//BHAGYA A JAI
//ROLL NO:B21CSB18
//DDL COMMANDS

1.QUESTION

Create the tables described below:
Classroom(building,room_number, capacity)
Department(dept_Name,building,budget)
Course(course_id,title,dept_name,credits)
Instructor(ID,name,dept_name,salary)
Student(ID,name,dept_name,tot_cred)
Section(course_id,sec_id, semester,year,building,room_number)
Teaches(ID,course_id,sec_id,semester,year)
Takes(ID,course_id,sec_id,semester,year,grade)

QUERY:

```
SQL> CREATE TABLE classroom( building varchar(10) , room_number int primary key , capacity
int );

SQL> CREATE TABLE department(dept_name varchar(30) primary key , building varchar(20) ,
budget int);

SQL> CREATE TABLE course ( course_id varchar(20) primary key , title varchar(50) , dept_name
varchar(20) , credits int ,foreign key (dept_name) references department(dept_name));

SQL> CREATE TABLE instructor ( ID int primary key , name varchar(30) , dept_name varchar(20)
, salary int , foreign key (dept_name) references department(dept_name));

SQL> CREATE TABLE student(ID int primary key,name varchar(20) , dept_name
varchar(20),tot_cred int , foreign key (dept_name) references department(dept_name));

SQL> CREATE TABLE section(course_id varchar(20), sec_id int ,semester varchar(20) , year int ,
building varchar(20) ,room_number int , foreign key (course_id) references course(course_id),
foreign key (room_number) references classroom(room_number));

SQL> CREATE TABLE teaches(ID int,course_id varchar(20),sec_id int,semester varchar(20),year
int,foreign key (ID) references instructor(ID),foreign key (course_id) references course(course_id));

SQL> CREATE TABLE takes(ID int,course_id varchar(20),sec_id int,semester varchar(20),year
int,grade varchar(10),foreign key (ID) references student(ID),foreign key (course_id) references
course(course_id));
```

OUTPUT :

```
SQL> describe classroom
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BUILDING                                           VARCHAR2(10)
 ROOM_NUMBER                               NOT NULL NUMBER(38)
 CAPACITY                                           NUMBER(38)

SQL> describe department
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DEPT_NAME                                 NOT NULL VARCHAR2(30)
 BUILDING                                           VARCHAR2(20)
 BUDGET                                             NUMBER(38)

SQL> describe course
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COURSE_ID                                 NOT NULL VARCHAR2(20)
 TITLE                                              VARCHAR2(50)
 DEPT_NAME                                          VARCHAR2(20)
 CREDITS                                            NUMBER(38)

SQL> describe instructor
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER(38)
 NAME                                               VARCHAR2(30)
 DEPT_NAME                                          VARCHAR2(20)
 SALARY                                             NUMBER(38)

SQL> describe student
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER(38)
 NAME                                               VARCHAR2(20)
 DEPT_NAME                                          VARCHAR2(20)
 TOT_CRED                                           NUMBER(38)

SQL> describe section
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COURSE_ID                                          VARCHAR2(20)
 SEC_ID                                             NUMBER(38)
 SEMESTER                                           VARCHAR2(20)
 YEAR                                               NUMBER(38)
 BUILDING                                           VARCHAR2(20)
 ROOM_NUMBER                                        NUMBER(38)

SQL> describe teaches
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                                 NUMBER(38)
 COURSE_ID                                          VARCHAR2(20)
 SEC_ID                                             NUMBER(38)
 SEMESTER                                           VARCHAR2(20)
 YEAR                                               NUMBER(38)

SQL> describe takes
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                                 NUMBER(38)
 COURSE_ID                                  VARCHAR2(20)
 SEC_ID                                NUMBER(38)
 SEMESTER                                   VARCHAR2(20)
 YEAR                                  NUMBER(38)
 GRADE                                    VARCHAR2(10)
```

2.QUESTION

Modify the table
a. Course such that the data type of course_id to varchar(10)
b. Department to add a new column dept_no of data type number
c. Student such that the contents of the column name should not be NULL
d. Classroom such that the default value for column capacity as 50
e. Department such that the contents of columndept_name should be unique
f. Instructor such that the values for the column salary should be greater than or equal to 40000

QUERY:

```
SQL> Alter table course modify(course_id varchar(10));

SQL> Alter table department add(department_no int);

SQL> Alter table student modify(name NOT NULL);

SQL> Alter table classroom modify(capacity default 50);

SQL> Alter table department add constraint unique_dept_name UNIQUE (dept_name);

SQL> Alter table instructor add constraint check_salary CHECK (salary >=40000);
```

OUTPUT :

```
SQL> describe course
 Name                              Null?    Type
 ----------------------------------------- -------- ----------------------------
 COURSE_ID                              NOT NULL VARCHAR2(10)
 TITLE                                    VARCHAR2(50)
 DEPT_NAME                                     VARCHAR2(20)
 CREDITS                                     NUMBER(38)

SQL> describe department
 Name                              Null?    Type
 ----------------------------------------- -------- ----------------------------
 DEPT_NAME                               NOT NULL VARCHAR2(30)
 BUILDING                                     VARCHAR2(20)
 BUDGET                                     NUMBER(38)
 DEPARTMENT_NO                                     NUMBER(38)

SQL> describe student
 Name                              Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                NOT NULL NUMBER(38)
 NAME                                 NOT NULL VARCHAR2(20)
 DEPT_NAME                                     VARCHAR2(20)
 TOT_CRED                                     NUMBER(38)
```

```
SQL> describe classroom
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BUILDING                                           VARCHAR2(10)
 ROOM_NUMBER                               NOT NULL NUMBER(38)
 CAPACITY                                           NUMBER(38)

SQL> describe instructor
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER(38)
 NAME                                               VARCHAR2(30)
 DEPT_NAME                                          VARCHAR2(20)
 SALARY                                             NUMBER(38)
```

3.QUESTION

Remove all constraints and modifications that are given to the database

QUERY

```
SQL> Alter table instructor drop constraint chk_salary;
SQL> Alter table student modify(name NULL);
SQL> Alter table department drop column department_no;
SQL> Alter table course modify(course_id varchar(20))
SQL> Alter table classroom modify( capacity default null);
```

OUTPUT

```
SQL> describe course
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COURSE_ID                                 NOT NULL VARCHAR2(20)
 TITLE                                              VARCHAR2(50)
 DEPT_NAME                                          VARCHAR2(20)
 CREDITS                                            NUMBER(38)

SQL> describe department
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DEPT_NAME                                 NOT NULL VARCHAR2(30)
 BUILDING                                           VARCHAR2(20)
 BUDGET                                             NUMBER(38)

SQL> describe student
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER(38)
 NAME                                               VARCHAR2(20)
 DEPT_NAME                                          VARCHAR2(20)
 TOT_CRED                                           NUMBER(38)

SQL> describe classroom
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BUILDING                                  VARCHAR2(10)
 ROOM_NUMBER                      NOT NULL NUMBER(38)
 CAPACITY                                  NUMBER(38)

SQL> describe instructor
 Name                            Null?    Type
 ---------------------------------------- -------- ----------------------------
 ID                              NOT NULL NUMBER(38)
 NAME                                     VARCHAR2(30)
 DEPT_NAME                                VARCHAR2(20)
 SALARY                                   NUMBER(38)
```

4.QUESTION
Change the name of table takes to subject

QUERY

```
SQL> Alter table takes rename to subject;
```

OUTPUT

```
SQL> describe subject
 Name                            Null?    Type
 ---------------------------------------- -------- ----------------------------
 ID                                       NUMBER(38)
 COURSE_ID                                VARCHAR2(20)
 SEC_ID                                   NUMBER(38)
 SEMESTER                                 VARCHAR2(20)
 YEAR                                     NUMBER(38)
 GRADE                                    VARCHAR2(10)
```

