

**Appendices -  
Advanced exercises**

# **ADV12** Dynamic execution

- ❖ Context : implement a dynamic execution of delivery with parent delivery
  - The link between deliveries and the execution logs are stored in 2 tables :

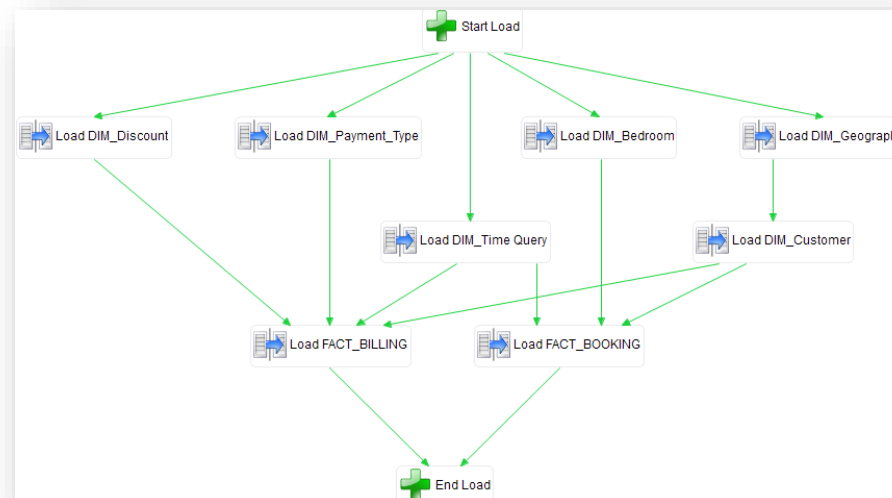
```
select * from DYNAMIC_LOAD.DLVRY_LINK
```

EXECT_PRCS	DLVRY_NAME	DLVRY_PRNT
Load ALL Datamart	Init LOAD_ALL_DATAMART	<null>
Load ALL Datamart	Load DIM_DISCOUNT	Init LOAD_ALL_DATAMART
Load ALL Datamart	Load DIM_PAYMENT_TYPE	Init LOAD_ALL_DATAMART
Load ALL Datamart	Load DIM_BEDROOM	Init LOAD_ALL_DATAMART
Load ALL Datamart	Load DIM_GEOGRAPHY	Init LOAD_ALL_DATAMART
Load ALL Datamart	Load DIM_TIME	Init LOAD_ALL_DATAMART
Load ALL Datamart	Load DIM_CUSTOMER	Load DIM_GEOGRAPHY
Load ALL Datamart	Load FACT_BILLING	Load DIM_CUSTOMER
Load ALL Datamart	Load FACT_BILLING	Load DIM_DISCOUNT
Load ALL Datamart	Load FACT_BILLING	Load DIM_TIME
Load ALL Datamart	Load FACT_BILLING	Load DIM_PAYMENT_TYPE
Load ALL Datamart	Load FACT_BOOKING	Load DIM_BEDROOM
Load ALL Datamart	Load FACT_BOOKING	Load DIM_CUSTOMER
Load ALL Datamart	Load FACT_BOOKING	Load DIM_TIME
Load ALL Datamart	Terminate LOAD_ALL_DATAMART	Load FACT_BOOKING
Load ALL Datamart	Terminate LOAD_ALL_DATAMART	Load FACT_BILLING

```
select * from DYNAMIC_LOAD.DLVRY EXEC LOG
```

EXECT_PRCS	EXECT_SEID	DLVRY_NAME	LOAD_STATUS	EXECT_START	EXECT_END
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Init LOAD_ALL_DATAMART	FINISHED	2020-07-08 18:52:43:125	2020-07-08 18:52:44:186
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Load DIM_BEDROOM	FINISHED	2020-07-08 18:52:44:536	2020-07-08 18:52:45:813
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Load DIM_DISCOUNT	FINISHED	2020-07-08 18:52:44:566	2020-07-08 18:52:45:730
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Load DIM_GEOGRAPHY	FINISHED	2020-07-08 18:52:44:597	2020-07-08 18:52:48:495
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Load DIM_PAYMENT_TYPE	FINISHED	2020-07-08 18:52:44:627	2020-07-08 18:52:45:995
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Load DIM_TIME	FINISHED	2020-07-08 18:52:44:661	2020-07-08 18:52:46:043
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Load DIM_CUSTOMER	FINISHED	2020-07-08 18:52:49:103	2020-07-08 18:52:50:166
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Load FACT_BILLING	FINISHED	2020-07-08 18:52:50:508	2020-07-08 18:53:00:012
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Load FACT_BOOKING	FINISHED	2020-07-08 18:52:50:537	2020-07-08 18:52:59:640
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Terminate LOAD_ALL_DATAMART	FINISHED	2020-07-08 18:53:00:501	2020-07-08 18:53:01:468

