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 s.ssn, fname "First Name", NVL(class_description,'No description is available yet') "Class Description"
FROM chave_student s, chave_student_class sc, chave_class c
WHERE s.ssn=sc.ssn AND sc.class_code=c.class_code;
```

	SSN	First Name	Class Description
1	409-56-7008	Abraham	Database Programming
2	648-92-1872	Reginald	No description is available yet
3	427-17-2319	Ann	Intro to principles
4	998-72-3567	Albert	Introduction to Computers
5	998-72-3567	Albert	No description is available yet
6	213-46-8915	Marjorie	Introduction to C programming
7	527-72-3246	Morningstar	No description is available yet
8	472-27-2349	Burt	Introduction to C programming
9	846-92-7186	Sheryl	Introduction to C programming
10	486-29-1786	Chastity	No description is available yet
11	267-41-2394	Michael	Intro to principles
12	172-32-1176	Johnson	Database Programming
13	672-71-3249	Akiko	Introduction to Computers
14	712-45-1867	Innes	Database Programming

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 "Last Name", c.class_code "Class Code"
FROM chave_student s, chave_student_class sc, chave_class c
WHERE s.ssn = sc.ssn
      AND sc.class_code = c.class_code
      AND c.class_description = 'Introduction to C programming';
```

	Last Name	Class Code
1	Green	32
2	Gringlesby	32
3	Hunter	32

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'
(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 NVL(class_description,'Other Classes') "Class Description", COUNT(s.ssn) "Number of Students"
FROM chave_student s, chave_student_class sc, chave_class c
WHERE s.ssn = sc.ssn AND
      sc.class_code = c.class_code AND
      TRUNC(MONTHS_BETWEEN(SYSDATE,dob)/12) > (SELECT AVG(TRUNC(MONTHS_BETWEEN(SYSDATE,dob)/12)) FROM chave_student)
GROUP BY NVL(class_description,'Other Classes')
HAVING COUNT(*) > 1
ORDER BY 2;
```

	Class Description	Number of Students
1	Other Classes	2
2	Database Programming	2
3	Introduction to C programming	2

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 "Class Code", class_description "Classes With No Students"
FROM chave_class
WHERE class_code NOT IN (SELECT class_code FROM chave_student_class);
```

	Class Code	Classes With No Students
1	14A	Operating systems

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

```
SELECT *
FROM chave_student s
WHERE NOT EXISTS (SELECT sc.ssn
                  FROM chave_student_class sc
                  WHERE sc.ssn = s.ssn);
```

	SSN	LNAME	FNAME	PHONE	ADDRESS	CITY	STATE	ZIP	DOB	SALARY
1	238-95-7766	Gren	Cheryl	415 548-7723	589 Darwin Ln.	Berkeley	CA	94705	(null)	45000
2	999-00-0000	Al	Cal	615 297-2723	22 Graybar House Rd.	Nashville	TN	37215	06-FEB-98	22000

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

```
CREATE TABLE chave_student_class_desc AS
SELECT s.ssn, fname "First Name", NVL(class_description,'No description') "Class Description"
FROM chave_student s, chave_student_class sc, chave_class c
WHERE s.ssn=sc.ssn AND sc.class_code=c.class_code
UNION
(SELECT ssn, fname, 'No classes' FROM chave_student
MINUS
SELECT s.ssn, fname, 'No classes' FROM chave_student s, chave_student_class sc WHERE s.ssn=sc.ssn);
```

```
table CHAVE_STUDENT_CLASS_DESC created.
```

	SSN	First Name	Class Description
1	172-32-1176	Johnson	Database Programming
2	213-46-8915	Marjorie	Introduction to C programming
3	238-95-7766	Cheryl	No classes
4	267-41-2394	Michael	Intro to principles
5	409-56-7008	Abraham	Database Programming
6	427-17-2319	Ann	Intro to principles
7	472-27-2349	Burt	Introduction to C programming
8	486-29-1786	Chastity	No description
9	527-72-3246	Morningstar	No description
10	648-92-1872	Reginald	No description
11	672-71-3249	Akiko	Introduction to Computers
12	712-45-1867	Innes	Database Programming
13	846-92-7186	Sheryl	Introduction to C programming
14	998-72-3567	Albert	Introduction to Computers
15	998-72-3567	Albert	No description
16	999-00-0000	Cal	No classes

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

```
CREATE TABLE chave_student_class_desc AS
SELECT s.ssn, fname "First Name", NVL(class_description,'No description') "Class Description"
FROM chave_student s, chave_student_class sc, chave_class c
WHERE s.ssn=sc.ssn AND sc.class_code=c.class_code
UNION
(SELECT s.ssn, fname, 'No classes' FROM chave_student s
WHERE NOT EXISTS (SELECT sc.ssn FROM chave_student_class sc WHERE sc.ssn = s.ssn));
```

```
table CHAVE_STUDENT_CLASS_DESC created.
```

	SSN	First Name	Class Description
1	172-32-1176	Johnson	Database Programming
2	213-46-8915	Marjorie	Introduction to C programming
3	238-95-7766	Cheryl	No classes
4	267-41-2394	Michael	Intro to principles
5	409-56-7008	Abraham	Database Programming
6	427-17-2319	Ann	Intro to principles
7	472-27-2349	Burt	Introduction to C programming
8	486-29-1786	Chastity	No description
9	527-72-3246	Morningstar	No description
10	648-92-1872	Reginald	No description
11	672-71-3249	Akiko	Introduction to Computers
12	712-45-1867	Innes	Database Programming
13	846-92-7186	Sheryl	Introduction to C programming
14	998-72-3567	Albert	Introduction to Computers
15	998-72-3567	Albert	No description
16	999-00-0000	Cal	No classes