//BHAGYA A JAI
//ROLL NO:B21CSB18
//DML COMMANDS

1.QUESTION
Insert data into given tables

QUERY

```
SQL> INSERT INTO classroom VALUES ( 'Packard' , 101 ,500);
SQL> INSERT INTO classroom VALUES ( 'Painter' ,514 ,10);
SQL> INSERT INTO classroom VALUES ( 'Taylor',3128 ,70);
SQL> INSERT INTO classroom VALUES ( 'Watson',100 ,30);
SQL> INSERT INTO classroom VALUES ( 'Watson',120 ,50);



SQL> INSERT INTO department VALUES('Biology','Watson',90000);
SQL> INSERT INTO department VALUES('Comp.Sci','Taylor',100000);
SQL> INSERT INTO department VALUES('Elec.Eng.','Taylor',85000);
SQL> INSERT INTO department VALUES('Finance','Painter',120000);
SQL> INSERT INTO department VALUES('History','Painter',50000);
SQL> INSERT INTO department VALUES('Music','Packard',80000);
SQL> INSERT INTO department VALUES('Physics','Watson',70000)


SQL> INSERT INTO course values('&course_id','&title','&dept_name',&credits);
Enter value for course_id: BIO_101
Enter value for title: Intro. to Biology
Enter value for dept_name: Biology
Enter value for credits: 4
old   1: INSERT INTO course values('&course_id','&title','&dept_name',&credits)
new   1: INSERT INTO course values('BIO_101','Intro. to Biology','Biology',4)

SQL> INSERT INTO course values('&course_id','&title','&dept_name',&credits);
Enter value for course_id: BIO_301
Enter value for title: Genetics
Enter value for dept_name: Biology
Enter value for credits: 4
old   1: INSERT INTO course values('&course_id','&title','&dept_name',&credits)
new   1: INSERT INTO course values('BIO_301','Genetics','Biology',4)

SQL> INSERT INTO course values('&course_id','&title','&dept_name',&credits);
Enter value for course_id: BIO_399
Enter value for title: Computational Biology
Enter value for dept_name: Biology
Enter value for credits: 3
old   1: INSERT INTO course values('&course_id','&title','&dept_name',&credits)
new   1: INSERT INTO course values('BIO_399','Computational Biology','Biology',3)

SQL> INSERT INTO course values('&course_id','&title','&dept_name',&credits);
Enter value for course_id: CS_101
Enter value for title: Intro. to Computer Science
Enter value for dept_name: Comp.Sci
Enter value for credits: 4
old   1: INSERT INTO course values('&course_id','&title','&dept_name',&credits)
new   1: INSERT INTO course values('CS_101','Intro. to Computer Science','Comp.Sci',4)

SQL> INSERT INTO course values('&course_id','&title','&dept_name',&credits);
Enter value for course_id: CS_190
Enter value for title: Game Design
Enter value for dept_name: Comp.Sci
Enter value for credits: 4
old   1: INSERT INTO course values('&course_id','&title','&dept_name',&credits)
new   1: INSERT INTO course values('CS_190','Game Design','Comp.Sci',4)

SQL> INSERT INTO course values('&course_id','&title','&dept_name',&credits);
Enter value for course_id: CS_315
Enter value for title: Robotics
Enter value for dept_name: Comp.Sci
Enter value for credits: 3
old   1: INSERT INTO course values('&course_id','&title','&dept_name',&credits)
new   1: INSERT INTO course values('CS_315','Robotics','Comp.Sci',3)
SQL> INSERT INTO course values('&course_id','&title','&dept_name',&credits);
Enter value for course_id: CS_319
Enter value for title: Image Processing
Enter value for dept_name: Comp.Sci
Enter value for credits: 3
old   1: INSERT INTO course values('&course_id','&title','&dept_name',&credits)
new   1: INSERT INTO course values('CS_319','Image Processing','Comp.Sci',3)

SQL> INSERT INTO course values('&course_id','&title','&dept_name',&credits);
Enter value for course_id: CS_347
Enter value for title: Database System Concepts
Enter value for dept_name: Comp.Sci
Enter value for credits: 3
old   1: INSERT INTO course values('&course_id','&title','&dept_name',&credits)
new   1: INSERT INTO course values('CS_347','Database System Concepts','Comp.Sci',3)

SQL> INSERT INTO course values('&course_id','&title','&dept_name',&credits);
Enter value for course_id: EE_181
Enter value for title: Intro.to Digital Systems
Enter value for dept_name: Elec.Eng.
Enter value for credits: 3
old   1: INSERT INTO course values('&course_id','&title','&dept_name',&credits)
new   1: INSERT INTO course values('EE_181','Intro.to Digital Systems','Elec.Eng.',3)

SQL> INSERT INTO course values('&course_id','&title','&dept_name',&credits);
Enter value for course_id: FIN_201
Enter value for title: Investment Banking
Enter value for dept_name: Finance
Enter value for credits: 3
old   1: INSERT INTO course values('&course_id','&title','&dept_name',&credits)
new   1: INSERT INTO course values('FIN_201','Investment Banking','Finance',3)
```

```
SQL> INSERT INTO course values('&course_id','&title','&dept_name',&credits);
Enter value for course_id: HIS_351
Enter value for title: World History
Enter value for dept_name: History
Enter value for credits: 3
old   1: INSERT INTO course values('&course_id','&title','&dept_name',&credits)
new   1: INSERT INTO course values('HIS_351','World History','History',3)

SQL> INSERT INTO course values('&course_id','&title','&dept_name',&credits);
Enter value for course_id: MU_199
Enter value for title: Music Video Producation
Enter value for dept_name: Music
Enter value for credits: 3
old   1: INSERT INTO course values('&course_id','&title','&dept_name',&credits)
new   1: INSERT INTO course values('MU_199','Music Video Producation','Music',3)

SQL> INSERT INTO course values('&course_id','&title','&dept_name',&credits);
Enter value for course_id: PHY_101
Enter value for title: Physical Principles
Enter value for dept_name: Physics
Enter value for credits: 4
old   1: INSERT INTO course values('&course_id','&title','&dept_name',&credits)
new   1: INSERT INTO course values('PHY_101','Physical Principles','Physics',4)

SQL> INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary);
Enter value for id: 10101
Enter value for name: Srinivasan
Enter value for dept_name: Comp.Sci
Enter value for salary: 65000
old   1: INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary)
new   1: INSERT INTO instructor values (10101 , 'Srinivasan' ,'Comp.Sci' ,65000)

SQL> INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary);
Enter value for id: 12121
Enter value for name: Wu
Enter value for dept_name: Finance
Enter value for salary: 90000
old   1: INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary)
new   1: INSERT INTO instructor values (12121 , 'Wu' ,'Finance' ,90000)

SQL> INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary);
Enter value for id: 15151
Enter value for name: Mozart
Enter value for dept_name: Music
Enter value for salary: 40000
old   1: INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary)
new   1: INSERT INTO instructor values (15151 , 'Mozart' ,'Music' ,40000)

SQL> INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary);
Enter value for id: 22222
Enter value for name: Einstein
Enter value for dept_name: Physics
Enter value for salary: 95000
old   1: INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary)
new   1: INSERT INTO instructor values (22222 , 'Einstein' ,'Physics' ,95000)

SQL> INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary);
Enter value for id: 32343
Enter value for name: El Said
Enter value for dept_name: History
Enter value for salary: 60000
old   1: INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary)
new   1: INSERT INTO instructor values (32343 , 'El Said' ,'History' ,60000)

SQL> INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary);
Enter value for id: 33456
Enter value for name: Gold
Enter value for dept_name: Physics
Enter value for salary: 87000
old   1: INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary)
new   1: INSERT INTO instructor values (33456 , 'Gold' ,'Physics' ,87000)

SQL> INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary);
Enter value for id: 45565
Enter value for name: Katz
Enter value for dept_name: Comp.Sci
Enter value for salary: 75000
old   1: INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary)
new   1: INSERT INTO instructor values (45565 , 'Katz' ,'Comp.Sci' ,75000)

SQL> INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary);
Enter value for id: 58583
Enter value for name: Califieri
Enter value for dept_name: History
Enter value for salary: 62000
old   1: INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary)
new   1: INSERT INTO instructor values (58583 , 'Califieri' ,'History' ,62000)

SQL> INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary);
Enter value for id: 76543
Enter value for name: Singh
Enter value for dept_name: Finance
Enter value for salary: 80000
old   1: INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary)
new   1: INSERT INTO instructor values (76543 , 'Singh' ,'Finance' ,80000)

SQL> INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary);
Enter value for id: 76766
Enter value for name: Crick
Enter value for dept_name: Biology
Enter value for salary: 72000
old   1: INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary)
new   1: INSERT INTO instructor values (76766 , 'Crick' ,'Biology' ,72000)
```

```
SQL> INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary);
Enter value for id: 83821
Enter value for name: Brandt
Enter value for dept_name: Comp.Sci
Enter value for salary: 92000
old   1: INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary)
new   1: INSERT INTO instructor values (83821 , 'Brandt' ,'Comp.Sci' ,92000)

SQL> INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary);
Enter value for id: 98345
Enter value for name: Kim
Enter value for dept_name: Elec.Eng.
Enter value for salary: 80000
old   1: INSERT INTO instructor values (&ID , '&name' ,'&dept_name' ,&salary)
new   1: INSERT INTO instructor values (98345 , 'Kim' ,'Elec.Eng.' ,80000)

SQL> INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred);
Enter value for id: 00128
Enter value for name: Zhang
Enter value for dept_name: Comp.Sci
Enter value for tot_cred: 102
old   1: INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred)
new   1: INSERT INTO student values(00128 , 'Zhang' ,'Comp.Sci',102)

SQL> INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred);
Enter value for id: 12345
Enter value for name: Shankar
Enter value for dept_name: Comp.Sci
Enter value for tot_cred: 32
old   1: INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred)
new   1: INSERT INTO student values(12345 , 'Shankar' ,'Comp.Sci',32)

SQL> INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred);
Enter value for id: 19991
Enter value for name: Brandt
Enter value for dept_name: History
Enter value for tot_cred: 80
old   1: INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred)
new   1: INSERT INTO student values(19991 , 'Brandt' ,'History',80)

SQL> INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred);
Enter value for id: 23121
Enter value for name: Chavez
Enter value for dept_name: Finance
Enter value for tot_cred: 110
old   1: INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred)
new   1: INSERT INTO student values(23121 , 'Chavez' ,'Finance',110)

SQL> INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred);
Enter value for id: 44553
Enter value for name: Peltier
Enter value for dept_name: Physics
Enter value for tot_cred: 56
old   1: INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred)
new   1: INSERT INTO student values(44553 , 'Peltier' ,'Physics',56)

SQL> INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred);
Enter value for id: 45678
Enter value for name: Levy
Enter value for dept_name: Physics
Enter value for tot_cred: 46
old   1: INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred)
new   1: INSERT INTO student values(45678 , 'Levy' ,'Physics',46)

SQL> INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred);
Enter value for id: 54321
Enter value for name: Williams
Enter value for dept_name: Comp.Sci
Enter value for tot_cred: 54
old   1: INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred)
new   1: INSERT INTO student values(54321 , 'Williams' ,'Comp.Sci',54)

SQL> INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred);
Enter value for id: 55739
Enter value for name: Sanchez
Enter value for dept_name: Music
Enter value for tot_cred: 38
old   1: INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred)
new   1: INSERT INTO student values(55739 , 'Sanchez' ,'Music',38)

SQL> INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred);
Enter value for id: 70557
Enter value for name: Snow
Enter value for dept_name: Physics
Enter value for tot_cred: 0
old   1: INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred)
new   1: INSERT INTO student values(70557 , 'Snow' ,'Physics',0)

SQL> INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred);
Enter value for id: 76543
Enter value for name: Brown
Enter value for dept_name: Comp.Sci
Enter value for tot_cred: 58
old   1: INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred)
new   1: INSERT INTO student values(76543 , 'Brown' ,'Comp.Sci',58)

SQL> INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred);
Enter value for id: 76653
Enter value for name: Aoi
Enter value for dept_name: Elec.Eng.
Enter value for tot_cred: 60
old   1: INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred)
new   1: INSERT INTO student values(76653 , 'Aoi' ,'Elec.Eng.',60)
```

```
SQL> INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred);
Enter value for id: 98765
Enter value for name: Bourikas
Enter value for dept_name: Elec.Eng.
Enter value for tot_cred: 98
old   1: INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred)
new   1: INSERT INTO student values(98765 , 'Bourikas' ,'Elec.Eng.',98)

SQL> INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred);
Enter value for id: 98988
Enter value for name: Tanaka
Enter value for dept_name: Biology
Enter value for tot_cred: 120
old   1: INSERT INTO student values(&ID , '&name' ,'&dept_name',&tot_cred)
new   1: INSERT INTO student values(98988 , 'Tanaka' ,'Biology',120)

SQL> INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number);
Enter value for course_id: BIO_101
Enter value for sec_id: 1
Enter value for semester: Summer
Enter value for year: 2009
Enter value for building: Painter
Enter value for room_number: 514
old   1: INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number)
new   1: INSERT INTO section values('BIO_101',1,'Summer',2009,'Painter',514)

SQL> INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number);
Enter value for course_id: BIO_301
Enter value for sec_id: 1
Enter value for semester: Summer
Enter value for year: 2010
Enter value for building: Painter
Enter value for room_number: 514
old   1: INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number)
new   1: INSERT INTO section values('BIO_301',1,'Summer',2010,'Painter',514)

SQL> INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number);
Enter value for course_id: CS_101
Enter value for sec_id: 1
Enter value for semester: Fall
Enter value for year: 2009
Enter value for building: Packard
Enter value for room_number: 101
old   1: INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number)
new   1: INSERT INTO section values('CS_101',1,'Fall',2009,'Packard',101)
```

```
SQL> INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number);
Enter value for course_id: CS_101
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
Enter value for building: Packard
Enter value for room_number: 101
old   1: INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number)
new   1: INSERT INTO section values('CS_101',1,'Spring',2010,'Packard',101)

SQL> INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number);
Enter value for course_id: CS_190
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2009
Enter value for building: Taylor
Enter value for room_number: 3128
old   1: INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number)
new   1: INSERT INTO section values('CS_190',1,'Spring',2009,'Taylor',3128)

SQL> INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number);
Enter value for course_id: CS_190
Enter value for sec_id: 2
Enter value for semester: Spring
Enter value for year: 2009
Enter value for building: Taylor
Enter value for room_number: 3128
old   1: INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number)
new   1: INSERT INTO section values('CS_190',2,'Spring',2009,'Taylor',3128)

SQL> INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number);
Enter value for course_id: CS_315
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
Enter value for building: Watson
Enter value for room_number: 120
old   1: INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number)
new   1: INSERT INTO section values('CS_315',1,'Spring',2010,'Watson',120)

SQL> INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number);
Enter value for course_id: CS_319
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
Enter value for building: Watson
Enter value for room_number: 100
old   1: INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number)
new   1: INSERT INTO section values('CS_319',1,'Spring',2010,'Watson',100)

SQL> INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number);
Enter value for course_id: CS_319
Enter value for sec_id: 2
Enter value for semester: Spring
Enter value for year: 2010
Enter value for building: Taylor
Enter value for room_number: 3128
old   1: INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number)
new   1: INSERT INTO section values('CS_319',2,'Spring',2010,'Taylor',3128)

SQL> INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number);
Enter value for course_id: CS_347
Enter value for sec_id: 1
Enter value for semester: Fall
Enter value for year: 2009
Enter value for building: Taylor
Enter value for room_number: 3128
old   1: INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number)
new   1: INSERT INTO section values('CS_347',1,'Fall',2009,'Taylor',3128)

SQL> INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number);
Enter value for course_id: EE_181
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2009
Enter value for building: Taylor
Enter value for room_number: 3128
old   1: INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number)
new   1: INSERT INTO section values('EE_181',1,'Spring',2009,'Taylor',3128)

SQL> INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number);
Enter value for course_id: FIN_201
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
Enter value for building: Packard
Enter value for room_number: 101
```

```
old   1: INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number)
new   1: INSERT INTO section values('FIN_201',1,'Spring',2010,'Packard',101)

SQL> INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number);
Enter value for course_id: HIS_351
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
Enter value for building: Painter
Enter value for room_number: 514
old   1: INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number)
new   1: INSERT INTO section values('HIS_351',1,'Spring',2010,'Painter',514)

SQL> INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number);
Enter value for course_id: MU_199
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
Enter value for building: Packard
Enter value for room_number: 101
old   1: INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number)
new   1: INSERT INTO section values('MU_199',1,'Spring',2010,'Packard',101)

SQL> INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number);
Enter value for course_id: PHY_101
Enter value for sec_id: 1
Enter value for semester: Fall
Enter value for year: 2009
Enter value for building: Watson
Enter value for room_number: 100
old   1: INSERT INTO section
values('&course_id',&sec_id,'&semester',&year,'&building',&room_number)
new   1: INSERT INTO section values('PHY_101',1,'Fall',2009,'Watson',100)

SQL> INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year);
Enter value for id: 10101
Enter value for course_id: CS_101
Enter value for sec_id: 1
Enter value for semester: Fall
Enter value for year: 2009
old   1: INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year)
new   1: INSERT INTO teaches values(10101,'CS_101',1,'Fall',2009)

SQL> INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year);
Enter value for id: 10101
Enter value for course_id: CS_315
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
old   1: INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year)
new   1: INSERT INTO teaches values(10101,'CS_315',1,'Spring',2010)

SQL> INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year);
Enter value for id: 10101
Enter value for course_id: CS_347
Enter value for sec_id: 1
Enter value for semester: Fall
Enter value for year: 2009
old   1: INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year)
new   1: INSERT INTO teaches values(10101,'CS_347',1,'Fall',2009)

SQL> INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year);
Enter value for id: 12121
Enter value for course_id: FIN_201
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
old   1: INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year)
new   1: INSERT INTO teaches values(12121,'FIN_201',1,'Spring',2010)

SQL> INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year);
Enter value for id: 15151
Enter value for course_id: MU_199
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
old   1: INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year)
new   1: INSERT INTO teaches values(15151,'MU_199',1,'Spring',2010)

SQL> INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year);
Enter value for id: 22222
Enter value for course_id: PHY_101
Enter value for sec_id: 1
Enter value for semester: Fall
Enter value for year: 2009
old   1: INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year)
new   1: INSERT INTO teaches values(22222,'PHY_101',1,'Fall',2009)

SQL> INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year);
Enter value for id: 32343
Enter value for course_id: HIS_351
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
old   1: INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year)
new   1: INSERT INTO teaches values(32343,'HIS_351',1,'Spring',2010)

SQL> INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year);
```

```
Enter value for id: 45565
Enter value for course_id: CS_101
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
old   1: INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year)
new   1: INSERT INTO teaches values(45565,'CS_101',1,'Spring',2010)

SQL> INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year);
Enter value for id: 45565
Enter value for course_id: CS_319
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
old   1: INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year)
new   1: INSERT INTO teaches values(45565,'CS_319',1,'Spring',2010)

SQL> INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year);
Enter value for id: 76766
Enter value for course_id: BIO_101
Enter value for sec_id: 1
Enter value for semester: Summer
Enter value for year: 2009
old   1: INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year)
new   1: INSERT INTO teaches values(76766,'BIO_101',1,'Summer',2009)

SQL> INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year);
Enter value for id: 76766
Enter value for course_id: BIO_301
Enter value for sec_id: 1
Enter value for semester: Summer
Enter value for year: 2010
old   1: INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year)
new   1: INSERT INTO teaches values(76766,'BIO_301',1,'Summer',2010)

SQL> INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year);
Enter value for id: 83821
Enter value for course_id: CS_190
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2009
old   1: INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year)
new   1: INSERT INTO teaches values(83821,'CS_190',1,'Spring',2009)

SQL> INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year);
Enter value for id: 83821
Enter value for course_id: CS_190
Enter value for sec_id: 2
Enter value for semester: Spring
Enter value for year: 2009
old   1: INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year)
new   1: INSERT INTO teaches values(83821,'CS_190',2,'Spring',2009)
```

```
SQL> INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year);
Enter value for id: 83821
Enter value for course_id: CS_319
Enter value for sec_id: 2
Enter value for semester: Spring
Enter value for year: 2010
old   1: INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year)
new   1: INSERT INTO teaches values(83821,'CS_319',2,'Spring',2010)

SQL> INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year);
Enter value for id: 98345
Enter value for course_id: EE_181
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2009
old   1: INSERT INTO teaches values(&ID,'&course_id',&sec_id,'&semester',&year)
new   1: INSERT INTO teaches values(98345,'EE_181',1,'Spring',2009)

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 00128
Enter value for course_id: CS_101
Enter value for sec_id: 1
Enter value for semester: Fall
Enter value for year: 2009
Enter value for grade: A
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(00128,'CS_101',1,'Fall',2009,'A')

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 00128
Enter value for course_id: CS_347
Enter value for sec_id: 1
Enter value for semester: Fall
Enter value for year: 2009
Enter value for grade: A-
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(00128,'CS_347',1,'Fall',2009,'A-')

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 12345
Enter value for course_id: CS_101
Enter value for sec_id: 1
Enter value for semester: Fall
Enter value for year: 2009
Enter value for grade: c
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(12345,'CS_101',1,'Fall',2009,'c')

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 12345
Enter value for course_id: CS_190
Enter value for sec_id: 2
Enter value for semester: Spring
Enter value for year: 2009
Enter value for grade: A
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(12345,'CS_190',2,'Spring',2009,'A')

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 12345
Enter value for course_id: CS_315
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
Enter value for grade: A
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(12345,'CS_315',1,'Spring',2010,'A')

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 12345
Enter value for course_id: CS_347
Enter value for sec_id: 1
Enter value for semester: Fall
Enter value for year: 2009
Enter value for grade: A
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(12345,'CS_347',1,'Fall',2009,'A')

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 19991
Enter value for course_id: HIS_351
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
Enter value for grade: B
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(19991,'HIS_351',1,'Spring',2010,'B')

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 23121
Enter value for course_id: FIN_201
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
Enter value for grade: C+
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(23121,'FIN_201',1,'Spring',2010,'C+')

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 44553
Enter value for course_id: PHY_101
Enter value for sec_id: 1
Enter value for semester: Fall
Enter value for year: 2009
Enter value for grade: B-
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(44553,'PHY_101',1,'Fall',2009,'B-')

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 45678
Enter value for course_id: CS_101
Enter value for sec_id: 1
Enter value for semester: Fall
Enter value for year: 2009
Enter value for grade: F
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(45678,'CS_101',1,'Fall',2009,'F')

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 45678
Enter value for course_id: CS_101
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
Enter value for grade: B+
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(45678,'CS_101',1,'Spring',2010,'B+')

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 45678
Enter value for course_id: CS_319
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
Enter value for grade: B
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(45678,'CS_319',1,'Spring',2010,'B')

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 54321
Enter value for course_id: CS_101
Enter value for sec_id: 1
Enter value for semester: Fall
Enter value for year: 2009
Enter value for grade: A-
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(54321,'CS_101',1,'Fall',2009,'A-')

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 54321
Enter value for course_id: CS_190
Enter value for sec_id: 2
Enter value for semester: Spring
Enter value for year: 2009
Enter value for grade: B+
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(54321,'CS_190',2,'Spring',2009,'B+')

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 55739
Enter value for course_id: MU_199
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
Enter value for grade: A-
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(55739,'MU_199',1,'Spring',2010,'A-')

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 76543
Enter value for course_id: CS_101
Enter value for sec_id: 1
Enter value for semester: Fall
Enter value for year: 2009
Enter value for grade: A
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(76543,'CS_101',1,'Fall',2009,'A')

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 76543
Enter value for course_id: CS_319
Enter value for sec_id: 2
Enter value for semester: Spring
Enter value for year: 2010
Enter value for grade: A
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(76543,'CS_319',2,'Spring',2010,'A')

SQL> INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 76653
Enter value for course_id: EE_181
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2009
Enter value for grade: C
old   1: INSERT INTO takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: INSERT INTO takes values(76653,'EE_181',1,'Spring',2009,'C')

SQL> insert into takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 98765
Enter value for course_id: CS_101
Enter value for sec_id: 1
Enter value for semester: Fall
Enter value for year: 2009
Enter value for grade: C-
old   1: insert into takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: insert into takes values(98765,'CS_101',1,'Fall',2009,'C-')
```

```
SQL> insert into takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 98765
Enter value for course_id: CS_315
Enter value for sec_id: 1
Enter value for semester: Spring
Enter value for year: 2010
Enter value for grade: B
old   1: insert into takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: insert into takes values(98765,'CS_315',1,'Spring',2010,'B')

SQL> insert into takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 98988
Enter value for course_id: BIO_101
Enter value for sec_id: 1
Enter value for semester: Summer
Enter value for year: 2009
Enter value for grade: A
old   1: insert into takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: insert into takes values(98988,'BIO_101',1,'Summer',2009,'A')

SQL> insert into takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade');
Enter value for id: 98988
Enter value for course_id: BIO_301
Enter value for sec_id: 1
Enter value for semester: Summer
Enter value for year: 2010
Enter value for grade: null
old   1: insert into takes values(&ID,'&course_id',&sec_id,'&semester',&year,'&grade')
new   1: insert into takes values(98988,'BIO_301',1,'Summer',2010,'null')
```

