SQL GRADED ASSIGNMENT

NAME:DHANALAKSHMI B

EMAIL ID:10DROSSDL@GMAIL.COM

Q1.

Create a table “student” with the structure/dictionary given above and insert 10 records given in the table created.

Create a table “faculty” with the structure/dictionary given above and insert 8 records given in the table created.

Create a table “course” with the structure/dictionary given above and insert 8 records given in the table created.

Create a table “registration” with the structure/dictionary given above and insert 18 records given in the table created.

QUERY:

```
use DSML; -- Making sure I am using the right database created for the assignment.
```

CREATING TABLE

```
CREATE TABLE student (sid VARCHAR2(3) NOT NULL,sname VARCHAR2(10) NOT NULL,sex VARCHAR2(3),major VARCHAR2(3),gpa NUMBER(3,2),PRIMARY KEY(sid));

CREATE TABLE faculty (fid VARCHAR2(3) NOT NULL ,fname VARCHAR2(10) NOT NULL,ext VARCHAR2(3),dept VARCHAR2(3),rank1 VARCHAR2(4),salary INT check (salary >0),PRIMARY KEY(fid));

CREATE TABLE course (CRSNBR VARCHAR2(6) NOT NULL,cname VARCHAR2(25) NOT NULL,credit NUMBER(1),maxenrl INT, fid VARCHAR2(3) NOT NULL,FOREIGN KEY (fid) REFERENCES faculty(fid));

CREATE TABLE registration (CRSNBR VARCHAR2(6)NOT NULL,sid VARCHAR2(3) NOT NULL,grade VARCHAR2(1),PRIMARY KEY(CRSNBR,SID), FOREIGN KEY (Sid) REFERENCES STUDENT(Sid));
```

INSERTING DATA

I INSERTED DATA USING IMPORT METHOD. STEPS FOR THE SAME ARE, HAVE DATA SAVED IN EXCEL SHEET PRIOR TO THIS. EXPAND THE TABLES(FILTERED) OPTION, SEARCH FOR THE REQUIRED TABLE, RIGHT CLICK ON THE OPTION->IMPORT DATA->BROWSE FOR FILE NAME->UNCLICK HEADER IF HEADER NOT GIVEN->NEXT->NEXT->FINISH.

BELOW IS ONE EXAMPLE OF THE INSERT QUERY.

```
INSERT INTO COURSE (CRSNBR, CNAME, CREDIT, MAXENRL, FID ) VALUES ('MGT630','INTRODUCTION TO MGMT',4,'30',138);
```

Or

```
INSERT INTO COURSE VALUES ('FIN60','MANAGERIAL FINANCE',4,'25',117);
```

SCREENSHOT:

DESCRIBE table_name;

STUDENT TABLE

Name	Null?	Type
SID	NOT NULL	VARCHAR2(3)
SNAME	NOT NULL	VARCHAR2(10)
SEX		VARCHAR2(3)
MAJOR		VARCHAR2(3)
GPA		NUMBER(3,2)

FACULTY TABLE

Name	Null?	Type
FID	NOT NULL	VARCHAR2(3)
FNAME	NOT NULL	VARCHAR2(10)
EXT		VARCHAR2(3)
DEPT		VARCHAR2(3)
RANK1		VARCHAR2(4)
SALARY		NUMBER(38)

COURSE TABLE

Name	Null?	Type
CRSNBR	NOT NULL	VARCHAR2(6)
CNAME	NOT NULL	VARCHAR2(25)
CREDIT		NUMBER(1)
MAXENRL		NUMBER(38)
FID	NOT NULL	VARCHAR2(3)

REGISTRATION TABLE

Name	Null?	Type
CRSNBR	NOT NULL	VARCHAR2(6)
SID	NOT NULL	VARCHAR2(3)
GRADE		VARCHAR2(1)

