

School of Information Technology

Department of Computer Science



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Faculty of Engineering, Built Environment and
Information Technology

COS326 Database Systems: Practical 7 2022

Release Date: 02 October 2022

Submission Date: 09 October 2022

Lecturer: Mr S.M Makura

Total: 50 Marks

Objectives

1. Learn how to use SQL to load XML data into an object-relational database.
2. Learn how to write SQL queries that access and process XML data in an object- relational database using XPath expressions.
3. Learn how to write SQL queries that generate XML documents and JSON objects from object relational database tables.

For this practical exercise you will use the PostgreSQL database and the file *staff_list.xml* available from ClickUP. When you are done: You must submit the following files:

- a. **LoadQueries.sql** which contains all statements necessary to create the database tables and functions for loading XML data into the PostgreSQL database tables.
- b. **SelectQueries.sql** which contains all statements that provide reports from the XML data stored in the database (SELECT statements).
- c. Compress the above documents into an archive and upload it to ClickUP **before** the due date/time. The file name for the archive must have your student number as part of the file name, e.g. **uXXXXXXXX** where XXXXXXXX is your student number.

Task 1: Load data from an XML file into database tables [30 marks]

You are provided with the file *staff_list.xml*. For this task, you will write plpgsql functions and SQL statements to load XML data from a file stored in the file system into PostgreSQL database tables. You should use the following tables:

Table name	Table columns	Description
xml_staff_list	filename, xml_document	stores the xml file content in one row. The content of the XML data file is loaded into this table.
xml_staff	entry_number, xml_data	stores the xml fragments of the column xml_document in xml_staff_list as many rows, one row per staff entry.
shred_staff_list	branchNo, staffNo, fname, lname, position, DOB, salary	Stores the shredded version of the column xml_document, i.e. in a table with one column for each XML tag value.

When you work in pgAdmin 4, the current working directory in MS Windows is the folder *C:\Program Files\PostgreSQL\14\data* (for Linux/Mac OS, refer to the PostgreSQL documentation). Create a folder *MyXMLfiles* under the *data* folder and copy the file *staff_list.xml* to this folder.

In pgAdmin 4, type the statements:

```
SELECT pg_ls_dir('.') and  
SELECT pg_ls_dir('./MyXMLfiles')
```

and observe the results.

The following statement is also useful:

```
SELECT name, setting FROM pg_settings WHERE name = 'data_directory'
```

For the loading of the XML document into PostgreSQL you will use the function *pg_read_file*. Refer to the PostgreSQL documentation for details of this function. The following SQL statement examples show how to obtain data from an xml file:

```
SELECT pg_read_file('./MyXMLfiles/staff_list.xml');           -- read xml data  
as text  
SELECT CAST(pg_read_file('./MyXMLfiles/staff_list.xml') AS xml); -- read & convert  
to xml
```

To be done:

1. Write the SQL statements to create the three database tables described in the table above. You are expected to determine the suitable data types for the table columns.
(3 marks)
2. Write the SQL statement that will read the content of file *staff_list.xml* and then load the file name and file content into one row of the *xml_staff_list* table.
(5 marks)
3. Write the plpgsql function *loadXMLfragments (text, xml)* which returns a boolean result. This function uses the XPath expression passed as the text argument (\$1) to obtain each <STAFF>...</STAFF> entry/fragment from the xml argument (\$2) and inserts records into the *xml_staff* table. Each inserted record consists of an entry number (with the first entry having entry_number = 1) and an xml fragment of the form <STAFF>...</STAFF>. The function returns true after successful insertion of all records. [Hint: use the xpath function to generate an array of the xml fragments].
(8 marks)
4. Now write an SQL statement which inserts xml fragments into the *xml_staff* table. The SQL statement should pass the contents of the *xml_document* column of the *xml_staff_list* table to the function *loadXMLfragments*. [Hint: use a sub-query.]
(3 marks)
5. Write the plpgsql function *loadshreddedXML (text, xml)* which returns a boolean result. This function uses the XPath expression passed as the text argument (\$1) to obtain each <STAFF>...</STAFF> entry/fragment from the xml argument (\$2). The function then obtains the values for branchNo, STAFFNO, FNAME, LNAME, POSITION, DOB, SALARY from the fragment and inserts a record into the *shred_staff_list* table. Each inserted record consists of the branchNo, staffNo, fname, lname, position, DOB and salary. The function returns true after successful insertion of all records.
(8 marks)
6. Now write an SQL statement which inserts xml tag values into the *shred_staff_list* table. The SQL statement should pass the contents of the *xml_document* column of the *xml_staff_list* table to the function *loadshreddedXML*.
(3 marks)

Marks will only be awarded for functions and queries that produce correct results.

Task 2: Query XML data stored in database tables**[14 marks]**

Write SQL statements to provide the following reports from the *XML_staff* table.

1. List all the details of the staff whose staff number is SL21. (2 marks)
2. List the first name, last name and branch number of all staff at branch number B003. (5 marks)
3. List the staff number and salary for all staff. The salary must be listed as integer values, and the report must have exactly one row for each staff. (3 marks)
4. Compute and show the total salary (aggregate) for all staff (2 marks)
5. Compute and show the average salary (aggregate) for all staff (2 marks)

Task 3: Generation of XML documents and JSON objects from database tables**[6 marks]**

1. Write the SQL statement to generate the XML schema document for the *shred_staff_list* table. (1 mark)
2. Write the SQL statement to obtain the details of all staff who earn a salary of 30000 from the *shred_staff_list* table and convert the details into an XML document. (2 marks)
3. Write the SQL statement to obtain the details of all staff who earn a salary of 30000 from the *shred_staff_list* table and convert the details into JSON objects. [Hint: use the *row_to_json* function]. (3 marks)