OUTPUT

```
SQL> SELECT * FROM classroom;

BUILDING   ROOM_NUMBER   CAPACITY
---------- ----------- ----------
Packard           101        500
Painter          514        10
Taylor          3128         70
Watson            100         30
Watson            120         50

SQL> SELECT * FROM department;

DEPT_NAME                    BUILDING               BUDGET
----------------------------- -------------------- ----------
Biology                    Watson                90000
Comp.Sci                     Taylor              100000
Elec.Eng.                   Taylor               85000
Finance                    Painter             120000
History                    Painter             50000
Music                      Packard               80000
```

```
Physics              Watson               70000

7 rows selected.

SQL> SELECT * FROM course;

COURSE_ID            TITLE
-------------------- --------------------------------------------------
DEPT_NAME               CREDITS
-------------------- ----------
BIO_101              Intro. to Biology
Biology                       4

BIO_301              Genetics
Biology                       4

BIO_399              Computational Biology
Biology                       3


COURSE_ID            TITLE
-------------------- --------------------------------------------------
DEPT_NAME               CREDITS
-------------------- ----------
CS_101               Intro. to Computer Science
Comp.Sci                      4

CS_190               Game Design
Comp.Sci                      4

CS_315               Robotics
Comp.Sci                      3


COURSE_ID            TITLE
-------------------- --------------------------------------------------
DEPT_NAME               CREDITS
-------------------- ----------
CS_319               Image Processing
Comp.Sci                      3

CS_347               Database System Concepts
Comp.Sci                      3

EE_181               Intro.to Digital Systems
Elec.Eng.                     3


COURSE_ID            TITLE
-------------------- --------------------------------------------------
DEPT_NAME               CREDITS
-------------------- ----------
```

```
FIN_201              Investment Banking
Finance                       3

HIS_351              World History
History                       3

MU_199               Music Video Producation
Music                         3


COURSE_ID            TITLE
-------------------- --------------------------------------------------
DEPT_NAME               CREDITS
-------------------- ----------
PHY_101              Physical Principles
Physics                       4


13 rows selected.

SQL> Commit;

Commit complete.

SQL> SELECT * FROM course;

COURSE_ID            TITLE
-------------------- --------------------------------------------------
DEPT_NAME               CREDITS
-------------------- ----------
BIO_101              Intro. to Biology
Biology                       4

BIO_301              Genetics
Biology                       4

BIO_399              Computational Biology
Biology                       3


COURSE_ID            TITLE
-------------------- --------------------------------------------------
DEPT_NAME               CREDITS
-------------------- ----------
CS_101               Intro. to Computer Science
Comp.Sci                      4

CS_190               Game Design
Comp.Sci                      4

CS_315               Robotics
Comp.Sci                      3
```

```
COURSE_ID          TITLE
-------------------- --------------------------------------------------
DEPT_NAME            CREDITS
-------------------- ----------
CS_319             Image Processing
Comp.Sci                  3

CS_347             Database System Concepts
Comp.Sci                  3

EE_181             Intro.to Digital Systems
Elec.Eng.                 3


COURSE_ID          TITLE
-------------------- --------------------------------------------------
DEPT_NAME            CREDITS
-------------------- ----------
FIN_201            Investment Banking
Finance                   3

HIS_351            World History
History                   3

MU_199             Music Video Producation
Music                     3


COURSE_ID          TITLE
-------------------- --------------------------------------------------
DEPT_NAME            CREDITS
-------------------- ----------
PHY_101            Physical Principles
Physics                   4


13 rows selected.

SQL> SELECT * FROM instructor;

      ID NAME                 DEPT_NAME            SALARY
---------- ------------------------------ -------------------- ----------
     10101 Srinivasan          Comp.Sci             65000
     12121 Wu                  Finance              90000
     15151 Mozart              Music                40000
     22222 Einstein            Physics              95000
     32343 El Said             History              60000
     33456 Gold                Physics              87000
     45565 Katz                Comp.Sci             75000
     58583 Califieri           History              62000
     76543 Singh                Finance              80000
     76766 Crick                Biology              72000
     83821 Brandt               Comp.Sci             92000

        ID NAME                 DEPT_NAME            SALARY
---------- -------------------- -------------------- ----------
     98345 Kim                  Elec.Eng.            80000

12 rows selected.

SQL> SELECT * FROM student;

        ID NAME            DEPT_NAME            TOT_CRED
---------- --------------- -------------------- ----------
       128 Zhang           Comp.Sci             102
     12345 Shankar         Comp.Sci             32
     19991 Brandt          History              80
     23121 Chavez          Finance              110
     44553 Peltier         Physics              56
     45678 Levy            Physics              46
     54321 Williams        Comp.Sci             54
     55739 Sanchez         Music                38
     70557 Snow            Physics              0
     76543 Brown           Comp.Sci             58
     76653 Aoi             Elec.Eng.            60

        ID NAME            DEPT_NAME            TOT_CRED
---------- --------------- -------------------- ----------
     98765 Bourikas        Elec.Eng.            98
     98988 Tanaka          Biology              120

13 rows selected.

SQL> SELECT * FROM section;

COURSE_ID            SEC_ID SEMESTER             YEAR
-------------------- ---------- -------------------- ----------
BUILDING             ROOM_NUMBER
-------------------- -----------
BIO_101                   1 Summer               2009
Painter              514

BIO_301                   1 Summer               2010
Painter              514

CS_101                    1 Fall                 2009
Packard              101


COURSE_ID            SEC_ID SEMESTER             YEAR
-------------------- ---------- -------------------- ----------
BUILDING             ROOM_NUMBER
```

```
-------------------- -----------
CS_101                  1 Spring              2010
Packard               101

CS_190                  1 Spring              2009
Taylor               3128

CS_190                  2 Spring              2009
Taylor               3128


COURSE_ID              SEC_ID SEMESTER              YEAR
-------------------- ---------- -------------------- ----------
BUILDING             ROOM_NUMBER
-------------------- -----------
CS_315                  1 Spring              2010
Watson                120

CS_319                  1 Spring              2010
Watson                100

CS_319                  2 Spring              2010
Taylor               3128


COURSE_ID              SEC_ID SEMESTER              YEAR
-------------------- ---------- -------------------- ----------
BUILDING             ROOM_NUMBER
-------------------- -----------
CS_347                  1 Fall                2009
Taylor               3128

EE_181                  1 Spring              2009
Taylor               3128

FIN_201                 1 Spring              2010
Packard               101


COURSE_ID              SEC_ID SEMESTER              YEAR
-------------------- ---------- -------------------- ----------
BUILDING             ROOM_NUMBER
-------------------- -----------
HIS_351                 1 Spring              2010
Painter               514

MU_199                  1 Spring              2010
Packard               101

PHY_101                 1 Fall                2009
Watson                100
```

```
15 rows selected.

SQL> SELECT * FROM teaches;

        ID COURSE_ID                SEC_ID SEMESTER                  YEAR
---------- -------------------- ---------- -------------------- ----------
     10101 CS_101                        1 Fall                       2009
     10101 CS_315                        1 Spring                     2010
     10101 CS_347                        1 Fall                       2009
     12121 FIN_201                       1 Spring                     2010
     15151 MU_199                        1 Spring                     2010
     22222 PHY_101                       1 Fall                       2009
     32343 HIS_351                       1 Spring                     2010
     45565 CS_101                        1 Spring                     2010
     45565 CS_319                        1 Spring                     2010
     76766 BIO_101                       1 Summer                     2009
     76766 BIO_301                       1 Summer                     2010

        ID COURSE_ID                SEC_ID SEMESTER                  YEAR
---------- -------------------- ---------- -------------------- ----------
     83821 CS_190                        1 Spring                     2009
     83821 CS_190                        2 Spring                     2009
     83821 CS_319                        2 Spring                     2010
     98345 EE_181                        1 Spring                     2009

15 rows selected.

SQL> select * from takes;

        ID COURSE_ID                SEC_ID SEMESTER                  YEAR
---------- -------------------- ---------- -------------------- ----------
GRADE
----------
       128 CS_101                        1 Fall                       2009
A

       128 CS_347                        1 Fall                       2009
A-

     12345 CS_101                        1 Fall                       2009
c


        ID COURSE_ID                SEC_ID SEMESTER                  YEAR
---------- -------------------- ---------- -------------------- ----------
GRADE
----------
     12345 CS_190                        2 Spring                     2009
A

     12345 CS_315                        1 Spring                     2010
```

```
A

     12345 CS_347                  1 Fall                 2009
A


        ID COURSE_ID              SEC_ID SEMESTER               YEAR
---------- -------------------- ---------- -------------------- ----------
GRADE
----------
     19991 HIS_351                  1 Spring               2010
B

     23121 FIN_201                  1 Spring               2010
C+

     44553 PHY_101                  1 Fall                 2009
B-


        ID COURSE_ID              SEC_ID SEMESTER               YEAR
---------- -------------------- ---------- -------------------- ----------
GRADE
----------
     45678 CS_101                   1 Fall                 2009
F

     45678 CS_101                   1 Spring               2010
B+

     45678 CS_319                   1 Spring               2010
B


        ID COURSE_ID              SEC_ID SEMESTER               YEAR
---------- -------------------- ---------- -------------------- ----------
GRADE
----------
     54321 CS_101                   1 Fall                 2009
A-

     54321 CS_190                   2 Spring               2009
B+

     55739 MU_199                   1 Spring               2010
A-


        ID COURSE_ID              SEC_ID SEMESTER               YEAR
---------- -------------------- ---------- -------------------- ----------
GRADE
----------
```

```
     76543 CS_101                   1 Fall                 2009
A

     76543 CS_319                   2 Spring               2010
A

     76653 EE_181                   1 Spring               2009
C


        ID COURSE_ID            SEC_ID SEMESTER              YEAR
---------- -------------------- ---------- -------------------- ----------
GRADE
----------
     98765 CS_101                   1 Fall                 2009
C-

     98765 CS_315                   1 Spring               2010
B

     98988 BIO_101                  1 Summer               2009
A


        ID COURSE_ID            SEC_ID SEMESTER              YEAR
---------- -------------------- ---------- -------------------- ----------
GRADE
----------
     98988 BIO_301                  1 Summer               2010
null


22 rows selected.
```