- To execute dynamically the equivalent of :



- ❖ 1 - Execute the SQL file (Init Environment.sql) to create the required tables on a Datamart connection

Init Environment.sql

Execute current SQL. Current SQL is the selected text or the complete file content if nothing is selected.

```

CREATE SCHEMA DYNAMIC_LOAD;

CREATE TABLE DYNAMIC_LOAD.DLVRY_EXEC_LOG
(EXECT_PRCS VARCHAR(100),
 EXECT_SEID VARCHAR(50),
 DLVRY_NAME VARCHAR(100),
 LOAD_STATUS VARCHAR(20),
 EXECT_START TIMESTAMP,
 EXECT_END TIMESTAMP);

CREATE TABLE DYNAMIC_LOAD.DLVRY_LINK
(EXECT_PRCS VARCHAR(100),
 DLVRY_NAME VARCHAR(100),
 DLVRY_PRNT VARCHAR(100));

INSERT INTO DYNAMIC_LOAD.DLVRY_LINK VALUES ('Load ALL Datamart','Init LOAD_ALL_DATAMART',NULL);
INSERT INTO DYNAMIC_LOAD.DLVRY_LINK VALUES ('Load ALL Datamart','Load DIM_DISCOUNT','Init LOAD_ALL_DATAMART');
INSERT INTO DYNAMIC_LOAD.DLVRY_LINK VALUES ('Load ALL Datamart','Load DIM_PAYMENT_TYPE','Init LOAD_ALL_DATAMART');
INSERT INTO DYNAMIC_LOAD.DLVRY_LINK VALUES ('Load ALL Datamart','Load DIM_BEDROOM','Init LOAD_ALL_DATAMART');
INSERT INTO DYNAMIC_LOAD.DLVRY_LINK VALUES ('Load ALL Datamart','Load DIM_GEOGRAPHY','Init LOAD_ALL_DATAMART');
    
```

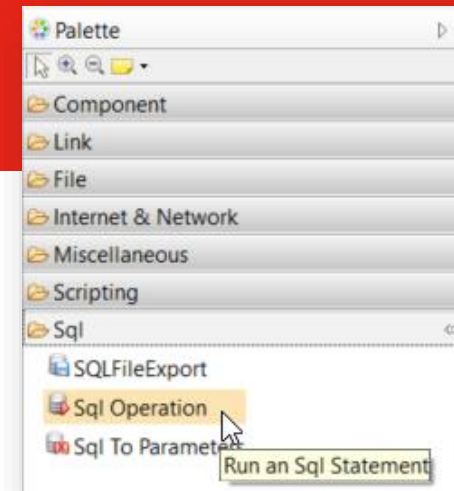
1

- ❖ 2 - Reverse the tables :

HSQL\_Datamart

- DYNAMIC\_LOAD
  - DLVRY\_EXEC\_LOG
    - EXECT\_PRCS
    - EXECT\_SEID
    - DLVRY\_NAME
    - LOAD\_STATUS
    - EXECT\_START
    - EXECT\_END
  - DLVRY\_LINK
    - EXECT\_PRCS
    - DLVRY\_NAME
    - DLVRY\_PRNT

