

# Week 7 – LAQ's

## Differences Between Implicit and Explicit Cursors

Implicit and explicit cursors are both mechanisms used in SQL databases to manage the context area for SQL statements. Here's a detailed comparison of the two:

| Feature      | Implicit Cursors  | Explicit Cursors   |
|--------------|---|--|
| Creation     | Automatically created by the database system when an SQL statement is executed. | Defined explicitly by the programmer using a <code>DECLARE</code> statement.       |
| Row Fetching | Can only fetch a single row at a time.  | Can fetch multiple rows, allowing more complex data retrieval.                     |
| Memory Usage | Uses anonymous buffer memory for storage.                                       | Uses user-defined memory space for storage.  |
| Closure      | Automatically closed after execution of the statement.                          | Must be explicitly closed by the programmer after use.                             |
| Efficiency   | Generally more efficient for simple operations due to less overhead.            | More control but potentially less efficient due to additional coding requirements. |

| Feature                     | Implicit Cursors  | Explicit Cursors  |
|-----------------------------|---|---|
| <b>Error Handling</b>       | More vulnerable to errors, such as data errors, since they handle exceptions automatically. | Less vulnerable to errors; programmers can handle exceptions explicitly.                                      |
| <b>Programmatic Control</b> | Provides less control over cursor behavior; not programmable by the user.                   | Provides more control over cursor operations, allowing programmers to manage fetching and closing explicitly. |

### Detailed Explanation

#### 1. Creation:

- **Implicit Cursors:** These are automatically created by the database whenever a SQL statement is executed that does not use an explicit cursor. For example, executing an INSERT, UPDATE, or DELETE statement creates an implicit cursor.
- **Explicit Cursors:** Programmers define these cursors using a DECLARE statement, allowing them to specify the SQL query and manage its lifecycle.

#### 2. Row Fetching:

- **Implicit Cursors:** Typically used for single-row queries where only one record is expected.
- **Explicit Cursors:** Designed for operations that may return multiple rows, allowing programmers to fetch each row individually.

#### 3. Memory Usage:

- **Implicit Cursors:** Use a temporary, anonymous memory area managed by the database.
- **Explicit Cursors:** Allocate user-defined memory space, which can be managed more effectively by the programmer.

#### 4. Closure:

- Implicit Cursors: Automatically close after the execution of the SQL statement, which simplifies coding but offers less control.
- Explicit Cursors: Require explicit closing by the programmer, which can help prevent memory leaks and manage resources better.

#### 5. Efficiency:

- Implicit Cursors: Generally faster for simple operations because they involve less overhead and are optimized for single-row processing.
- Explicit Cursors: While they provide greater flexibility and control, they may introduce additional overhead due to their complexity.

#### 6. Error Handling:

- Implicit Cursors: Handle errors automatically but can lead to issues if not properly managed, especially in complex transactions.
- Explicit Cursors: Allow for custom error handling logic, making it easier to manage exceptions and unexpected results.

#### 7. Programmatic Control:

- Implicit Cursors: Offer limited control; the database manages their lifecycle.
- Explicit Cursors: Provide full control over how and when data is fetched and processed, allowing for more complex logic in applications.

#### Conclusion

In summary, implicit cursors are simpler and more efficient for straightforward operations, while explicit cursors provide greater flexibility and control for handling complex queries that require multiple rows or specific error handling strategies. The choice between using implicit or explicit cursors depends on the specific requirements of the application and the complexity of the SQL operations being performed.