

microgen

DBCClarity Developer™



Product Overview and Examples

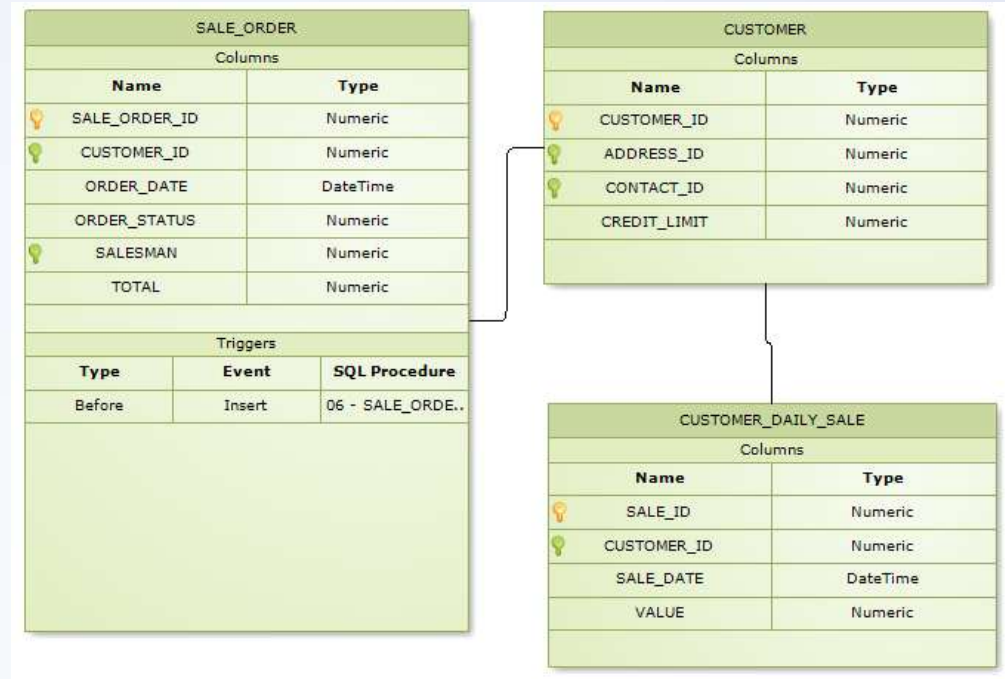
Advanced SQL Procedure Example

Advanced SQL Procedures

- Example

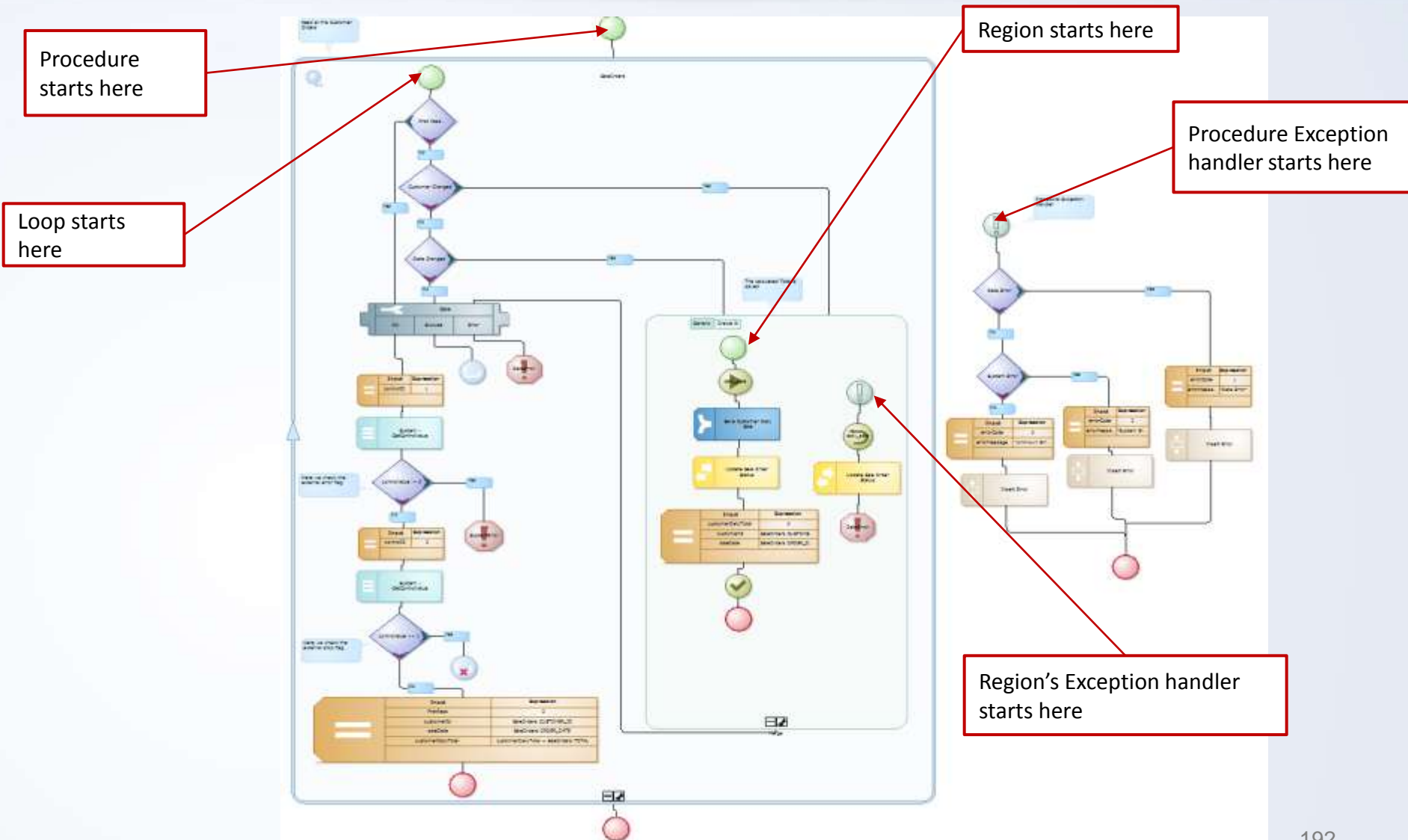
Example Scenario

- Calculate the daily sale for each customer.
- This is a recalculation, the entries may already exist and then only the values should be updated.
- The sale order can be marked as “excluded” and the record must be skipped.
- The sale order can be marked as “error” and the recalculation must stop.
- After each updated or inserted daily sale, the status of the included sale orders must change to “processed”
- The calculation is a time consuming operation and it has to be interrupted on demand or if any external system error occurs.



$$\text{CUSTOMER_DAILY_SALE}(\text{CUSTOMER_ID}, \text{SALE_DATE}) = \text{SUM}(\text{SALE_ORDER}(\text{CUSTOMER_ID}, \text{ORDER_DATE}))$$

SQL Procedures – The Diagram



Loop Block based on a SQL Rule

To calculate the daily sale from the Customer Orders table, use a Loop over the result-set of a query returning all the customer orders sorted by the customer and date.