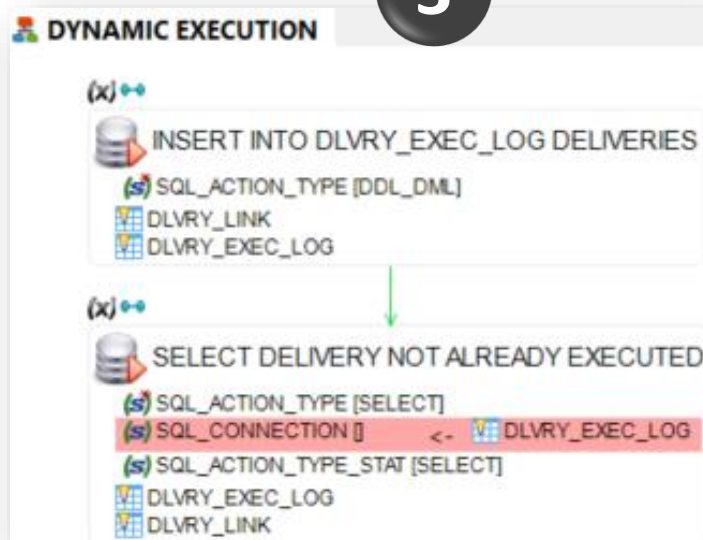
2



## ❖ 3 - Create a process with 2 SQL Operation actions

- The 1<sup>st</sup> SELECT has “DDL\_DML” as SQL\_ACTION\_TYPE
- The 2<sup>nd</sup> SELECT has “SELECT” as SQL\_ACTION\_TYPE

3

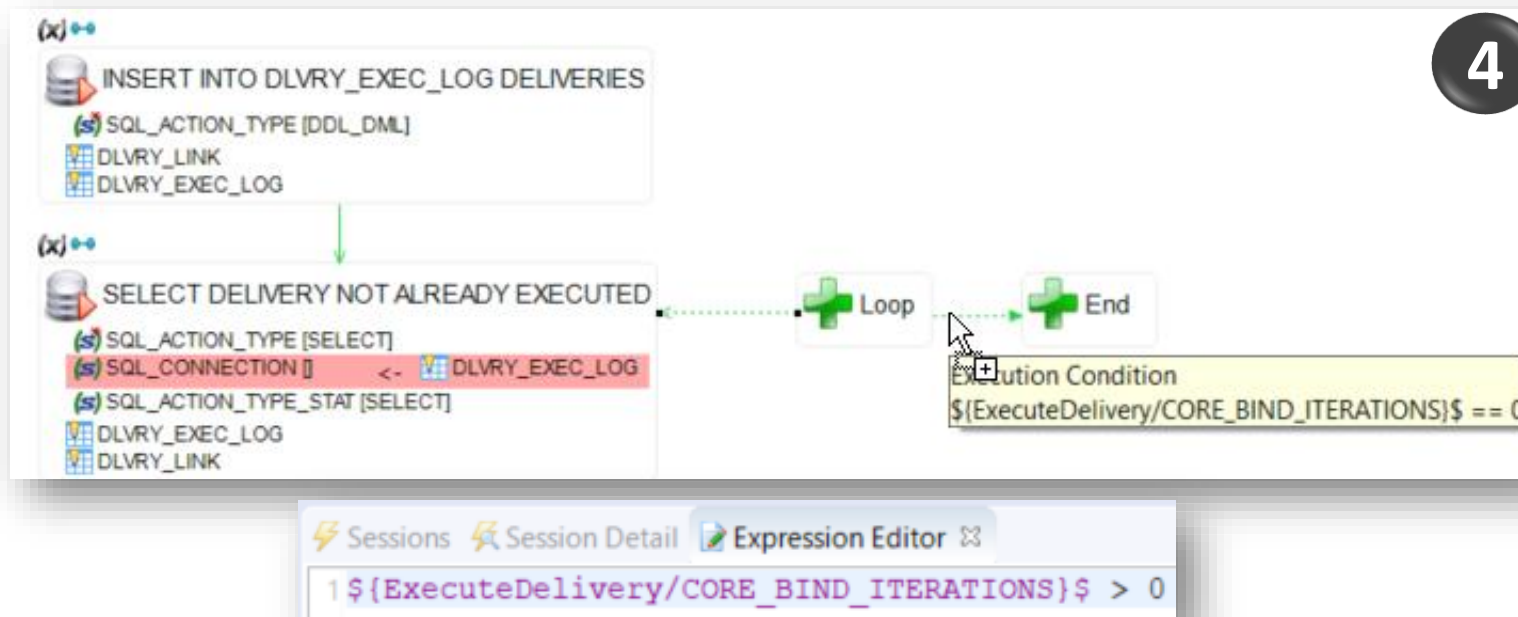
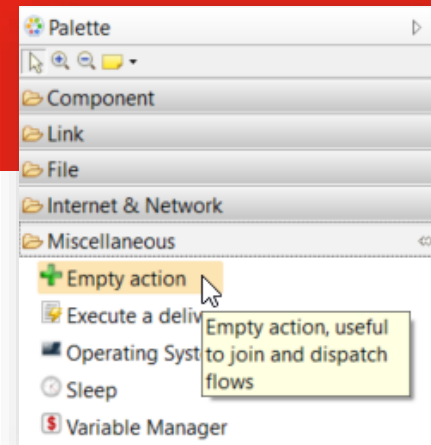