2.QUESTION

Modify the table

- Student such that change the name of student as ‘Mozart’ whose Id is 45678
- Department such that budget of history department is 100000 and building as Taylor
- Instructor such that the salary of each instructor increases by 10%
- Course such that the credits of all courses under computer science department is 4
- Student such that additional 10 points should be given to the total credits of students who have total credits in between 20 to 50
- Instructor such that a salary raise of 5% to be given to instructors whose salary is less than the average salary.
- Teaches such that for the course with course id ‘CS-101’ which is under ‘Fall’ semester , change year of the course to 2010

QUERY

```
SQL> update student set name='Mozart' where ID=45678;
SQL> update department set budget=100000,building='Taylor' where dept_name ='History';
SQL> update instructor set salary=1.1*salary;
SQL> update course set credits=4 where dept_name='Comp.Sci';
SQL> update student set tot_cred=tot_cred+10 where tot_cred>20 and tot_cred<50;
SQL> update instructor set salary=1.05*salary where salary<(select avg(salary) from instructor );
SQL> update teaches set year=2010 where course_id='CS_101' and semester ='Fall';
```

OUTPUT

```
SQL> select * from student;

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
       128 Zhang                Comp.Sci                    102 24-JUL-03
     12345 Shankar              Comp.Sci                     32 01-JAN-01
     19991 Brandt               History                      80 10-JUN-02
     23121 Chavez               Finance                     110 25-MAY-03
     44553 Peltier              Physics                      56 30-APR-04
     45678 Mozart               Physics                      46 27-AUG-99
     54321 Williams             Comp.Sci                     54 01-OCT-06
     55739 Sanchez              Music                        38 07-FEB-95
     70557 Snow                 Physics                       0 08-APR-08
     76543 Brown                Comp.Sci                     58 09-MAR-09
     76653 Aoi                  Elec.Eng.                    60 10-MAY-10

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
     98765 Bourikas             Elec.Eng.                    98 07-JUN-11
     98988 Tanaka               Biology                     120 08-APR-12

SQL> select * from department;

DEPT_NAME                      BUILDING                 BUDGET
------------------------------ -------------------- ----------
Biology                        Watson                    90000
Comp.Sci                       Taylor                   100000
Elec.Eng.                      Taylor                    85000
Finance                        Painter                  120000
History                        Taylor                   100000
Music                          Packard                   80000
Physics                        Watson                    70000

SQL> select * from instructor;

        ID NAME                           DEPT_NAME                SALARY
---------- ------------------------------ -------------------- ----------
     10101 Srinivasan                     Comp.Sci                  71500
     12121 Wu                             Finance                   99000
     15151 Mozart                         Music                     44000
     22222 Einstein                       Physics                  104500
     32343 El Said                        History                   66000
     33456 Gold                           Physics                   95700
     45565 Katz                           Comp.Sci                  82500
     58583 Califieri                      History                   68200
     76543 Singh                          Finance                   88000
     76766 Crick                          Biology                   79200
     83821 Brandt                         Comp.Sci                 101200

        ID NAME                           DEPT_NAME                SALARY
---------- ------------------------------ -------------------- ----------
     98345 Kim                            Elec.Eng.                 88000

SQL> select * from course;

COURSE_ID            TITLE
-------------------- --------------------------------------------------
DEPT_NAME               CREDITS
-------------------- ----------
BIO_101              Intro. to Biology
Biology                       4

BIO_301              Genetics
Biology                       4

BIO_399              Computational Biology
Biology                       3


COURSE_ID            TITLE
-------------------- --------------------------------------------------
DEPT_NAME               CREDITS
-------------------- ----------
CS_101               Intro. to Computer Science
Comp.Sci                      4

CS_190               Game Design
Comp.Sci                      4

CS_315               Robotics
Comp.Sci                      4


COURSE_ID            TITLE
-------------------- --------------------------------------------------
DEPT_NAME               CREDITS
-------------------- ----------
CS_319               Image Processing
Comp.Sci                      4

CS_347               Database System Concepts
Comp.Sci                      4

EE_181               Intro.to Digital Systems
Elec.Eng.                     3


COURSE_ID            TITLE
-------------------- --------------------------------------------------
```

```
DEPT_NAME              CREDITS
-------------------- ----------
FIN_201            Investment Banking
Finance                    3

HIS_351            World History
History                    3

MU_199             Music Video Producation
Music                      3


COURSE_ID          TITLE
-------------------- --------------------------------------------------
DEPT_NAME              CREDITS
-------------------- ----------
PHY_101            Physical Principles
Physics                    4

SQL> select * from student;

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
       128 Zhang                Comp.Sci                    102 24-JUL-03
     12345 Shankar              Comp.Sci                     42 01-JAN-01
     19991 Brandt               History                      80 10-JUN-02
     23121 Chavez               Finance                     110 25-MAY-03
     44553 Peltier              Physics                      56 30-APR-04
     45678 Mozart               Physics                      56 27-AUG-99
     54321 Williams             Comp.Sci                     54 01-OCT-06
     55739 Sanchez              Music                        48 07-FEB-95
     70557 Snow                 Physics                       0 08-APR-08
     76543 Brown                Comp.Sci                     58 09-MAR-09
     76653 Aoi                  Elec.Eng.                    60 10-MAY-10

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
     98765 Bourikas             Elec.Eng.                    98 07-JUN-11
     98988 Tanaka               Biology                     120 08-APR-12

SQL> select * from instructor;

        ID NAME                           DEPT_NAME                SALARY
---------- ------------------------------ -------------------- ----------
     10101 Srinivasan                     Comp.Sci                  75075
     12121 Wu                             Finance                   99000
     15151 Mozart                         Music                     46200
     22222 Einstein                       Physics                  104500
     32343 El Said                        History                   69300
     33456 Gold                           Physics                   95700
     45565 Katz                           Comp.Sci                  82500
     58583 Califieri                      History                   71610
     76543 Singh                    Finance              88000
     76766 Crick                    Biology              83160
     83821 Brandt                   Comp.Sci             101200

        ID NAME                     DEPT_NAME            SALARY
---------- ------------------------------ -------------------- ----------
     98345 Kim                      Elec.Eng.            88000
```

```
SQL> select * from teaches;

        ID COURSE_ID            SEC_ID SEMESTER             YEAR
---------- -------------------- ---------- -------------------- ----------
     10101 CS_101                    1 Fall                 2010
     10101 CS_315                    1 Spring               2010
     10101 CS_347                    1 Fall                 2009
     12121 FIN_201                   1 Spring               2010
     15151 MU_199                    1 Spring               2010
     22222 PHY_101                   1 Fall                 2009
     32343 HIS_351                   1 Spring               2010
     45565 CS_101                    1 Spring               2010
     45565 CS_319                    1 Spring               2010
     76766 BIO_101                   1 Summer               2009
     76766 BIO_301                   1 Summer               2010

        ID COURSE_ID            SEC_ID SEMESTER             YEAR
---------- -------------------- ---------- -------------------- ----------
     83821 CS_190                    1 Spring               2009
     83821 CS_190                    2 Spring               2009
     83821 CS_319                    2 Spring               2010
     98345 EE_181                    1 Spring               2009
```

3.QUESTION

Delete all the student information whose total credits is zero.

QUERY

```
SQL> delete from student where tot_cred=0;
```

OUTPUT

```
SQL> select * from student;

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
       128 Zhang                Comp.Sci                    102 24-JUL-03
     12345 Shankar              Comp.Sci                     42 01-JAN-01
     19991 Brandt               History                      80 10-JUN-02
     23121 Chavez               Finance                     110 25-MAY-03
     44553 Peltier              Physics                      56 30-APR-04
     45678 Mozart               Physics                      56 27-AUG-99
     54321 Williams             Comp.Sci                     54 01-OCT-06
     55739 Sanchez              Music                        48 07-FEB-95
     76543 Brown                Comp.Sci                     58 09-MAR-09
     76653 Aoi                  Elec.Eng.                    60 10-MAY-10
     98765 Bourikas            Elec.Eng.                 98 07-JUN-11

        ID NAME              DEPT_NAME            TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
     98988 Tanaka              Biology                  120 08-APR-12
        ID NAME                        DEPT_NAME              SALARY
---------- ------------------------------ -------------------- ----------
     98345 Kim                         Elec.Eng.              88000
```

4.QUESTION
Delete the record from instructor whose id starts with '765'

QUERY
SQL> delete from instructor where ID LIKE '765%';

OUTPUT
SQL> select * from instructor;

```
        ID NAME                        DEPT_NAME              SALARY
---------- ------------------------------ -------------------- ----------
     10101 Srinivasan                   Comp.Sci               75075
     12121 Wu                          Finance                99000
     15151 Mozart                       Music                  46200
     22222 Einstein                     Physics               104500
     32343 El Said                      History                69300
     33456 Gold                        Physics                95700
     45565 Katz                        Comp.Sci               82500
     58583 Califieri                    History               71610
     76766 Crick                       Biology                83160
     83821 Brandt                       Comp.Sci              101200
     98345 Kim                         Elec.Eng.              88000
```

5.QUESTION
Delete all courses that have never been offered (i.e., which do not occur in the section relation)

QUERY
SQL> delete from course where course_id not in (select course_id from section);

OUTPUT
SQL> select *from course;

```
COURSE_ID           TITLE
-------------------- --------------------------------------------------
DEPT_NAME              CREDITS
-------------------- ----------
BIO_101              Intro. to Biology
Biology                      4

BIO_301              Genetics
Biology                      4

CS_101              Intro. to Computer Science
```

```
Comp.Sci                 4


COURSE_ID          TITLE
-------------------- --------------------------------------------------
DEPT_NAME              CREDITS
-------------------- ----------
CS_190              Game Design
Comp.Sci                 4

CS_315              Robotics
Comp.Sci                 4

CS_319              Image Processing
Comp.Sci                 4


COURSE_ID          TITLE
-------------------- --------------------------------------------------
DEPT_NAME              CREDITS
-------------------- ----------
CS_347              Database System Concepts
Comp.Sci                 4

EE_181              Intro.to Digital Systems
Elec.Eng.                3

FIN_201             Investment Banking
Finance                  3


COURSE_ID          TITLE
-------------------- --------------------------------------------------
DEPT_NAME              CREDITS
-------------------- ----------
HIS_351             World History
History                  3

MU_199              Music Video Producation
Music                    3

PHY_101             Physical Principles
Physics                  4
```

//BHAGYA A JAI
//ROLL NO:B21CSB18
//DQL COMMANDS

1.QUESTION
Retrieve the names of all instructors along with department names

QUERY
SQL> select name,dept_name from instructor;

OUTPUT

```
NAME                       DEPT_NAME
------------------------------ --------------------
Srinivasan                   Comp.Sci
Wu                         Finance
Mozart                      Music
Einstein                    Physics
El Said                     History
Gold                        Physics
Katz                        Comp.Sci
Califieri                   History
Crick                       Biology
Brandt                      Comp.Sci
Kim                         Elec.Eng.
```

2.QUESTION
Retrieve the semester names from teaches relation and avoid duplicates

QUERY
SQL> select semester from teaches;

OUTPUT

```
SEMESTER
--------------------
Fall
Spring
Fall
Spring
Spring
Fall
Spring
Spring
Spring
Summer
Summer

SEMESTER
--------------------
Spring
Spring
Spring
Spring
```

```
SQL> select distinct semester from teaches;

SEMESTER
--------------------
Spring
Summer
Fall
```

3.QUESTION

Retrieve the name of instructor along with 10% raise in their salary

QUERY

```
SQL> select name ,1.1*salary from instructor;
```

OUTPUT

```
NAME                          1.1*SALARY
------------------------------ ----------
Srinivasan                      82582.5
Wu                            108900
Mozart                           50820
Einstein                        114950
El Said                          76230
Gold                           105270
Katz                            90750
Califieri                        78771
Crick                           91476
Brandt                          111320
Kim                              96800
```

4.QUESTION

Retrieve the id and name of instructor whose salary is greater than 70000 and working under computer science department

QUERY

```
SQL> select ID ,name from instructor where salary>70000 and dept_name='Comp.Sci';
```

OUTPUT

```
      ID NAME
---------- ------------------------------
    10101 Srinivasan
    45565 Katz
    83821 Brandt
```

5.QUESTION

Retrieve the names of all instructors along with their department names and department building names

QUERY

```
SQL> select I.name,I.dept_name,D.building from instructor I,department D where
I.dept_name=D.dept_name;
```

OUTPUT

```
NAME                     DEPT_NAME          BUILDING
----------------------------- -------------------- --------------------
Crick                   Biology          Watson
Brandt                   Comp.Sci          Taylor
Katz                    Comp.Sci          Taylor
Srinivasan                 Comp.Sci          Taylor
Kim                     Elec.Eng.         Taylor
Wu                      Finance          Painter
El Said                  History          Taylor
Califieri                History          Taylor
Mozart                    Music           Packard
Einstein                 Physics           Watson
Gold                     Physics          Watson
```

