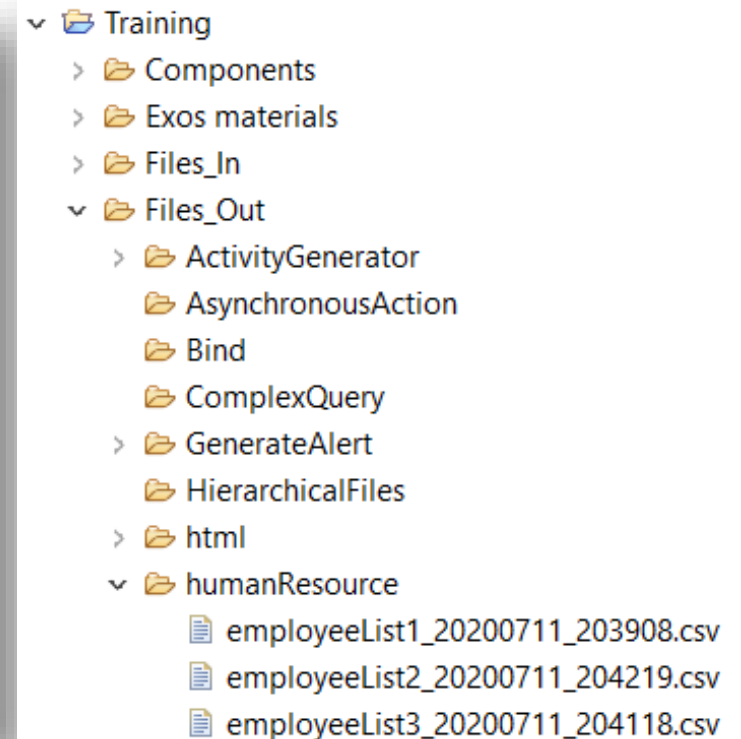
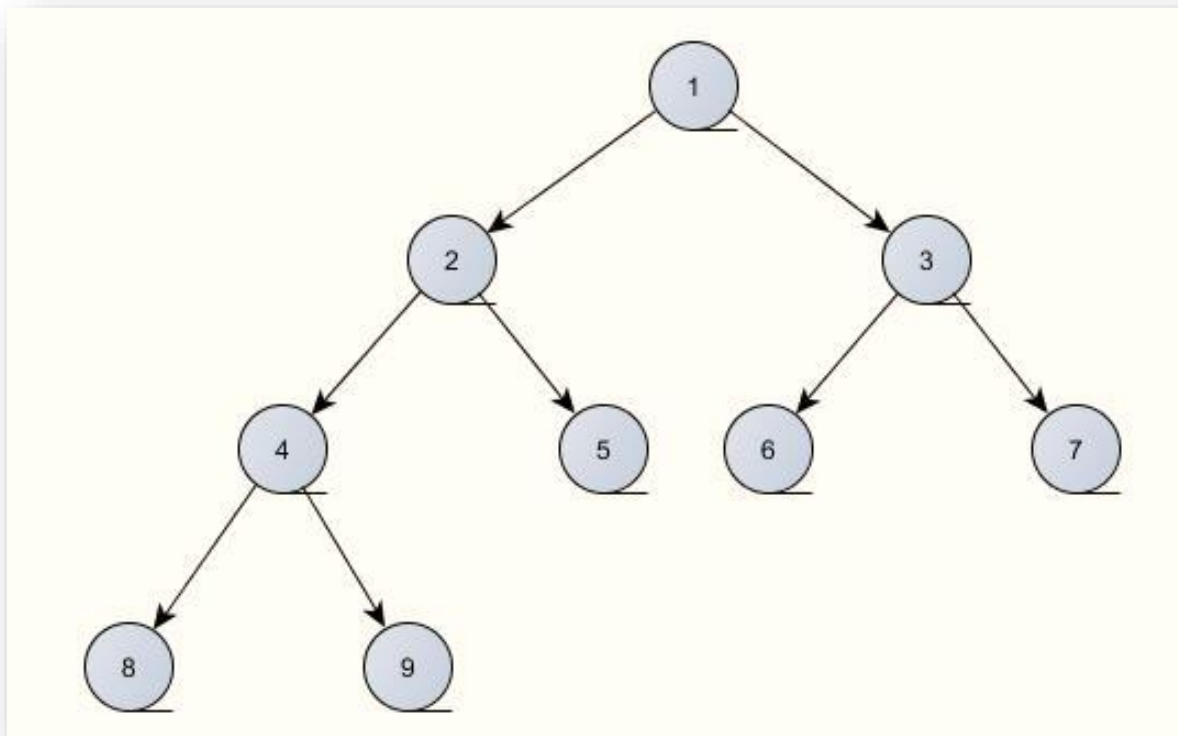


