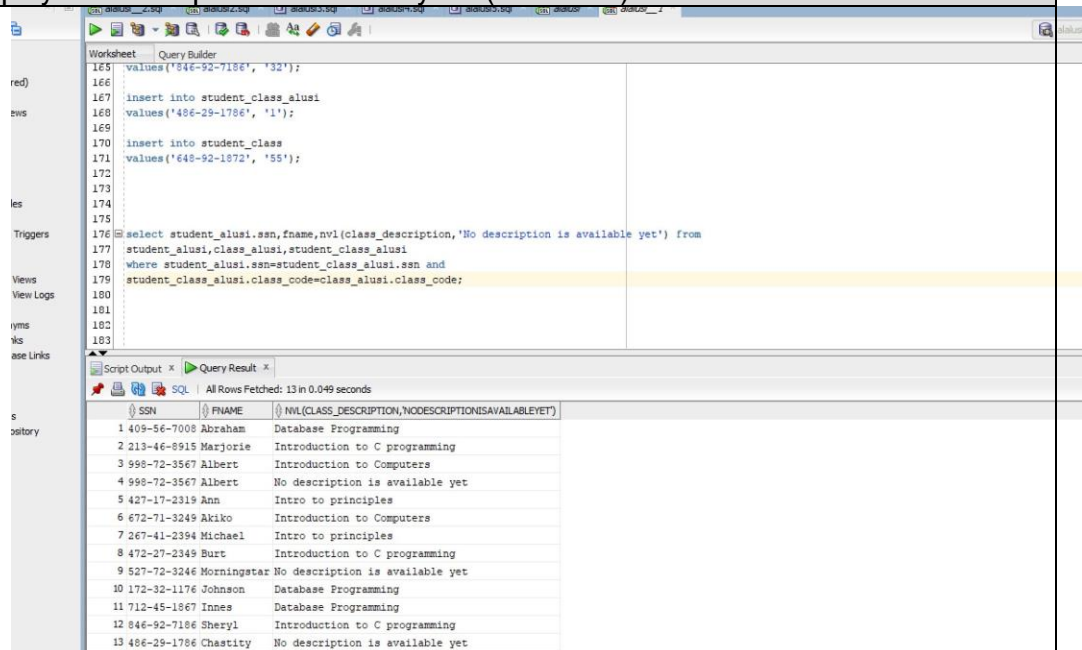


Follow the same formatting guidelines as the previous homework assignment.

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)																																										
	 <p>The screenshot shows the SQL Developer interface. The 'Worksheet' tab is active, displaying a SQL script with the following content:</p> <pre> 165 values('846-92-7186', '32'); 166 167 insert into student_class_alusi 168 values('486-29-1786', '1'); 169 170 insert into student_class 171 values('648-92-1872', '55'); 172 173 174 175 176 select student_alusi.ssn, fname, nvl(class_description, 'No description is available yet') from 177 student_alusi, class_alusi, student_class_alusi 178 where student_alusi.ssn=student_class_alusi.ssn and 179 student_class_alusi.class_code=class_alusi.class_code; 180 181 182 183 </pre> <p>The 'Query Result' tab is also visible, showing the results of the query. The results are as follows:</p> <table> <tr> <th>SSN</th><th>FNAME</th><th>NVL(CLASS_DESCRIPTION, NODESCRIPTIONISAVAILABLEYET)</th></tr> <tr><td>1 409-56-7008</td><td>Abraham</td><td>Database Programming</td></tr> <tr><td>2 213-46-8915</td><td>Marjorie</td><td>Introduction to C programming</td></tr> <tr><td>3 998-72-3567</td><td>Albert</td><td>Introduction to Computers</td></tr> <tr><td>4 998-72-3567</td><td>Albert</td><td>No description is available yet</td></tr> <tr><td>5 427-17-2319</td><td>Ann</td><td>Intro to principles</td></tr> <tr><td>6 672-71-3249</td><td>Akiko</td><td>Introduction to Computers</td></tr> <tr><td>7 267-41-2394</td><td>Michael</td><td>Intro to principles</td></tr> <tr><td>8 472-27-2349</td><td>Burt</td><td>Introduction to C programming</td></tr> <tr><td>9 527-72-3246</td><td>Morningstar</td><td>No description is available yet</td></tr> <tr><td>10 172-32-1176</td><td>Johnson</td><td>Database Programming</td></tr> <tr><td>11 712-45-1867</td><td>Innes</td><td>Database Programming</td></tr> <tr><td>12 846-92-7186</td><td>Sheryl</td><td>Introduction to C programming</td></tr> <tr><td>13 486-29-1786</td><td>Chastity</td><td>No description is available yet</td></tr> </table>	SSN	FNAME	NVL(CLASS_DESCRIPTION, NODESCRIPTIONISAVAILABLEYET)	1 409-56-7008	Abraham	Database Programming	2 213-46-8915	Marjorie	Introduction to C programming	3 998-72-3567	Albert	Introduction to Computers	4 998-72-3567	Albert	No description is available yet	5 427-17-2319	Ann	Intro to principles	6 672-71-3249	Akiko	Introduction to Computers	7 267-41-2394	Michael	Intro to principles	8 472-27-2349	Burt	Introduction to C programming	9 527-72-3246	Morningstar	No description is available yet	10 172-32-1176	Johnson	Database Programming	11 712-45-1867	Innes	Database Programming	12 846-92-7186	Sheryl	Introduction to C programming	13 486-29-1786	Chastity	No description is available yet
SSN	FNAME	NVL(CLASS_DESCRIPTION, NODESCRIPTIONISAVAILABLEYET)																																									
1 409-56-7008	Abraham	Database Programming																																									
2 213-46-8915	Marjorie	Introduction to C programming																																									
3 998-72-3567	Albert	Introduction to Computers																																									
4 998-72-3567	Albert	No description is available yet																																									
5 427-17-2319	Ann	Intro to principles																																									
6 672-71-3249	Akiko	Introduction to Computers																																									
7 267-41-2394	Michael	Intro to principles																																									
8 472-27-2349	Burt	Introduction to C programming																																									
9 527-72-3246	Morningstar	No description is available yet																																									
10 172-32-1176	Johnson	Database Programming																																									
11 712-45-1867	Innes	Database Programming																																									
12 846-92-7186	Sheryl	Introduction to C programming																																									
13 486-29-1786	Chastity	No description is available yet																																									
2	Give a listing of only the lname and the class_code for students who are taking 'Introduction to C programming'. (Inner join)																																										

