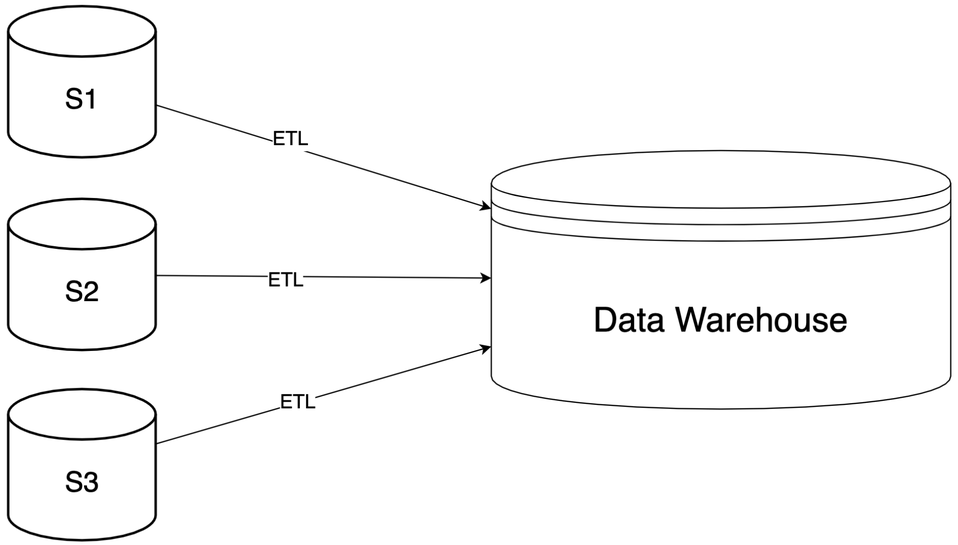
Master thesis GFDM

Project introduction

Data integration

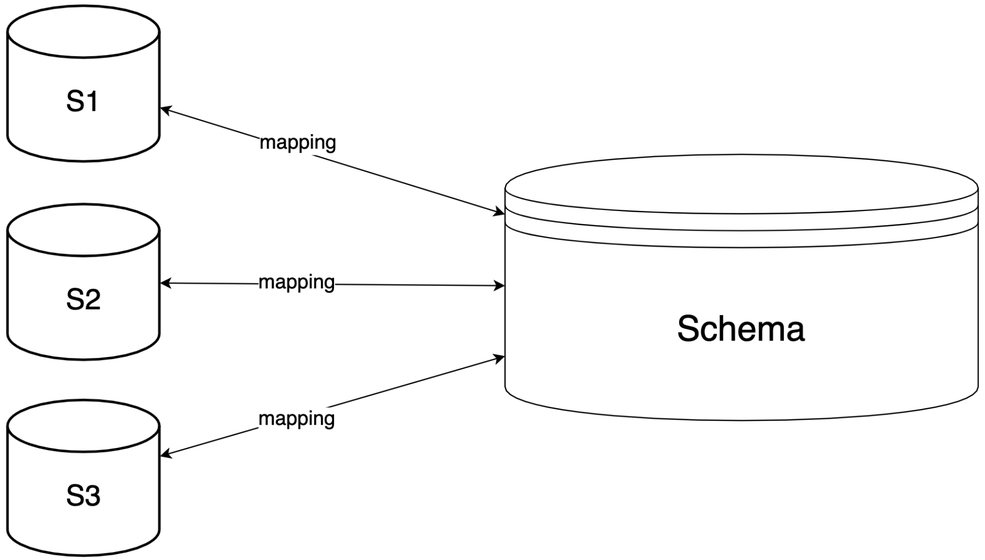
Physical data integration

1. ETL
2. Cyclic



Virtual data integration

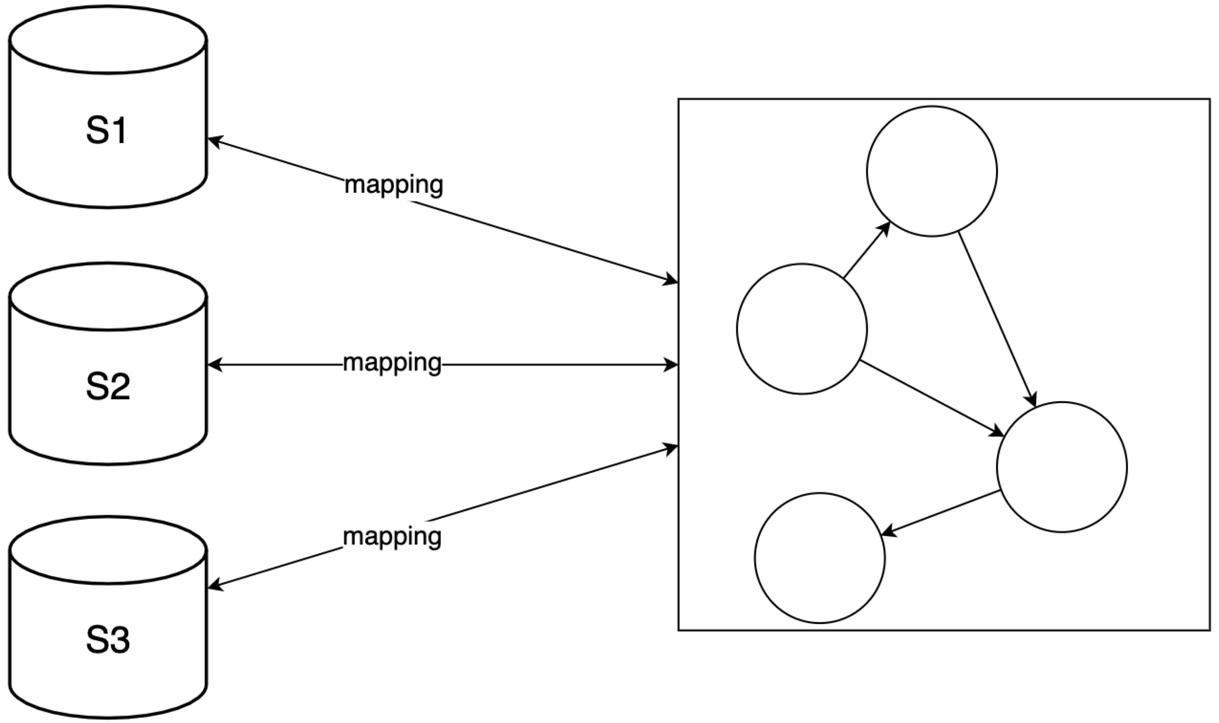
1. Mappings
2. Data freshness
3. No data transfer



Graph data integration

Graph data integration

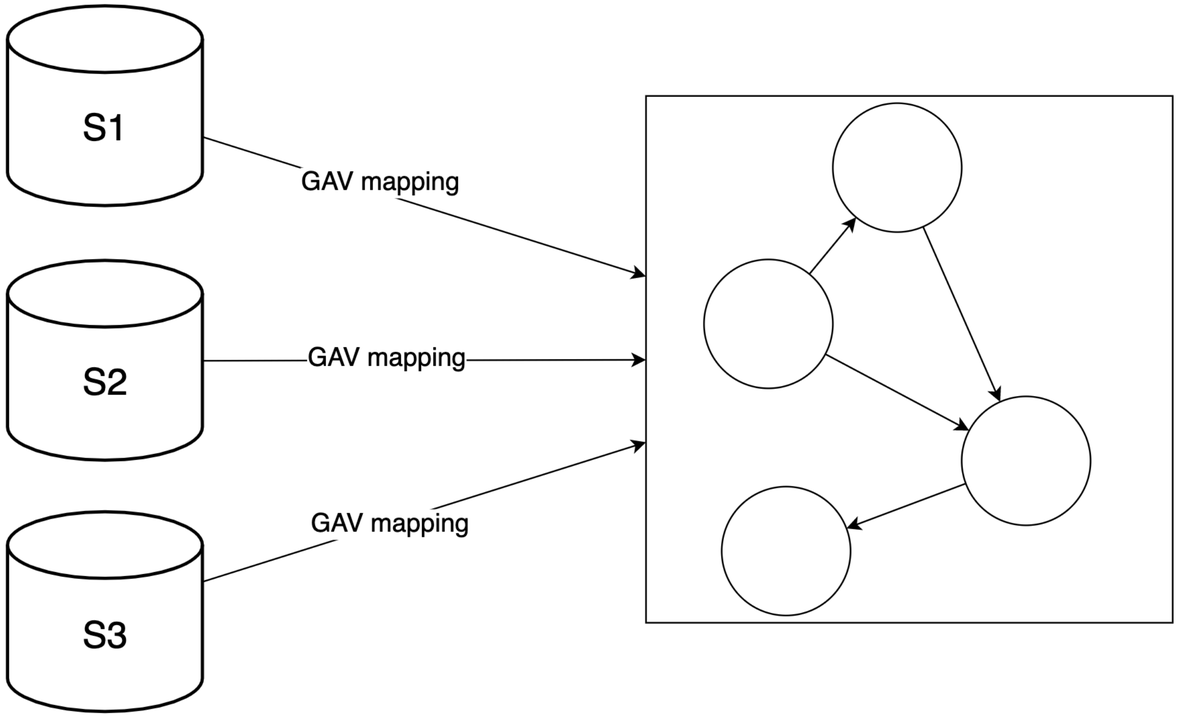
1. Schema easy to extend
2. Exploit graph semantic for queries
3. Easy to read and access for non domain experts