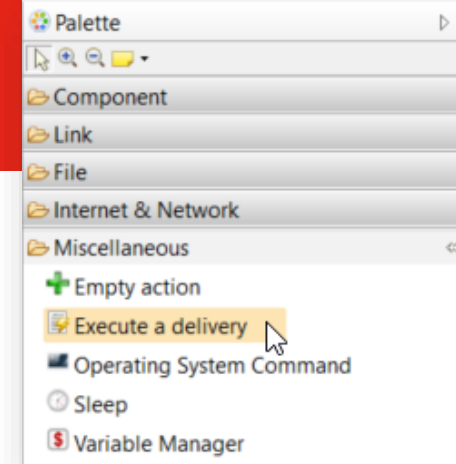
```
1 INSERT INTO %x({$DLVRY_EXEC_LOG/tech:physicalPath()})x%
2 SELECT DISTINCT EXECT_PRCS, '${/CORE_SESSION_ID}$', DLVRY_NAME, NULL, NULL, NULL
3 FROM %x({$DLVRY_LINK/tech:physicalPath()})x%
4 WHERE EXECT_PRCS='${~/EXECT_PRCS}$'
```

```
1 SELECT A.DLVRY_NAME
2 FROM %x({$DLVRY_EXEC_LOG/tech:physicalPath()})x% A
3 INNER JOIN %x({$DLVRY_LINK/tech:physicalPath()})x% B
4 ON A.DLVRY_NAME=B.DLVRY_NAME
5 AND A.EXECT_PRCS=B.EXECT_PRCS
6 WHERE A.LOAD_STATUS IS NULL AND B.DLVRY_PRNT IS NULL
7 AND A.EXECT_PRCS='${~/EXECT_PRCS}$' AND A.EXECT_SEID='${/CORE_SESSION_ID}$'
8 UNION
9 SELECT DLVRY_NAME
10 FROM (SELECT A.DLVRY_NAME, --L.DLVRY_PRNT,
11 SUM(CASE WHEN B.LOAD_STATUS = 'FINISHED' THEN 1 ELSE 0 END) AS NB_TREATED,
12 COUNT(*) AS NB_TOTAL
13 FROM %x({$DLVRY_EXEC_LOG/tech:physicalPath()})x% A
14 INNER JOIN %x({$DLVRY_LINK/tech:physicalPath()})x% L
15 ON A.DLVRY_NAME=L.DLVRY_NAME
16 AND A.EXECT_PRCS=L.EXECT_PRCS
17 INNER JOIN %x({$DLVRY_EXEC_LOG/tech:physicalPath()})x% B
18 ON L.DLVRY_PRNT=B.DLVRY_NAME AND A.EXECT_PRCS=B.EXECT_PRCS
19 AND A.EXECT_SEID=B.EXECT_SEID
20 WHERE A.LOAD_STATUS IS NULL AND A.EXECT_PRCS='${~/EXECT_PRCS}$'
21 AND A.EXECT_SEID='${/CORE_SESSION_ID}$'
22 GROUP BY A.DLVRY_NAME--, L.DLVRY_PRNT
23 )
24 WHERE NB_TREATED=NB_TOTAL
```