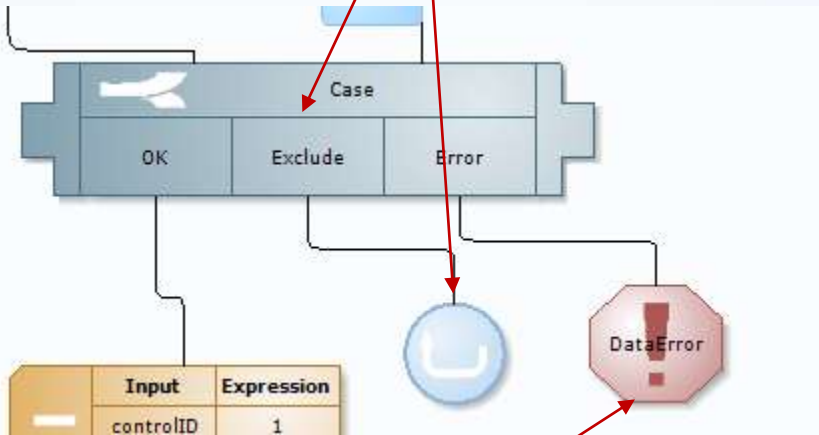


Attribute	Order
SALE_ORDER.CUSTOMER_ID	ASC
SALE_ORDER.ORDER_DATE	ASC

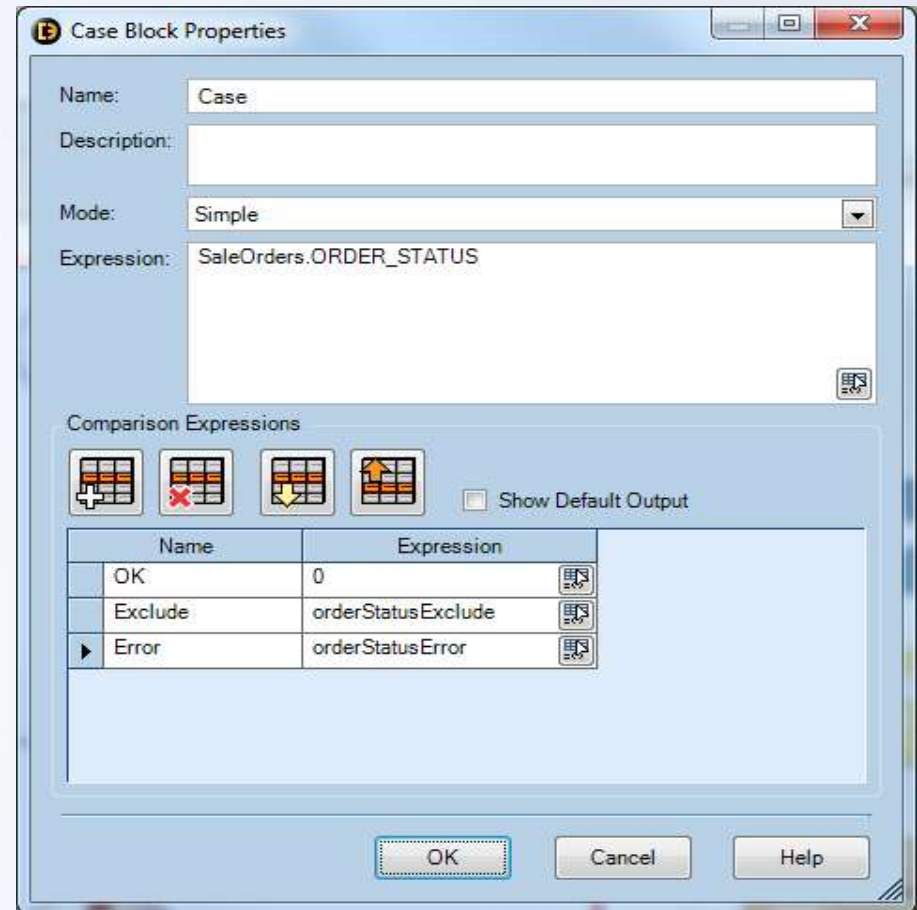
Name	Generic Expression
CUSTOMER_ID	SALE_ORDER.CUSTOMER_ID
ORDER_DATE	SALE_ORDER.ORDER_DATE
TOTAL	SALE_ORDER.TOTAL
ORDER_STATUS	SALE_ORDER.ORDER_STATUS
SALE_ORDER_ID	SALE_ORDER.SALE_ORDER_ID

Case, Continue Loop, Throw Exception microgen DBClarity Developer™

If the record is marked as “excluded”, continue the loop and do not go to the calculation