Virtual Data integration

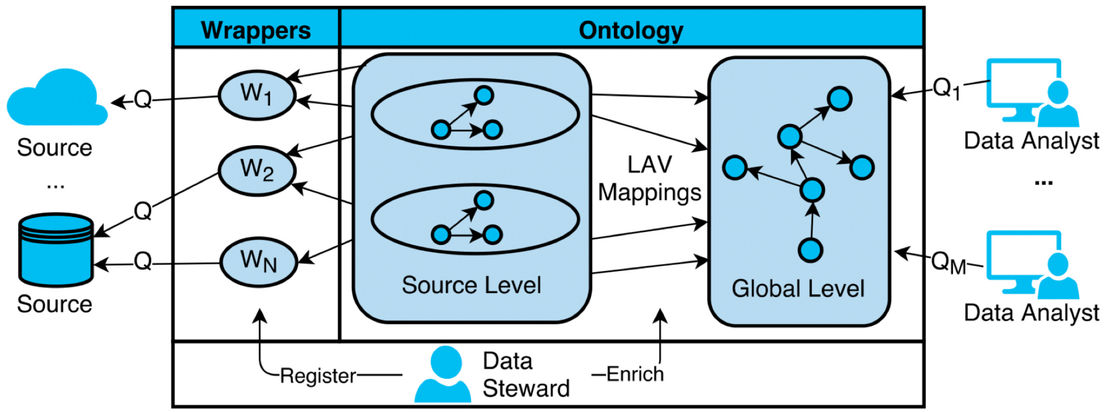
Ontology based data access

1. GAV

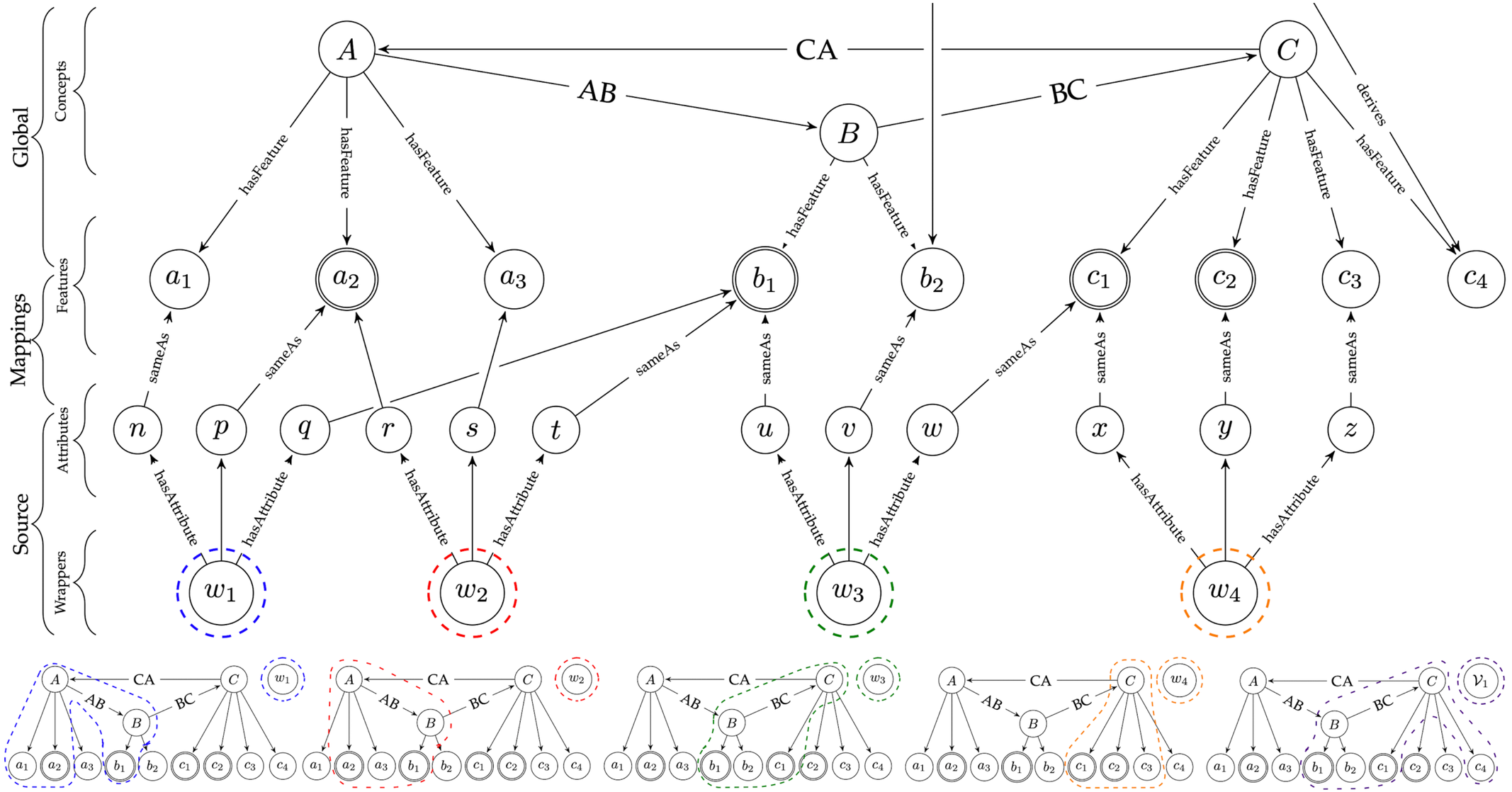


Ontology mediated queries

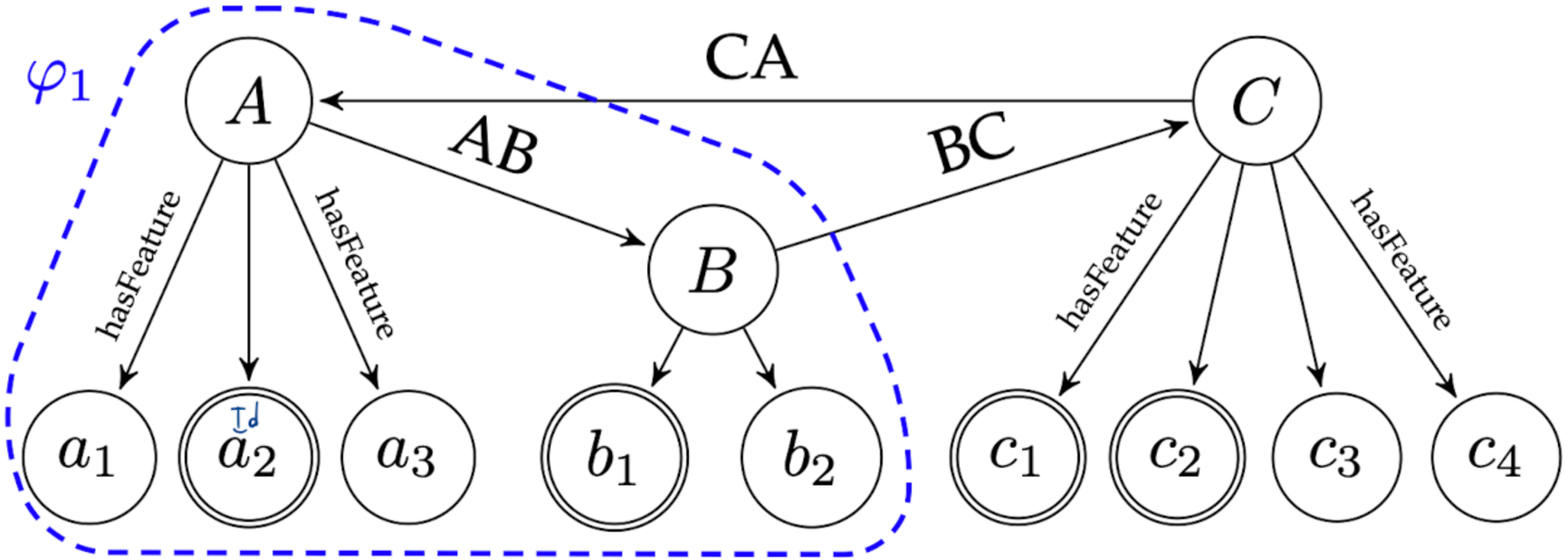
1. Wrappers
2. LAV



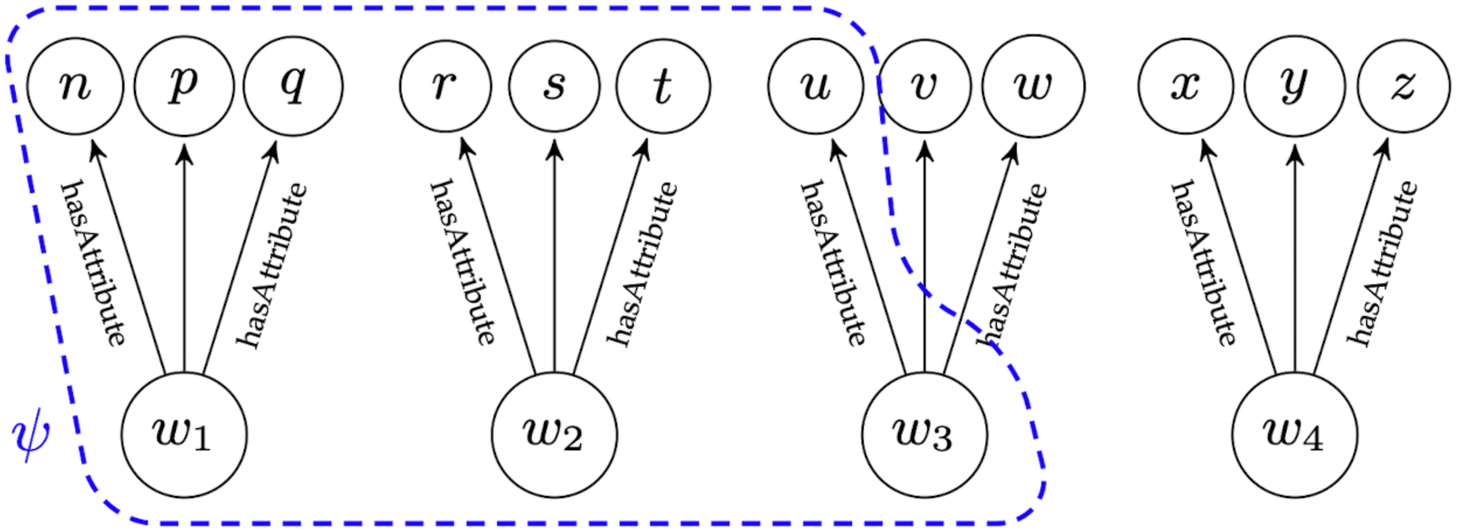
The GFDM system, Schema modeling



Querying over the graph



The rewriting algorithm



$$h_{\mathcal{I}}(\psi) = \pi_{n,p,s,u,v}((W_1 \times W_2 \times W_3)|p = r \wedge q = u \wedge t = u)$$

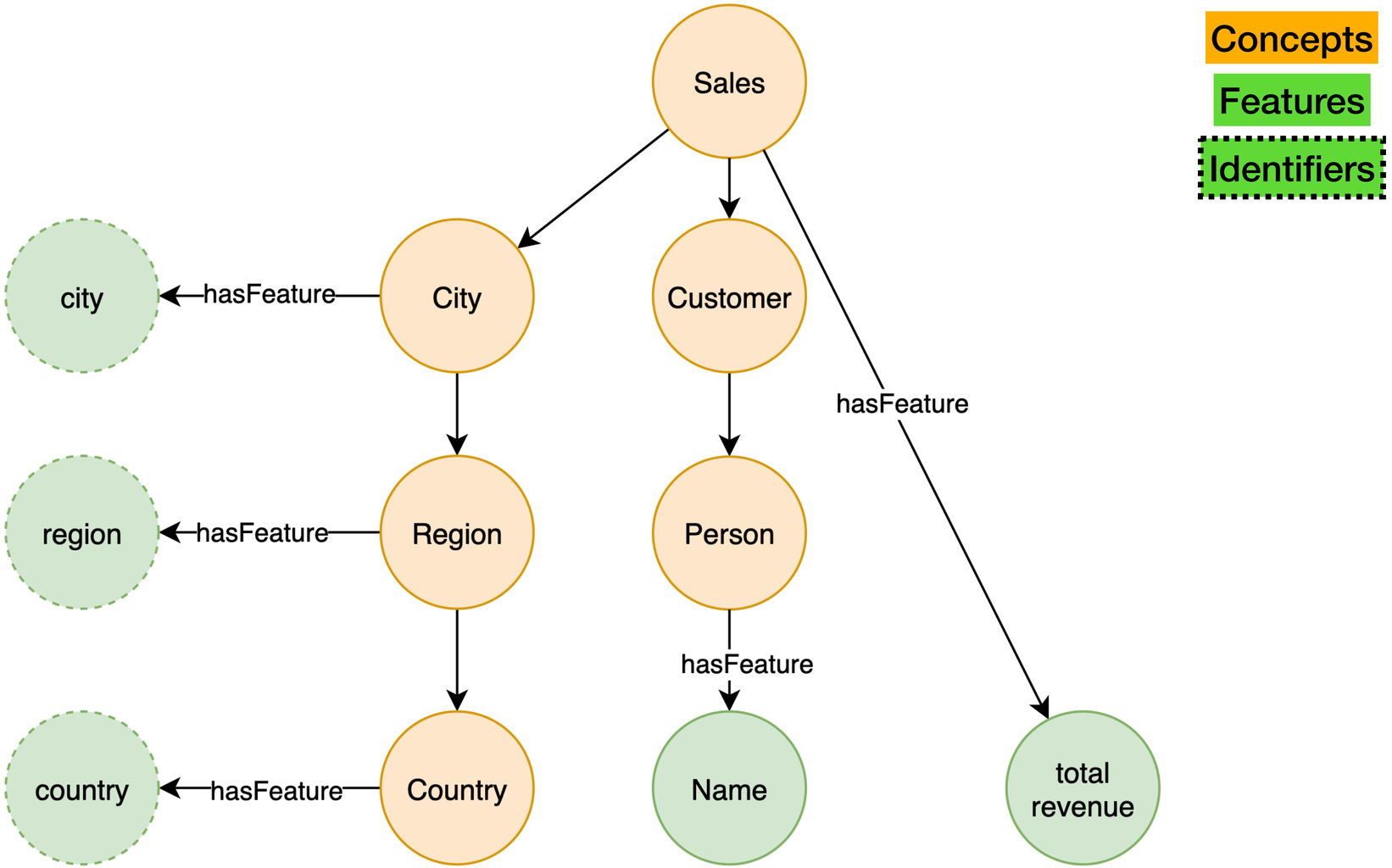
$exec(h_{\mathcal{I}}(\psi))$:

	n	p	s	u	v
t_1	n_1	p_1	s_1	u_1	v_1
t_2	n_2	p_2	s_2	u_2	v_2

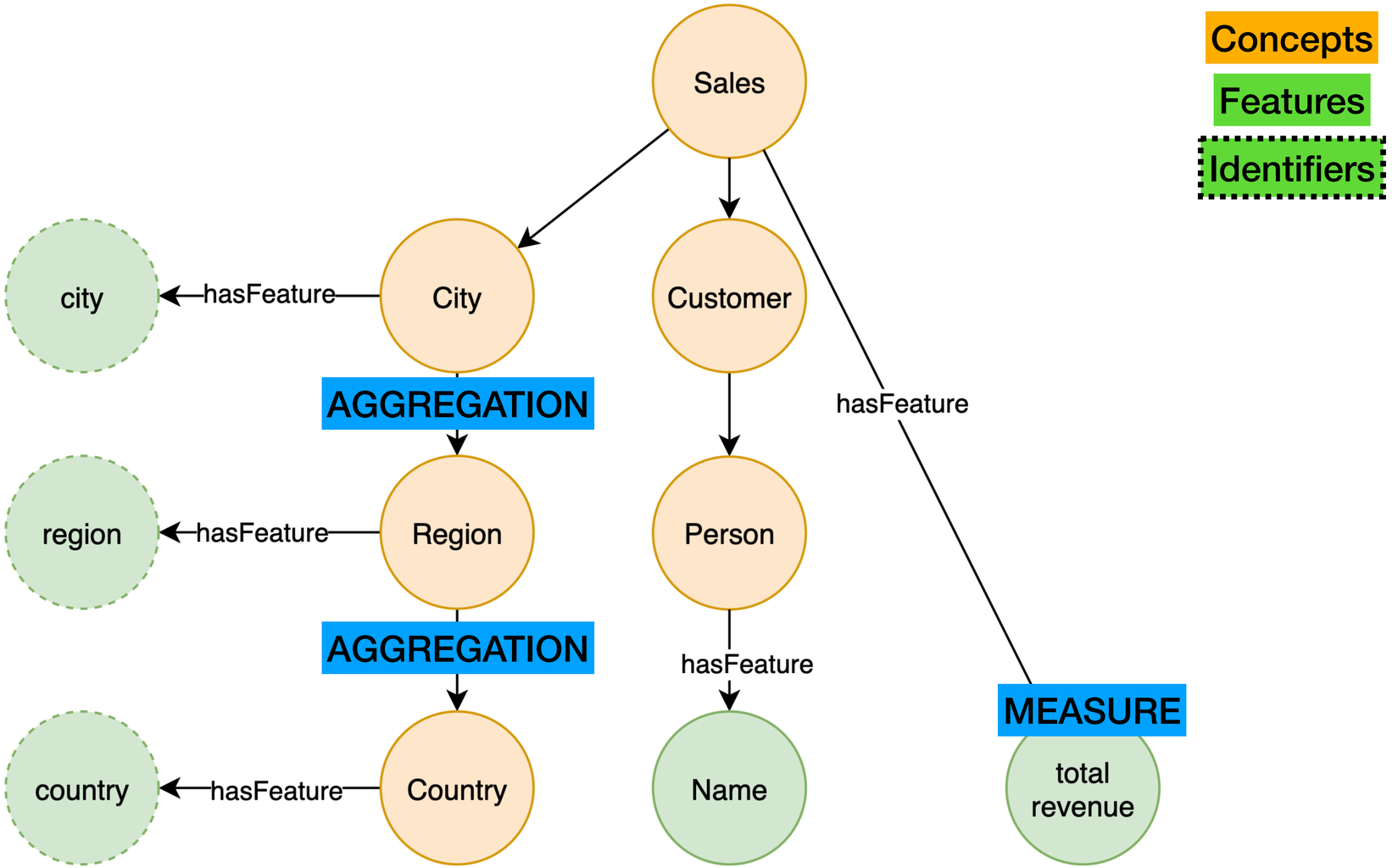
Goals

1. Model a **Multidimensional schema** by means of Knowledge Graph syntax
    1. Facts
    2. Mesures
    3. Dimensions
    4. Levels
    5. Aggregation function
2. The extension should be most transparent as possible to the user
3. Support **implicit** aggregations to automatically deliver the right data granularity (Roll-up)
    1. Automatic join of dimensional data
    2. Automatic GROUP-BY
    3. Automatic aggregation of measures

