**Active Transformations      Passive Transformations**      **Both Active &  Passive Transformation**  
1. Aggregator[Aggregator Transformation Icon](http://informatica-tutorials-for-beginners.blogspot.in/2013/11/aggregator-transformation.html)                                1. Data Masking                                 1. Custom[https://2.bp.blogspot.com/-GG5WRDdiXFE/UpCGL6fsKKI/AAAAAAAAAKs/_SVKYdE67z8/s320/image003.png](http://2.bp.blogspot.com/-GG5WRDdiXFE/UpCGL6fsKKI/AAAAAAAAAKs/_SVKYdE67z8/s1600/image003.png)  
2. Application Source Qualifier&[https://4.bp.blogspot.com/-qvwNeY8_9zs/UpCGLyAbgcI/AAAAAAAAAKo/SA4XFPCFiXk/s1600/image005.png](http://4.bp.blogspot.com/-qvwNeY8_9zs/UpCGLyAbgcI/AAAAAAAAAKo/SA4XFPCFiXk/s1600/image005.png)    2. Expression[https://3.bp.blogspot.com/-6z4N9TaTKVw/UpCGMeDslFI/AAAAAAAAAK0/rVy-99fas80/s320/image007.png](http://3.bp.blogspot.com/-6z4N9TaTKVw/UpCGMeDslFI/AAAAAAAAAK0/rVy-99fas80/s1600/image007.png)                               2. Lookup[https://1.bp.blogspot.com/-znITgftobFA/UpCGMngr1oI/AAAAAAAAALM/f5dDew4G_9o/s320/image009.png](http://1.bp.blogspot.com/-znITgftobFA/UpCGMngr1oI/AAAAAAAAALM/f5dDew4G_9o/s1600/image009.png)  
3. Source Qualifier[https://3.bp.blogspot.com/-QKETL1GQI6Q/UpCGMwbfb8I/AAAAAAAAALI/0vhjnBa9Z90/s1600/image011.png](http://3.bp.blogspot.com/-QKETL1GQI6Q/UpCGMwbfb8I/AAAAAAAAALI/0vhjnBa9Z90/s1600/image011.png)                       3. External Procedure                          3. Unstructured Data[https://3.bp.blogspot.com/-WYRWClhbQPg/UpCGNDvfCmI/AAAAAAAAALE/QBJBD4g34nU/s320/image013.png](http://3.bp.blogspot.com/-WYRWClhbQPg/UpCGNDvfCmI/AAAAAAAAALE/QBJBD4g34nU/s1600/image013.png)  
4. Sorter[https://4.bp.blogspot.com/-5s8Wy_Xep-4/UpCGNgaObxI/AAAAAAAAALU/SUWZoUVug18/s320/image015.png](http://4.bp.blogspot.com/-5s8Wy_Xep-4/UpCGNgaObxI/AAAAAAAAALU/SUWZoUVug18/s1600/image015.png)                                       4. Input                                              4. SQL[https://3.bp.blogspot.com/-vngp1VJfvOU/UpCGN-gx3rI/AAAAAAAAAMA/r5I5V0mNvNE/s320/image017.png](http://3.bp.blogspot.com/-vngp1VJfvOU/UpCGN-gx3rI/AAAAAAAAAMA/r5I5V0mNvNE/s1600/image017.png)  
5. Union[https://3.bp.blogspot.com/-fslNzWCZlUY/UpCGOA5zNhI/AAAAAAAAALk/7RUN7dSdoXo/s320/image019.png](http://3.bp.blogspot.com/-fslNzWCZlUY/UpCGOA5zNhI/AAAAAAAAALk/7RUN7dSdoXo/s1600/image019.png)                                        5. Output                                           5. Java[https://4.bp.blogspot.com/-YS4dsDh2Y-o/UpCGOgK6vAI/AAAAAAAAALo/EF8a5R3GhJ4/s320/image021.png](http://4.bp.blogspot.com/-YS4dsDh2Y-o/UpCGOgK6vAI/AAAAAAAAALo/EF8a5R3GhJ4/s1600/image021.png)  
6. Update Strategy[https://4.bp.blogspot.com/-OqvaMEq4H-w/UpCGPNPm8kI/AAAAAAAAAL4/L6lI_HMA8Ag/s320/image023.png](http://4.bp.blogspot.com/-OqvaMEq4H-w/UpCGPNPm8kI/AAAAAAAAAL4/L6lI_HMA8Ag/s1600/image023.png)                         6. Sequence Generator[https://2.bp.blogspot.com/-62aS5ECYdto/UpCGPFRvzdI/AAAAAAAAALw/d13RpFxtpzA/s320/image025.png](http://2.bp.blogspot.com/-62aS5ECYdto/UpCGPFRvzdI/AAAAAAAAALw/d13RpFxtpzA/s1600/image025.png)  
7. Rank[https://3.bp.blogspot.com/-ySslVBOWTIk/UpCGP-W8qcI/AAAAAAAAAMI/LsnpPjLfQrk/s320/image027.png](http://3.bp.blogspot.com/-ySslVBOWTIk/UpCGP-W8qcI/AAAAAAAAAMI/LsnpPjLfQrk/s1600/image027.png)                                        7. Stored Procedure[https://3.bp.blogspot.com/-EB6KofsekyY/UpCGQPUbG9I/AAAAAAAAAMU/yWVXCQdu6bw/s320/image029.png](http://3.bp.blogspot.com/-EB6KofsekyY/UpCGQPUbG9I/AAAAAAAAAMU/yWVXCQdu6bw/s1600/image029.png)  
8. Router[https://3.bp.blogspot.com/-LAE27Nt-Wnk/UpCGQRwEysI/AAAAAAAAAMY/77HIv-vdKC0/s320/image031.png](http://3.bp.blogspot.com/-LAE27Nt-Wnk/UpCGQRwEysI/AAAAAAAAAMY/77HIv-vdKC0/s1600/image031.png)                                      8. HTTP[https://3.bp.blogspot.com/-fPbSm2QuYZQ/UpCGQrKv8VI/AAAAAAAAAMc/eLhMQZMJ7yY/s320/image033.png](http://3.bp.blogspot.com/-fPbSm2QuYZQ/UpCGQrKv8VI/AAAAAAAAAMc/eLhMQZMJ7yY/s1600/image033.png)  
9. Filter[https://3.bp.blogspot.com/-ufvQvDdhKr4/UpCGRD8rt_I/AAAAAAAAAMk/5I-iPG8LXsE/s320/image035.png](http://3.bp.blogspot.com/-ufvQvDdhKr4/UpCGRD8rt_I/AAAAAAAAAMk/5I-iPG8LXsE/s1600/image035.png)   
10. Normalizer[https://4.bp.blogspot.com/-e8FSell_gG4/UpCGRTlmMpI/AAAAAAAAAM0/fRAUTMaa080/s320/image037.png](http://4.bp.blogspot.com/-e8FSell_gG4/UpCGRTlmMpI/AAAAAAAAAM0/fRAUTMaa080/s1600/image037.png)   
11. Transaction Control[https://1.bp.blogspot.com/-8zuec1l72Sk/UpCGRf5SWCI/AAAAAAAAAM4/j4yV2HG-IZs/s320/image039.png](http://1.bp.blogspot.com/-8zuec1l72Sk/UpCGRf5SWCI/AAAAAAAAAM4/j4yV2HG-IZs/s1600/image039.png)   
12. Joiner[https://3.bp.blogspot.com/-NJZ1PLJWCoY/UpCGSHMG24I/AAAAAAAAANA/l-oK4qzroXg/s320/image041.png](http://3.bp.blogspot.com/-NJZ1PLJWCoY/UpCGSHMG24I/AAAAAAAAANA/l-oK4qzroXg/s1600/image041.png)   
13. XML Source Qualifier[https://1.bp.blogspot.com/-L3EZN2AsIuY/UpCGTI-0QDI/AAAAAAAAANY/BGveFL9PVVk/s320/image043.png](http://1.bp.blogspot.com/-L3EZN2AsIuY/UpCGTI-0QDI/AAAAAAAAANY/BGveFL9PVVk/s1600/image043.png)   
14. XML Generator[https://1.bp.blogspot.com/-Be4HSbpGAng/UpCDhadcH6I/AAAAAAAAAKQ/N7AiAZLCIwo/s1600/image045.png](http://1.bp.blogspot.com/-Be4HSbpGAng/UpCDhadcH6I/AAAAAAAAAKQ/N7AiAZLCIwo/s1600/image045.png)  
15. XML Parser[https://2.bp.blogspot.com/-iZ8dXyHCZ00/UpCDhie8urI/AAAAAAAAAKY/drU8ACToBo4/s1600/image047.png](http://2.bp.blogspot.com/-iZ8dXyHCZ00/UpCDhie8urI/AAAAAAAAAKY/drU8ACToBo4/s1600/image047.png)

**Connected and Unconnected Transformation (Both)**              **Connected Transformation**  
1. External Procedure                                                                Rest All transformations are connected.  
2. Stored Procedure[https://3.bp.blogspot.com/-EB6KofsekyY/UpCGQPUbG9I/AAAAAAAAAMU/yWVXCQdu6bw/s320/image029.png](http://3.bp.blogspot.com/-EB6KofsekyY/UpCGQPUbG9I/AAAAAAAAAMU/yWVXCQdu6bw/s1600/image029.png)  
3. Lookup [https://1.bp.blogspot.com/-znITgftobFA/UpCGMngr1oI/AAAAAAAAALM/f5dDew4G_9o/s320/image009.png](http://1.bp.blogspot.com/-znITgftobFA/UpCGMngr1oI/AAAAAAAAALM/f5dDew4G_9o/s1600/image009.png)  
  
**MultiGroup Transformations**  
1. Union[https://3.bp.blogspot.com/-fslNzWCZlUY/UpCGOA5zNhI/AAAAAAAAALk/7RUN7dSdoXo/s320/image019.png](http://3.bp.blogspot.com/-fslNzWCZlUY/UpCGOA5zNhI/AAAAAAAAALk/7RUN7dSdoXo/s1600/image019.png)  
2. Router[https://3.bp.blogspot.com/-LAE27Nt-Wnk/UpCGQRwEysI/AAAAAAAAAMY/77HIv-vdKC0/s320/image031.png](http://3.bp.blogspot.com/-LAE27Nt-Wnk/UpCGQRwEysI/AAAAAAAAAMY/77HIv-vdKC0/s1600/image031.png)  
3. Joiner[https://3.bp.blogspot.com/-NJZ1PLJWCoY/UpCGSHMG24I/AAAAAAAAANA/l-oK4qzroXg/s320/image041.png](http://3.bp.blogspot.com/-NJZ1PLJWCoY/UpCGSHMG24I/AAAAAAAAANA/l-oK4qzroXg/s1600/image041.png)  
4. Custom[https://2.bp.blogspot.com/-GG5WRDdiXFE/UpCGL6fsKKI/AAAAAAAAAKs/_SVKYdE67z8/s320/image003.png](http://2.bp.blogspot.com/-GG5WRDdiXFE/UpCGL6fsKKI/AAAAAAAAAKs/_SVKYdE67z8/s1600/image003.png)  
5. Unstructured Data[https://3.bp.blogspot.com/-WYRWClhbQPg/UpCGNDvfCmI/AAAAAAAAALE/QBJBD4g34nU/s320/image013.png](http://3.bp.blogspot.com/-WYRWClhbQPg/UpCGNDvfCmI/AAAAAAAAALE/QBJBD4g34nU/s1600/image013.png)   
6. XML Source Qualifier[https://1.bp.blogspot.com/-L3EZN2AsIuY/UpCGTI-0QDI/AAAAAAAAANY/BGveFL9PVVk/s320/image043.png](http://1.bp.blogspot.com/-L3EZN2AsIuY/UpCGTI-0QDI/AAAAAAAAANY/BGveFL9PVVk/s1600/image043.png)   
7. XML Generator[https://1.bp.blogspot.com/-Be4HSbpGAng/UpCDhadcH6I/AAAAAAAAAKQ/N7AiAZLCIwo/s1600/image045.png](http://1.bp.blogspot.com/-Be4HSbpGAng/UpCDhadcH6I/AAAAAAAAAKQ/N7AiAZLCIwo/s1600/image045.png)  
8. XML Parser[https://2.bp.blogspot.com/-iZ8dXyHCZ00/UpCDhie8urI/AAAAAAAAAKY/drU8ACToBo4/s1600/image047.png](http://2.bp.blogspot.com/-iZ8dXyHCZ00/UpCDhie8urI/AAAAAAAAAKY/drU8ACToBo4/s1600/image047.png)  
9. XML Target Definition  
  
**Blocking Transformations**   
1.Custom transformation with Input may block property enabled.[https://2.bp.blogspot.com/-GG5WRDdiXFE/UpCGL6fsKKI/AAAAAAAAAKs/_SVKYdE67z8/s320/image003.png](http://2.bp.blogspot.com/-GG5WRDdiXFE/UpCGL6fsKKI/AAAAAAAAAKs/_SVKYdE67z8/s1600/image003.png)  
2.Joiner transformation configured for unsorted Input.[https://3.bp.blogspot.com/-NJZ1PLJWCoY/UpCGSHMG24I/AAAAAAAAANA/l-oK4qzroXg/s320/image041.png](http://3.bp.blogspot.com/-NJZ1PLJWCoY/UpCGSHMG24I/AAAAAAAAANA/l-oK4qzroXg/s1600/image041.png)

**SQL Transformations:**

You can pass the database connection information to the SQL transformation as input data at run time. The transformation processes external SQL scripts or SQL queries that you create in an SQL editor. The SQL transformation processes the query and returns rows and database errors.

When you create an SQL transformation, you configure the following options:

**Mode:-**The SQL transformation runs in one of the following modes:

**Script mode.**An SQL transformation running in script mode runs SQL scripts from text files.

**Query Mode**

* When you configure the SQL transformation to run in query mode, you create an active transformation.
* When an SQL transformation runs in query mode, it executes an SQL query that you define in the transformation.
* You pass strings or parameters to the query from the transformation input ports to change the query statement or the query data.

You can create the following types of SQL queries in the SQL transformation:

* **Static SQL query**. The query statement does not change, but you can use query parameters to change the data. The Integration Service prepares the query once and runs the query for all input rows.
* **Dynamic SQL query**. You can change the query statements and the data. The Integration Service prepares a query for each input row.
* When the number of output ports is more than the number of columns in the SELECT clause, the extra ports receive a NULL value.
* When the number of output ports is less than the number of columns in the SELECT clause, the Integration Service generates a row error.
* You can use string substitution instead of parameter binding in a query. However, the input ports must be string data types.

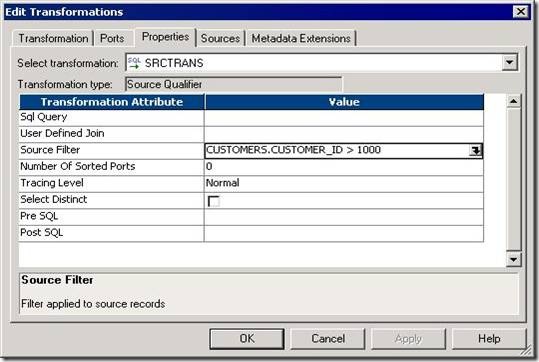
**SQL Transformation Properties**

After you create the SQL transformation, you can define ports and set attributes in the following transformation tabs:

* **Ports.**Displays the transformation ports and attributes that you create on the SQL Ports tab.
* **Properties**. SQL transformation general properties.
* **SQL Settings**. Attributes unique to the SQL transformation.
* **SQL Ports**. SQL transformation ports and attributes.

