

SQL Inside PL/SQL: A Beginner's Guide

Welcome back! Today we're diving into the exciting world of PL/SQL, exploring how we can use SQL within our programs. This powerful feature allows us to interact with databases in a seamless and efficient way. Let's get started!

— por Mayko Silva



Introducing SELECT INTO

Declaration

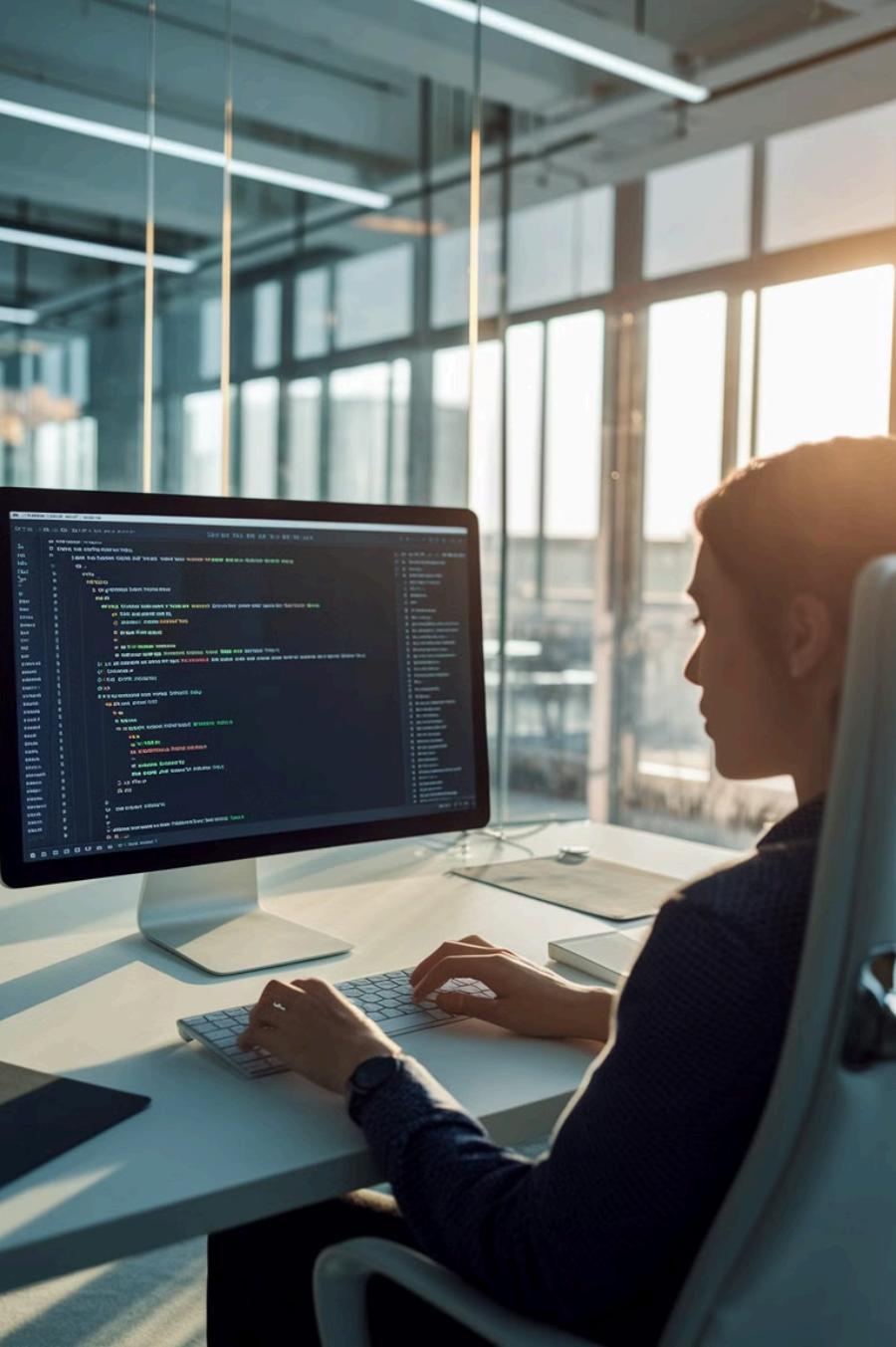
We begin by declaring variables to store our data: `v_emp_name` (`VARCHAR2`) and `v_emp_salary` (`NUMBER`).