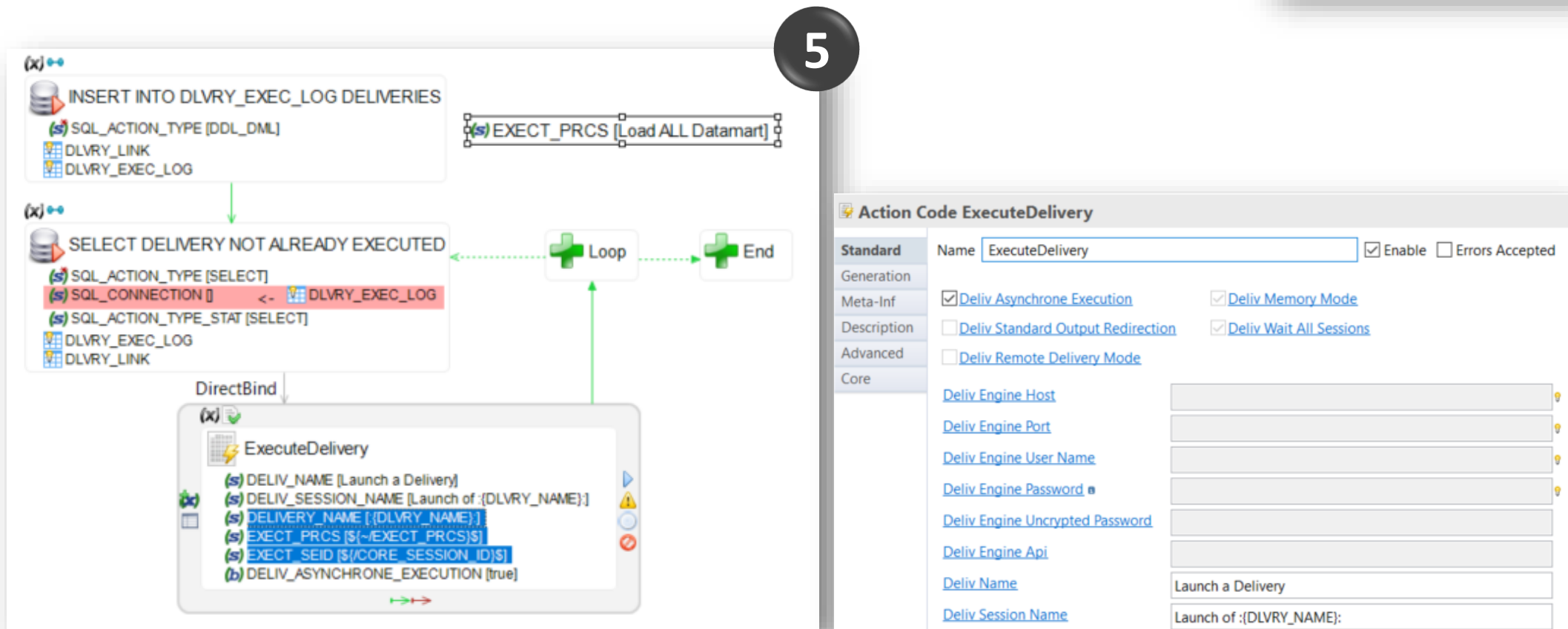
- ❖ 4 - Add two empty actions (Loop and End)
  - Add two opposite execution conditions on the 2 links
  - Both links that arrive to “SELECT DELIVERY...” action must be “Not mandatory”

