SQL ASSIGNMENT ON STUDENT INFORMATION SYSTEM

NAME - JAYSHREE

EMAIL ID - jayshree2406@gmail.com

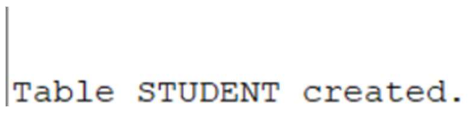
Q1. Create a table “student” with the structure/dictionary given above and insert 10 records given in the table created.

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.

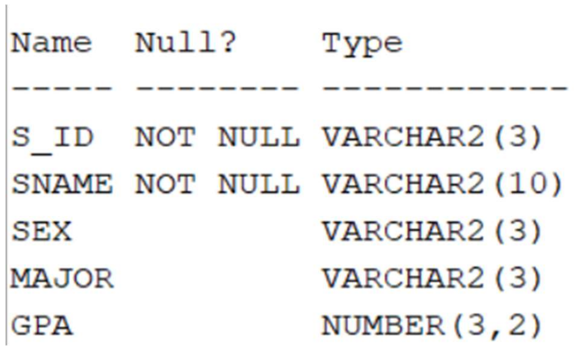
Create a table “registration” with the structure/dictionary given above and insert 18 records given in the table created.;

```
CREATE TABLE Student(
S_ID varchar2(3) NOT NULL PRIMARY KEY ,
SNAME varchar2(10) not null,
SEX varchar2(3),
MAJOR varchar2(3),
GPA decimal(3,2));
```



```
Table STUDENT created.
```

describe student;



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

SQL ASSIGNMENT ON STUDENT INFORMATION SYSTEM

CREATE TABLE Faculty(

FID varchar2(3) not null primary key,

FNAME varchar2(10)not null,

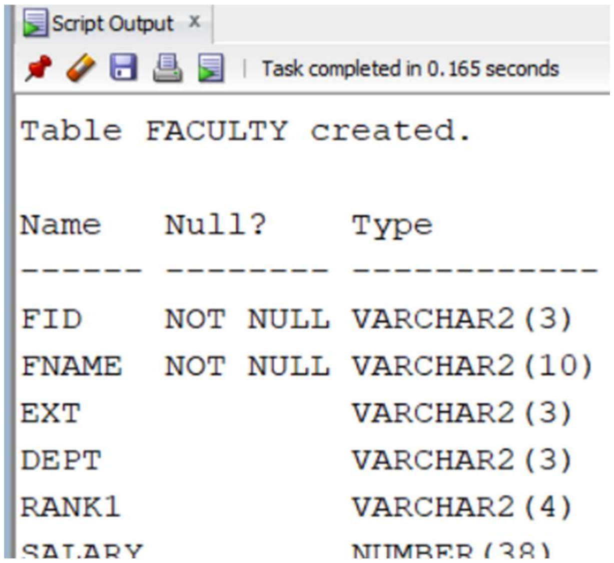
EXT varchar2(3),

DEPT varchar2(3),

RANK1 varchar2(4),

SALARY int);

describe faculty;



```
Table FACULTY created.

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)
```

CREATE TABLE Course(

CRSNBR varchar2(6) not null,

CNAME varchar2(25) not null,

CREDIT int,

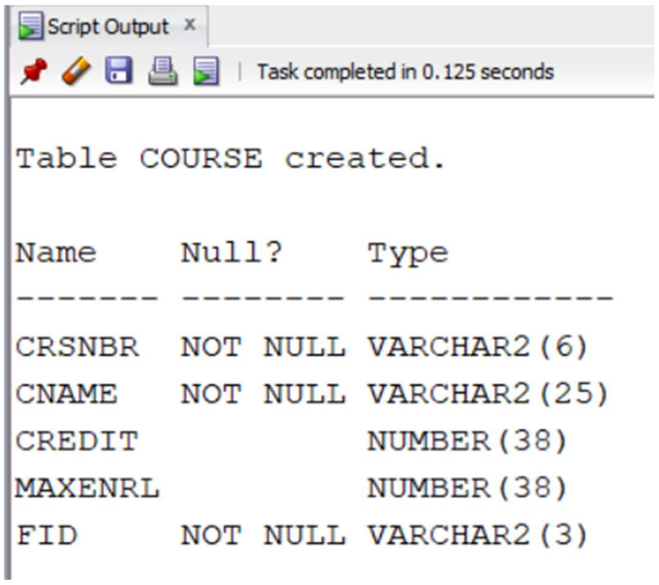
MAXENRL int,

FID varchar2(3) not null,

foreign key(FID)references faculty(FID));

describe course;

