# 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

**Release Date:** 18 August 2022
**Submission Date:** 26 August 2022

**Lecturer:** Mr S.M Makura

**Total: 50 Marks**

## A. Objectives

1. Get exposure to the PostgreSQL object-relational DBMS (ORDBMS).
2. Learn how to implement domains, types (UDTs), sequences, tables, functions and table inheritance in an ORDBMS.
3. Appreciate the differences between a relational DBMS, object-oriented DBMS and object-relational DBMS.

### B. Submission Procedure:

You should have PostgreSQL installed on your computer in order to complete this practical.

1. Your practical submission must consist of the following files:
   - **a. UML.pdf –** This file contains the class diagram for the database. Your name and student number must appear in this document.
   - **b. CreateStatements.sql -** This file contains all statements necessary to create the database 'objects'
     i.e. domains, types, sequences, tables and functions.
   - **c. InsertQueries.sql -** This file contains all statements that add to the content of the database (INSERT statements).
   - **d. SelectQueries.sql -** This file contains all statements that provide reports from the database (SELECT statements)
   - e. Compress the above documents into an archive (zip file) and upload it to ClickUP using the Practical 3 submission link **before** the due date and time. The file name for the archive must have your student number as part of the file name, e.g. ***uXXXXXXXX.zip*** (*XXXXXXXX is your student number)*
2. The practical will be marked through a live demo on Discord.

**NO LATE** submissions will be accepted after the submission date and time has lapsed. Do not wait till the last minute to submit and start giving excuses that you faced technical challenges when you tried to submit.

# Question 1: PostgresSQL ORDMS

## Task 1: Domains, sequences, user-defined types and table inheritance [30 marks]

The Student Records Unit of a university needs to keep records of all **students**: **undergraduates** and **postgraduates,** as well as **degree programs,** and **courses.** For each **student** the student number, name (title, first name, surname), date of birth, degree program, and the year of study (1st year, 2nd year, etc), should be recorded. For an **undergraduate student**, the courses currently registered for, should additionally be recorded. For a **postgraduate student**, the category (part time or full time), and supervisor (title, first name, surname), should additionally be recorded. For each **degree program** the degree name, number of years, department, and faculty should be recorded. For each **course**, the course code, course name, course credits, and department should be recorded.

The above information is summarised as follows:

**Entities, attributes and functions:**

| **Entity** | **Attributes (values)** | **functions (and examples of return values)** |
|---|---|---|
| **Student** | student number (of six numeric characters), full names (title, first name, surname), where title is one of : Ms, Mev, Miss, Mrs, Mr, Mnr), date of birth, degree code (e.g. BSc), year of study (e.g. 1, 2, 3, etc) | *personFullNames* as 'title first name surname' (e.g. Ms Good Student) (e.g. Mr Serious Student)<br><br>*ageInYears* (e.g. 22 years) (computed from date of birth) |
| **Undergraduate** (is-a student) | **additional attributes:**<br><br>courseRegistration (an array of course codes, e.g. ['COS301', 'COS302', 'COS326',….] ) | *isRegisteredFor(…)* returns true if the student is registered for the course with the specified course code.<br><br>*isFinalYearStudent(…)* returns true if the student is in the final year of his/her degree program. |
| **Postgraduate** (is-a student) | **additional attributes:** category (part time or full time), supervisor (title, first name, surname) where title is one of : Ms, Mev, Miss, Mrs, Mr, Mnr, Dr, Prof | *isFullTime(..), isPartTime(..)* The *is..* functions return a Boolean value (true or false )<br><br>*personFullNames* as 'title first name surname' (e.g. Prof. Very Serious) |
| **DegreeProgram** | degree code, (e.g. BSc) degree name (e.g. Bachelor of Science), number of years, faculty | |
| **Course** | course code (e.g. COS326), course name (e.g. Database Systems), department (e.g. Computer Science), credits | |

## Your task as the student is to:

1. Create a UML class diagram to show the above relationships (2 marks)
2. Identify the classes that should be implemented as tables.
3. Write SQL statements to create:
   a. all necessary domains, enum and structured types (CREATE DOMAIN, CREATE TYPE …) (5 marks)
   b. a sequence that will generate a surrogate (primary) key for each table. (CREATE SEQUENCE). You must specify the start value for the sequence. (3 marks)
   c. the tables to store the objects in the class hierarchy (hierachies) (CREATE TABLE … (INHERITS) ) (8 marks)
   d. all the functions for the tables. For functions whose input arguments are UDTs and the output is text, consider using the CAST operator. **HINT**: Use the following syntax for all functions:

```
CREATE FUNCTION functionName( pmt1Type, pmt2Type,....)
      RETURNS returntype AS
       $$
            SELECT expression to compute AS functionname;
       $$ LANGUAGE SQL;
```

(12 marks)

4. **Create a database in PostgreSQL called *studentsDB* and run all the SQL statements in (3) above to create the database 'objects'.** As you create the database 'objects' in pgAdmin 4, right click on your database in the object browser and select 'Refresh' so that you can see the created 'objects'. **Note:** marks for part (3) will only be awarded if the database 'objects' actually get created.

