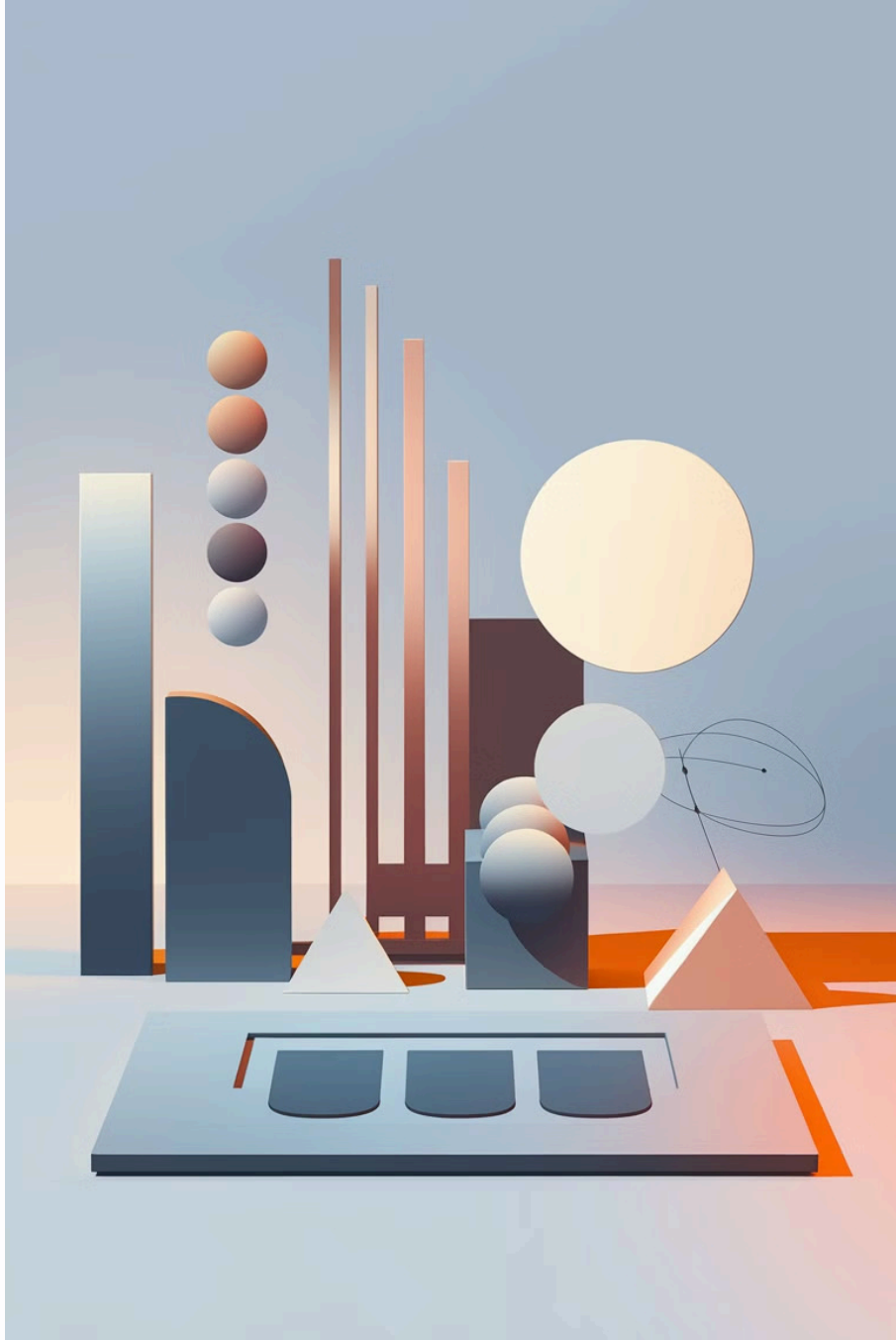
Variables and Data Types in PL/SQL

por Mayko Silva



Common PL/SQL Data Types

VARCHAR2

Stores variable-length character strings such as names, addresses, or any text of varying length.

