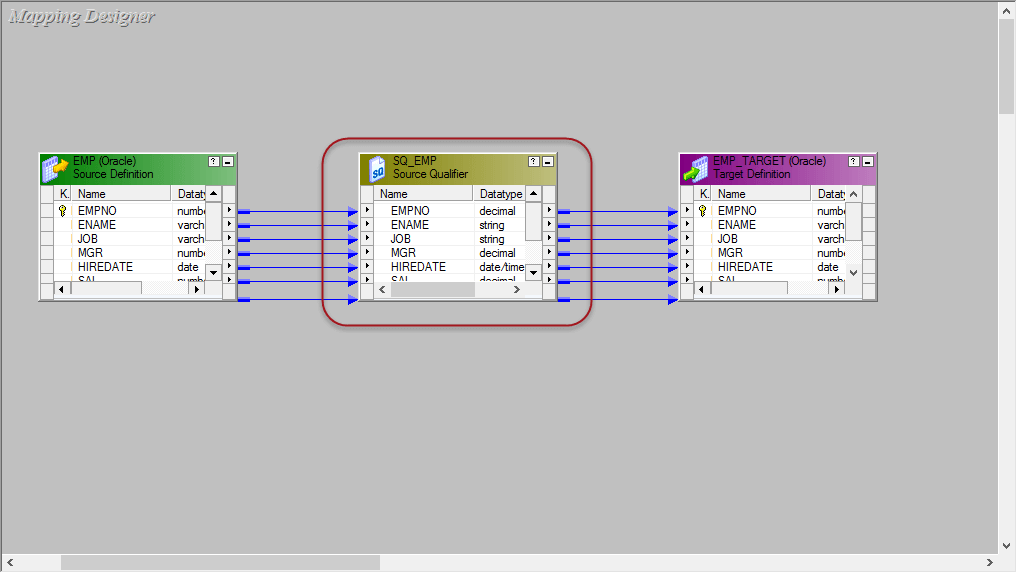
**What is Source Qualifier Transformation?**

Source qualifier transformation is an active, connected transformation which is used to represent the rows that the integrations service read. Whenever we add a relational source or a flat file to a mapping, a source qualifier transformation is required. When we add a source to a mapping, source qualifier transformation is added automatically. With source qualifier, we can define and override how the data is fetched from the source.

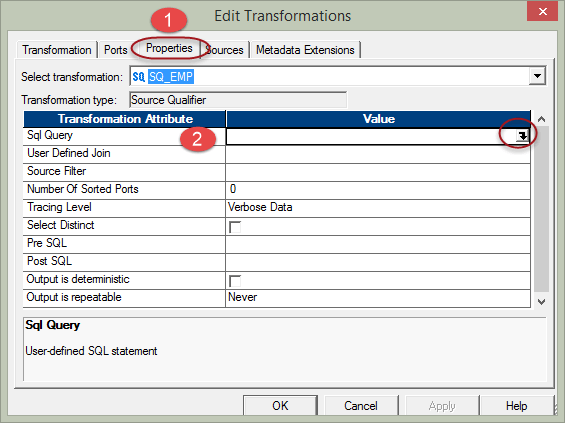
In the following example we would be modifying source qualifier of our mapping “m\_emp\_emp\_target”, so instead of returning all the columns it will return only selected columns.

**Step 1** – Open mapping “m\_emp\_emp\_target” in mapping designer.



**Step 2** – Double click on the source Qualifier transformation “SQ\_EMP”. It will open edit transformation property window for it. Then

1. Click on the properties tab
2. Click on the[SQL](https://www.guru99.com/sql.html)Query Modify option, this will open an SQL editor window



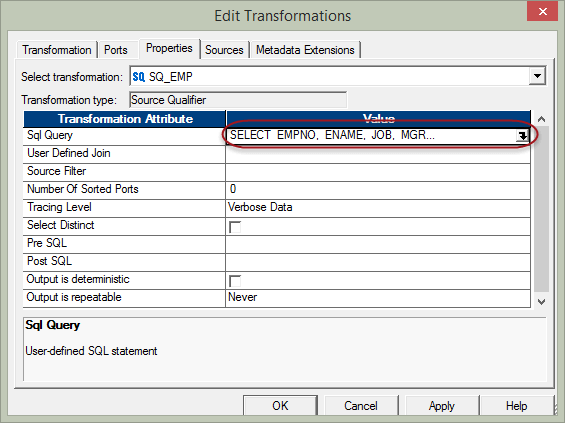
**Step 3** – In the SQL editor window

1. Enter the following query

SELECT EMPNO, ENAME, JOB, MGR FROM EMP

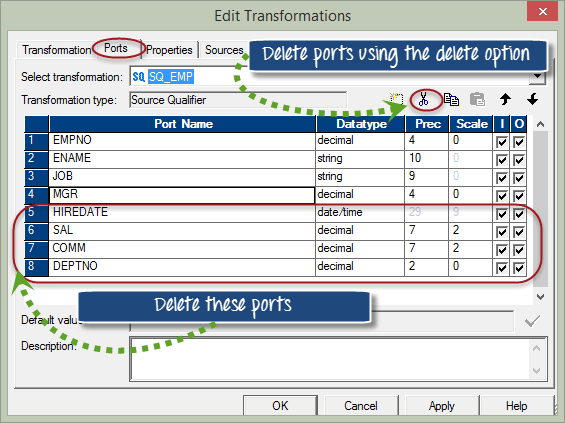
**Note** – we are selecting the columns EMPNO, ENAME, JOB & MANAGER from the source, so we have kept only those in the select query

1. Select OK Button

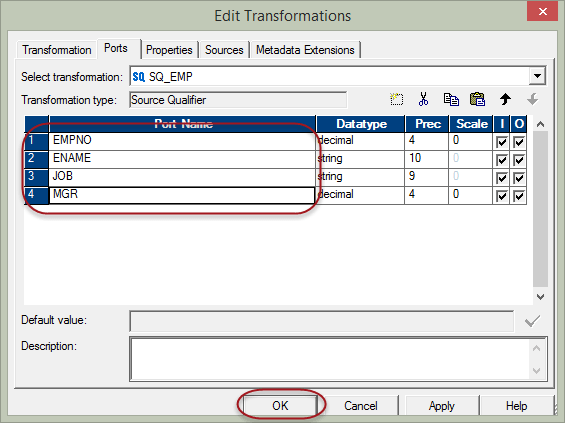


**Step 4** – In the “edit transformations” window,

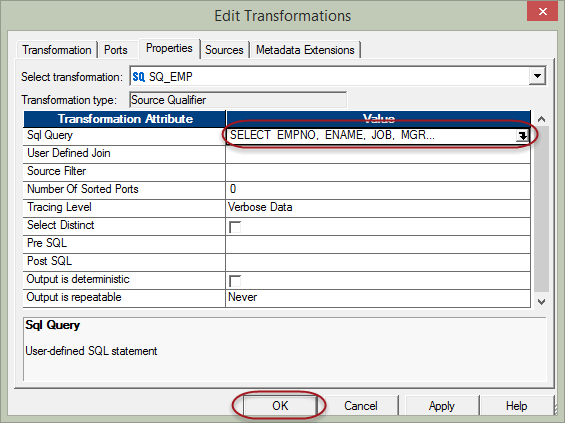
1. Select Ports tab from the menu
2. Under ports tab, you will see all the ports. Keep only the ports EMPNO, ENAME, JOB, MGR and delete other ports



**Step 5** – After deletion of ports, Select OK Button

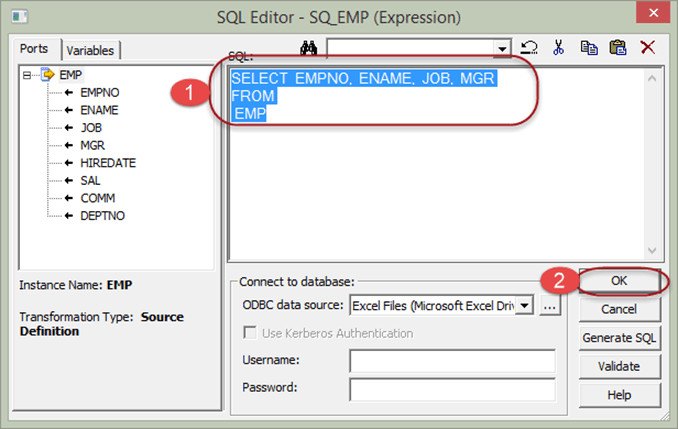


Now, again click on properties tab in Edit Transformations window, and you will see only those data that you have selected.



When you click on “OK” button it will open SQL Editor Window, and

1. It will confirm the data you have selected are correct and ready for loading into the target table
2. Click on OK button to process further



Save the mapping (using ctrl+s Shortcut) and execute the workflow, after execution only the selected columns will be loaded into the target.

In this way, you can override in source qualifier what columns needs to be fetched from the source & this is the only way to override what specific columns will be brought inside the mapping.

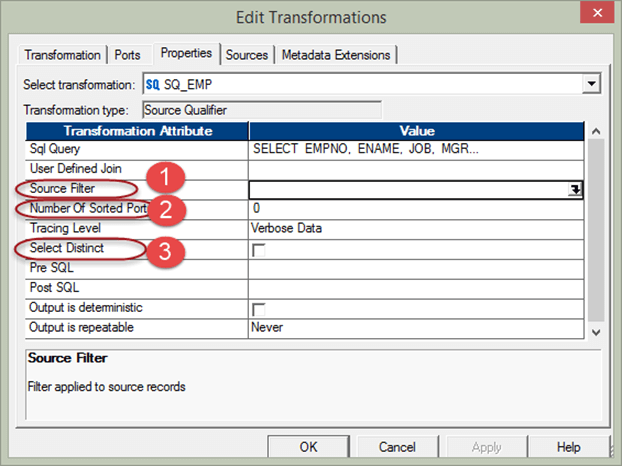
**Properties of Source Qualifier**

You can use various properties of Source Qualifier, to determine what type of source data needs to transform to target table.

1. **Source Filter** – Using the source filter property you can filter the number of source records. For example, you want to fetch only the employees of deptno 10, then you can enter the filter condition deptno=10 in source filter property and execute the data.
2. **Number for sorted ports** – In source qualifier transformation, you can also sort the input records based on the ports number. So when the data is passed on to the transformations inside the mapping, it will read the port number and sort the data accordingly.

As data can be sorted based on a single or multiple ports, you have to give the number of ports which will be used in sorting. If you give value as 1, then only empno data will be sorted. If you give value as 2 then on empno and ename on both columns data will be sorted.

1. **Select Distinct** – you can fetch only distinct records from the source using this property. When you select the select distinct option, only distinct combination of source data will be fetched by source qualifier.