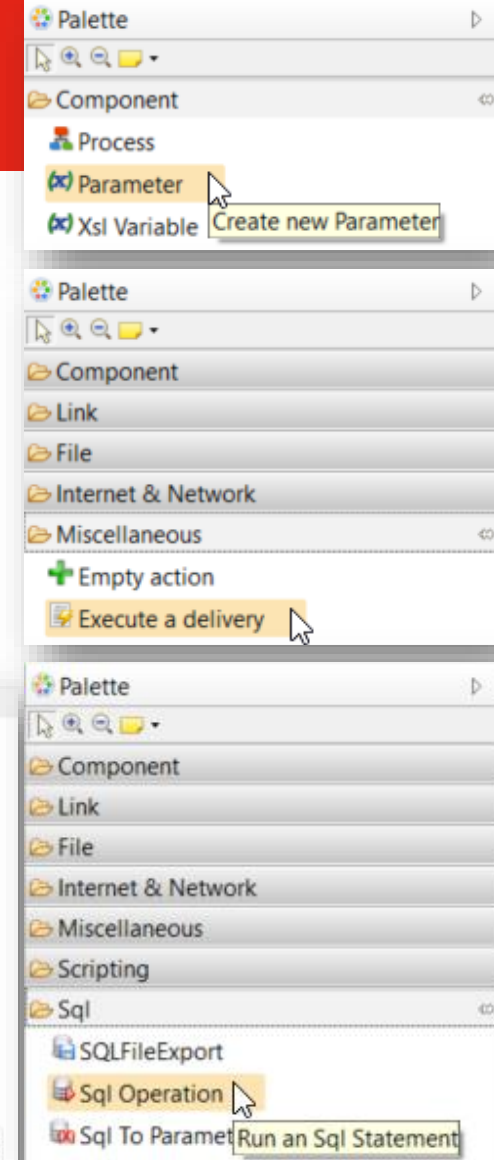




## ❖ 5 - Add an ExecuteDelivery action and a bind link

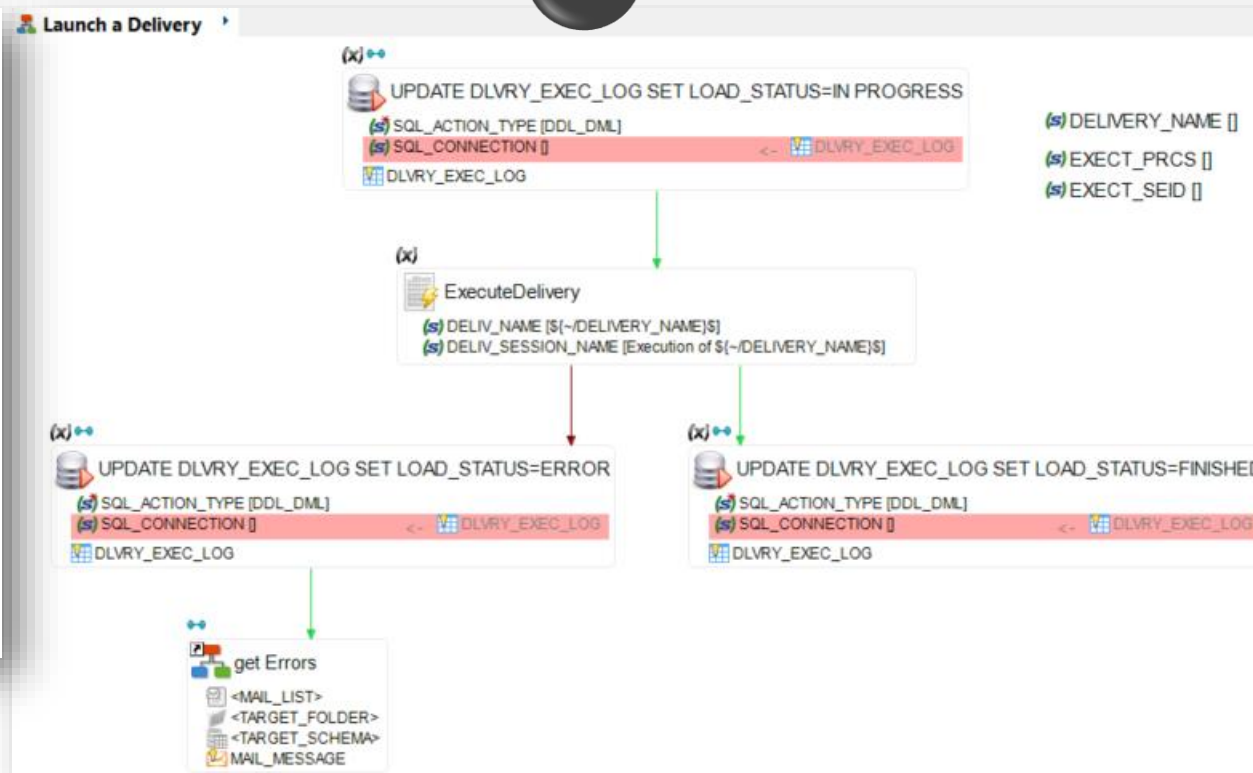
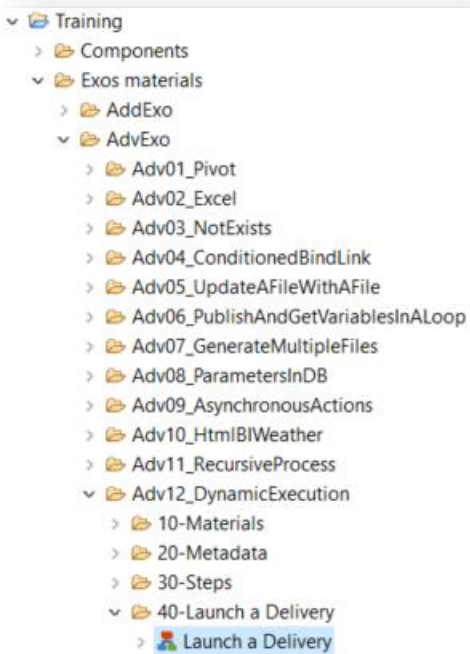
- Add 3 parameters in ExecuteDelivery Action
- Add a parameter in the process