**Source Qualifier Transformation:**

* Active and Connected Transformation.
* The Source Qualifier transformation represents the rows that the Power Center Server reads when it runs a session.
* It is only transformation that is not reusable.

[](http://lh5.ggpht.com/_MbhSjEtmzI8/Ta-TAOhZ7QI/AAAAAAAAAVY/VtRKZq9mWuE/s1600-h/clip_image004%5b4%5d.jpg)

**Creating Mapping:**

1. Open folder where we want to create the mapping.
2. Click Tools -> Mapping Designer.
3. Click Mapping-> Create-> Give mapping name. Ex: m\_SQ\_example
4. Drag EMP, DEPT, Target.
5. Right Click SQ\_EMP and Select Delete from the mapping.
6. Right Click SQ\_DEPT and Select Delete from the mapping.
7. Click Transformation -> Create -> Select Source Qualifier from List -> Give Name -> Click Create
8. Select EMP and DEPT both. Click OK.
9. Link all as shown in above picture.
10. Edit SQ -> Properties Tab -> Open User defined Join -> Give Join condition EMP.DEPTNO=DEPT.DEPTNO. Click Apply -> OK
11. Mapping -> Validate
12. Repository -> Save

* Create Session and Workflow as described earlier. Run the Workflow and see the data in target table.
* Make sure to give connection information for all tables.

**SQ PROPERTIES TAB**

**1) SOURCE FILTER:**

We can enter a source filter to reduce the number of rows the Power Center Server queries.

**Note:** When we enter a source filter in the session properties, we override the customized SQL query in the Source Qualifier transformation.

Validate the mapping. Save it. Now refresh session and save the changes. Now run the workflow and see output.

**2) NUMBER OF SORTED PORTS:**

When we use sorted ports, the Power Center Server adds the ports to the ORDER BY clause in the default query.By default it is 0. If we change it to 1, then the data will be sorted by column that is at the top in SQ. Example: DEPTNO in above figure.

**Steps:**

1. In the Mapping Designer, open a Source Qualifier transformation.
2. Select the Properties tab.
3. Enter any number instead of zero for Number of Sorted ports.
4. Click Apply -> Click OK.

**3) SELECT DISTINCT:**

If we want the Power Center Server to select unique values from a source, we can use the Select Distinct option.

* Just check the option in Properties tab to enable it.

**4) PRE and POST SQL Commands**

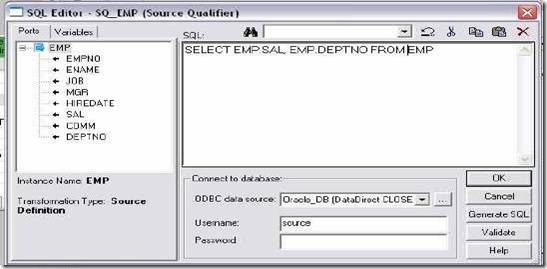
* The Power Center Server runs pre-session SQL commands against the source database before it reads the source.
* It runs post-session SQL commands against the source database after it writes to the target.
* Use a semi-colon (;) to separate multiple statements.

**5) USER DEFINED JOINS**

Entering a user-defined join is similar to entering a custom SQL query. However, we only enter the contents of the WHERE clause, not the entire query.

* We can specify equi join, left outer join and right outer join only. We Cannot specify full outer join. To use full outer join, we need to write SQL Query.

|  |  |
| --- | --- |
| **Join Type** | **Syntax** |
| Equi Join | DEPT.DEPTNO=EMP.DEPTNO |
| Left Outer Join | {EMP LEFT OUTER JOIN DEPT ON DEPT.DEPTNO=EMP.DEPTNO} |
| Right Outer Join | {EMP RIGHT OUTER JOIN DEPT ON DEPT.DEPTNO=EMP.DEPTNO} |

[](http://lh5.ggpht.com/_MbhSjEtmzI8/Ta-TE1Qe9-I/AAAAAAAAAVw/Waw5ttoySuY/s1600-h/clip_image054%5b6%5d.jpg)

**6) SQL QUERY**

We can enter an SQL statement supported by our source database. Before entering the query, connect all the input and output ports we want to use in the mapping.

**Example:** As in our case, we can’t use full outer join in user defined join, **we can write SQL query for FULL OUTER JOIN:**

SELECT DEPT.DEPTNO, DEPT.DNAME, DEPT.LOC, EMP.EMPNO, EMP.ENAME, EMP.JOB, EMP.SAL, EMP.COMM, EMP.DEPTNO FROM EMP FULL OUTER JOIN DEPT ON DEPT.DEPTNO=EMP.DEPTNO WHERE SAL>2000

**Important Points:**

* When creating a custom SQL query, the SELECT statement must list the port names in the order in which they appear in the transformation.

*Example: DEPTNO is top column; DNAME is second in our SQ   mapping.*

*So when we write SQL Query, SELECT statement have name DEPTNO first, DNAME second and so on. SELECT DEPT.DEPTNO, DEPT.DNAME*

**FILTER TRANFORMATION:**

* Active and connected transformation.

We can filter rows in a mapping with the Filter transformation. We pass all the rows from a source transformation through the Filter transformation, and then enter a Filter condition for the transformation. All ports in a Filter transformation are input/output and only rows that meet the condition pass through the Filter Transformation.

**Creating Mapping:**

1. Click Transformation -> Create -> Select Filter from list. Give name and click  Create. Now click done.
2. Pass ports from SQ\_EMP to Filter Transformation.
3. Edit Filter Transformation. Go to Properties Tab
4. Click the Value section of the Filter condition, and then click the Open button.
5. The Expression Editor appears.
6. Enter the filter condition you want to apply.
7. Click Validate to check the syntax of the conditions you entered.

**How to filter out rows with null values?**

To filter out rows containing null values or spaces, use the ISNULL and IS\_SPACES Functions to test the value of the port. For example, if we want to filter out rows that  Contain NULLs in the FIRST\_NAME port, use the following condition:

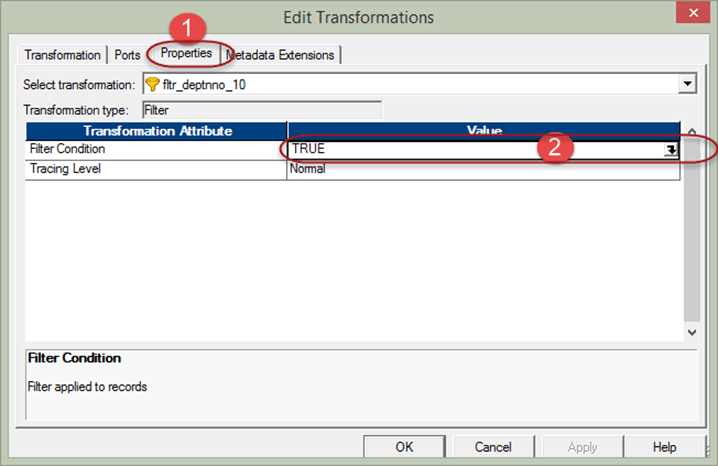
IIF (ISNULL (FIRST\_NAME), FALSE, TRUE)

This condition states that if the FIRST\_NAME port is NULL, the return value is FALSE and the row should be discarded. Otherwise, the row passes through to the next Transformation.

**Performance tuning:**

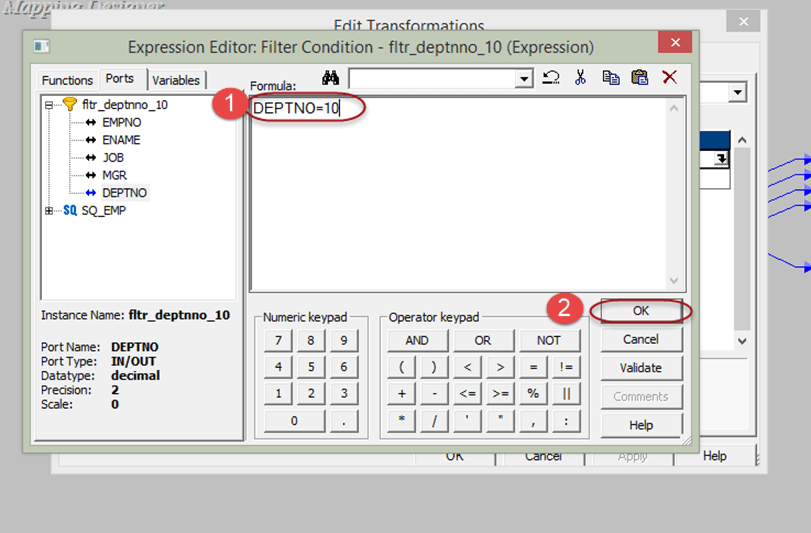
Filter transformation is used to filter off unwanted fields based on conditions we Specify.

1. Use filter transformation as close to source as possible so that unwanted data gets Eliminated sooner.
2. If elimination of unwanted data can be done by source qualifier instead of filter,Then eliminate them at Source Qualifier itself.
3. Use conditional filters and keep the filter condition simple, involving TRUE/FALSE or 1/0

[](http://cdn.guru99.com/images/informatica/083115_1041_Introductio8.png)

**Step 7** – Then in the filter condition expression editor

1. Enter filter condition – deptno=10
2. Select OK button

[](http://cdn.guru99.com/images/informatica/083115_1041_Introductio9.png)

**What is Aggregator Transformation?**

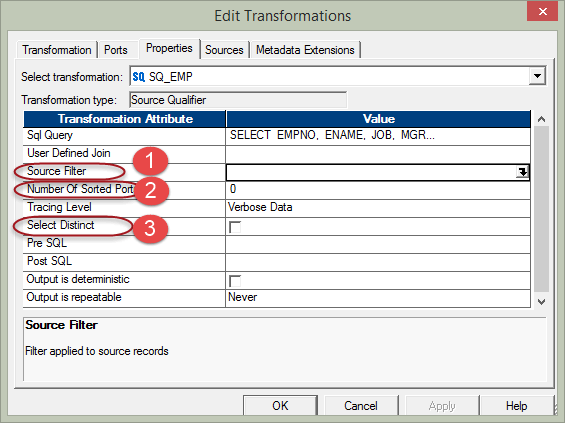
Aggregator transformation is an active transformation is used to performs aggregate calculations like sum, average, etc.For example, if you want to calculate the sum of salaries of all employees department wise, we can use the Aggregator Transformation.

The aggregate operations are performed over a group of rows, so a temporary placeholder is required to store all these records and perform the calculations.

**Configuring the aggregator transformation:**  
  
You can configure the following components in aggregator transformation

* **Aggregate Cache**: The integration service stores the group values in the index cache and row data in the data cache.
* **Aggregate Expression**: You can enter expressions in the output port or variable port.
* **Group by Port:** This tells the integration service how to create groups. You can configure input, input/output or variable ports for the group.
* **Sorted Input**: This option can be used to improve the session performance. You can use this option only when the input to the aggregator transformation in sorted on group by ports.
* **Properties of Aggregator Transformation:**  
    
  The below table illustrates the properties of aggregator transformation

|  |  |
| --- | --- |
| **Property** | **Description** |
| Cache Directory | The Integration Service creates the index and data cache files. |
| Tracing Level | Amount of detail displayed in the session log for this transformation. |
| Sorted Input | Indicates input data is already sorted by groups. Select this option only if the input to the Aggregator transformation is sorted. |
| Aggregator Data Cache Size | Default cache size is 2,000,000 bytes. Data cache stores row data. |
| Aggregator Index Cache Size | Default cache size is 1,000,000 bytes. Index cache stores group by ports data |
| Transformation Scope | Specifies how the Integration Service applies the transformation logic to incoming data |

[](http://cdn.guru99.com/images/informatica/083115_1111_AggregatorT1.png)

For this, aggregator cache memory is used. This is a temporary main memory which is allocated to the aggregator transformation to perform such operations.

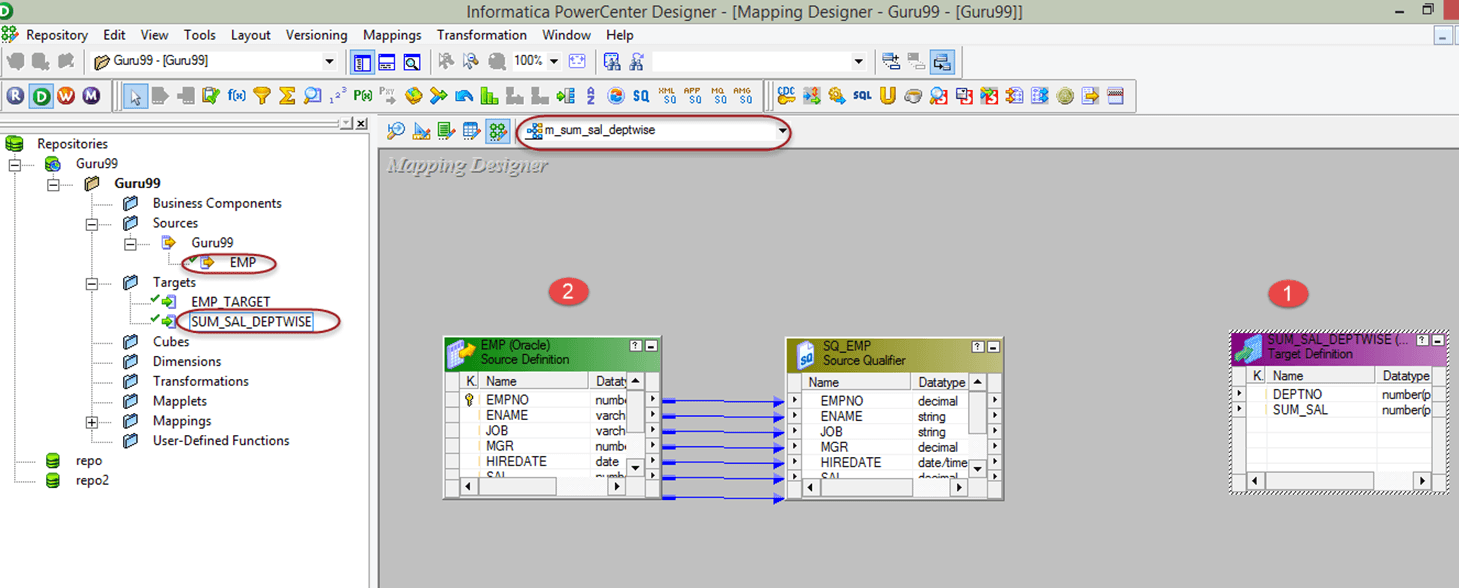
In this example, we will calculate the sum of salaries department wise. For this, we require a new column to store this sum. So, first of all, we will prepare a new column.

**Step 1** – Create a new database target table, for example, say "sum\_sal\_deptwise", using the below script. You will see the new database target table is created under Target folder in next step.