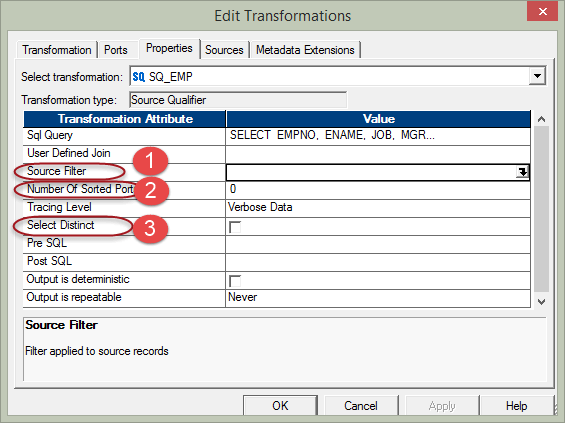


**What is Aggregator Transformation?**

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.



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.

[**Download the above Create\_table\_sal\_deptwise.txt File**](https://drive.google.com/uc?export=download&id=0ByI5-ZLwpo25X2NOdDZ3MGxXUTA)

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

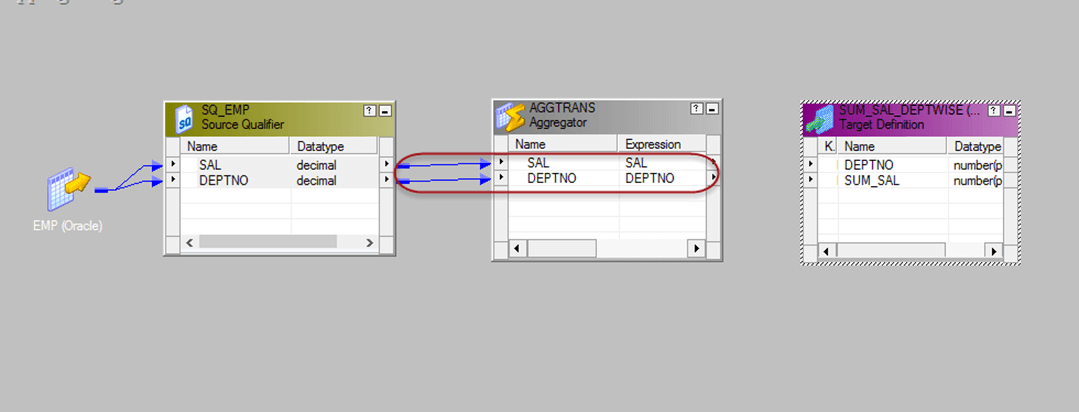
In order to create new mapping, we need source table (EMP) and target table (sum\_sal\_deptwise) both in mapping designer for that we need to

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

**Step 3** – In the mapping,

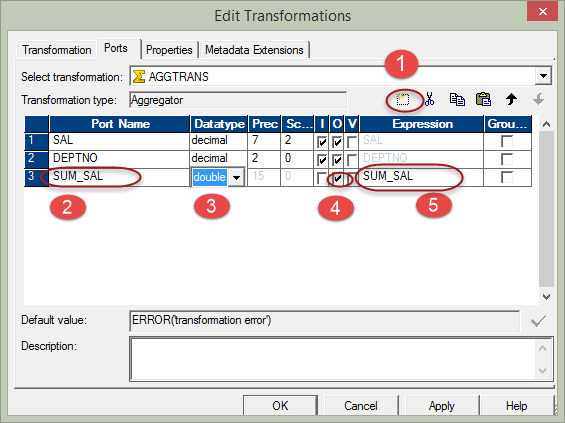
1. From the Source Qualifier, delete the columns empno, ename, job, mgr, hiredate & comm so leaving only the columns deptno and sal.
2. Create a new aggregator transformation using the toolbox menu as shown in screen shot. When you click on the aggregator icon, a new aggregator transformation will be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

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



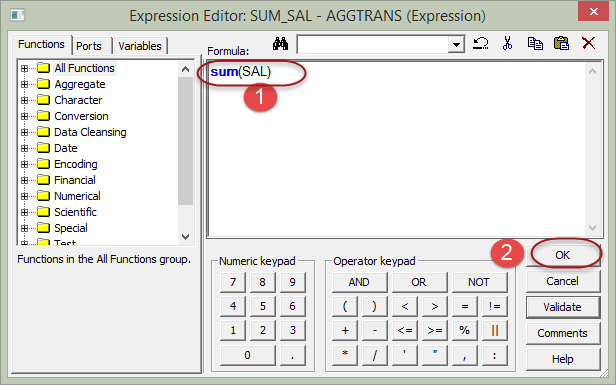
**Step 5** – Double click on the aggregator transformation to open its properties, and then

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

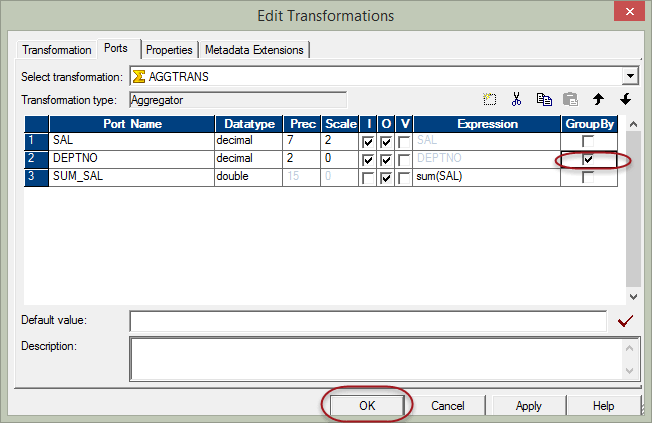


**Step 6** – In the expression window

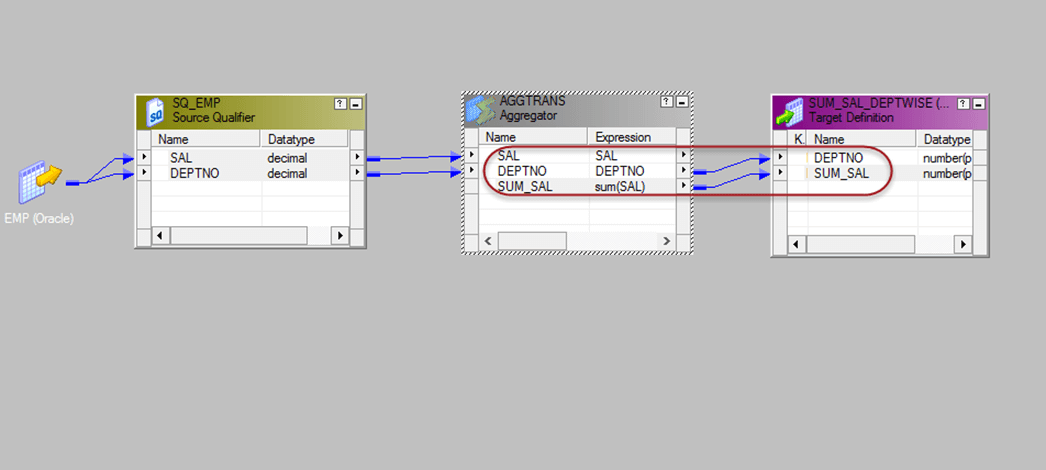
1. Add expression- sum(SAL), you have to write this expression.
2. Select Ok Button, this will bring back the edit transformation window.



**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)



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



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.

**What is Router Transformation?**

Router transformation is an active and connected transformation which is similar to filter transformation, used to filter the source data.

The additional functionality provided beside filtering is that the discarded data (filtered out data) can also be collected in the mapping, as well as the multiple filter conditions can be applied to get multiple sets of data.

For example, when filtering the data form deptno =10, we can also get those records where deptno is not equal to 10. So, router transformation gives multiple output groups, and each output group can have its own filter condition.

In addition there is also a default group, this default group has those record sets which doesn’t satisfy any of the group conditions. For example, if you have created two groups for the filter conditions deptno=10 & dept=20 respectively, then those records which are not having deptno 10 and 20 will be passed into this default group. In short the data which is rejected by the filter groups will be collected by this default group and sometimes there can be a requirement to store these rejected data. In such scenarios, default output group can be useful.

To allow multiple filter condition, the router transformation provides group option.

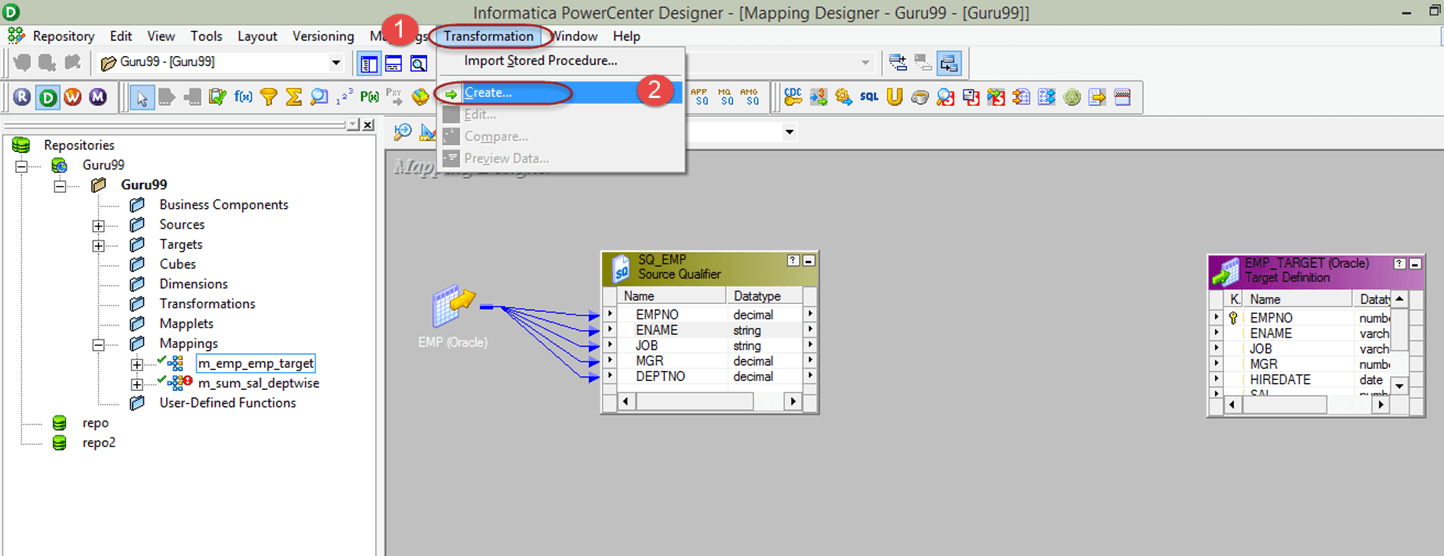
* There is a default input group which takes input data
* There is also a default output group which provides all those data which is not passed by any filter condition
* For every filter condition, an output group is created in router transformation. You can connect different targets to these different groups.