## Task 2: Inserting data into the Database tables [6 marks]

Use the *INSERT INTO* SQL statement to add the following data into the database. Execute some SELECT statements to confirm that you entered the data correctly.

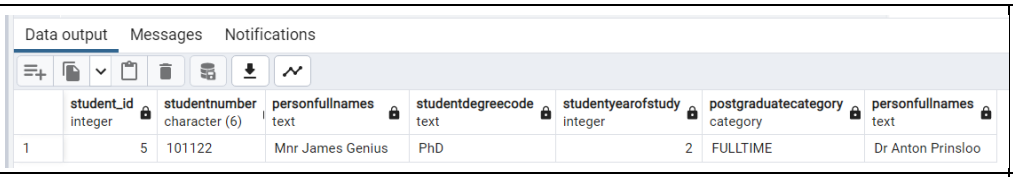
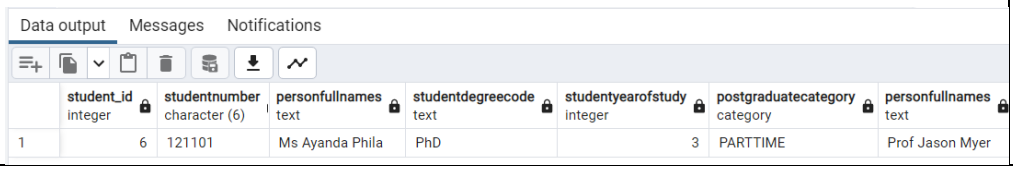
| | **Attribute values:** note that the values of attributes **… key** are generated by the SEQUENCEs that you created | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Degree Program** | **degree key** | **degree code** | **degree name** | **number of years** | **Faculty** | | | | |
| | | BSc | Bachelor of Science | 3 | EBIT | | | | |
| | | BIT | Bachelor of IT | 4 | EBIT | | | | |
| | | PhD | Philosophiae Doctor | 5 | EBIT | | | | |
| | | | | | | | | | |
| **Course** | **course key** | **course code** | **course name** | **department** | **credits** | | | | |
| | | COS301 | Software Engineering | Computer Science | 40 | | | | |
| | | COS326 | Database Systems | Computer Science | 20 | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | MTH301 | Discrete Mathematics | Mathematics | 15 | | | | |
| | | PHL301 | Logical | Philosophy | 15 | | | | |
| | | | | | | | | | |
| **Undergraduate** | **student key** | **student number** | **student name (title, fname, surname)** | **date of birth (dd-mm-yyyy)** | **degree code** | **year of study** | **courseRegistration** | | |
| | | 140010 | choose title & names | 10-01-1996 | BSc | 3 | COS301, COS326, MTH301 | | |
| | | 140015 | choose title & names | 25-05-1995 | BSc | 3 | COS301, PHL301, MTH301 | | |
| | | 131120 | choose title & names | 30-01-1995 | BIT | 3 | COS301, COS326, PHL301 | | |
| | | 131140 | choose title & names | 20-02-1996 | BIT | 4 | COS301, COS326, MTH301, PHL301 | | |
| | | | | | | | | | |
| **Postgraduate** | **student key** | **student number** | **student name (title, fname, sname)** | **date of birth** | **degree code** | **year of study** | **category** | **supervisor (title,fname, sname)** | |
| | | 101122 | choose title & names | 15-06-1987 | PhD | 2 | full time | choose title & names | |
| | | 121101 | choose title & names | 27-04-1985 | PhD | 3 | part tim | choose title & names | |

## Task 3: Querying the Database [14 marks]

Write SELECT statements to provide the reports described in the following table:

| Required report: List of all | Columns in report, e.g., | Marks |
|---|---|---|
| 1. **students personal details** (you must use the *personFullNames* and ageInYears functions) | Data output \| Messages \| Notifications<br>student_id [PK] integer \| studentnumber character (6) \| personfullnames text \| ageinyears integer<br>1 \| 1 \| 140010 \| Mr Sheunesu Makura \| 25<br>2 \| 2 \| 140015 \| Mrs Thando Mandela \| 27<br>3 \| 3 \| 131120 \| Mev Janie Kruger \| 27<br>4 \| 4 \| 131140 \| Miss Palesa Mohlare \| 26<br>5 \| 5 \| 101122 \| Mnr James Genius \| 35<br>6 \| 6 \| 121101 \| Ms Ayanda Phila \| 37 | **2** |
| 2.**undergraduate students** registration details (you must use the *personFullNames* function) | Data output \| Messages \| Notifications<br>student_id integer \| studentnumber character (6) \| personfullnames text \| studentdegreecode text \| studentyearofstudy integer \| courseregistration character[] (6)<br>1 \| 1 \| 140010 \| Mr Sheunesu Makura \| BSc \| 3 \| {COS301,COS326,MTH301}<br>2 \| 2 \| 140015 \| Mrs Thando Mandela \| BSc \| 3 \| {COS301,PHL301,MTH301}<br>3 \| 3 \| 131120 \| Mev Janie Kruger \| BIT \| 3 \| {COS301,COS326,PHL301}<br>4 \| 4 \| 131140 \| Miss Palesa Mohlare \| BIT \| 4 \| {COS301,COS326,MTH301,PHL301} | **2** |
| 3. **postgraduate students** registration details (you must use the *personFullNames* function) | Data output \| Messages \| Notifications<br>student_id integer \| studentnumber character (6) \| personfullnames text \| studentdegreecode text \| studentyearofstudy integer \| postgraduatecategory category \| personfullnames text<br>1 \| 5 \| 101122 \| Mnr James Genius \| PhD \| 2 \| FULLTIME \| Dr Anton Prinsloo<br>2 \| 6 \| 121101 \| Ms Ayanda Phila \| PhD \| 3 \| PARTTIME \| Prof Jason Myer | **2** |
| 4.**undergraduate students** registration details for final year students (you must use the *isFinalYear* function) | Data output \| Messages \| Notifications<br>student_id integer \| studentnumber character (6) \| personfullnames text \| studentdegreecode text \| studentyearofstudy integer \| courseregistration character[] (6)<br>1 \| 1 \| 140010 \| Mr Sheunesu Makura \| BSc \| 3 \| {COS301,COS326,MTH301}<br>2 \| 2 \| 140015 \| Mrs Thando Mandela \| BSc \| 3 \| {COS301,PHL301,MTH301}<br>3 \| 4 \| 131140 \| Miss Palesa Mohlare \| BIT \| 4 \| {COS301,COS326,MTH301,PHL301} | **3** |
| 5.**undergraduate students** registration details for students registered for , e.g. COS326 (you must use the *isRegisteredFor* functions) | Data output \| Messages \| Notifications<br>student_id integer \| studentnumber character (6) \| personfullnames text \| studentdegreecode text \| studentyearofstudy integer \| courseregistration character[] (6)<br>1 \| 1 \| 140010 \| Mr Sheunesu Makura \| BSc \| 3 \| {COS301,COS326,MTH301}<br>2 \| 3 \| 131120 \| Mev Janie Kruger \| BIT \| 3 \| {COS301,COS326,PHL301}<br>3 \| 4 \| 131140 \| Miss Palesa Mohlare \| BIT \| 4 \| {COS301,COS326,MTH301,PHL301} | **3** |

| | | |
|---|---|---|
| 6. **full-time postgraduate students** registration details (you must use the *isFullTime*function) | Data output Messages Notifications<br>  <br>student_id integer \| studentnumber character (6) \| personfullnames text \| studentdegreecode text \| studentyearofstudy integer \| postgraduatecategory category \| personfullnames text<br>1 \| 5 \| 101122 \| Mnr James Genius \| PhD \| 2 \| FULLTIME \| Dr Anton Prinsloo | **1** |
| 7. **part-time postgraduate students** registration details (you must use the *isPartTime*function) | Data output Messages Notifications<br>  <br>student_id integer \| studentnumber character (6) \| personfullnames text \| studentdegreecode text \| studentyearofstudy integer \| postgraduatecategory category \| personfullnames text<br>1 \| 6 \| 121101 \| Ms Ayanda Phila \| PhD \| 3 \| PARTTIME \| Prof Jason Myer | **1** |

## Additional instructions to aid you in the practical

**In order to create the *.sql files, proceed as follows:**

a. Open the **pgAdmin 4** query tool.

b. For the file **CreateStatements.sql :** copy all the CREATE statements into the query pane and select Save. Take careful note of the quotation marks in your queries. They must be the 'pure text quotation marks' and not the 'MS Word quotation marks'. Save the file by clicking the save button (see figure below), and type in the file name and save in the desired location.

c. For the file **InsertQueries.sql** : Open a new query connection window (see figure below) and copy all the INSERT statements into the query pane and select Save. Again, save the file by clicking the save button (see figure below), and type in the file name and save in the desired location.

d. For the file **SelectQueries.sql:** Open a new query connection window (see Figure below) and copy all the SELECT statements into the query pane and select Save. Again, save the file by clicking the save button (see figure below), and type in the file name and save in the desired location.

e. Test that your queries are working by doing the following: (1) delete the database you created (2) create the database again (3) open the query tool. (4) open the file Create Statements.sql and click on the execute query button. Repeat this for the InsertQueries.sql and then for the SelectStatements.sql file, you need to execute your queries **one by one** to see the output for each sql query statement.