**Step 2 –** Create a New mapping "m\_ sum\_sal\_deptwise".

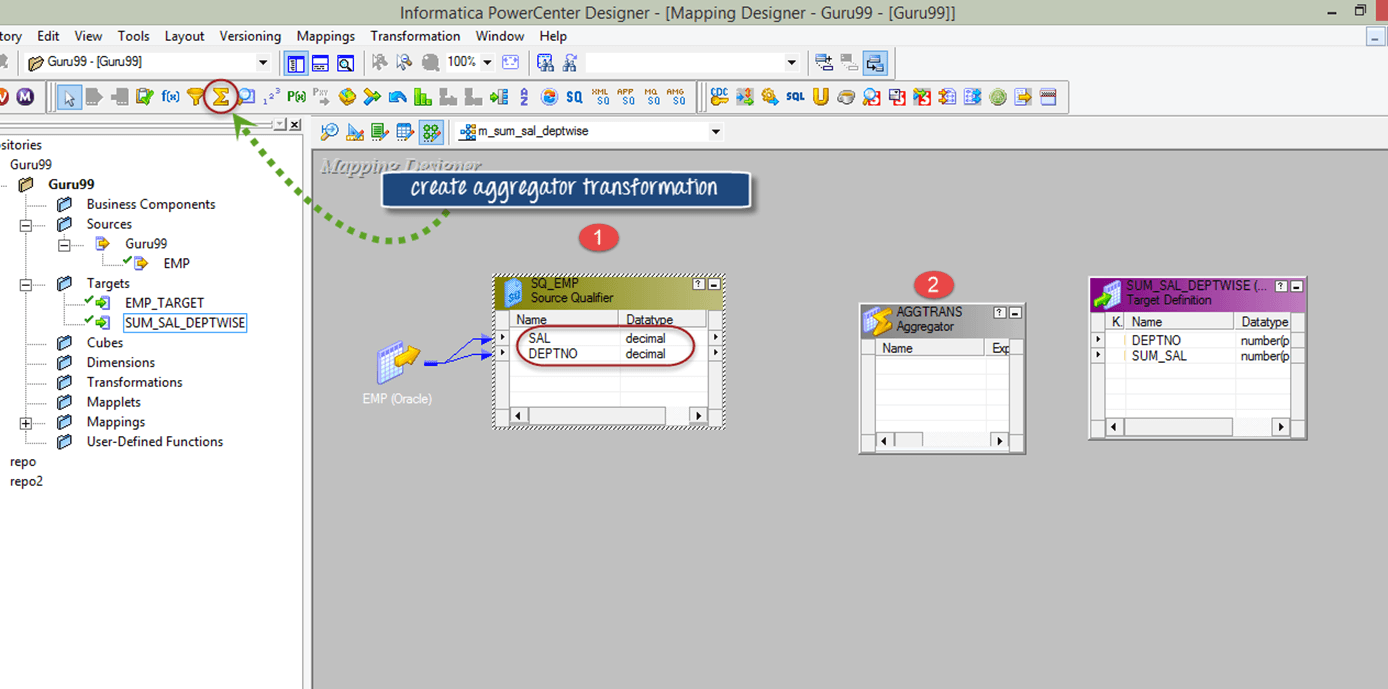
In order to create new mapping, we need source table (EMP) and target table (sum\_sal\_deptwise) both in mapping designer for that we need to

1. Import the target table "sum\_sal\_deptwise" in the mapping.
2. Import the source table "emp".

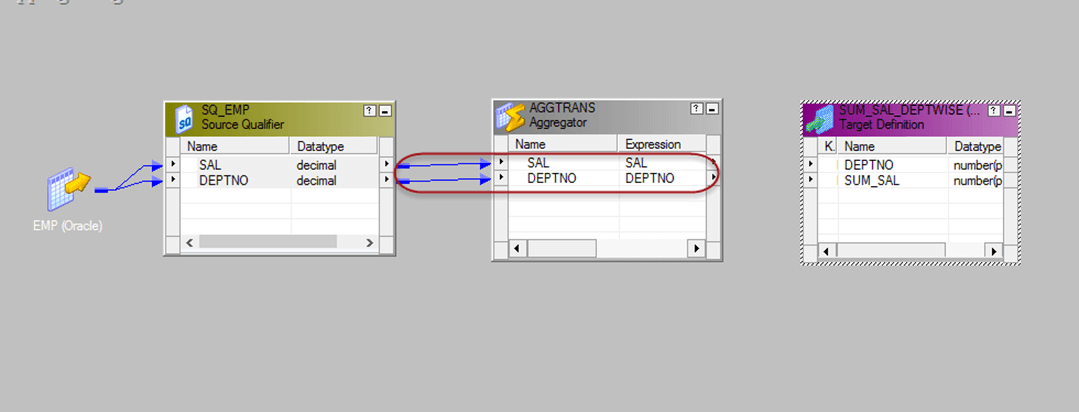
[](http://cdn.guru99.com/images/informatica/083115_1111_AggregatorT3.png)

**Step 3** – In the mapping,

1. From the Source Qualifier, delete the columns empno, ename, job, mgr, hiredate & comm so leaving only the columns deptno and sal.
2. Create a new aggregator transformation using the toolbox menu as shown in screen shot. When you click on the aggregator icon, a new aggregator transformation will be created.

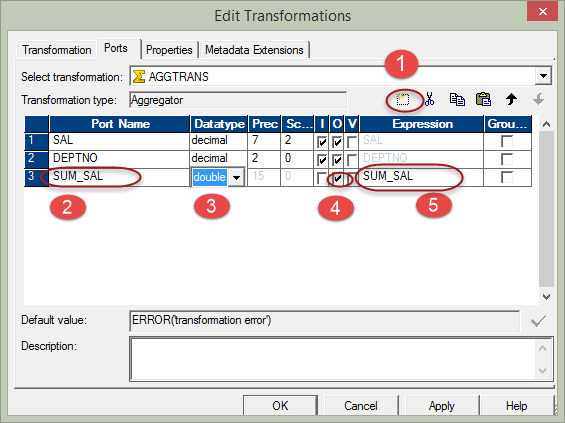
[](http://cdn.guru99.com/images/informatica/083115_1111_AggregatorT4.png)

**Step 4** - Drag and drop SAL & DEPTNO columns from source qualifier (SQ\_EMP) to the aggregator transformation

[](http://cdn.guru99.com/images/informatica/083115_1111_AggregatorT5.png)

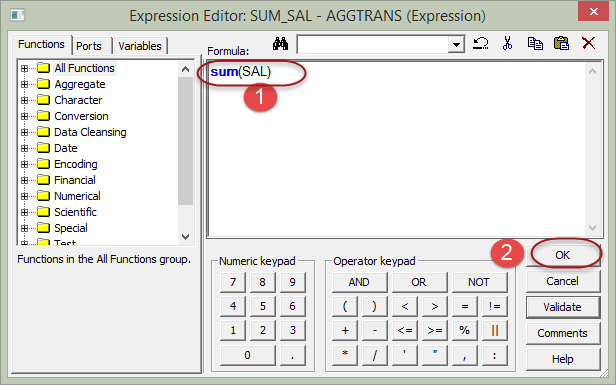
**Step 5** – Double click on the aggregator transformation to open its properties, and then

1. Add a new port in the transformation
2. Rename the port name to SUM\_SAL
3. Change the data type of this new port to double
4. Make this port as output port by selecting the checkbox of the output port.
5. Click on the expression option

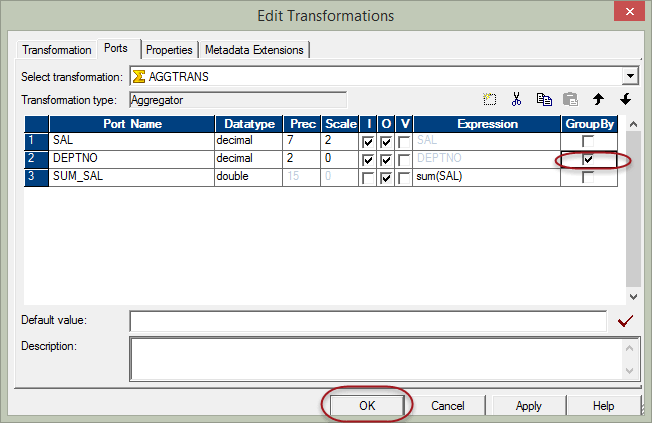
[](http://cdn.guru99.com/images/informatica/083115_1111_AggregatorT6.png)

**Step 6** – In the expression window

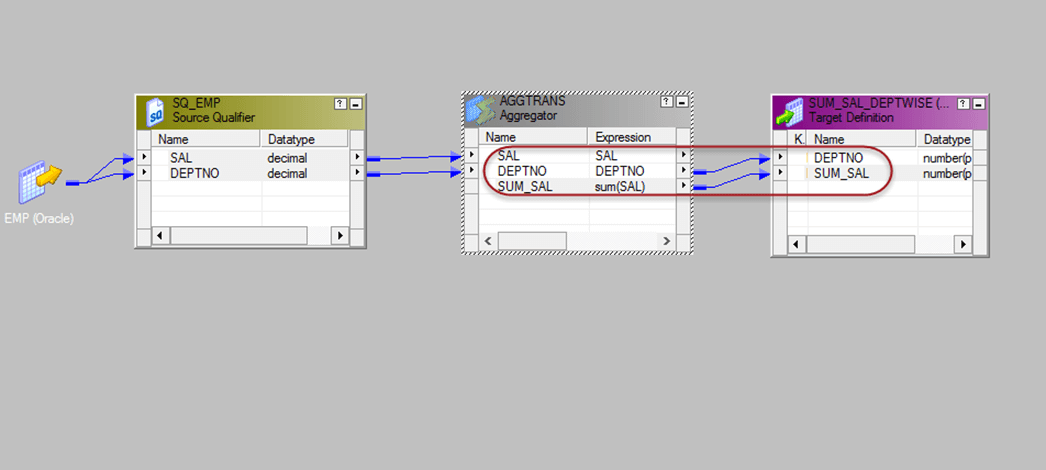
1. Add expression- sum(SAL), you have to write this expression.
2. Select Ok Button, this will bring back the edit transformation window.

[](http://cdn.guru99.com/images/informatica/083115_1111_AggregatorT7.png)

**Step 7** – In edit transformation window, select option "GroupBy" by marking the check box against the deptno column and Click Ok ( by selecting group by against the deptno, we are instructing Informatica to group salaries by deptno)

[](http://cdn.guru99.com/images/informatica/083115_1111_AggregatorT8.png)

**Step 8** – Link the deptno and sum\_sal columns from aggregator transformation to the target table

[](http://cdn.guru99.com/images/informatica/083115_1111_AggregatorT9.png)

Now save the mapping and execute it after creating a new session for this mapping. The target table would contain the sum of salaries department wise. In this way, we can use aggregator transformation to calculate aggregate results.

Components of the Aggregator Transformation

**1) Aggregate Expressions**

**Single Level Aggregate Function: MAX(SAL)**

**Nested Aggregate Function: MAX( COUNT( ITEM ))**

**Conditional Clauses**

We can use conditional clauses in the aggregate expression to reduce the number of rows used in the aggregation. The conditional clause can be any clause that evaluates to TRUE or FALSE.

* SUM( COMMISSION, COMMISSION > QUOTA )

**Non-Aggregate Functions**

* IIF( MAX( QUANTITY ) > 0, MAX( QUANTITY ), 0))

**2) Group By Ports**

* Indicates how to create groups.

**3) Using Sorted Input**

* Use to improve session performance.
* To use sorted input, we must pass data to the Aggregator transformation sorted by group by port, in ascending or descending order.
* When we use this option, we tell Aggregator that data coming to it is already sorted.
* We check the Sorted Input Option in Properties Tab of the transformation.
* If the option is checked but we are not passing sorted data to the transformation, then the session fails.

**4) Aggregator Caches**

* The Power Center Server stores data in the aggregate cache until it completes Aggregate calculations.
* It stores group values in an index cache and row data in the data cache. If the Power Center Server requires more space, it stores overflow values in cache files.

**Note:** The Power Center Server uses memory to process an Aggregator transformation with sorted ports. It does not use cache memory. We do not need to configure cache memory for Aggregator transformations that use sorted ports.

1)**Example: To calculate MAX, MIN, AVG and SUM of salary of EMP table.**

* EMP will be source table.
* Create a target table EMP\_AGG\_EXAMPLE in target designer. Table should contain DEPTNO, MAX\_SAL, MIN\_SAL, AVG\_SAL and SUM\_SAL
* Create the shortcuts in your folder.

**Creating Mapping:**

1. Open folder where we want to create the mapping.

2. Click Tools -> Mapping Designer.

3. Click Mapping-> Create-> Give mapping name. Ex: m\_agg\_example

4. Drag EMP from source in mapping.

5. Click Transformation -> Create -> Select AGGREGATOR from list. Give name and click Create. Now click done.

6. Pass SAL and DEPTNO only from SQ\_EMP to AGGREGATOR Transformation.

7. Edit AGGREGATOR Transformation. Go to Ports Tab

8. Create 4 output ports: OUT\_MAX\_SAL, OUT\_MIN\_SAL, OUT\_AVG\_SAL,

OUT\_SUM\_SAL

9. Open Expression Editor one by one for all output ports and give the

calculations. Ex: MAX(SAL), MIN(SAL), AVG(SAL),SUM(SAL)

10. Click Apply -> Ok.

11. Drag target table now.

12. Connect the output ports from Rank to target table.

13. Click Mapping -> Validate

14. Repository -> Save

* Create Session and Workflow as described earlier. Run the Workflow and see the data in target table.
* Make sure to give connection information for all tables.

**Sorter Transformation**:

* Connected and Active Transformation
* The Sorter transformation allows us to sort data.
* We can sort data in ascending or descending order according to a specified sort key.
* We can also configure the Sorter transformation for case-sensitive sorting, and specify whether the output rows should be distinct.

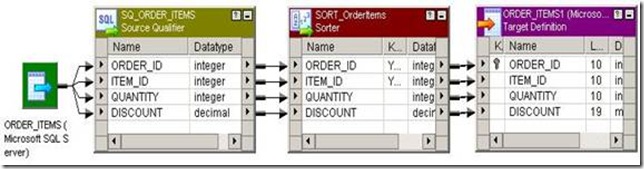
**Sorter Transformation Properties**

**1. Sorter Cache Size:**The Power Center Server uses the Sorter Cache Size property to determine the maximum amount of memory it can allocate to perform the sort operation.

**2. Case Sensitive:**The Case Sensitive property determines whether the Power Center Server considers case when sorting data. When we enable the Case Sensitive property, the Power Center Server sorts uppercase characters higher than lowercase characters.

**3. Work Directory**:Directory Power Center Server uses to create temporary files while it sorts data.

**4. Distinct:**Check this option if we want to remove duplicates. Sorter will sort data according to all the ports when it is selected.

[](http://lh4.ggpht.com/_MbhSjEtmzI8/Ta8k4dCPytI/AAAAAAAAAUo/kxx56Sjmdjw/s1600-h/clip_image002%5b4%5d.jpg)

**Example: Sorting data of EMP by ENAME**

* Source is EMP table.
* Create a target table EMP\_SORTER\_EXAMPLE in target designer. Structure same as EMP table.
* Create the shortcuts in your folder.

**Creating Mapping:**

5. Click Transformation -> Create -> Select Sorter from list. Give name and click Create. Now click done.

6. Pass ports from SQ\_EMP to Sorter Transformation.

7. Edit Sorter Transformation. Go to Ports Tab

8. Select ENAME as sort key. CHECK mark on KEY in front of ENAME.

9. Click Properties Tab and Select Properties as needed.

10. Click Apply -> Ok.

11. Drag target table now.

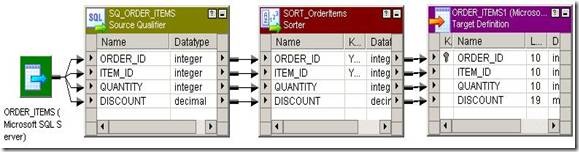
12. Connect the output ports from Sorter to target table.

