Database mapping of XBRL instance documents based on the WIP taxonomy

Using Altova MapForce or RaptorXML+XBRL Server

Presented by: Alexander Falk
January 6, 2016

XBRL → Database Mapping of WIP: Overview

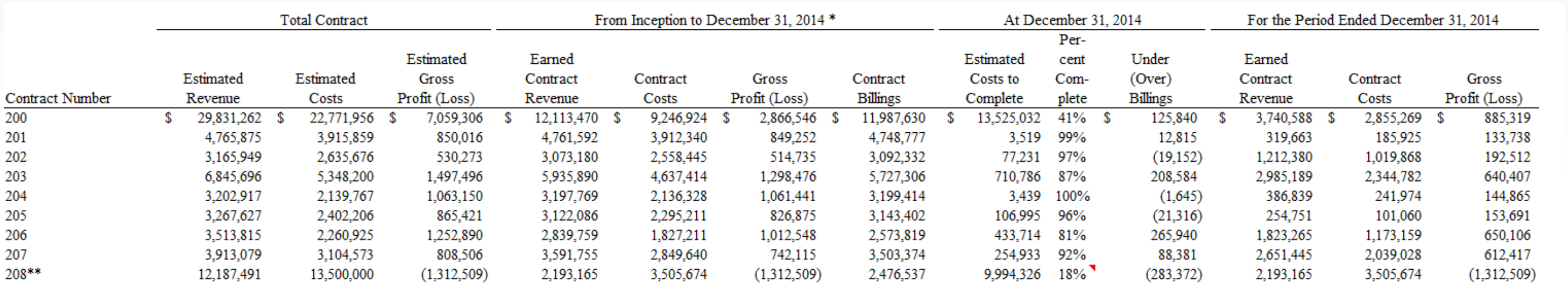
- MapForce can be used to graphically design the XBRL → Database Mapping rules/logic.

- Alternatively, if more control & programming is desired, XBRL instance documents can be analyzed and processed by RaptorXML+XBRL Server and data can be extracted into a database via Python scripts

- If desired, FlowForce and MapForce Server can be used as a workflow & mapping engine to automate either of the above processes using date/time and/or event triggers (e.g. a WIP instance document arriving in a certain directory)

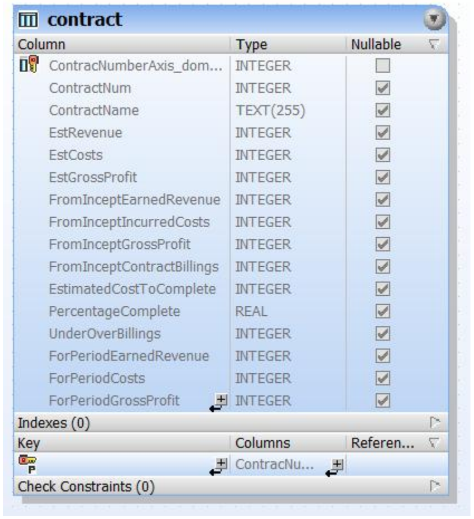
A sample target database

- For the purpose of this demonstration, we created a very simple target DB that has just one table with columns that closely model the WIP spreadsheet:



	Total Contract			From Inception to December 31, 2014 *				At December 31, 2014			For the Period Ended December 31, 2014		
Contract Number	Estimated Revenue	Estimated Costs	Estimated Gross Profit (Loss)	Earned Contract Revenue	Contract Costs	Gross Profit (Loss)	Contract Billings	Estimated Costs to Complete	Per-cent Com-plete	Under (Over) Billings	Earned Contract Revenue	Contract Costs	Gross Profit (Loss)
200	$ 29,831,262	$ 22,771,956	$ 7,059,306	$ 12,113,470	$ 9,246,924	$ 2,866,546	$ 11,987,630	$ 13,525,032	41%	$ 125,840	$ 3,740,588	$ 2,855,269	$ 885,319
201	4,765,875	3,915,859	850,016	4,761,592	3,912,340	849,252	4,748,777	3,519	99%	12,815	319,663	185,925	133,738
202	3,165,949	2,635,676	530,273	3,073,180	2,558,445	514,735	3,092,332	77,231	97%	(19,152)	1,212,380	1,019,868	192,512
203	6,845,696	5,348,200	1,497,496	5,935,890	4,637,414	1,298,476	5,727,306	710,786	87%	208,584	2,985,189	2,344,782	640,407
204	3,202,917	2,139,767	1,063,150	3,197,769	2,136,328	1,061,441	3,199,414	3,439	100%	(1,645)	386,839	241,974	144,865
205	3,267,627	2,402,206	865,421	3,122,086	2,295,211	826,875	3,143,402	106,995	96%	(21,316)	254,751	101,060	153,691
206	3,513,815	2,260,925	1,252,890	2,839,759	1,827,211	1,012,548	2,573,819	433,714	81%	265,940	1,823,265	1,173,159	650,106
207	3,913,079	3,104,573	808,506	3,591,755	2,849,640	742,115	3,503,374	254,933	92%	88,381	2,651,445	2,039,028	612,417
208**	12,187,491	13,500,000	(1,312,509)	2,193,165	3,505,674	(1,312,509)	2,476,537	9,994,326	18%	(283,372)	2,193,165	3,505,674	(1,312,509)

- Clearly, any real-world database will be more complex and have a relational database model involving multiple tables, but the same technology we will demonstrate here can be applied to mapping the data from XBRL to any number of relational tables.



contract

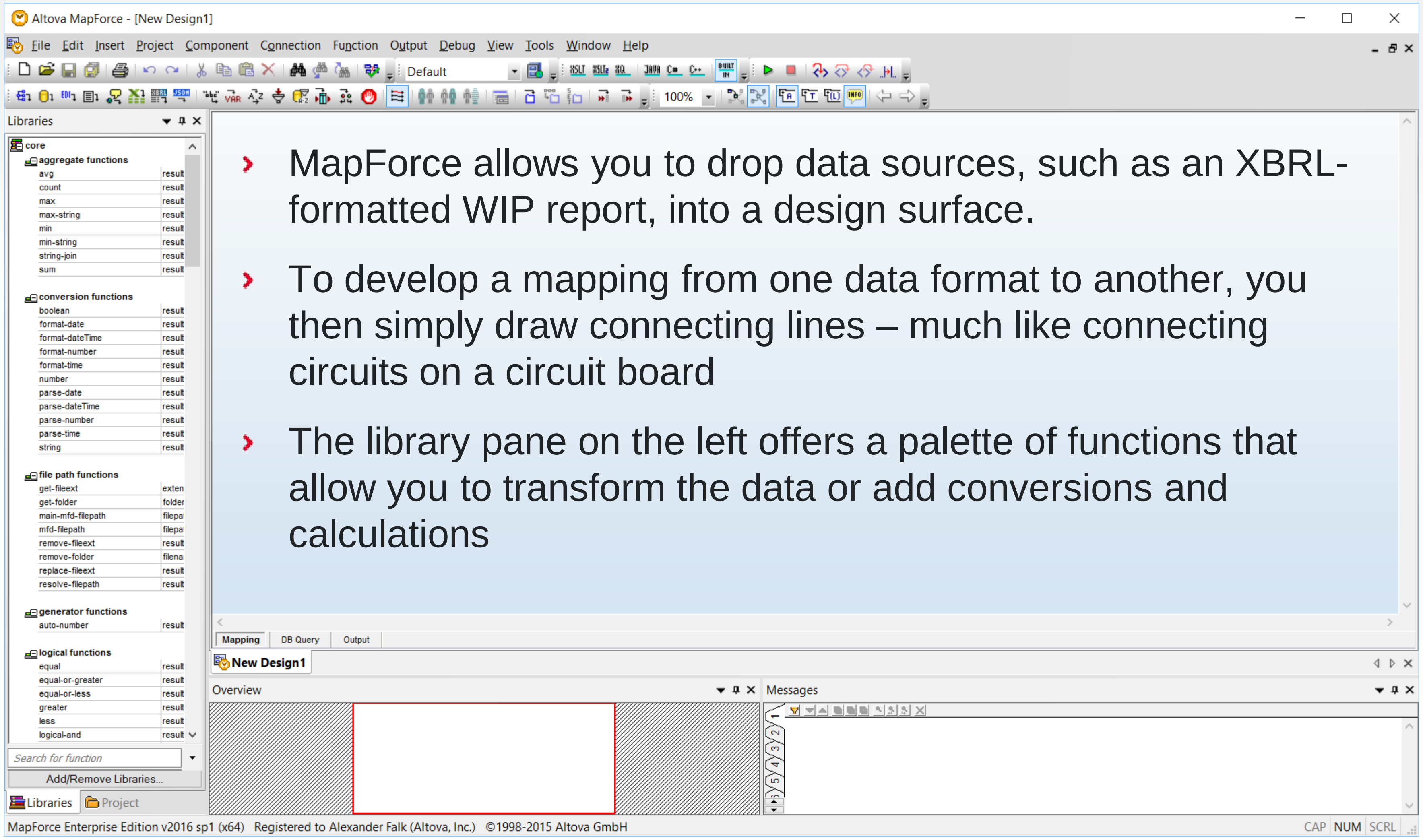
Column	Type	Nullable
ContracNumberAxis_dom...	INTEGER	☐
ContractNum	INTEGER	✓
ContractName	TEXT(255)	✓
EstRevenue	INTEGER	✓
EstCosts	INTEGER	✓
EstGrossProfit	INTEGER	✓
FromInceptEarnedRevenue	INTEGER	✓
FromInceptIncurredCosts	INTEGER	✓
FromInceptGrossProfit	INTEGER	✓
FromInceptContractBillings	INTEGER	✓
EstimatedCostToComplete	INTEGER	✓
PercentageComplete	REAL	✓
UnderOverBillings	INTEGER	✓
ForPeriodEarnedRevenue	INTEGER	✓
ForPeriodCosts	INTEGER	✓
ForPeriodGrossProfit	INTEGER	✓