If the record is marked as “error”, raise the DataError Exception



SELECT Statement and Break Loop

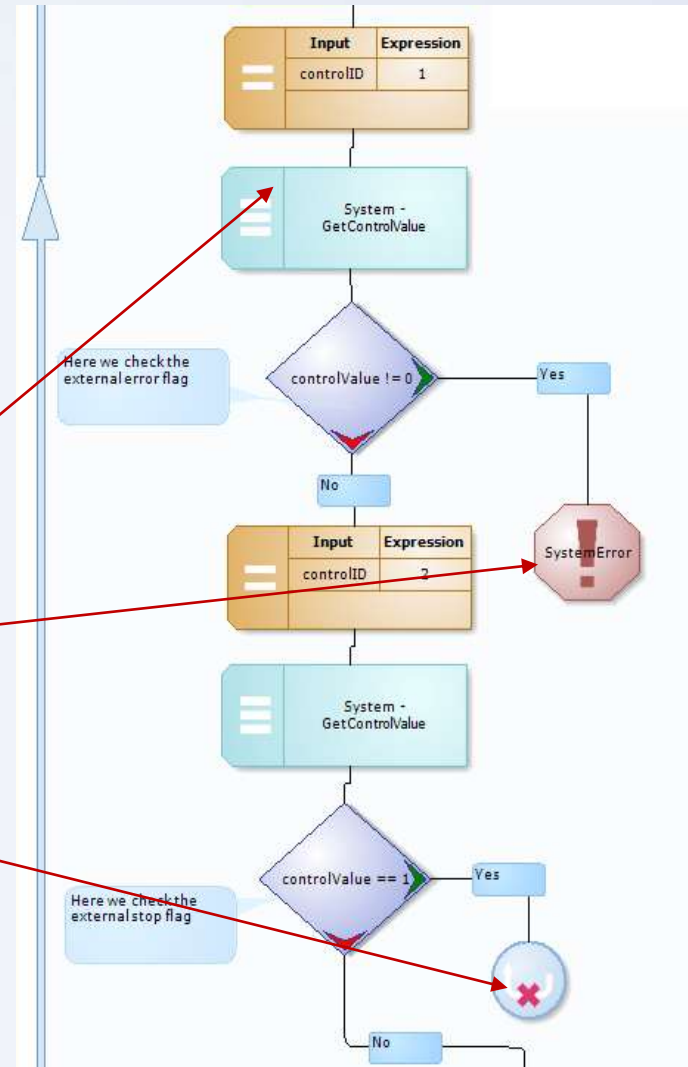
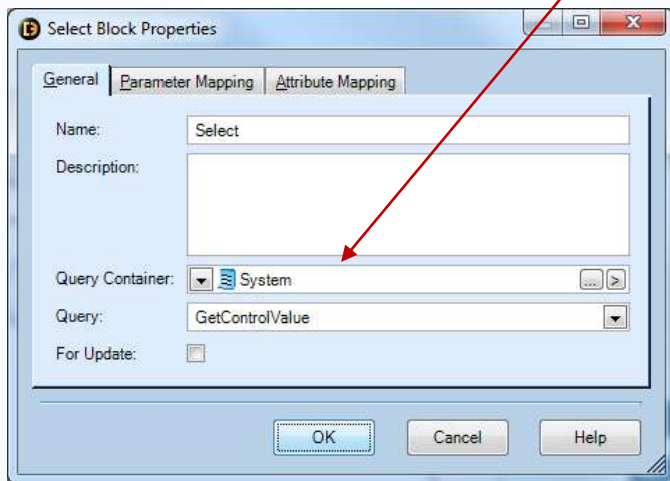
- To interact with the system, read special control data from a system table, using the SELECT Block.
- It produces a SELECT statement and stores the result in a variable:

```
SELECT  
CONTROL_VALUE as CONTROL_VALUE INTO controlValue  
FROM  
SYSTEM_CONTROL  
WHERE  
CONTROL_ID = controlID;
```

Query stored in EDF, but it can be a SQL Rule

Raise the SystemError exception

Break the Loop and exit the procedure



SQL Procedure Elements

Calculation done in a Region Block to catch the exception thrown only by the operations inside

Begin a transaction

Call the Merge SQL Rule

Update the Sale Order status to "processed"