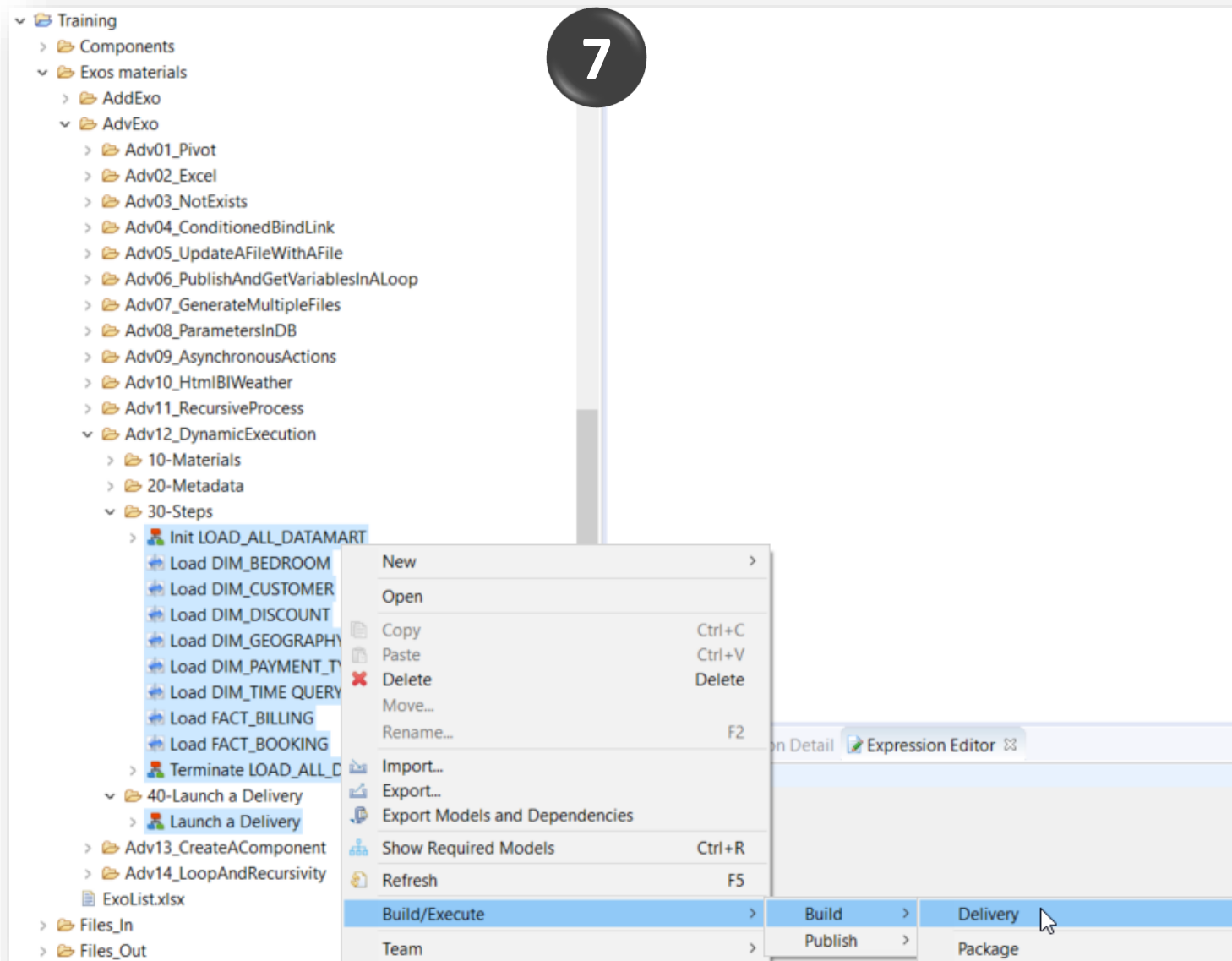


6

- ❖ 6 – Use the process named “Launch a Delivery”
  - With 3 parameters, a ExecuteDelivery ands 3 SQL Operations (UPDATE) actions with following status
    - IN PROGRESS, ERROR, FINISHED



## ❖ 7 - Build the following mappings and processes





## ❖ 8 - Execute DYNAMIC EXECUTION process

8

Start Time	Name	Status	Elapsed T...
▼ 2020/07/08 18:52:42.767	DYNAMIC EXECUTION	✓ EXECUTED	00:00:19.0...
> 2020/07/08 18:53:00.373	Launch of Terminate LOAD_ALL_DATAMART	✓ EXECUTED	00:00:01.1...
> 2020/07/08 18:52:50.411	Launch of Load FACT_BOOKING	✓ EXECUTED	00:00:09.2...
> 2020/07/08 18:52:50.382	Launch of Load FACT_BILLING	✓ EXECUTED	00:00:09.6...
> 2020/07/08 18:52:48.975	Launch of Load DIM_CUSTOMER	✓ EXECUTED	00:00:01.2...
> 2020/07/08 18:52:44.530	Launch of Load DIM_TIME	✓ EXECUTED	00:00:01.5...
> 2020/07/08 18:52:44.500	Launch of Load DIM_PAYMENT_TYPE	✓ EXECUTED	00:00:01.5...
> 2020/07/08 18:52:44.469	Launch of Load DIM_GEOGRAPHY	✓ EXECUTED	00:00:04.0...
> 2020/07/08 18:52:44.439	Launch of Load DIM_DISCOUNT	✓ EXECUTED	00:00:01.3...
> 2020/07/08 18:52:44.408	Launch of Load DIM_BEDROOM	✓ EXECUTED	00:00:01.4...
> 2020/07/08 18:52:42.997	Launch of Init LOAD_ALL_DATAMART	✓ EXECUTED	00:00:01.2...

HSQL_Datamart(5).sql					
HSQL_Datamart (_SfGD0A_IeekWbKD5uGsVQ)/sa					
Limit Rows: 100 PUBLIC					
<pre>SELECT * FROM DYNAMIC_LOAD.DLVRY_EXEC_LOG ORDER BY EXECT_SEID, EXECT_START</pre>					