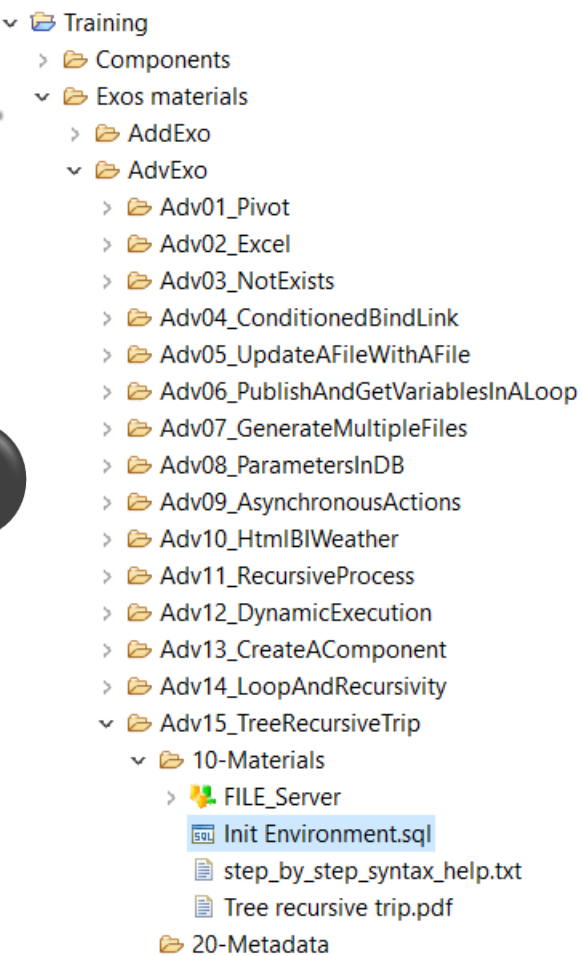
**Appendices -  
Advanced exercises**

# **ADV15 Tree recursive trip**

❖ Purpose : study 3 ways to browse a employee tree stored in database to generate a employee' list file

- 1<sup>st</sup> sol : With a recursivity in a physical process
- 2<sup>nd</sup> sol : With a mapping and a recursive query metadata
- 3<sup>th</sup> sol : With a recursive SQL query in a SQL File export action



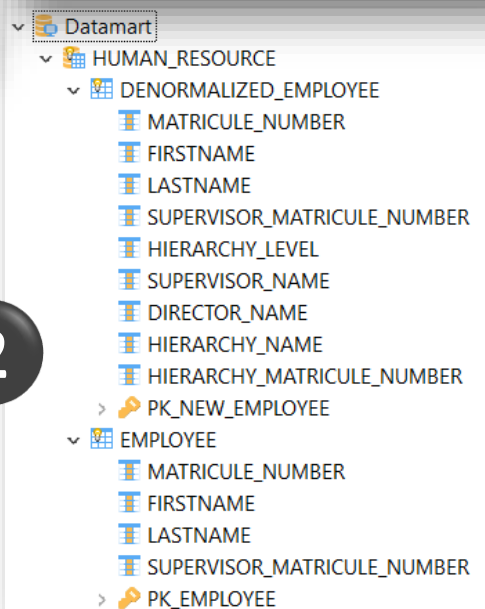


1

```
SQL Init Environment.sql
DROP SCHEMA HUMAN_RESOURCE IF EXISTS CASCADE;
CREATE SCHEMA HUMAN_RESOURCE;

CREATE TABLE HUMAN_RESOURCE.EMPLOYEE (
  MATRICULE_NUMBER CHAR(3) NOT NULL,
  FIRSTNAME VARCHAR(100),
  LASTNAME VARCHAR(100),
  SUPERVISOR_MATRICULE_NUMBER CHAR(3),
  CONSTRAINT PK_EMPLOYEE PRIMARY KEY (MATRICULE_NUMBER));

INSERT INTO HUMAN_RESOURCE.EMPLOYEE VALUES ('001', 'Ken', 'Sanchez', '000');
INSERT INTO HUMAN_RESOURCE.EMPLOYEE VALUES ('002', 'David', 'Bradley', '001');
INSERT INTO HUMAN_RESOURCE.EMPLOYEE VALUES ('003', 'Dave', 'King', '001');
INSERT INTO HUMAN_RESOURCE.EMPLOYEE VALUES ('004', 'Brian', 'Welcker', '001');
```