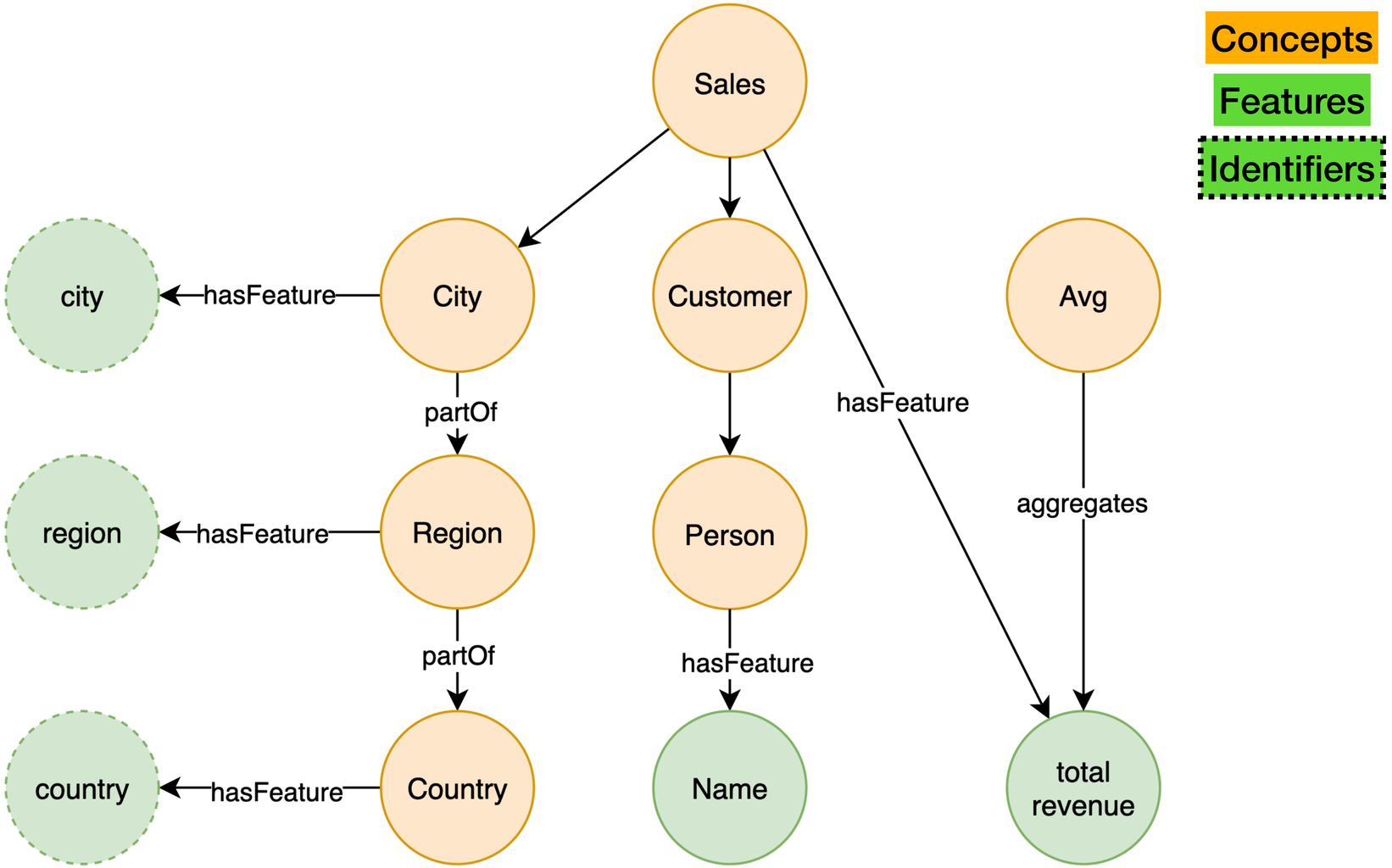
The model


Concepts
Features
Identifiers
Sales
City
Customer
city
hasFeature
region
hasFeature
Region
Person
hasFeature
hasFeature
country
hasFeature
Country
Name
total
revenue