Indexes (0)

Key	Columns	Referen...
P	ContracNu...	

Check Constraints (0)

Quick introduction to MapForce

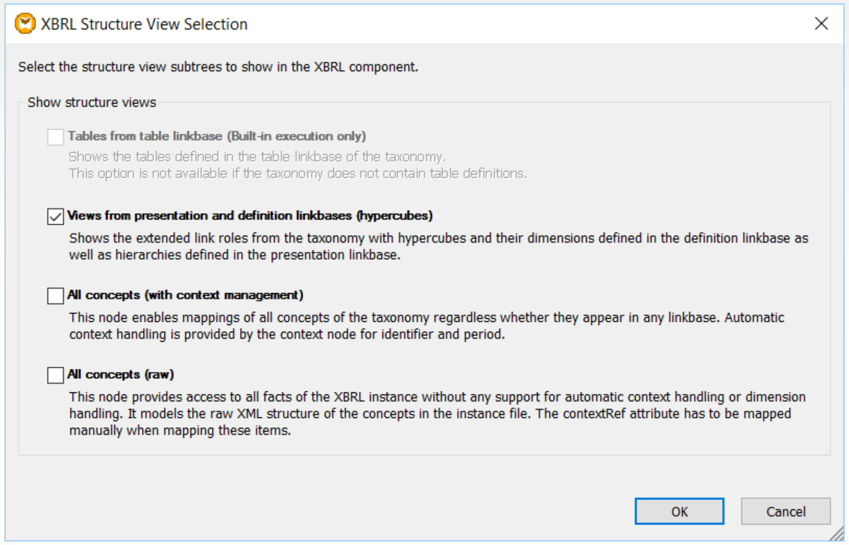


- MapForce allows you to drop data sources, such as an XBRL-formatted WIP report, into a design surface.
- To develop a mapping from one data format to another, you then simply draw connecting lines – much like connecting circuits on a circuit board
- The library pane on the left offers a palette of functions that allow you to transform the data or add conversions and calculations

MapForce Enterprise Edition v2016 sp1 (x64) Registered to Alexander Falk (Altova, Inc.) ©1998-2015 Altova GmbH

Adding an XBRL data source

- When you use an XBRL formatted WIP report as the data source in MapForce, the WIP Taxonomy is automatically processed and you can show the structure of the XBRL WIP either as raw concepts or based on the XBRL financial table defined.



XBRL Structure View Selection

Select the structure view subtrees to show in the XBRL component.

Show structure views

[ ] Tables from table linkbase (Built-in execution only)
Shows the tables defined in the table linkbase of the taxonomy.
This option is not available if the taxonomy does not contain table definitions.

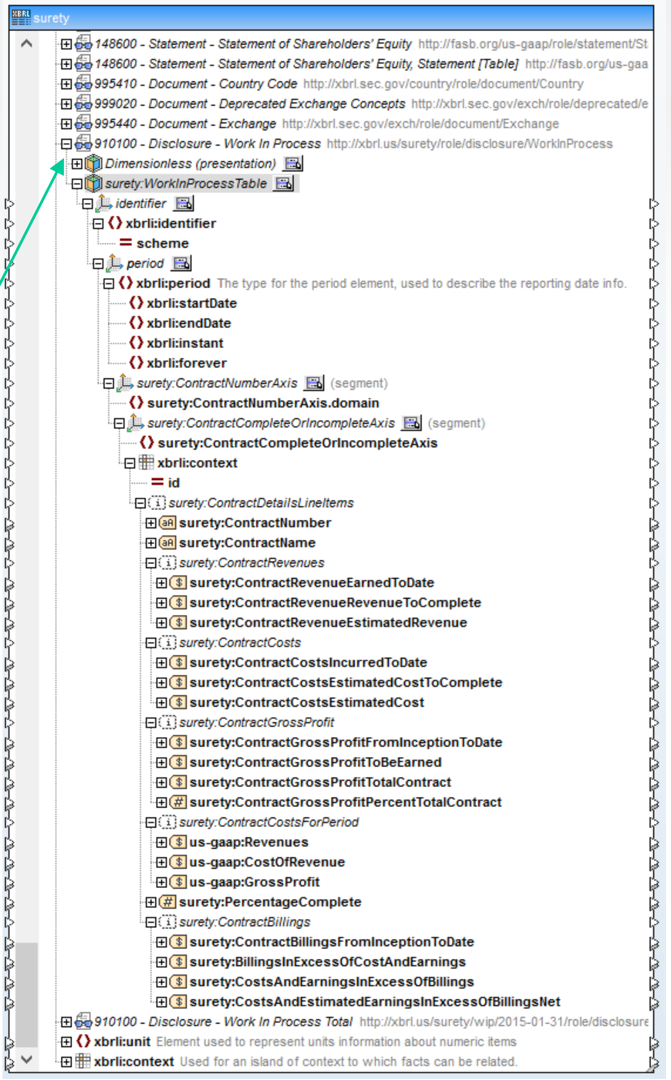
[✓] Views from presentation and definition linkbases (hypercubes)
Shows the extended link roles from the taxonomy with hypercubes and their dimensions defined in the definition linkbase as well as hierarchies defined in the presentation linkbase.

[ ] All concepts (with context management)
This node enables mappings of all concepts of the taxonomy regardless whether they appear in any linkbase. Automatic context handling is provided by the context node for identifier and period.

[ ] All concepts (raw)
This node provides access to all facts of the XBRL instance without any support for automatic context handling or dimension handling. It models the raw XML structure of the concepts in the instance file. The contextRef attribute has to be mapped manually when mapping these items.