2

❖ 2 - Reverse the schema and the tables

- ❖ 3 – Create a process with one empty action and two SQL Operations (one in “SELECT” as SQL ACTION TYPE)
- Two links between the empty action and each SQL operation with a execution condition on each link
- The two selected links are “not mandatory”

**Generate recursive employeeList1**

3

Execution

Execution Type Successful

Triggering Behavior Mandatory

Condition

`__ctx__.getVariableValue("~/hierarchical_level")==0`

**Start**

Execution

Execution Type Successful

Triggering Behavior Not Mandatory

Condition

`__ctx__.getVariableValue("~/hierarchical_level")!=0`

**TRUNCATE denormalized\_employee**

DENORMALIZED\_EMPLOYEE

**SELECT current\_employee**

SQL\_ACTION\_TYPE [SELECT]

EMPLOYEE

Sessions Session Detail Expression Editor Step Detail Variable Statistic Prop

```

1 SELECT MATRICULE_NUMBER, FIRSTNAME, LASTNAME,
2       FIRSTNAME+' '+LASTNAME AS NAME, SUPERVISOR_MATRICULE_NUMBER,
3       %e(rhino){ __ctx__.retValue = __ctx__.getVariableValue("../hi
4       CASE WHEN %e(rhino){ __ctx__.retValue = __ctx__.getVariableVa
5       CASE WHEN %e(rhino){ __ctx__.retValue = __ctx__.getVariableVa
6       CASE WHEN %e(rhino){ __ctx__.retValue = __ctx__.getVariableVa
7       CASE WHEN %e(rhino){ __ctx__.retValue = __ctx__.getVariableVa
8 FROM   %x{$EMPLOYEE/tech:physicalPath()}x%
9 WHERE  SUPERVISOR_MATRICULE_NUMBER=' %e(rhino){ __ctx__.retValue =

```

Sessions Session Detail Expression Editor Step Detail Variable Statistic Prop

1 Truncate table %x{\$DENORMALIZED\_EMPLOYEE/tech:physicalPath()}x%

Palette

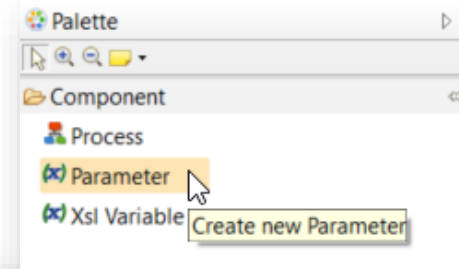
- Component
- Link
- File
- Internet & Network
- Miscellaneous
  - Empty action

Empty action, useful to join and dispatch flows
  - Execute a delivery
  - Operating System
  - Sleep
  - Variable Manager
- Scripting
  - SQLFileExport
  - Sql Operation

Run an Sql Statement
  - Sql To Parameters

## ❖ 4 - Create the following parameters

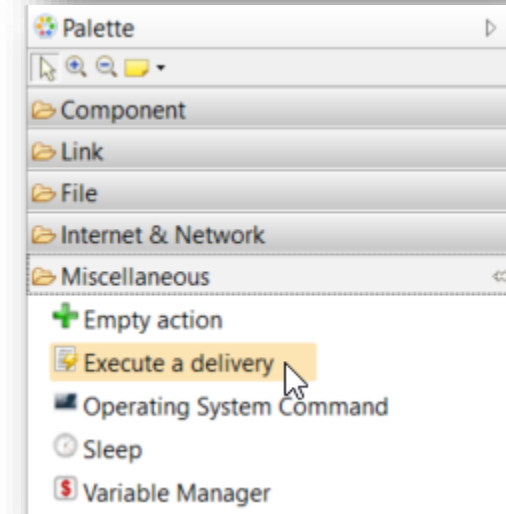
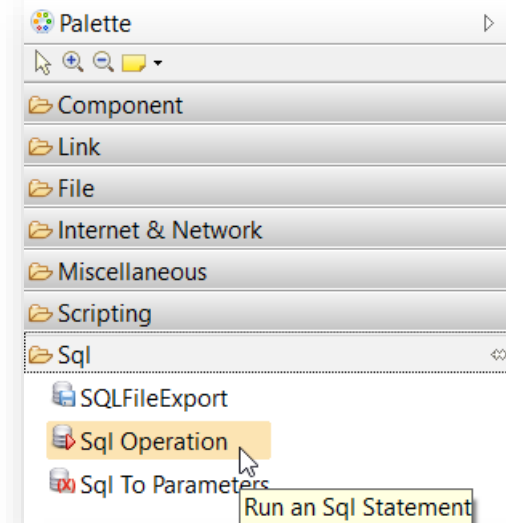
4