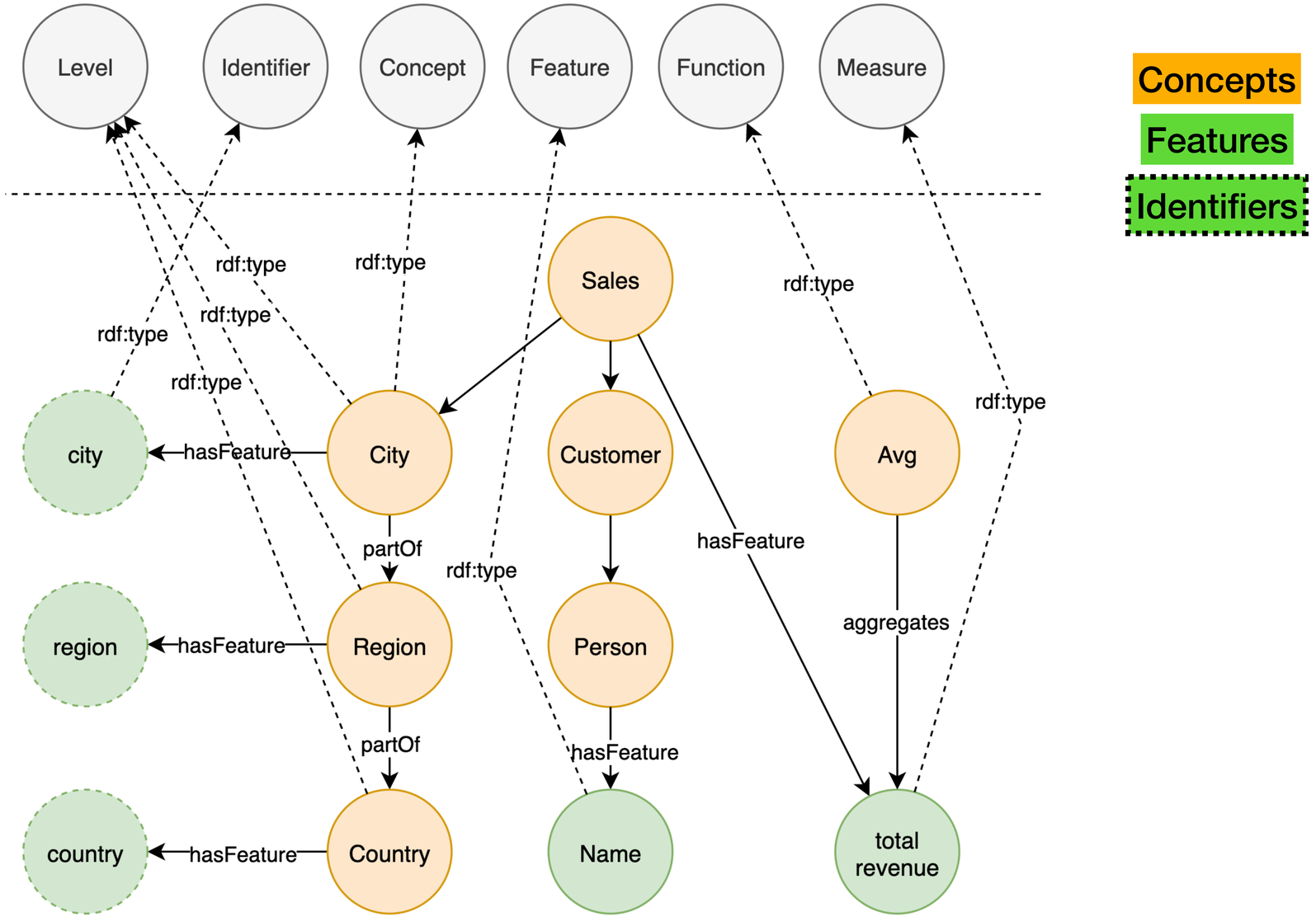
Make a KG support aggregations



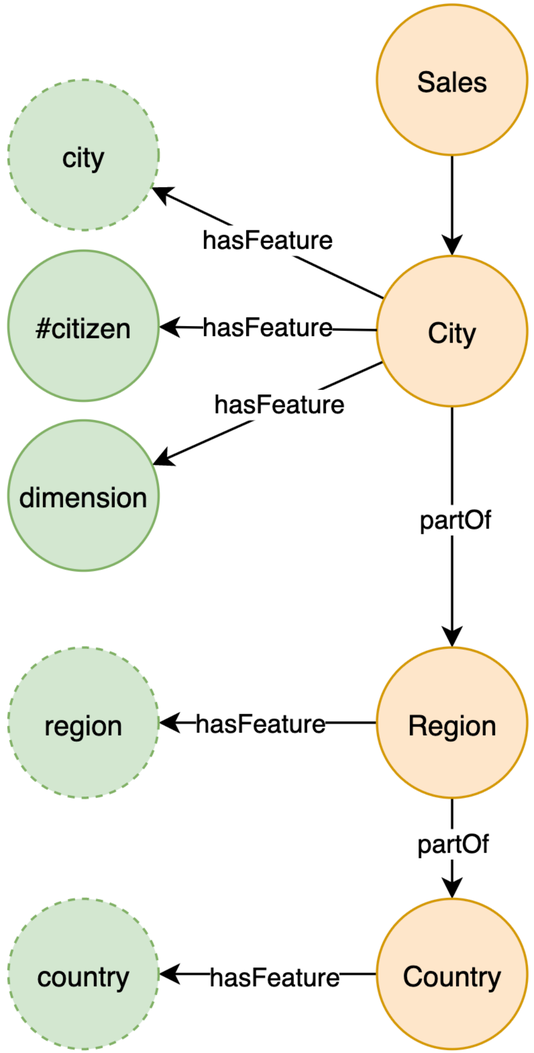
The model


Concepts
Features
Identifiers
Sales
city
hasFeature
City
Customer
Avg
partOf
hasFeature
region
hasFeature
Region
Person
aggregates
partOf
hasFeature
country
hasFeature
Country
Name
total revenue

Typing


Level
Identifier
Concept
Feature
Function
Measure
Concepts
Features
Identifiers
rdf:type
rdf:type
rdf:type
rdf:type
rdf:type
rdf:type
rdf:type
rdf:type
Sales
city
hasFeature
City
Customer
Avg
partOf
hasFeature
region
hasFeature
Region
Person
aggregates
partOf
hasFeature
country
hasFeature
Country
Name
total revenue

Dimensions in detail



Concepts

Features

Identifiers

Dimension and levels.

Concepts

Features

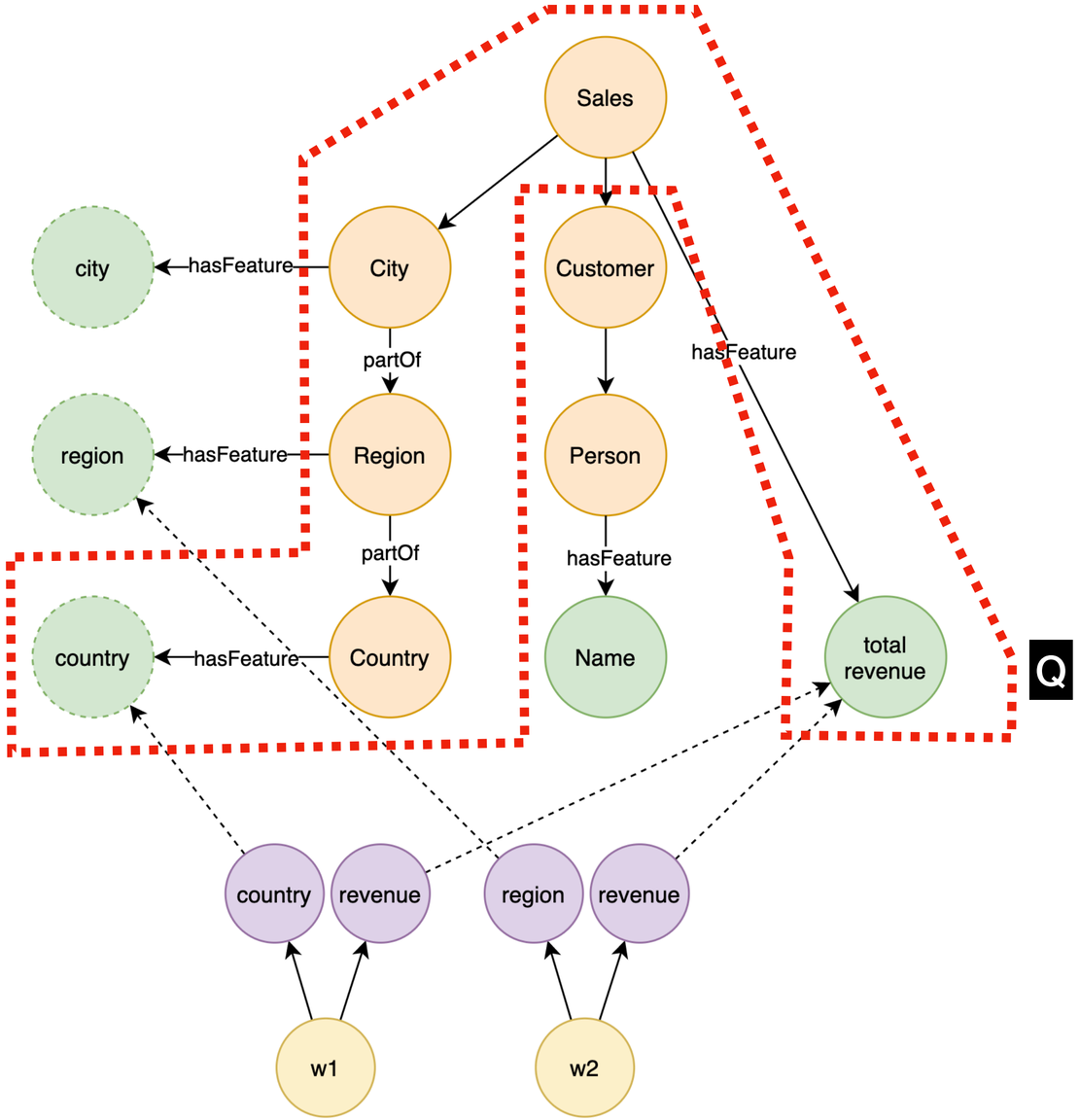
Identifiers

Attributes

Wrappers

Query

Implicit aggregation Model



Given a query Q.

We want to have *implicit aggregation* to *enrich* the query output.

Without implicit aggregations (Roll-up) it would not be possible to have the output of *w2* since it is in a lower aggregation level.

Roll-up, Joining dimensional data

Concepts

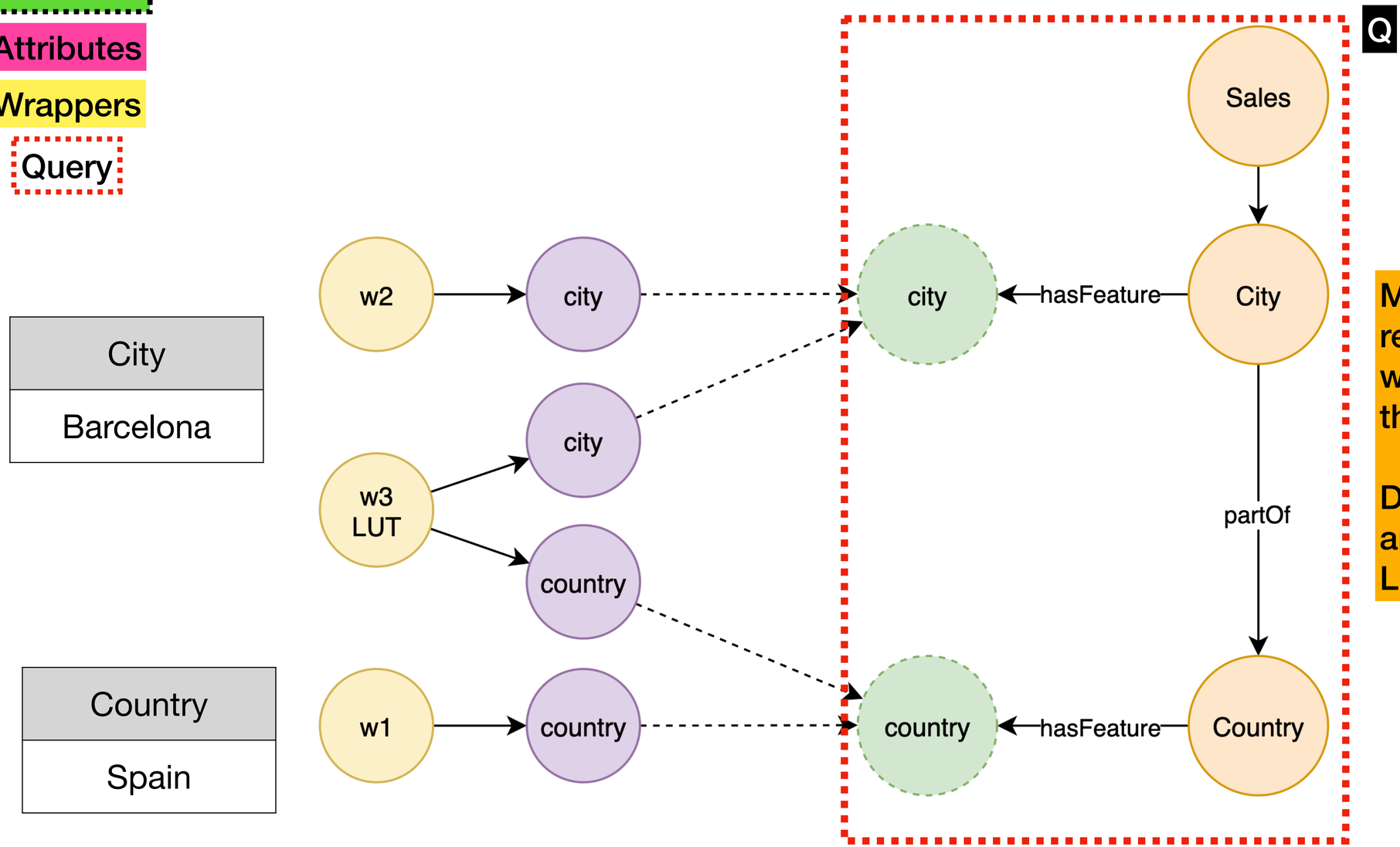
Features

Identifiers

Attributes

Wrappers

Query



Missing dimensional data related to country in *w2* , a wrapper used as LUT will be the solution to Roll-up.