13. Click Mapping -> Validate

14. Repository -> Save

* Create Session and Workflow as described earlier. Run the Workflow and see the data in target table.
* Make sure to give connection information for all tables.

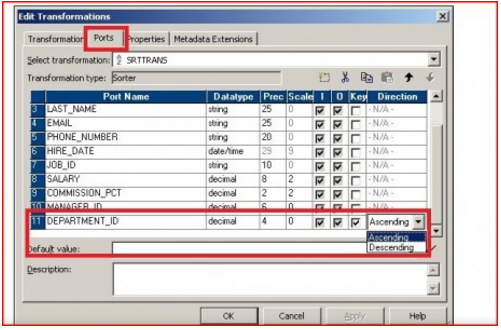
**Sample Sorter Mapping :**

[](http://lh5.ggpht.com/_MbhSjEtmzI8/Ta8k5idX96I/AAAAAAAAAUw/0dZKuBcGJ0s/s1600-h/clip_image002%5b5%5d%5b5%5d.jpg)

**Performance Tuning:**

Sorter transformation is used to sort the input data.

1. While using the sorter transformation, configure sorter cache size to be larger than the input data size.
2. Configure the sorter cache size setting to be larger than the input data size while Using sorter transformation.
3. At the sorter transformation, use hash auto keys partitioning or hash user keys Partitioning.



**Expression Transformation:**

* Passive and connected transformation.

Use the Expression transformation to calculate values in a single row before we write to the target. For example, we might need to adjust employee salaries, concatenate first and last names, or convert strings to numbers.

Use the Expression transformation to perform any non-aggregate calculations.

**Example:** Addition, Subtraction, Multiplication, Division, Concat, Uppercase conversion, lowercase conversion etc.

We can also use the Expression transformation to test conditional statements before we output the results to target tables or other transformations. Example: IF, Then, Decode

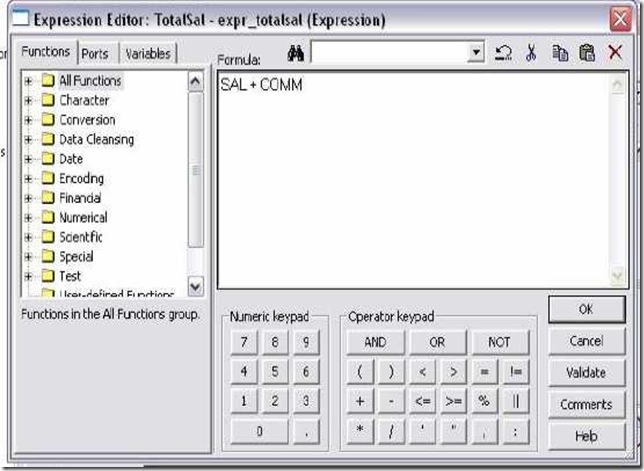
There are 3 types of ports in Expression Transformation:

* Input
* Output
* Variable: Used to store any temporary calculation.

We can enter multiple expressions in a single Expression transformation. We can create any number of output ports in the transformation.

**Example:** **Calculating Total Salary of an Employee**

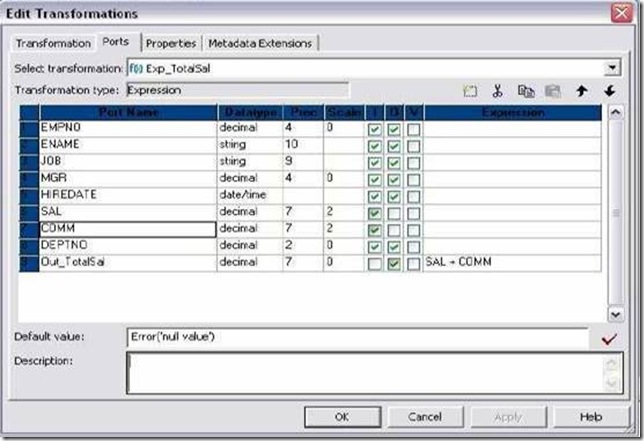
* Import the source table EMP in Shared folder. If it is already there, then don’t  import.
* In shared folder, create the target table Emp\_Total\_SAL. Keep all ports as in EMP table except Sal and Comm in target table. Add Total\_SAL port to store the calculation.
* Create the necessary shortcuts in the folder.

[](http://lh4.ggpht.com/_MbhSjEtmzI8/Ta7u2Ietc7I/AAAAAAAAARc/gnRa_C5vB6w/s1600-h/clip_image001%5b4%5d.jpg)

**Creating Mapping:**

1. Click Transformation -> Create -> Select Expression from list. Give name and click Create. Now click done.
2. Link ports from SQ\_EMP to Expression Transformation.
3. Edit Expression Transformation. As we do not want Sal and Comm in target, remove check from output port for both columns.
4. Now create a new port out\_Total\_SAL. Make it as output port only.
5. Click the small button that appears in the Expression section of the dialog box and enter the expression in the Expression Editor.
6. Enter expression SAL + COMM. You can select SAL and COMM from Ports tab in expression editor.

Create Session and Workflow as described earlier. Run the workflow and see the data in target table.

[](http://lh6.ggpht.com/_MbhSjEtmzI8/Ta7u3-ucXrI/AAAAAAAAARk/j-mqygiOupQ/s1600-h/clip_image031%5b4%5d.jpg)

As COMM is null, Total\_SAL will be null in most cases. Now open your mapping and expression transformation. Select COMM port, In Default Value give 0. Now apply changes. Validate Mapping and Save.

Refresh the session and validate workflow again. Run the workflow and see the result again.

Now use ERROR in Default value of COMM to skip rows where COMM is null.

**Syntax:** ERROR(‘Any message here’)

Similarly, we can use ABORT function to abort the session if COMM is null.

**Syntax:**ABORT(‘Any message here’)

Make sure to double click the session after doing any changes in mapping. It will prompt that mapping has changed. Click OK to refresh the mapping. Run workflow after validating and saving the workflow.

**Performance tuning :**

Expression transformation is used to perform simple calculations and also to do Source lookups.

1. Use operators instead of functions.
2. Minimize the usage of string functions.
3. If we use a complex expression multiple times in the expression transformer, then Make that expression as a variable. Then we need to use only this variable for all computations.

**JOINER TRANSFORMATION:**

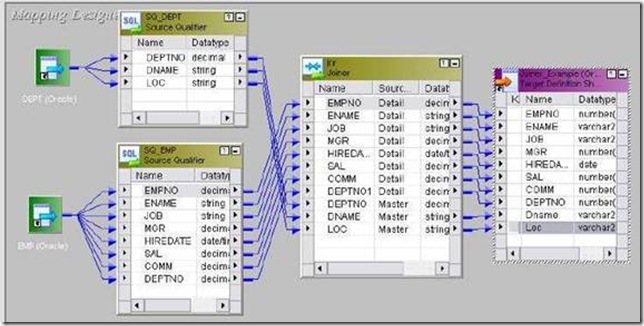
* Connected and Active Transformation
* Used to join source data from two related heterogeneous sources residing in Different locations or file systems. Or, we can join data  from the same source.
* If we need to join 3 tables, then we need 2 Joiner Transformations.
* The Joiner transformation joins two sources with at least one matching port. The Joiner transformation uses a condition that matches one or more pairs of Ports between the two sources.

**Example: To join EMP and DEPT tables.**

* EMP and DEPT will be source table.
* Create a target table JOINER\_EXAMPLE in target designer. Table should Contain all ports of EMP table plus DNAME and LOC as shown below.
* Create the shortcuts in your folder.

**Creating Mapping:**

1. Open folder where we want to create the mapping.
2. Click Tools -> Mapping Designer.
3. Click Mapping-> Create-> Give mapping name. Ex: m\_joiner\_example
4. Drag EMP, DEPT, and Target. Create Joiner Transformation. Link as shown below.

[](http://lh3.ggpht.com/_MbhSjEtmzI8/Ta7yhxA9hKI/AAAAAAAAAR8/vkBxgPp5IYc/s1600-h/clip_image002%5b4%5d.jpg)

5. Specify the join condition in Condition tab. See steps on next page.

6. Set Master in Ports tab. See steps on next page.

7. Mapping -> Validate

8. Repository -> Save.

* Create Session and Workflow as described earlier. Run the Work flow and see the data in target table.
* Make sure to give connection information for all tables.

**JOIN CONDITION:**

**Example:** DEPTNO=DEPTNO1 in above.

* Only = operator is allowed.

**Note:** The Joiner transformation does not match null values.

**MASTER and DETAIL TABLES**

In Joiner, one table is called as MASTER and other as DETAIL.

* MASTER table is always cached. We can make any table as MASTER.
* Edit Joiner Transformation -> Ports Tab -> Select M for Master table.

Table with less number of rows should be made MASTER to improve Performance.

**JOINER TRANSFORMATION PROPERTIES TAB**

* **Case-Sensitive String Comparison:** If selected, the Power Center Server uses case-sensitive string comparisons when performing joins on string columns.
* **Cache Directory:** Specifies the directory used to cache master or detail rows and the index to these rows.
* **Join Type:** Specifies the type of join: Normal, Master Outer, Detail Outer, or Full Outer.

**Tracing Level**

**Joiner Data Cache Size**

**Joiner Index Cache Size**

**Sorted Input**

**Types of Joins:**

**Note:** A normal or master outer join performs faster than a full outer or detail outer join.

**Normal Join:**With a normal join, the Power Center Server discards all rows of data from the master and detail source that do not match, based on the condition.

* All employees of 10, 20 and 30 will be there as only they are matching.

**Master Outer Join:**This join keeps all rows of data from the detail source and the matching rows from the master source. It discards the unmatched rows from the master source.

* All data of employees of 10, 20 and 30 will be there.
* There will be employees of DEPTNO 50 and corresponding DNAME and LOC Columns will be NULL.

**Detail Outer Join:**This join keeps all rows of data from the master source and the matching rows from the detail source. It discards the unmatched rows from the detail source.

* All employees of 10, 20 and 30 will be there.
* There will be one record for DEPTNO 40 and corresponding data of EMP columns will be NULL.

**Full Outer Join:**A full outer join keeps all rows of data from both the master and detail sources.

* All data of employees of 10, 20 and 30 will be there.
* There will be employees of DEPTNO 50 and corresponding DNAME and LOC Columns will be NULL.
* There will be one record for DEPTNO 40 and corresponding data of EMP Columns will be NULL.

**USING SORTED INPUT**

* Use to improve session performance.
* to use sorted input, we must pass data to the Joiner transformation sorted by the ports that are used in Join Condition.

**JOINER CACHES**

Joiner always caches the MASTER table. We cannot disable caching. It builds Index cache and Data Cache based on MASTER table.

**1) Joiner Index Cache:**

* All Columns of MASTER table used in Join condition are in JOINER INDEX CACHE.

**· Example:** DEPTNO in our mapping.

**2) Joiner Data Cache:**

* Master column not in join condition and used for output to other transformation or target table are in Data Cache.

**· Example:** DNAME and LOC in our mapping example.

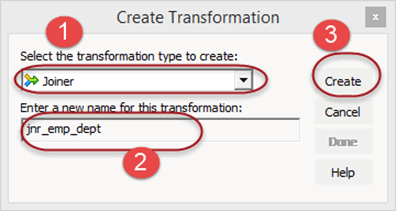
**Performance Tuning:**

* Perform joins in a database when possible.
* Join sorted data when possible.
* For a sorted Joiner transformation, designate as the master source the source with fewer duplicate key values.
* Joiner can't be used in following conditions:

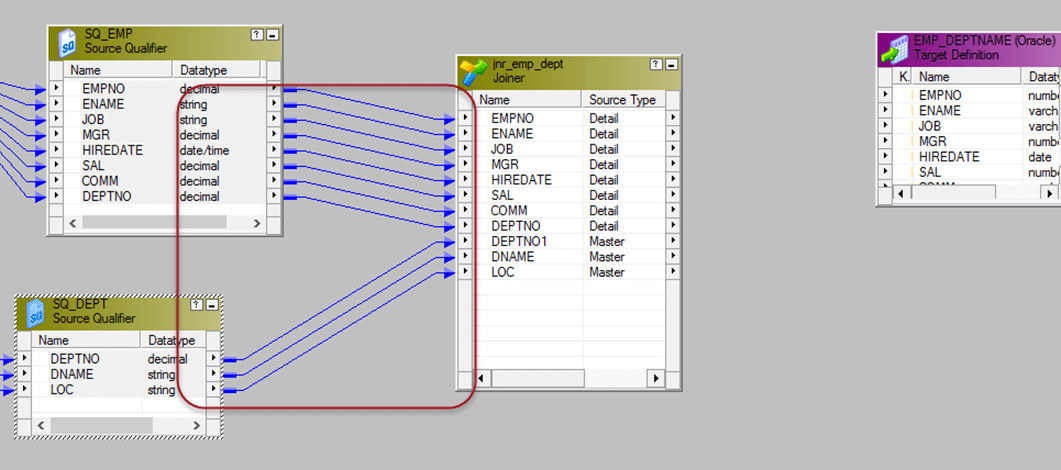
1. Either input pipeline contains an Update Strategy transformation.
2. We connect a Sequence Generator transformation directly before the Joiner transformation.

