Error Handling in ETL

**Scenario 1 🡪 Fail Component**

In SSMS --

Use Zensar\_DB

go

create table table1 (a int, b int) -- Source Table

create table table2 (a int, b int **check (b >= 10))** -- Destination or Target Table

Insert Into Table1 Values(1,**11**)

Insert Into Table1 Values(2,**12**)

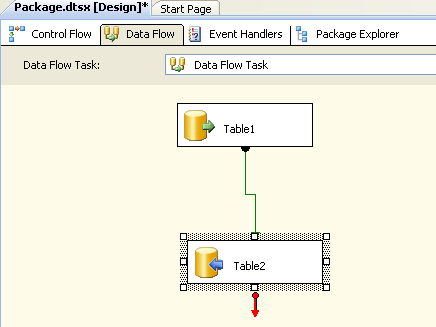
**Insert Into Table1 Values(3,7)**

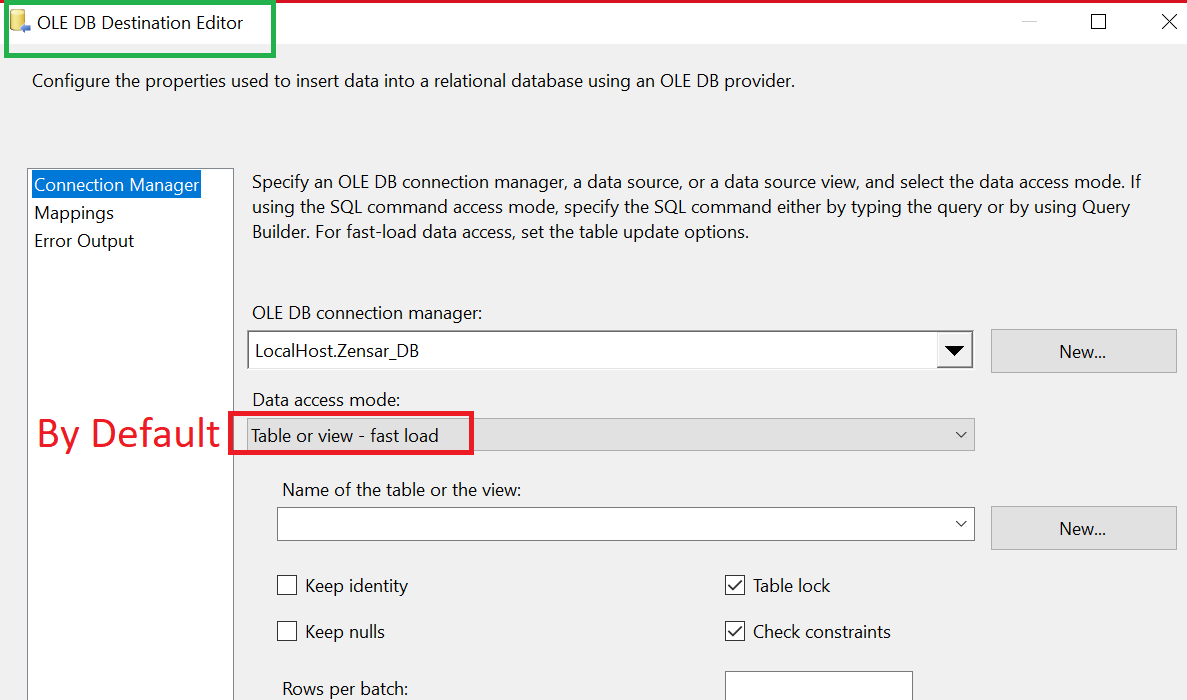
Insert Into Table1 Values(4,13)