STUDENT DATA

	SID	SNAME	SEX	MAJOR	GPA
1	987	POIRIER	F	MGT	3.2
2	763	PARKER	F	FIN	2.7
3	218	RICHARDS	M	ACC	2.4
4	359	PELNICK	F	FIN	3.6
5	862	FAGIN	M	MGT	2.2
6	748	MEGLIN	M	MGT	2.8
7	506	LEE	M	FIN	2.7
8	581	GAMBREL	F	MKT	3.8
9	372	QUICK	F	MGT	3.5
10	126	ANDERSON	M	ACC	3.7

FACULTY DATA

	FID	FNAME	EXT	DEPT	RANK1	SALARY
1	36	BARGES	325	MGT	ASSO	35000
2	117	JARDIN	212	FIN	FULL	33000
3	98	KENEDY	176	ACC	ASSO	30000
4	75	SAMPLE	171	MKT	ASST	25000
5	138	WARD	125	MGT	INST	20000
6	219	PETERS	220	FIN	FULL	45000
7	151	DARDEN	250	ACC	ASSO	37000
8	113	PIERCE	205	MGT	INST	22000

COURSE DATA

	CRSNBR	CNAME	CREDIT	MAXENRL	FID
1	MGT630	INTRODUCTION TO MGMT	4	30	138
2	FIN601	MANAGERIAL FINANCE	4	25	117
3	MKT610	MARKETING FOR MANAGERS	3	35	75
4	MKT661	TAXATION	3	30	98
5	FIN602	INVESTMENT SKILLS	3	25	219
6	ACC601	BASIC ACCOUNTING	4	25	98
7	MGT681	INTERL. MANAGEMENT	3	20	36
8	MKT670	PRODUCT MARKETING	3	20	75

REGISTRATION DATA

	CRSNBR	SID	GRADE
1	MGT630	987	A
2	FIN602	987	B
3	MKT610	987	A
4	FIN601	763	B
5	FIN602	763	B
6	ACC610	763	B
7	ACC610	218	A
8	ACC661	218	A
9	MGT630	218	C
10	MGT630	359	F
11	MGT681	359	B
12	MKT610	359	A
13	MKT610	862	A
14	MKT670	862	A
15	ACC610	862	B
16	MGT630	748	C
17	MGT681	748	B
18	FIN601	748	A

Q2.

Retrieve the list of students in alphabetical order.

QUERY:

SELECT * FROM STUDENT ORDER BY SNAME;

SCREENSHOT:

	SID	SNAME	SEX	MAJOR	GPA
1	126	ANDERSON	M	ACC	3.7
2	862	FAGIN	M	MGT	2.2
3	581	GAMBREL	F	MKT	3.8
4	506	LEE	M	FIN	2.7
5	748	MEGLIN	M	MGT	2.8
6	763	PARKER	F	FIN	2.7
7	359	PELNICK	F	FIN	3.6
8	987	POIRIER	F	MGT	3.2
9	372	QUICK	F	MGT	3.5
10	218	RICHARDS	M	ACC	2.4

Q3.

Display a list of female students with a GPA above 3.25.

QUERY:

```
SELECT * FROM STUDENT WHERE GPA > 3.25 AND SEX ='F';
```

SCREENSHOT:

	SID	SNAME	SEX	MAJOR	GPA
1	359	PELNICK	F	FIN	3.6
2	581	GAMBREL	F	MKT	3.8
3	372	QUICK	F	MGT	3.5

Q4.

Retrieve the names, majors, and GPA of all students who have a GPA above 3.5 and who are majoring in either accounting or finance.

QUERY:

```
SELECT SNAME,MAJOR,GPA FROM STUDENT WHERE GPA > 3.5 AND MAJOR IN ('ACC','FIN');
```

SCREENSHOT:

	SNAME	MAJOR	GPA
1	PELNICK	FIN	3.6
2	ANDERSON	ACC	3.7

Q5.

Next year every faculty member will receive a 5% salary increase. List the names of each faculty member, his/her current salary, and next year’s salary.