SQL ASSIGNMENT ON STUDENT INFORMATION SYSTEM



Script Output ×

Task completed in 0.125 seconds

```
Table COURSE created.

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

CREATE TABLE Registration(

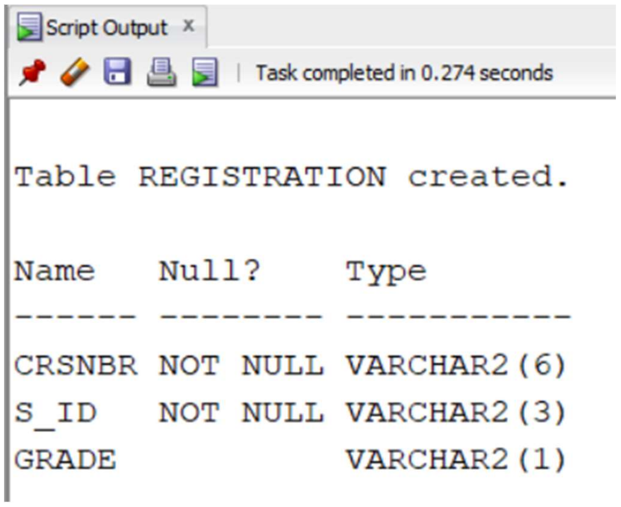
CRSNBR varchar2(6),

S_ID varchar2(3),

GRADE varchar2(1),

PRIMARY KEY(CRSNBR,S_ID));

DESCRIBE REGISTRATION;



Script Output ×

Task completed in 0.274 seconds

```
Table REGISTRATION created.

Name   Null?    Type
------ -------- -----------
CRSNBR NOT NULL VARCHAR2(6)
S_ID   NOT NULL VARCHAR2(3)
GRADE           VARCHAR2(1)
```

--STUDENT TABLE

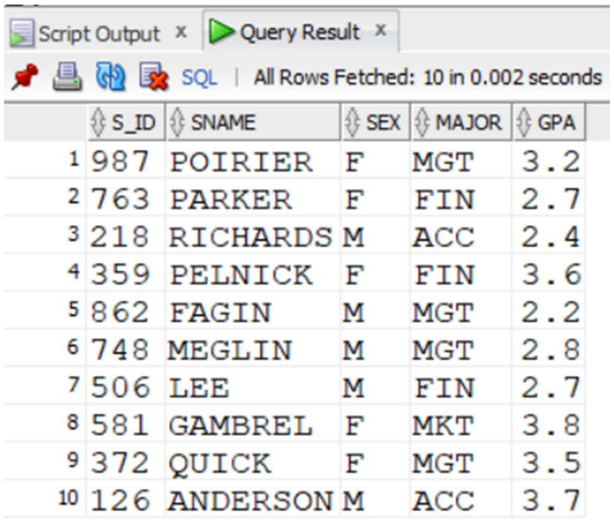
INSERT INTO Student values(987 ,'POIRIER', 'F', 'MGT', 3.2);

INSERT INTO Student values(763, 'PARKER', 'F', 'FIN', 2.7);

INSERT INTO Student values(218, 'RICHARDS', 'M', 'ACC', 2.4);

SQL ASSIGNMENT ON STUDENT INFORMATION SYSTEM

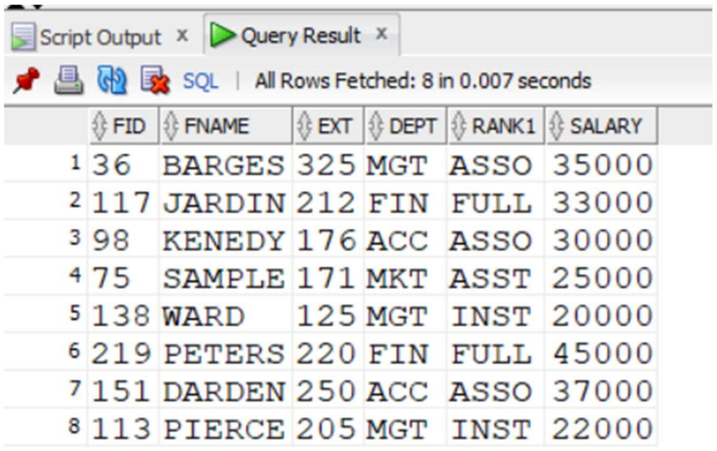
```
INSERT INTO Student values(359, 'PELNICK', 'F', 'FIN', 3.6);

INSERT INTO Student values(862, 'FAGIN', 'M', 'MGT', 2.2);

INSERT INTO Student values(748, 'MEGLIN', 'M', 'MGT', 2.8);

INSERT INTO Student values(506, 'LEE', 'M', 'FIN', 2.7);

INSERT INTO Student values(581, 'GAMBREL', 'F', 'MKT', 3.8);

