**Checking For Null Values And Handling Null Values**

**Project Name: statistical machine learning Team ID : PNT2022TMID04163**

**approaches to liver disease prediction**

**Checking For Null Values And Handling Null Values:**

## NULL Values:

It is not possible to test for NULL values with comparison operators, such as =, <, or <>.

We will have to use the IS NULL and IS NOT NULL operators instead.

### NULL Syntax

SELECT column\_namesFROM table\_name  
WHERE column\_name IS NULL;

### NOT NULL Syntax

SELECT column\_namesFROM table\_name  
WHERE column\_name IS NOT NULL;

## NULL Operator

The IS NULL operator is used to test for empty values (NULL values).

The following SQL lists all customers with a NULL value in the "Address" field:

### Example

SELECT CustomerName, ContactName, Address  
FROM Customers  
WHERE Address IS NULL;

# Handling Null Values:

A null value in a relational database is used when the value in a column is unknown or missing. A null is neither an empty string (for character or datetime data types) nor a zero value (for numeric data types).

## Nulls and Three-Valued Logic

Allowing null values in column definitions introduces three-valued logic into your application. A comparison can evaluate to one of three conditions:

* True
* False
* Unknown

**Example:**

* static private void WorkWithSqlNulls()
* {
* DataTable table = new DataTable();
* // Specify the SqlType for each column.
* DataColumn idColumn =
* table.Columns.Add("ID", typeof(SqlInt32));
* DataColumn descColumn =
* table.Columns.Add("Description", typeof(SqlString));
* // Add some data.
* DataRow nRow = table.NewRow();
* nRow["ID"] = 123;
* nRow["Description"] = "Side Mirror";
* table.Rows.Add(nRow);
* // Add null values.
* nRow = table.NewRow();
* nRow["ID"] = SqlInt32.Null;
* nRow["Description"] = SqlString.Null;
* table.Rows.Add(nRow);
* // Initialize variables to use when
* // extracting the data.
* SqlBoolean isColumnNull = false;
* SqlInt32 idValue = SqlInt32.Zero;
* SqlString descriptionValue = SqlString.Null;
* // Iterate through the DataTable and display the values.
* foreach (DataRow row in table.Rows)
* {
* // Assign values to variables. Note that you
* // do not have to test for null values.
* idValue = (SqlInt32)row["ID"];
* descriptionValue = (SqlString)row["Description"];
* // Test for null value in ID column.
* isColumnNull = idValue.IsNull;
* // Display variable values in console window.
* Console.Write("isColumnNull={0}, ID={1}, Description={2}",
* isColumnNull, idValue, descriptionValue);
* Console.WriteLine();
* }

**Output:**

* isColumnNull=False, ID=123, Description=Side Mirror
* isColumnNull=True, ID=Null, Description=Null