Creating Router Transformation

**Step 1** – Create a mapping having source “EMP” and target “EMP\_TARGET.”

**Step 2** – Then in the mapping

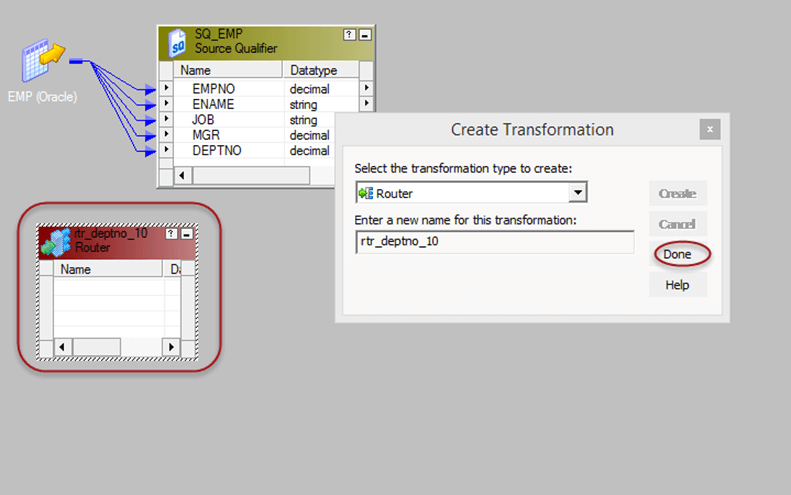
1. Select Transformation menu
2. Select create option



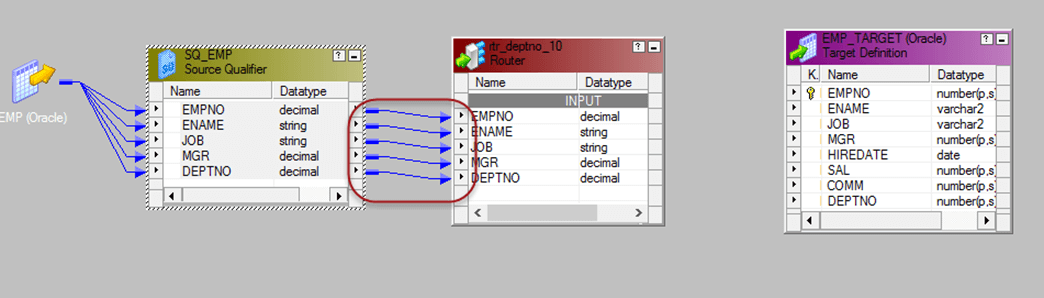
**Step 3** – In the create transformation window

1. Select router transformation
2. Enter a name for the transformation “rtr\_deptno\_10”
3. Select Create option

**Step 4** – The router transformation will be created in the mapping, select done option in the window

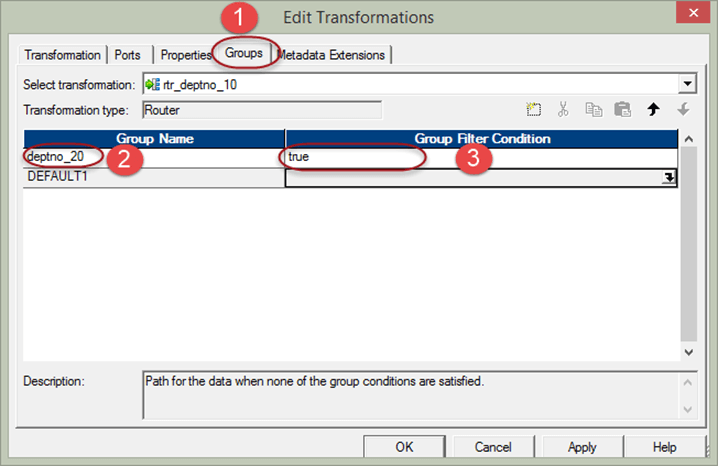


**Step 5** – Drag and drop all the columns from Source qualifier to router transformation

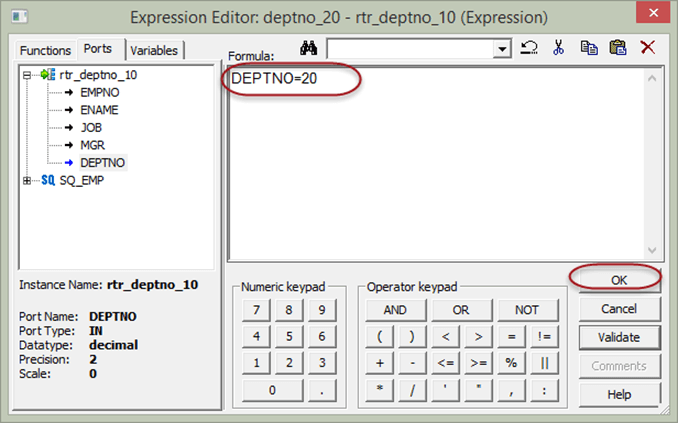


**Step 6** – Double click on the router transformation, then in the transformation property of it

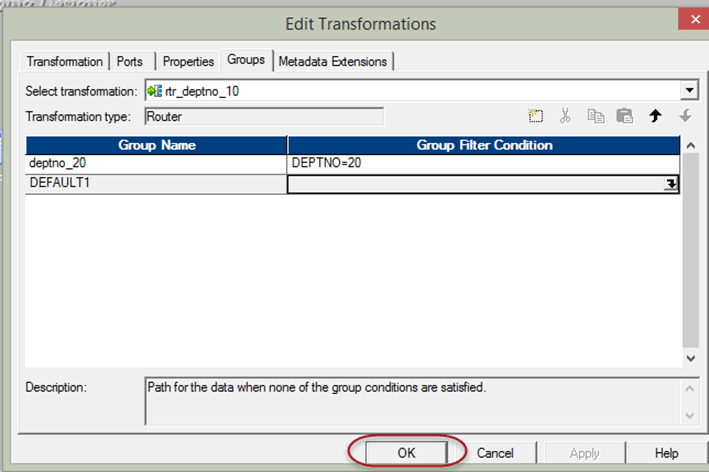
1. Select group tab
2. Enter group name “deptno\_20”
3. Click on the group filter condition



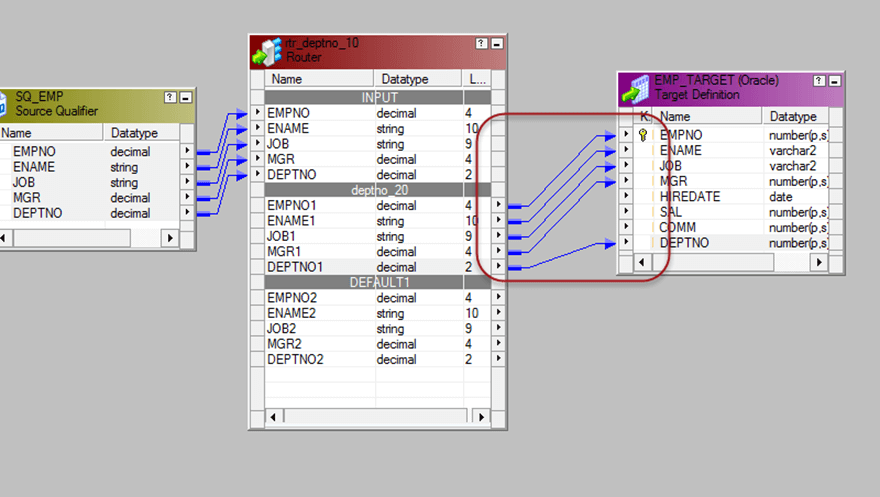
**Step 7** – In the expression editor, enter filter condition deptno=20 and select OK button.



**Step 8** – Select OK button in the group window



**Step 9** – Connect the ports from the group deptno\_20 of router transformation to target table ports



Now, when you execute this mapping, the filtered records will get loaded into the target table.

**What is Joiner Transformation?**

Joiner transformation is an active and connected transformation that provides you the option to create joins in Informatica. The joins created using joiner transformation are similar to the joins in databases. The advantage of joiner transformation is that joins can be created for heterogeneous systems (different databases).

In joiner transformation, there are two sources which we are going to use it for joins. These two sources are called

* Master Source
* Detail Source

In the properties of joiner transformation, you can select which data source can be Master and which source can be detail source.

During execution, the master source is cached into the memory for joining purpose. So it is recommended to select the source with less number of records as the master source.

The following joins can be created using joiner transformation

1. Master outer join

In Master outer join, all records from the Detail source are returned by the join and only matching rows from the master source are returned.

1. Detail outer join

In detail outer join only matching rows are returned from the detail source, and all rows from the master source are returned.

1. Full outer join

In full outer join, all records from both the sources are returned. Master outer and Detail outer joins are equivalent to left outer joins in SQL.

1. Normal join

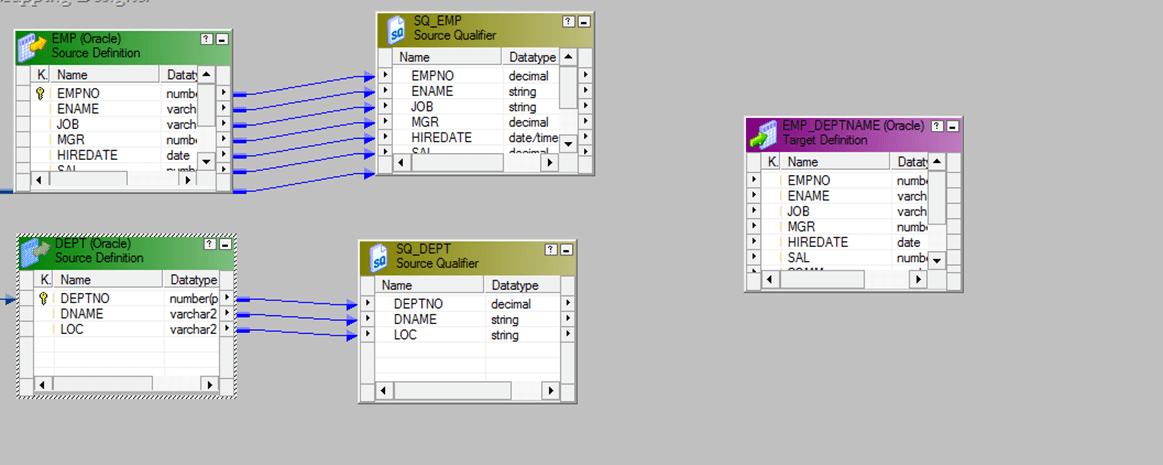
In normal join only matching rows are returned from both the sources.