Query and Retrieval

Our `SELECT` statement retrieves the first name, last name, and salary for the employee with ID 100.

Data Storage

The `INTO` clause assigns the retrieved values to our declared variables, making them available for use.



Beyond Basics: Cursor FOR Loop

1

Iteration

The FOR loop allows us to process each employee record found by the SELECT statement.

2

Data Retrieval

The SELECT statement retrieves the employee ID and full name for employees in department 30.

3

Output

Inside the loop, we print the ID and full name of each employee, demonstrating the iterative nature of the loop.



Why Use SQL in PL/SQL?

Data Manipulation

PL/SQL extends SQL's capabilities, allowing complex operations like conditional logic, loops, and error handling.

Data Validation

We can validate data retrieved from the database before using it, ensuring data quality and accuracy.

Stored Procedures

PL/SQL enables us to create reusable code blocks called stored procedures, optimizing database performance and reducing code redundancy.

Understanding Cursor Attributes



%ROWTYPE

This attribute allows us to declare variables that match the structure of a database table, making data handling efficient.



Cursor Attributes

Cursors provide information about the current record, such as %FOUND, %NOTFOUND, and %ROWCOUNT, enabling us to handle data effectively.

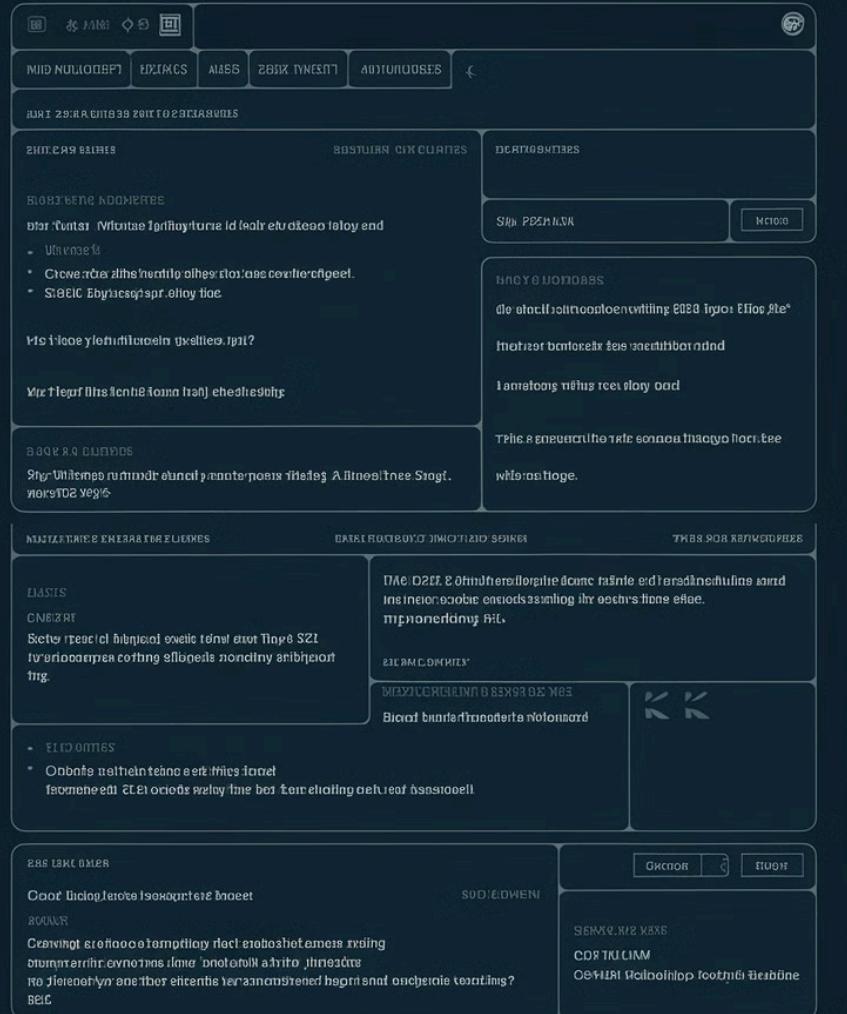


Cursor Loops

FOR loops are ideal for iterating over a set of records, providing a structured approach to processing data retrieved by a cursor.