The screenshot shows a SQL Developer window with a query script in the 'Worksheet' tab. The script includes several SQL statements: inserting a student, inserting a class, selecting student and class details, and selecting class descriptions for students older than the average age. The 'Query Result' tab shows the output of the last query, which is a table with two columns: LNAME and CLASS_CODE. The results are:

LNAME	CLASS_CODE
1 Green	32
2 Gringlesby	32
3 Hunter	32

- 3 Give a listing 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)

The screenshot shows a SQL Developer window with a complex query script in the 'Worksheet' tab. The script uses a subquery to find the average age of students and then selects class descriptions and the number of students enrolled in each class for students older than the average age. The 'Query Result' tab shows the output of the last query, which is a table with two columns: CLASS_DESCRIPTION and NUMBEROFSTUDENTS. The results are:

CLASS_DESCRIPTION	NUMBEROFSTUDENTS
1 Introduction to C programming	2
2 Database Programming	2

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

Worksheet Query Builder

```

200
201
202
203
204
205
206 SELECT NVL(c.class_description, 'Other Classes') class_description, count(s.ssn) NumberOfStudents
207 FROM Student_alusi s
208 INNER JOIN student_class_alusi sc ON sc.ssn = s.ssn
209 INNER JOIN class_alusi c ON c.class_code = sc.class_code
210 WHERE ROUND((SYSDATE - TO_DATE(Dob))/365.25, 5) >= (SELECT AVG(ROUND((SYSDATE - TO_DATE(Dob))/365.25, 5)) FROM Student_alusi)
211 GROUP BY sc.class_code, c.class_description
212 HAVING COUNT(sc.ssn) > 1
213 ORDER BY COUNT(sc.ssn);
214
215
216
217
218

```

Script Output x Query Result x

SQL All Rows Fetched: 1 in 0.045 seconds

CLASS_CODE	CLASS_DESCRIPTION
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)

Worksheet Query Builder

```

215
216
217
218
219
220
221
222
223
224
225
226
227
228 SELECT * FROM Class_alusi WHERE class_code NOT IN (SELECT DISTINCT class_code FROM student_class_alusi);
229
230
231
232
233 SELECT * FROM student_alusi s WHERE NOT EXISTS (SELECT sc.SSN FROM student_class_alusi sc WHERE sc.SSN = s.SSN);

```

Script Output x Query Result x

SQL All Rows Fetched: 3 in 0.043 seconds

SSN	LNAME	FNAME	PHONE	ADDRESS	CITY	STATE	ZIP	DOB	SALARY
1 648-92-1872	Blotchet-Halls	Reginald	503 745-6402	55 Hillsdale Bl.	Corvallis	OR	97330	01-FEB-77	43000
2 238-95-7766	Gren	Cheryl	415 548-7723	589 Darwin Ln.	Berkeley	CA	94705	(null)	45000
3 999-00-0000	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)

Worksheet Query Builder

```

253
254
255
256
257
258 CREATE TABLE student_class_description_alusi AS
259 (SELECT s.ssn, s.fname, s.lname, NVL( c.class_description, 'No Description') Class_Description
260 FROM STUDENT_alusi s INNER JOIN STUDENT_CLASS_alusi sc on sc.ssn = s.ssn
261 INNER JOIN Class_alusi c ON c.class_code = sc.class_code
262 UNION
263 (SELECT s.ssn, s.fname, s.lname, 'No Description' Class_Description
264 FROM student_alusi s
265 MINUS
266 SELECT s.ssn, s.fname, s.lname, NVL( c.class_description, 'No Description') Class_Description
267 FROM STUDENT_alusi s INNER JOIN STUDENT_CLASS_alusi sc on sc.ssn = s.ssn
268 INNER JOIN Class_alusi c ON c.class_code = sc.class_code
269 )
270 )
271 );

```

Script Output x Query Result x

Task completed in 0.053 seconds

table STUDENT_CLASS_ALUSI_DESCRIPTION created.
table STUDENT_CLASS_DESCRIPTION_ALUSI created.