**Step 3** – From the transformation menu, select create option.

1. Select joiner transformation
2. Enter transformation name "jnr\_emp\_dept"
3. Select create option

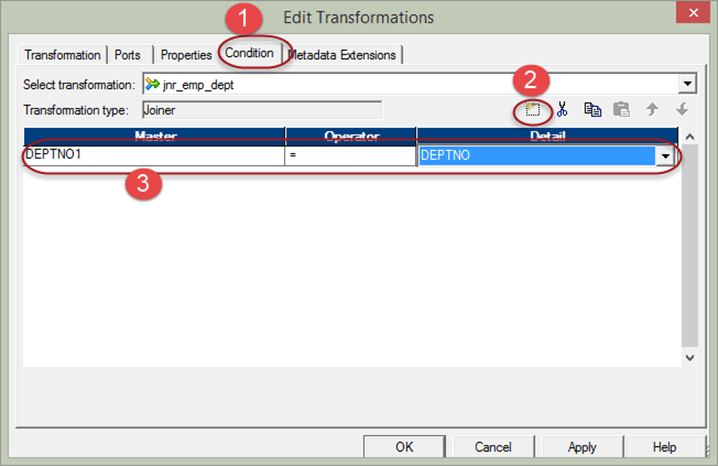
[](http://cdn.guru99.com/images/informatica/083115_1139_Joinertrans3.png)

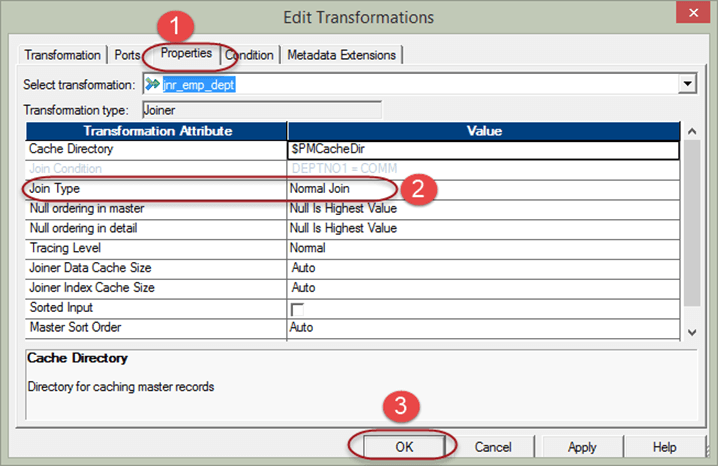
**Step 4** – Drag and drop all the columns from both the source qualifiers to the joiner transformation

[](http://cdn.guru99.com/images/informatica/083115_1139_Joinertrans4.png)

**Step 5** - Double click on the joiner transformation, then in the edit transformation window

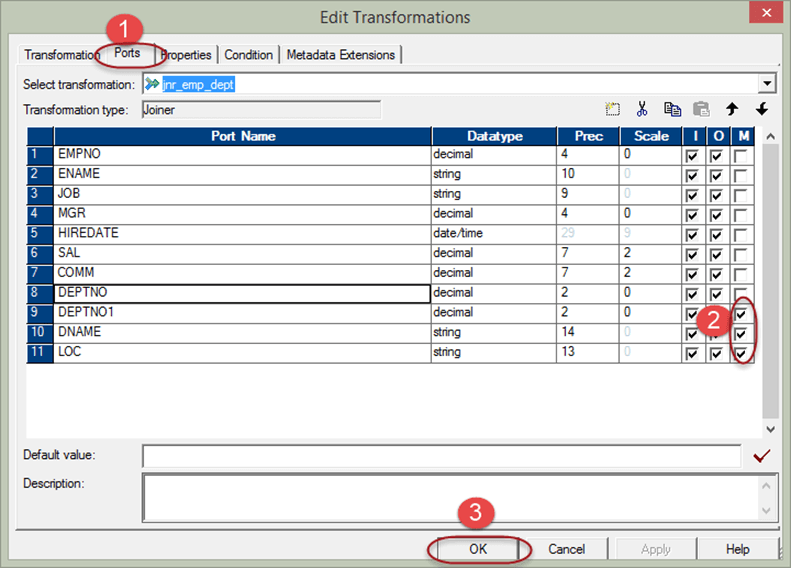
1. Select condition tab
2. Click on add new condition icon
3. Select deptno in master and detail columns list

[](http://cdn.guru99.com/images/informatica/083115_1139_Joinertrans5.png)

[](http://cdn.guru99.com/images/informatica/083115_1139_Joinertrans6.png)

For performance optimization, we assign the master source to the source table pipeline which is having less no of records. To perform this task –

**Step 7** –Double click on the joiner transformation to open edit properties window, and then

[](http://cdn.guru99.com/images/informatica/083115_1139_Joinertrans7.png)

**Step 8** – Link the relevant columns from joiner transformation to target table

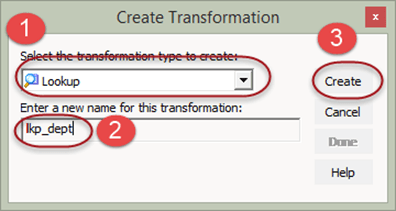
**LOOKUP TRANFORMATION:**

* Passive Transformation
* Can be Connected or Unconnected. Dynamic lookup is connected.
* Use a Lookup transformation in a mapping to look up data in a flat file or a relational table, view, or synonym.
* We can import a lookup definition from any flat file or relational database to which both the PowerCenter Client and Server can connect.
* We can use multiple Lookup transformations in a mapping.

**We can use the Lookup transformation to perform following:**

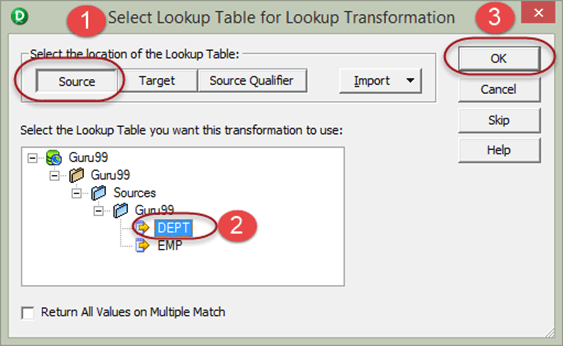
* **Get a related value:** EMP has DEPTNO but DNAME is not there. We use Lookup to get DNAME from DEPT table based on Lookup Condition.
* **Perform a calculation:** We want only those Employees who’s SAL > Average (SAL). We will write Lookup Override query.
* **Update slowly changing dimension tables:** Most important use. We can use a Lookup transformation to determine whether rows already exist in the target.

**Step 2** – Create a new transformation using transformation menu then

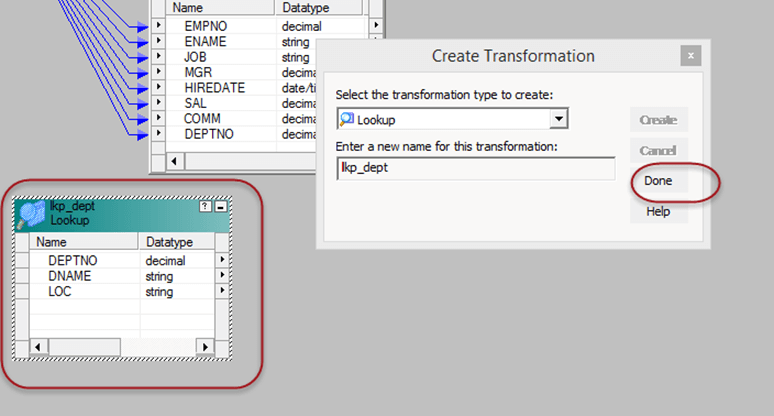
[](http://cdn.guru99.com/images/informatica/083115_1246_LookupandRe2.png)

**Step 3** – This will open lookup table window, in this window

1. Select source button
2. Select DEPT table
3. Select Ok Button

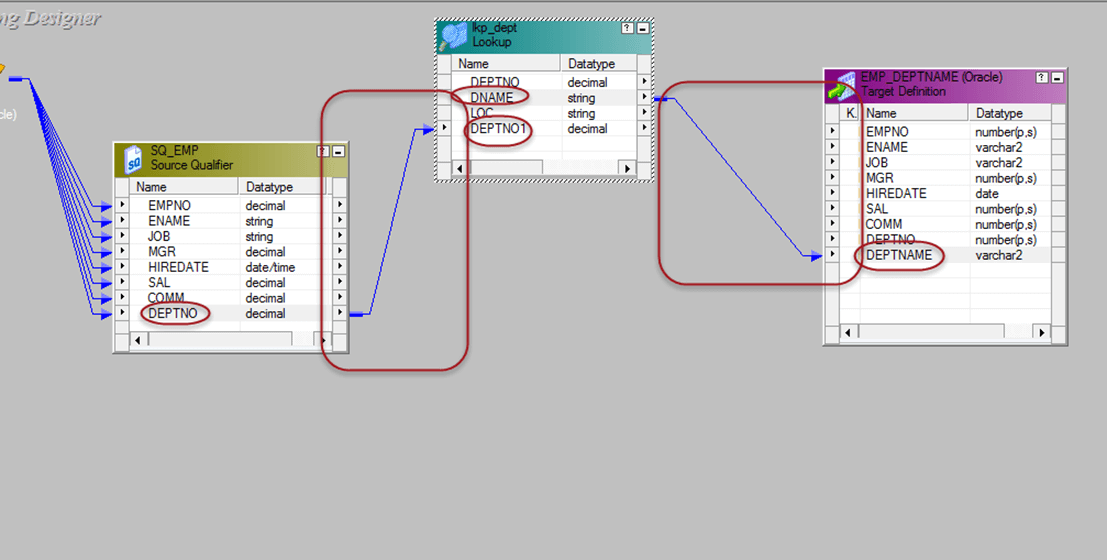
[](http://cdn.guru99.com/images/informatica/083115_1246_LookupandRe3.png)

**Step 4** - Lookup transformation will be created with the columns of DEPT table, now select done button

[](http://cdn.guru99.com/images/informatica/083115_1246_LookupandRe4.png)

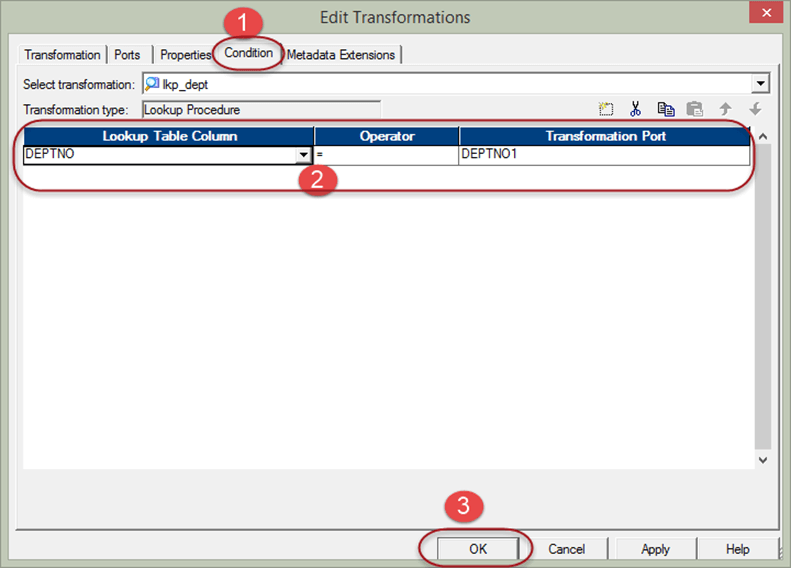
**Step 5** - Drag and drop DEPTNO column from source qualifier to the lookup transformation, this will create a new column DEPTNO1 in lookup transformation. Then link the DNAME column from lookup transformation to the target table.

The lookup transformation will lookup and return department name based upon the DEPTNO1 value.

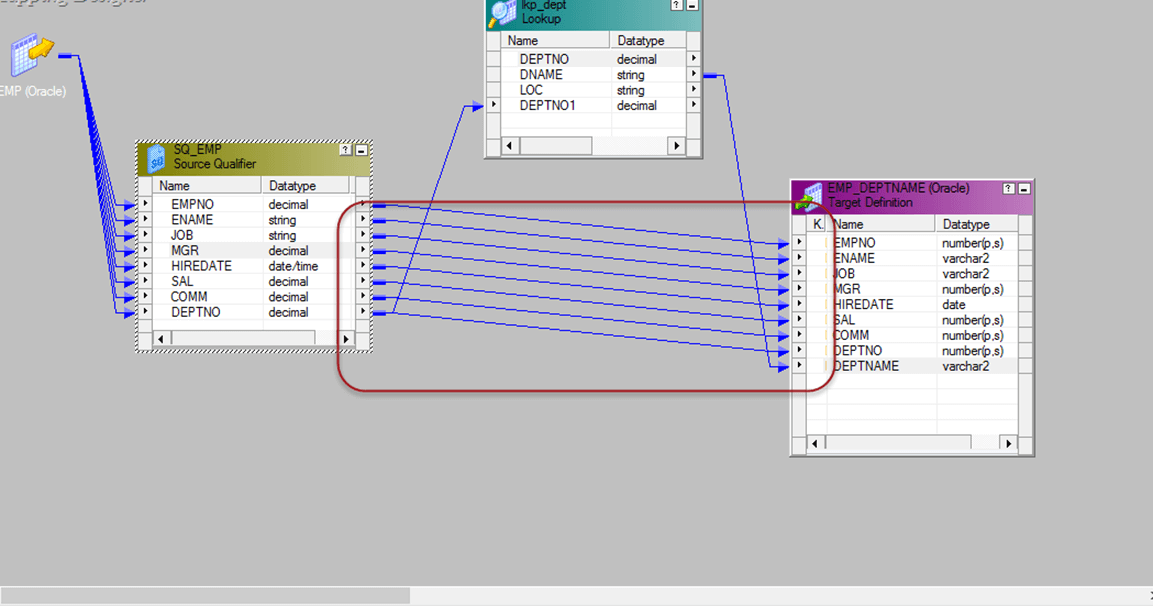
[](http://cdn.guru99.com/images/informatica/083115_1246_LookupandRe5.png)

**Step 6** – Double click on the lookup transformation. Then in the edit transformation window

1. Select condition tab
2. Set the condition column to DEPTNO = DEPTNO1
3. Select Ok Button

[](http://cdn.guru99.com/images/informatica/083115_1246_LookupandRe6.png)

**Step 7** – Link rest of the columns from source qualifier to the target table

[](http://cdn.guru99.com/images/informatica/083115_1246_LookupandRe7.png)

Now, save the mapping and execute it after creating the session and workflow. This mapping will fetch the department names using lookup transformation.

The lookup transformation is set to lookup on dept table. And the joining condition is set based on dept number.

**Reusable Transformation**

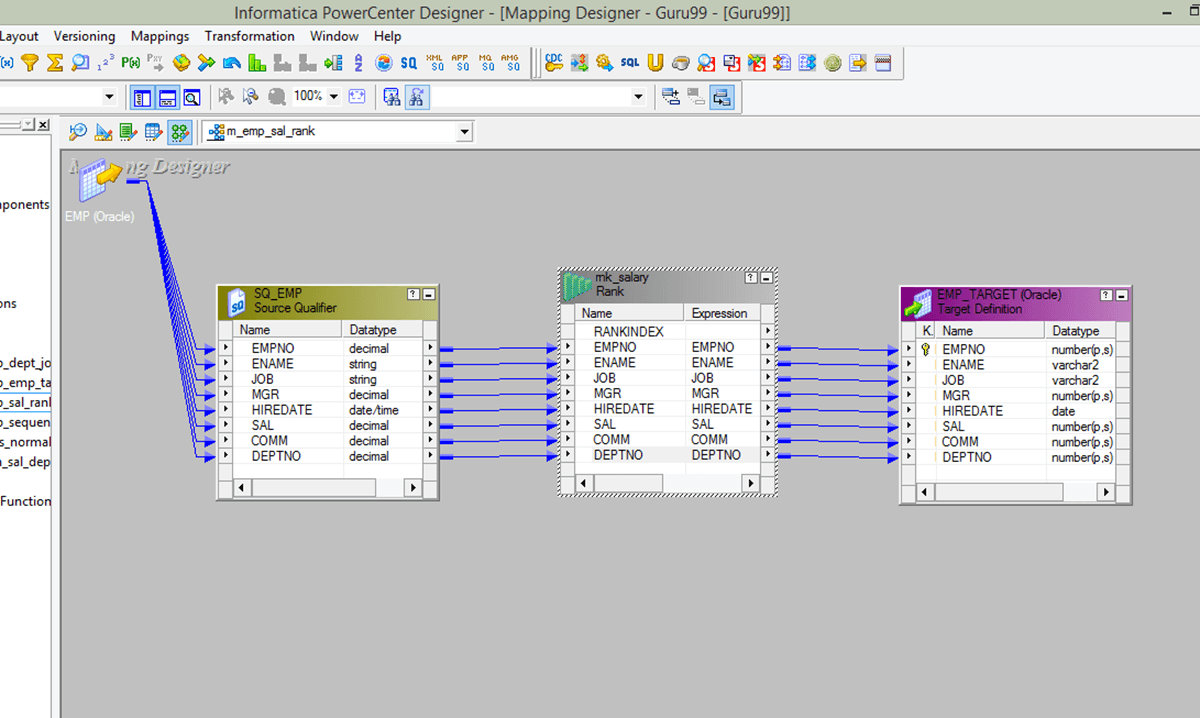
A normal transformation is an object that belongs to a mapping and can be used inside that mapping only. However, by making a transformation reusable it can be re-used inside several mappings.