Dimensional data relationship are represented by means of LUT (*w3*).

The LAV mappings and join semantics

Concepts

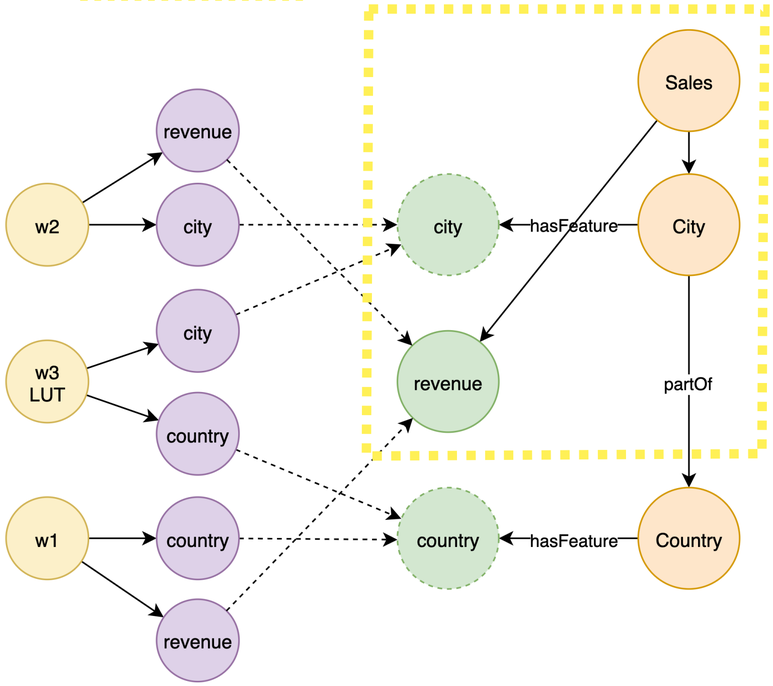
Features

Identifiers

Attributes

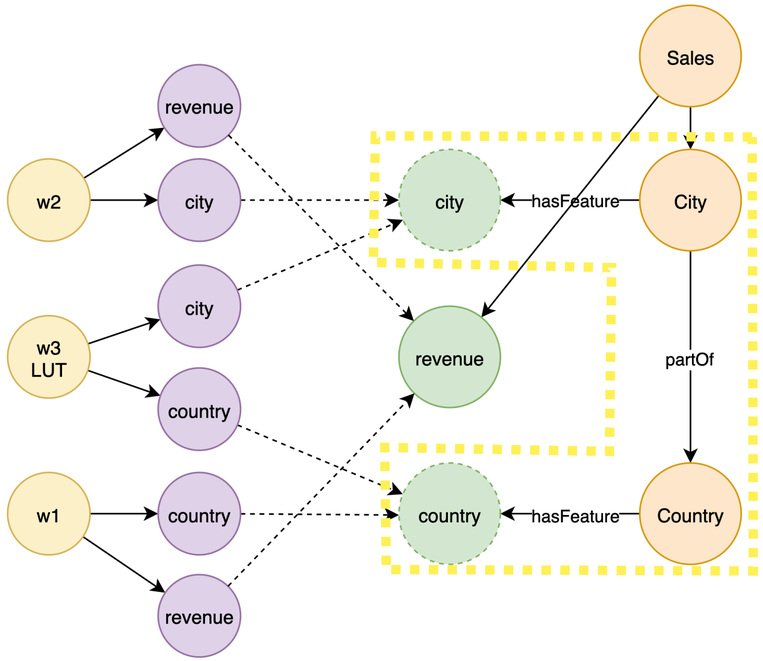
Wrappers

Mappings



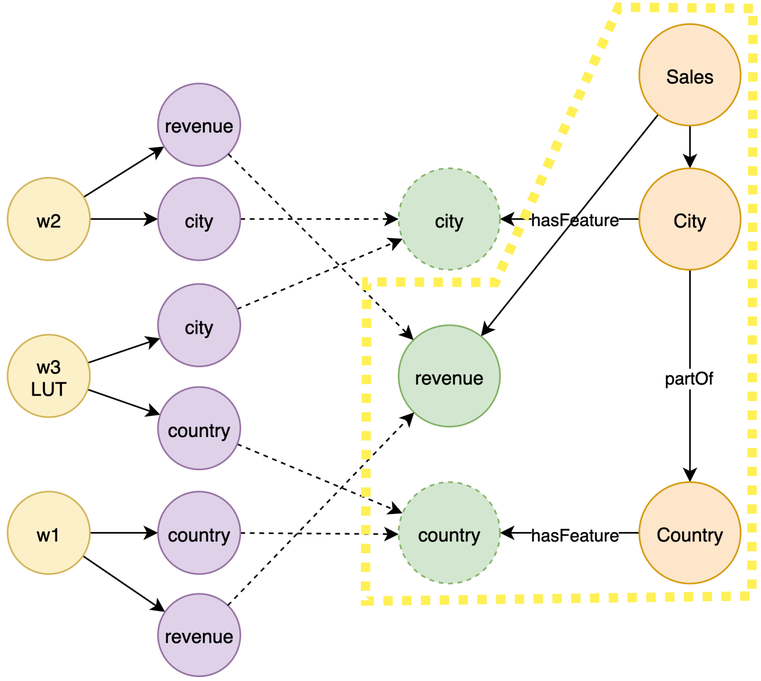
Mapping for w2

⋈



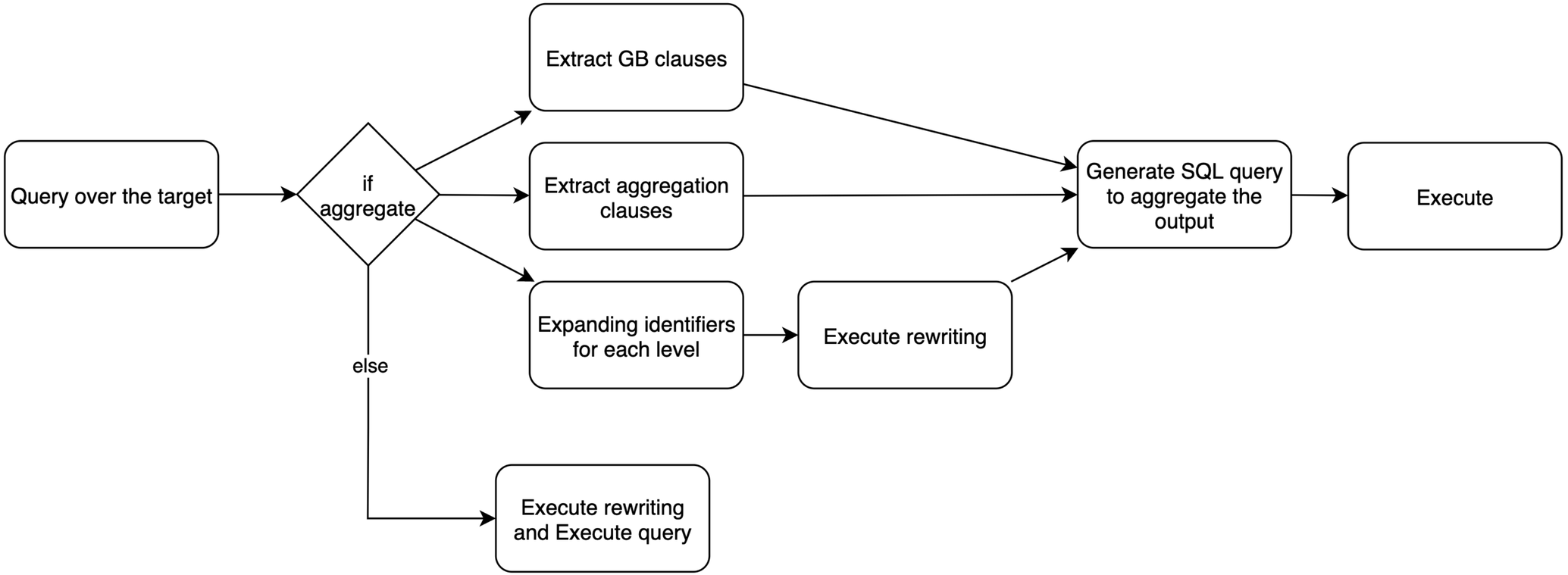
Mapping for w3

⋈

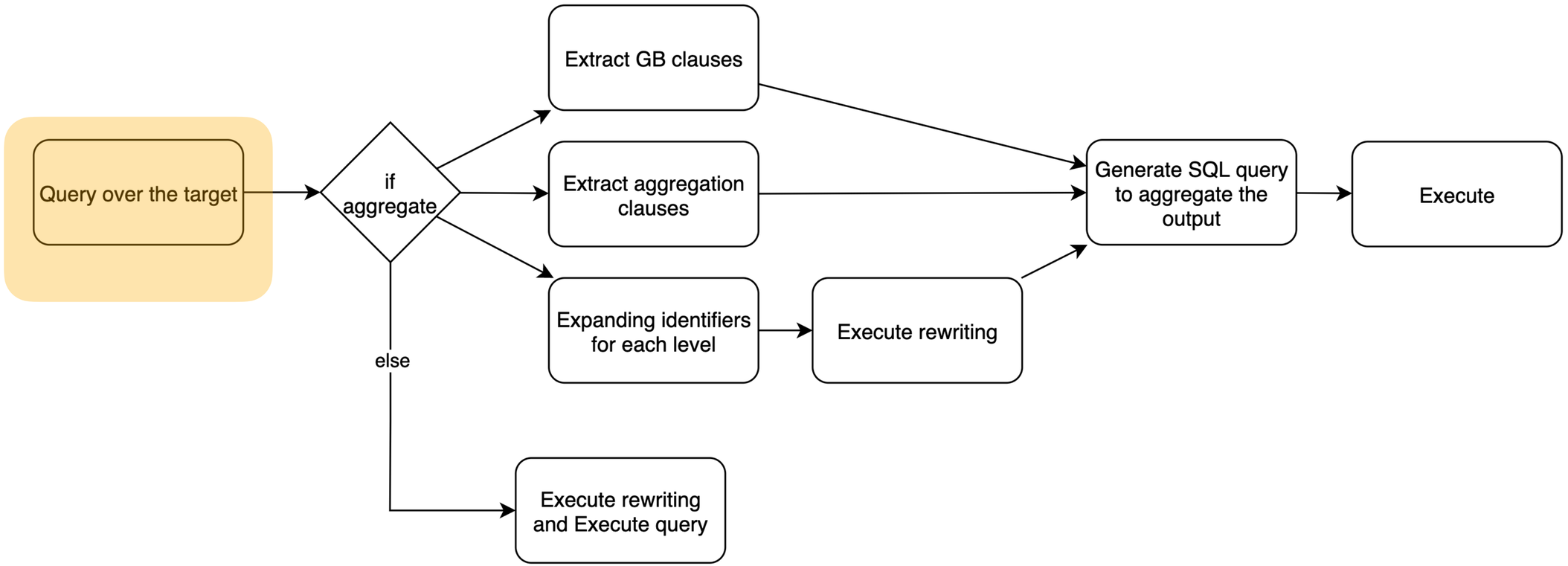


Mapping for w1

The Algorithm



The query


Query over the target
