Igor Oleshko
CSC 134
05/08/2022
Professor Sabzevary

Homework 6: Join

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)

```
81  SELECT ssn, fname, nvl(class_description, 'no description is available yet'
82  FROM student NATURAL JOIN student_class NATURAL JOIN class;
83
```

SSN	FNAME	CLASS_DESCRIPTION
172-32-1176	Johnson	Database Programming
213-46-8915	Marjorie	Introduction to C programming
267-41-2394	Michael	Intro to principles
409-56-7008	Abraham	Database Programming
427-17-2319	Ann	Intro to principles
472-27-2349	Burt	Introduction to C programming
486-29-1786	Chastity	no description is available yet
527-72-3246	Morningstar	no description is available yet
648-92-1872	Reginald	no description is available yet
672-71-3249	Akiko	Introduction to Computers
712-45-1867	Innes	Database Programming
846-92-7186	Sheryl	Introduction to C programming

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

```
87  SELECT lname, c.class_code
88  FROM student s, student_class sc, class c
89  WHERE s.ssn = sc.ssn AND sc.class_code = c.cl
90  class_description LIKE 'Introduction to C pro
```

LNAME	CLASS_CODE
Green	32
Gringlesby	32
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)

```
93  SELECT nvl(class_description, 'other classes'), COUNT(*)
94  FROM student s NATURAL JOIN student_class sc NATURAL JOIN class c
95  WHERE MONTHS_BETWEEN(sysdate, dob)/12 > (SELECT AVG(MONTHS_BETWEEN(sysdate, dol
96  GROUP BY class_description
97  HAVING COUNT(*) > 1
98  ORDER BY 2;
```

NVL(CLASS_DESCRIPTION,'OTHERCLASSES')	COUNT(*)
Introduction to C programming	2
other classes	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)

```
102  SELECT class_description, class_code FROM class
103  WHERE class_code NOT IN (SELECT class_code FROM st
```

CLASS_DESCRIPTION	CLASS_CODE
Operating systems	14A

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

```
107  SELECT ssn, fname FROM student s
108  WHERE NOT EXISTS (SELECT * FROM studen
109  WHERE s.ssn = sc.ssn);
```

SSN	FNAME
238-95-7766	Cheryl
999-00-0000	Cal

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)

```
114  CREATE TABLE new_table AS
115  SELECT fname, lname, nvl(class_description, 'no description') cl
116      FROM student s INNER JOIN student_class sc ON s.ssn = sc.ssn
117      INNER JOIN class c ON sc.class_code = c.class_code
118  UNION
119  (
120      SELECT fname, lname, 'no classes' FROM student
121          MINUS
122      SELECT fname, lname, 'no classes' FROM student_class sc INNE
123      ON s.ssn = sc.ssn
124  )
```

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)

```
128  CREATE TABLE new_table AS
129  SELECT fname, lname, nvl(class_description, 'no description') c
130      FROM student s INNER JOIN student_class sc ON s.ssn = sc.ss
131      INNER JOIN class c ON sc.class_code = c.class_code
132  UNION
133  (
134      SELECT fname, lname, 'no classes' FROM student s
135      WHERE NOT EXISTS (SELECT * FROM student_class sc
136      WHERE s.ssn = sc.ssn)
137  )
```

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)

```
141  CREATE VIEW new_view AS
142  SELECT class_description, fname, lname FROM
143  student s INNER JOIN student_class sc ON s.ssn =
144  INNER JOIN class c ON sc.class_code = c.class_cc
145  WHERE (MONTHS_BETWEEN(sysdate, dob)/12) >
146  (SELECT AVG(MONTHS_BETWEEN(sysdate, dob)/12) FRC
```

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)

```
151  SELECT class_description, lname
152  FROM student CROSS JOIN class
153      MINUS
154  SELECT class_description, lname
155  FROM student NATURAL JOIN student_class NATURAL
```

Introduction to Computers	Locksley
Introduction to Computers	O'Leary
Introduction to Computers	White
Introduction to Computers	del Castillo

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

```
158  SELECT constraint_name, column_name
159  FROM user_constraints NATURAL JOIN all_co
160  WHERE table_name = 'STUDENT'
161  AND constraint_type = 'P';
```

CONSTRAINT_NAME	COLUMN_NAME
STUDENT_PK	SSN

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

```
165  SELECT constraint_name, column_name
166  FROM user_constraints NATURAL JOIN all_cc
167  WHERE table_name = 'STUDENT'
168  AND constraint_type = 'U';
```

CONSTRAINT_NAME	COLUMN_NAME
STUDENT_UK	LNAME
STUDENT_UK	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

```
82  SELECT constraint_name, column_name, r_const
83  FROM user_constraints NATURAL JOIN all_cons_
84  WHERE table_name = 'STUDENT_CLASS'
85  AND constraint type = 'R';
```

CONSTRAINT_NAME	COLUMN_NAME	R_CONSTRAINT_NAME

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

```
179  SELECT constraint_name, search_cor
180  FROM user_constraints
181  WHERE table_name = 'STUDENT'
182  AND constraint_type = 'C';
```

CONSTRAINT_NAME	SEARCH_CONDITION
SYS_C0086737843	"SSN" IS NOT NULL
SYS_C0086737844	"LNAME" IS NOT NULL
SYS_C0086737845	"FNAME" IS NOT NULL