Locate and expand the surety:WorkInProcessTable

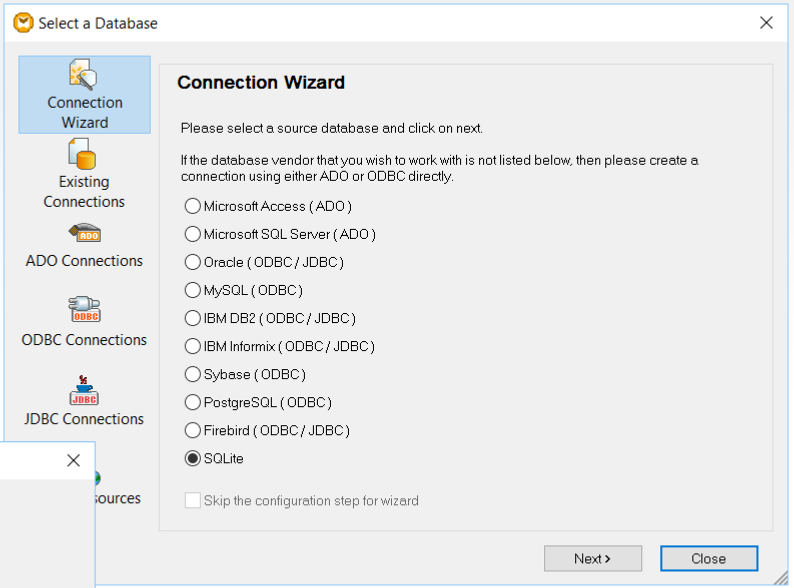
- The WIP taxonomy is based on and includes the full US-GAAP taxonomy, so there are many financial statement presentations included
- Scroll down to the bottom of the XBRL control in MapForce and locate the **910100 – Disclosure – Work in Process** presentation
- Expand that node by clicking on the plus in front, then click on the **surety:WorkInProcessTable** XBRL table and hit the * key on the numeric keypad to expand all its children
- This is the graphical representation of the source XBRL instance data that we will be mapping from. On the right side of each element/fact is a triangle where you can start connections from the source to the target.



```
surety
⊞ 148600 - Statement - Statement of Shareholders' Equity  http://fasb.org/us-gaap/role/statement/St
⊞ 148600 - Statement - Statement of Shareholders' Equity, Statement [Table]  http://fasb.org/us-gaa
⊞ 995410 - Document - Country Code  http://xbrl.sec.gov/country/role/document/Country
⊞ 999020 - Document - Deprecated Exchange Concepts  http://xbrl.sec.gov/exch/role/deprecated/e
⊞ 995440 - Document - Exchange  http://xbrl.sec.gov/exch/role/document/Exchange
⊟ 910100 - Disclosure - Work In Process  http://xbrl.us/surety/role/disclosure/WorkInProcess
  ⊞ Dimensionless (presentation)
  ⊟ surety:WorkInProcessTable
    ⊟ identifier
      ⊟ xbrli:identifier
          scheme
      ⊟ period
        ⊟ xbrli:period  The type for the period element, used to describe the reporting date info.
            xbrli:startDate
            xbrli:endDate
            xbrli:instant
            xbrli:forever
        ⊟ surety:ContractNumberAxis  (segment)
            surety:ContractNumberAxis.domain
          ⊟ surety:ContractCompleteOrIncompleteAxis  (segment)
              surety:ContractCompleteOrIncompleteAxis
            ⊟ xbrli:context
                id
              ⊟ surety:ContractDetailsLineItems
                ⊞ surety:ContractNumber
                ⊞ surety:ContractName
                ⊟ surety:ContractRevenues
                  ⊞ surety:ContractRevenueEarnedToDate
                  ⊞ surety:ContractRevenueRevenueToComplete
                  ⊞ surety:ContractRevenueEstimatedRevenue
                ⊟ surety:ContractCosts
                  ⊞ surety:ContractCostsIncurredToDate
                  ⊞ surety:ContractCostsEstimatedCostToComplete
                  ⊞ surety:ContractCostsEstimatedCost
                ⊟ surety:ContractGrossProfit
                  ⊞ surety:ContractGrossProfitFromInceptionToDate
                  ⊞ surety:ContractGrossProfitToBeEarned
                  ⊞ surety:ContractGrossProfitTotalContract
                  ⊞ surety:ContractGrossProfitPercentTotalContract
                ⊟ surety:ContractCostsForPeriod
                  ⊞ us-gaap:Revenues
                  ⊞ us-gaap:CostOfRevenue
                  ⊞ us-gaap:GrossProfit
                ⊞ surety:PercentageComplete
                ⊟ surety:ContractBillings
                  ⊞ surety:ContractBillingsFromInceptionToDate
                  ⊞ surety:BillingsInExcessOfCostAndEarnings
                  ⊞ surety:CostsAndEarningsInExcessOfBillings
                  ⊞ surety:CostsAndEstimatedEarningsInExcessOfBillingsNet
⊞ 910100 - Disclosure - Work In Process Total  http://xbrl.us/surety/wip/2015-01-31/role/disclosure
⊞ xbrli:unit  Element used to represent units information about numeric items
⊞ xbrli:context  Used for an island of context to which facts can be related.
```

Adding a database target

- Adding a database target involves first connecting to a database source using the connection wizard
- Then you select which tables from that database you want to use in your mapping project – we‘ll pick „contract“


Select a Database

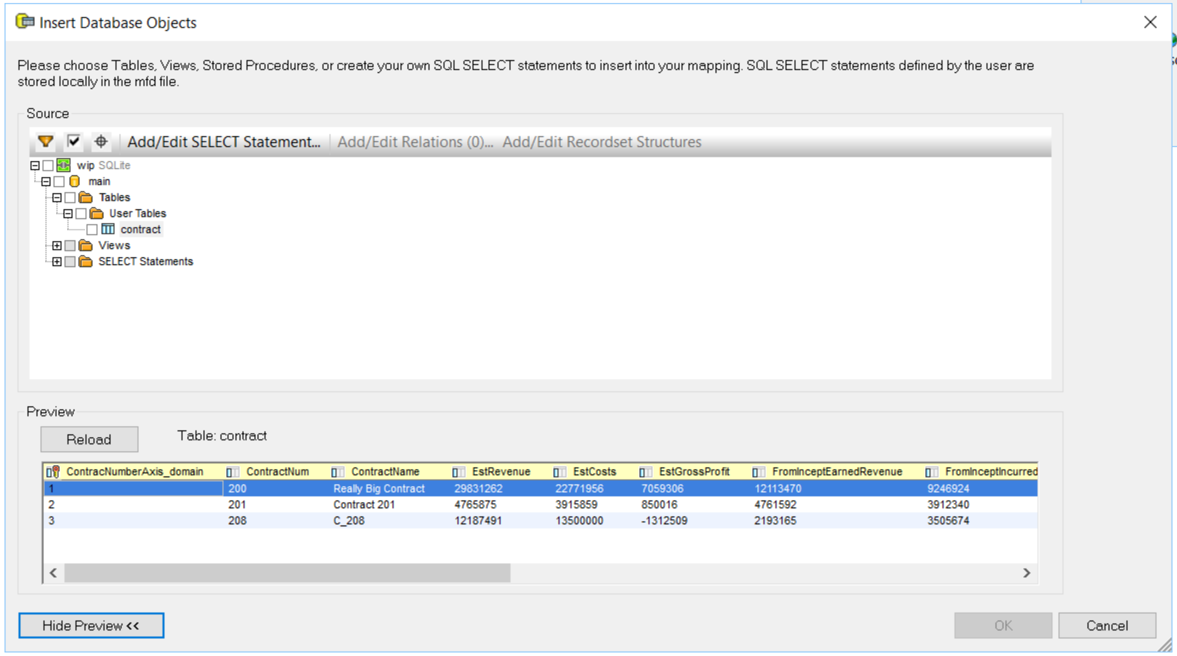
Connection Wizard

Please select a source database and click on next.

If the database vendor that you wish to work with is not listed below, then please create a connection using either ADO or ODBC directly.

- ○ Microsoft Access (ADO)
- ○ Microsoft SQL Server (ADO)
- ○ Oracle (ODBC / JDBC)
- ○ MySQL (ODBC)
- ○ IBM DB2 (ODBC / JDBC)
- ○ IBM Informix (ODBC / JDBC)
- ○ Sybase (ODBC)
- ○ PostgreSQL (ODBC)
- ○ Firebird (ODBC / JDBC)
- ● SQLite

☐ Skip the configuration step for wizard


Insert Database Objects

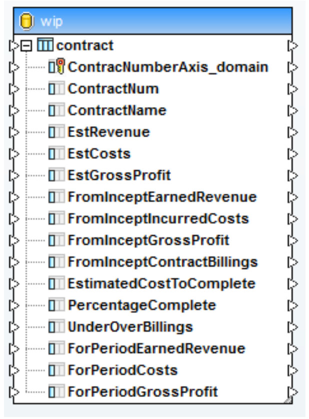
Please choose Tables, Views, Stored Procedures, or create your own SQL SELECT statements to insert into your mapping. SQL SELECT statements defined by the user are stored locally in the mfd file.

