Figure: Network generated with two SQL statements from a local instance of the Reactome database (version reactome_28), representing a bipartite graph of directed relationships between molecules (large nodes, colored according to legend) and reactions (small yellow nodes). The first two queries (see below, query 1 & 2) followed by union merge are used to generate a network consisting of relationships between "molecules" and "reactions". To demonstrate the simplicity of using CytoSQL to enrich an existing network, useful node attributes were added for reactions and molecules, with one extra statement (query 3). It loads molecule and reaction descriptions for selected (or all) nodes.
Query 1: Loading Input - Reaction network

```sql
SELECT
    ReactionlikeEvent_2_input.input AS input,
    ReactionlikeEvent_2_input.DB_ID AS reaction,
    ReactionlikeEvent_2_input.input_class AS molecule_class,
    speciesinfo.eventID,
    speciesinfo.species,
    speciesinfo.speciesname,
    "reaction"
FROM
    ReactionlikeEvent_2_input
INNER JOIN
    (SELECT
        Event_2_species.DB_ID as eventID,
        species, _displayName AS speciesname
    FROM
        Event_2_species
    LEFT JOIN
        DatabaseObject
    ON
        Event_2_species.species = DatabaseObject.DB_ID)
    AS speciesinfo
ON
    ReactionlikeEvent_2_input.DB_ID = speciesinfo.eventID
WHERE
    speciesinfo.speciesname = "Homo sapiens"
```

*CytoSQL Mode: Create Network*

*CytoSQL mappings: source node, target node, source, edge, source, source, target*

(Note: the 7th “reaction” field in the SELECT statement is a fixed value column that is mapped as a target node attribute, to facilitate the differentiation of “reaction” nodes from “molecular” nodes.)

Query 2: Loading Output - Reaction network

```sql
SELECT
    ReactionlikeEvent_2_output.DB_ID AS reaction,
    ReactionlikeEvent_2_output.output AS output,
    ReactionlikeEvent_2_output.output_class AS molecule_class,
    speciesinfo.eventID,
    speciesinfo.species,
    speciesinfo.speciesname,
    "reaction"
FROM
    ReactionlikeEvent_2_output
INNER JOIN
    (SELECT
        Event_2_species.DB_ID as eventID,
        species, _displayName AS speciesname
    FROM
        Event_2_species
    LEFT JOIN
        DatabaseObject
    ON
        Event_2_species.species = DatabaseObject.DB_ID)
    AS speciesinfo
ON
    ReactionlikeEvent_2_output.DB_ID = speciesinfo.eventID
```
WHERE
    speciesinfo.speciesname = 'Homo sapiens'

_CytoSQL Mode: Create Network_
_CytoSQL mappings: source node, target node, target, edge, target, target, source_

_The networks created with Query 1 and Query 2 are merged with the “Union” option._

(Note: the 7th “reaction” field in the SELECT statement is a fixed value column that is mapped as a source node attribute, to facilitate the differentiation of “reaction” nodes from “molecular” nodes.)

_Query 3: Loading node descriptions_

SELECT
    DB_ID,
    _displayName
FROM
    DatabaseObject
WHERE
    DB_ID = ?

_CytoSQL Mode: Update node attributes_
_CytoSQL bind variable: ID_