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;