Source
Add/Edit SELECT Statement...
Add/Edit Relations (0)...
Add/Edit Recordset Structures
wip SQLite
main
Tables
User Tables
contract
Views
SELECT Statements
Preview
Reload

Table: contract

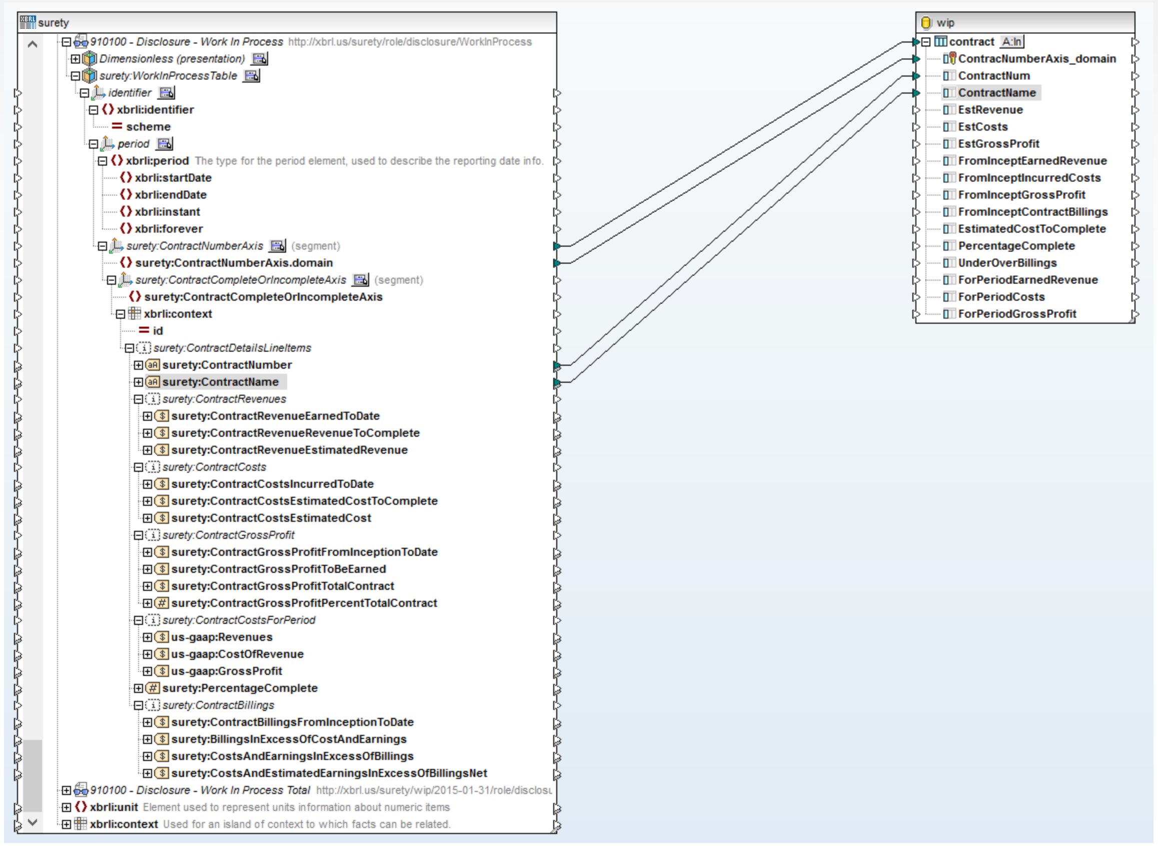
ContracNumberAxis_domain	ContractNum	ContractName	EstRevenue	EstCosts	EstGrossProfit	FromInceptEarnedRevenue	FromInceptIncurred
1	200	Really Big Contract	29831262	22771956	7059306	12113470	9246924
2	201	Contract 201	4765875	3915859	850016	4761592	3912340
3	208	C_208	12187491	13500000	-1312509	2193165	3505674

- This creates a database object in our mapping project that looks like this



Start making connections

- Now we can start making connections from the source to the target to create our mapping
- We‘re focusing on the **ContractNumberAxis** and will connect that to each row in the contract table in the DB
- Start drawing lines from the triangle nodes on the left XBRL object to the triangle nodes on the right database table object...



```
surety
⊟ 910100 - Disclosure - Work In Process  http://xbrl.us/surety/role/disclosure/WorkInProcess
  ⊞ Dimensionless (presentation)
  ⊟ surety:WorkInProcessTable
    ⊟ identifier
      ⊟ xbrli:identifier
          scheme
      ⊟ period
        ⊟ xbrli:period  The type for the period element, used to describe the reporting date info.
            xbrli:startDate
            xbrli:endDate
            xbrli:instant
            xbrli:forever
        ⊟ surety:ContractNumberAxis  (segment)
            surety:ContractNumberAxis.domain
          ⊟ surety:ContractCompleteOrIncompleteAxis  (segment)
              surety:ContractCompleteOrIncompleteAxis
            ⊟ xbrli:context
                id
              ⊟ surety:ContractDetailsLineItems
                ⊞ surety:ContractNumber
                ⊞ surety:ContractName
                ⊟ surety:ContractRevenues
                  ⊞ surety:ContractRevenueEarnedToDate
                  ⊞ surety:ContractRevenueRevenueToComplete
                  ⊞ surety:ContractRevenueEstimatedRevenue
                ⊟ surety:ContractCosts
                  ⊞ surety:ContractCostsIncurredToDate
                  ⊞ surety:ContractCostsEstimatedCostToComplete
                  ⊞ surety:ContractCostsEstimatedCost
                ⊟ surety:ContractGrossProfit
                  ⊞ surety:ContractGrossProfitFromInceptionToDate
                  ⊞ surety:ContractGrossProfitToBeEarned
                  ⊞ surety:ContractGrossProfitTotalContract
                  ⊞ surety:ContractGrossProfitPercentTotalContract
                ⊟ surety:ContractCostsForPeriod
                  ⊞ us-gaap:Revenues
                  ⊞ us-gaap:CostOfRevenue
                  ⊞ us-gaap:GrossProfit
                ⊞ surety:PercentageComplete
                ⊟ surety:ContractBillings
                  ⊞ surety:ContractBillingsFromInceptionToDate
                  ⊞ surety:BillingsInExcessOfCostAndEarnings
                  ⊞ surety:CostsAndEarningsInExcessOfBillings
                  ⊞ surety:CostsAndEstimatedEarningsInExcessOfBillingsNet
⊞ 910100 - Disclosure - Work In Process Total  http://xbrl.us/surety/wip/2015-01-31/role/disclosu
⊞ xbrli:unit  Element used to represent units information about numeric items
⊞ xbrli:context  Used for an island of context to which facts can be related.
```

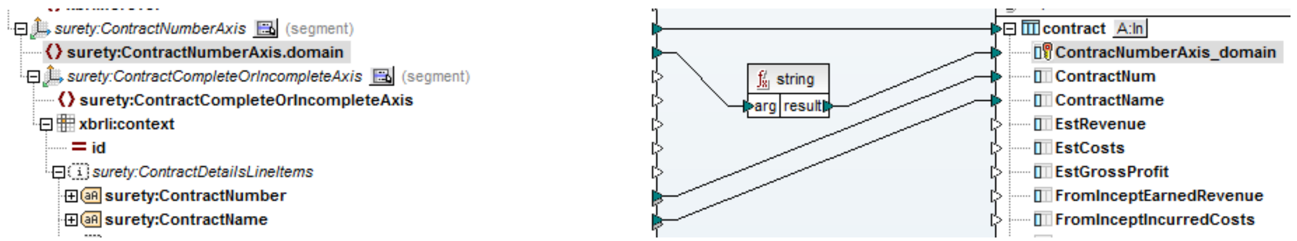
```
wip
⊟ contract  A:In
    ContracNumberAxis_domain
    ContractNum
    ContractName
    EstRevenue
    EstCosts
    EstGrossProfit
    FromInceptEarnedRevenue
    FromInceptIncurredCosts
    FromInceptGrossProfit
    FromInceptContractBillings
    EstimatedCostToComplete
    PercentageComplete
    UnderOverBillings
    ForPeriodEarnedRevenue
    ForPeriodCosts
    ForPeriodGrossProfit
```