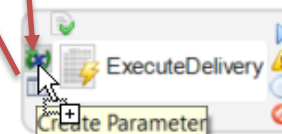
## ❖ 5 - Create one “SQL operation” and one “ExecuteDeliv” actions

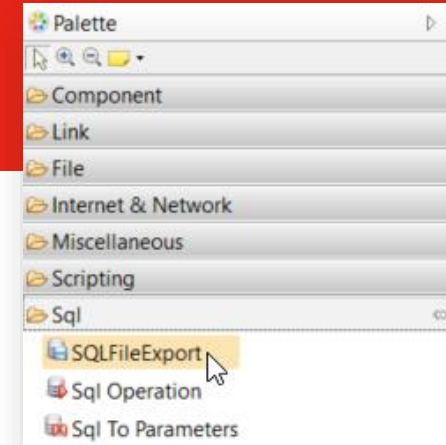
- The SQL Operation has “INSERT” as “SQL\_ACTION\_TYPE” and as “SQL\_ACTION\_TYPE\_STAT”
- Create the same parameters in the execDeliv as in the previous slide
- Create two bind links and a successful link as displayed



```

1 INSERT INTO %x{$DENORMALIZED_EMPLOYEE/tech:physicalPath()}%x%
2 VALUES (: {MATRICULE_NUMBER} :, : {FIRSTNAME} :, : {LASTNAME} :,
3         : {SUPERVISOR_MATRICULE_NUMBER} :, : {HIERARCHICAL_LEVEL} :,
4         : {SUPERVISOR_NAME} :, : {DIRECTOR_NAME} :, : {HIERARCHY_NAME} :,
5         CASE WHEN : {HIERARCHY_MATRICULE_NUMBER} :=' ' THEN : {MATRICULE_NUMBER} :
6         ELSE : {HIERARCHY_MATRICULE_NUMBER} :+'.':+ : {MATRICULE_NUMBER} : END)
  
```





## ❖ 6 - Create one "SQL FileExport"

- The link between the execDeliv and the SQLFileExport has an execution and a generation conditions

6

**Workflow Diagram:**

- Start** (Green Plus)
- TRUNCATE denormalized\_employee** (Database icon)
- DENORMALIZED\_EMPLOYEE** (Table icon)
- SELECT current\_employee** (Database icon)
- SQL\_ACTION\_TYPE [SELECT]** (Variable icon)
- EMPLOYEE** (Table icon)
- DirectBind** (Green arrow)
- INSERT current\_employee** (Database icon)
- DENORMALIZED\_EMPLOYEE** (Table icon)
- SQLFileExport** (Database icon)
- SQL\_EXP\_FILENAME [%x(\$humanResource\_fold...]** (Variable icon)
- SQL\_EXP\_GENERATE\_HEADER [true]** (Variable icon)
- DENORMALIZED\_EMPLOYEE** (Table icon)
- humanResource\_folder** (Folder icon)

**ExecDeliv Recursive treatment of current employee subordinates:**

```

(b) generateReport [true]
(i) hierarchical_level [0]
(s) hierarchyMatriculeNumber []
(s) hierarchyName []
(s) supervisor [000]
(s) supervisorName []
(s) directorName []
(x)
(i) supervisor [{MATRICULE_NUMBER}]
(i) hierarchical_level [{HIERARCHICAL_LEVEL}]
(s) supervisorName [{NAME}]
(s) directorName [%e(rhino){ if ($_.hierarchical_level)$==0) {.(NAME):} else {$. /...
(s) generateReport [%e(rhino){ __ctx__refValue =~$~/generateReport$}e(rhino)...
(s) hierarchyName [%e(rhino){ if ($_.hierarchical_level)$==0) {.(NAME):} else {$. /...
(s) hierarchyMatriculeNumber [%e(rhino){ if ($_.hierarchical_level)$==0) {.(MATRI...
(s) DELIV_NAME [Generate recursive employeeList1]
(s) DELIV_SESSION_NAME [Exec hierarchical level %e(jython){ __ctx__refValue =...
  