In this example, we will join emp and dept tables using joiner transformation

**Step 1** – Create a new target table EMP\_DEPTNAME in the database using the below script and import the table in Informatica targets.

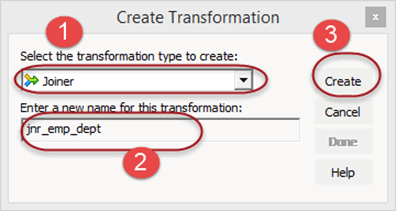
[**Download the above emp\_deptname.sql File**](https://drive.google.com/uc?export=download&id=0ByI5-ZLwpo25bVRfOXdQRkNRMGM)

**Step 2** – Create a new mapping and import source tables “EMP” and “DEPT” and target table which we created in the previous step

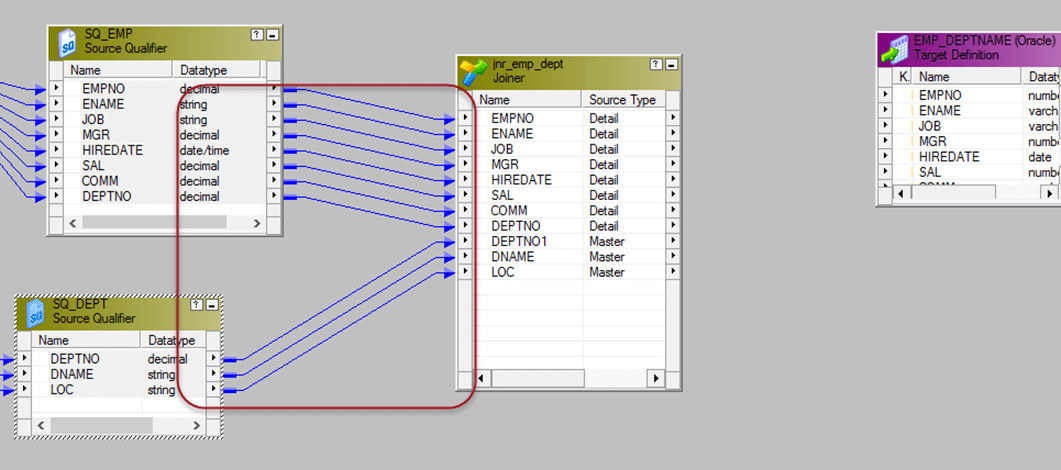


**Step 3** – From the transformation menu, select create option.

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

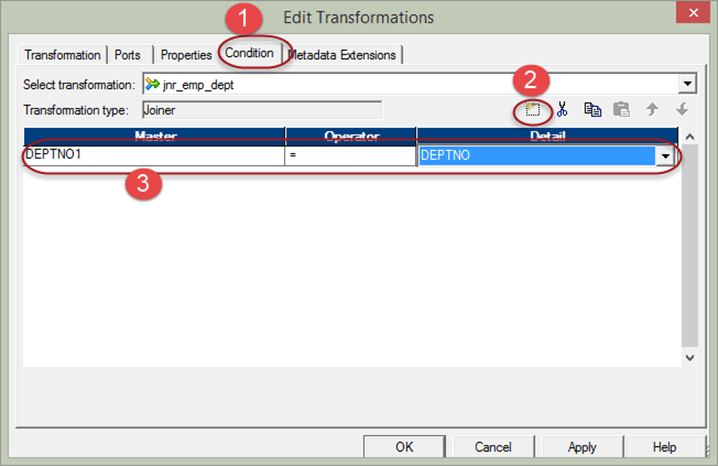


**Step 4** – Drag and drop all the columns from both the source qualifiers to the joiner transformation



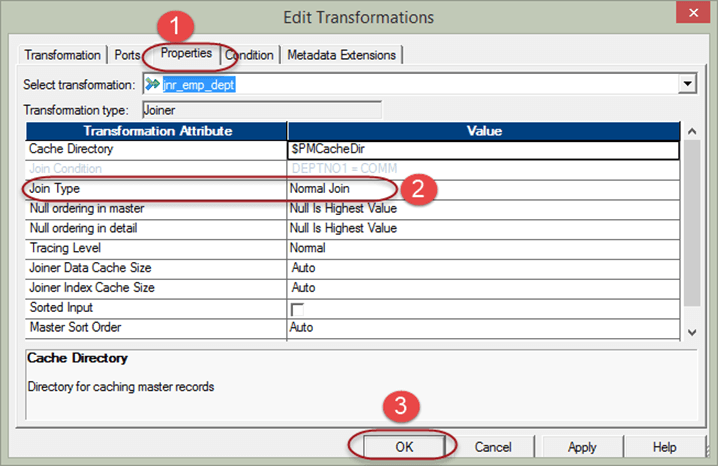
**Step 5** – Double click on the joiner transformation, then in the edit transformation window

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



**Step 6** – Then in the same window

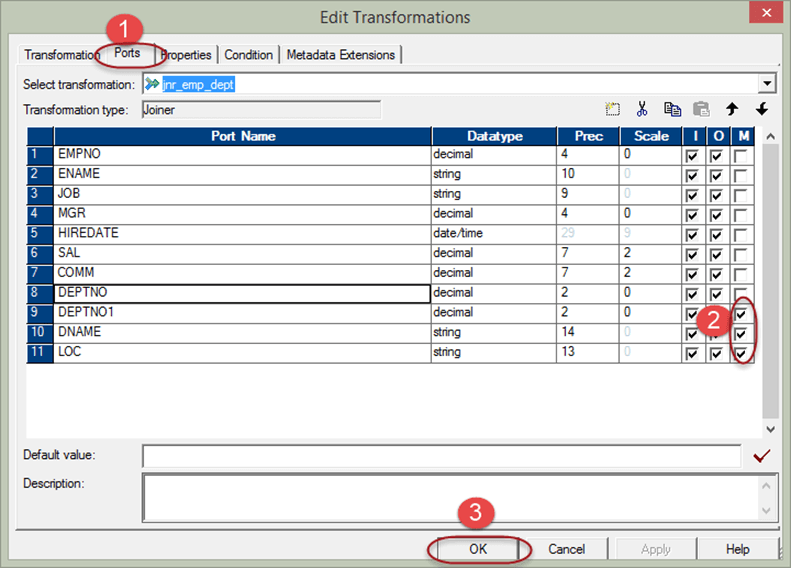
1. Select properties tab
2. Select normal Join as join type
3. Select OK Button



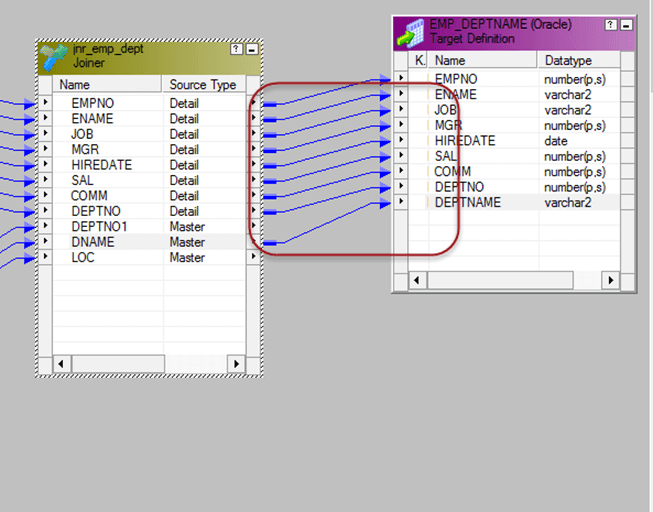
For performance optimization, we assign the master source to the source table pipeline which is having less no of records. To perform this task –

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

1. Select ports tab
2. Select any column of a particular source which you want to make a master
3. Select OK



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



Now save the mapping and execute it after creating session and workflow for it. The join will be created using Informatica joiner, and relevant details will be fetched from both the tables.

**What is Rank Transformation?**

Rank transformation is an active and connected transformation that performs the filtering of data based on group and ranks. For example, you want to get ten records of employees having highest salary, such kind of filtering can be done by rank transformation.

Rank transformation also provides the feature to do ranking based on groups. Like if you want to get top ten salaried employee department wise, then this grouping can be done with this transformation.

Rank transformation is an active transformation, as it affects the number of output rows.

The rank transformation has an output port by which it assigns a rank to the rows.

Our requirement is to load top 3 salaried employees for each department; we will implement this using rank transformation.

**Step 1** – Create a mapping having source EMP and target EMP\_TARGET

**Step 2**– Then in the mapping

1. Select transformation menu
2. Select create option

**Step 3** – In the create transformation window

1. Select rank transformation
2. Enter transformation name “rnk\_salary”
3. Select Create button

**Step 4** – The rank transformation will be created in the mapping, select done button in the window

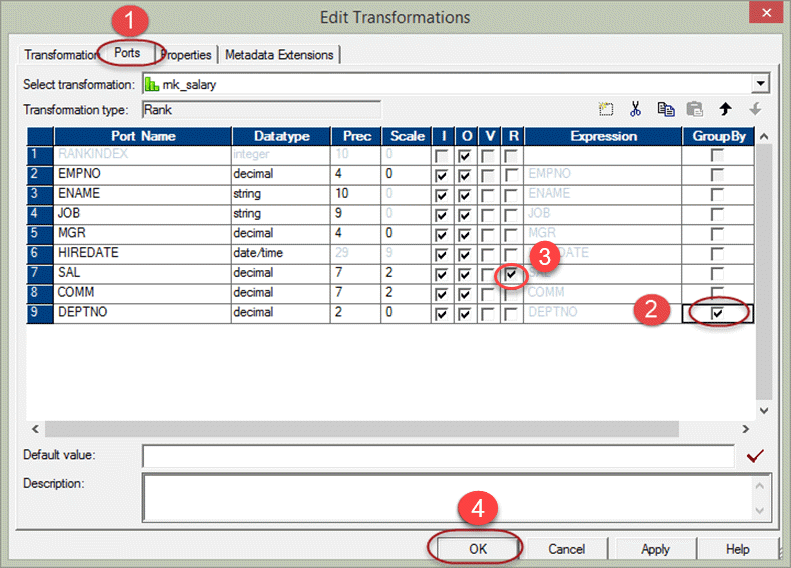
**Step 5** – Connect all the ports from source qualifier to the rank transformation

