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

School of Information Technology
Department of Computer Science

COS326 Database Systems

Practical 3 2025

Release Date: 08 August 2025

Submission Date: 17 August 2025 @ 23:59Hrs

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_Surname_Initials***(*XXXXXXXX is your student number) e.g u12345678_Smith_JN*

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]

Scenario

Codealot Pvt Ltd were impressed by your apps you developed using ObjectDB. In the interview, you also had mentioned that you have experience in using PostgreSQL. They would now like to test your proficiency in using PostgreSQL. They give you the following information:

Codealot is a private company with company divisions in 4 provinces in South Africa. They have the Software Development division in Gauteng, the Human Resources division in North West, Finance division in Free State and Auditing division in the Western Cape. They would like to a database created keep records for all its employees. They employ two types of employees namely Full-Time employees and Part time employees. For each employee, the employee number, employee name (title, first name, surname), date of birth, contract code, and the year of hire must be recorded. The part time employees have a mentor to help them when they start working.
For a full-time employee, the province where they work, should additionally be recorded. Full time employees may work in one or more provinces. For the part time employees, the mentor (title, first name, surname) should additionally be recorded. For each contract, the contract name, contract type, number of years of the contract should be recorded. For the province, the province name, provincial code, and the department located in that province should be recorded.

Source: Makura S.M (2025)

The above information is summarised as follows:

Entities, attributes and functions:

Entity	**Attributes (values)**	**functions (and examples of return**
Employee	Employee 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, contract code (e.g. CAL 113 and CAL 114), year hired (e.g. 2015, 2016… etc)	*personFullNames* as 'title first name surname' (e.g. Ms Thando Mandela) (e.g. Mnr Adriaan Graaff) *ageInYears* (e.g. 22 years)

Full Time Employee (is-a employee)	**additional attributes:** provincialRegistration (an array of provincial codes, e.g. [‘GP’, ‘NW’, ‘WC’,….])	*isLocatedAt(…)* returns true if the employee is located at the province with the specified provincial code
Part Time Employee (is-a employee)	**additional attributes:** mentor (title, first name, surname) where title is one of : Ms, Mev, Miss, Mrs, Mr, Mnr, Dr, Prof	*personFullNames* as ‘title first name surname’ (e.g. Prof. Very Serious)
Contract	Contract number, (e.g. CAL 113) Contract Type (e.g. Full time), contract number of years (e.g 5 years or 2 years)	
Province	provincial code (e.g. GP), provincial name (e.g. Gauteng), department (e.g. Auditing)	

Your task as the student is to:

1. Create a UML class diagram to show the above relationships (4 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 User Defined Types (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 plpgsql;
```

(12 marks)

4. **Create a database in PostgreSQL called *employeesDB* 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]

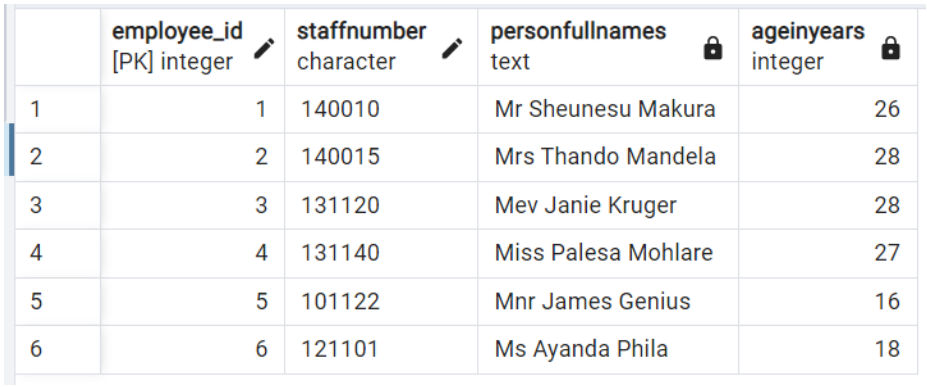
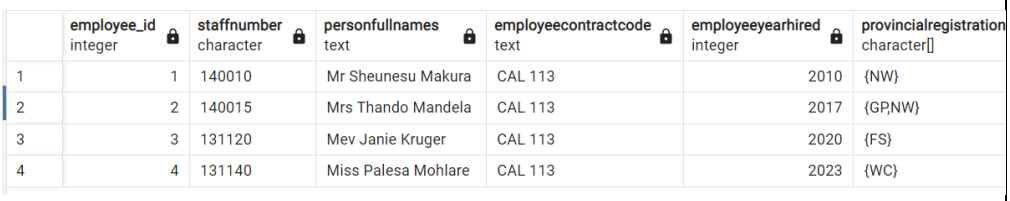
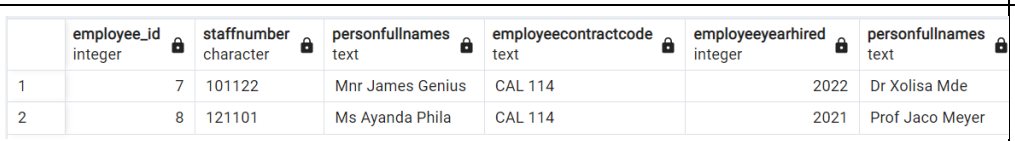
Use the *INSERT INTO* SQL statement to add the following data into the database. Execute some
SELECT statements to confirm that you entered the data correctly.

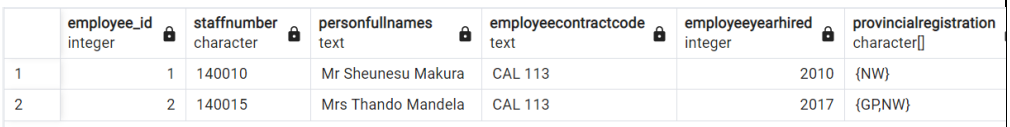
	Attribute values: note that the values of attributes **... key** are generated by the SEQUENCEs that you created							
Contract	**Contract key**	**Contract Code**	**Contract Type**	**Number of years**				
		CAL 113	Full Time	5				
		CAL 114	Part Time	2				
Province	**Province key**	**Provincial code**	**Province name**	**Department**				
		GP	Gauteng	Software Development				
		WC	Western Cape	Auditing				
		FS	Free State	Finance				
		NW	North West	Human Resources				
Full Time	**Employee key**	**Employee number**	**Employee name (title, fname, surname)**	**Date of birth (dd-mm- yyyy)**	**Contract code**	**Year Hired**	**provinceRegistration**	
		140010	choose title & names	10-01-1999	CAL 113	2010	NW	
		140015	choose title & names	25-05-1997	CAL 113	2017	GP, NW	
		131120	choose title & names	30-01-1997	CAL 113	2020	FS	
		131140	choose title & names	20-02-1998	CAL 113	2023	WC	

Part Time	**Employee key**	**Employee number**	**Employee name (title, fname,**	**Date of birth (dd-mm-yyyy)**	**Contract code**	**Year Hired**	**Mentor (title,fname, sname)**
		101122	choose title & names	15-06-2009	CAL 114	2022	choose title & names
		121101	choose title & names	27-04-2007	CAL 114	2021	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. **employee personal details** (you **must** use the *personFullNames* and ageInYears functions)		**2**
