KEY(SUB_NO));
```

```
CREATE TABLE T_ENROL(
SUB_NO NUMBER(5),
STU_NO VARCHAR2(9),
ENR_GRADE NUMBER(3),
CONSTRAINT ENR_SUB_NO_FK1 FOREIGN KEY(SUB_NO) REFERENCES T_SUBJECT(SUB_NO),
--CONSTRAINT ENR_STU_NO_FK2 FOREIGN KEY(STU_NO) REFERENCES STUDENT(STU_NO),
CONSTRAINT ENR_PK1 PRIMARY KEY(SUB_NO,STU_NO));
DROP TABLE T_ENROL;
SELECT *
FROM USER_CONSTRAINTS
WHERE TABLE_NAME='T_ENROL';
--CHECK?
DROP TABLE T_STUDENT;

CREATE TABLE T_STUDENT(
STU_NO CHAR(9),
STU_NAME VARCHAR2(12)
CONSTRAINT U_STU_NAME UNIQUE,
STU_DEPT VARCHAR2(20)
CONSTRAINT N_STU_DEPT NOT NULL,
STU_GRADE NUMBER(1),
STU_GENDER CHAR(1)
CONSTRAINT C_STU_GENDER CHECK (STU_GENDER IN('M','F')),
STU_HEIGHT NUMBER(5,2),
STU_WEIGHT NUMBER(5,2),
```

```
CONSTRAINT P_STU_NO PRIMARY KEY(STU_NO));

SELECT * FROM USER_CONSTRAINTS

WHERE TABLE_NAME='T_STUDENT';

--제약조건의 비활성화/활성화?

ALTER TABLE T_STUDENT

DISABLE CONSTRAINT N_STU_DEPT;

ALTER TABLE T_STUDENT

ENABLE CONSTRAINT N_STU_DEPT;


--제약조건의 삭제방법?

ALTER TABLE T_ENROL

DROP CONSTRAINT ENR_SUB_NO_FK1 CASCADE;

SELECT * FROM USER_CONSTRAINTS WHERE TABLE_NAME='T_ENROL';


--VIEW?

--단순뷰

CREATE OR REPLACE VIEW V_STUDENT1

AS

SELECT * FROM STUDENT

WHERE STU_DEPT='컴퓨터정보';

SELECT * FROM V_STUDENT1;

--조인뷰?

CREATE OR REPLACE VIEW V_ENROL1

AS

SELECT SUB_NAME,SUB_NO,STU_NO,ENR_GRADE

FROM ENROL NATURAL JOIN SUBJECT;
```

```
SELECT * FROM V_ENROL1;

--학과별 평균신장보다 큰 학생들의 학번, 이름, 신장을 검색하라

--IN LINE VIEW

SELECT STU_NO,STU_NAME,A.STU_DEPT,STU_HEIGHT

FROM STUDENT A,(SELECT STU_DEPT,AVG(STU_HEIGHT) AS AVG_HEIGHT

FROM STUDENT GROUP BY STU_DEPT) B

WHERE A.STU_DEPT=B.STU_DEPT

AND A.STU_HEIGHT>B.AVG_HEIGHT;
```