**20.**

**1)1. Implicit Cursor:**

**An Implicit Cursor is automatically created by Oracle when a SQL query is executed in PL/SQL. It is used for SELECT INTO, INSERT, UPDATE, and DELETE operations, and you do not need to explicitly declare or manage them. Oracle handles the opening, fetching, and closing of the cursor automatically for these types of SQL statements.**

* **Advantages:**
  + **Simpler to use as it requires no explicit declaration or management.**
  + **Oracle automatically handles the cursor for you, making it easier for simple queries.**
* **Disadvantages:**
  + **Provides limited control over fetching data.**
  + **Cannot fetch multiple rows in complex scenarios, such as when more than one row is returned.**

**2. Explicit Cursor:**

**An Explicit Cursor is explicitly declared by the programmer in PL/SQL when you need more control over data retrieval. It is used when you want to fetch multiple rows, process them one by one, and manage the cursor's lifecycle (open, fetch, close).**

* **Advantages:**
  + **Provides more control over the SQL query execution.**
  + **Allows you to process multiple rows.**
  + **You can fetch rows iteratively and handle complex queries with ease.**
* **Disadvantages:**
  + **Requires more code for declaration, fetching, and closing.**
  + **Can be more complex to manage.**

**2)** **DECLARE**

**-- Declare a cursor to fetch items with price greater than 10000**

**CURSOR item\_cursor IS**

**SELECT item\_id, item\_name, price**

**FROM store**

**WHERE price > 10000;**

**-- Declare a variable to store the count of items**

**item\_count NUMBER := 0;**

**-- Declare a record variable to hold the data from the cursor**

**item\_record item\_cursor%ROWTYPE;**

**BEGIN**

**-- Open the cursor**

**OPEN item\_cursor;**

**-- Loop through the cursor and count the number of items**

**LOOP**

**-- Fetch the data into item\_record**

**FETCH item\_cursor INTO item\_record;**

**-- Exit the loop when no more rows are fetched**

**EXIT WHEN item\_cursor%NOTFOUND;**

**-- Increment the item count**

**item\_count := item\_count + 1;**

**END LOOP;**

**-- Close the cursor**

**CLOSE item\_cursor;**

**-- Display the count of items with price greater than 10000**

**DBMS\_OUTPUT.PUT\_LINE('Number of items with price greater than 10000: ' || item\_count);**

**END;**