2.**Full Time Employee** details including the province and department (you **must** use the *personFullNames* function)		**3**
3. **Part Time Employee** details including the province and department (you **must** use the *personFullNames* function)		**3**

4.**Full Time Employee** contract details (you **must** use the *personFullNames* & *durationofEmployment* functions)		**3**
5.**Employee** details of employee working at the North West province. (you must use the *personFullNames* & *isLocatedAt()* functions)		**3**

	employee_id integer	staffnumber character	personfullnames text	employeecontractcode text	employeeyearhired integer	provincialregistration character[]	durationofemployment integer
1	1	140010	Mr Sheunesu Makura	CAL 113	2010	{NW}	13
2	2	140015	Mrs Thando Mandela	CAL 113	2017	{GP,NW}	6
3	3	131120	Mev Janie Kruger	CAL 113	2020	{FS}	3
4	4	131140	Miss Palesa Mohlare	CAL 113	2023	{WC}	0

	employee_id integer	staffnumber character	personfullnames text	employeecontractcode text	employeeyearhired integer	provincialregistration character[]
1	1	140010	Mr Sheunesu Makura	CAL 113	2010	{NW}
2	2	140015	Mrs Thando Mandela	CAL 113	2017	{GP,NW}

Additional instructions to aid you in the practical

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

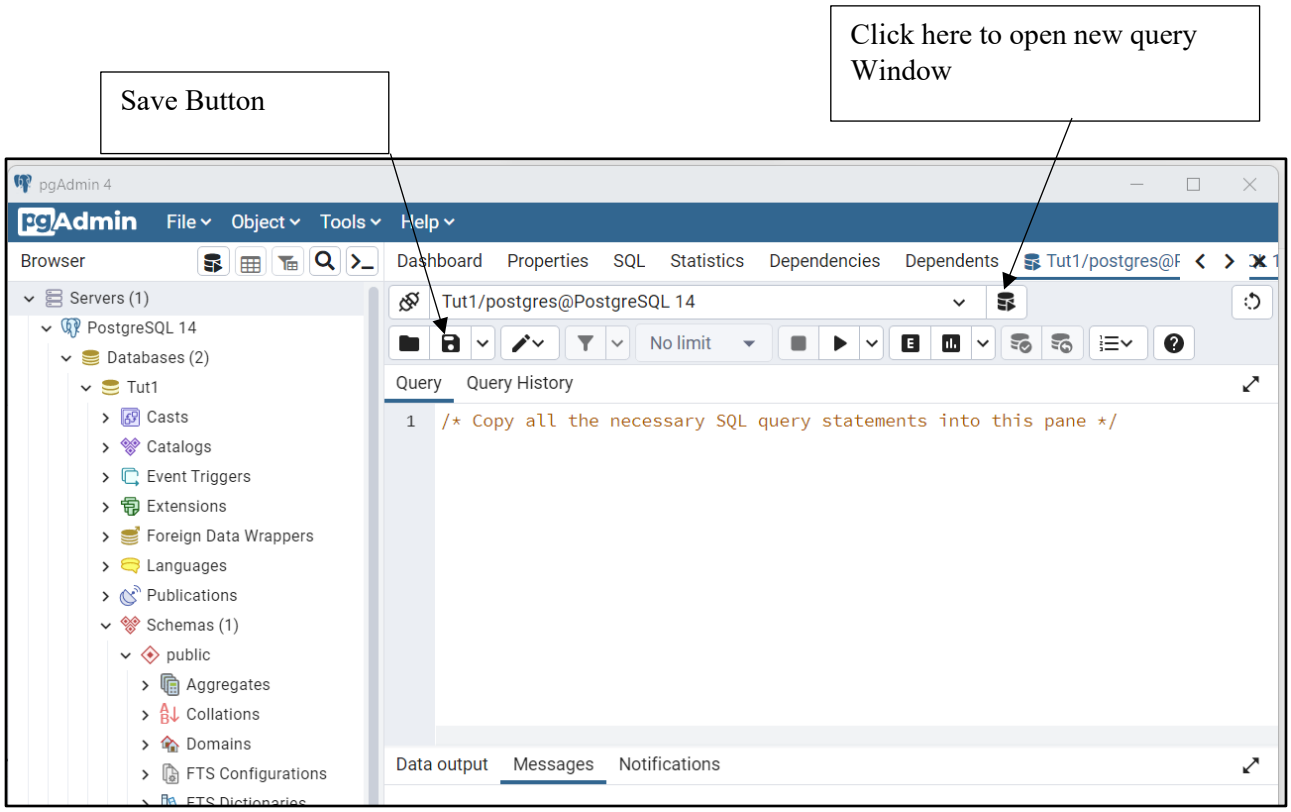
a. Open the **pgAdmin 4** query tool.

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.


Save Button
Click here to open new query Window
pgAdmin 4
pgAdmin
File
Object
Tools
Help
Browser
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Tut1/postgres@PostgreSQL 14
No limit
Query
Query History
1
/* Copy all the necessary SQL query statements into this pane */
Data output
Messages
Notifications
Servers (1)
PostgreSQL 14
Databases (2)
Tut1
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configurations