Assignment 3 CIS3530

Due: Nov 28th Total marks: 40 (+10 for bonus question)

Learning Goals: by the end of this assignment you should be able to write PL/pgsql functions, stored procedures and triggers.

Submission Instructions: There are 6 sql files to submit plus an additional one if you attempt the bonus question.

Questions 1 to 5 are based on the IRCU schema you must have created for Questions 2 and 5 of Assignment 2. You may use your own tables and data from assignment 2 or use csv files given in A3_csv_files.zip to create data for this assignment.

Question 1 (4 marks)

Write a PL/pgsql function that returns the number of agents hired by a university. For example,

SELECT numberAgents ('Toronto');

```
numberagents
--------------
           3
(1 row)
```

Name the file to submit: A3Q1_lastname_firstname.sql

Question 2 (6 marks)

Write a PL/pgsql function that takes an agent id as input and returns the max commission earned by the agent. If the agent id is not found in the schema, then the function displays a message that it is an invalid id, along with the complete list of agents in the schema sorted on agent id. A sample scenario is shown below. (Reminder: RAISE NOTICE is used to display messages and RAISE EXCEPTION is used to display a pl/pgsql error message.

```
SELECT find_max_commission (1) AS "MAX Commission";

MAX Commission
----------------
           8.00
(1 row)
```

```
SELECT find_max_commission (10);

NOTICE:  Invalid agent id. Valid agent ids are:
NOTICE:  1: MAHIA JAIN
NOTICE:  2: SUKH VIN
NOTICE:  3: JIN LU
NOTICE:  4: HOANG CHEN
NOTICE:  5: ELON MACRON
NOTICE:  6: VANSH SUD
NOTICE:  8: JIM SPIDER
NOTICE:  11: ASHA CHUG
NOTICE:  14: HUN HAO
NOTICE:  20: MELISSA MENSAH
NOTICE:  40: RITU TRIVEDI
ERROR:  Invalid agent id
CONTEXT:  PL/pgSQL function find_max_commission(numeric) line 15 at
RAISE
```

Name the file to submit: A3Q2_lastname_firstname.sql

Question 3 (8 marks)

Write a PL/pgsql function that generates an email for each agent (in addition to the gmail one) given their aid using their first name, last name and name of the university. Note that domain for universities of Brock and Trent have letter 'u' after their name (e.g. brocku.ca), Windsor, Sherbrooke and Toronto have letter 'u' before their name (e.g. utoronto.ca) and Guelph has letters 'uo' before its name (e.g. uoguelph.ca). For example,

```
SELECT generateEmail(1);

     generateemail
-----------------------
 mahia.jain.india@uoguelph.ca
(1 row)
```

Name the file to submit: A3Q3_lastname_firstname.sql

Question 4. (12 marks)

Write a PL/pgsql procedure that lists all countries, total number of students that each university has recruited from different countries and a list of degrees offered by them. For each university, the display should be sorted in descending order of number of students. A sample report is shown below.

In addition to displaying the report, for each University in this report, you must add a new row to a table named recruit_stats. For example, <'Guelph', 1488, 3> is added for University of Guelph.

The structure of the table is:

University	VARCHAR
numRecruited	INTEGER
NumDegreesOffered	INTEGER

Sample Report:

****

University of Guelph

Recruited:

980 students from China

430 students from India

78 students from USA

Total number of students = 1488

Degree offered:

M. Eng.

MAC

MSW

****

University of Windsor

Recruited:

3000 students from China

1000 students from India

690 students from France

Total number of students = 4690

Degree offered:

B. Sc.

M. Eng.

MAC

MSW

BAC

****

And so on (for each University in the database)

Name the file to submit: A3Q4_lastname_firstname.sql

Question 5 (10 marks)

a. Write a trigger called ensure_case for table Agents that change the letters in first and last names of agents to uppercase before the data is actually inserted or updated in the table. For example, if the insert statement given by a user is:

INSERT INTO Agents (fname, lname, aid) VALUES ('Ritu', 'chaturvedi', 13);

this trigger must change the letters in fname to RITU and lname to CHATURVEDI before the row gets inserted into table Agents.

b. Write a trigger for table University so that anytime the url is changed, an alert message is displayed on the screen (for example, 'University of Guelph is changing its url').

Name the files to submit: A3Q5a_lastname_firstname.sql and A3Q5b_lastname_firstname.sql

(BONUS Question) Question 6: 10 marks

Create a table called POSSIBLE_IDS with 1 field called ID (VARCHAR (60)).

Create a stored procedure/function called generate_id that takes the first name, last name and date of birth of an employee (stored as VARCHAR), and generates a list of possible login ids for the person by using combinations of first name, lastname, age and the sun sign of the person and stores this list in the table POSSIBLE_IDS.

You may use built-in functions such as current_date, age and date_part to get the age. For example,

SELECT (date_part('year', age(current_date, '1976-01-12')));

```
 date_part
-----------
        41
(1 row)
```

To read more on date functions, visit
https://www.postgresql.org/docs/8.4/static/functions-datetime.html#FUNCTIONS-DATETIME-EXTRACT

For sun-signs, you may use the following:

- Aries - March 21 - April 20
- Taurus - April 21 - May 21
- Gemini - May 22 - June 21
- Cancer - June 22 - July 22
- Leo - July 23 -August 21
- Virgo - August 22 - September 23
- Libra - September 24 - October 23
- Scorpio - October 24 - November 22
- Sagittarius - November 23 - December 22
- Capricorn - December 23 - January 20
- Aquarius - January 21 - February 19
- Pisces - February 20- March 20

Sample Input/Output:

SELECT generate_id ('Ash','Bagley','1976-01-12');

generates the following ids and stores these results in table POSSIBLE_IDS. You do not need to use an algorithm to generate these combinations – you may hard code them.

Ash_Bagley
Ash_Bagley_41
Ash_41
Bagley_41
Ash_Bagley_Capricorn
Bagley_Capricorn
Ash_Capricorn
Ash_Bagley_Capricorn_41

Name of file to submit: A3Bonus_lastname_firstname.sql that includes (1) create table command for POSSIBLE_IDS and (2) script for procedure generate_id.