INSERT INTO Student values(372, 'QUICK', 'F', 'MGT', 3.5);

INSERT INTO Student values(126, 'ANDERSON', 'M', 'ACC', 3.7);
```



	S_ID	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 TABLE

INSERT INTO Faculty values(036, 'BARGES', 325, 'MGT', 'ASSO', 35000);

INSERT INTO Faculty values(117, 'JARDIN', 212, 'FIN', 'FULL', 33000);

INSERT INTO Faculty values(098, 'KENEDY', 176, 'ACC', 'ASSO', 30000);

INSERT INTO Faculty values(075, 'SAMPLE', 171, 'MKT', 'ASST', 25000);

INSERT INTO Faculty values(138, 'WARD', 125, 'MGT', 'INST', 20000);

INSERT INTO Faculty values(219, 'PETERS', 220, 'FIN', 'FULL', 45000);

INSERT INTO Faculty values(151, 'DARDEN', 250, 'ACC', 'ASSO', 37000);

INSERT INTO Faculty values(113, 'PIERCE', 205, 'MGT', 'INST', 22000);
```

select * from faculty;



```
   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 TABLE

INSERT INTO Course values('MGT630', 'INTRODUCTION TO MGMT', 4, 30, 138);

INSERT INTO Course values('FIN601', 'MANAGERIAL FINANCE', 4, 25, 117);

INSERT INTO Course values('MKT610', 'MARKETING FOR MANAGERS', 3, 35, 075);

INSERT INTO Course values('MKT661', 'TAXATION', 3, 30, 098);

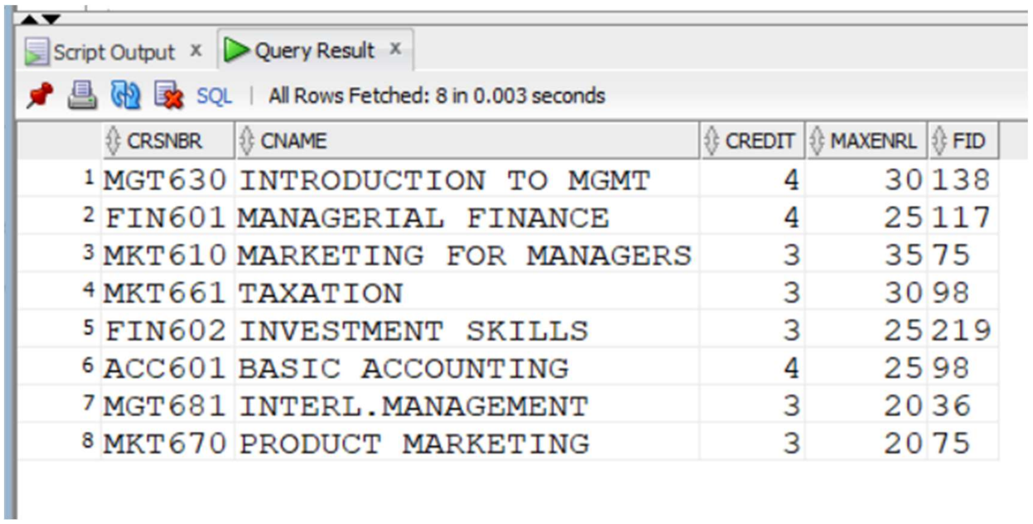
INSERT INTO Course values('FIN602', 'INVESTMENT SKILLS', 3, 25, 219);

INSERT INTO Course values('ACC601', 'BASIC ACCOUNTING', 4, 25, 098);

INSERT INTO Course values('MGT681', 'INTERL.MANAGEMENT', 3, 20, 036);

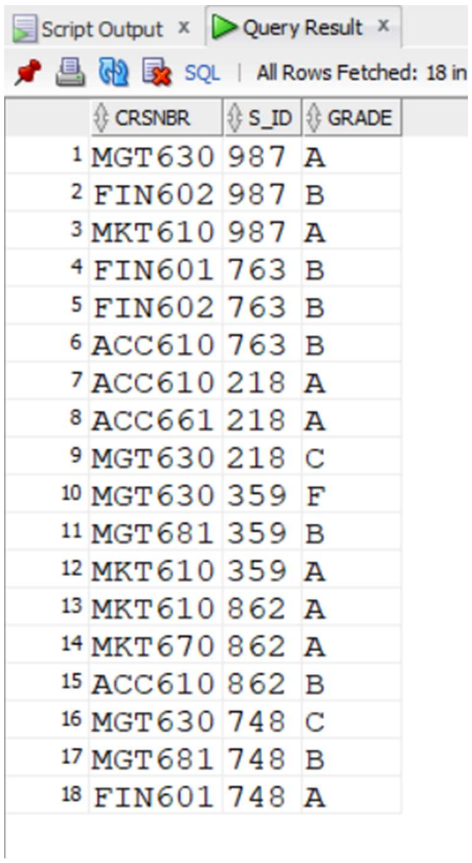
INSERT INTO Course values('MKT670', 'PRODUCT MARKETING', 3, 20, 075);

select * from Course;