For example, a lookup transformation which fetches employee details based on employee number can be used at multiple mappings wherever employee details are required.

By using reusable transformation, it reduces the overwork of creating same functionality again.

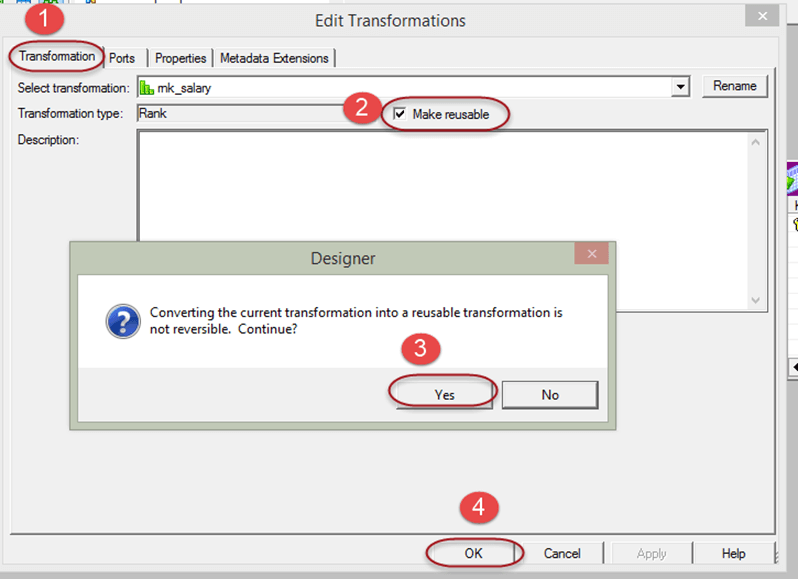
To make a transformation reusable –

**Step1** – Open the mapping which is having the transformation, here we are making the rank transformation reusable.

[](http://cdn.guru99.com/images/informatica/083115_1246_LookupandRe8.png)

**Step 2** – Double click on the transformation to open edit transformation window. Then

1. Select Transformation tab in the window
2. Select the check box to make transformation reusable
3. Select yes in the confirmation window
4. Select OK in the transformation properties window.

[](http://cdn.guru99.com/images/informatica/083115_1246_LookupandRe9.png)

**Connected and Unconnected Lookup**

|  |  |
| --- | --- |
| **Connected Lookup** | **Unconnected Lookup** |
| Receives input values directly from the pipeline. | Receives input values from the result of a :LKP expression in another transformation. |
| We can use a dynamic or static cache. | We can use a static cache. |
| Cache includes all lookup columns used in the mapping. | Cache includes all lookup/output ports in the lookup condition and the lookup/return port. |
| If there is no match for the lookup condition, the Power Center Server returns the default value for all output ports. | If there is no match for the lookup condition, the Power Center Server returns NULL. |
| If there is a match for the lookup condition, the Power Center Server returns the result of the lookup condition for all lookup/output ports. | If there is a match for the lookup condition,the Power Center Server returns the result of the lookup condition into the return port. |
| Pass multiple output values to another transformation. | Pass one output value to another transformation. |
| Supports user-defined default values | Does not support user-defined default values. |

**2. Ports:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ports** | **Lookup**  **Type** | **Number**  **Needed** | **Description** |
| I | Connected  Unconnected | Minimum 1 | Input port to Lookup. Usually ports used for Join condition are Input ports. |
| O | Connected  Unconnected | Minimum 1 | Ports going to another transformation from Lookup. |
| L | Connected  Unconnected | Minimum 1 | Lookup port. The Designer automatically Designates each column in the lookup source as a lookup (L) and output port (O). |
| R | Unconnected | 1 Only | Return port. Use only in unconnected Lookup t/f only. |

**3. Properties Tab**

|  |  |  |
| --- | --- | --- |
| **Options** | **Lookup Type** | **Description** |
| Lookup SQL Override | Relational | Overrides the default SQL statement to query the lookup table. |
| Lookup Table Name | Relational | Specifies the name of the table from which the transformation looks up and caches values. |
| Lookup Caching Enabled | Flat File, Relational | Indicates whether the Power Center Server caches lookup values during the session. |
| Lookup Policy on Multiple Match | Flat File, Relational | Determines what happens when the Lookup transformation finds multiple rows that match the lookup condition. Options: Use First Value or Use Last Value or Use Any Value or Report Error |
| Lookup Condition | Flat File, Relational | Displays the lookup condition you set in the Condition tab. |
| Connection Information | Relational | Specifies the database containing the lookup table. |
| Source Type | Flat File, Relational | Lookup is from a database or flat file. |
| Lookup Cache Directory Name | Flat File, Relational | Location where cache is build. |
| Lookup Cache Persistent | Flat File, Relational | Whether to use Persistent Cache or not. |
| Dynamic Lookup Cache | Flat File, Relational | Whether to use Dynamic Cache or not. |
| Recache From Lookup Source | Flat File, Relational | To rebuild cache if cache source changes and we are using Persistent Cache. |
| Insert Else Update | Relational | Use only with dynamic caching enabled. Applies to rows entering the Lookup transformation with the row type of insert. |
| Lookup Data Cache Size | Flat File, Relational | Data Cache Size |
| Lookup Index Cache Size | Flat File, Relational | Index Cache Size |
| Cache File Name Prefix | Flat File, Relational | Use only with persistent lookup cache. Specifies the file name prefix to use with persistent lookup cache files. |

**Creating Mapping:**

1. Open folder where we want to create the mapping.

2. Click Tools -> Mapping Designer.

3. Click Mapping-> Create-> Give name. Ex: m\_CONN\_LOOKUP\_EXAMPLE

4. Drag EMP and Target table.

5. Connect all fields from SQ\_EMP to target except DNAME and LOC.

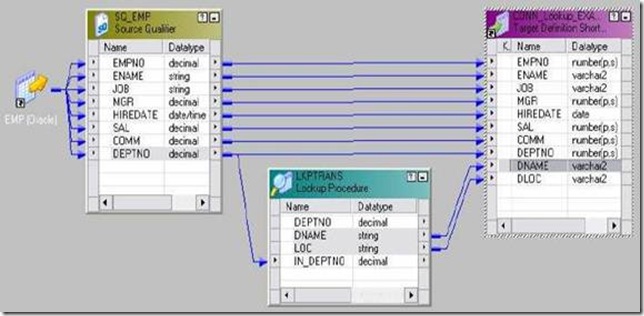
6. Transformation-> Create -> Select LOOKUP from list. Give name and click

Create.

7. The Following screen is displayed.

8. As DEPT is the Source definition, click Source and then Select DEPT.

9. Click Ok.

[](http://lh4.ggpht.com/_MbhSjEtmzI8/Ta8dCemLNDI/AAAAAAAAATU/s5ndyRJQ3ys/s1600-h/clip_image135%5b4%5d.jpg)

10. Now Pass DEPTNO from SQ\_EMP to this Lookup. DEPTNO from SQ\_EMP will be named as DEPTNO1. Edit Lookup and rename it to IN\_DEPTNO in ports tab.

11. Now go to CONDITION tab and add CONDITION.

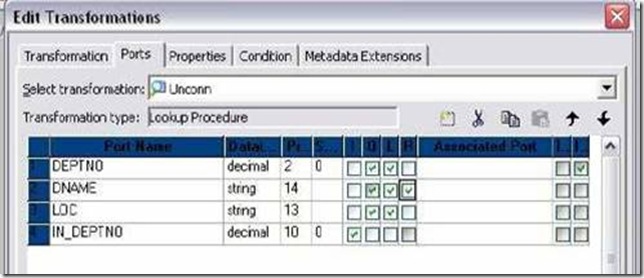
DEPTNO = IN\_DEPTNO and Click Apply and then OK.

Link the mapping as shown below:

12. We are not passing IN\_DEPTNO and DEPTNO to any other transformation from LOOKUP; we can edit the lookup transformation and remove the OUTPUT check from them.

13. Mapping -> Validate

14. Repository -> Save

We use Connected Lookup when we need to return more than one column from Lookup table.There is no use of Return Port in Connected Lookup.[](http://lh3.ggpht.com/_MbhSjEtmzI8/Ta8dDmARLHI/AAAAAAAAATc/Hz5coTB-5zQ/s1600-h/clip_image137%5b4%5d.jpg)

**4. Unconnected Lookup Transformation**

An unconnected Lookup transformation is separate from the pipeline in the mapping. We write an expression using the :LKP reference qualifier to call the lookup within another transformation.

**Creating Mapping:**

1. Open folder where we want to create the mapping.

2. Click Tools -> Mapping Designer.

3. Click Mapping-> Create-> Give name. Ex: m\_UNCONN\_LOOKUP\_EXAMPLE

4. Drag EMP and Target table.

5. Now Transformation-> Create -> Select EXPRESSION from list. Give name

and click Create. Then Click Done.

6. Pass all ports from SQ\_EMP to EXPRESSION transformation.

7. Connect all fields from EXPRESSION to target except DNAME.

8. Transformation-> Create -> Select LOOKUP from list. Give name and click

Create.

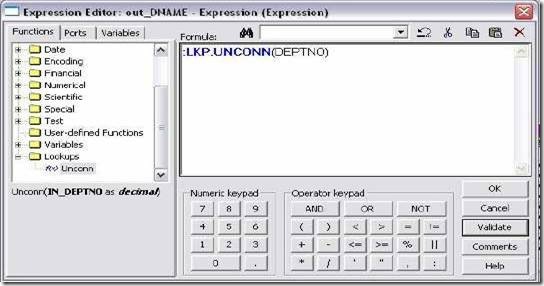
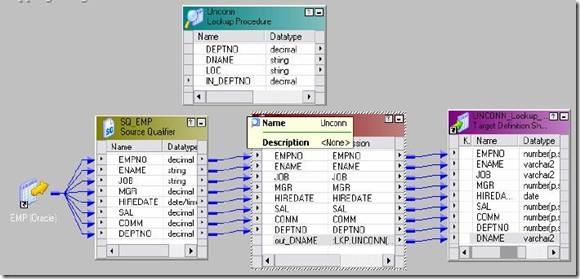
9. Follow the steps as in Connected above to create Lookup on DEPT table.

10. Click Ok.

11. Now Edit the Lookup Transformation. Go to Ports tab.

12. As DEPTNO is common in source and Lookup, create a port IN\_DEPTNO

ports tab. Make it Input port only and Give Datatype same as DEPTNO.

13. Designate DNAME as Return Port. Check on R to make it.[](http://lh3.ggpht.com/_MbhSjEtmzI8/Ta8dEyaAcxI/AAAAAAAAATk/UQiUJuuEUcA/s1600-h/clip_image139%5b4%5d.jpg)[](http://lh4.ggpht.com/_MbhSjEtmzI8/Ta8dGopK5NI/AAAAAAAAATs/aLS5ZNIfY1g/s1600-h/clip_image141%5b4%5d.jpg)

14. Now add a condition in Condition Tab.

DEPTNO = IN\_DEPTNO and Click Apply and then OK.

15. Now we need to call this Lookup from Expression Transformation.

16. Edit Expression t/f and create a new output port out\_DNAME of data type as DNAME. Open the Expression editor and call Lookup as given below:

We double click Unconn in bottom of Functions tab and as we need only

DEPTNO, we pass only DEPTNO as input.

17. Validate the call in Expression editor and Click OK.

18. Mapping -> Validate

19. Repository Save.

**5. Lookup Caches**

**Lookup Cache Files**

1. **Lookup Index Cache:**

* Stores data for the columns used in the lookup condition.

2. **Lookup Data Cache:**

* For a connected Lookup transformation, stores data for the connected output ports, not including ports used in the lookup condition.
* For an unconnected Lookup transformation, stores data from the return port.

**Update Strategy Transformation**

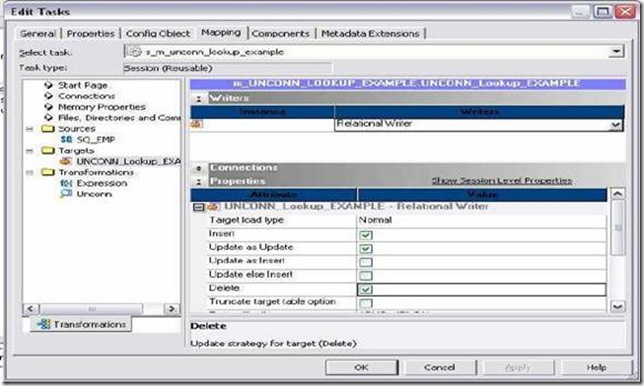
Active and Connected TransformationTill now, we have only inserted rows in our target tables. What if we want to update, delete or reject rows coming from source based on some condition?

**Example:** If Address of a CUSTOMER changes, we can update the old address or keep both old and new address. One row is for old and one for new. This way we maintain the historical data.

Update Strategy is used with Lookup Transformation. In DWH, we create a Lookup on target table to determine whether a row already exists or not. Then we insert, update, delete or reject the source record as per business need.

In Power Center, we set the update strategy at two different levels:

1. Within a session
2. Within a Mapping
3. **Update Strategy within a session:**When we configure a session, we can instruct the IS to either treat all rows in the same way or use instructions coded into the session mapping to flag rows for different database operations.
4. **Session Configuration:**Edit Session -> Properties -> Treat Source Rows as: (Insert, Update, Delete, and Data Driven). Insert is default. Specifying Operations for Individual Target Tables:

[](http://lh3.ggpht.com/_MbhSjEtmzI8/Ta-cYsL54oI/AAAAAAAAAXg/OOrxdNJpbGA/s1600-h/clip_image002%5b4%5d.jpg)

You can set the following update strategy options:

**Insert:** Select this option to insert a row into a target table.

**Delete:** Select this option to delete a row from a table.

**Update:** We have the following options in this situation:

**Truncate table:** Select this option to truncate the target table before loading data.

**2. Flagging Rows within a Mapping**

Within a mapping, we use the Update Strategy transformation to flag rows for insert, delete, update, or reject.

|  |  |  |
| --- | --- | --- |
| **Operation** | **Constant** | **Numeric Value** |
| INSERT | DD\_INSERT | 0 |
| UPDATE | DD\_UPDATE | 1 |
| DELETE | DD\_DELETE | 2 |
| REJECT | DD\_REJECT | 3 |

**Update Strategy Expressions:**

Frequently, the update strategy expression uses the IIF or DECODE function from the transformation language to test each row to see if it meets a particular condition.

IIF( ( ENTRY\_DATE > APPLY\_DATE), DD\_REJECT, DD\_UPDATE )

Or

IIF( ( ENTRY\_DATE > APPLY\_DATE), 3, 2 )

* The above expression is written in Properties Tab of Update Strategy T/f.
* DD means DATA DRIVEN

**Forwarding Rejected Rows:**We can configure the Update Strategy transformation to either pass rejected rows to the next transformation or drop them

**Performance tuning:**

1. Use Update Strategy transformation as less as possible in the mapping.
2. Do not use update strategy transformation if we just want to insert into target table, instead use direct mapping, direct filtering etc.
3. For updating or deleting rows from the target table we can use Update Strategy transformation itself.