Working with Multiple Tables



1 Join Operations

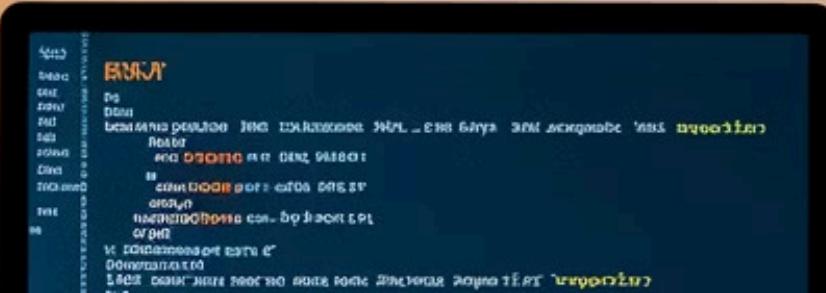
Join statements connect data from different tables based on common columns, enabling us to retrieve related information.

2 Subqueries

Subqueries are queries nested inside other queries, allowing us to filter data based on results from a smaller query.

3 Data Integrity

By combining data from multiple tables, we can ensure data consistency and maintain relationships between different entities.



Handling Data Errors



EXCEPTION Handling

1 PL/SQL's EXCEPTION keyword allows us to gracefully handle runtime errors, preventing program crashes and ensuring proper data handling.



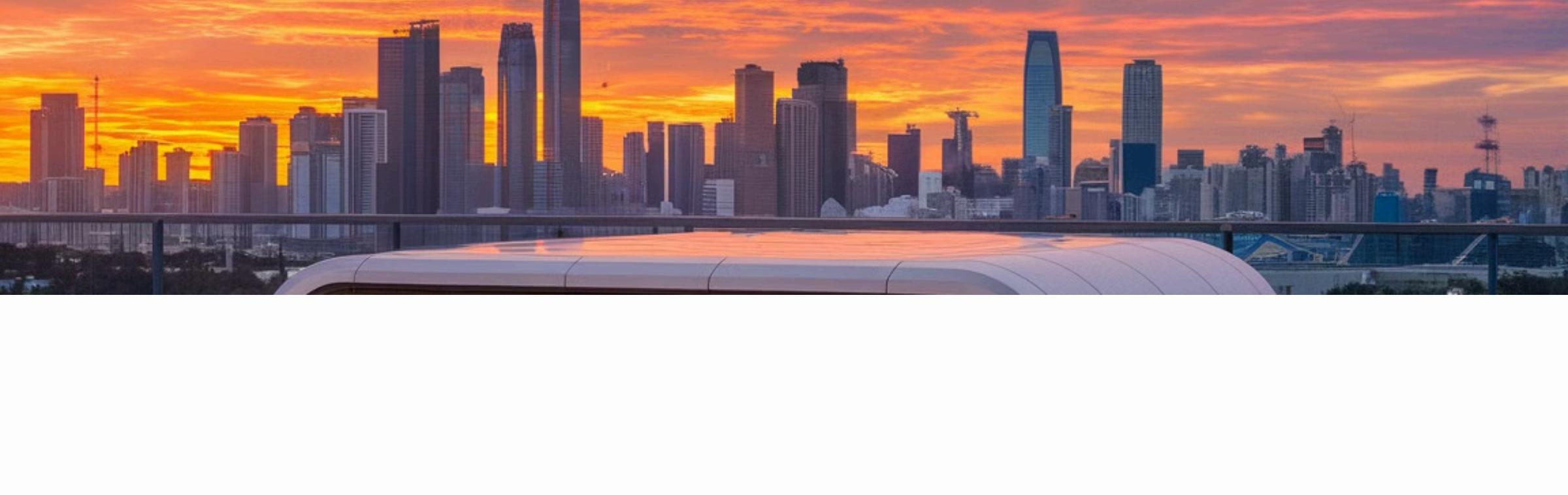
Error Messages

2 When an error occurs, we can retrieve specific error codes and messages, enabling us to diagnose and address the issue.



Data Validation

3 Error handling can be used to validate data before processing, preventing data inconsistencies and ensuring data quality.



PL/SQL: A Powerful Tool

PL/SQL provides a powerful and flexible framework for interacting with databases, offering a wide range of capabilities. By mastering these concepts, you'll be well-equipped to build sophisticated and reliable database applications.