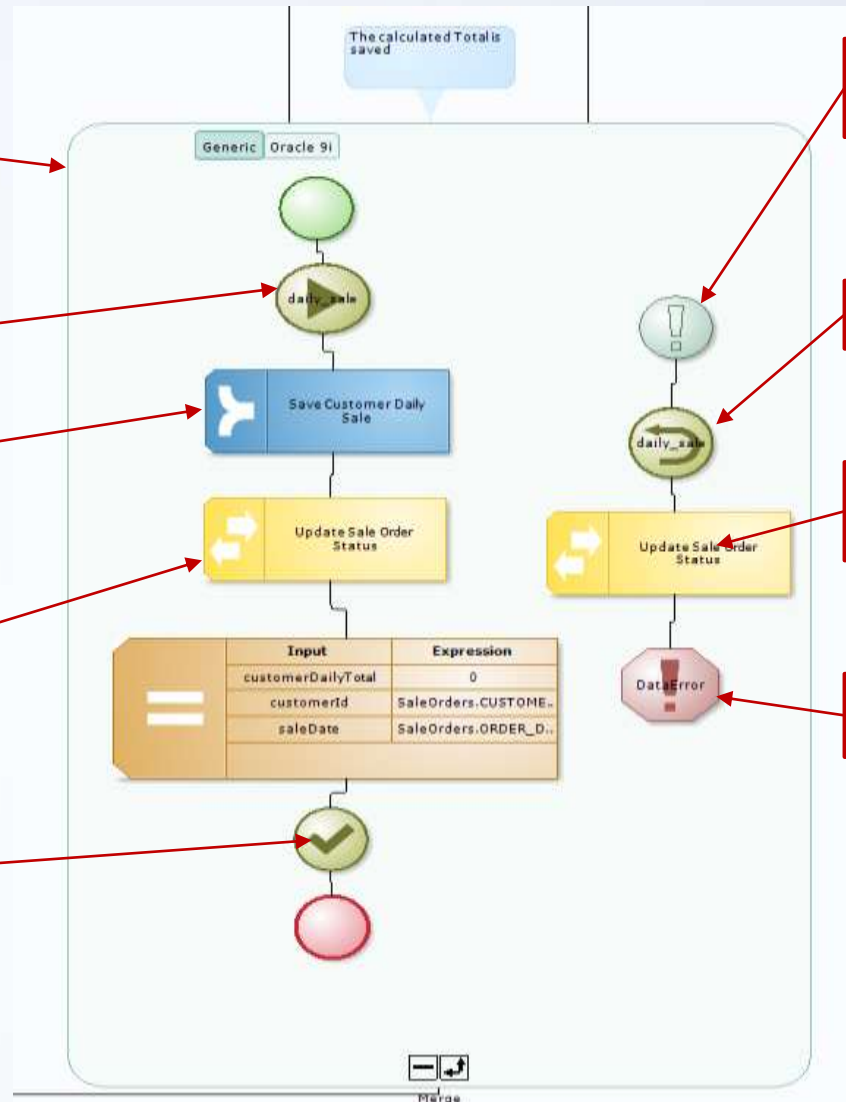
Commit the transaction

Local Exceptions handler begins here

Rollback the transaction

