# Pentaho Metadata MQL Outer Join Scenarios

Below are a list of scenarios that the outer join algorithm will need to support.

## *Scenario 1: Two Tables are Outer Joined*

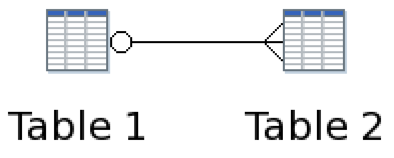
Table2 is Outer Joined to Table1

Table1 Data

| PrimaryKey | ForeignKey |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |

Table2 Data

| PrimaryKey | ForeignKey |
|---|---|
| 1 | 1 |
| 2 | 2 |



SQL Expected:

FROM Table1 LEFT OUTER JOIN Table2 ON (Table1.PrimaryKey = Table2.ForeignKey)

Results:

| Table1.PrimaryKey | Table2.PrimaryKey |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | NULL |

## *Scenario 1a: Two Tables are Outer Joined with a constraint*

MQL Selections: Table1.PrimaryKey, Table2.PrimaryKey
MQL Constraint: Table2.PrimaryKey > 1

SQL Expected:

FROM Table1 LEFT OUTER JOIN Table2 ON (Table1.PrimaryKey = Table2.ForeignKey AND Table2.PrimaryKey > 1)

Results:

| Table1.PrimaryKey | Table2.PrimaryKey |
|---|---|
| 1 | NULL |
| 2 | 2 |
| 3 | NULL |

## *Scenario 1b: Two Tables are Outer Joined with an aggregate*

MQL Selections: Table1.PrimaryKey,  SUM(Table2.PrimaryKey)

SQL Expected:

SELECT Table1.PrimaryKey, SUM(Table2.PrimaryKey) FROM Table1 LEFT OUTER JOIN Table2 ON (Table1.PrimaryKey = Table2.ForeignKey) GROUP BY Table1.PrimaryKey

Results:

| Table1.PrimaryKey | Table2.PrimaryKey |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | NULL |

## *Scenario 1c: Two Tables are Outer Joined with an aggregate constraint*

MQL Selections: Table1.PrimaryKey,  SUM(Table2.PrimaryKey)
MQL Constraint: SUM(Table2.PrimaryKey) > 1

SQL Expected:

SELECT Table1.PrimaryKey, SUM(Table2.PrimaryKey) FROM Table1 LEFT OUTER JOIN Table2 ON (Table1.PrimaryKey = Table2.ForeignKey) GROUP BY Table1.PrimaryKey HAVING SUM(Table2.PrimaryKey) > 1

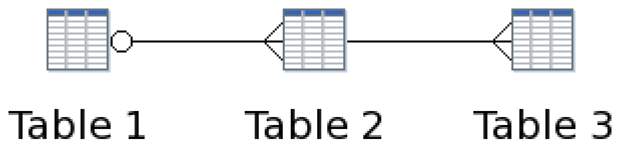
| Table1.PrimaryKey | Table2.PrimaryKey |
|---|---|
| 2 | 2 |

* Note that the outer join is no longer in effect due to the having constraint clause.

## *Scenario 2: Two Joined Tables are Outer Joined To a Single Table*

Additional Table3 added as a regular join to Table2

| Table3.PrimaryKey | Table3.ForeignKey |
|---|---|
| 1 | 1 |
| 3 | 3 |



SQL Expected:

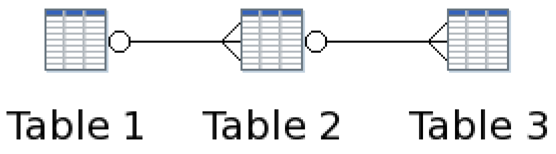
FROM Table1 LEFT OUTER JOIN (Table2 JOIN Table3 ON (Table2.PrimaryKey = Table3.ForeignKey)) ON (Table1.PrimaryKey = Table2.ForeignKey)

Results:

| Table1.PrimaryKey | Table2.PrimaryKey | Table3.PrimaryKey |
|---|---|---|
| 1 | 1 | 1 |
| 2 | NULL | NULL |
| 3 | NULL | NULL |

## *Scenario 3: Three Tables are Outer Joined*

Additional Table3 added as an outer join to Table2



SQL Expected:

```
FROM
        (Table1 LEFT OUTER JOIN Table2 ON (Table1.PrimaryKey = Table2.ForeignKey))
                 LEFT OUTER JOIN Table3 ON (Table2.PrimaryKey = Table3.ForeignKey)
```

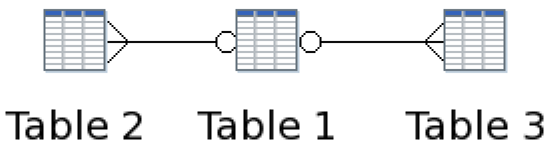
Equivalent algorithm generated SQL, starting at Table 1 (see below):

```
FROM
        (Table3 RIGHT OUTER JOIN Table2 ON (Table2.PrimaryKey=Table3.ForeignKey))
                 RIGHT OUTER JOIN Table1 ON (Table1.PrimaryKey=Table2.ForeignKey)
```

| Table1.PrimaryKey | Table2.PrimaryKey | Table3.PrimaryKey |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 2 | NULL |
| 3 | NULL | NULL |

## *Scenario 4: Two outer joins on a single Table*

Additional Table3 added as outer join to Table1

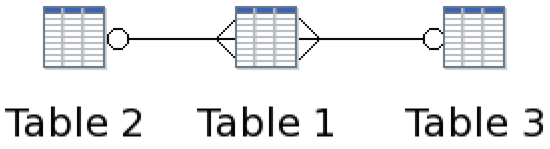


FROM
    Table1 LEFT OUTER JOIN Table2 ON (Table2.PrimaryKey = Table1.PrimaryKey)
        LEFT OUTER JOIN  Table3 ON (Table3.PrimaryKey = Table1.ForeignKey)

Equivalent algorithm generated SQL, starting at Table 1 (see below):
(Table3 RIGHT OUTER JOIN (Table2 RIGHT OUTER JOIN Table1 ON (Table2.PrimaryKey = Table1.PrimaryKey)) ON (Table3.PrimaryKey = Table1.ForeignKey)

| Table1.PrimaryKey | Table2.PrimaryKey | Table3.PrimaryKey |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 2 | NULL |
| 3 | NULL | 3 |

## *Scenario 5: Two outer joins in the opposite direction*



This scenario's results depend on the order of the joins. Most likely, a new field will be added to the defined Relations or Business Tables within the metadata model to dictate this order.

```
FROM
        Table1 RIGHT OUTER JOIN Table2 ON (Table2.PrimaryKey = Table1.PrimaryKey)
                RIGHT OUTER JOIN  Table3 ON (Table3.PrimaryKey = Table1.ForeignKey)
```

| Table1.PrimaryKey | Table2.PrimaryKey | Table3.PrimaryKey |
|---|---|---|
| 1 | 1 | 1 |
| NULL | NULL | 3 |

OR

```
FROM
        Table1 RIGHT OUTER JOIN Table3 ON (Table3.PrimaryKey = Table1.PrimaryKey)
                RIGHT OUTER JOIN  Table2 ON (Table2.PrimaryKey = Table1.ForeignKey)
```

| Table1.PrimaryKey | Table2.PrimaryKey | Table3.PrimaryKey |
|---|---|---|
| 1 | 1 | 1 |
| NULL | 2 | NULL |

* Special Note: Some databases, including Hypersonic, do not support the nested join syntax used in these examples. The algorithm used will need to be dialect aware and generate valid SQL for the specified database.