QUERY:

```
SELECT FNAME,SALARY,(.05*SALARY+SALARY) AS NEXT_YEAR_SALARY FROM FACULTY;
```

SCREENSHOT:

	FNAME	SALARY	NEXT_YEAR_SALARY
1	BARGES	35000	36750
2	JARDIN	33000	34650
3	KENEDY	30000	31500
4	SAMPLE	25000	26250
5	WARD	20000	21000
6	PETERS	45000	47250
7	DARDEN	37000	38850
8	PIERCE	22000	23100

Q6.

Retrieve the average GPA from student where major=’MGT’.

QUERY:

```
SELECT AVG(GPA) FROM STUDENT
WHERE MAJOR ='MGT';
```

Or

```
SELECT MAJOR,AVG(GPA)
FROM STUDENT
GROUP BY MAJOR HAVING MAJOR ='MGT';
```

SCREENSHOT:

	AVG(GPA)
1	2.925

Or

	MAJOR	AVG(GPA)
1	MGT	2.925

Q7.

Create a new table rgn_copy and copy the data from the REGISTRATION table to the rgn_copy table. Change the grade to F in rgn_copy table where course no is MGT681.

QUERY:

```
CREATE TABLE RGN_COPY AS SELECT * FROM REGISTRATION;
UPDATE RGN_COPY SET GRADE ='F' WHERE CRSNBR ='MGT681';
```

SCREENSHOT:

```
Table RGN_COPY1 created.
```

9	MGT630	218	C
10	MGT630	359	F
11	MGT681	359	F
12	MKT610	359	A
13	MKT610	862	A

Q8.

Create a new table std_copy and copy the data from the student table to the std_copy table. A student whose ID number is 748 leaves the University. First delete the course in which student 748 is enrolled from the rgn_copy table. Then remove the student from the table std_copy.

QUERY:

```
CREATE TABLE STD_COPY AS SELECT * FROM STUDENT;

DELETE FROM RGN_COPY WHERE SID =748;

Or

DELETE FROM RGN_COPY

WHERE SID IN

        (SELECT S.SID

        FROM RGN_COPY R

        INNER JOIN STD_COPY S ON R.SID =S.SID

        WHERE R.SID = 748);

DELETE FROM STD_COPY WHERE SID = 748;
```

SCREENSHOT:

SQL | All Rows Fetched: 15 in 0.004 seconds

	CRSNBR	SID	GRADE
1	MGT630	987	A
2	FIN602	987	B
3	MKT610	987	A
4	FIN601	763	B
5	FIN602	763	B
6	ACC610	763	B
7	ACC610	218	A
8	ACC661	218	A
9	MGT630	218	C
10	MGT630	359	F
11	MGT681	359	F
12	MKT610	359	A
13	MKT610	862	A
14	MKT670	862	A
15	ACC610	862	B

SQL | All Rows Fetched: 9 in 0.004 seconds

	SID	SNAME	SEX	MAJOR	GPA
1	987	POIRIER	F	MGT	3.2
2	763	PARKER	F	FIN	2.7
3	218	RICHARDS	M	ACC	2.4
4	359	PELNICK	F	FIN	3.6
5	862	FAGIN	M	MGT	2.2
6	506	LEE	M	FIN	2.7
7	581	GAMBREL	F	MKT	3.8
8	372	QUICK	F	MGT	3.5
9	126	ANDERSON	M	ACC	3.7

Q9.

Delete the tables rgn_copy and std_copy from the database.

QUERY:

```
DELETE FROM RGN_COPY;

DELETE FROM STD_COPY;
```

SCREENSHOT:

```
15 rows deleted.
```

```
9 rows deleted.
```

Q10.

Create a table IPMFA with the following structure:

FID Character (3) where null values are not allowed; FNAME Varchar2(10) where null values are not allowed, EXT Varchar2(3) where null values are not allowed, DEPT Varchar2(3), RANK1 Varchar2(4), SALARY as integer. In this table, FID is the primary key.

