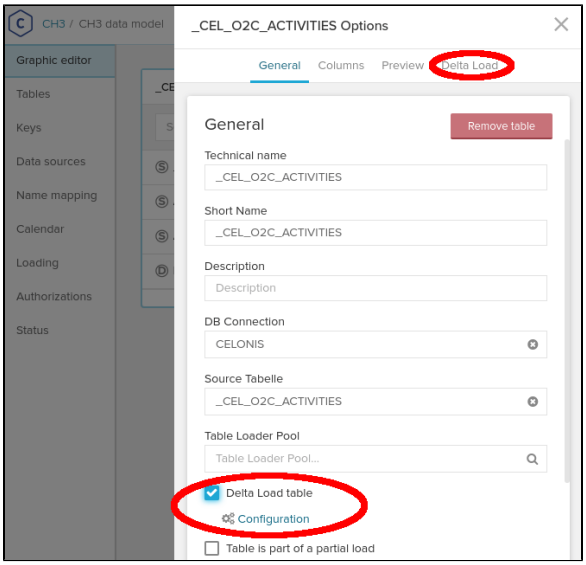
Delta Load

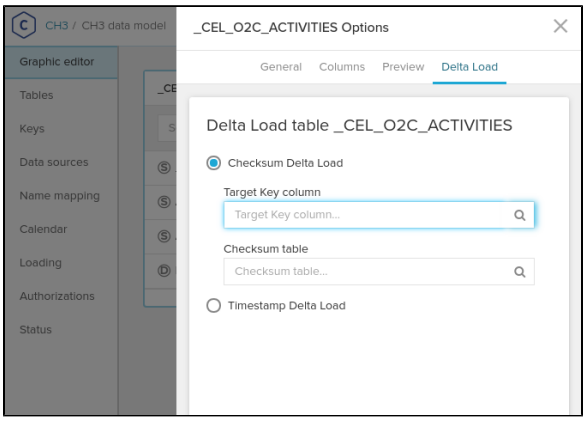
A Delta Load can significantly reduce the amount of data that has to be fetched from a source DB during the Data Model load.

To set up a Delta Load, CPM4 allows to partition the input tables. Having split the input tables into partitions, CPM4 then only fetches those partitions from the DB that it identified as changed since the last load. For identifying if the data inside partitions changed, CPM4 currently provides two strategies: "Timestamp Delta Load" and "Checksum Delta Load". For each table in the Data Model, the activation and selection of the partitioning strategy can be done in the Table configuration of the Data Model Editor:

The "Delta Load table" checkbox is available if the table is loaded from a database. As soon as it is checked, the "Delta Load" tab in the menu bar on the top appears:



In the Delta Load configuration tab, one of the available strategies can be chosen:



Timestamp Delta Load

This strategy partitions a table along a timestamp column. When a Data Model load is triggered in CPM4, this strategy checks for each partition if the number of entries, the maximum value or both changed. The retrieved values are called checksums and compared to the values from the previous load. If a checksum changes, the corresponding partition is reloaded from the source database.

Pro	Con
+ Easy to configure	• Might misses updates
+ Applicable in many cases	• Checksums can be expensive to calculate on source DB

Example

In the below example, the column "Eventtime" is used as Timestamp Delta Load column. Here, the partitioning is based on month and we use the count of timestamps for the checksum.

First Load

We first load this table from the source DB.

Second Load

Now the input data on the source DB changed with an additional entry.

Case	Activity	Eventtime
1	A	01.10.2019
1	B	02.10.2019
1	C	04.11.2019

During the first Data Model load, checksum labels are created in the background (not visible for the user). The checksum is here the count of rows per month.

Case	Activity	Eventtime	Checksum (Count of timestamps per month)
1	A	01.10.2019	2
1	B	02.10.2019	2
1	C	04.11.2019	1

Case	Activity	Eventtime
1	A	01.10.2019
1	B	02.10.2019
1	C	04.11.2019
1	D	16.10.2019

When we now trigger the Data Model load again, the checksums in the background (not visible for the user) are recalculated. The count per month changes for the month October from 2 to 3. The checksum for November stays the same as no value for November was added.

Case	Activity	Eventtime	Checksum (Count of timestamps per month)
1	A	01.10.2019	3
1	B	02.10.2019	3
1	C	04.11.2019	1
1	D	16.10.2019	3

Now, only the data of the table that is in the month October is updated in Celonis, the data for which the Eventtime is in November is not updated. Even if the activity name in November changed in the source DB from "C" to "F", this value would not be updated with the defined Delta Load because the row count did not change.

Checksum Delta Load

To be more flexible, the Checksum strategy allows you to define yourself which partitions are reloaded. The strategy relies on a separate checksum table. In this table you define the partitions and its checksums.

If you want a partition from the Source Table to be reloaded during the Data Model load, you only have to change the Checksum value from the Partition Table. A Data Model reload will only load those partitions for which the Checksum value of the Partition Key changed. The exact value of the Checksum is not of importance.

Example

In this example the source table is partitioned by country code:

Source Table		
Costumer	**City**	**Country**
Costumer A	Bristol	GB
Costumer B	Paris	FR
Costumer C	London	GB

Partition Table	
Partition Key	**Checksum**
GB	1
FR	1

If in this example the checksum for FR is set to 2 and a Reload from Source is triggered, all costumers from France are reloaded. The customers from Great Britain will not be updated.

The partition table is not managed by CPM4 and has to be maintained by the user. Also it is noteworthy that partitions which are not mentioned in the Partition table are not fetched from the source database and are therefore not available in CPM4. In the example above, customers from Spain would not be loaded because there is no Partition Key ES in the Partition table.

Pro	**Con**

+ Very flexible	• User needs to implement logic to maintain partition table