if aggregate
Extract GB clauses
Extract aggregation clauses
Expanding identifiers for each level
Execute rewriting
else
Execute rewriting and Execute query
Generate SQL query to aggregate the output
Execute

The query

Concepts

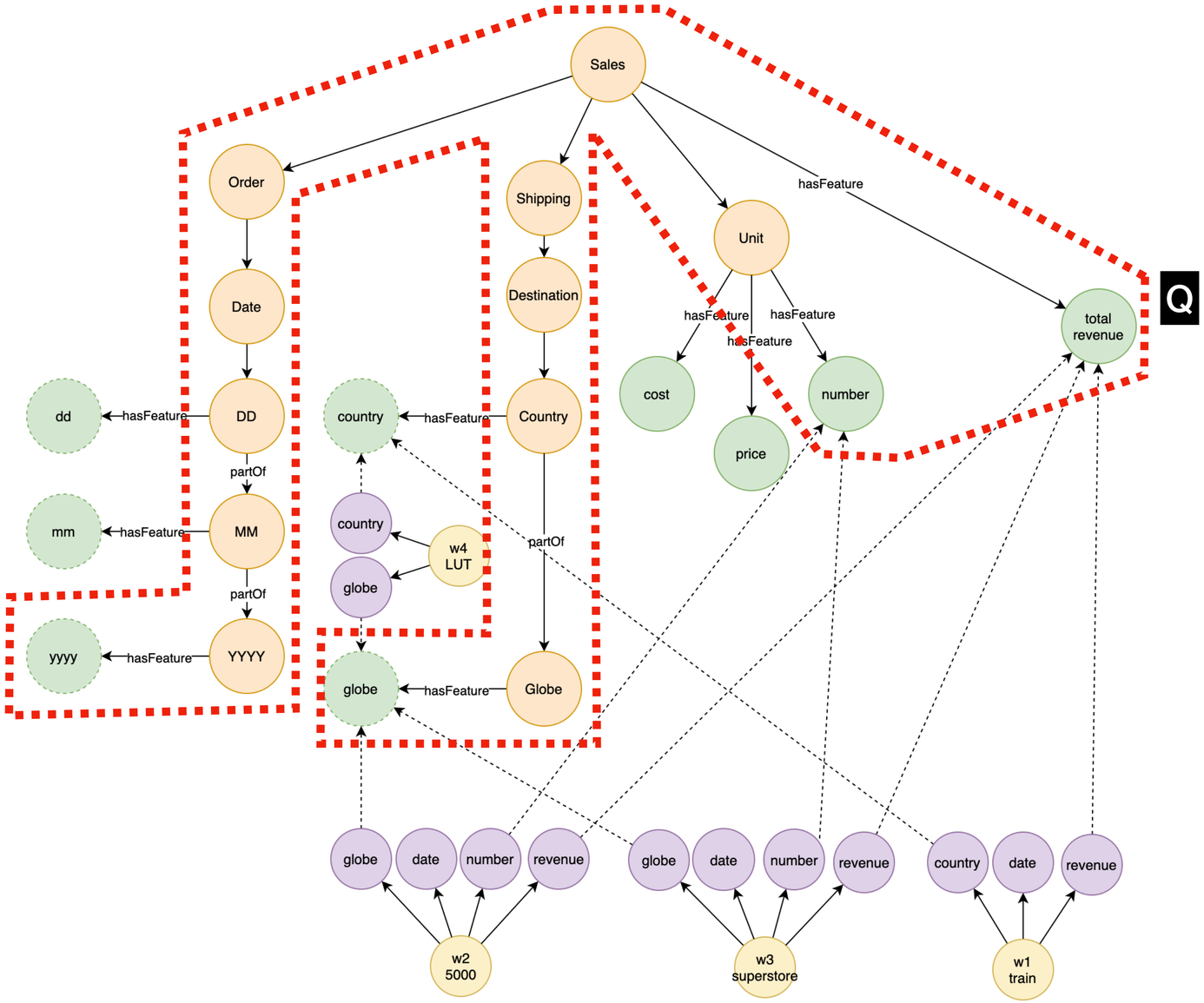
Features

Identifiers

Attributes

Wrappers

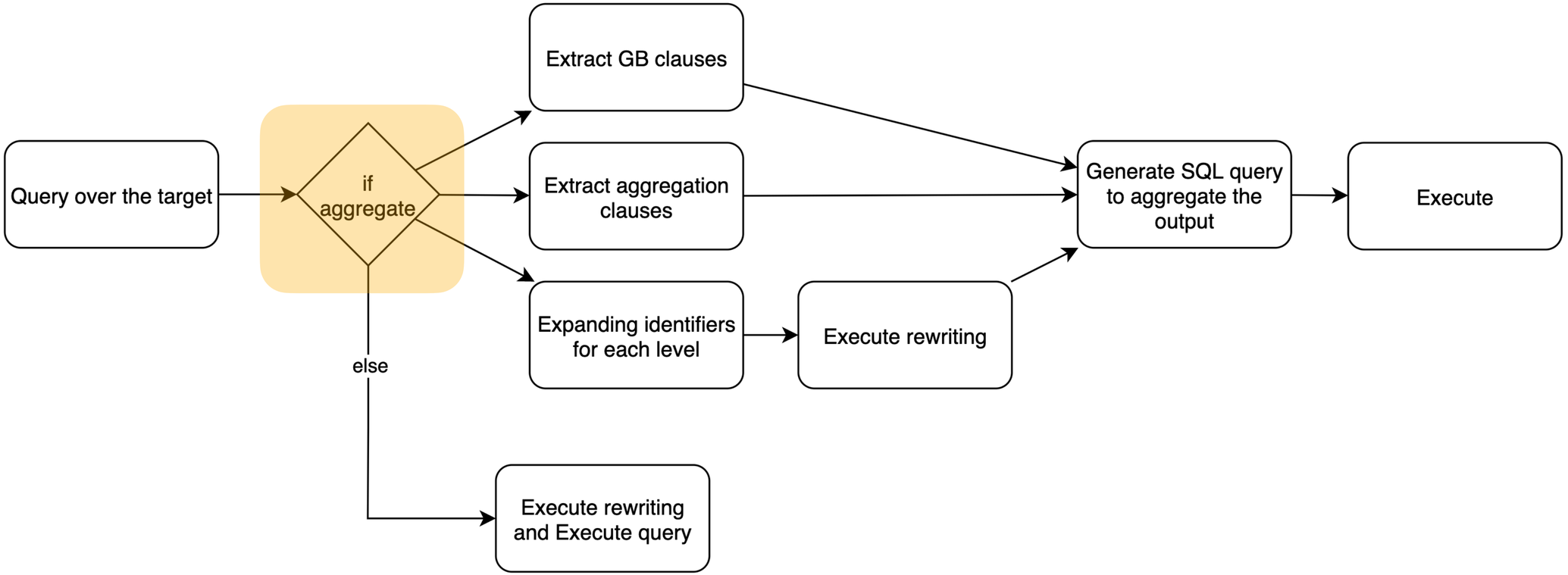
Query



Semantic:

Revenue and product sold in Year and Globe

The aggregation conditions



When does implicit aggregation can be performed?