```

**SQLFileExport Configuration:**

Category	Property	Value
Execution	Execution Type	Successful
	Triggering Behavior	Mandatory
	Condition	__ctx__getVariableValue("~/hierarchical_level") = 0
	Core	
Generation	Generation Type	Anyway
	Condition	md:paramValue(\$this,'generateReport') = 'true'
	Description	

**File Browser:**

- Server
  - humanResource\_folder
    - employeeList2

**Action Code SQLFileExport:**

Category	Property	Value
Standard	Name	SQLFileExport
	Generation	
Meta-Inf	Sql Exp File Append	<input type="checkbox"/>
	Sql Exp Generate If Empty	<input type="checkbox"/>
Description	Sql Exp Generate Header	<input checked="" type="checkbox"/>
	Sql Exp Add Record Number	<input type="checkbox"/>
Advanced	Sql Exp Boolean As Numeric	<input checked="" type="checkbox"/>
	Sql Exp Absolute Numeric	<input type="checkbox"/>
Core	Sql Exp Row Sep On Last Row	<input type="checkbox"/>
	Sql Exp Use Numeric Grouping Char	<input type="checkbox"/>
	Sql Exp Row Sep On First Row When Append	<input checked="" type="checkbox"/>
	Sql Exp Enclose All Fields	<input type="checkbox"/>
Sql Exp Filename		%x(\$humanResource_folder/tech:path0)x%/employeeList1_%x(md:formatDate("yyyyMMdd_HHmss"))x%.csv

**Expression Editor:**

```

1 SELECT *
2 FROM HUMAN_RESOURCE.DENORMALIZED_EMPLOYEE
3 ORDER BY HIERARCHY_MATRICULE_NUMBER
  
```

**File Browser (Output Path):**

- Training
  - Files\_Out
    - employeeList1\_20200711\_203908.csv

## ❖ 7 - Create a query in hotel datamart database

7

type filter text

**Datamart**

- HUMAN\_RESOURCE
  - DENORMALIZED\_EMPLOYEE
  - EMPLOYEE
- QF\_HUMAN\_RESOURCE
  - q\_denormalized\_employee**
    - MATRICULE\_NUMBER
    - FIRSTNAME
    - LASTNAME
    - SUPERVISOR\_MATRICULE\_NUMBER
    - HIERARCHICAL\_LEVEL
    - SUPERVISOR\_NAME
    - DIRECTOR\_NAME
    - HIERARCHICAL\_NAME
    - HIERARCHY\_MATRICULE\_NUMBER

Standard	Name	Expression
Customization	q_denormalized_employee	WITH RECURSIVE denormalized_employee (matricule_number,firstname,lastname,supervisor_matricule_number,hierarchical_level,supervisor_name,director_name,hierarchical_name,hierarchy_matricule_number) AS ( SELECT matricule_number,firstname,lastname,supervisor_matricule_number,1 as hierarchical_level,null as supervisor_name,firstname+''+lastname as director_name, CAST(null AS VARCHAR(500)) as hierarchical_name,CAST(matricule_number AS VARCHAR(100)) AS hierarchy_matricule_number FROM HUMAN_RESOURCE.employee WHERE supervisor_matricule_number='000' UNION SELECT a.matricule_number,a.firstname,a.lastname,a.supervisor_matricule_number,b.hierarchical_level+1,b.firstname+''+b.lastname as supervisor_name, b.director_name,CASE WHEN hierarchical_name IS null THEN b.firstname+''+b.lastname ELSE b.hierarchical_name+'/' +b.firstname+''+b.lastname END as hierarchical_name, b.hierarchy_matricule_number+''+a.matricule_number AS hierarchy_matricule_number FROM HUMAN_RESOURCE.employee a INNER JOIN denormalized_employee b ON a.supervisor_matricule_number=b.matricule_number) SELECT * FROM DENORMALIZED_EMPLOYEE ORDER BY HIERARCHY_MATRICULE_NUMBER

## ❖ 8 - Create a Variable set metadata to be able to generate a file name with current date & time

8

type filter text

**Set**

- v\_current\_moment**
  - c\_hotel\_datamart

Standard	Name	Refresh Query
Description	v_current_moment	SELECT CAST(YEAR(current_timestamp)*10000+MONTH(current_timestamp)*100+DAY(current_timestamp) AS VARCHAR)+'_'+ SUBSTR(CAST(HOUR(current_timestamp)*10000+MINUTE(current_timestamp)*100+SECOND(current_timestamp) AS VARCHAR),1,6) FROM (VALUES(0))
Customization		
Externalize		
Core		
	Default Value	
	<input type="checkbox"/> Use native Xpath Expression	
	Saving Connection	
	Refresh Connection	Set.c_hotel_datamart
	Type	String