QUERY:

```
CREATE TABLE IPMFA

(FID CHARACTER(3) NOT NULL,

FNAME VARCHAR2(10) NOT NULL,

EXT VARCHAR2(3)NOT NULL,

DEPT VARCHAR2(3),

RANKI VARCHAR2(4),

SALARY INT,

PRIMARY KEY(FID));
```

SCREENSHOT:

```
Table IPMFA created.
```

```
Name    Null?    Type
------ -------- ------------
FID     NOT NULL CHAR(3)
FNAME   NOT NULL VARCHAR2(10)
EXT     NOT NULL VARCHAR2(3)
DEPT             VARCHAR2(3)
RANKI            VARCHAR2(4)
SALARY           NUMBER(38)
```

Q11.

Create a table IPMCO with the following structure:

CRSNBR Varchar2(6) with null values not allowed, CNAME Varchar2 25) with null values not allowed, CREDIT as integer, MAXENRL as integer, FID Varchar2(3) with null values not allowed. Now, introduce FID as Foreign Key and then reference to IPMFAC table considering FID of IPMFAC table and FID of IPMCO as common field.

QUERY:

```
CREATE TABLE IPMCO
(CRSNBR Varchar2(6) NOT NULL,
CNAME VARCHAR2(25) NOT NULL,
CREDIT INT,
MAXENRL INT,
FID CHARACTER(3)NOT NULL,
FOREIGN KEY(FID) REFERENCES IPMFA(FID));
```

SCREENSHOT:

```
Table IPMCO created.

Name    Null?    Type
------- -------- ------------
CRSNBR  NOT NULL VARCHAR2(6)
CNAME   NOT NULL VARCHAR2(25)
CREDIT           NUMBER(38)
MAXENRL          NUMBER(38)
FID     NOT NULL CHAR(3)
```

Q12.

Create a view “Roster” that enables the individual to visualize selected data from the STUDENT, REGISTRATION, COURSE and FACULTY tables as being one table. This view includes course number, course name, name of person teaching the course, student ID and student name.

Display course number, course name, student ID, and student name from view “Roster” for the course number “FIN601”.

QUERY:

```
CREATE VIEW ROSTER AS
(SELECT C.CRSNBR ,C.CNAME,F.FNAME,S.SID,S.SNAME
FROM STUDENT S
INNER JOIN REGISTRATION R ON S.SID = R.SID
INNER JOIN COURSE C ON R.CRSNBR = C.CRSNBR
```

```
INNER JOIN FACULTY F ON F.FID = C.FID);
```

Or

```
CREATE VIEW "ROSTER1" AS
(
SELECT
R.CRSNBR,C.CNAME,F.FNAME,S.SID,S.SNAME
FROM STUDENT S,REGISTRATION R,COURSE C,FACULTY F
WHERE S.SID = R.SID
AND R.CRSNBR = C.CRSNBR
AND C.FID = F.FID);
```

```
B) SELECT * FROM ROSTER1;
SELECT CRSNBR ,CNAME,SID,SNAME  FROM ROSTER
WHERE CRSNBR ='FIN601';
```

SCREENSHOT:

```
View ROSTER created.
```

	CRSNBR	CNAME	FNAME	SID	SNAME
1	MGT681	INTERL. MANAGEMENT	BARGES	359	PELNICK
2	MGT681	INTERL. MANAGEMENT	BARGES	748	MEGLIN
3	FIN601	MANAGERIAL FINANCE	JARDIN	763	PARKER
4	FIN601	MANAGERIAL FINANCE	JARDIN	748	MEGLIN
5	MKT610	MARKETING FOR MANAGERS	SAMPLE	987	POIRIER
6	MKT610	MARKETING FOR MANAGERS	SAMPLE	359	PELNICK
7	MKT610	MARKETING FOR MANAGERS	SAMPLE	862	FAGIN
8	MGT630	INTRODUCTION TO MGMT	WARD	987	POIRIER
9	MGT630	INTRODUCTION TO MGMT	WARD	218	RICHARDS
10	MGT630	INTRODUCTION TO MGMT	WARD	359	PELNICK
11	MGT630	INTRODUCTION TO MGMT	WARD	748	MEGLIN
12	FIN602	INVESTMENT SKILLS	PETERS	987	POIRIER
13	FIN602	INVESTMENT SKILLS	PETERS	763	PARKER