6.QUESTION

Retrieve the instructor names and course identifiers for instructors in the computer science department

QUERY

```
SQL> select I.name,T.course_id from instructor I,teaches T where I.dept_name='Comp.Sci' and I.ID=T.ID;
```

OUTPUT

```
NAME                     COURSE_ID
----------------------------- --------------------
Srinivasan                 CS_101
Srinivasan                 CS_315
Srinivasan                 CS_347
Katz                     CS_101
Katz                     CS_319
Brandt                    CS_190
Brandt                    CS_190
Brandt                    CS_319
```

7.QUESTION

Retrieve the details of course for which title starts with ‘Intro’

QUERY

```
SQL> select * from course where title Like 'Intro.%';
```

OUTPUT

```
COURSE_ID          TITLE
-------------------- --------------------------------------------------
DEPT_NAME              CREDITS
-------------------- ----------
BIO_101            Intro. to Biology
Biology                   4

CS_101            Intro. to Computer Science
Comp.Sci                   4

EE_181            Intro.to Digital Systems
Elec.Eng.                  3
```

```
SQL> select * from course where title Like '%Comp%';

COURSE_ID          TITLE
-------------------- --------------------------------------------------
DEPT_NAME             CREDITS
-------------------- ----------
CS_101              Intro. to Computer Science
Comp.Sci                     4

SQL> select * from student where name like '___';

      ID NAME               DEPT_NAME           TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
    76653 Aoi               Elec.Eng.                 60 10-MAY-10
```

8.QUESTION

Retrieve the details of course in which title contains the substring ‘Comp’

QUERY

```
SQL> select * from course where title Like '%Comp%';
```

OUTPUT

```
COURSE_ID          TITLE
-------------------- --------------------------------------------------
DEPT_NAME             CREDITS
-------------------- ----------
CS_101              Intro. to Computer Science
Comp.Sci                     4
```

9.QUESTION

Retrieve the information of students who have exactly 3 characters in their names

QUERY

```
SQL> select * from student where name like '___';
```

OUTPUT

```
      ID NAME               DEPT_NAME           TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
    76653 Aoi               Elec.Eng.                 60 10-MAY-10
```

10.QUESTION

Retrieve the information of Instructors who have at least 3 characters in their names

QUERY

```
SQL> select * from instructor where length(name)>=3;
```

OUTPUT

```
      ID NAME                          DEPT_NAME              SALARY
---------- ------------------------------ -------------------- ----------
    10101 Srinivasan                  Comp.Sci              75075
    15151 Mozart                      Music                46200
    22222 Einstein                    Physics             104500
    32343 El Said                     History              69300
    33456 Gold                        Physics              95700
    45565 Katz                        Comp.Sci              82500
    58583 Califieri                   History              71610
    76766 Crick                       Biology              83160
    83821 Brandt                      Comp.Sci             101200
    98345 Kim                         Elec.Eng.             88000
```

//BHAGYA A JAI
//ROLL NO:B21CSB18
//AGGREGATE FUNCTIONS

1.QUESTION
Find the sum of the salaries of all instructors, the maximum salary, the minimum salary, and the average salary

QUERY
```
SQL> select sum(salary) as SUM,min(salary) as MIN,max(salary) as MAX from instructor;
```

OUTPUT
```
       SUM        MIN        MAX
---------- ---------- ----------
    916245      46200     104500
```

2.QUESTION
Find the sum of the salaries of all instructors of the 'History' department, as well as the maximum salary, the minimum salary, and the average salary in this department.

QUERY
```
SQL> select sum(salary) as SUM_History,min(salary) as MIN_History,max(salary) as
MAX_History,avg(salary) as AVG_History from instructor where dept_name='History';
```

OUTPUT
```
SUM_HISTORY MIN_HISTORY MAX_HISTORY AVG_HISTORY
----------- ----------- ----------- -----------
     140910       69300       71610       70455
```

3.QUESTION
Retrieve the total number of instructors in the institution

QUERY
```
SQL> select count(name) as Count_instructors from instructor;
```

OUTPUT
```
COUNT_INSTRUCTORS
-----------------
               11
```

4.QUESTION
Retrieve the total number of instructors in the 'Computer science' department

QUERY
```
SQL> select count(name) as Count_CSinstructors from instructor where dept_name='Comp.Sci';
```

OUTPUT
```
COUNT_CSINSTRUCTORS
-------------------
                  3
```

5.QUESTION

Count the number of distinct salary values in the database

QUERY

```
SQL> select count(distinct salary) from instructor;
```

OUTPUT

```
COUNT(DISTINCTSALARY)
---------------------
                   11
```

//BHAGYA A JAI
//ROLL NO:B21CSB18
//TCL COMMANDS

1.QUESTION
Change the name of the student with id 44553 as john
QUERY
update student set name = 'John’' where ID =44553;
OUTPUT

```
        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
       128 Zhang                Comp.Sci                    102 24-JUL-03
     12345 Shankar              Comp.Sci                     42 01-JAN-01
     19991 Brandt               History                      80 10-JUN-02
     23121 Chavez               Finance                     110 25-MAY-03
     44553 John                 Physics                      56 30-APR-04
     45678 Mozart               Physics                      56 27-AUG-99
     54321 Williams             Comp.Sci                     54 01-OCT-06
     55739 Sanchez              Music                        48 07-FEB-95
     76543 Brown                Comp.Sci                     58 09-MAR-09
     76653 Aoi                  Elec.Eng.                    60 10-MAY-10
     98765 Bourikas             Elec.Eng.                    98 07-JUN-11

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
     98988 Tanaka               Biology                     120 08-APR-12
```

2.QUESTION
Commit all the operations
QUERY
SQL> Commit;

3.QUESTION
Change the name of the student with id 44553 as Peltier
QUERY
SQL> update student set name = 'Peltier' where ID =44553;
OUTPUT
SQL> select * from student;

```
        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
       128 Zhang                Comp.Sci                    102 24-JUL-03
     12345 Shankar              Comp.Sci                     42 01-JAN-01
     19991 Brandt               History                      80 10-JUN-02
     23121 Chavez               Finance                     110 25-MAY-03
     44553 Peltier              Physics                      56 30-APR-04
     45678 Mozart               Physics                      56 27-AUG-99
     54321 Williams             Comp.Sci                     54 01-OCT-06
     55739 Sanchez              Music                        48 07-FEB-95
     76543 Brown                Comp.Sci                     58 09-MAR-09
     76653 Aoi                  Elec.Eng.                    60 10-MAY-10
     98765 Bourikas             Elec.Eng.                    98 07-JUN-11
```

```
        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
     98988 Tanaka               Biology                     120 08-APR-12
```

4.QUESTION

Rollback all the operations

QUERY

```
SQL> Rollback;
```

OUTPUT

```
SQL> select * from student;

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
       128 Zhang                Comp.Sci                    102 24-JUL-03
     12345 Shankar              Comp.Sci                     42 01-JAN-01
     19991 Brandt               History                      80 10-JUN-02
     23121 Chavez               Finance                     110 25-MAY-03
     44553 John                 Physics                      56 30-APR-04
     45678 Mozart               Physics                      56 27-AUG-99
     54321 Williams             Comp.Sci                     54 01-OCT-06
     55739 Sanchez              Music                        48 07-FEB-95
     76543 Brown                Comp.Sci                     58 09-MAR-09
     76653 Aoi                  Elec.Eng.                    60 10-MAY-10
     98765 Bourikas             Elec.Eng.                    98 07-JUN-11

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
     98988 Tanaka               Biology                     120 08-APR-12
```

5.QUESTION

Insert a new row into a table student, make 3 updates to the new row and after 2 updates set savepoints

QUERY

```
SQL> insert into student(ID,name,dept_name,tot_cred) values (12354,'A','Finance',20);
```

OUTPUT

```
SQL> select * from student;

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
       128 Zhang                Comp.Sci                    102 24-JUL-03
     12345 Shankar              Comp.Sci                     42 01-JAN-01
     19991 Brandt               History                      80 10-JUN-02
     23121 Chavez               Finance                     110 25-MAY-03
     44553 John                 Physics                      56 30-APR-04
     45678 Mozart               Physics                      56 27-AUG-99
     54321 Williams             Comp.Sci                     54 01-OCT-06
     55739 Sanchez              Music                        48 07-FEB-95
     76543 Brown                Comp.Sci                     58 09-MAR-09
     76653 Aoi                  Elec.Eng.                    60 10-MAY-10
     98765 Bourikas             Elec.Eng.                    98 07-JUN-11

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
     98988 Tanaka               Biology                     120 08-APR-12
     12354 A                    Finance                      20

SQL> savepoint update1;

Savepoint created.

SQL> update student set ID = 56312 where ID =12354;

1 row updated.

SQL> select * from student;

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
       128 Zhang                Comp.Sci                    102 24-JUL-03
     12345 Shankar              Comp.Sci                     42 01-JAN-01
     19991 Brandt               History                      80 10-JUN-02
     23121 Chavez               Finance                     110 25-MAY-03
     44553 John                 Physics                      56 30-APR-04
     45678 Mozart               Physics                      56 27-AUG-99
     54321 Williams             Comp.Sci                     54 01-OCT-06
     55739 Sanchez              Music                        48 07-FEB-95
     76543 Brown                Comp.Sci                     58 09-MAR-09
     76653 Aoi                  Elec.Eng.                    60 10-MAY-10
     98765 Bourikas             Elec.Eng.                    98 07-JUN-11

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
     98988 Tanaka               Biology                     120 08-APR-12
     56312 A                    Finance                      20

SQL> savepoint update2;
Savepoint created.
SQL> update student set name = 'ABC' where ID =56312;
1 row updated.

SQL> update student set dept_name ='History'where ID=56312;

1 row updated.

SQL> select * from student;

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
       128 Zhang                Comp.Sci                    102 24-JUL-03
     12345 Shankar              Comp.Sci                     42 01-JAN-01
     19991 Brandt               History                      80 10-JUN-02
     23121 Chavez               Finance                     110 25-MAY-03
     44553 John                 Physics                      56 30-APR-04
     45678 Mozart               Physics                      56 27-AUG-99
     54321 Williams             Comp.Sci                     54 01-OCT-06
     55739 Sanchez              Music                        48 07-FEB-95
     76543 Brown                Comp.Sci                   58 09-MAR-09
     76653 Aoi                  Elec.Eng.                  60 10-MAY-10
     98765 Bourikas             Elec.Eng.                  98 07-JUN-11

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
     98988 Tanaka               Biology                   120 08-APR-12
     56312 ABC                  History                    20
```

6.QUESTION

Rollback to each savepoint

QUERY

```
SQL> rollback to update2;
SQL> rollback to update1;
```

OUTPUT

```
SQL> select * from student;
        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
       128 Zhang                Comp.Sci                  102 24-JUL-03
     12345 Shankar              Comp.Sci                   42 01-JAN-01
     19991 Brandt               History                    80 10-JUN-02
     23121 Chavez               Finance                   110 25-MAY-03
     44553 John                 Physics                    56 30-APR-04
     45678 Mozart               Physics                    56 27-AUG-99
     54321 Williams             Comp.Sci                   54 01-OCT-06
     55739 Sanchez              Music                      48 07-FEB-95
     76543 Brown                Comp.Sci                   58 09-MAR-09
     76653 Aoi                  Elec.Eng.                  60 10-MAY-10
     98765 Bourikas             Elec.Eng.                  98 07-JUN-11

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
     98988 Tanaka               Biology                   120 08-APR-12
     56312 A                    Finance                    20
SQL> select * from student;
        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
       128 Zhang                Comp.Sci                  102 24-JUL-03
     12345 Shankar              Comp.Sci                   42 01-JAN-01
     19991 Brandt               History                    80 10-JUN-02
     23121 Chavez               Finance                   110 25-MAY-03
     44553 John                 Physics                    56 30-APR-04
     45678 Mozart               Physics                    56 27-AUG-99
     54321 Williams             Comp.Sci                   54 01-OCT-06
     55739 Sanchez              Music                      48 07-FEB-95
     76543 Brown                Comp.Sci                   58 09-MAR-09
     76653 Aoi                  Elec.Eng.                  60 10-MAY-10
     98765 Bourikas             Elec.Eng.                  98 07-JUN-11

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
     98988 Tanaka               Biology                   120 08-APR-12
     12354 A                    Finance                    20
```

//BHAGYA A JAI
//EXPERIMENT NO:6
//DCL COMMANDS

1.QUESTION

Give select permission to all users for a table that you created already

QUERY

```
SQL> Grant select on student to public;
```

OUTPUT

```
Grant succeeded
```

2.QUESTION

Give all permission to all users for a table that you created already

QUERY

```
SQL> Grant all privileges on student to public;
```

OUTPUT

```
Grant succeeded.
```

3.QUESTION

Retrieve all the permissions that you are given

QUERY

```
SQL> revoke select on student from public;
SQL> revoke all privileges on student from public;
```

OUTPUT

```
Revoke succeeded.
```

//BHAGYA A JAI
//ROLL NO:B21CSB18
//VIEWS AND ASSERTIONS

1.QUESTION
Create a view named ‘yearwise’ from table section which shows the details for the courses on 2009
QUERY

```
SQL> create view yearwise AS
  2  select * from section
  3  where year=2009;
```

OUTPUT

```
SQL> select * from yearwise;

COURSE_ID                SEC_ID SEMESTER                 YEAR
-------------------- ---------- -------------------- ----------
BUILDING             ROOM_NUMBER
-------------------- -----------
BIO_101                       1 Summer                     2009
Painter                      514

CS_101                        1 Fall                       2009
Packard                      101

CS_190                        1 Spring                     2009
Taylor                      3128


COURSE_ID                SEC_ID SEMESTER                 YEAR
-------------------- ---------- -------------------- ----------
BUILDING             ROOM_NUMBER
-------------------- -----------
CS_190                        2 Spring                     2009
Taylor                      3128

CS_347                        1 Fall                       2009
Taylor                      3128

EE_181                        1 Spring                     2009
Taylor                      3128


COURSE_ID                SEC_ID SEMESTER                 YEAR
-------------------- ---------- -------------------- ----------
BUILDING             ROOM_NUMBER
-------------------- -----------
PHY_101                       1 Fall                       2009
Watson                       100
```