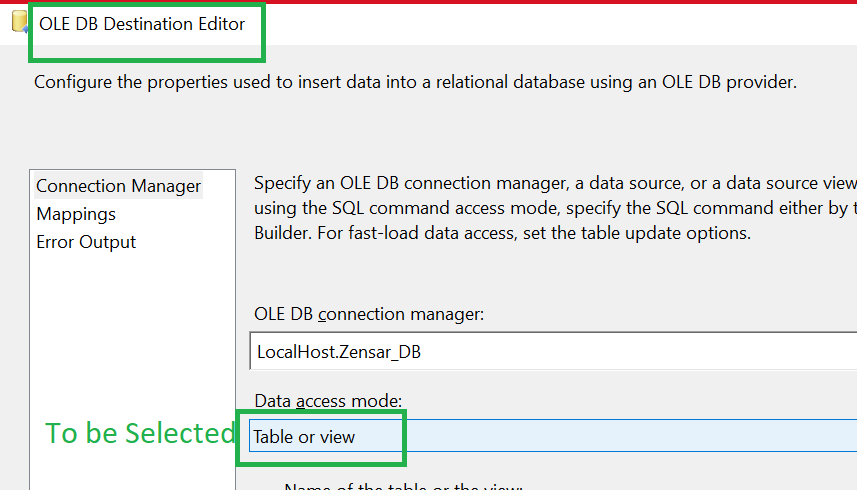
go

select \* from table1

In SSIS –





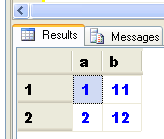


Run the package.

Error will come for Table 2.

In SSMS –

select \* from table2

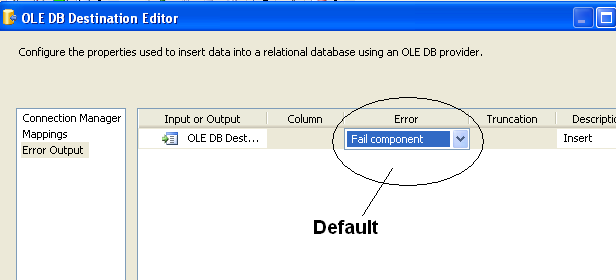


In the default setup the rows till the error come get transferred.

After the error record, even if the next record is correct, still it is not getting transferred.

The control is coming out.

This is because in the Error Output tab of OLE DB Destination (Table 2) the default value is **Fail Component** as shown next –

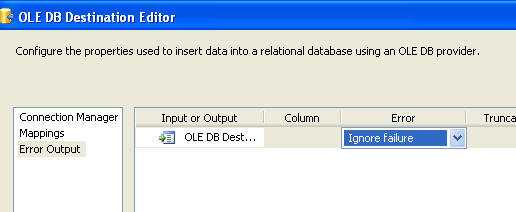


**Scenario 2 🡪Ignore Failure**

**In SSMS Delete all the rows of Table 2**

**Delete from table2**

In SSIS change the value from Fail Component to **Ignore Failure** as shown next –



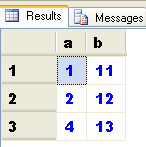
Run the package.

There is no error.

In SSMS –

See the Table2 records

Select \* from table2



Due to **Ignore Failure** even if a error comes it will ignore that error and continue processing next rows.

**Scenario 3 🡪 Redirect Row**

Row which have error can be transferred to another table.

In SSMS –

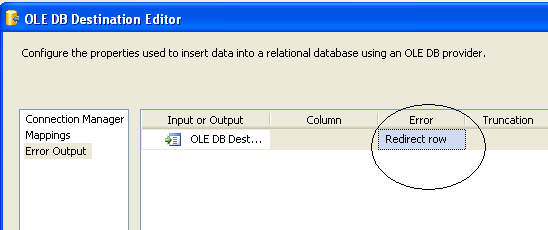
Delete from Table2

Select \* into **Bad\_Rows** from Table2

This new table Bad\_Rows will have the rows which did not get transferred to table2.

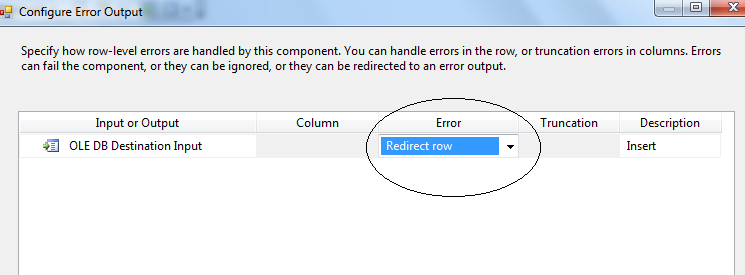
In SSIS –

Change the value to Redirect row as shown next –



**Add one more OLE DB Destination**. Rename it as Table 3 (Error Rows). Connect the red arrow of Table 2 to Table 3 (Error Rows).

