Follow the same formatting guidelines as the previous homework assignment.

**Copy and paste the contents of student.txt (Same as the previous lab) into your SQLPlus session. Rename the tables such that they are all prefixed with the first five letters of your lastname such as sabze_student. Make sure that the tables (student, classes and student_class) are all renamed properly before you continue.**

**Use only a single SQL statement for each of the following questions**

| | |
|---|---|
| 1 | Give a listing of all the ssns,first names and the class descriptions of all the classes the students are taking. If there are no class _descriptions display 'No description is available yet'. (USE NVL) |

```
select MS.ssn, NVL(MC.class_description,'No description is available yet') "Class_description", MS.fname
from mendo_class MC, mendo_student MS, mendo_student_class MSC
where MC.class_code = MSC.class_code and MSC.ssn = MS.ssn;
```

| SSN | Class_description | FNAME |
|---|---|---|
| 172-32-1176 | Database Programming | Johnson |
| 213-46-8915 | Introduction to C programming | Marjorie |
| 267-41-2394 | Intro to principles | Michael |
| 409-56-7008 | Database Programming | Abraham |
| 427-17-2319 | Intro to principles | Ann |
| 472-27-2349 | Introduction to C programming | Burt |
| 486-29-1786 | No description is available yet | Chastity |
| 527-72-3246 | No description is available yet | Morningstar |
| 648-92-1872 | No description is available yet | Reginald |
| 672-71-3249 | Introduction to Computers | Akiko |

Page 1 of 2 (1-10 of 14 rows)

| | |
|---|---|
| 2 | Give a listing of only the lname and the class_code for students who are taking 'Introduction to C programming'. (Inner join) |

```
select lname, MC.class_code
from mendo_class MC, mendo_student MS, mendo_student_class MSC
where MC.class_code = MSC.class_code and MSC.ssn = MS.ssn and class_description='Introduction to C programming';
```

| LNAME | CLASS_CODE |
|---|---|
| Green | 32 |
| Gringlesby | 32 |
| Hunter | 32 |

Page 1 of 1 (1-3 of 3 rows)

| | |
|---|---|
| 3 | Give a lising of all the class_descriptions and the number of students enrolled in each class for all students who are older than the average age where the total number of students for the class is more than 1 student. Order by the number of students. If there is no class description replace it with 'Other Classes'<br>(Note: Take it in steps. First do all those who are older than the average age, then do the group by, then add the having clause and then the order and then combine everything together) |

```
select MSC.class_code,class_description,count(*) "number of students"
from mendo_class MC, mendo_student MS, mendo_student_class MSC
where MS.ssn = MSC.ssn and MSC.class_Code = MC.class_code and months_between(sysdate,dob)/12>(select avg(months_between(sysdate,dob)/12) from mendo_student)
group by MSC.class_Code, MC.class_description
having count(*)>1;
```

| | CLASS_CODE | CLASS_DESCRIPTION | number of students |
|---|---|---|---|
| ☑ | 37 | Database Programming | 2 |
| ☑ | 32 | Introduction to C programming | 2 |

Page 1 of 1 (1-2 of 2 rows)

| | |
|---|---|
| 4 | Give a listing of all the classes for which no students are enrolled in (use in or not in clause) (subquery) |

```
select class_code
from mendo_class
where class_code not in(select distinct(Class_code) from mendo_student_class);
```

| | CLASS_CODE |
|---|---|
| ☑ | 14A |

Page 1 of 1 (1-1 of 1 rows)

| | |
|---|---|
| 5 | Give a listing of all the students who are not enrolled in any classes (Note: Use Exists or not Exists) |

```
select fname, lname from mendo_student s left outer join mendo_student_class sc
on s.ssn = sc.ssn where not exists (select * from mendo_student s where s.ssn = sc.ssn);
```

| | FNAME | LNAME |
|---|---|---|
| ☑ | Cheryl | Gren |
| ☑ | Cal | Al |

Page 1 of 1 (1-2 of 2 rows)

| | |
|---|---|
| 6 | create a new table that contains the list of all the students<br>and class_descriptions. Include In this table the list of all<br>students who are not enrolled in any classes (display no classes). If there are no class descriptions then display 'no description'<br>(Use combination of inner join, union and minus)<br>(Note: minus will deal with the students who are not enrolled in any classes) |

```
CREATE TABLE mendo_new_table AS
SELECT fname, lname, NVL(class_description, 'No Description') AS "Class Description"
FROM mendo_student MS inner join mendo_student_class MSC
on MS.ssn = MSC.ssn inner join mendo_class MC on MSC.class_code = MC.class_code
UNION
((SELECT fname, lname, NVL(class_description, 'No Description') AS "Class Description"
FROM mendo_student MS inner join mendo_student_class MSC
on MS.ssn = MSC.ssn inner join mendo_class MC on MSC.class_code = MC.class_code)
MINUS
(SELECT fname, lname, NVL(class_description, 'No Description') AS "Class Description"
FROM mendo_student MS inner join mendo_student_class MSC
on MS.ssn = MSC.ssn inner join mendo_class MC on MSC.class_code = MC.class_code));

select * from mendo_new_table;
```