2.QUESTION
Create a view named ‘studentlist’ from table student which shows the details of students in alphabetical order
QUERY

```
SQL> create view studentlist as
  2  select * from student
  3  order by name;
```

OUTPUT

```
SQL> select * from studentlist;

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
     12354 A                    Finance                      20
     76653 Aoi                  Elec.Eng.                    60 10-MAY-10
     98765 Bourikas             Elec.Eng.                    98 07-JUN-11
     19991 Brandt               History                      80 10-JUN-02
     76543 Brown                Comp.Sci                     58 09-MAR-09
     23121 Chavez               Finance                     110 25-MAY-03
     44553 John                 Physics                      56 30-APR-04
     45678 Mozart               Physics                      56 27-AUG-99
     55739 Sanchez              Music                        48 07-FEB-95
     12345 Shankar              Comp.Sci                     42 01-JAN-01
     98988 Tanaka               Biology                     120 08-APR-12

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
     54321 Williams             Comp.Sci                     54 01-OCT-06
       128 Zhang                Comp.Sci                    102 24-JUL-03
```

3.QUESTION

Create a view named ‘teacher’ which shows the details of teacher who are not taking any class

QUERY

```
SQL> create view teacher as
  2  select * from instructor
  3  where ID not in(select distinct ID from teaches);
```

OUTPUT

```
SQL> select * from teacher;

        ID NAME                           DEPT_NAME                SALARY
---------- ------------------------------ -------------------- ----------
     33456 Gold                           Physics                   95700
     58583 Califieri                      History                   71610
```

4.QUESTION

Insert a new row into view studentlist and update the name of student as ‘saviour’ to the new row

QUERY

```
SQL> insert into student(ID,name,dept_name,tot_cred) values (36541,'Joy','Comp.Sci',80);
SQL> update student set name = 'Saviour' where Id=36541;
```

OUTPUT

```
SQL> select * from studentlist;

        ID NAME                 DEPT_NAME              TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
     12354 A                    Finance                      20
     76653 Aoi                  Elec.Eng.                    60 10-MAY-10
     98765 Bourikas             Elec.Eng.                    98 07-JUN-11
     19991 Brandt               History                      80 10-JUN-02
     76543 Brown                Comp.Sci                    58 09-MAR-09
     23121 Chavez               Finance                    110 25-MAY-03
     44553 John                Physics                    56 30-APR-04
     45678 Mozart               Physics                    56 27-AUG-99
     55739 Sanchez              Music                      48 07-FEB-95
     36541 Saviour              Comp.Sci                    80
     12345 Shankar              Comp.Sci                    42 01-JAN-01

        ID NAME                 DEPT_NAME             TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
     98988 Tanaka               Biology                    120 08-APR-12
     54321 Williams             Comp.Sci                    54 01-OCT-06
       128 Zhang               Comp.Sci                   102 24-JUL-03
```

5.QUESTION

Delete the new row added to the view studentlist

QUERY

```
SQL> delete from student where ID =36541;
```

OUTPUT

```
SQL> select * from studentlist;

        ID NAME                 DEPT_NAME             TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
     12354 A                   Finance                    20
     76653 Aoi                 Elec.Eng.                  60 10-MAY-10
     98765 Bourikas             Elec.Eng.                  98 07-JUN-11
     19991 Brandt              History                    80 10-JUN-02
     76543 Brown                Comp.Sci                    58 09-MAR-09
     23121 Chavez               Finance                    110 25-MAY-03
     44553 John                Physics                    56 30-APR-04
     45678 Mozart               Physics                    56 27-AUG-99
     55739 Sanchez              Music                      48 07-FEB-95
     12345 Shankar              Comp.Sci                    42 01-JAN-01
     98988 Tanaka               Biology                    120 08-APR-12

        ID NAME                 DEPT_NAME             TOT_CRED DATE_
---------- -------------------- -------------------- ---------- ---------
     54321 Williams             Comp.Sci                    54 01-OCT-06
       128 Zhang               Comp.Sci                   102 24-JUL-03
```

6.QUESTION

Drop all the views created

QUERY

```
SQL> drop view studentlist;
SQL> drop view teacher;
SQL> drop view yearwise;
```

OUTPUT

View dropped

//BHAGYA A JAI
//ROLL NO:B21CSB18
//BUILT IN FUNCTIONS

1.QUESTION

Evaluate the following using Built in functions
a. Cos(absolute(-10))* e^2) ,print the rounded value
b. Log10 (((5^4)%14)) , print the result as 2 digit in decimal part
c. Sin(30)+tan(60), also print the sign of the result

QUERY

```
SQL> select round(cos(abs(-10)*power(exp(1),2)),2) AS RESULT1 from dual;
SQL> select log((sqrt(mod(power(5,4), 14))),10) as result2 from dual;
SQL> select (sin(30) +tan(30)) as RESULT3 from dual;
```

OUTPUT

```
  RESULT1
----------
      .06

  RESULT2
----------
2.09590327

  RESULT3
----------
-7.3933628
```

2.QUESTION

Find the greatest preceding or the least succeeding integer of 12.9

QUERY

```
SQL> select ceil(12.9) as ANSWER from dual;
```

OUTPUT

```
   ANSWER
----------
       13
```

3.QUESTION

Display Name of instructor as Uppercase, lowercase letters, and also shows as first letter in capital in separate columns

QUERY

```
SQL> select upper(name) as UPPER ,lower(name) as LOWER, initcap(name) as CAPITALISED
from instructor;
```

OUTPUT

```
UPPER                          LOWER
------------------------------ ------------------------------
CAPITALISED
------------------------------
SRINIVASAN                     srinivasan
Srinivasan

WU                             wu
Wu

MOZART                         mozart
```

```
Mozart


UPPER                     LOWER
------------------------------ ------------------------------
CAPITALISED
------------------------------
EINSTEIN                    einstein
Einstein

EL SAID                    el said
El Said

GOLD                      gold
Gold


UPPER                     LOWER
------------------------------ ------------------------------
CAPITALISED
------------------------------
KATZ                      katz
Katz

CALIFIERI                   califieri
Califieri

CRICK                     crick
Crick


UPPER                     LOWER
------------------------------ ------------------------------
CAPITALISED
------------------------------
BRANDT                     brandt
Brandt

KIM                      kim
Kim
```

4.QUESTION

Replace the '-' in Course_id with '/' and display it as new column

QUERY

```
SQL> select course_id, replace(course_id,'_','\') as Modified_cid from course;
```

OUTPUT

```
COURSE_ID          MODIFIED_CID
-------------------- --------------------
BIO_101            BIO\101
BIO_301            BIO\301
CS_101            CS\101
CS_190            CS\190
CS_315            CS\315
```

```
CS_319              CS\319
CS_347              CS\347
EE_181              EE\181
FIN_201             FIN\201
HIS_351             HIS\351
MU_199              MU\199

COURSE_ID           MODIFIED_CID
-------------------- --------------------
PHY_101             PHY\101
```

5.QUESTION

Display the name of instructor with department name as full name

QUERY

```
SQL> select concat(name,dept_name) as Fullname from instructor;
```

OUTPUT

```
FULLNAME
----------------------------------------
SrinivasanComp.Sci
WuFinance
MozartMusic
EinsteinPhysics
El SaidHistory
GoldPhysics
KatzComp.Sci
CalifieriHistory
CrickBiology
BrandtComp.Sci
KimElec.Eng.
```

6.QUESTION

Display the last 3 numbers from course id

QUERY

```
SQL> select substr(course_id,-3,3) as lastnumbers from course;
```

OUTPUT

```
LAS
---
101
301
101
190
315
319
347
181
201
351
199

LAS
---
101
```

//BHAGYA A JAI
//ROLL NO:B21CSB18
//HAVING AND GROUP BY

1.QUESTION

For each department, retrieve the department name, the number of instructors in the department, and their average salary

QUERY

```
SQL> select dept_name,count(name) as numberofinstr,avg(salary) as avsalary from instructor group by dept_name;
```

OUTPUT

```
DEPT_NAME          NUMBEROFINSTR   AVSALARY
-------------------- ------------- ----------
Physics                      2     100100
Comp.Sci                       3 86258.3333
Finance                      1      99000
Elec.Eng.                     1      88000
Biology                      1      83160
Music                        1      46200
History                      2      70455
```

2.QUESTION

For each course, retrieve the course_id, the semester, and the number of students who takes that course

QUERY

```
SQL> select course_id , semester ,count(ID) as numstudents from subject group by course_id,semester;
```

OUTPUT

```
COURSE_ID          SEMESTER           NUMSTUDENTS
-------------------- -------------------- -----------
CS_347             Fall                     2
PHY_101              Fall                    1
BIO_101             Summer                    1
CS_101             Fall                     6
FIN_201             Spring                   1
CS_101             Spring                   1
BIO_301             Summer                    1
CS_190             Spring                   2
CS_315             Spring                   2
HIS_351             Spring                   1
CS_319             Spring                   2

COURSE_ID          SEMESTER           NUMSTUDENTS
-------------------- -------------------- -----------
EE_181             Spring                   1
MU_199              Spring                   1
```

3.QUESTION

For each course on which more than two students taken, retrieve the course_id, the semester, and the number of students who takes that course

QUERY

```
SQL> select course_id,semester,count(ID) as numstudents from subject group by course_id,semester having count(ID)>2;
```

OUTPUT

```
COURSE_ID          SEMESTER           NUMSTUDENTS
-------------------- -------------------- -----------
CS_101             Fall                       6
```

4.QUESTION

For each course, retrieve the course_id, the instructor name, and the number of courses taken from each department

QUERY

```
SQL> select course_id,name,count(*) as numcourses  from instructor,teaches  where instructor.ID =teaches.ID  group by course_id,name;
```

OUTPUT

```
COURSE_ID   NAME       NUMCOURSES
----------- ------------ -----------
CS_101      Srinivasan          2
CS_315      Srinivasan          1
CS_347      Srinivasan          1
FIN_201     Wu                1
MU_199      Mozart              1
PHY_101     Einstein            1
HIS_351     El Said            1
CS_101      Katz              1
CS_319      Katz              2
BIO_101     Crick              1
BIO_301     Crick              1
CS_190      Brandt             2
CS_319      Brandt             1
EE_181      Kim               1
```

5.QUESTION

For each department that has more than two instructors ,retrieve the department name and the number of its instructors who are making more than $80000

QUERY

```
SELECT d.DEPT_NAME, COUNT(*) AS NUM_INSTRUCTORS_OVER_80000
FROM instructor i
JOIN department d ON i.DEPT_NAME = d.DEPT_NAME
WHERE i.SALARY > 80000
GROUP BY d.DEPT_NAME
HAVING COUNT(*) > 2
ORDER BY NUM_INSTRUCTORS_OVER_80000 DESC, d.DEPT_NAME;
```

OUTPUT

```
DEPT_NAME    NUM_INSTRUCTORS_OVER_80000
------------ --------------------------
Comp.Sci      2
Finance       2
```

//BHAGYA A JAI
//ROLL NO:B21CSB18
//PL/SQL

1.QUESTION

Write a PL SQL program which assigns a message and display it.

PROGRAM

```
declare
msg varchar(20):= 'welcome to plsql';
begin
dbms_output.put_line(msg);
end;
/
```

OUTPUT

```
SQL> set serveroutput on
SQL> @program1.sql
welcome to plsql
PL/SQL procedure successfully completed.
```

2.QUESTION

Write a PL SQL conditional statement program for Grading system

PROGRAM

```
declare
score number := 85;
grade varchar2(2);
begin
if score>=90 then grade := 'A';
elsif score >=80 then grade:='B';
elsif score >=70 then grade:='C';
elsif score >=60 then grade:='D';
else grade:= 'F';
end if;
dbms_output.put_line('Grade: ' || grade);
end;
/
```

OUTPUT

```
SQL> set serveroutput on
SQL> @program2.sql
Grade: B
PL/SQL procedure successfully completed.
```

3.QUESTION

Using Varray write a PL SQL program to find total marks of each of 5 students in 2 subject

PROGRAM

```
DECLARE
   type namesarray IS VARRAY(5) OF VARCHAR2(10);
   type gradeA IS VARRAY(5) OF INTEGER;
   type gradeB IS VARRAY(5) OF INTEGER;
   names namesarray;
   marksA gradeA;
   marksB gradeB;
   total integer;
   totalmarks integer;
BEGIN
   names := namesarray('A', 'B', 'C', 'D', 'E');
   marksA:= gradeA(98, 97, 78, 87, 92);
   marksB:= gradeB(89, 97, 88, 81, 99);
   total := names.count;
   dbms_output.put_line('Total '|| total || ' Students');
   FOR i in 1 .. total LOOP
      totalmarks :=0;
      totalmarks := marksA(i)+marksB(i);
      dbms_output.put_line('Student: ' || names(i) || '
      Marks: ' ||totalmarks );
   END LOOP;
END;
/
```

OUTPUT

```
SQL> set serveroutput on
SQL> @program3.sql
Total 5 Students
Student: A
      Marks: 187
Student: B
      Marks: 194
Student: C
      Marks: 166
Student: D
      Marks: 168
Student: E
      Marks: 191
PL/SQL procedure successfully completed.
```

4.QUESTION

Using While loop write a PL SQL program to find factorial of any number

PROGRAM

```
declare
num number:=5;
factorial number := 1;
begin
while num>0 loop
factorial := factorial*num;
num :=num-1;
end loop;
dbms_output.put_line('Factorial: '|| factorial);
end;
/
```

OUTPUT

```
SQL> set serveroutput on
SQL> @program4.sql
Factorial: 120
PL/SQL procedure successfully completed.
```

5.QUESTION

Using simple loop write a PL SQL program to display multiplication table of 3 upto 10

PROGRAM

```
declare
multiplier number := 3;
begin
for i in 1..10 loop
dbms_output.put_line(multiplier ||'*'||i||'='||multiplier*i);
end loop;
end;
/
```

OUTPUT

```
SQL> set serveroutput on
SQL> @program5.sql
3*1=3
3*2=6
3*3=9
3*4=12
3*5=15
3*6=18
3*7=21
3*8=24
3*9=27
3*10=30
PL/SQL procedure successfully completed.
```

6.QUESTION

Write a PLSQL program to find the odd and even numbers between 1 to 10 and insert it into a table and display it.

