Course: Database management
Unit: Advanced SQL 2
Material: PostgreSQL PL/pgSQL
Teamwork: Groups of two or three

Complete the PL/pgSQL tutorial
https://www.postgresqltutorial.com/postgresql-plpgsql/

This section shows you step by step on how to use the PL/pgSQL to develop PostgreSQL user-defined functions and stored procedures.

PL/pgSQL procedural language adds many procedural elements, e.g., control structures, loops, and complex computations, to extend standard SQL. It allows you to develop complex functions and stored procedures in PostgreSQL that may not be possible using plain SQL.

PL/pgSQL procedural language is similar to the Oracle PL/SQL. The following are reasons to learn PL/pgSQL:

- PL/pgSQL is easy to learn and simple to use.
- PL/pgSQL comes with PostgreSQL by default. The user-defined functions and stored procedures developed in PL/pgSQL can be used like any built-in functions and stored procedures.
- PL/pgSQL inherits all user-defined types, functions, and operators.
- PL/pgSQL has many features that allow you to develop complex functions and stored procedures.
- PL/pgSQL can be defined to be trusted by the PostgreSQL database server.

Let's get started programming with PL/pgSQL.

Section 1. Getting started

- Introduction to PostgreSQL PL/pgSQL – introduce you to the PostgreSQL PL/pgSQL and explain to you their advantages and disadvantages.
- Dollar-quoted string constants – learn how to use dollar-quoted string constant syntax.
- Block Structure – introduce you to the PL/pgSQL block structure and show you how to develop and execute anonymous blocks.

Section 2. Variables & constants

- Variables – show you how to declare variables in PL/pgSQL.
- Select into – guide you on how to use the select into to select data and assign it to a variable.
- Row type variables – learn how to use the row variables to store a complete row of a result set.

- Record type variables – show you how to declare record variables to hold a single row of a result set.
- Constants – guide you on how to use constants to make the code more readable and easier to maintain.

Section 3. Reporting messages and errors

- Raising errors and reporting messages – show you how to report messages and raise errors in PL/pgSQL.
- Assert – show you how to use the assert statement to add debugging checks to PL/pgSQL code.

Section 4. Control structures

- If statement – introduce you to three forms of the if statement.
- Case statements – explain case statements including the simple and searched case statements.
- Loop statements – show you how to use loop statements to execute a block of code repeatedly based on a condition.
- While loop – learn how to use the while loop statement to create a pre-test loop.
- For loop – show you how to use the for loop statement to iterate over rows of a result set.
- Exit – guide you on how to use the exit statement to terminate a loop.
- Continue – provide you with a way to use the continue statement to skip the current loop iteration and start a new one.

Section 5. User-defined functions

- Create Function – show you how to develop a user-defined function by using the create function statement.
- Function parameter modes – introduce you to various parameter modes including IN, OUT, and INOUT.
- Function overloading – introduce you to the function overloading.
- Functions that return a table – show you how to develop a function that returns a table.
- Drop function – learn how to remove an existing function.

Section 6. Exception handling

- Handling exception – show you how to use the exception clause to catch and handle exceptions.

Section 7. Stored procedures

- Create procedure – show you how to create and call a stored procedure.
- Drop procedure – learn how to drop a stored procedure.

Section 8. Cursors

- Cursors – show you how to use cursors to process a result set, row by row.

Section 9. Trigger functions

- Trigger procedures using PL/pgSQL – apply PL/pgSQL to define trigger procedures.