**Step 6**– Double click on the rank transformation and it will open “edit transformation window”. In this window

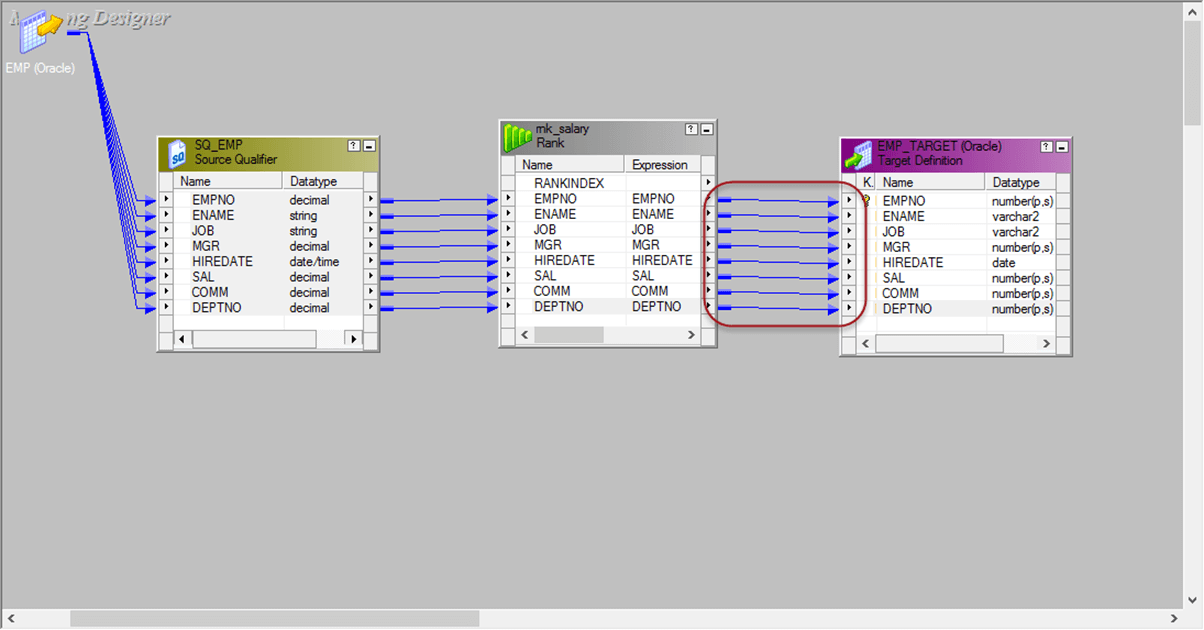
1. Select properties menu
2. Select “Top” option from the Top/Bottom property
3. Enter 3 in the number of ranks

**Step 7** – In the “edit transformation” window again

1. Select ports tab
2. Select group by option for the Department number column
3. Select Rank in the Salary Column
4. Select ok button



**Step 8** –Connect the ports from rank transformation to the target table



Now, save the mapping and execute it after creating session and workflow. The source qualifier will fetch all the records, but rank transformation will pass only records having three high salaries for each department.

**What is Sequence Generator Transformation?**

Sequence generator transformation is passive so it does not affect the number of input rows. The sequence generator is used to generate primary key values & it’s used to generate numeric sequence values like 1, 2, 3, 4, 5 etc.

For example, you want to assign sequence values to the source records, then you can use sequence generator. The generated sequence values can be like 5, 10, 15, 20, 25 etc. or 10, 20, 30, 40, 50 etc. depending upon the configured properties of the transformation.

Sequence generator has two output ports

* CURRVAL
* NEXTVAL

CURRVAL port value is always NEXTVAL+1.

To generate the sequence numbers, we always use the NEXTVAL column.

Properties of Sequence Generator Transformation

* Start Value – It is the first value that will be generated by the transformation, the default value is 0.
* Increment by – This is the number by which you want to increment the values. The default value is 1.
* End value – It is the maximum value that the transformation should generate.
* Cycle – if this option is set then after reaching the end of the value, the transformation restarts from the start value.

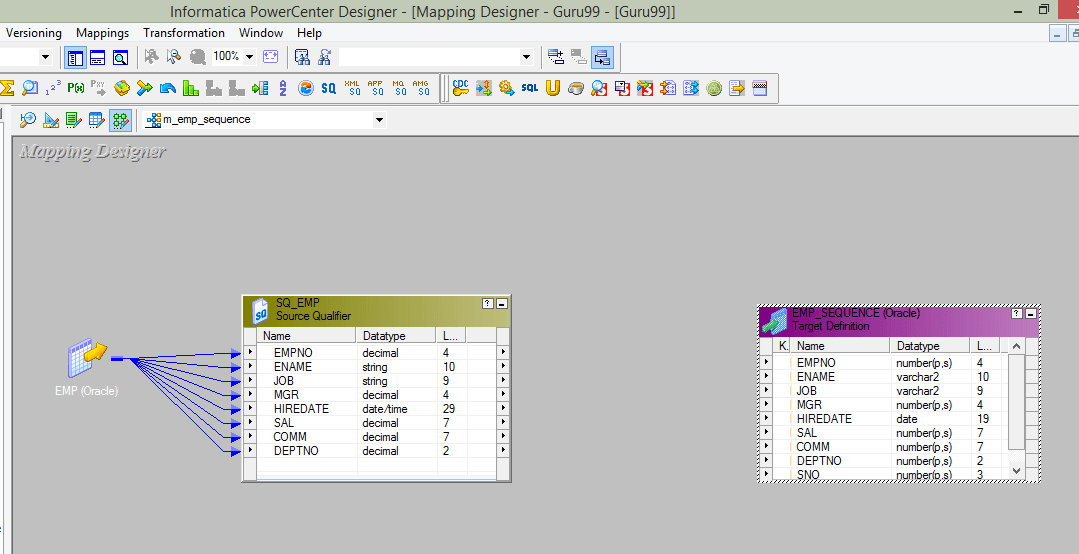
In this example, we will generate sequence numbers and store in the target.

**Step 1** – Create a target table with the following script.

[**Download the above emp\_sequence.sql File**](https://drive.google.com/uc?export=download&id=0ByI5-ZLwpo25VDZKTWR5dWNMMGM)

**Step 2**– Import the table in Informatica as target table

**Step 3** – Create a new mapping and import EMP source and EMP\_SEQUENCE target table

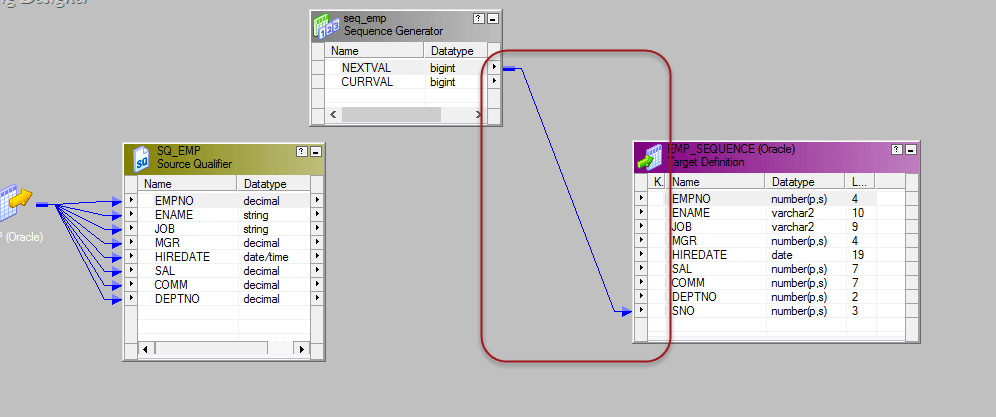


**Step 4** – Create a new transformation in the mapping

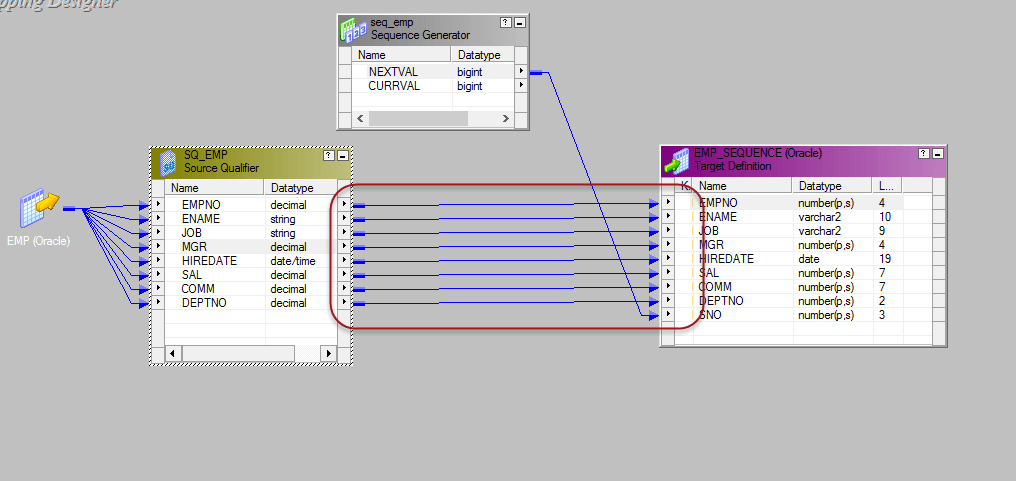
1. Select sequence transformation as the type
2. Enter transformation name “seq\_emp”
3. Select Create option

**Step 5** – Sequence generator transformation will be created, select the done option

**Step 6** – Link the NEXTVAL column of sequence generator to SNO column in target

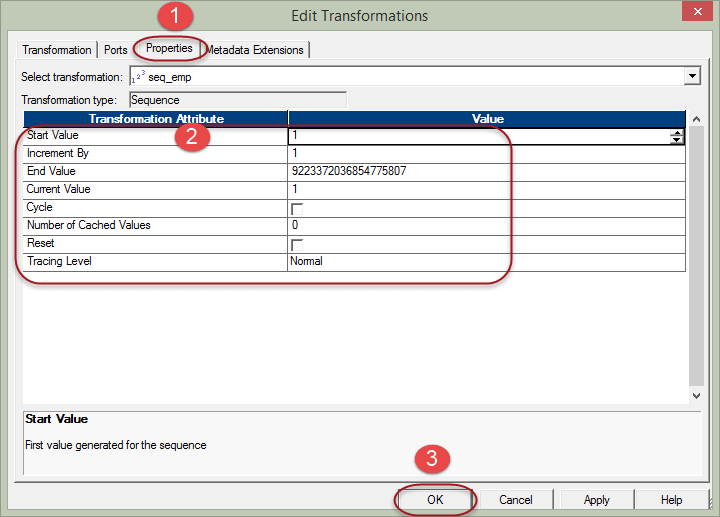


**Step 7** – link the other columns from source qualifier transformation to the target table



**Step 8** – Double click on the sequence generator to open property window, and then

1. Select the properties tab
2. Enter the properties with Start value =1, leave the rest properties as default
3. Select OK button



Now save the mapping and execute it after creating the session and workflow.