PROGRAM

```
CREATE TABLE even_numbers (
   number_value NUMBER
);

CREATE TABLE odd_numbers (
   number_value NUMBER
);

DECLARE
   i NUMBER;
BEGIN
   FOR i IN 1..10 LOOP
      IF MOD(i, 2) = 0 THEN

         INSERT INTO even_numbers (number_value) VALUES (i);
      ELSE

         INSERT INTO odd_numbers (number_value) VALUES (i);
      END IF;
   END LOOP;
   COMMIT;
END;
```

```
/

SELECT * FROM even_numbers;
SELECT * FROM odd_numbers;
```

OUTPUT

```
SQL> @program6
Table created.
Table created.
PL/SQL procedure successfully completed.
NUMBER_VALUE
------------
         2
         4
         6
         8
        10


NUMBER_VALUE
------------
         1
         3
         5
         7
         9
```

7. QUESTION

Write a PL/SQL block to find the square and cube of first 10 numbers and insert these values into a table

PROGRAM

```
CREATE TABLE square_cube_results (
   number_value NUMBER,
   square_value NUMBER,
   cube_value NUMBER
);
DECLARE
   i NUMBER;
   square_val NUMBER;
   cube_val NUMBER;
BEGIN
   FOR i IN 1..10 LOOP
      square_val := i * i;
      cube_val := i * i * i;

      INSERT INTO square_cube_results (number_value, square_value, cube_value)
      VALUES (i, square_val, cube_val);
   END LOOP;
   COMMIT; -- Commit the changes
END;
/
SELECT * FROM square_cube_results;
```

OUTPUT

```
SQL> @program7
```

```
Table created.
PL/SQL procedure successfully completed.
NUMBER_VALUE SQUARE_VALUE CUBE_VALUE
------------ ------------ ----------
           1            1          1
           2            4          8
           3            9         27
           4           16         64
           5           25        125
           6           36        216
           7           49        343
           8           64        512
           9           81        729
          10          100       1000

10 rows selected.
```

8.QUESTION

Write a PL/SQL block to accept cid and update Emi to half of its original value and display appropriate message based on the existence of the record in customer table

PROGRAM

```
CREATE TABLE customers (
    cid NUMBER PRIMARY KEY,
    emi NUMBER
);
INSERT INTO customers (cid, emi) VALUES (1, 1000);
INSERT INTO customers (cid, emi) VALUES (2, 2000);
INSERT INTO customers (cid, emi) VALUES (3, 3000);
INSERT INTO customers (cid, emi) VALUES (4, 4000);
INSERT INTO customers (cid, emi) VALUES (5, 5000);
select * from customers;
GRANT SELECT, UPDATE ON customers TO public;
DECLARE
    v_cid NUMBER;
    v_emi NUMBER;
BEGIN
    v_cid := &cid;
    SELECT emi INTO v_emi FROM customers WHERE cid = v_cid;
    IF SQL%FOUND THEN
        UPDATE customers SET emi = emi / 2 WHERE cid = v_cid;
        COMMIT;
        DBMS_OUTPUT.PUT_LINE('EMI updated successfully. New EMI is half of the original
EMI.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Customer with CID ' || v_cid || ' not found in the customer
table.');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
END;
/
```

```
Select * from customers;
```

OUTPUT

```
SQL> @program8
Table created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
       CID        EMI
---------- ----------
         1       1000
         2       2000
         3       3000
         4       4000
         5       5000
Grant succeeded.
Enter value for cid: 4
old   5:     v_cid := &cid;
new   5:     v_cid := 4;
PL/SQL procedure successfully completed.
       CID        EMI
---------- ----------
         1       1000
         2       2000
         3       3000
         4       2000
         5       5000
```

//BHAGYA A JAI
//ROLL NO:B21CSB18
//PROCEDURES

1.QUESTION

Create a procedure to find the minimum of two numbers

PROGRAM

```
set serveroutput on;
CREATE OR REPLACE PROCEDURE min_num (
  a in int, b in int
)
IS
BEGIN
   IF a>b then

      dbms_output.put_line('minimum of ' || a || ' and ' || b || ' is : ' || b);
   ELSE

      dbms_output.put_line('minimum of ' || a || ' and ' || b || ' is : ' || a);
   END IF;
END min_num;
/
```

OUTPUT

```
SQL> @procdure1
Procedure created.
SQL> exec min_num(4,7);
minimum of 4 and 7 is : 4
PL/SQL procedure successfully completed.
```

2.QUESTION

Create a procedure to convert a user input binary number to its decimal number

PROGRAM

```
CREATE OR REPLACE PROCEDURE binary_to_decimal (
   binary_input IN VARCHAR2
)
IS
   binary_length NUMBER;
   decimal_value NUMBER := 0;
BEGIN
   binary_length := LENGTH(binary_input);

   FOR i IN REVERSE 1..binary_length LOOP
      IF SUBSTR(binary_input, i, 1) = '1' THEN
         decimal_value := decimal_value + POWER(2, binary_length - i);
      END IF;
   END LOOP;

   dbms_output.put_line('Decimal equivalent of ' || binary_input || ' is: ' || decimal_value);
END binary_to_decimal;
/
```

OUTPUT

```
SQL> @procdure2
```

```
Procedure created.
SQL> exec binary_to_decimal(11);
Decimal equivalent of 11 is: 3
PL/SQL procedure successfully completed.
```

3.QUESTION

Create a procedure to find roots of a quadratic equation

PROGRAM

```
CREATE OR REPLACE PROCEDURE quadratic_roots (
   a IN NUMBER,
   b IN NUMBER,
   c IN NUMBER
)
IS
   discriminant NUMBER;
   root1 NUMBER;
   root2 NUMBER;
BEGIN
   discriminant := b * b - 4 * a * c;

   IF discriminant > 0 THEN
      -- Two distinct real roots
      root1 := (-b + SQRT(discriminant)) / (2 * a);
      root2 := (-b - SQRT(discriminant)) / (2 * a);
      dbms_output.put_line('Two distinct real roots:');
      dbms_output.put_line('Root 1: ' || root1);
      dbms_output.put_line('Root 2: ' || root2);
   ELSIF discriminant = 0 THEN
      -- One real root (double root)
      root1 := -b / (2 * a);
      dbms_output.put_line('One real root (double root):');
      dbms_output.put_line('Root 1: ' || root1);
   ELSE
      -- Complex roots (no real roots)
      dbms_output.put_line('Complex roots (no real roots)');
   END IF;
END quadratic_roots;
/
```

OUTPUT

```
SQL> @procdure3
Procedure created.
SQL> exec quadratic_roots(1,-9,20);
Two distinct real roots:
Root 1: 5
Root 2: 4
PL/SQL procedure successfully completed.
```

4.QUESTION

Using procedure increment the salary with 20 percent for the staff whose id is given by user.

PROGRAM

```
CREATE OR REPLACE PROCEDURE increase_salary(p_instructor_id IN NUMBER) IS
   v_old_salary NUMBER;
   v_new_salary NUMBER;
BEGIN
   SELECT salary INTO v_old_salary FROM instructor WHERE ID = p_instructor_id;
   IF v_old_salary IS NOT NULL THEN
      -- Calculate the new salary (20% increment)
      v_new_salary := v_old_salary * 1.2;

      UPDATE instructor SET salary = v_new_salary WHERE ID = p_instructor_id;

      DBMS_OUTPUT.PUT_LINE('Salary for Instructor ID ' || p_instructor_id || ' increased to ' ||
v_new_salary);
   ELSE
      DBMS_OUTPUT.PUT_LINE('Instructor with ID ' || p_instructor_id || ' not found');
   END IF;
END;
/
DECLARE
   v_instructor_id NUMBER;
BEGIN
   v_instructor_id := &Enter_Instructor_ID;
   increase_salary(v_instructor_id);
END;
/
```

OUTPUT

```
SQL> @procdure4
Procedure created.
Enter value for enter_instructor_id: 98345
old   5:     v_instructor_id := &Enter_Instructor_ID;
new   5:     v_instructor_id := 98345;
Salary for Instructor ID 98345 increased to 105600
PL/SQL procedure successfully completed.
SQL> select * from instructor;

      ID NAME                     DEPT_NAME            SALARY
---------- ------------------------------ -------------------- ----------
     10101 Srinivasan               Comp.Sci              75075
     12121 Wu                      Finance            99000
     15151 Mozart                   Music              46200
     22222 Einstein                 Physics            104500
     32343 El Said                  History            69300
     33456 Gold                    Physics            95700
     45565 Katz                    Comp.Sci             82500
     58583 Califieri                History            71610
     76766 Crick                   Biology            83160
     83821 Brandt                   Comp.Sci             101200
     98345 Kim                     Elec.Eng.          105600

11 rows selected.
```

5.QUESTION

Using procedure delete the entry from staff table whose id is given by user

PROGRAM

```
CREATE OR REPLACE PROCEDURE delete_entry(p_staff_id IN NUMBER) IS
BEGIN
   DELETE FROM staff WHERE ID = p_staff_id;
   DBMS_OUTPUT.PUT_LINE('Staff with ID ' || p_staff_id || ' deleted.');
EXCEPTION
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Error: ' || SQLCODE || ' - ' || SQLERRM);
END;
/

DECLARE
   v_staff_id NUMBER;
BEGIN
   v_staff_id := &Enter_Staff_ID;
   delete_entry(v_staff_id);
END;
/
```

OUTPUT

```
SQL> @procdure5
Procedure created.
Enter value for enter_staff_id: 2
old   4:     v_staff_id := &Enter_Staff_ID;
new   4:     v_staff_id := 2;
Staff with ID 2 deleted.
PL/SQL procedure successfully completed.
SQL> select * from staff;
        ID NAME                     SALARY
---------- -------------------- ----------
         1 abc                       20000
         3 ghi                       40000
```

//BHAGYA A JAI
//ROLL NO:B21CSB18
//FUNCTIONS

1.QUESTION

Create a function to find the factorial of n number

PROGRAM

```
SQL> create or replace function factorial (n in number) return number is  2 result number:=1;
 3 begin
 4 if n<0 then
 5 dbms_output.put_line('input must be a positive number');  6 elsif n=0 or n=1 then
 7 return 1;
 8 else
 9 for i in 2..n loop
 10 result:=result*i;
 11 end loop;
 12 return result;
 13 end if;
 14 end factorial;
 15 /
Function created.
SQL> declare
 2 result number;
 3 x number;
 4 begin
 5 x:=&x;
 6 result:=factorial(x);
 7 dbms_output.put_line('Factorial of ' ||x||' is '||result);
 8 end;
 9 /
```

OUTPUT

```
Enter value for x: 5
old 5: x:=&x;
new 5: x:=5;
Factorial of 5 is 120
PL/SQL procedure successfully completed.
```

2.QUESTION

Create a function to find the reverse of a given number and also check whether the number is palindrome or not

PROGRAM

```
SQL> CREATE OR REPLACE FUNCTION reverse_number(n IN NUMBER) RETURN
NUMBER IS  2 x NUMBER := n;
 3 reversed_number NUMBER := 0;
 4 BEGIN
 5 WHILE x > 0 LOOP
 6 reversed_number := reversed_number * 10 + MOD(x, 10);  7 x := TRUNC(x / 10);
 8 END LOOP;
 9
 10
 11 if reversed_number=n then
 12 dbms_output.put_line('PALINDROME');
 13 else
 14 dbms_output.put_line('NOT A PALINDROME');
 15 end if;
 16 RETURN reversed_number;
 17 END reverse_number;
 18 /
Function created.
SQL> DECLARE
 2 result NUMBER;
 3 y number;
 4 BEGIN
 5 y:=&y;
 6 result := reverse_number(y);
 7 DBMS_OUTPUT.PUT_LINE('Reversed number is: ' || result);  8 END;
 9 /
```

OUTPUT

```
Enter value for y: 12321
old 5: y:=&y;
new 5: y:=12321;
PALINDROME
Reversed number is: 12321
PL/SQL procedure successfully completed.
SQL> DECLARE
 2 result NUMBER;
 3 y number;
 4 BEGIN
 5 y:=&y;
 6 result := reverse_number(y);
 7 DBMS_OUTPUT.PUT_LINE('Reversed number is: ' || result);  8 END;
 9 /
Enter value for y: 123
old 5: y:=&y;
new 5: y:=123;
NOT A PALINDROME
Reversed number is: 321
PL/SQL procedure successfully completed.
```

3.QUESTION

Create a function to check the given number is armstrong or not

PROGRAM

```
SQL> create or replace function armstrong(n in number,p in number) return number is  2 x
number :=n;
 3 arm number:=0;
 4 digit number:=0;
 5 begin
 6 while x>0 loop
 7 digit:=mod(x,10);
 8 arm:=arm + power(digit,p);
 9 x:=trunc(x/10);
 10 end loop;
 11 if arm=n then
 12 dbms_output.put_line('it is an armstrong number');  13 else
 14 dbms_output.put_line('it is not an armstrong number');  15 end if;
 16 return arm;
 17 end armstrong;
 18 /
Function created.
SQL> DECLARE
 2 result NUMBER;
 3 y number;
 4 length number;
 5 BEGIN
 6 y:=&y;
 7 length:=&length;
 8 result := armstrong(y,length);
 9 DBMS_OUTPUT.PUT_LINE('resulting number is: ' || result);
10 END;
 11 /
```

OUTPUT

```
Enter value for y: 8208
old 6: y:=&y;
new 6: y:=8208;
Enter value for length: 4
old 7: length:=&length;
new 7: length:=4;
it is an armstrong number
resulting number is: 8208
PL/SQL procedure successfully completed.
```

4.QUESTION

Create a function to find sum of n natural numbers and also insert it into a table having attributes number and sum