NUMBER

Holds numeric values. This can store integers and floating-point numbers. Example: you can use to hold salaries, quantities, or any numerical data type.

DATE

Contains date and time information. You can use for birthdays, appointment times, or any date-related data.

BOOLEAN

Serves to represent logical values: possible values TRUE, FALSE, or NULL. You can use for flags, condition checks, or any true-false scenarios.

Declaring Variables in PL/SQL

Example Declarations

- v_name VARCHAR2(50)
- v_salary NUMBER
- v_hire_date DATE
- v_is_active BOOLEAN

Initialization

Some variables are being initialized:

- name initialized with 'Alice'
- hire date initialized with SYSDATE
- is_active initialized with TRUE
- v_salary is not initialized

Structure of PL/SQL Code

1

DECLARE Section

This is where you define your variables

2

BEGIN Block

This is where you can use the variables you've declared

3

Operations

Perform operations like assign values or use them in conditionals in queries

Key Points to Remember

Declaration

Variables must be declared before they can be used

Data Type

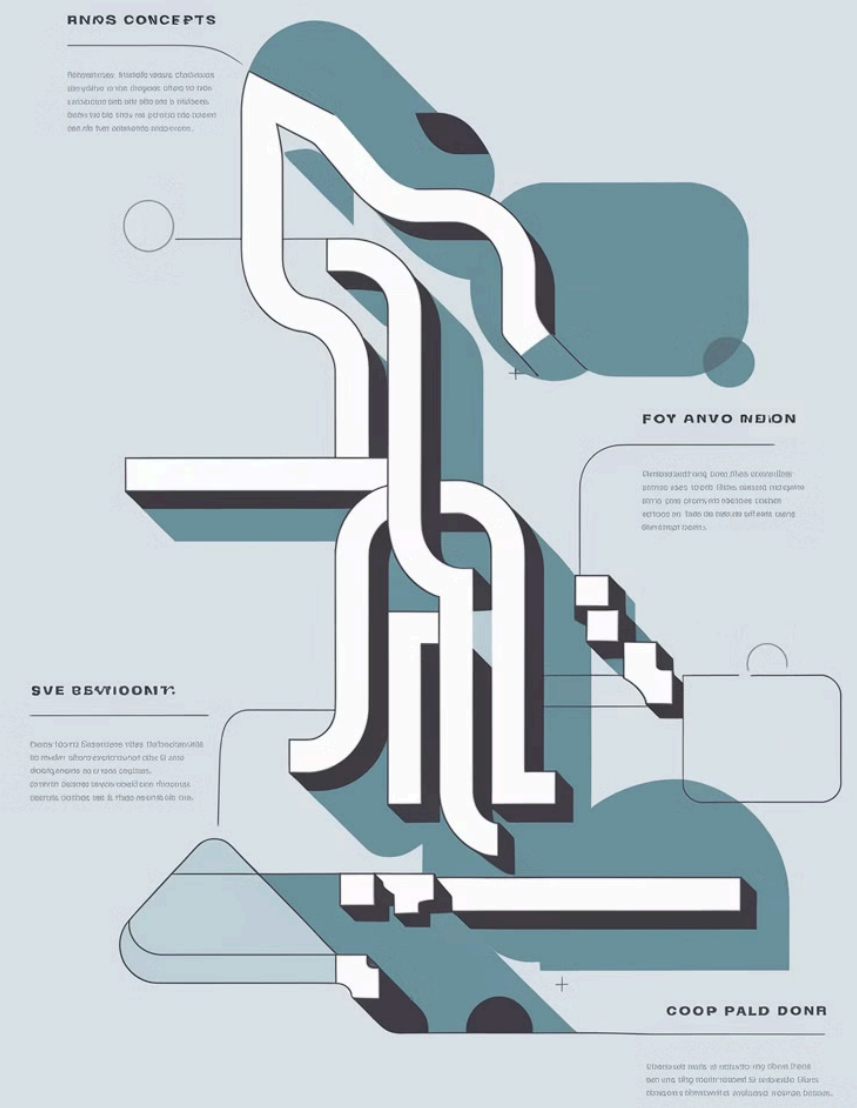
The data type of a variable determines what kind of data it can hold

Initialization

Variables can be initialized at declaration, but this is optional

Naming

It's important to use meaningful variable names to improve code readability



Variable Declaration Process

1

Step 1: Choose Data Type

Determine the appropriate data type for your variable based on the kind of data it will store.