Add datatype conversions where necessary

- Sometimes you may find that the datatypes in the XBRL instance and in your target database are not compatible and you’ll get an error message in the mapping validation:

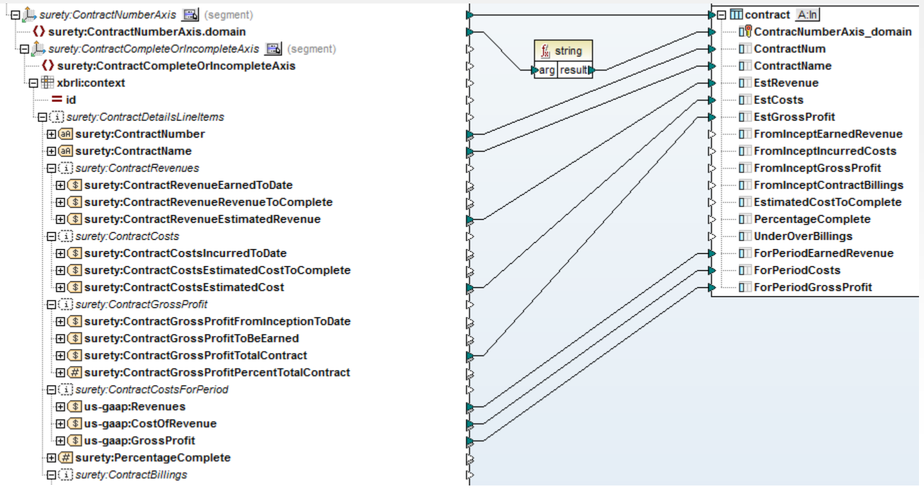
```
New Design1: Mapping validation failed - 1 error(s), 0 warning(s)
       surety:ContractNumberAxis.domain => ContracNumberAxis_domain: Incompatible datatypes.
       There are no valid values of the source type that are valid values of the target type.
                   Source: mf:node  Target: sqlite:integer
```

- If that happens, add a manual conversion from the function library on the left – sometimes it is even more practical to convert to an intermediate datatype, like string



Make more connections, then start looking at the output

- MapForce includes on-demand output preview as well as an interactive visual debugger that makes developing these mappings very easy
- Once you‘ve made a few more connections, it is time to start looking at the output


surety:ContractNumberAxis (segment)
surety:ContractNumberAxis.domain
surety:ContractCompleteOrIncompleteAxis (segment)
surety:ContractCompleteOrIncompleteAxis
xbrli:context
id
surety:ContractDetailsLineItems
surety:ContractNumber
surety:ContractName
surety:ContractRevenues
surety:ContractRevenueEarnedToDate
surety:ContractRevenueRevenueToComplete
surety:ContractRevenueEstimatedRevenue
surety:ContractCosts
surety:ContractCostsIncurredToDate
surety:ContractCostsEstimatedCostToComplete
surety:ContractCostsEstimatedCost
surety:ContractGrossProfit
surety:ContractGrossProfitFromInceptionToDate
surety:ContractGrossProfitToBeEarned
surety:ContractGrossProfitTotalContract
surety:ContractGrossProfitPercentTotalContract
surety:ContractCostsForPeriod
us-gaap:Revenues
us-gaap:CostOfRevenue
us-gaap:GrossProfit
surety:PercentageComplete
surety:ContractBillings
string
arg
result
contract A:In
ContracNumberAxis_domain
ContractNum
ContractName
EstRevenue
EstCosts
EstGrossProfit
FromInceptEarnedRevenue
FromInceptIncurredCosts
FromInceptGrossProfit
FromInceptContractBillings
EstimatedCostToComplete
PercentageComplete
UnderOverBillings
ForPeriodEarnedRevenue
ForPeriodCosts
ForPeriodGrossProfit

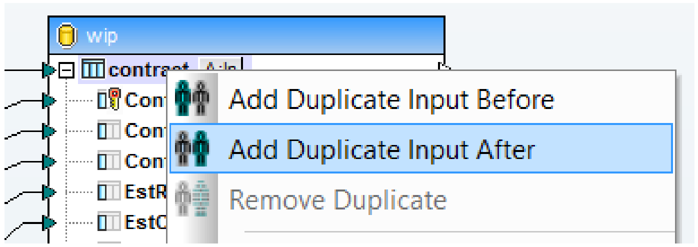
- Since our target is a database, the output preview will be SQL commands that will be executed against the database server:

```
INSERT INTO "contract" ("ContractNum", "ContractName", "EstRevenue", "EstCosts", "EstGrossProfit", "ForPeriodEarnedRevenue", "ForPeriodCosts", "ForPeriodGrossProfit") VALUES (NULL, NULL, 29831262, 22771956, 7059306, NULL, NULL, NULL)
INSERT INTO "contract" ("ContractNum", "ContractName", "EstRevenue", "EstCosts", "EstGrossProfit", "ForPeriodEarnedRevenue", "ForPeriodCosts", "ForPeriodGrossProfit") VALUES (NULL, NULL, 4765875, 3915859, 850016, NULL, NULL, NULL)
INSERT INTO "contract" ("ContractNum", "ContractName", "EstRevenue", "EstCosts", "EstGrossProfit", "ForPeriodEarnedRevenue", "ForPeriodCosts", "ForPeriodGrossProfit") VALUES (NULL, NULL, 12187491, 13500000, -1312509, NULL, NULL, NULL)
INSERT INTO "contract" ("ContractNum", "ContractName", "EstRevenue", "EstCosts", "EstGrossProfit", "ForPeriodEarnedRevenue", "ForPeriodCosts", "ForPeriodGrossProfit") VALUES (200, 'Really Big Contract', NULL, NULL, NULL, 3740588, 2855269, 885319)
INSERT INTO "contract" ("ContractNum", "ContractName", "EstRevenue", "EstCosts", "EstGrossProfit", "ForPeriodEarnedRevenue", "ForPeriodCosts", "ForPeriodGrossProfit") VALUES (201, 'Contract 201', NULL, NULL, NULL, 319663, 185925, 133738)
INSERT INTO "contract" ("ContractNum", "ContractName", "EstRevenue", "EstCosts", "EstGrossProfit", "ForPeriodEarnedRevenue", "ForPeriodCosts", "ForPeriodGrossProfit") VALUES (208, 'C_208', NULL, NULL, NULL, 2193165, 3505674, -1312509)
```

- Clearly, that doesn‘t look quite right yet. We have three contracts in our WIP example XBRL instance, yet we‘re trying to create six rows in the database with only some columns being filled with values and the rest being NULL...

