

# DatabaseSink

## Parameter

- connection: The name of an existing database connection (see [Database Feature](#) how to create one)  
Or:
- type: The JDBC connection type (e.g. mysql)
- db: The Database to access
- user: The username for the database
- password: The password for the user
- host: The hostname of the database
- port: The port of the database connection  
Or:
- jdbc: The jdbc connect string
- table: The table to write to
- drop: Should the table be dropped and recreated
- truncate: Should existing elements be dropped before writing to the table (but keeping the Table)
- BatchSize: How many elements should be written, before they are committed to the database (default is 10)
- BatchTimeout: How long should the system wait (in ms) even if not BatchSize elements are buffer. This is useful for cases where the datarate is not constant. Default is 0, i.e. no timeout, wait for ever. If the query the closed the batch will be written also.

The following parameters configure a new recovery mechanism, that can be used, if the database connection is temporarily unavailable (still under progress and may change).

- recoveryenabled (boolean): A flag that indicates, if mechanisms for recovery should be used.
- recoverystoretype (String): When recovery is used, the data needs to be stored for later restart. Here the store type can be defined: Currently, FileStore and MemStore are available. Further stores (e.g. an external database will be added in the future).
- recoveryoptions (String-Map): Each recoverystore may need different types of further information. E.g. FileStore needs the filename, where to temporarily store elements for recovery.

## Example

```
sink = DATABASESINK({
    connection = 'con1',
    table = 'testTable2',
    primarykeys = ['ts'],
    batchsize = 100,
    drop = false,
    recoverystoretype = 'FileStore',
    recoveryenabled = false,
    recoveryoptions = [
        ['filename', 'g:/tmp/testTable1Recovery.store']
    ]
},
input
)
```

Further examples can be found under [Database Feature](#)

## Remark

You could use a [Map operator](#) before the database sink if you need to map the input to a database schema or a prepared statement.