- ❖ 9 - Update the File metadata
  - Drag & drop the variable on the file level
  - Update the “Physical Name” property to use the variable

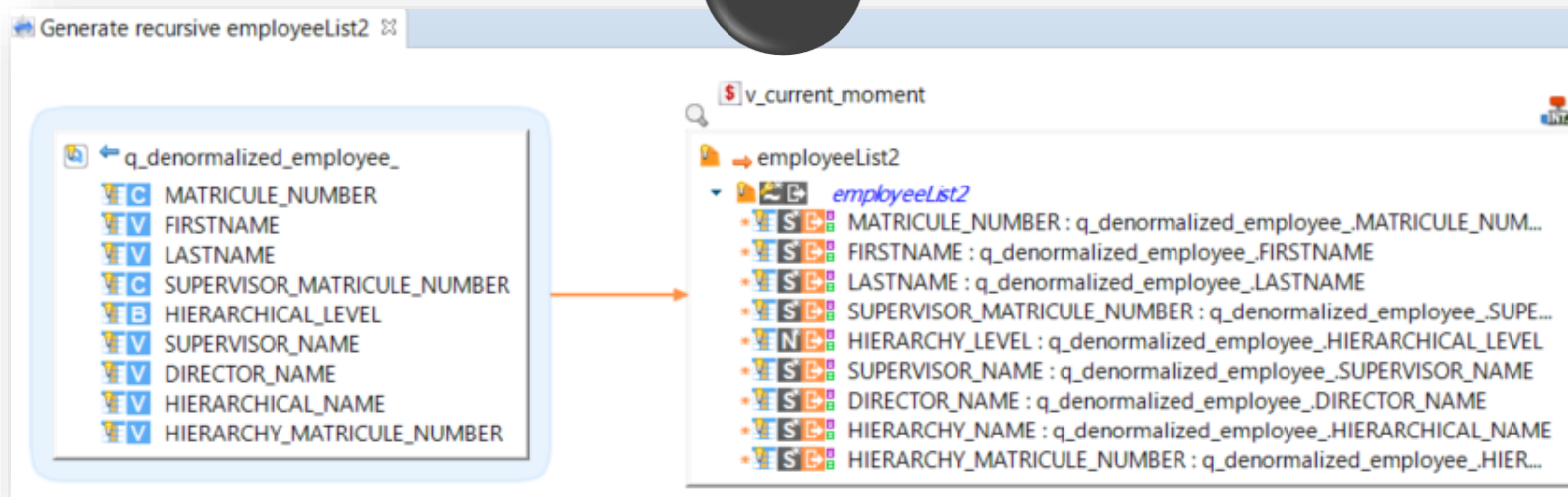
9

The screenshot shows the FILE\_Server.md interface. On the left, a tree view shows the hierarchy: Server > humanResource\_folder > employeeList2. The employeeList2 folder is expanded, showing a list of properties: MATRICULE\_NUMBER, FIRSTNAME, LASTNAME, SUPERVISOR\_MATRICULE\_NUMBER, HIERARCHY\_LEVEL, SUPERVISOR\_NAME, DIRECTOR\_NAME, HIERARCHY\_NAME, HIERARCHY\_MATRICULE\_NUMBER, and Meta Data Link v\_current\_moment. The Meta Data Link v\_current\_moment is highlighted. In the center, a list of properties for employeeList2 is shown: Standard, Reading Behaviour, Deprecated, Reverse Helper, Transformation, Customization, Externalize, and Core. The Core property is selected. On the right, a table shows the configuration for the Core property:

Property	Value
Name	employeeList2
Physical Name	employeeList2_%{v_current_moment}%.csv
Type	DELIMITED
Heading Lines To Skip	0
Ending Lines To Skip	0
Header Line Position	1
Line separator	0D0A
Field separator	3B
String delimiter	
Decimal separator	2E

- ❖ 10 – Use the query and the file in a new mapping
  - Drag & drop the variable in the mapping