Duration contexts vs. Instant contexts

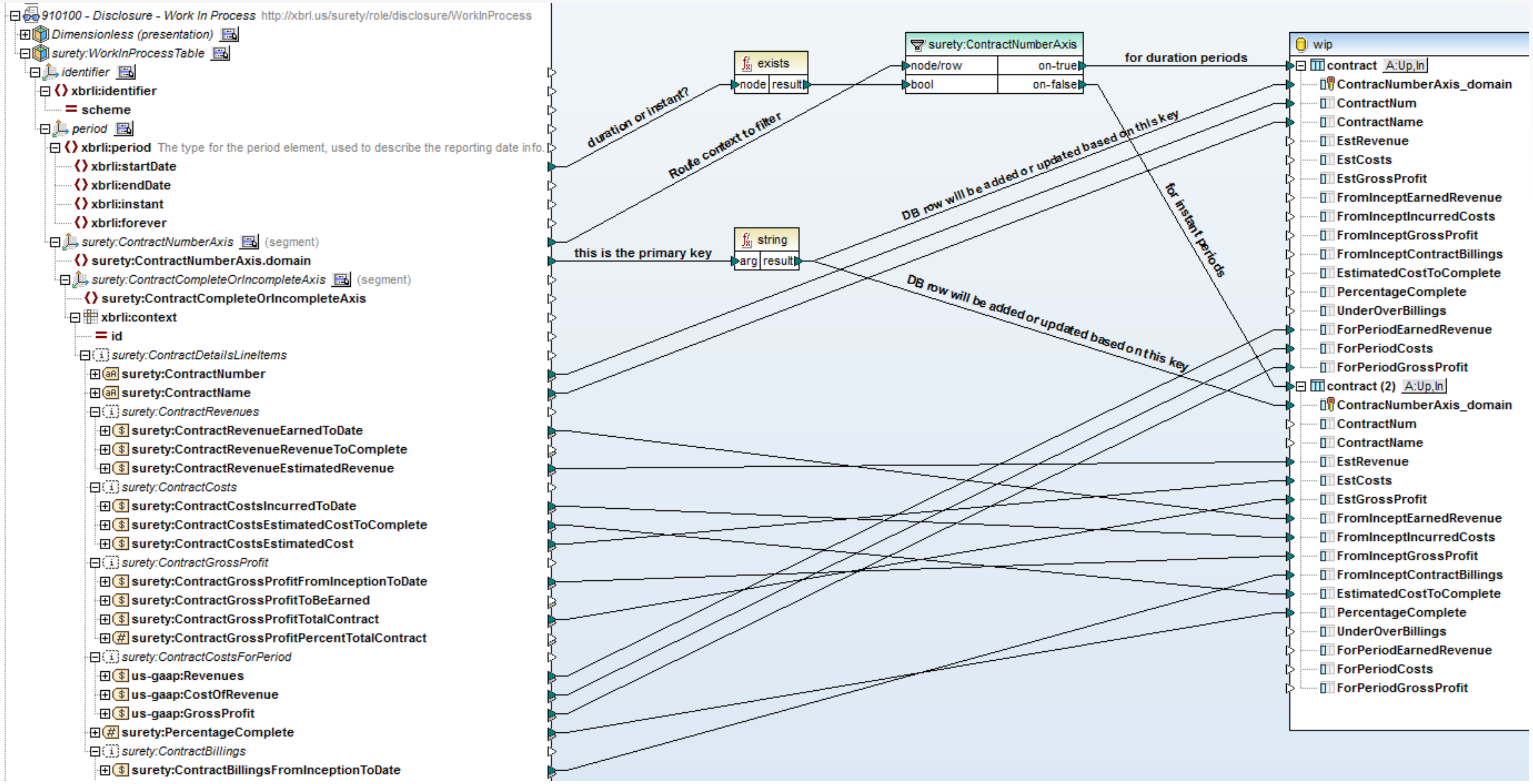
- The reason for these six rows being created is that MapForce has found six *contexts** in the XBRL instance document, and it has mapped each to a row in the database
- So we need to understand the structure of our XBRL document a bit better: for each contract we have an instant context and a duration context in the XBRL file, because each contract is associated with data that is reported as an instant, such as % Complete; or reported as a duration such as Earned Contract Revenue, from Inception to Dec 31, 2014
- Fortunately, we can easily resolve this in MapForce and map them both to the same row in the database using the ContractNumberAxis as the primary key
- To do that, we right click on the contract table in our database object on the right and pick Add Duplicate Input After, which gives us two instances of the table that we can now map to separately from instant vs. duration contexts:



** All facts have an associated context which defines the reporting entity and time period, and it also defines dimensional information.*

Next step: filter contexts and map them differently

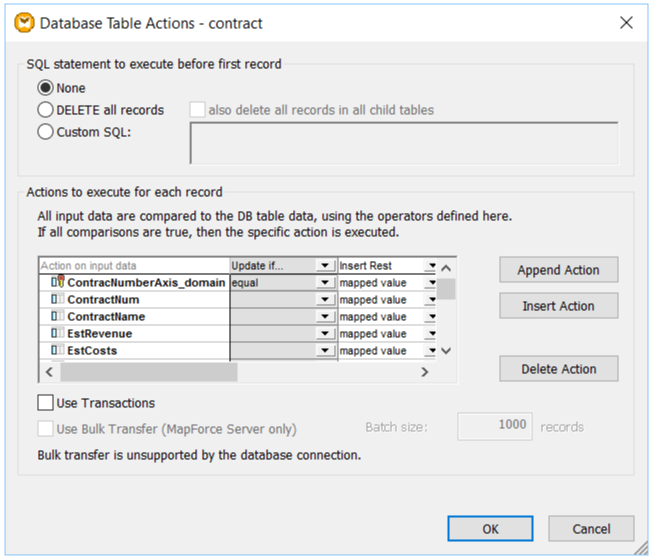
- Now that we have two representations of the table on the right, we can filter XBRL contexts depending on if they are instant or duration and map them differently – as long as we keep the primary key connected to both:



910100 - Disclosure - Work In Process http://xbrl.us/surety/role/disclosure/WorkInProcess
Dimensionless (presentation)
surety:WorkInProcessTable
identifier
xbrli:identifier
scheme
period
xbrli:period The type for the period element, used to describe the reporting date info.
xbrli:startDate
xbrli:endDate
xbrli:instant
xbrli:forever
surety:ContractNumberAxis (segment)
surety:ContractNumberAxis.domain
surety:ContractCompleteOrIncompleteAxis (segment)
surety:ContractCompleteOrIncompleteAxis
xbrli:context
id
surety:ContractDetailsLineItems
surety:ContractNumber
surety:ContractName
surety:ContractRevenues
surety:ContractRevenueEarnedToDate
surety:ContractRevenueRevenueToComplete
surety:ContractRevenueEstimatedRevenue
surety:ContractCosts
surety:ContractCostsIncurredToDate
surety:ContractCostsEstimatedCostToComplete
surety:ContractCostsEstimatedCost
surety:ContractGrossProfit
surety:ContractGrossProfitFromInceptionToDate
surety:ContractGrossProfitToBeEarned
surety:ContractGrossProfitTotalContract
surety:ContractGrossProfitPercentTotalContract
surety:ContractCostsForPeriod
us-gaap:Revenues
us-gaap:CostOfRevenue
us-gaap:GrossProfit
surety:PercentageComplete
surety:ContractBillings
surety:ContractBillingsFromInceptionToDate
exists
node
result
duration or instant?
Route context to filter
surety:ContractNumberAxis
node/row
on-true
bool
on-false
for duration periods
for instant periods
string
arg
result
this is the primary key
DB row will be added or updated based on this key
DB row will be added or updated based on this key
wip
contract A:Up,In
ContracNumberAxis_domain
ContractNum
ContractName
EstRevenue
EstCosts
EstGrossProfit
FromInceptEarnedRevenue
FromInceptIncurredCosts
FromInceptGrossProfit
FromInceptContractBillings
EstimatedCostToComplete
PercentageComplete
UnderOverBillings
ForPeriodEarnedRevenue
ForPeriodCosts
ForPeriodGrossProfit
contract (2) A:Up,In
ContracNumberAxis_domain
ContractNum
ContractName
EstRevenue
EstCosts
EstGrossProfit
FromInceptEarnedRevenue
FromInceptIncurredCosts
FromInceptGrossProfit
FromInceptContractBillings
EstimatedCostToComplete
PercentageComplete
UnderOverBillings
ForPeriodEarnedRevenue
ForPeriodCosts
ForPeriodGrossProfit

Database Table Actions

- Since we‘re now mapping two separate contexts to the same table in the database, we also have to tell MapForce what database table actions to perform based on the primary key
- We do that by clicking the small button to the right of each contract table and specify the database actions in the following dialog
- We simply Insert another Action column before the “Insert All” column and specify that we want to Update the data if the primary key matches, otherwise we insert a new row into the DB
- For all the input data items we then elect to insert the “mapped value” into the database
- This will allow us to map some elements/facts from one context and then grab other facts from a different context as long as the primary key matches


Database Table Actions - contract

SQL statement to execute before first record

[✓ None]
DELETE all records — also delete all records in all child tables
Custom SQL:

Actions to execute for each record

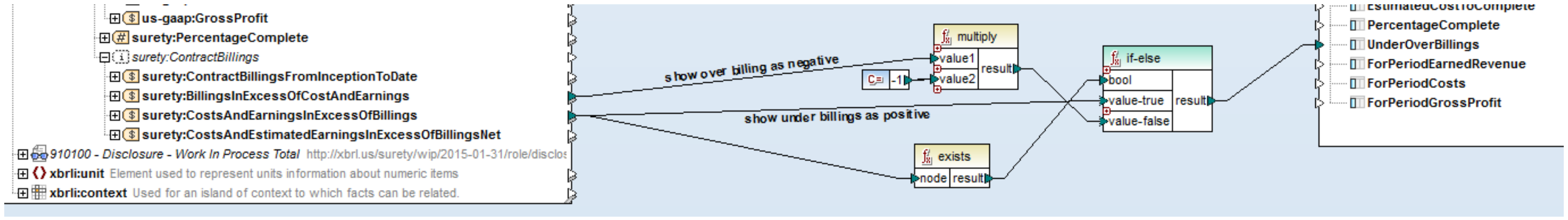
All input data are compared to the DB table data, using the operators defined here.
If all comparisons are true, then the specific action is executed.