```
   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 TABLE
INSERT INTO Registration values('MGT630', 987, 'A');
INSERT INTO Registration values('FIN602', 987, 'B');
INSERT INTO Registration values('MKT610', 987, 'A');
INSERT INTO Registration values('FIN601', 763, 'B');
INSERT INTO Registration values('FIN602', 763, 'B');
INSERT INTO Registration values('ACC610', 763, 'B');
INSERT INTO Registration values('ACC610', 218, 'A');
INSERT INTO Registration values('ACC661', 218, 'A');
INSERT INTO Registration values('MGT630', 218, 'C');
INSERT INTO Registration values('MGT630', 359, 'F');
INSERT INTO Registration values('MGT681', 359, 'B');
INSERT INTO Registration values('MKT610', 359, 'A');
INSERT INTO Registration values('MKT610', 862, 'A');
INSERT INTO Registration values('MKT670', 862, 'A');
INSERT INTO Registration values('ACC610', 862, 'B');
INSERT INTO Registration values('MGT630', 748, 'C');
INSERT INTO Registration values('MGT681', 748, 'B');
INSERT INTO Registration values('FIN601', 748, 'A');
select * from registration;
```



Script Output × | Query Result ×

SQL | All Rows Fetched: 18 in

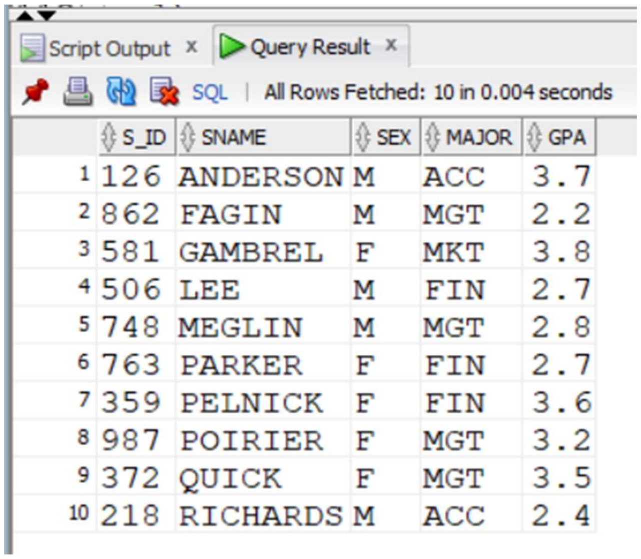
	CRSNBR	S_ID	GRADE
1	MGT630	987	A
2	FIN602	987	B
3	MKT610	987	A
4	FIN601	763	B
5	FIN602	763	B
6	ACC610	763	B
7	ACC610	218	A
8	ACC661	218	A
9	MGT630	218	C
10	MGT630	359	F
11	MGT681	359	B
12	MKT610	359	A
13	MKT610	862	A
14	MKT670	862	A
15	ACC610	862	B
16	MGT630	748	C
17	MGT681	748	B
18	FIN601	748	A

Q2. Retrieve the list of students in alphabetical order;

sol)

SELECT * from student

ORDER BY sname;



Script Output × | Query Result ×

SQL | All Rows Fetched: 10 in 0.004 seconds

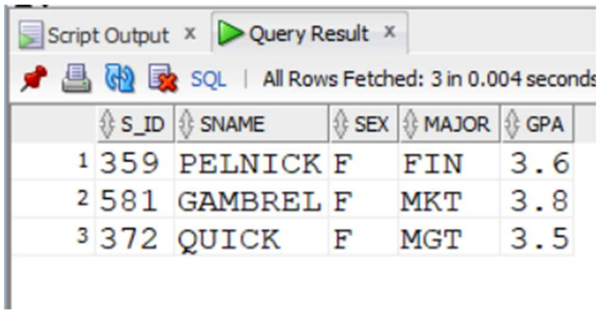
	S_ID	SNAME	SEX	MAJOR	GPA
1	126	ANDERSON	M	ACC	3.7