The sno column in the target would contain the sequence numbers generated by the sequence generator transformation.

In our example, the sequences will be like 1 – Scott, 2 – King, 3 – Adam, 4 – Miller, etc.

**What is Transaction Control Transformation?**

Transaction Control is an active and connected transformation which allows us to commit or rollback transactions during the execution of the mapping. Commit and rollback operations are of significant importance as it guarantees the availability of data.

When processing a high volume of data, there can be a situation when to commit the data to the target. If a commit is performed too frequently, then it will be an overhead to the system. If a commit is performed too late then in the case of failure there are chances of data loss.

So to provide flexibility Transaction control transformation is provided.

**TCL COMMIT & ROLLBACK Commands**

There are five in-built variables available in this transformation to handle the operation.

TC\_CONTINUE\_TRANSACTION

In tc\_continue\_transaction there are no operations performed, the process of data load continues as it is.

TC\_COMMIT\_BEFORE

In tc\_commit\_before when this flag is found set, a commit is performed before the processing of current row.

TC\_COMMIT\_AFTER

In tc\_commit\_after the current row is processed then a commit is performed.

TC\_ROLLBACK\_BEFORE

In tc\_rollback\_before, rollback is performed first then data is processed to write.

TC\_ROLLBACK\_AFTER

In tc\_rollback\_after data is processed then the rollback is performed.

**Example:**

In this example, we will commit data to the target when dept no =20 condition is found true

**Step 1** – Create a mapping with EMP as source and EMP\_TARGET as target

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

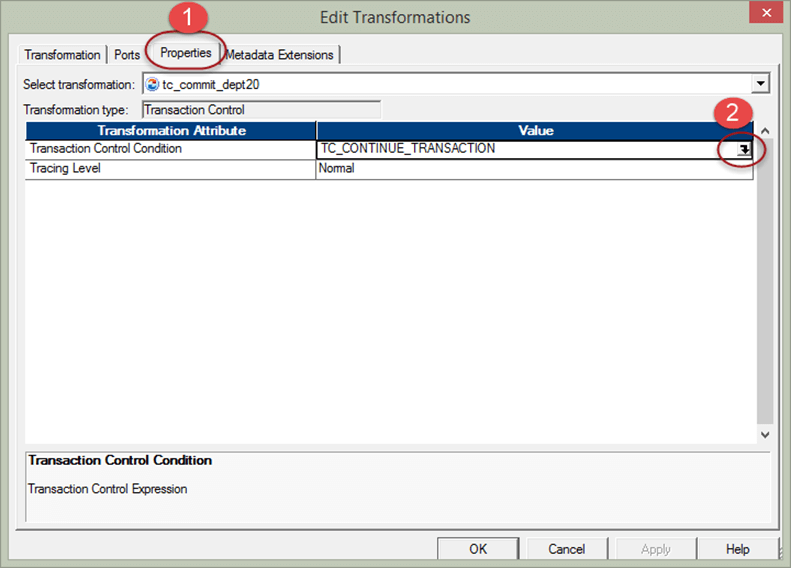
1. Select a transaction control as the new transformation
2. Enter transformation name “tc\_commit\_dept20”
3. Select create option

**Step 3** – The transaction control transformation will be created, select done button

**Step 4** – Drag and drop all the columns from source qualifier to the transaction control transformation then link all the columns from transaction control transformation to the target table

**Step 5** – Double click on the transaction control transformation and then in the edit property window

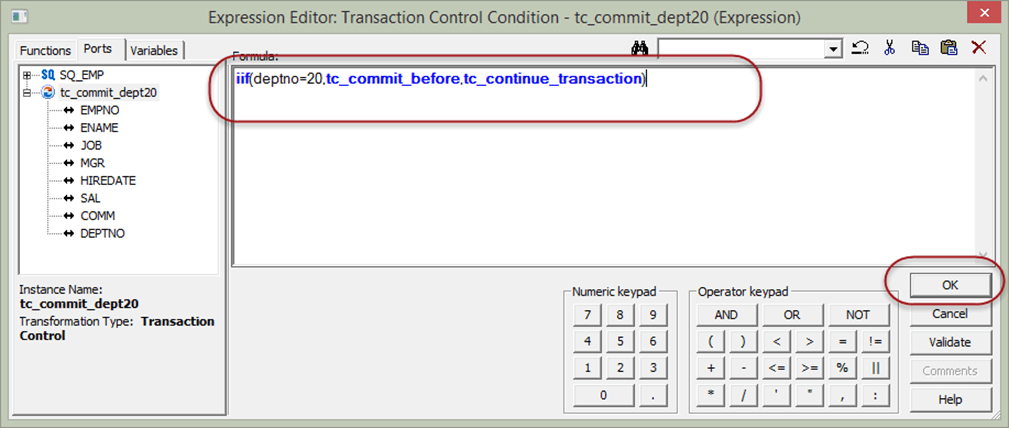
1. Select property tab
2. Click on the transaction control editor icon



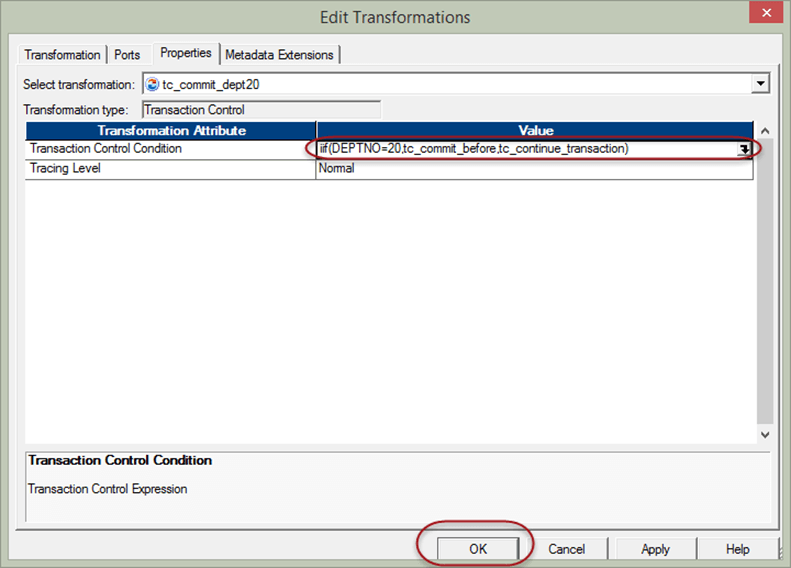
**Step 6** –in the expression editor enter the expression –

**“**iif(deptno=20,tc\_commit\_before,tc\_continue\_transaction)” and select OK

It means if deptno 20 is found then commit transaction in target, else continue the current processing.



**Step 7** – Select OK in the previous window



Now save the mapping and execute it after creating session and workflows. This mapping will commit the data to the target whenever department number 20 is found in the data.

**What is Lookup Transformation?**

Lookup transformation is a passive transformation used to look up a source, source qualifier, or target to get the relevant data. Basically, it’s a kind of join operation in which one of the joining tables is the source data, and the other joining table is the lookup table.

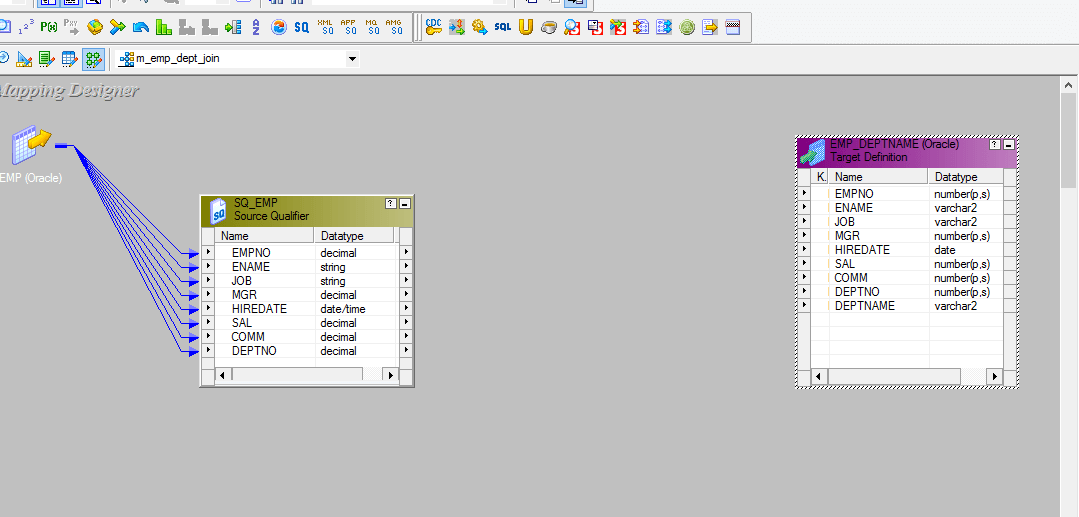
In previous topics, we used joiner transformation to join “emp” and “dept” table to bring department names. In this section, we will implement the same using lookup transformation.

**In this tutorial, you will learn:**

**Create Lookup Transformation**

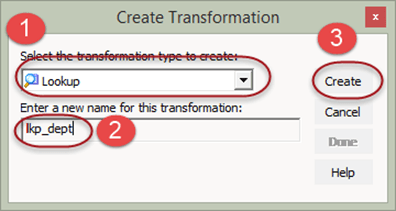
Following are the steps to create Lookup Transformation in Informatica

**Step 1** – Create a new mapping with EMP as source and EMP\_DEPTNAME as target



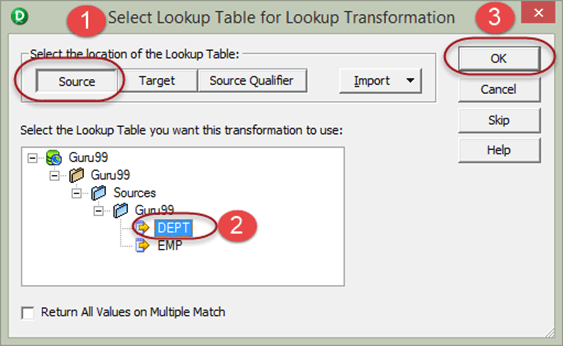
**Step 2** – Create a new transformation using transformation menu then