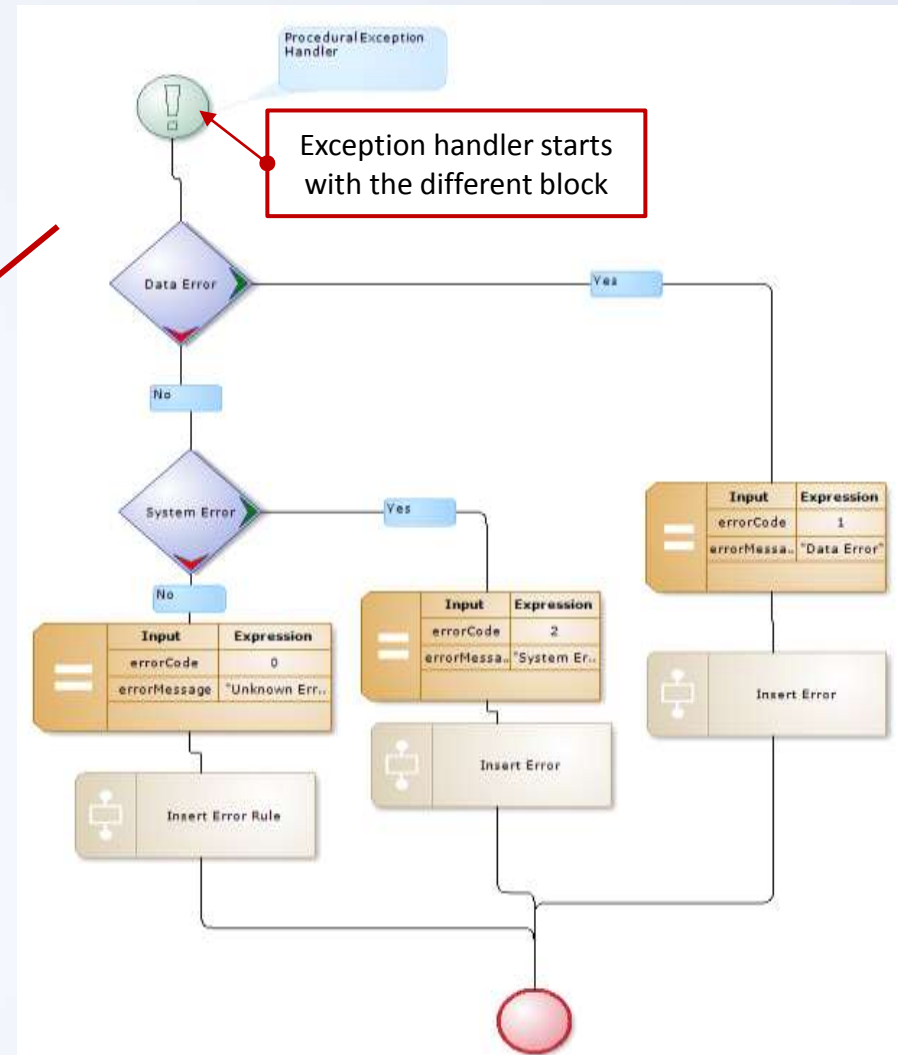
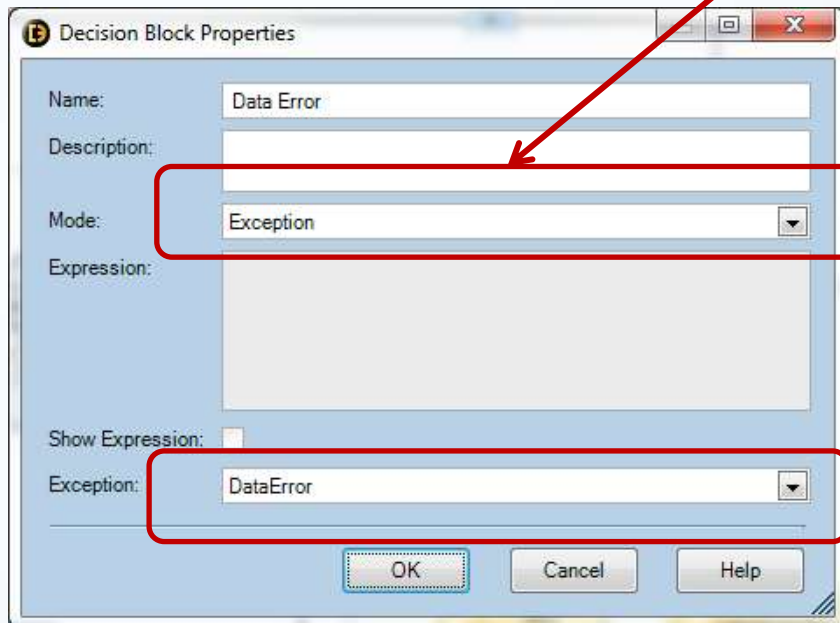
Update the Sale Order status to "error"