1 [SELECT * FROM DYNAMIC_...] Messages					
EXECT_PRCS	EXECT_SEID	DLVRY_NAME	LOAD_STATUS	EXECT_START	EXECT_END
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Init LOAD_ALL_DATAMART	FINISHED	2020-07-08 18:52:43:125	2020-07-08 18:52:44:186
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Load DIM_BEDROOM	FINISHED	2020-07-08 18:52:44:536	2020-07-08 18:52:45:813
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Load DIM_DISCOUNT	FINISHED	2020-07-08 18:52:44:566	2020-07-08 18:52:45:730
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Load DIM_GEOGRAPHY	FINISHED	2020-07-08 18:52:44:597	2020-07-08 18:52:48:495
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Load DIM_PAYMENT_TYPE	FINISHED	2020-07-08 18:52:44:627	2020-07-08 18:52:45:995
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Load DIM_TIME	FINISHED	2020-07-08 18:52:44:661	2020-07-08 18:52:46:043
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Load DIM_CUSTOMER	FINISHED	2020-07-08 18:52:49:103	2020-07-08 18:52:50:166
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Load FACT_BILLING	FINISHED	2020-07-08 18:52:50:508	2020-07-08 18:53:00:012
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Load FACT_BOOKING	FINISHED	2020-07-08 18:52:50:537	2020-07-08 18:52:59:640
Load ALL Datamart	c0a8000a01732f57fe84f3c25edba589	Terminate LOAD_ALL_DATAMART	FINISHED	2020-07-08 18:53:00:501	2020-07-08 18:53:01:468