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)

SQL Worksheet History

Worksheet Query Builder

```

264 FROM student_alusi s
265 MINUS
266 SELECT s.ssn, s.fname, s.lname, NVL( c.class_description, 'No Description') Class_Description
267 FROM STUDENT_alusi s INNER JOIN STUDENT_CLASS_alusi sc on sc.ssn = s.ssn
268 INNER JOIN Class_alusi c ON c.class_code = sc.class_code
269 )
270 )
271 );
272
273
274
275 CREATE TABLE new_table_alusi AS(SELECT fname||' '||lname AS "Name", NVL(class_description, 'No description') AS "Class Description"
276 FROM student_alusi, class_alusi, student_class_alusi WHERE student_alusi.ssn=student_class_alusi.ssn AND student_class_alusi.class_code=class_alu
277 UNION (SELECT fname||' '||lname AS "Name", 'No classes' FROM (SELECT ssn FROM student_alusi
278 WHERE NOT EXISTS (SELECT ssn FROM student_class_alusi WHERE student_alusi.ssn=student_class_alusi.ssn)) NATURAL JOIN student_alusi));
279
280

```

Script Output x

Task completed in 0.066 seconds

Error starting at line : 275 in command -
CREATE TABLE new_table_alusi AS(SELECT fname||' '||lname AS "Name", NVL(class_description, 'No description') AS "Class Description"
FROM student_alusi, class_alusi, student_class_alusi WHERE student_alusi.ssn=student_class_alusi.ssn AND student_class_alusi.class_code=class_alu
UNION (SELECT fname||' '||lname AS "Name", 'No classes' FROM (SELECT ssn FROM student_alusi
WHERE NOT EXISTS (SELECT ssn FROM student_class_alusi WHERE student_alusi.ssn=student_class_alusi.ssn)) NATURAL JOIN student_alusi));
Error at Command Line : 278 Column : 133
Error report -
SQL Error: ORA-00933: SQL command not properly ended
00933. 00000 - "SQL command not properly ended"
'Cause:
'Action:
table NEW_TABLE_ALUSI created.

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)

The screenshot shows an SQL Worksheet with a query editor and a results pane. The query is as follows:

```

286
287
288
289
290
291
292
293
294
295
296 SELECT fname||' '||lname AS "Name", NVL(class_description, 'Other Classes') AS "class description" FROM class_alusi, student_alusi, student_class_a
297 WHERE student_alusi.ssn=student_class_alusi.ssn AND class_alusi.class_code=student_class_alusi.class_code AND
298 TRUNC(MONTHS_BETWEEN(SYSDATE, DOB)/12)>(SELECT AVG(TRUNC(MONTHS_BETWEEN(SYSDATE, DOB)/12)) FROM student_alusi);
299
300
301
302
303

```

The results pane shows the following data:

Name	class description
1 Abraham Bennet	Database Programming
2 Marjorie Green	Introduction to C programming
3 Innes del Castillo	Database Programming
4 Sheryl Hunter	Introduction to C programming
5 Chastity Locksley	Other Classes

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)

The screenshot shows an SQL Worksheet with a query editor. The query is as follows:

```

300
301
302
303
304
305
306
307
308
309
310 SELECT class_description, lname FROM class_alusi, student_alusi MINUS SELECT lname, class_description FROM student_alusi, class_alusi, student_clas
311 WHERE class_alusi.class_code=student_class_alusi.class_code AND student_class_alusi.ssn=student_alusi.ssn;
312

```

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.

SQL Worksheet | History

Worksheet | Query Builder

```
317
318
319
320
321
322
323
324
325
326
327 SELECT NVL(c.class_description,'Other Classes') AS "Class Description",
328 COUNT(sc.class_code) AS "Number of Students"
329 FROM student_class_alusi sc INNER JOIN class_alusi c ON c.class_code = sc.class_code
330 WHERE sc.ssn IN (SELECT ssn FROM student_alusi
331 WHERE (TRUNC(MONTHS_BETWEEN(sysdate,dob)/12)) >
332 (SELECT AVG(TRUNC(MONTHS_BETWEEN(sysdate,dob)/12)) FROM student_alusi))
333 GROUP BY c.class_description HAVING COUNT(sc.class_code) > 1 ORDER BY class_description;
334
```

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.05 seconds

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