**RANK TRANSFORMATION**

* Active and connected transformation

The Rank transformation allows us to select only the top or bottom rank of data. It Allows us to select a group of top or bottom values, not just one value

**Rank Transformation Properties :**

* Cache Directory where cache will be made.
* Top/Bottom Rank as per need
* Number of Ranks Ex: 1, 2 or any number
* Case Sensitive Comparison can be checked if needed
* Rank Data Cache Size can be set
* Rank Index Cache Size can be set

**Ports in a Rank Transformation :**

|  |  |  |
| --- | --- | --- |
| **Ports** | **Number Required** | **Description** |
| I | 1 Minimum | Port to receive data from another transformation. |
| O | 1 Minimum | Port we want to pass to other transformation. |
| V | not needed | can use to store values or calculations to use in an expression. |
| R | Only 1 | Rank port. Rank is calculated according to it. The Rank port is an input/output port. We must link the Rank port to another transformation. Example: Total Salary |

**Rank Index**

The Designer automatically creates a RANKINDEX port for each Rank transformation. The Power Center Server uses the Rank Index port to store the ranking position for Each row in a group.

**For example,** if we create a Rank transformation that ranks the top five salaried employees, the rank index numbers the employees from 1 to 5.

* The RANKINDEX is an output port only.
* We can pass the rank index to another transformation in the mapping or directly to a target.
* We cannot delete or edit it.

**Defining Groups**

Rank transformation allows us to group information. For example: If we want to select the top 3 salaried employees of each Department, we can define a group for Department.

1) **Example: Finding Top 5 Salaried Employees**

* EMP will be source table.
* Create a target table EMP\_RANK\_EXAMPLE in target designer. Structure should be same as EMP table. Just add one more port Rank\_Index to store RANK INDEX.
* Create the shortcuts in your folder.

**Creating Mapping:**

1. Open folder where we want to create the mapping.
2. Click Tools -> Mapping Designer.
3. Click Mapping-> Create-> Give mapping name. Ex: m\_rank\_example
4. Drag EMP from source in mapping.
5. Create an EXPRESSION transformation to calculate TOTAL\_SAL.
6. Click Transformation -> Create -> Select RANK from list. Give name and click Create. Now click done.
7. Pass ports from Expression to Rank Transformation.
8. Edit Rank Transformation. Go to Ports Tab
9. Select TOTAL\_SAL as rank port. Check R type in front of TOTAL\_SAL.
10. Click Properties Tab and Select Properties as needed.
11. Top in Top/Bottom and Number of Ranks as 5.
12. Click Apply -> Ok.
13. Drag target table now.
14. Connect the output ports from Rank to target table.
15. Click Mapping -> Validate
16. Repository -> Save

* Create Session and Workflow as described earlier. Run the Workflow and see the data in target table.
* Make sure to give connection information for all tables.

**2) Example: Finding Top 2 Salaried Employees for every DEPARTMENT**

* Open the mapping made above. Edit Rank Transformation.
* Go to Ports Tab. Select Group By for DEPTNO.
* Go to Properties tab. Set Number of Ranks as 2.
* Click Apply -> Ok.
* Mapping -> Validate and Repository Save.

1)**RANK INDEX CACHE:**

The index cache holds group information from the group by ports. If we are Using Group By on DEPTNO, then this cache stores values 10, 20, 30 etc.

* All Group By Columns are in RANK INDEX CACHE. Ex. DEPTNO

2) **RANK DATA CACHE:**

It holds row data until the Power Center Server completes the ranking and is Generally larger than the index cache. To reduce the data cache size, connect Only the necessary input/output ports to subsequent transformations.

* All Variable ports if there, Rank Port, All ports going out from RANK Transformations are stored in RANK DATA CACHE.
* **Example:** All ports except DEPTNO In our mapping example.

**UNION TRANSFORMATION**

  Active and Connected transformation.Union transformation is a multiple input group transformation that you can use to merge data from multiple pipelines or pipeline branches into one pipeline branch. It merges data from multiple sources similar to the **UNION ALL** SQL statement to Combine the results from two or more SQL statements.

**Union Transformation Rules and Guidelines**

* we can create multiple input groups, but only one output group.
* The Union transformation does not remove duplicate rows. To remove Duplicate rows, we must add another transformation such as a Router or Filter Transformation.
* we cannot use a Sequence Generator or Update Strategy transformation upstream from a Union transformation.

**Union Transformation Components**

**Transformation tab:** We can rename the transformation and add a description.

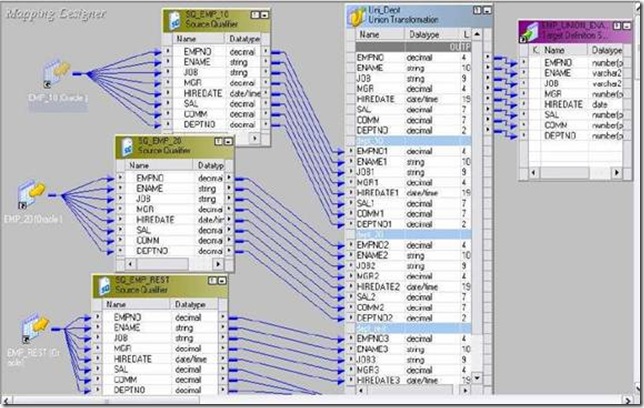
**Properties tab:** We can specify the tracing level.

**Groups tab:** We can create and delete input groups. The Designer displays groups we create on the Ports tab.

**Group Ports tab:** We can create and delete ports for the input groups. The Designer displays ports we create on the Ports tab.

**Example: to combine data of tables EMP\_10, EMP\_20 and EMP\_REST**

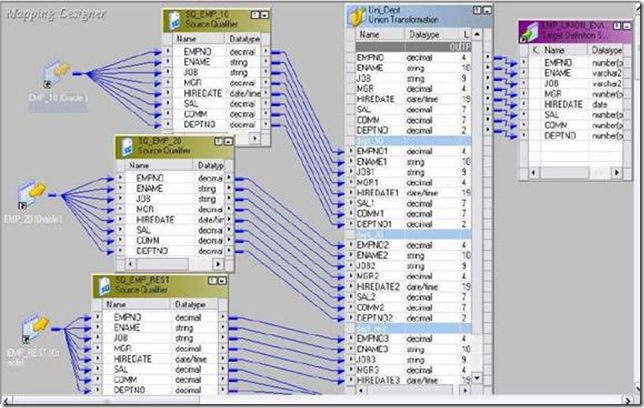
* Import tables EMP\_10, EMP\_20 and EMP\_REST in shared folder in Sources.
* Create a target table EMP\_UNION\_EXAMPLE in target designer. Structure should be same EMP table.
* Create the shortcuts in your folder.

[](http://lh6.ggpht.com/_MbhSjEtmzI8/Ta-Zs2ZndxI/AAAAAAAAAXQ/n9vWEtve7HM/s1600-h/clip_image002%5b4%5d.jpg)

**Creating Mapping:**

1. Open folder where we want to create the mapping.
2. Click Tools -> Mapping Designer.
3. Click Mapping-> Create-> Give mapping name. Ex: m\_union\_example
4. Drag EMP\_10, EMP\_20 and EMP\_REST from source in mapping.
5. Click Transformation -> Create -> Select Union from list. Give name and click Create. Now click done.
6. Pass ports from SQ\_EMP\_10 to Union Transformation.
7. Edit Union Transformation. Go to Groups Tab
8. One group will be already there as we dragged ports from SQ\_DEPT\_10 to Union Transformation.
9. As we have 3 source tables, we 3 need 3 input groups. Click add button to add 2 more groups. See Sample Mapping
10. We can also modify ports in ports tab.
11. Click Apply -> Ok.
12. Drag target table now.
13. Connect the output ports from Union to target table.
14. Click Mapping -> Validate
15. Repository -> Save

* Create Session and Workflow as described earlier. Run the Workflow and see the data in target table.
* Make sure to give connection information for all 3 source Tables.

[](http://lh6.ggpht.com/_MbhSjEtmzI8/Ta-ZuP_aprI/AAAAAAAAAXY/h5pQB6d1fMg/s1600-h/clip_image002%5b5%5d%5b3%5d.jpg)

**NORMALIZER TRANSFORMATION**

Active and Connected Transformation.

* The Normalizer transformation normalizes records from COBOL and relational sources, allowing us to organize the data.
* Use a Normalizer transformation instead of the Source Qualifier transformation when we normalize a COBOL source.
* We can also use the Normalizer transformation with relational sources to create multiple rows from a single row of data.

**Example 1:** To create 4 records of every employee in EMP table.

* EMP will be source table.
* Create target table Normalizer\_Multiple\_Records. Structure same as EMP and datatype of HIREDATE as VARCHAR2.
* Create shortcuts as necessary.

**Creating Mapping :**

1. Open folder where we want to create the mapping.
2. Click Tools -> Mapping Designer.
3. Click Mapping-> Create-> Give name. Ex: m\_ Normalizer\_Multiple\_Records
4. Drag EMP and Target table.
5. Transformation->Create->Select Expression-> Give name, Click create, done.
6. Pass all ports from SQ\_EMP to Expression transformation.
7. Transformation-> Create-> Select Normalizer-> Give name, create & done.
8. Try dragging ports from Expression to Normalizer. Not Possible.
9. Edit Normalizer and Normalizer Tab. Add columns. Columns equal to columns in EMP table and datatype also same.
10. Normalizer doesn’t have DATETIME datatype. So convert HIREDATE to char in expression t/f. Create output port out\_hdate and do the conversion.
11. Connect ports from Expression to Normalizer.
12. Edit Normalizer and Normalizer Tab. As EMPNO identifies source records and we want 4 records of every employee, give OCCUR for EMPNO as 4.
13. [](http://lh6.ggpht.com/_MbhSjEtmzI8/Ta71Yt4NnwI/AAAAAAAAASE/DbLCHkAz-5I/s1600-h/clip_image002%5b3%5d.jpg)
14. Click Apply and then OK.
15. Add link as shown in mapping below:
16. Mapping -> Validate
17. Repository -> Save

* Make session and workflow.
* Give connection information for source and target table.
* Run workflow and see result.

**Example 2:** To break rows into columns

**Source:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Roll\_Number | Name | ENG | HINDI | MATHS |
| 100 | Amit | 78 | 76 | 90 |
| 101 | Rahul | 76 | 78 | 87 |
| 102 | Jessie | 65 | 98 | 79 |

**Target :**

|  |  |  |
| --- | --- | --- |
| Roll\_Number | Name | Marks |
| 100 | Amit | 78 |
| 100 | Amit | 76 |
| 100 | Amit | 90 |
| 101 | Rahul | 76 |
| 101 | Rahul | 78 |
| 101 | Rahul | 87 |
| 102 | Jessie | 65 |
| 102 | Jessie | 98 |
| 102 | Jessie | 79 |

* Make source as a flat file. Import it and create target table.
* Create Mapping as before. In Normalizer tab, create only 3 ports Roll\_Number, Name and Marks as there are 3 columns in target table.
* Also as we have 3 marks in source, give Occurs as 3 for Marks in Normalizer tab.
* Connect accordingly and connect to target.
* Validate and Save
* Make Session and workflow and Run it. Give Source File Directory and Source File name for source flat file in source properties in mapping tab of session.
* See the result.

**SEQUENCE GENERATOR T/F**

* Passive and Connected Transformation.
* The Sequence Generator transformation generates numeric values.
* Use the Sequence Generator to create unique primary key values, replace missing primary keys, or cycle through a sequential range of numbers.

We use it to generate Surrogate Key in DWH environment mostly. When we want to Maintain history, then we need a key other than Primary Key to uniquely identify the record. So we create a Sequence 1,2,3,4 and so on. We use this sequence as the key. Example: If EMPNO is the key, we can keep only one record in target and can’t maintain history. So we use Surrogate key as Primary key and not EMPNO.

**Sequence Generator Ports :**

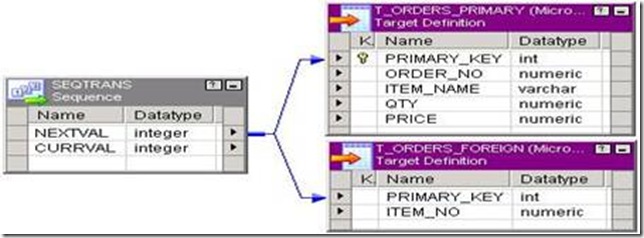
The Sequence Generator transformation provides two output ports: NEXTVAL and CURRVAL.

* We cannot edit or delete these ports.
* Likewise, we cannot add ports to the transformation.

**NEXTVAL:**

Use the NEXTVAL port to generate sequence numbers by connecting it to a Transformation or target.

**For example**, we might connect NEXTVAL to two target tables in a mapping to generate unique primary key values.

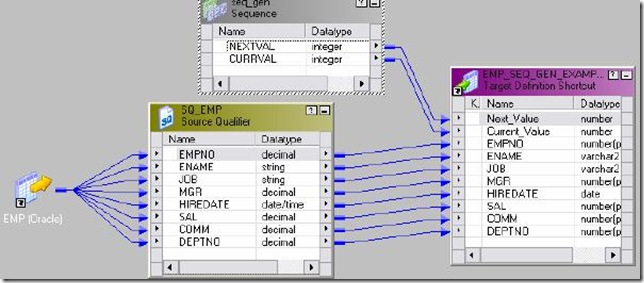
[](http://lh3.ggpht.com/_MbhSjEtmzI8/Ta8jgJVv-gI/AAAAAAAAAUY/E1FNUCi-T-c/s1600-h/clip_image002%5b4%5d.jpg)

Sequence in Table 1 will be generated first. When table 1 has been loaded, only then Sequence for table 2 will be generated.

**CURRVAL:**

CURRVAL is NEXTVAL plus the Increment By value.

* We typically only connect the CURRVAL port when the NEXTVAL port is Already connected to a downstream transformation.
* If we connect the CURRVAL port without connecting the NEXTVAL port, the Integration Service passes a constant value for each row.
* when we connect the CURRVAL port in a Sequence Generator Transformation, the Integration Service processes one row in each block.
* We can optimize performance by connecting only the NEXTVAL port in a Mapping.

[](http://lh4.ggpht.com/_MbhSjEtmzI8/Ta8jhrs24ZI/AAAAAAAAAUg/fHtZ07MO8Tg/s1600-h/clip_image003%5b4%5d.jpg)

**Example:** To use Sequence Generator transformation

* EMP will be source.
* Create a target EMP\_SEQ\_GEN\_EXAMPLE in shared folder. Structure same as EMP. Add two more ports NEXT\_VALUE and CURR\_VALUE to the target table.
* Create shortcuts as needed.

**Creating Mapping:**

1. Open folder where we want to create the mapping.

2. Click Tools -> Mapping Designer.

3. Click Mapping-> Create-> Give name. Ex: m\_seq\_gen\_example

4. Drag EMP and Target table.

5. Connect all ports from SQ\_EMP to target table.

6. Transformation -> Create -> Select Sequence Generator for list -> Create -> Done

7. Connect NEXT\_VAL and CURR\_VAL from Sequence Generator to target.

8. Validate Mapping

9. Repository -> Save

* Create Session and then workflow.
* Give connection information for all tables.
* Run workflow and see the result in table.