Re-raise the exception



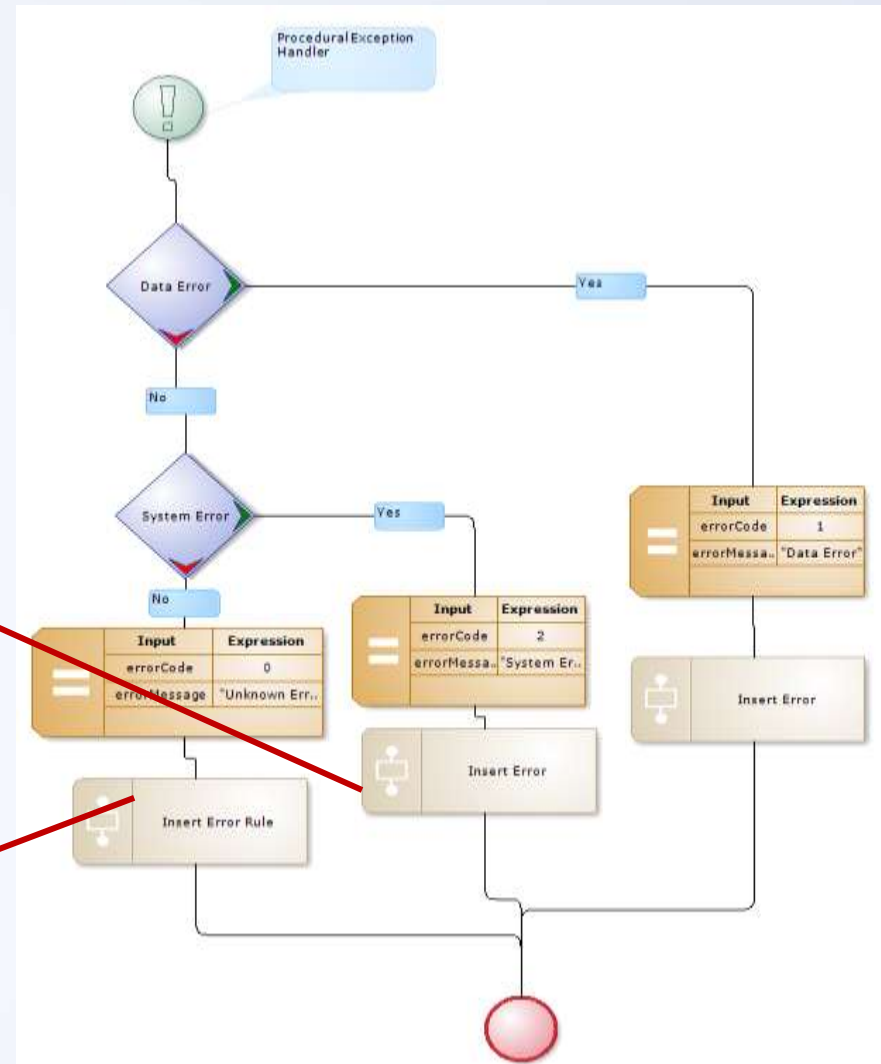
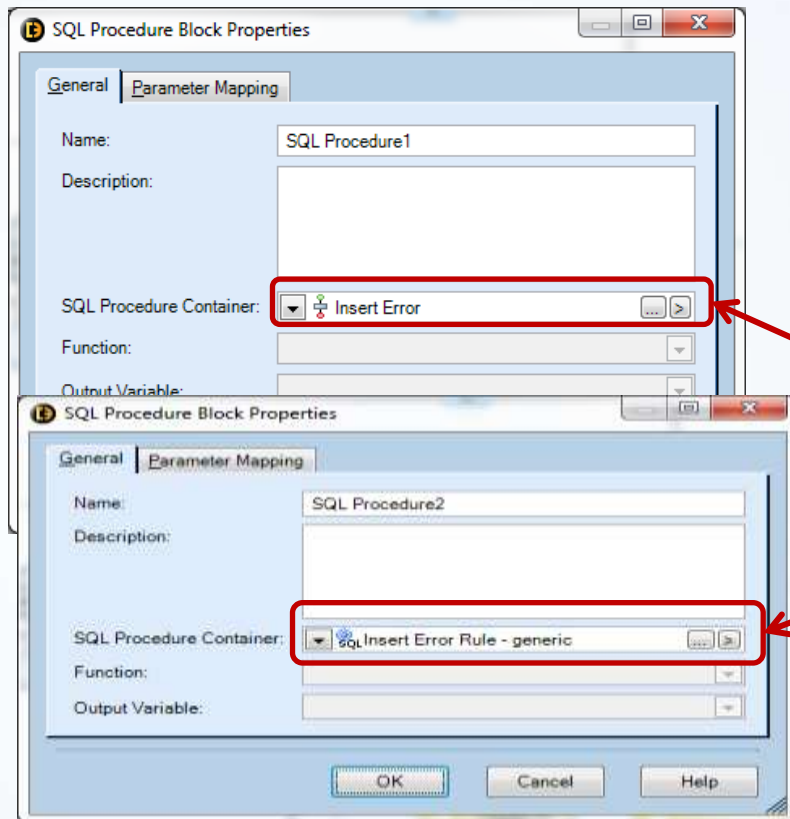
Exception Handler