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)

```
DROP VIEW chave_student_classes_view;

CREATE VIEW chave_student_classes_view AS SELECT s.ssn, fname "First Name", NVL(class_description,'No description') "Class Description"
FROM chave_student s, chave_student_class sc, chave_class c
WHERE s.ssn=sc.ssn AND
      sc.class_code=c.class_code AND
      TRUNC(MONTHS_BETWEEN(SYSDATE,dob)/12) > (SELECT AVG(TRUNC(MONTHS_BETWEEN(SYSDATE,dob)/12)) FROM chave_student)
ORDER BY 1;

SELECT * FROM chave_student_classes_view;
```

```
view CHAVE_STUDENT_CLASSES_VIEW created.
```

	SSN	First Name	Class Description
1	213-46-8915	Marjorie	Introduction to C programming
2	409-56-7008	Abraham	Database Programming
3	486-29-1786	Chastity	No description
4	648-92-1872	Reginald	No description
5	712-45-1867	Innes	Database Programming
6	846-92-7186	Sheryl	Introduction to C programming

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 s.ssn, fname || ' ' || lname AS "Name", c.class_code AS "Class Code", NVL(class_description,'No description') AS "Class Description"
FROM chave_student s, chave_student_class sc, chave_class c
MINUS
SELECT s.ssn, fname || ' ' || lname, c.class_code, NVL(class_description,'No description')
FROM chave_student s, chave_student_class sc, chave_class c
WHERE s.ssn=sc.ssn AND sc.class_code=c.class_code
ORDER BY 1;
```

NOTE: The whole output was 91 lines long. Only the first 18 lines are shown below.

	SSN	Name	Class Code	Class Description
1	172-32-1176	Johnson White	1	No description
2	172-32-1176	Johnson White	14A	Operating systems
3	172-32-1176	Johnson White	3	Introduction to Computers
4	172-32-1176	Johnson White	32	Introduction to C programming
5	172-32-1176	Johnson White	34	Intro to principles
6	172-32-1176	Johnson White	55	No description
7	213-46-8915	Marjorie Green	1	No description
8	213-46-8915	Marjorie Green	14A	Operating systems
9	213-46-8915	Marjorie Green	3	Introduction to Computers
10	213-46-8915	Marjorie Green	34	Intro to principles
11	213-46-8915	Marjorie Green	37	Database Programming
12	213-46-8915	Marjorie Green	55	No description
13	238-95-7766	Cheryl Gren	1	No description
14	238-95-7766	Cheryl Gren	14A	Operating systems
15	238-95-7766	Cheryl Gren	3	Introduction to Computers
16	238-95-7766	Cheryl Gren	32	Introduction to C programming
17	238-95-7766	Cheryl Gren	34	Intro to principles
18	238-95-7766	Cheryl Gren	37	Database Programming

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 uc.table_name, uc.constraint_name, r_constraint_name, column_name
FROM user_constraints uc, user_cons_columns ucc
WHERE uc.constraint_name = ucc.constraint_name AND
      uc.table_name='CHAVE_STUDENT' AND
      constraint_type = 'P';
```

	TABLE_NAME	CONSTRAINT_NAME	R_CONSTRAINT_NAME	COLUMN_NAME
1	CHAVE_STUDENT	CHAVE_STUDENT_PK	(null)	SSN

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

```
SELECT uc.table_name, uc.constraint_name, r_constraint_name, column_name
FROM user_constraints uc, user_cons_columns ucc
WHERE uc.constraint_name = ucc.constraint_name AND
      uc.table_name='CHAVE_STUDENT' AND
      constraint_type = 'U';
```

	TABLE_NAME	CONSTRAINT_NAME	R_CONSTRAINT_NAME	COLUMN_NAME
1	CHAVE_STUDENT	CHAVE_STUDENT_UK	(null)	LNAME
2	CHAVE_STUDENT	CHAVE_STUDENT_UK	(null)	FNAME

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 uc.table_name, uc.constraint_name, r_constraint_name, column_name
FROM user_constraints uc, user_cons_columns ucc
WHERE uc.constraint_name = ucc.constraint_name AND
      uc.table_name='CHAVE_STUDENT_CLASS' AND
      constraint_type = 'R';
```

	TABLE_NAME	CONSTRAINT_NAME	R_CONSTRAINT_NAME	COLUMN_NAME
1	CHAVE_STUDENT_CLASS	CHAVE_ST_FK	CHAVE_STUDENT_PK	SSN

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

```
SELECT table_name, constraint_name, r_constraint_name, status
FROM user_constraints
WHERE table_name='CHAVE_STUDENT' AND
      constraint_type = 'C';
```

	TABLE_NAME	CONSTRAINT_NAME	R_CONSTRAINT_NAME	STATUS
1	CHAVE_STUDENT	SYS_C0046377	(null)	ENABLED
2	CHAVE_STUDENT	SYS_C0046378	(null)	ENABLED
3	CHAVE_STUDENT	SYS_C0046379	(null)	ENABLED