**Points to Ponder:**

* If Current value is 1 and end value 10, no cycle option. There are 17 records in source. In this case session will fail.
* If we connect just CURR\_VAL only, the value will be same for all records.
* If Current value is 1 and end value 10, cycle option there. Start value is 0.
* There are 17 records in source. Sequence: 1 2 – 10. 0 1 2 3 –
* To make above sequence as 1-10 1-20, give Start Value as 1. Start value is used along with Cycle option only.
* If Current value is 1 and end value 10, cycle option there. Start value is 1.
* There are 17 records in source. Session runs. 1-10 1-7. 7 will be saved in repository. If we run session again, sequence will start from 8.
* Use reset option if you want to start sequence from CURR\_VAL every time.

**Stored procedure:**

* Passive Transformation
* Connected and Unconnected Transformation
* Stored procedures are stored and run within the database.

A Stored Procedure transformation is an important tool for populating and Maintaining databases. Database administrators create stored procedures to Automate tasks that are too complicated for standard SQL statements.

**Use of Stored Procedure in mapping:**

* Check the status of a target database before loading data into it.
* Determine if enough space exists in a database.
* Perform a specialized calculation.
* Drop and recreate indexes. Mostly used for this in projects.

Data Passes Between IS and Stored Procedure One of the most useful features of stored procedures is the ability to send data to the stored procedure, and receive data from the stored procedure. There are three types of data that pass between the Integration Service and the stored procedure:

**Input/output parameters:** Parameters we give as input and the parameters returned from Stored Procedure.

**Return values:** Value returned by Stored Procedure if any.

**Status codes:** Status codes provide error handling for the IS during a workflow. The stored procedure issues a status code that notifies whether or not the stored procedure completed successfully. We cannot see this value. The IS uses it to determine whether to continue running the session or stop. Specifying when the Stored Procedure Runs

**Normal:** The stored procedure runs where the transformation exists in the mapping on a row-by-row basis. We pass some input to procedure and it returns some calculated values. Connected stored procedures run only in normal mode.

**Pre-load of the Source:** Before the session retrieves data from the source, the stored procedure runs. This is useful for verifying the existence of tables or performing joins of data in a temporary table.

**Post-load of the Source:**After the session retrieves data from the source, the stored procedure runs. This is useful for removing temporary tables.

**Pre-load of the Target:** Before the session sends data to the target, the stored procedure runs. This is useful for dropping indexes or disabling constraints.

**Post-load of the Target:** After the session sends data to the target, the stored procedure runs. This is useful for re-creating indexes on the database.

**Using a Stored Procedure in a Mapping :**

1. Create the stored procedure in the database.
2. Import or create the Stored Procedure transformation.
3. Determine whether to use the transformation as connected or unconnected.
4. If connected, map the appropriate input and output ports.
5. If unconnected, either configure the stored procedure to run pre- or post-session, or configure it to run from an expression in another transformation.
6. Configure the session.

**Stored Procedures:**

**Connect to Source database and create the stored procedures given below:**

CREATE OR REPLACE procedure sp\_agg (in\_deptno in number, max\_sal out number,

min\_sal out number, avg\_sal out number, sum\_sal out number)

As

Begin

select max(Sal),min(sal),avg(sal),sum(sal) into max\_sal,min\_sal,avg\_sal,sum\_sal

from emp where deptno=in\_deptno group by deptno;

End;

/

CREATE OR REPLACE procedure sp\_unconn\_1\_value(in\_deptno in number, max\_sal out number)

As

Begin

Select max(Sal) into max\_sal from EMP where deptno=in\_deptno;

End;

/

**1. Connected Stored Procedure T/F**

**Example:** To give input as DEPTNO from DEPT table and find the MAX, MIN, AVG and SUM of SAL from EMP table.

* DEPT will be source table. Create a target table SP\_CONN\_EXAMPLE with fields DEPTNO, MAX\_SAL, MIN\_SAL, AVG\_SAL & SUM\_SAL.
* Write Stored Procedure in Database first and Create shortcuts as needed.

**Creating Mapping:**

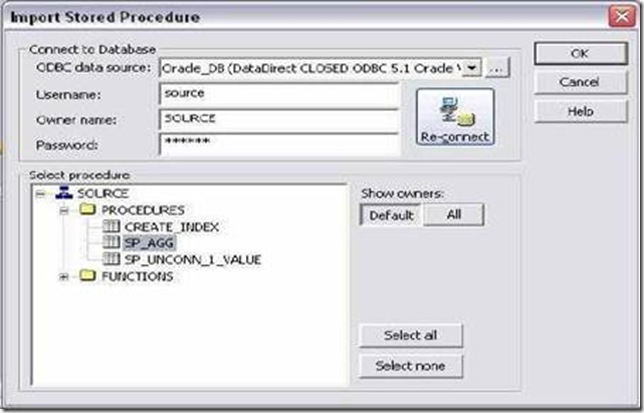
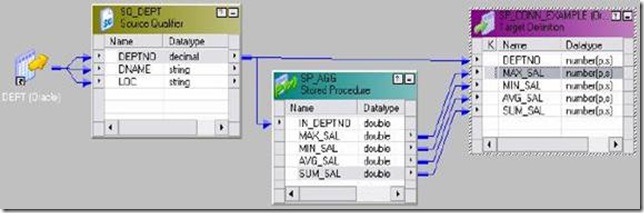
1. Open folder where we want to create the mapping.

2. Click Tools -> Mapping Designer.

3. Click Mapping-> Create-> Give name. Ex: m\_SP\_CONN\_EXAMPLE

4. Drag DEPT and Target table.

5. Transformation -> Import Stored Procedure -> Give Database Connection -> Connect -> Select the procedure sp\_agg from the list.

[](http://lh5.ggpht.com/_MbhSjEtmzI8/Ta-W-bQgvXI/AAAAAAAAAV4/Ch265raCRbI/s1600-h/clip_image030%5b4%5d.jpg)[](http://lh3.ggpht.com/_MbhSjEtmzI8/Ta-XA8vgHMI/AAAAAAAAAWA/HfvvBwFifTA/s1600-h/clip_image032%5b5%5d.jpg)

6. Drag DEPTNO from SQ\_DEPT to the stored procedure input port and also to DEPTNO port of target.

7. Connect the ports from procedure to target as shown below:

8. Mapping -> Validate

9. Repository -> Save

* Create Session and then workflow.
* Give connection information for all tables.
* Give connection information for Stored Procedure also.
* Run workflow and see the result in table.

[](http://lh5.ggpht.com/_MbhSjEtmzI8/Ta-XCGapAUI/AAAAAAAAAWI/HRI1ugVXLhU/s1600-h/clip_image034%5b4%5d.jpg)

**2. Unconnected Stored Procedure T/F :**

An unconnected Stored Procedure transformation is not directly connected to the flow of data through the mapping. Instead, the stored procedure runs either:

* From an expression: Called from an expression transformation.
* Pre- or post-session: Runs before or after a session.

Method of returning the value of output parameters to a port:

* Assign the output value to a local variable.
* Assign the output value to the system variable PROC\_RESULT. (See Later)

**Example 1:** DEPTNO as input and get MAX of Sal as output.

* DEPT will be source table.
* Create a target table with fields DEPTNO and MAX\_SAL of decimal data type.
* Write Stored Procedure in Database first and Create shortcuts as needed.

**Creating Mapping:**

1. Open folder where we want to create the mapping.

2. Click Tools -> Mapping Designer.

3. Click Mapping-> Create-> Give name. Ex: m\_sp\_unconn\_1\_value

4. Drag DEPT and Target table.

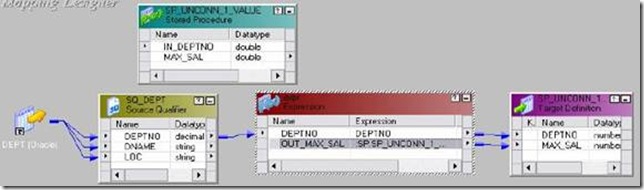
5. Transformation -> Import Stored Procedure -> Give Database Connection -> Connect -> Select the procedure sp\_unconn\_1\_value from the list. Click OK.

6. Stored Procedure has been imported.

7. T/F -> Create Expression T/F. Pass DEPTNO from SQ\_DEPT to Expression T/F.

8. Edit expression and create an output port OUT\_MAX\_SAL of decimal data type.

9. Open Expression editor and call the stored procedure as below:Click OK and connect the port from expression to target as in mapping below:

[](http://lh6.ggpht.com/_MbhSjEtmzI8/Ta-XDowYl2I/AAAAAAAAAWQ/c24X1tGV5eA/s1600-h/clip_image036%5b4%5d.jpg)

10. Mapping -> Validate

11. Repository Save.

* Create Session and then workflow.
* Give connection information for all tables.
* Give connection information for Stored Procedure also.
* Run workflow and see the result in table.

**PROC\_RESULT use:**

* If the stored procedure returns a single output parameter or a return value, we the reserved variable PROC\_RESULT as the output variable.

**Example:** DEPTNO as Input and MAX Sal as output :

:SP.SP\_UNCONN\_1\_VALUE(DEPTNO,PROC\_RESULT)

* If the stored procedure returns multiple output parameters, you must create variables for each output parameter.

**Example:** DEPTNO as Input and MAX\_SAL, MIN\_SAL, AVG\_SAL and SUM\_SAL

as output then:

1. Create four variable ports in expression VAR\_MAX\_SAL,VAR\_MIN\_SAL, VAR\_AVG\_SAL and iVAR\_SUM\_SAL.

2. Create four output ports in expression OUT\_MAX\_SAL, OUT\_MIN\_SAL, OUT\_AVG\_SAL and OUT\_SUM\_SAL.

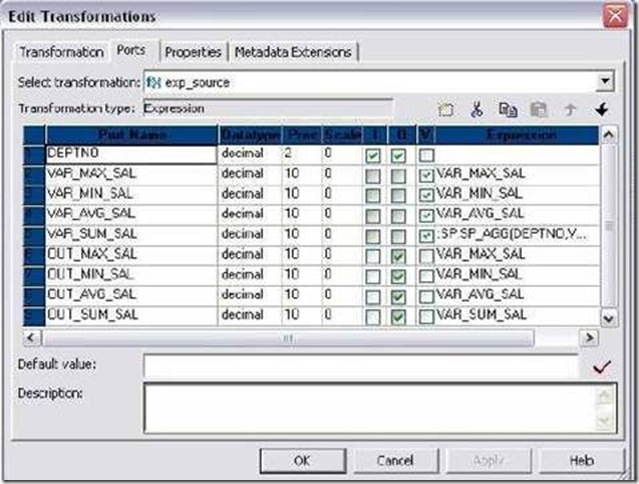
3. Call the procedure in last variable port says VAR\_SUM\_SAL.

:SP.SP\_AGG (DEPTNO, VAR\_MAX\_SAL,VAR\_MIN\_SAL, VAR\_AVG\_SAL, PROC\_RESULT)

**Example 2:**

DEPTNO as Input and MAX\_SAL, MIN\_SAL, AVG\_SAL and SUM\_SAL as O/P Stored Procedure to drop index in Pre Load of Target Stored Procedure to create index in Post Load of Target

* DEPT will be source table. Create a target table SP\_UNCONN\_EXAMPLE with fields DEPTNO, MAX\_SAL, MIN\_SAL, AVG\_SAL & SUM\_SAL.
* Write Stored Procedure in Database first and Create shortcuts as needed. Stored procedures are given below to drop and create index on target.Make sure to create target table first. Stored Procedures to be created in next example in Target Database:

[clip_image007[4]](http://lh6.ggpht.com/_MbhSjEtmzI8/Ta-XFC4y2TI/AAAAAAAAAWY/3L1MQkI_oaw/s1600-h/clip_image007%5b4%5d%5b2%5d.gif)[](http://lh3.ggpht.com/_MbhSjEtmzI8/Ta-XHCyP2hI/AAAAAAAAAWg/zzRRhL_B1io/s1600-h/clip_image038%5b5%5d.jpg)

Create or replace procedure CREATE\_INDEX

As

Begin

Execute immediate 'create index unconn\_dept on SP\_UNCONN\_EXAMPLE(DEPTNO)';

End;

/

Create or replace procedure DROP\_INDEX

As

Begin

Execute immediate 'drop index unconn\_dept';

End;

/

**Creating Mapping:**

1. Open folder where we want to create the mapping.

2. Click Tools -> Mapping Designer.

3. Click Mapping-> Create-> Give name. Ex: m\_sp\_unconn\_1\_value

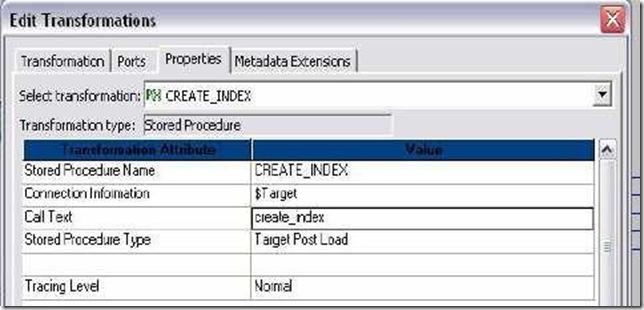
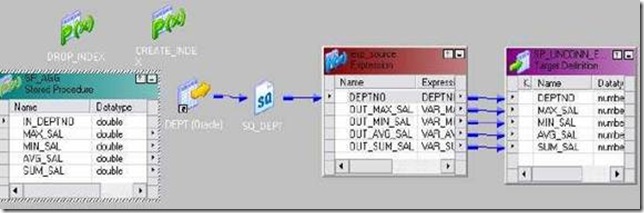
4. Drag DEPT and Target table.

5. Transformation -> Import Stored Procedure -> Give Database Connection -> Connect -> Select the procedure sp\_agg from the list. Click OK.

6. Stored Procedure has been imported.

7. T/F -> Create Expression T/F. Pass DEPTNO from SQ\_DEPT to Expression T/F.

8. Edit Expression and create 4 variable ports and 4 output ports as shown below:

[](http://lh4.ggpht.com/_MbhSjEtmzI8/Ta-XIUrLwmI/AAAAAAAAAWo/f_HMEifcsjk/s1600-h/clip_image039%5b4%5d.jpg)[](http://lh5.ggpht.com/_MbhSjEtmzI8/Ta-XJ_YW0ZI/AAAAAAAAAWw/rW6m0eDSfLw/s1600-h/clip_image041%5b4%5d.jpg)

9. Call the procedure in last variable port VAR\_SUM\_SAL.

10. :SP.SP\_AGG (DEPTNO, VAR\_MAX\_SAL, VAR\_MIN\_SAL, VAR\_AVG\_SAL, PROC\_RESULT)

11. Click Apply and Ok.

12. Connect to target table as needed.

13. Transformation -> Import Stored Procedure -> Give Database Connection for target -> Connect -> Select the procedure CREATE\_INDEX and DROP\_INDEX from the list. Click OK.

14. Edit DROP\_INDEX -> Properties Tab -> Select Target Pre Load as Stored Procedure Type and in call text write drop\_index. Click Apply -> Ok.

15. Edit CREATE\_INDEX -> Properties Tab -> Select Target Post Load as Stored Procedure Type and in call text write create\_index. Click Apply -> Ok.

16. Mapping -> Validate

17. Repository -> Save

* Create Session and then workflow.
* Give connection information for all tables.
* Give connection information for Stored Procedures also.
* Also make sure that you execute the procedure CREATE\_INDEX on database before using them in mapping. This is because, if there is no INDEX on target table, DROP\_INDEX will fail and Session will also fail.
* Run workflow and see the result in table.