2

Step 2: Name Variable

Choose a meaningful name that reflects the variable's purpose.

3

Step 3: Declare Variable

Use the DECLARE section to formally declare the variable with its chosen name and data type.

4

Step 4: Initialize (Optional)

Optionally, you can initialize the variable with a starting value.

Using Variables in PL/SQL

Assignment

Assign values to variables using the := operator

Calculations

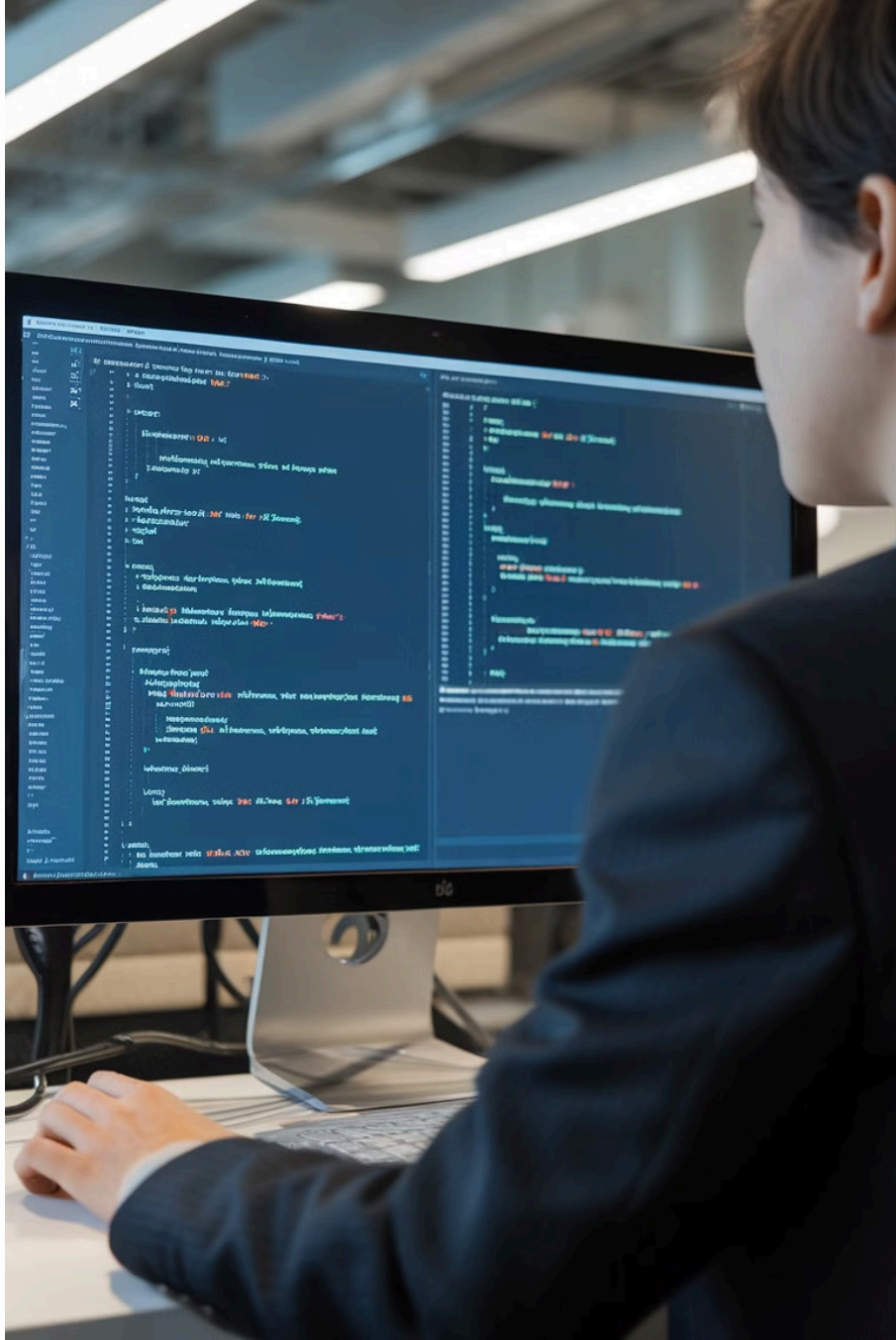
Use variables in mathematical operations