- Exception handler is the control flow alternative to the normal flow
- Starts when the exception is thrown from the normal flow
- The only blocks allowed directly after the Catch Exception are Decision and End



Calling other Procedures

- Calling stored procedures
 - Implemented as a SQL Procedure
 - Implemented as a SQL Rule
 - Declared in an External Call Format



External Stored Procedures

The screenshot displays the DBClarity Developer interface for configuring external stored procedures. At the top, the 'Name' field is set to 'Procedures' and the 'Description' field is empty. The 'Database' is set to 'Oracle', 'Version' to '11g', and 'Interface' to 'OCI'. A checkbox for 'Case sensitive names with special characters' is present.

Below this, a table lists functions and their associated stored procedures and SQL rules. A red box highlights the 'InsertError' function, with an arrow pointing to the 'SQL Rule / SQL Procedure' column.

Function Name	Schema Name	Stored Procedure	SQL Rule / SQL Procedure
NewContact		NewContact	SQL 26 - INSERT with par
InsertError		InsertError	

Below the table, a parameter mapping table is shown:

Name	Native Type	Data Type	Length	Precision	Scale	Direction	Default value type	Default value
pErrorCode	NUMBER	Numeric	0	38	0	Input	None	
pErrorMessage	VARCHAR2	String	4000	0	0	Input	None	

A red box highlights the 'InsertError' function in the table, with an arrow pointing to the 'SQL Procedure Block Properties' dialog box. This dialog box has two tabs: 'General' and 'Parameter Mapping'. The 'Parameter Mapping' tab is active, showing the 'Name' as 'InsertError', 'Description' as empty, 'SQL Procedure Container' as 'Procedures', 'Function' as 'InsertError', and 'Output Variable' as empty. The 'OK', 'Cancel', and 'Help' buttons are at the bottom.

On the right side, a diagram shows a call block in a rule. The call block is labeled 'InsertError' and has two input fields: 'errorCode' with a value of '1' and 'errorMessage' with a value of 'Data Error'. A red arrow points from the 'SQL Procedure Block Properties' dialog to this call block.

Two red boxes with arrows provide instructions: 'Use an External Call Format to refer to the SQL Procedure' (pointing to the 'InsertError' function in the table) and 'Use an External Call block in a Rule to call the stored procedure' (pointing to the 'InsertError' call block in the rule diagram).

More Information

- Download the product and find more information, including product workshops and training videos, visit the Microgen website www.microgen.com/dbclarity
- Send your product questions and feedback to the Microgen team at dbclarity.feedback@microgen.com
- Contact Microgen Support for all DBClarity Developer related support queries at dbclarity.support@microgen.com
- Follow DBClarity Developer on Twitter @**MCGN_DBClarity**

This document is designed to provide a training overview of Microgen DBClarity Developer™. Information in this document is subject to change without notice and does not represent a commitment on the part of Microgen.

The information contained in this document is proprietary and confidential to Microgen plc and must not, therefore, be disclosed to any third-party without the express written permission of Microgen. In addition, no part of this document may be reproduced or transmitted in any form or by any means electronic or mechanical including photocopying, recording or information storage and retrieval systems, for any purpose other than the recipient's personal use without the express written permission of Microgen.