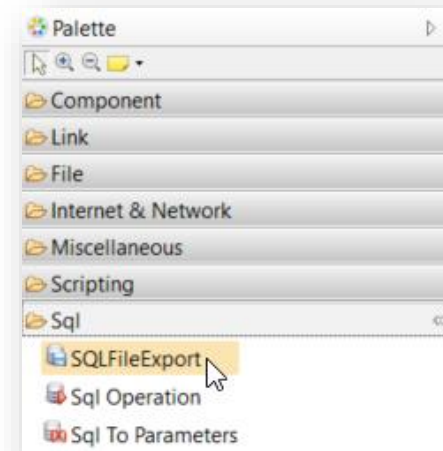
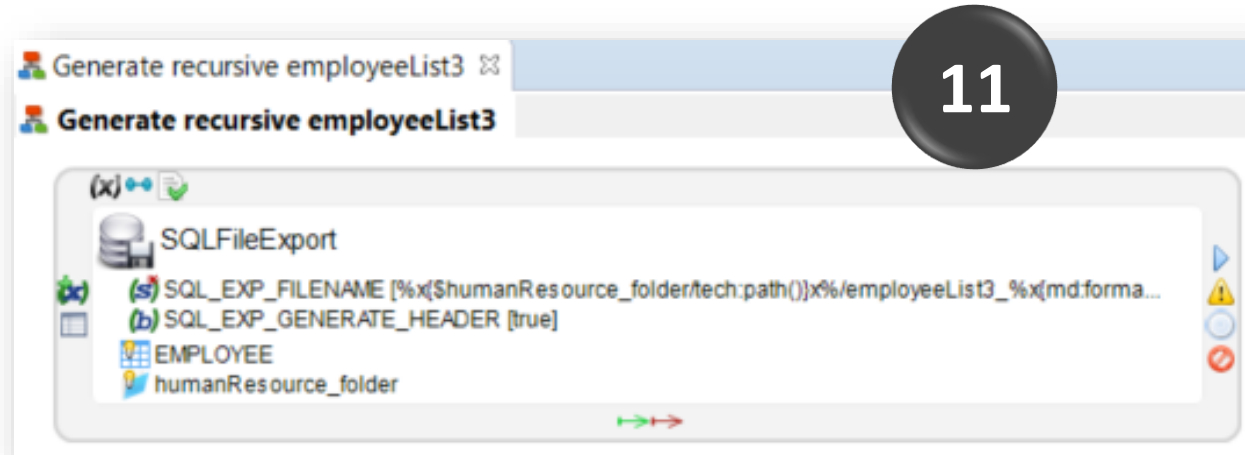
10



The screenshot shows a window titled "Template I\_employeeList2". It has two tabs: "Standard" and "Core". The "Standard" tab is selected, showing the following configuration:
 

- Action Process INTEGRATION File
- ☒ Do Insert ☐ Use Distinct
- ☒ Truncate Target File
- Generate Header

- ❖ 11 – Create a new process with a SQLFileExport action
  - Drag & drop EMPLOYEE table and the file folder on the action



```

1 WITH RECURSIVE denormalized_employee (matricule_number, first
2     supervisor_name, direct
3 SELECT matricule_number, firstname, lastname, supervisor_matricul
4     null as supervisor_name, firstname+' '+lastname as direct
5 FROM HUMAN_RESOURCE.employee
6 WHERE supervisor_matricule_number='000'
7 UNION
8 SELECT a.matricule_number, a.firstname, a.lastname, a.supervisor
9     b.firstname+' '+b.lastname as supervisor_name, b.direct
10 CASE WHEN hierarchical_name IS null THEN b.firstname+'
11     b.hierarchy_matricule_number+'.'+a.matricule_number AS
12 FROM HUMAN_RESOURCE.employee a
13     INNER JOIN denormalized_employee b ON a.supervisor_matr
14 )
15 SELECT * FROM DENORMALIZED_EMPLOYEE
16 ORDER BY HIERARCHY_MATRICULE_NUMBER
    
```

## Code SQLFileExport

Name

- |  |  |
|--|--|
| <input type="checkbox"/> <a href="#">Sql Exp File Append</a>                                 | <input type="checkbox"/> <a href="#">Sql Exp Generate If Empty</a>         |
| <input checked="" type="checkbox"/> <a href="#">Sql Exp Generate Header</a>                  | <input type="checkbox"/> <a href="#">Sql Exp Add Record Number</a>         |
| <input checked="" type="checkbox"/> <a href="#">Sql Exp Boolean As Numeric</a>               | <input type="checkbox"/> <a href="#">Sql Exp Absolute Numeric</a>          |
| <input type="checkbox"/> <a href="#">Sql Exp Row Sep On Last Row</a>                         | <input type="checkbox"/> <a href="#">Sql Exp Use Numeric Grouping Char</a> |
| <input checked="" type="checkbox"/> <a href="#">Sql Exp Row Sep On First Row When Append</a> | <input type="checkbox"/> <a href="#">Sql Exp Enclose All Fields</a>        |

Sql Exp Filename

%x(\$humanResource\_folder/tech:path())x%/employeeList3\_%x(md.formatDate("yyyyMMdd\_HH:mm:ss"))x%.csv

