**19.**

**1) Sequential Control in PL/SQL**

**Sequential control refers to the default execution flow in PL/SQL, where statements are executed one after the other in the order in which they are written. In addition to the regular flow, PL/SQL allows branching (e.g., using IF statements) and looping (e.g., using FOR or WHILE loops) to alter the order of execution.**

**GOTO Statement in PL/SQL**

**The GOTO statement allows the program to jump to a specified label in the code. It's used to transfer control to another part of the program. While GOTO is rarely used in modern programming due to its potential for making code harder to read, it is sometimes used for control flow in PL/SQL.**

**Syntax:**

**sql**

**Copy code**

**GOTO label\_name;**

**2)** **DECLARE**

**i NUMBER := 1; -- Variable to store the current number**

**BEGIN**

**-- Loop through numbers from 1 to 10**

**LOOP**

**IF i = 5 THEN**

**-- Skip number 5 using GOTO**

**i := i + 1; -- Increment the number**

**GOTO skip\_number; -- Jump to the label 'skip\_number'**

**END IF;**

**-- Print the number**

**DBMS\_OUTPUT.PUT\_LINE(i);**

**-- Increment the number**

**i := i + 1;**

**-- Exit the loop when i exceeds 10**

**IF i > 10 THEN**

**EXIT;**

**END IF;**

**END LOOP;**

**-- Label to skip printing the number 5**

**skip\_number:**

**NULL; -- No operation, just a placeholder for the GOTO statement**

**END;**