

School of Information Technology

Department of Computer Science



COS326 Database Systems

Release Date: 15 August 2024

Submission Date: 22 August 2024

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. **uXXXXXXXXX.zip** (*XXXXXXXXX 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: PostgreSQL ORDMS

Task 1: Domains, sequences, user-defined types and table inheritance

[30 marks]

Scenario

You have been hired by the University of Pretoria ITS division as a graduate software developer. ITS uses PostgreSQL to store staff records and have given you the task to use PostgreSQL to store student records.

The student records that need to be kept of all **students** are: **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.

Source: Makura S.M (2024)

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)	<i>personFullNames</i> as 'title first name surname' (e.g. Ms Good Student) (e.g. Mr Serious Student) <i>ageInYears</i> (e.g. 22 years) (computed from date of birth)
Undergraduate (is-a student)	additional attributes: courseRegistration (an array of course codes, e.g. ['COS301', 'COS302', 'COS326',....])	<i>isRegisteredFor(...)</i> returns true if the student is registered for the course with the specified course code. <i>isFinalYearStudent(...)</i> 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	<i>isFullTime(..)</i> , <i>isPartTime(..)</i> The <i>is..</i> functions return a Boolean value (true or false) <i>personFullNames</i> 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 (hierarchies) (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.

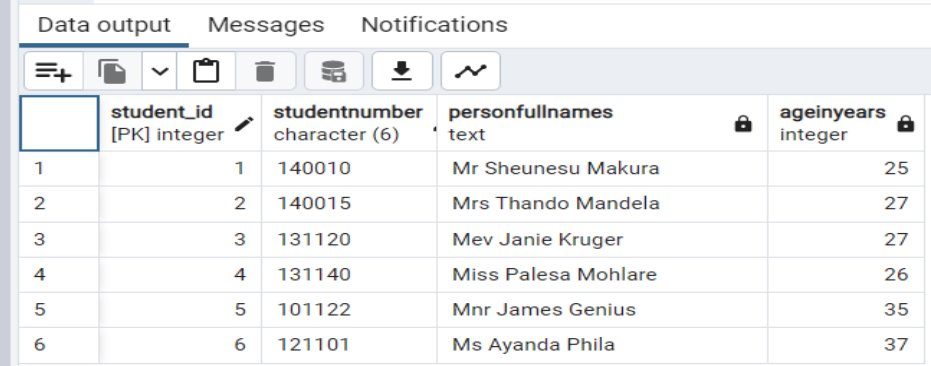
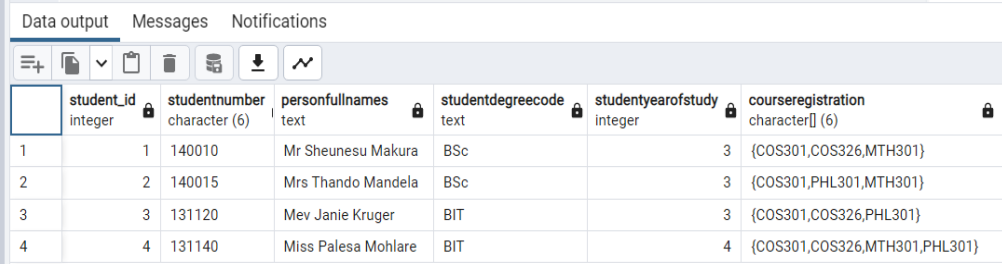
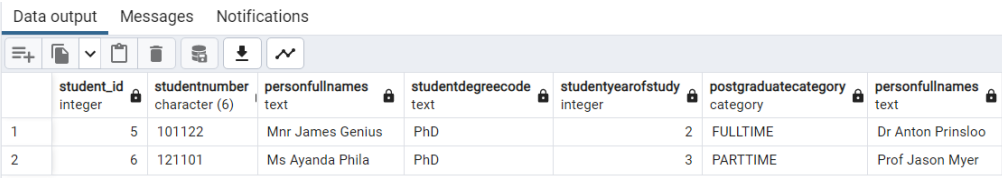
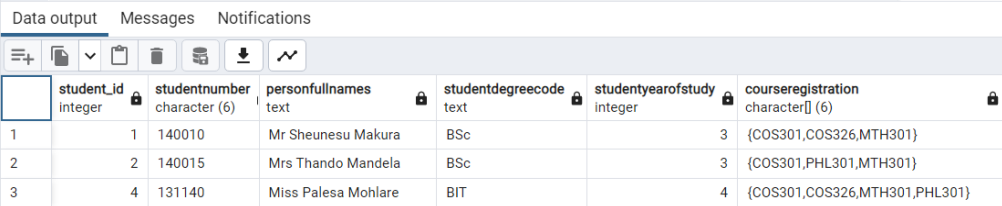
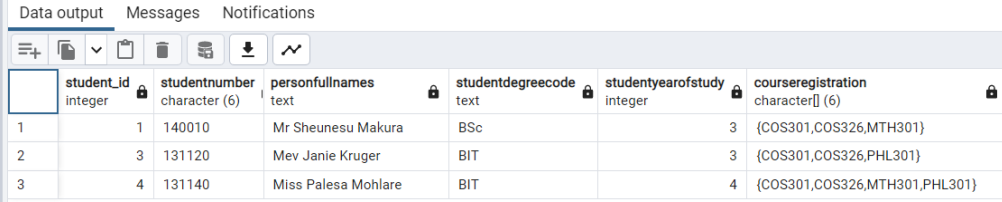
	Attribute values: note that the values of attributes ... key are generated by the SEQUENCES that you created								
Degree Program	degree key	degree code	degree name	number of years	Faculty				
		BSc	Bachelor of Science	3	EBIT				
		BIT	Bachelor of IT	4	EBIT				
		PhD	Philosophiae Doctor	5	EBIT				

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 Reasoning	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 time	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 <i>personFullNames</i> and <i>ageInYears</i> functions)		2
2. undergraduate students registration details (you must use the <i>personFullNames</i> function)		2
3. postgraduate students registration details (you must use the <i>personFullNames</i> function)		2
4. undergraduate students registration details for final year students (you must use the <i>isFinalYear</i> function)		3
5. undergraduate students registration details for students registered for, e.g. COS326 (You must use the <i>isRegisteredFor</i> function)		3

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

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 **CreateStatements.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.