2	862	FAGIN	M	MGT	2.2
3	581	GAMBREL	F	MKT	3.8
4	506	LEE	M	FIN	2.7
5	748	MEGLIN	M	MGT	2.8
6	763	PARKER	F	FIN	2.7
7	359	PELNICK	F	FIN	3.6
8	987	POIRIER	F	MGT	3.2
9	372	QUICK	F	MGT	3.5
10	218	RICHARDS	M	ACC	2.4

Q3. Display a list of female students with a GPA above 3.25.;

sol)

SELECT * FROM student

WHERE sex **LIKE** 'F' **AND** gpa > 3.25;



All Rows Fetched: 3 in 0.004 seconds

	S_ID	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

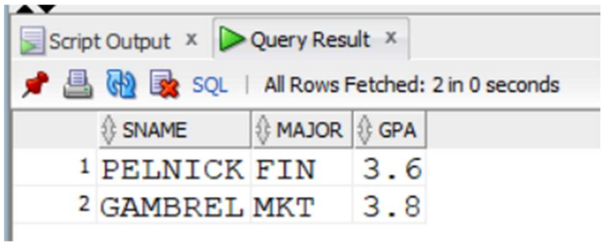
Q4. Retrieve the names, majors, and GPA of all students who have a GPA above 3.5 and who are majoring in either accounting or finance;

sol)

SELECT sname, major,gpa

FROM student

WHERE major in ('MKT' , 'FIN') **AND** gpa > 3.5;



All Rows Fetched: 2 in 0 seconds

	SNAME	MAJOR	GPA
1	PELNICK	FIN	3.6
2	GAMBREL	MKT	3.8

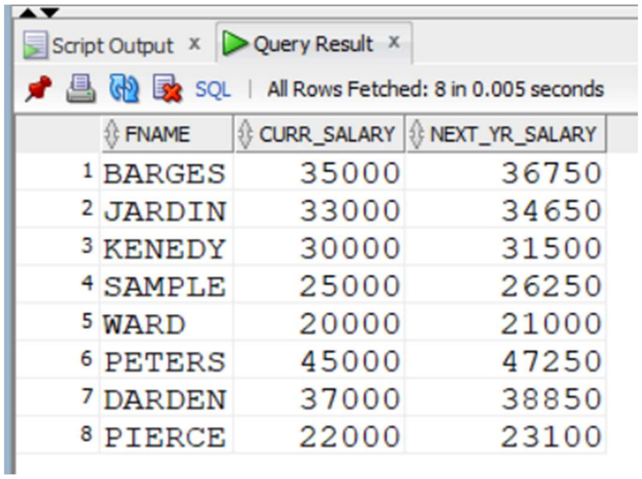
Q5. Next year every faculty member will receive a 5% salary increase. List the names of each faculty member, his/her current salary, and next years salary;

sol)

SELECT fname,salary as "CURR_SALARY",

1.05*(salary) **AS** "NEXT_YR_SALARY"

FROM faculty;



All Rows Fetched: 8 in 0.005 seconds

	FNAME	CURR_SALARY	NEXT_YR_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

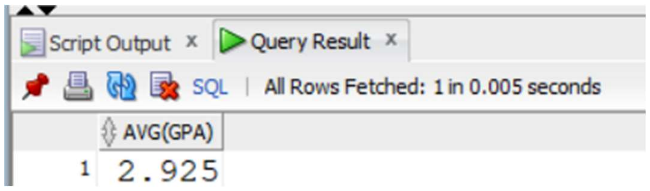
Q6. Retrieve the average GPA from student where major='MGT'.;

sol)

SELECT AVG(gpa)

FROM student

WHERE major like 'MGT';



All Rows Fetched: 1 in 0.005 seconds

	AVG(GPA)
1	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.;

SOL)

CREATE TABLE rgn_copy

AS select *

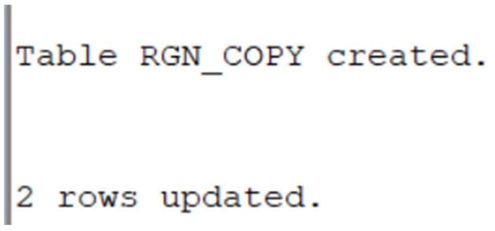
FROM registration;

UPDATE rgn_copy

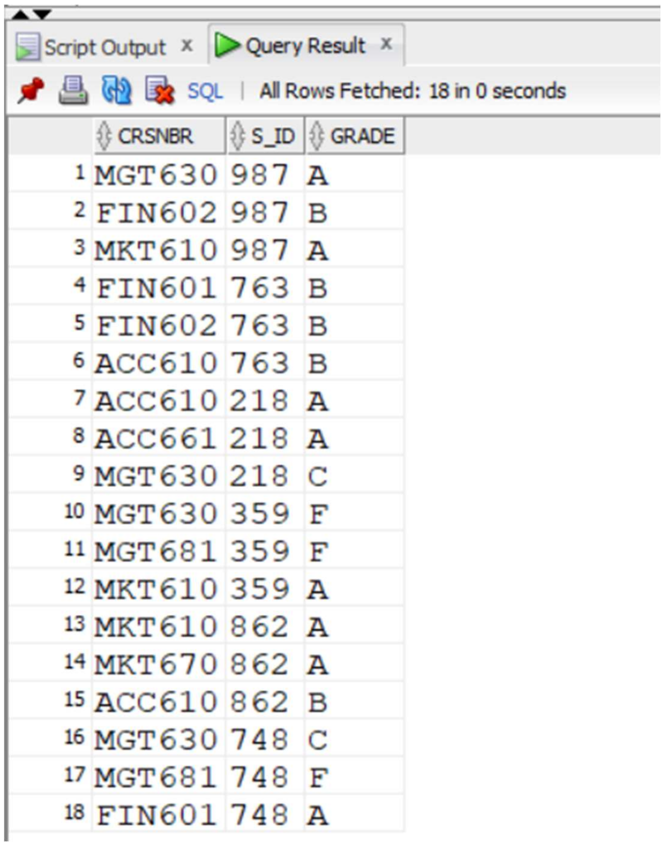
SET grade = 'F'

WHERe crsnbr = 'MGT681';

SELECT * FROM rgn_copy;