| FNAME | LNAME | Class Description |
|---|---|---|
| Abraham | Bennet | Database Programming |
| Akiko | Yokomoto | Introduction to Computers |
| Albert | Greeenr | Introduction to Computers |
| Albert | Greeenr | No Description |
| Ann | Dull | Intro to principles |
| Burt | Gringlesby | Introduction to C programming |
| Chastity | Locksley | No Description |
| Innes | del Castillo | Database Programming |
| Johnson | White | Database Programming |
| Marjorie | Green | Introduction to C programming |

Page 1 of 2 (1-10 of 14 rows)

| 7 | repeat question 6 using a combination of inner join, union and not exists<br>(Note: Not exists will deal with the students who are not enrolled in any classes) |
|---|---|

```
select fname, lname, 'no classes' from mendo_student s where not exists
(select * from mendo_student_class sc where s.ssn = sc.ssn)
UNION
select fname, lname, class_description from mendo_student s natural join mendo_student_class
natural join mendo_class where class_description is not null
UNION
select fname, lname, 'no description' from mendo_student s natural join mendo_student_class
natural join mendo_class where class_description is null;
```

| FNAME | LNAME | 'NOCLASSES' |
|---|---|---|
| Abraham | Bennet | Database Programming |
| Akiko | Yokomoto | Introduction to Computers |
| Albert | Greeenr | Introduction to Computers |
| Albert | Greeenr | no description |
| Ann | Dull | Intro to principles |
| Burt | Gringlesby | Introduction to C programming |
| Cal | Al | no classes |
| Chastity | Locksley | no description |
| Cheryl | Gren | no classes |
| Innes | del Castillo | Database Programming |

Page 1 of 2 (1-10 of 16 rows)

| 8 | **create a view.** We want to find out which courses are being taken by the different students for all those whose age is greater than the average age. Give a listing of the course descriptions and student names (Inner join) |
|---|---|

```
create view CourseTaken as
select class_description, fname, lname from mendo_student s natural join
mendo_student_class sc natural join mendo_class
where floor(months_between(sysdate, dob) / 12) >
(select avg(floor(months_between(sysdate, dob) / 12)) from mendo_student);

select * from CourseTaken;
```

| CLASS_DESCRIPTION | FNAME | LNAME |
|---|---|---|
| Introduction to C programming | Marjorie | Green |
| Database Programming | Abraham | Bennet |
| | Chastity | Locksley |
| | Reginald | Blotchet-Halls |
| Database Programming | Innes | del Castillo |
| Introduction to C programming | Sheryl | Hunter |

Page 1 of 1 (1-6 of 6 rows)

**9** We want to find out the courses that each student is not enrolled in.
Give a listing of the course descriptions, and the students (lname) who are not taking that specific course
(Use a cartesian product and union it with a minus)

```
select lname, class_description from mendo_student, mendo_class
MINUS
select lname, class_description from mendo_student natural join
mendo_class natural join mendo_student_class;
```

| LNAME | CLASS_DESCRIPTION |
|---|---|
| Al | Database Programming |
| Al | Intro to principles |
| Al | Introduction to C programming |
| Al | Introduction to Computers |
| Al | Operating systems |
| Al | |
| Bennet | Intro to principles |
| Bennet | Introduction to C programming |
| Bennet | Introduction to Computers |
| Bennet | Operating systems |

Page 1 of 8 (1-10 of 76 rows)

**10** Use the system catalog tables to display the results to find out the following:(Note show me the SQL syntax along with your results) Only a single SQL statement for each question.

**a)** Primary key name and the columns that make up the primary key for **student** table

```
select constraint_name, column_name from user_constraints natural join
user_cons_columns where table_name = 'MENDO_STUDENT' AND constraint_type = 'P';
```

**b)** Unique key name and the columns that make up the unique key for the **student** table

```
select constraint_name, column_name from user_constraints natural join
user_cons_columns where table_name = 'MENDO_STUDENT' AND constraint_type = 'U';
```

**c)** Foreign key name, the columns that make up the foreign key and the columns it references in the parent table for **student_class** table

```
select constraint_name, search_condition r_constraint_name
from user_constraints natural join user_cons_columns
where table_name = 'MENDO_STUDENT_CLASS' AND constraint_type = 'R';
```

**d)** Name of all the check constraints and their conditions for the **student** table

```
select constraint_name, search_condition from user_constraints natural join
user_cons_columns where table_name = 'MENDO_STUDENT' AND constraint_type = 'C';
```