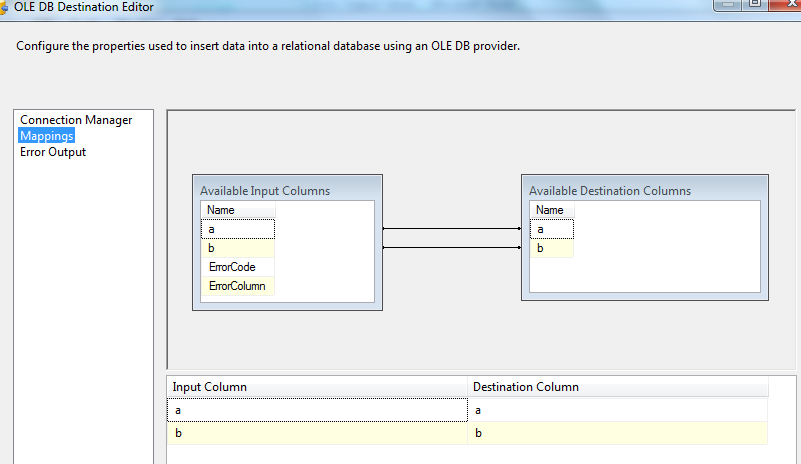
In the Configure Error Output window Redirect Row will be the option by default selected as shown next –



Open the OLE DB Destination Editor of the Table 3 (Error Rows).

Select Bad\_Rows.

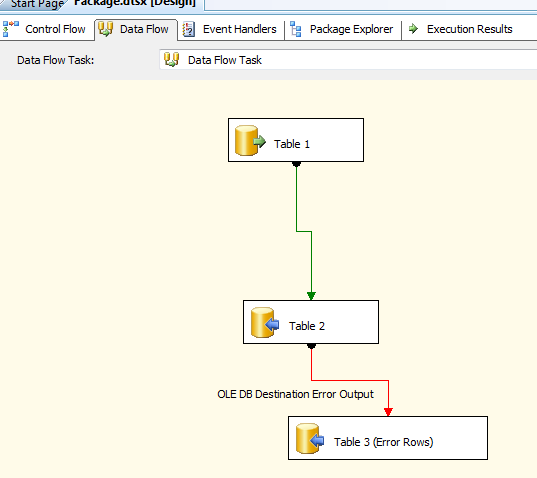
In Mappings keep the default a & B columns mapped as shown next –



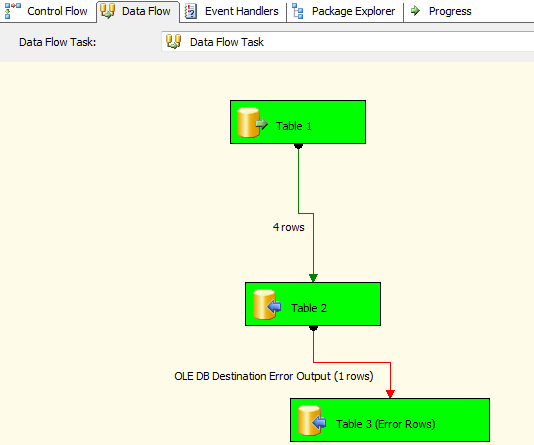
*(Note 🡪 ErrorCode and ErrorColumn are the internal lineage values. They are NOT important from business perspective. So ignore them.*

*The lineage identifiers are assigned automatically when you first create the package.)*

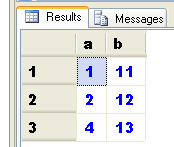
Finally the Data Flow will look like this –



Run the package –



Select \* from table2



Select \* from Table3