```
Table RGN_COPY created.

2 rows updated.
```



Script Output × | Query Result ×

SQL | All Rows Fetched: 18 in 0 seconds

	CRSNBR	S_ID	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
16	MGT630	748	C
17	MGT681	748	F
18	FIN601	748	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;

sol)

CREATE TABLE std_copy

AS SELECT *

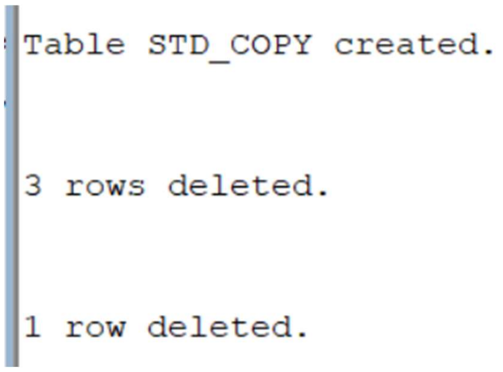
FROM student;

DELETE FROM rgn_copy

WHERE s_id = 748;

DELETE FROM std_copy

WHERE s_id = 748;



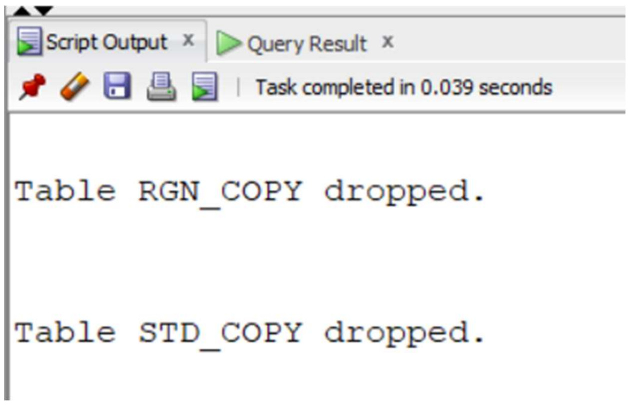
```
Table STD_COPY created.


3 rows deleted.


1 row deleted.
```

Q9. Delete the tables rgn_copy and std_copy from the database;

sol)

DROP TABLE rgn_copy;

DROP TABLE std_copy;



Task completed in 0.039 seconds

```
Table RGN_COPY dropped.


Table STD_COPY dropped.
```

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.;

sol)

CREATE TABLE IPMFA(

FID **VARCHAR2(3) NOT NULL PRIMARY KEY,**

FNAME **VARCHAR2(10) NOT NULL,**

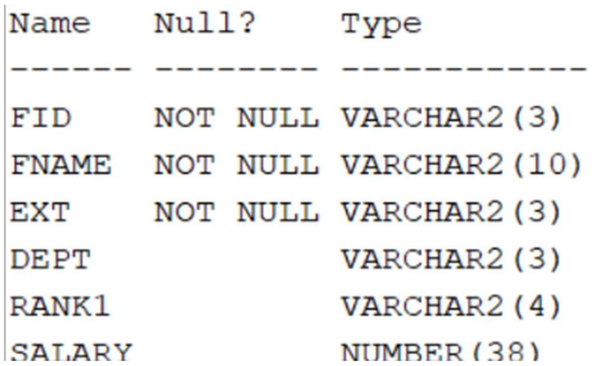
EXT **VARCHAR2(3) NOT NULL,**

DEPT **VARCHAR2(3),**

RANK1 **VARCHAR2(4),**

salary **INT**);

DESCRIBE ipmfa;



