---

1. In an SSIS Package Execute SQL Task that returns one value from a SQL query what ResultSet type must be selected?

Answer:
Use the **"Single Row"** ResultSet type.

- It is specifically used when the SQL query returns **only one row** with one or more columns.
- The values are mapped to variables via the **Result Set** tab in the task editor.

---

2. What is the purpose of Parameter Mapping in Execute SQL Task and how does the indexing of question marks (?) relate to it?

Answer:
Parameter Mapping allows you to pass SSIS variable values into your SQL query when using **parameterized queries**.

- If you're using **ODBC or OLE DB** connection types, question marks ? are used as **placeholders** in the SQL query (called **positional parameters**).
- The order of the ? placeholders **matters**: the **first ? corresponds to Parameter Name 0**, the second to 1, and so on in the Parameter Mapping tab.
- For example:
- SELECT * FROM Students WHERE Name = ? AND Age = ?
  - You'd need to map variables in order:
    Parameter Name 0 → @StudentName
    Parameter Name 1 → @Age

---

3. Describe how you can view the value stored in an SSIS variable at runtime using Breakpoints.

Answer:

1. **Open the Control Flow designer**.
2. Right-click on the task (e.g., Script Task or Data Flow Task) where you want to debug and choose **Edit Breakpoints**.
3. Enable a breakpoint such as "Break at the beginning of the task".

4. Run the package in **Debug mode**.
5. When execution pauses, go to **Debug → Windows → Locals** or **Watch** window.
6. Expand the **Variables** node to view the current values of your SSIS variables at that point in execution.

---

4. Explain how to configure a Script Task to print a string variable named @StudentName. Provide the C# line used for display.

Answer:

1. Open the **Script Task Editor**.
2. In **ReadOnlyVariables**, add: User::StudentName
3. Click **Edit Script** and in the Main() method, write:

```
string studentName = Dts.Variables["User::StudentName"].Value.ToString();

System.Windows.Forms.MessageBox.Show(studentName);
```

- This will pop up a message box showing the value of @StudentName when the script task runs.

---

5. How do you use two separate SQL queries in SSIS where the first query stores its result in a variable, and the second one uses that variable as a parameter?

Answer:

1. Use **Execute SQL Task 1**:
    - Query returns a value (e.g., SELECT MAX(ID) FROM Students)
    - Set ResultSet to **Single Row**
    - Map result to a variable, e.g., User::MaxStudentID
2. Use **Execute SQL Task 2**:
    - Query like SELECT * FROM Enrollments WHERE StudentID = ?
    - Set parameter mapping:
        - Parameter Name: 0
        - Variable Name: User::MaxStudentID

  - This uses the output from Task 1 as input to Task 2.

---

6. In an Execute SQL Task with multiple parameters (?), how do you match variables to specific positions?

Answer:
Use the **Parameter Mapping** tab:

- Parameters are **indexed starting at 0**.
- Match SSIS variables to their respective ? placeholders by their **order of appearance** in the SQL.

Example:

SELECT * FROM Table WHERE Col1 = ? AND Col2 = ?

Parameter Name	Variable Name
0	User::Var1
1	User::Var2

Make sure to use the correct **Data Type** in the parameter mapping as well (e.g., VARCHAR, INT, etc.).

---

7. What is the role of DelayValidation when working with dynamic file paths?

Answer:
DelayValidation = True tells SSIS to **postpone validation** of the task/component until runtime.

- This is essential when a file path (or connection string) is set dynamically through an expression or variable, and might **not exist at design-time**.
- Prevents errors like “File does not exist” during **package validation** before execution.

---

8. What happens if a result set is set to "Full Result Set" but the SQL query only returns a single value?

Answer:

- SSIS will still expect a **recordset (table-like structure)**, even if it only has one row and one column.

- You must store the result into an **Object-type variable**, and use a **Foreach Loop Container** or **Script Task** to extract the data.
- If you try to treat it like a single value, you'll get **type mismatch** or runtime errors.

To simplify:

- If returning one value → use **Single Row**
- If returning a list/table → use **Full Result Set**