- ❖ 12 - After the execution of the 2 processes and the mapping, compare the results and the duration :

Start Time	Name	Status	Elapsed Time
2020/07/11 20:42:19.720	Generate recursive employeeList2	✓ EXECUTED	00:00:00.171
2020/07/11 20:41:17.985	Generate recursive employeeList3	✓ EXECUTED	00:00:00.127
2020/07/11 20:38:57.817	Generate recursive employeeList1	✓ EXECUTED	00:00:10.409
2020/07/11 20:39:03.27	Exec hierarchical level 1 - Employee matricule number 010 - Supervisor matricule number 000	✓ EXECUTED	00:00:04.540
2020/07/11 20:39:05	Exec hierarchical level 2 - Employee matricule number 015 - Supervisor matricule number 010	✓ EXECUTED	00:00:01.824
2020/07/11 20:39	Exec hierarchical level 3 - Employee matricule number 018 - Supervisor matricule number 015	✓ EXECUTED	00:00:00.298
2020/07/11 20:39	Exec hierarchical level 3 - Employee matricule number 017 - Supervisor matricule number 015	✓ EXECUTED	00:00:00.297
2020/07/11 20:39	Exec hierarchical level 3 - Employee matricule number 016 - Supervisor matricule number 015	✓ EXECUTED	00:00:00.303
2020/07/11 20:39:03	Exec hierarchical level 2 - Employee matricule number 011 - Supervisor matricule number 010	✓ EXECUTED	00:00:01.826
2020/07/11 20:39	Exec hierarchical level 3 - Employee matricule number 014 - Supervisor matricule number 011	✓ EXECUTED	00:00:00.302
2020/07/11 20:39	Exec hierarchical level 3 - Employee matricule number 013 - Supervisor matricule number 011	✓ EXECUTED	00:00:00.296
2020/07/11 20:39	Exec hierarchical level 3 - Employee matricule number 012 - Supervisor matricule number 011	✓ EXECUTED	00:00:00.301
2020/07/11 20:38:58.14	Exec hierarchical level 1 - Employee matricule number 001 - Supervisor matricule number 000	✓ EXECUTED	00:00:04.944
2020/07/11 20:39:01	Exec hierarchical level 2 - Employee matricule number 004 - Supervisor matricule number 001	✓ EXECUTED	00:00:01.421
2020/07/11 20:39	Exec hierarchical level 3 - Employee matricule number 009 - Supervisor matricule number 004	✓ EXECUTED	00:00:00.305
2020/07/11 20:39	Exec hierarchical level 3 - Employee matricule number 006 - Supervisor matricule number 004	✓ EXECUTED	00:00:00.296
2020/07/11 20:38:58	Exec hierarchical level 2 - Employee matricule number 003 - Supervisor matricule number 001	✓ EXECUTED	00:00:02.034
2020/07/11 20:38	Exec hierarchical level 3 - Employee matricule number 005 - Supervisor matricule number 003	✓ EXECUTED	00:00:01.426
2020/07/11 20	Exec hierarchical level 4 - Employee matricule number 008 - Supervisor matricule number 005	✓ EXECUTED	00:00:00.297
2020/07/11 20	Exec hierarchical level 4 - Employee matricule number 007 - Supervisor matricule number 005	✓ EXECUTED	00:00:00.303
2020/07/11 20:38:58	Exec hierarchical level 2 - Employee matricule number 002 - Supervisor matricule number 001	✓ EXECUTED	00:00:00.304

Training
Components
Exos materials
Files_In
Files_Out
ActivityGenerator
AsynchronousAction
Bind
ComplexQuery
GenerateAlert
HierarchicalFiles
html
humanResource
employeeList1_20200711_203908.csv
employeeList2_20200711_204219.csv
employeeList3_20200711_204118.csv

Training
Components
Exos materials
AddExo
AdvExo
Adv01_Pivot
Adv02_Excel
Adv03_NotExists
Adv04_ConditionedBindLink
Adv05_UpdateAFileWithAFile
Adv06_PublishAndGetVariablesInALoop
Adv07_GenerateMultipleFiles
Adv08_ParametersInDB
Adv09_AsynchronousActions
Adv10_HtmlBIWeather
Adv11_RecursiveProcess
Adv12_DynamicExecution
Adv13_CreateAComponent
Adv14_LoopAndRecursivity
Adv15_TreeRecursiveTrip
10-Materials
20-Metadata
Generate recursive employeeList1
Generate recursive employeeList2
Generate recursive employeeList3