```
Name   Null?    Type
------ -------- ------------
FID    NOT NULL VARCHAR2(3)
FNAME  NOT NULL VARCHAR2(10)
EXT    NOT NULL VARCHAR2(3)
DEPT            VARCHAR2(3)
RANK1           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.;

sol)

CREATE TABLE IPMCO(

CRSNBR **VARCHAR2(6) NOT NULL,**

CNAME **VARCHAR2(25) NOT NULL,**

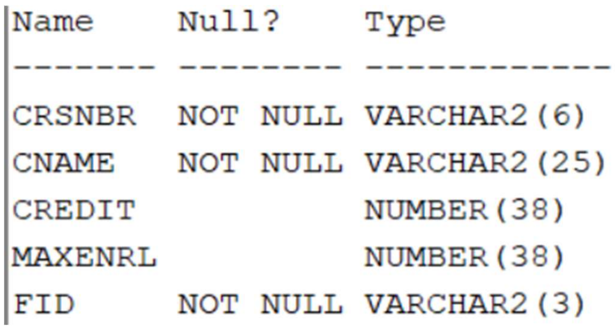
CREDIT **INT,**

MAXENRL **INT,**

FID **VARCHAR2(3) NOT NULL,**

FOREIGN KEY (FID) **REFERENCES** IPMFA(FID));

DESCRIBE IPMCO;



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

SOL)

CREATE VIEW Roster

AS SELECT c.crsnbr, c.cname,

f.fname, s.s_id,s.sname

FROM course c

LEFT JOIN registration r

ON c.crsnbr = r.crsnbr

LEFT JOIN faculty f

ON c.fid=f.fid

LEFT JOIN student s

ON s.s_id = r.s_id;

--**DISPLAY ROSTER VIEW HAVING COURSE NAME FIN601**

SELECT crsnbr as "COURSE NO",

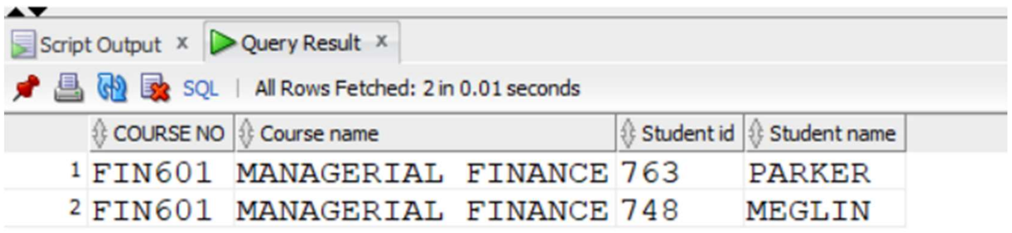
CNAME as "Course name",

S_ID **AS** "Student id",

SNAME **AS** "Student name"

FROM roster

WHERE crsnbr ='FIN601';



All Rows Fetched: 2 in 0.01 seconds

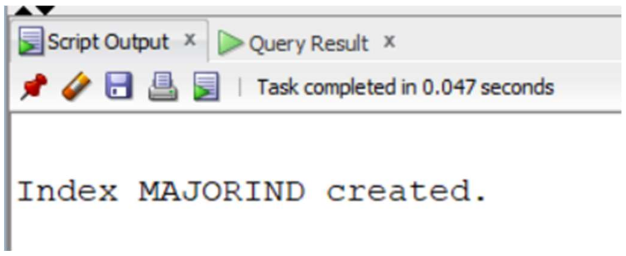
	COURSE NO	Course name	Student id	Student name
1	FIN601	MANAGERIAL FINANCE	763	PARKER
2	FIN601	MANAGERIAL FINANCE	748	MEGLIN

Q13. Create an index “MAJORIND” using the MAJOR column of Student to improve performance, MAJOR descending;

sol)

CREATE INDEX MAJORIND

ON student (major desc);



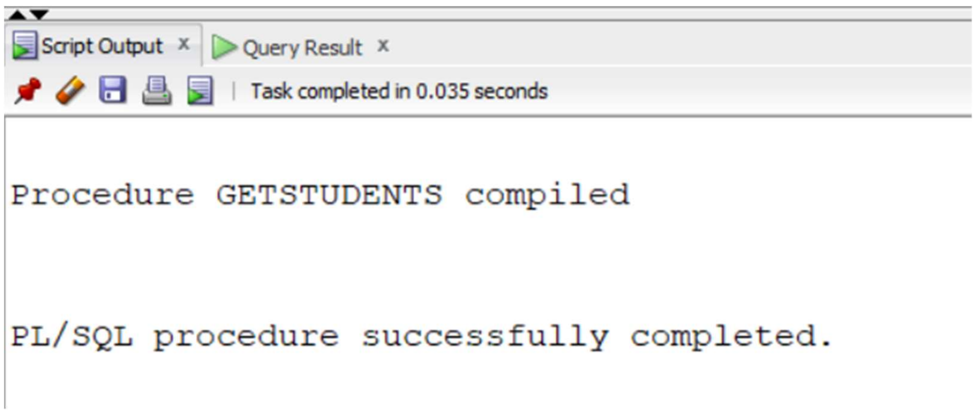
Task completed in 0.047 seconds

```
Index MAJORIND created.
```