PROGRAM

```
SQL> CREATE OR REPLACE FUNCTION calculate_and_insert_sum(n IN NUMBER) RETURN
NUMBER  IS
 2 total_sum NUMBER := 0;
 3 BEGIN
 4 FOR i IN 1..n LOOP
 5 total_sum := total_sum + i;
 6 END LOOP;
 7 INSERT INTO num_sum VALUES (n, total_sum);
 8 RETURN total_sum;
 9 END calculate_and_insert_sum;
 10 /
Function created.
SQL> declare
 2 result number;
 3 x number;
 4 i number;
 5 begin
 6 x:=&x;
 7 result:=calculate_and_insert_sum(x);
 8 for rec in (select sum from num_sum) loop
 9 dbms_output.put_line('sum of '||x||' natural numbers are :  '||rec.sum);
 10 end loop;
 11 end;
 12 /
```

OUTPUT

```
Enter value for x: 4
old 6: x:=&x;
new 6: x:=4;
sum of 4 natural numbers are : 10
PL/SQL procedure successfully completed.
```

5.QUESTION

Create a function to find square root and cube root of a given number and then insert it into table having attributes number, square root and cube root

PROGRAM

```
SQL> CREATE OR REPLACE FUNCTION calcroots(n IN NUMBER) RETURN NUMBER IS  2
sqroot NUMBER := 0;
 3 cuberoot NUMBER := 0;
 4 BEGIN
 5 sqroot := SQRT(n);
 6 cuberoot := POWER(n, 1/3);
 7
 8 INSERT INTO num_roots VALUES (n, sqroot, cuberoot);  9
 10 RETURN sqroot+cuberoot;
 11 END calcroots;
 12 /
SQL> DECLARE
 2 result NUMBER;
 3 x NUMBER;
 4 BEGIN
 5 x := &x;
 6 result := calcroots(x);
 7
 8 FOR rec IN (SELECT num, ROUND(sq, 2) as sq, ROUND(cube, 2) as cube FROM  num_roots
WHERE num = x) LOOP
 9 DBMS_OUTPUT.PUT_LINE('Square root and cube root of ' || x || ' is: ' ||
rec.sq || ', ' || rec.cube);
 10 END LOOP;
 11
 12 END;
 13 /
```

OUTPUT

```
Enter value for x: 27
old 5: x := &x;
new 5: x := 27;
Square root and cube root of 27 is: 5.2, 3 PL/SQL procedure successfully completed.
SQL> select * from num_roots;
 NUM SQ CUBE ---------- ---------- ----------  27 5.19615242
3
```

//BHAGYA A JAI
//ROLL NO:B21CSB18
//TRIGGERS

1.QUESTION

Write a trigger on the Course table which shows the old values and new values of credits after any updations on credits on Course table.

PROGRAM

```
CREATE OR REPLACE TRIGGER course_credit_update_trigger
BEFORE UPDATE ON Course
FOR EACH ROW
DECLARE
   v_old_credits NUMBER;
   v_new_credits NUMBER;
BEGIN
   v_old_credits := :OLD.credits;
   v_new_credits := :NEW.credits;

   DBMS_OUTPUT.PUT_LINE('Course ID: ' || :OLD.course_id);
   DBMS_OUTPUT.PUT_LINE('Old Credits: ' ||  v_old_credits );
   DBMS_OUTPUT.PUT_LINE('New Credits: ' ||v_new_credits);
END;
/
```

OUTPUT

```
SQL> UPDATE Course SET credits=3 WHERE course_id='CS-190';
Course ID: CS-190
Old Credits: 4
New Credits: 3

1 row updated.
```

2.QUESTION

Write a trigger to insert the old and new values of salary in the Instructor table into a new table when the Instructor table is updated and salary difference should be displayed.

PROGRAM

```
CREATE TABLE new_tab (oldsal number,newsal number);

CREATE OR REPLACE TRIGGER Trig2
AFTER UPDATE ON Instructor
FOR EACH ROW
DECLARE
   v_old_salary NUMBER;
   v_new_salary NUMBER;
   diff NUMBER;
BEGIN
   v_old_salary := :OLD.salary;
   v_new_salary := :NEW.salary;
   diff:=v_new_salary-v_old_salary;

   Insert into new_tab values(v_old_salary,v_new_salary);

   DBMS_OUTPUT.PUT_LINE('Instructor ID: ' || :OLD.ID);
   DBMS_OUTPUT.PUT_LINE('Old Salary: ' || v_old_salary );
   DBMS_OUTPUT.PUT_LINE('New Salary: ' ||v_new_salary);
   DBMS_OUTPUT.PUT_LINE('Difference: ' ||diff);
END;
/
```

OUTPUT

```
SQL> update Instructor set salary=76000 where ID=10101;
Instructor ID: 10101
Old Salary: 75075
New Salary: 76000
Difference: 925

1 row updated.
```

3.QUESTION

Write a trigger to ensure that no course of credits less than 3 can be inserted in the Course table.

PROGRAM

```
CREATE OR REPLACE TRIGGER Trig3
BEFORE INSERT ON Course
FOR EACH ROW
DECLARE
   v_new_credits NUMBER;
   new_id varchar(10);
BEGIN
   v_new_credits := :NEW.credits;
   new_id := :NEW.course_id;

   if v_new_credits<3 then
         DBMS_OUTPUT.PUT_LINE('Cannot insert course with less than 3 credits.');
         --DELETE FROM Course WHERE course_id=new_id;
         RAISE_APPLICATION_ERROR(-20001, 'Credits must be 3 or greater.');
   end if;
END;
/
```

OUTPUT

```
SQL>  INSERT INTO Course values('CS-102','Intro to DBMS','Comp.Sci',2);
Cannot insert course with less than 3 credits.
 INSERT INTO Course values('CS-102','Intro to DBMS','Comp.Sci',2)
          *
ERROR at line 1:
ORA-20001: Credits must be 3 or greater.
ORA-06512: at "R5B49.TRIG3", line 11
ORA-04088: error during execution of trigger 'R5B49.TRIG3'
```

4.QUESTION

Write a trigger to ensure that no row with year 2009 and semester Fall can be deleted from the Takes table.

PROGRAM

```
CREATE OR REPLACE TRIGGER Trig4
BEFORE DELETE ON Subject
FOR EACH ROW
DECLARE
```

```
    v_year NUMBER;
    v_sem varchar(10);
BEGIN
    v_year := :OLD.year;
    v_sem := :OLD.semester;

    if v_year=2009 and v_sem='Fall' then
            RAISE_APPLICATION_ERROR(-20002, 'Deletion of rows with year 2009 and semester fall is not allowed.');
    end if;
END;
/
```

OUTPUT

```
SQL> DELETE FROM Subject WHERE year=2009;
DELETE FROM Subject WHERE year=2009
            *
ERROR at line 1:
ORA-20002: Deletion of rows with year 2009 and semester fall is not allowed.
ORA-06512: at "R5B49.TRIG4", line 9
ORA-04088: error during execution of trigger 'R5B49.TRIG4'
```

5.QUESTION

Delete all the triggers created.

COMMANDS

```
SQL> DROP TRIGGER Trig2;
SQL> DROP TRIGGER Trig3;
SQL> DROP TRIGGER Trig4;
SQL> DROP TRIGGER course_credit_update_trigger;
```

OUTPUT

```
Trigger dropped.
Trigger dropped.
Trigger dropped.
Trigger dropped.
```

//BHAGYA A JAI
//ROLL NO:B21CSB18
//CURSORS

1.QUESTION
Write a PL / SQL program to create a cursor that displays the name, department and salary of each instructor in the INSTRUCTOR table whose salary is less than that specified by a passed-in parameter value.
PROGRAM

```
DECLARE
   iname Instructor%RowType;
   CURSOR cur1(ssal number) IS
         SELECT * FROM Instructor WHERE salary<ssal;
BEGIN
   OPEN cur1(70000);
   LOOP
   FETCH cur1 INTO iname;
   EXIT WHEN cur1%NOTFOUND;
   dbms_output.put_line('ID :'|| iname.ID||' Salary :'||iname.salary);
end loop;
END;
/
```

OUTPUT

```
ID :15151 Salary :46200
ID :32343 Salary :69300

PL/SQL procedure successfully completed.
```

2.QUESTION
Write a PL/SQL program to create a cursor that displays the department name, number of instructors and number of courses listed in each department.
PROGRAM

```
DECLARE
   dname Department.dept_name%Type;
   numin NUMBER;
   numc NUMBER;
   CURSOR cur1 IS
         SELECT I.dept_name,count( distinct I.ID),count(distinct C.course_id)
         from Department D ,Instructor I,Course C
         where C.dept_name=I.dept_name
         Group By I.dept_name;
BEGIN
   OPEN cur1;
   LOOP
   FETCH cur1 INTO dname,numin,numc;
   EXIT WHEN cur1%NOTFOUND;
   dbms_output.put_line('Department name :'|| dname||' Count Instructor:'||numin||' Count Course:'||
numc);
end loop;
END;
/
```

OUTPUT

```
Department name :Biology Count Instructor:1 Count Course:2
Department name :Comp.Sci Count Instructor:3 Count Course:5
Department name :Elec.Eng Count Instructor:1 Count Course:1
Department name :Finance Count Instructor:1 Count Course:1
Department name :History Count Instructor:2 Count Course:1
Department name :Music Count Instructor:1 Count Course:1
Department name :Physics Count Instructor:2 Count Course:1

PL/SQL procedure successfully completed.
```

3.QUESTION

Write a PL/SQL program to display instructor id, name and salary of 5 highest paid instructors using cursor.

PROGRAM

```
DECLARE
   iname Instructor%RowType;
   CURSOR cur1 IS
         SELECT *
          FROM (
         SELECT *
         FROM Instructor
         ORDER BY salary DESC
      )
      WHERE ROWNUM <= 5;
BEGIN
   OPEN cur1;
   LOOP
   FETCH cur1 INTO iname;
   EXIT WHEN cur1%NOTFOUND;

   dbms_output.put_line('Name :'|| iname.name||' ID:'||iname.ID||' Salary:'||iname.salary);

end loop;
END;
/
```

OUTPUT

```
Name :Einstein ID:22222 Salary:104500
Name :Brandt ID:83821 Salary:101200
Name :Wu ID:12121 Salary:99000
Name :Gold ID:33456 Salary:95700
Name :Kim ID:98345 Salary:88000

PL/SQL procedure successfully completed.
```

4.QUESTION

Write a PL/SQL program to increase salary of instructors in department specified by a parameter using cursor. The salary increase is 20% for instructors making less than 80,000 and 12% for the employees making 80,000 or more.

PROGRAM

```
DECLARE
```

```
   v_instructor_id   Instructor.ID%TYPE;
   v_name            Instructor.name%TYPE;
   v_salary          Instructor.salary%TYPE;


   CURSOR instructor_cursor(p_dept varchar) IS
      SELECT ID, name, salary
      FROM Instructor
      WHERE dept_name = p_dept;

BEGIN
   OPEN instructor_cursor('Comp.Sci');

   LOOP
      FETCH instructor_cursor INTO v_instructor_id, v_name, v_salary;
      EXIT WHEN instructor_cursor%NOTFOUND;

      IF v_salary < 80000 THEN
         v_salary := v_salary * 1.20;
      ELSE
         v_salary := v_salary * 1.12;
      END IF;

      UPDATE Instructor
      SET salary = v_salary
      WHERE ID = v_instructor_id;

      DBMS_OUTPUT.PUT_LINE('Instructor ID: ' || v_instructor_id);
      DBMS_OUTPUT.PUT_LINE('Name: ' || v_name);
      DBMS_OUTPUT.PUT_LINE('Old Salary: ' || v_salary / (CASE WHEN v_salary < 80000
THEN 1.20 ELSE 1.12 END));
      DBMS_OUTPUT.PUT_LINE('New Salary: ' || v_salary);
      DBMS_OUTPUT.PUT_LINE('--------------------------');
   END LOOP;

   CLOSE instructor_cursor;
END;
/
```

OUTPUT

```
Instructor ID: 10101
Name: Srinivasan
Old Salary: 80437.5
New Salary: 90090
--------------------------
Instructor ID: 45565
Name: Katz
Old Salary: 82500
New Salary: 92400
--------------------------
Instructor ID: 83821
Name: Brandt
```

```
Old Salary: 101200
New Salary: 113344
-------------------------

PL/SQL procedure successfully completed.
```

5.QUESTION

Write a PL / SQL program to create a cursor that displays all rows in Takes relation with grade * A' and display the total number of rows.

PROGRAM

```
Declare
   v_student_id   Subject.ID%TYPE;
   v_course_id    Subject.course_id%TYPE;
   v_grade        Subject.grade%TYPE;


   CURSOR grade_a_cursor IS
      SELECT ID, course_id, grade
      FROM Subject
      WHERE grade = 'A';

   v_total_rows NUMBER := 0;
BEGIN
   OPEN grade_a_cursor;

   LOOP
      FETCH grade_a_cursor INTO v_student_id, v_course_id, v_grade;
      EXIT WHEN grade_a_cursor%NOTFOUND;

      DBMS_OUTPUT.PUT_LINE('Student ID: ' || v_student_id);
      DBMS_OUTPUT.PUT_LINE('Course ID: ' || v_course_id);
      DBMS_OUTPUT.PUT_LINE('Grade: ' || v_grade);
      DBMS_OUTPUT.PUT_LINE('--------------------------');
      v_total_rows := v_total_rows + 1;
   END LOOP;

   CLOSE grade_a_cursor;
   DBMS_OUTPUT.PUT_LINE('Total Rows with Grade A: ' || v_total_rows);
END;
/
```

OUTPUT

```
Student ID: 128
Course ID: CS-101
Grade: A
-------------------------
Student ID: 12345
Course ID: CS-190
Grade: A
-------------------------
Student ID: 12345
Course ID: CS-315
Grade: A
```

```
--------------------------
Student ID: 12345
Course ID: CS-347
Grade: A
--------------------------
Student ID: 76543
Course ID: CS-101
Grade: A
--------------------------
Student ID: 76543
Course ID: CS-319
Grade: A
--------------------------
Student ID: 98988
Course ID: BIO-101
Grade: A
--------------------------
Total Rows with Grade A: 7

PL/SQL procedure successfully completed.
```