1. Select lookup transformation as the transformation
2. Enter transformation name “lkp\_dept”
3. Select create option

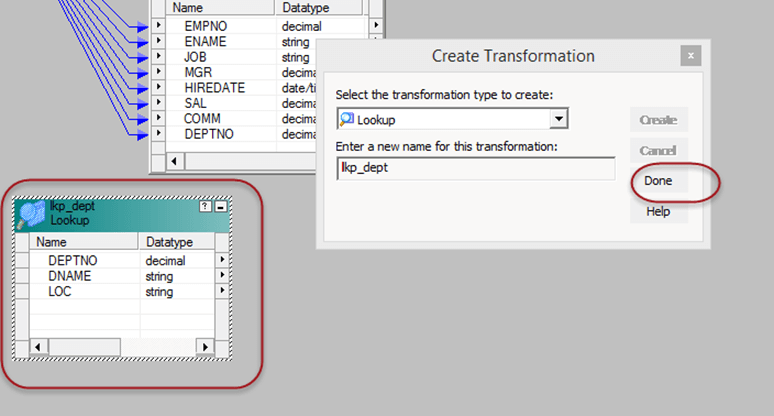


**Step 3** – This will open lookup table window, in this window

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

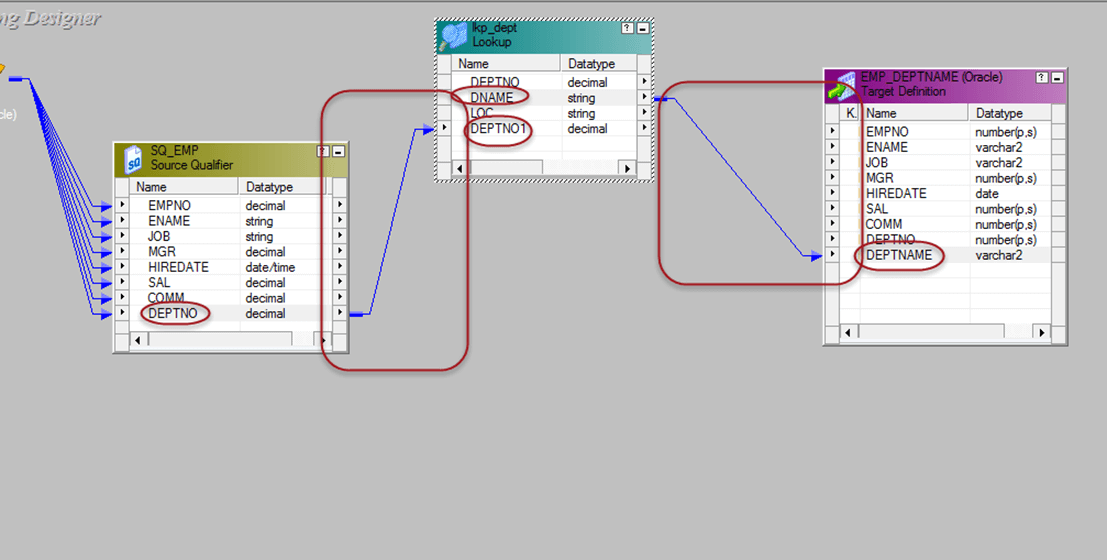


**Step 4** – Lookup transformation will be created with the columns of DEPT table, now select done button



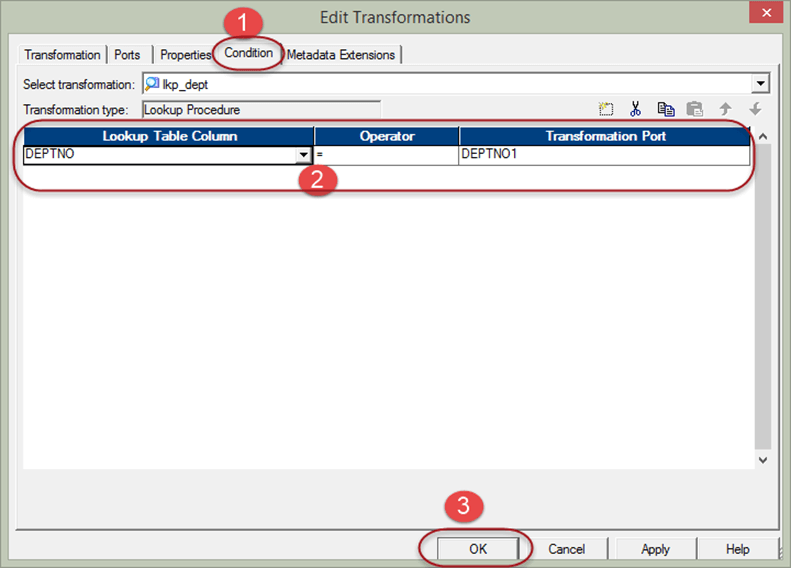
**Step 5** – Drag and drop DEPTNO column from source qualifier to the lookup transformation, this will create a new column DEPTNO1 in lookup transformation. Then link the DNAME column from lookup transformation to the target table.

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

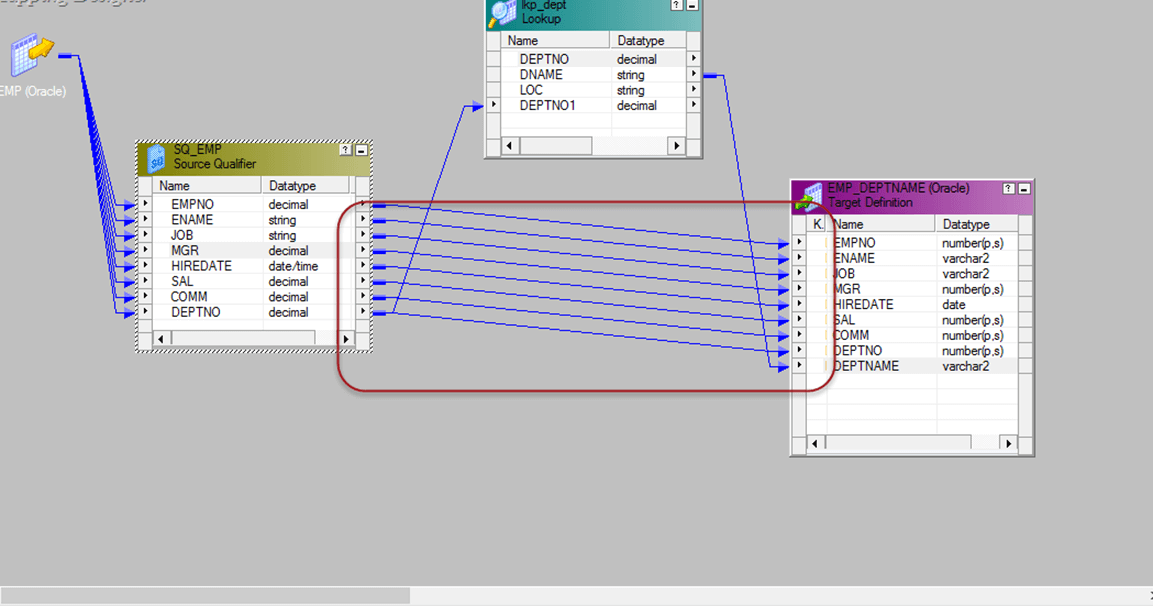


**Step 6** – Double click on the lookup transformation. Then in the edit transformation window

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



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



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.

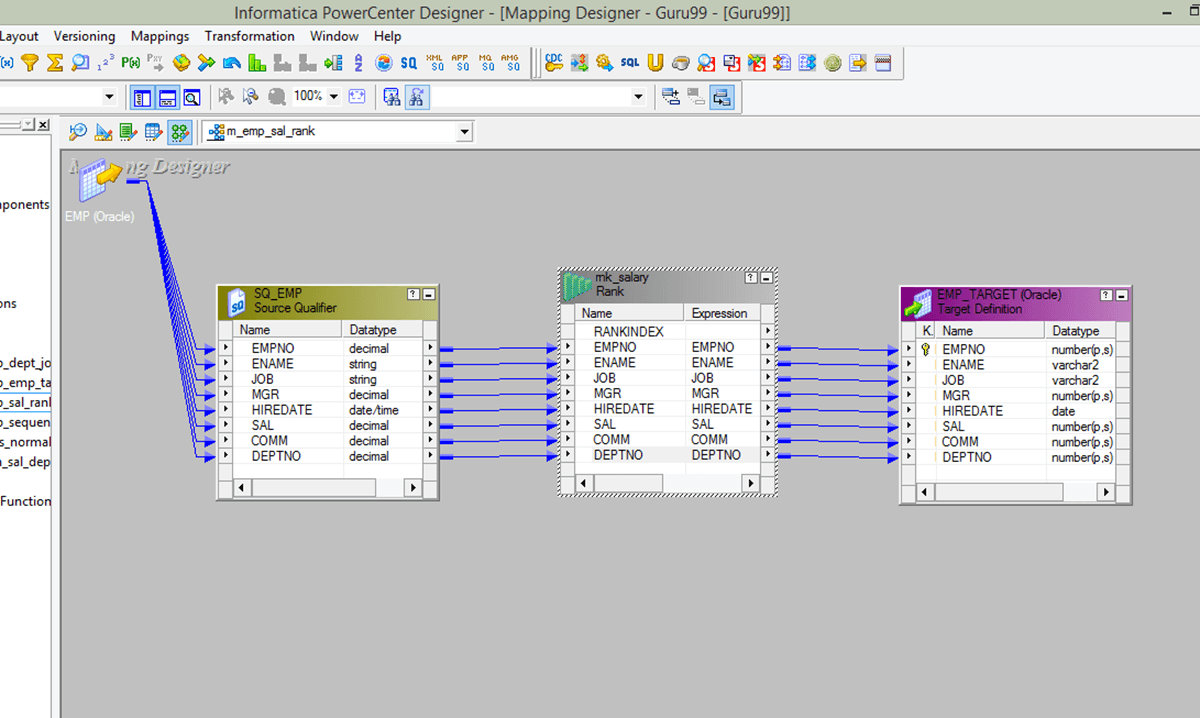
For example, a lookup transformation which fetches employee details based on employee number can be used at multiple mappings wherever employee details are required.

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

**How to Create Reusable Transformation**

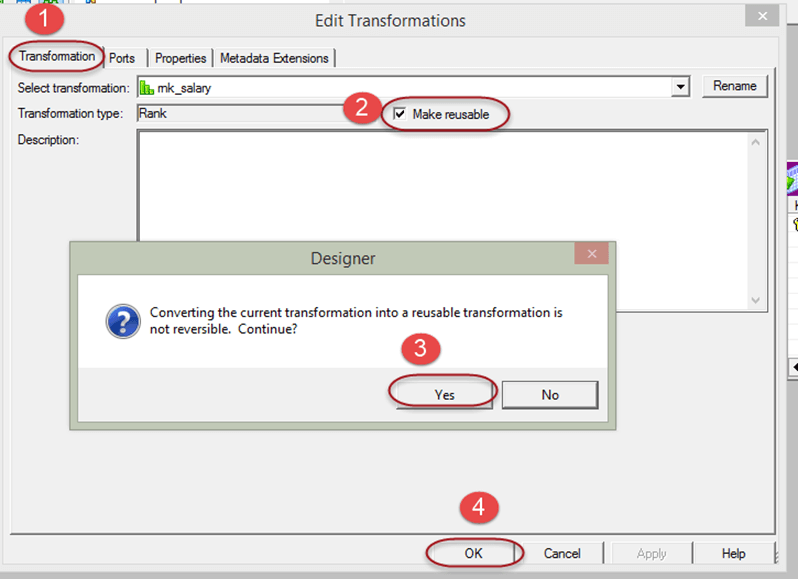
Following are the steps to create Reusable Transformation

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



**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.



This will make the transformation reusable.