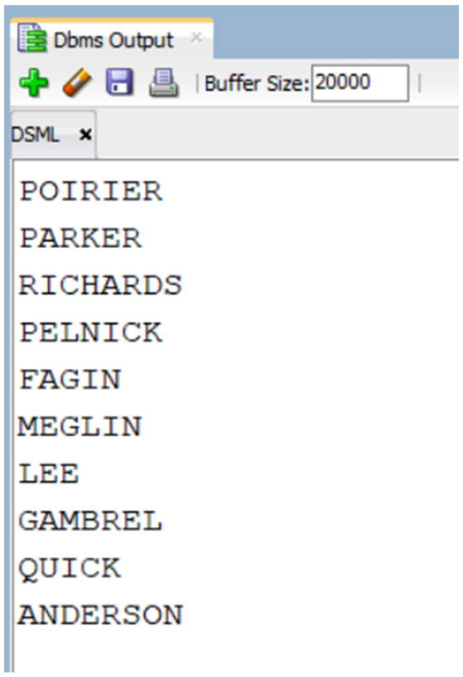
Q14. Write a stored procedure named “Getstudents” : To list all the sname of table Student;

sol)

```
CREATE OR REPLACE PROCEDURE Getstudents
AS
BEGIN
  FOR i in (select sname from student)
  LOOP
    dbms_output.put_line(i.sname);
  END LOOP;
END;
/
EXECUTE Getstudents;
```



```
Procedure GETSTUDENTS compiled


PL/SQL procedure successfully completed.
```



```
POIRIER
PARKER
RICHARDS
PELNICK
FAGIN
MEGLIN
LEE
GAMBREL
QUICK
ANDERSON
```

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.;

sol)

CREATE OR REPLACE TRIGGER salary_changes

BEFORE UPDATE OR DELETE OR INSERT ON faculty

FOR EACH ROW

When (NEW.FID>0)

DECLARE

 sal_diff **NUMBER;**

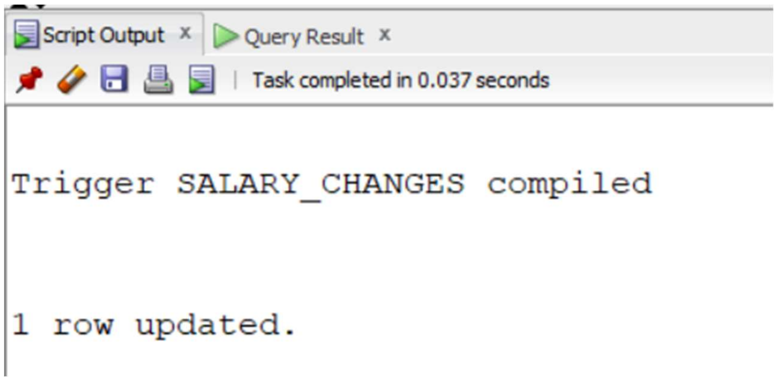
BEGIN

 sal_diff:=:NEW.salary -:OLD.salary;

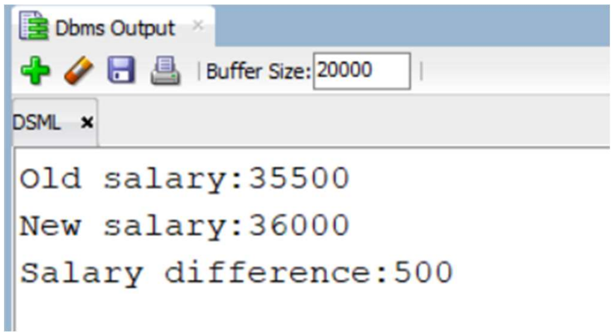
 dbms_output.put_line('Old salary:'|| :OLD.salary);

 dbms_output.put_line('New salary:' || :NEW.salary);

```
 dbms_output.put_line('Salary difference:' ||sal_diff);
END;
/


--UPDATE TABLE FOR TRIGGER ACTIVATION UPDATE FACULTY
UPDATE FACULTY
SET salary = salary+500
WHERE fid=36;
/
DROP TRIGGER
SALARY_CHANGES;
```



```
Trigger SALARY_CHANGES compiled


1 row updated.
```



```
Old salary:35500
New salary:36000
Salary difference:500
```