Conditionals

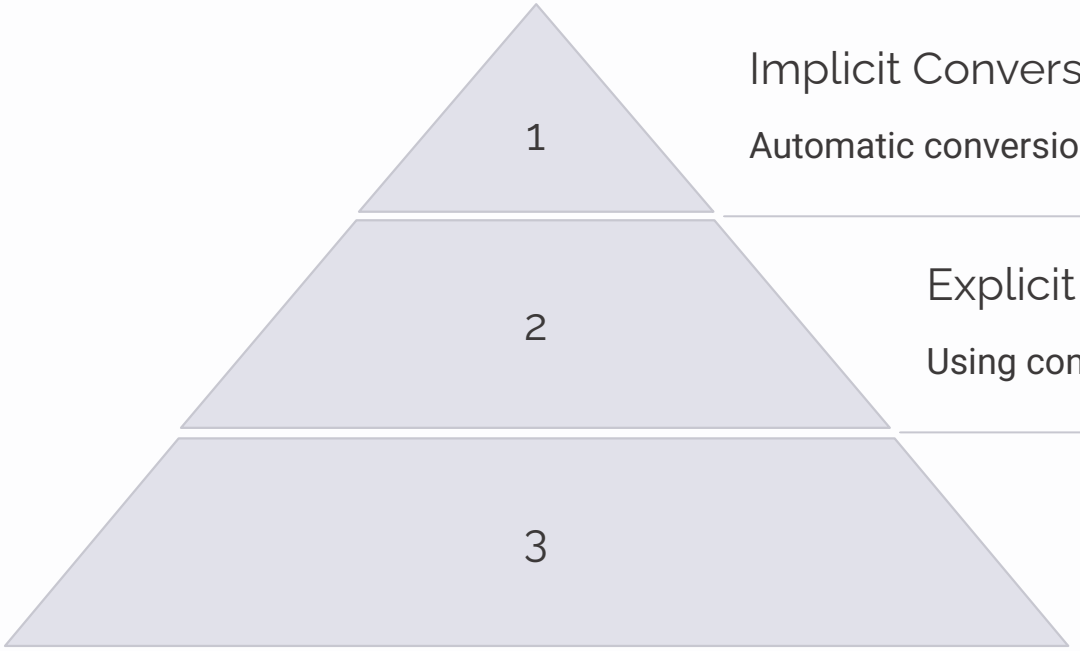
Use variables in IF statements and CASE expressions

Queries

Include variables in SQL queries within PL/SQL blocks



Data Type Conversion



Implicit Conversion

Automatic conversion by PL/SQL

Explicit Conversion

Using conversion functions

Type Compatibility

Ensuring data types match

Understanding data type conversion is crucial when working with variables of different types in PL/SQL. It helps prevent errors and ensures smooth data manipulation.

Best Practices for Variables

1

Use Descriptive Names

Improves code readability

2

Initialize When Possible

Prevents unintended null values

3

Use Appropriate Data Types

Ensures data integrity

4

Scope Management

Declare variables in the appropriate block

Conclusion

4

Main Data Types

VARCHAR2, NUMBER, DATE, and BOOLEAN are the most commonly used data types in PL/SQL.

2

Code Sections

PL/SQL code typically consists of a DECLARE section for variable declarations and a BEGIN block for using those variables.

5

Key Points

Remember the five key points: declaration before use, data type determines content, optional initialization, DECLARE section for definitions, and use meaningful variable names.

Understanding variables and data types is crucial for PL/SQL programming. They form the building blocks for more complex operations and data manipulations in your PL/SQL code. See you in the next lesson!