	CRSNBR	CNAME	SID	SNAME
1	FIN601	MANAGERIAL FINANCE	763	PARKER
2	FIN601	MANAGERIAL FINANCE	748	MEGLIN

Q13.

Create an index “MAJORIND” using the MAJOR column of Student to improve performance, MAJOR descending.

QUERY:

```
CREATE  INDEX MAJORIND ON STUDENT ( MAJOR DESC);
```

SCREENSHOT:

	INDEX_OWNER	INDEX_NAME	TABLE_OWNER	TABLE_NAME	COLUMN_NAME	COLUMN_POSITION	DESCEND
1	SYSTEM	MAJORIND	SYSTEM	STUDENT	SYS_NC00006$	1	DESC

Q14.

Write a stored procedure named “Getstudents” : To list all the sname of table Student.

QUERY:

```
CREATE OR REPLACE PROCEDURE Getstudents IS
BEGIN
 FOR var IN (SELECT SNAME FROM STUDENT)
 LOOP
    DBMS_OUTPUT.PUT_LINE(var.SNAME);
 END LOOP;
END;
/
EXECUTE Getstudents;
```

SCREENSHOT:

```
Procedure GETSTUDENTS compiled

PL/SQL procedure successfully completed.
Connecting to the database DSML.
POIRIER
PARKER
RICHARDS
PELNICK
FAGIN
MEGLIN
LEE
GAMBREL
QUICK
ANDERSON
Process exited.
Disconnecting from the database DSML.
```

Q15.

Create trigger, “salary_changes” to display the following information:

Old salary:

New salary:

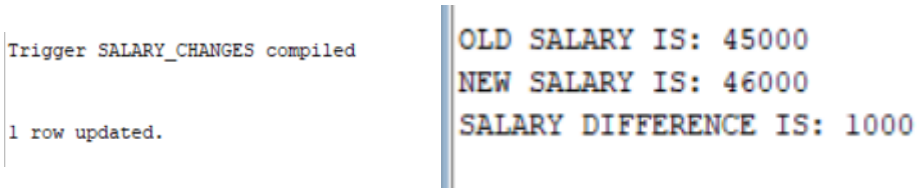
Salary difference:

The trigger will be fired when the salary difference is observed in the Faculty table.

QUERY:

```
CREATE OR REPLACE TRIGGER SALARY_CHANGES
BEFORE DELETE OR INSERT OR UPDATE ON FACULTY
FOR EACH ROW
WHEN (NEW.FID > 0)
DECLARE
  SALARY_DIFFERENCE NUMBER;
BEGIN
  SALARY_DIFFERENCE := :NEW.SALARY  - :OLD.SALARY;
  DBMS_OUTPUT.PUT_LINE('OLD SALARY IS: ' || :OLD. SALARY);
  DBMS_OUTPUT.PUT_LINE('NEW SALARY IS: ' || :NEW.SALARY);
  DBMS_OUTPUT.PUT_LINE('SALARY DIFFERENCE IS: ' || SALARY_DIFFERENCE);
END;
/
UPDATE FACULTY SET SALARY = SALARY + 1000 WHERE FID=219;
```

SCREENSHOT:



```
Trigger SALARY_CHANGES compiled


1 row updated.
```

```
OLD SALARY IS: 45000
NEW SALARY IS: 46000
SALARY DIFFERENCE IS: 1000
```

5	138	WARD	125	MGT	INST	20000
6	219	PETERS	220	FIN	FULL	46000