Action on input data	Update if...	Insert Rest
ContracNumberAxis_domain	equal	mapped value
ContractNum		mapped value
ContractName		mapped value
EstRevenue		mapped value
EstCosts		mapped value

Use Transactions
Use Bulk Transfer (MapForce Server only) Batch size: 1000 records
Bulk transfer is unsupported by the database connection.

Calculations and if-else statements

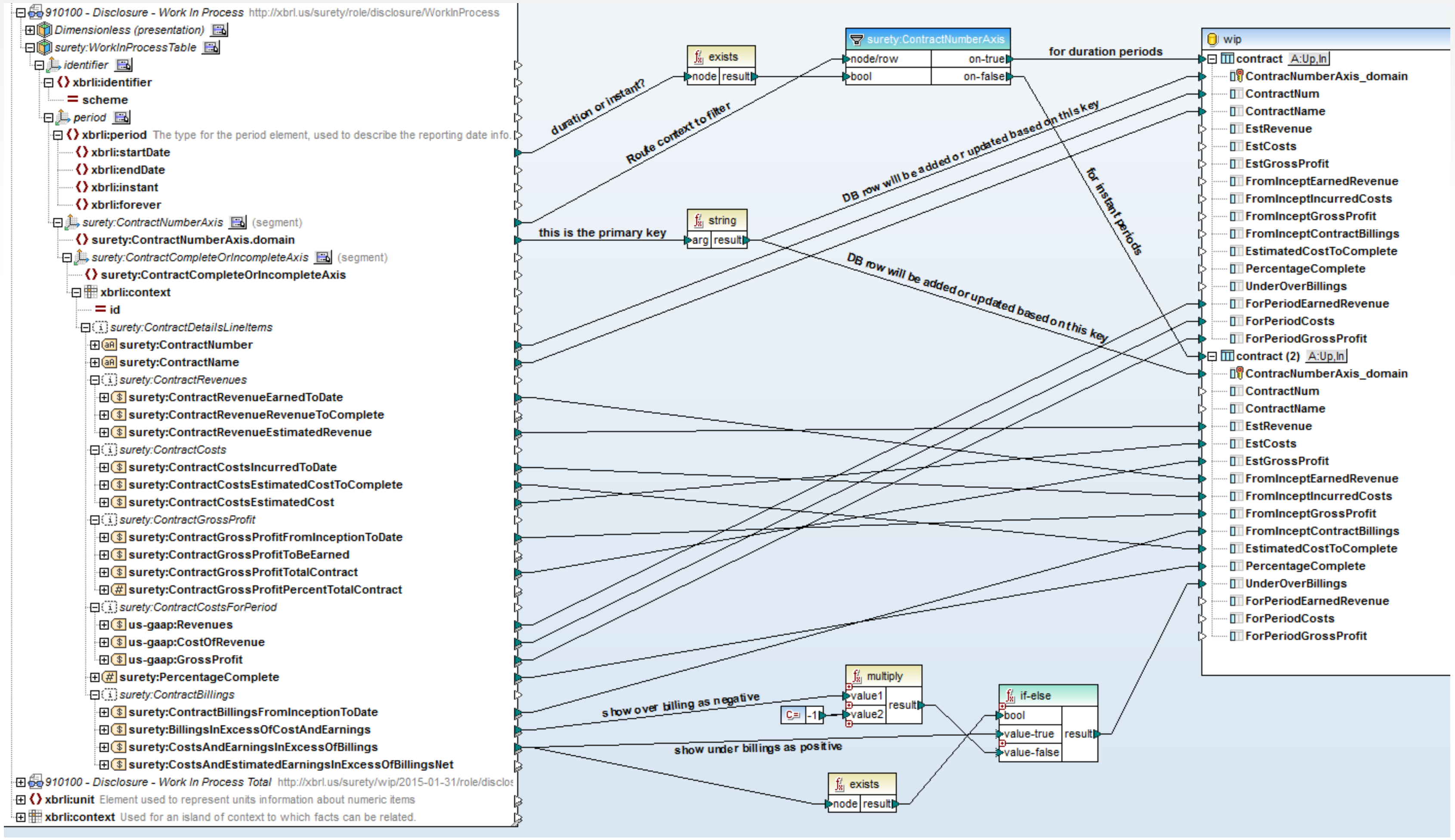
- Last, but not least, you sometimes may need to do more complex calculations or map values differently depending on what inputs you have. We will look at one example of how to do that
- In our XBRL taxonomy we have two separate facts that report billings over or under cost and earnings, but we want to map them to just one database field using either positive or negative values:
  - surety:BillingsInExcessOfCostAndEarnings
  - surety:CostsAndEarningsInExcessOfBillings
- We can do that by multiplying one of them by -1 and then use an if-else statement to map one or the other to the target column in the database, depending on if that fact exists in the source XBRL:



us-gaap:GrossProfit
surety:PercentageComplete
surety:ContractBillings
surety:ContractBillingsFromInceptionToDate
surety:BillingsInExcessOfCostAndEarnings
surety:CostsAndEarningsInExcessOfBillings
surety:CostsAndEstimatedEarningsInExcessOfBillingsNet
910100 - Disclosure - Work In Process Total http://xbrl.us/surety/wip/2015-01-31/role/disclos
xbrli:unit Element used to represent units information about numeric items
xbrli:context Used for an island of context to which facts can be related.
show over billing as negative
show under billings as positive
-1
multiply
value1
value2
result
if-else
bool
value-true
value-false
result
exists
node
result
EstimatedCostToComplete
PercentageComplete
UnderOverBillings
ForPeriodEarnedRevenue
ForPeriodCosts
ForPeriodGrossProfit

Putting it all together

- Here is the complete mapping that we have now created with all components and logic included:

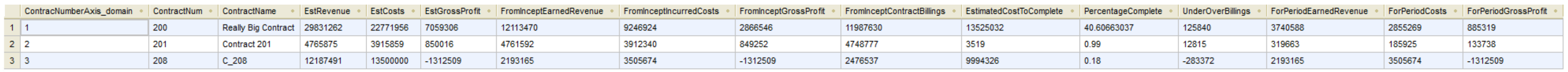


910100 - Disclosure - Work In Process http://xbrl.us/surety/role/disclosure/WorkInProcess
Dimensionless (presentation)
surety:WorkInProcessTable
identifier
xbrli:identifier
scheme
period
xbrli:period The type for the period element, used to describe the reporting date info.
xbrli:startDate
xbrli:endDate
xbrli:instant
xbrli:forever
surety:ContractNumberAxis (segment)
surety:ContractNumberAxis.domain
surety:ContractCompleteOrIncompleteAxis (segment)
surety:ContractCompleteOrIncompleteAxis
xbrli:context
id
surety:ContractDetailsLineItems
surety:ContractNumber
surety:ContractName
surety:ContractRevenues
surety:ContractRevenueEarnedToDate
surety:ContractRevenueRevenueToComplete
surety:ContractRevenueEstimatedRevenue
surety:ContractCosts
surety:ContractCostsIncurredToDate
surety:ContractCostsEstimatedCostToComplete
surety:ContractCostsEstimatedCost
surety:ContractGrossProfit
surety:ContractGrossProfitFromInceptionToDate
surety:ContractGrossProfitToBeEarned
surety:ContractGrossProfitTotalContract
surety:ContractGrossProfitPercentTotalContract
surety:ContractCostsForPeriod
us-gaap:Revenues
us-gaap:CostOfRevenue
us-gaap:GrossProfit
surety:PercentageComplete
surety:ContractBillings
surety:ContractBillingsFromInceptionToDate
surety:BillingsInExcessOfCostAndEarnings
surety:CostsAndEarningsInExcessOfBillings
surety:CostsAndEstimatedEarningsInExcessOfBillingsNet
910100 - Disclosure - Work In Process Total http://xbrl.us/surety/wip/2015-01-31/role/disclos
xbrli:unit Element used to represent units information about numeric items
xbrli:context Used for an island of context to which facts can be related.
exists
node
result
duration or instant?
Route context to filter
surety:ContractNumberAxis
node/row
on-true
bool
on-false
for duration periods
for instant periods
DB row will be added or updated based on this key
this is the primary key
string
arg
result
DB row will be added or updated based on this key
wip
contract A:Up,In
ContracNumberAxis_domain
ContractNum
ContractName
EstRevenue
EstCosts
EstGrossProfit
FromInceptEarnedRevenue
FromInceptIncurredCosts
FromInceptGrossProfit
FromInceptContractBillings
EstimatedCostToComplete
PercentageComplete
UnderOverBillings
ForPeriodEarnedRevenue
ForPeriodCosts
ForPeriodGrossProfit
contract (2) A:Up,In
ContracNumberAxis_domain
ContractNum
ContractName
EstRevenue
EstCosts
EstGrossProfit
FromInceptEarnedRevenue
FromInceptIncurredCosts
FromInceptGrossProfit
FromInceptContractBillings
EstimatedCostToComplete
PercentageComplete
UnderOverBillings
ForPeriodEarnedRevenue
ForPeriodCosts
ForPeriodGrossProfit
multiply
value1
value2
result
-1
show over billing as negative
if-else
bool
value-true
value-false
result
show under billings as positive
exists
node
result

Looking at the final result

- If we now switch to the Output tab we get a SQL Script that we can execute against the database and we see the following output, which produces precisely three rows in the target database as shown below

```
UPDATE "contract" SET "ContractNum" = 200, "ContractName" = 'Really Big Contract', "ForPeriodEarnedRevenue" = 3740588, "ForPeriodCosts" = 2855269, "ForPeriodGrossProfit" =
885319 WHERE ("contract"."ContracNumberAxis_domain"=1)
-->>> OK. 0 row(s).

INSERT INTO "contract" ("ContracNumberAxis_domain", "ContractNum", "ContractName", "ForPeriodEarnedRevenue", "ForPeriodCosts", "ForPeriodGrossProfit") VALUES (1, 200,
'Really Big Contract', 3740588, 2855269, 885319)
-->>> OK. 1 row(s).

UPDATE "contract" SET "ContractNum" = 201, "ContractName" = 'Contract 201', "ForPeriodEarnedRevenue" = 319663, "ForPeriodCosts" = 185925, "ForPeriodGrossProfit" = 133738
WHERE ("contract"."ContracNumberAxis_domain"=2)
-->>> OK. 0 row(s).

INSERT INTO "contract" ("ContracNumberAxis_domain", "ContractNum", "ContractName", "ForPeriodEarnedRevenue", "ForPeriodCosts", "ForPeriodGrossProfit") VALUES (2, 201,
'Contract 201', 319663, 185925, 133738)
-->>> OK. 1 row(s).

UPDATE "contract" SET "ContractNum" = 208, "ContractName" = 'C_208', "ForPeriodEarnedRevenue" = 2193165, "ForPeriodCosts" = 3505674, "ForPeriodGrossProfit" = -1312509 WHERE
("contract"."ContracNumberAxis_domain"=3)
-->>> OK. 0 row(s).

INSERT INTO "contract" ("ContracNumberAxis_domain", "ContractNum", "ContractName", "ForPeriodEarnedRevenue", "ForPeriodCosts", "ForPeriodGrossProfit") VALUES (3, 208,
'C_208', 2193165, 3505674, -1312509)
-->>> OK. 1 row(s).

UPDATE "contract" SET "EstRevenue" = 29831262, "EstCosts" = 22771956, "EstGrossProfit" = 7059306, "FromInceptEarnedRevenue" = 12113470, "FromInceptIncurredCosts" = 9246924,
"FromInceptGrossProfit" = 2866546, "FromInceptContractBillings" = 11987630, "EstimatedCostToComplete" = 13525032, "PercentageComplete" = 40.60663037, "UnderOverBillings" =
125840 WHERE ("contract"."ContracNumberAxis_domain"=1)
-->>> OK. 1 row(s).

UPDATE "contract" SET "EstRevenue" = 4765875, "EstCosts" = 3915859, "EstGrossProfit" = 850016, "FromInceptEarnedRevenue" = 4761592, "FromInceptIncurredCosts" = 3912340,
"FromInceptGrossProfit" = 849252, "FromInceptContractBillings" = 4748777, "EstimatedCostToComplete" = 3519, "PercentageComplete" = 0.99, "UnderOverBillings" = 12815 WHERE
("contract"."ContracNumberAxis_domain"=2)
-->>> OK. 1 row(s).

UPDATE "contract" SET "EstRevenue" = 12187491, "EstCosts" = 13500000, "EstGrossProfit" = -1312509, "FromInceptEarnedRevenue" = 2193165, "FromInceptIncurredCosts" = 3505674,
"FromInceptGrossProfit" = -1312509, "FromInceptContractBillings" = 2476537, "EstimatedCostToComplete" = 9994326, "PercentageComplete" = 0.18, "UnderOverBillings" = -283372
WHERE ("contract"."ContracNumberAxis_domain"=3)
-->>> OK. 1 row(s).
```



	ContracNumberAxis_domain	ContractNum	ContractName	EstRevenue	EstCosts	EstGrossProfit	FromInceptEarnedRevenue	FromInceptIncurredCosts	FromInceptGrossProfit	FromInceptContractBillings	EstimatedCostToComplete	PercentageComplete	UnderOverBillings	ForPeriodEarnedRevenue	ForPeriodCosts	ForPeriodGrossProfit
1	1	200	Really Big Contract	29831262	22771956	7059306	12113470	9246924	2866546	11987630	13525032	40.60663037	125840	3740588	2855269	885319
2	2	201	Contract 201	4765875	3915859	850016	4761592	3912340	849252	4748777	3519	0.99	12815	319663	185925	133738
3	3	208	C_208	12187491	13500000	-1312509	2193165	3505674	-1312509	2476537	9994326	0.18	-283372	2193165	3505674	-1312509

Summary of MapForce data mapping approach

- As we have demonstrated here, it is easy to develop a data mapping from an XBRL-formatted WIP report to a database using MapForce
- This data mapping can now be applied to any XBRL instance document that uses the same taxonomy!
- The mapping process for new instances can either be done interactively using MapForce itself
- Or it can be automated by using MapForce Server and FlowForce Server to have the mapping be executed either based on a time-trigger or an event-trigger (e.g. when a WIP instance document is received in a certain directory).
- MapForce Server and FlowForce Server can be deployed either in your local IT infrastructure or in the cloud. They are available for Linux and Windows operating systems.

Alternative approach: RaptorXML+XBRL Server

- If more control, a more advance programming logic, or more complexity in the data model is required, or if the number of XBRL-formatted WIP reports to be ingested is huge so that performance optimizations for parallel processing are required, there is an alternative approach that we offer:
  - Altova's RaptorXML+XBRL Server is an XBRL processing engine that is focused on high-speed and parallel processing on modern multi-core CPUs to achieve advanced throughput for XBRL validation
  - RaptorXML+XBRL Server comes with a built-in Python interpreter that allows a developer to add post-validation programming logic

How to get started with RaptorXML+XBRL Server

- We have recently published the full sources for an example XBRL-to-database mapping project that is using RaptorXML+XBRL Server. This example is based on downloading the EDGAR company financial filings from the SEC, processing them, and writing them to a SQL database.
- Using these sample sources can provide a great template for how to process XBRL instance documents for WIP, too:
  1. Download, clone, or fork the sources from GitHub: https://github.com/altova/SECDB
  2. Download and install the **RaptorXML+XBRL Server** software from here: http://www.altova.com/download-trial-server.html
  3. You can request a free 30-day license key-code for all Altova products

Thank You!

For more information, please see our blog and website:

http://blog.altova.com

http://www.altova.com

Altova is a proud member of XBRL.US and the Center for Data Quality