**What is Normalizer Transformation?**

Normalizer is an active transformation, used to convert a single row into multiple rows and vice versa. It is a smart way of representing your data in more organized manner.

If in a single row there is repeating data in multiple columns, then it can be split into multiple rows. Sometimes we have data in multiple occurring columns. For example

| **Student Name** | **Class 9 Score** | **Class 10 Score** | **Class 11 Score** | **Class 12 Score** |
| --- | --- | --- | --- | --- |
| Student 1 | 50 | 60 | 65 | 80 |
| Student 2 | 70 | 64 | 83 | 77 |

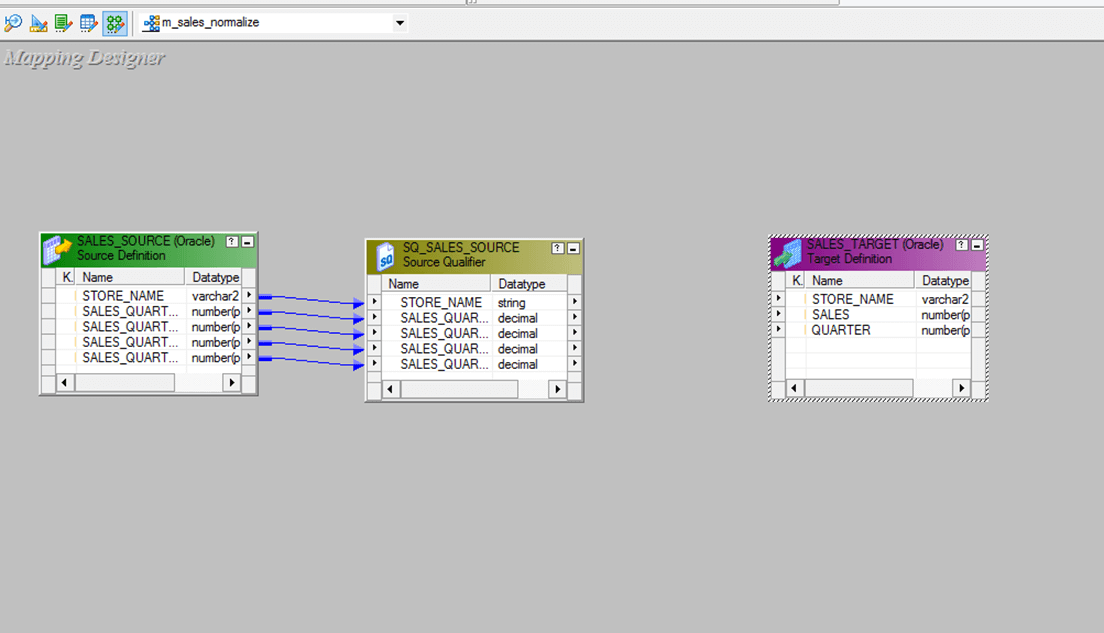
In this case, the class score column is repeating in four columns. Using normalizer, we can split these in the following data set.

| **Student Name** | **Class** | **Score** |
| --- | --- | --- |
| Student 1 | 9 | 50 |
| Student 1 | 10 | 60 |
| Student 1 | 11 | 65 |
| Student 1 | 12 | 80 |
| Student 2 | 9 | 70 |
| Student 2 | 10 | 64 |
| Student 2 | 11 | 83 |
| Student 2 | 12 | 77 |

**Step 1** – Create source table “sales\_source” and target table “sales\_target” using the script and import them in Informatica

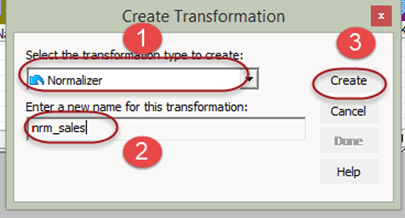
[**Download the above Sales\_Source.txt File**](https://drive.google.com/uc?export=download&id=0ByI5-ZLwpo25aTJ6cDJfTVNZczA)

**Step 2 –**Create a mapping having source “sales\_source” and target table “sales\_target”

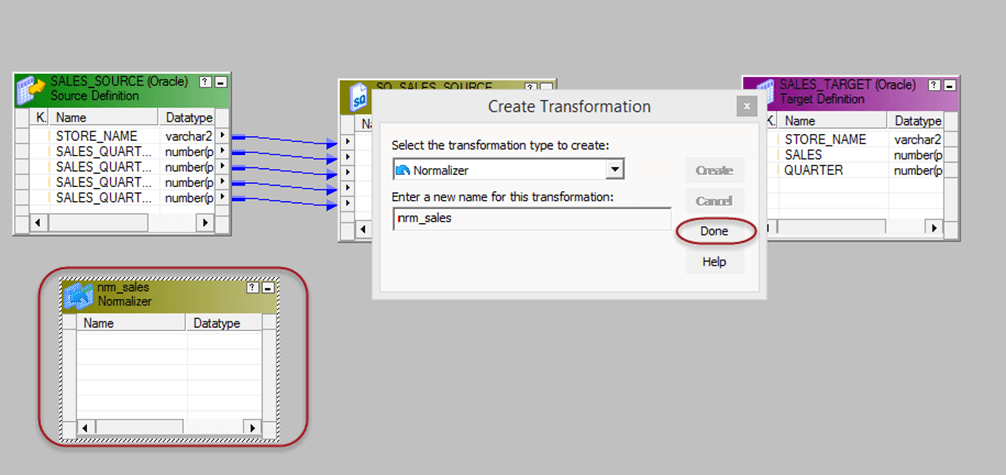


**Step 3** – From the transformation menu create a new transformation

1. Select normalizer as transformation
2. Enter name, “nrm\_sales”
3. Select create option

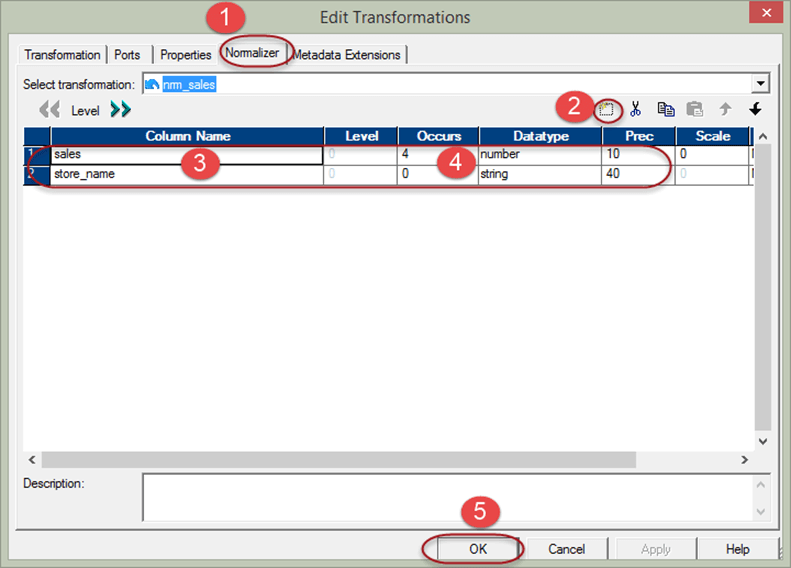


**Step 4** – The transformation will be created, select done option

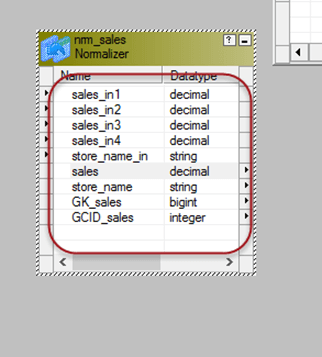


**Step 5** – Double click on the normalizer transformation, then

1. Select normalizer tab
2. Click on icon to create two columns
3. Enter column names
4. Set number of occurrence to 4 for sales and 0 for store name
5. Select OK button

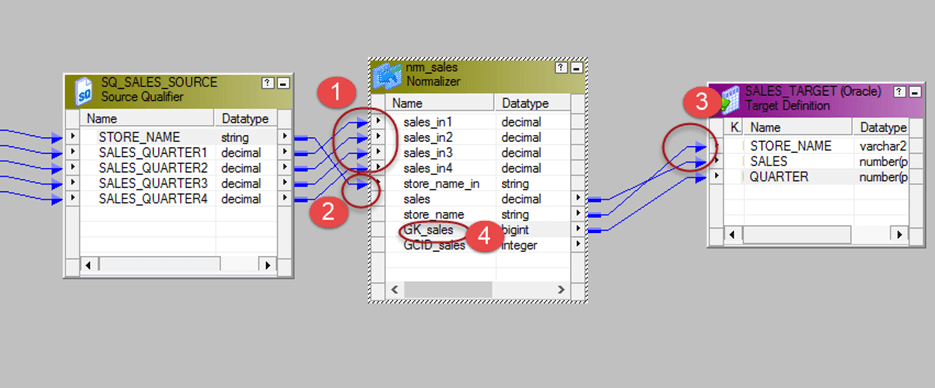


Columns will be generated in the transformation. You will see 4 number of sales column as we set the number of occurrences to 4.



**Step 6** – Then in the mapping

1. Link the four column of source qualifier of the four quarter to the normalizer columns respectively.
2. Link store name column to the normalizer column
3. Link store\_name & sales columns from normalizer to target table
4. Link GK\_sales column from normalizer to target table



Save the mapping and execute it after creating session and workflow. For each quarter sales of a store, a separate row will be created by the normalizer transformation.

The output of our mapping will be like –

| **Store Name** | **Quarter** | **Sales** |
| --- | --- | --- |
| DELHI | 1 | 150 |
| DELHI | 2 | 240 |
| DELHI | 3 | 455 |
| DELHI | 4 | 100 |
| MUMBAI | 1 | 100 |
| MUMBAI | 2 | 500 |
| MUMBAI | 3 | 350 |
| MUMBAI | 4 | 340 |

The source data had repeating columns namely QUARTER1, QUARTER2, QUARTER3, and QUARTER4. With the help of normalizer, we have rearranged the data to fit into a single column of QUARTER and for one source record four records are created in the target.

In this way, you can normalize data and create multiple records for a single source of data.