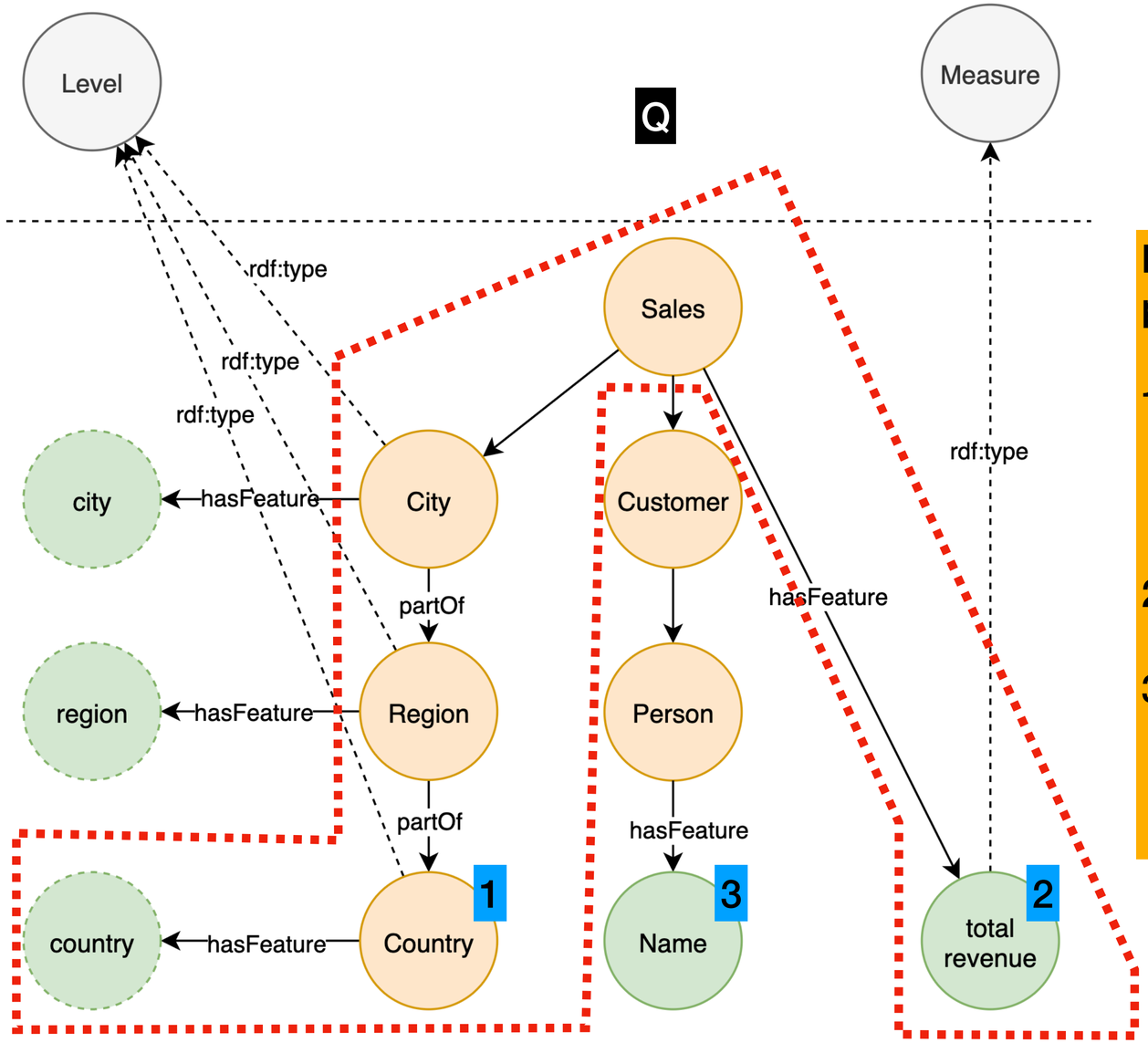
Concepts

Features

Identifiers

RDFS class

Query



Implicit aggregation is performed when:

1. A concept instance of Level is in Q and that concept is related to an Identifier
2. A feature instance of Measure is in Q
3. There are no concepts typed as Level and / or with features not being measures

When does implicit aggregation can be performed?

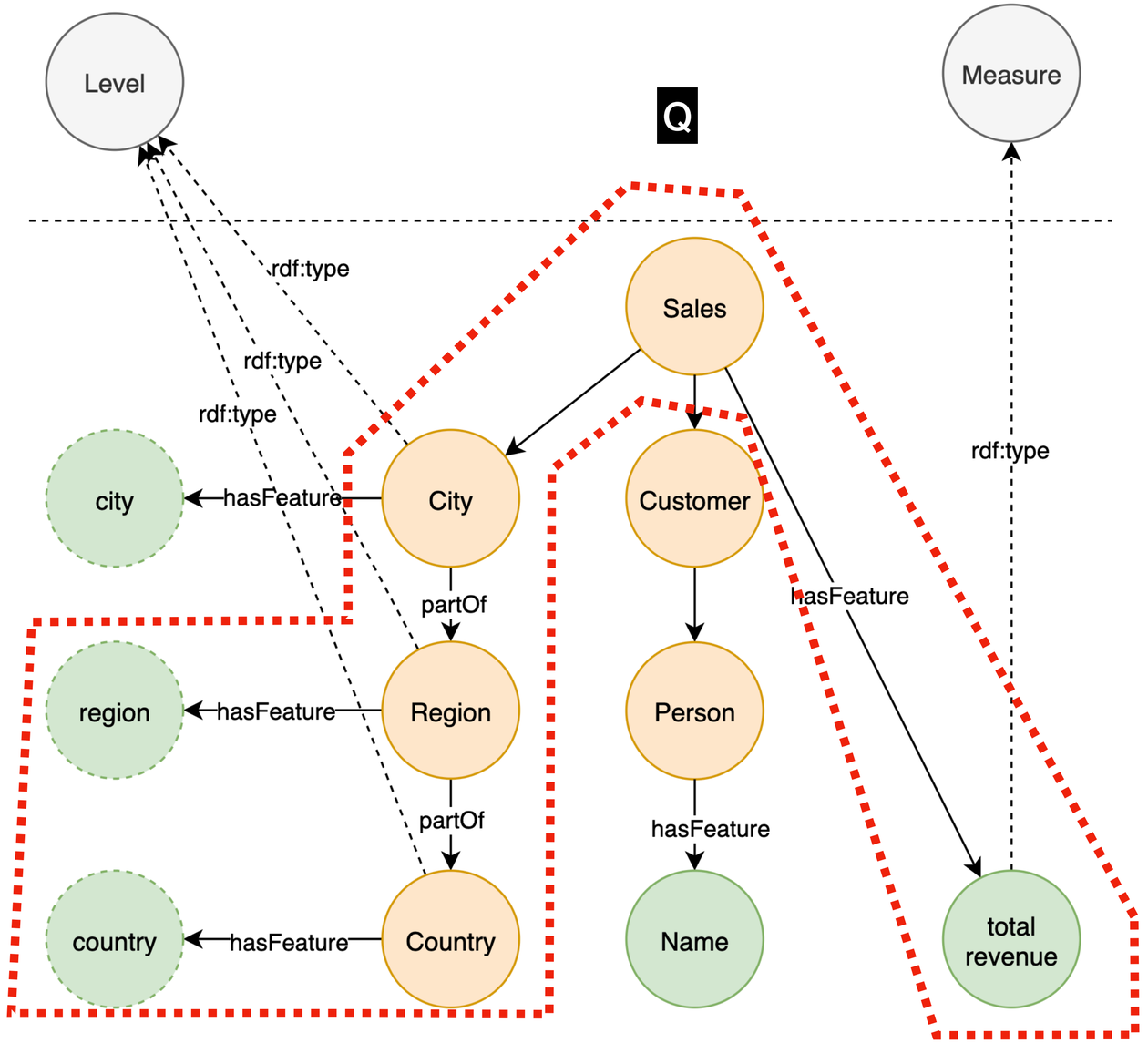
Concepts

Features

Identifiers

RDFS class

Query



In this case also make sense to have implicit aggregation.

The aggregation will be executed Rolling-up cities into region.

When does implicit aggregation can be performed?

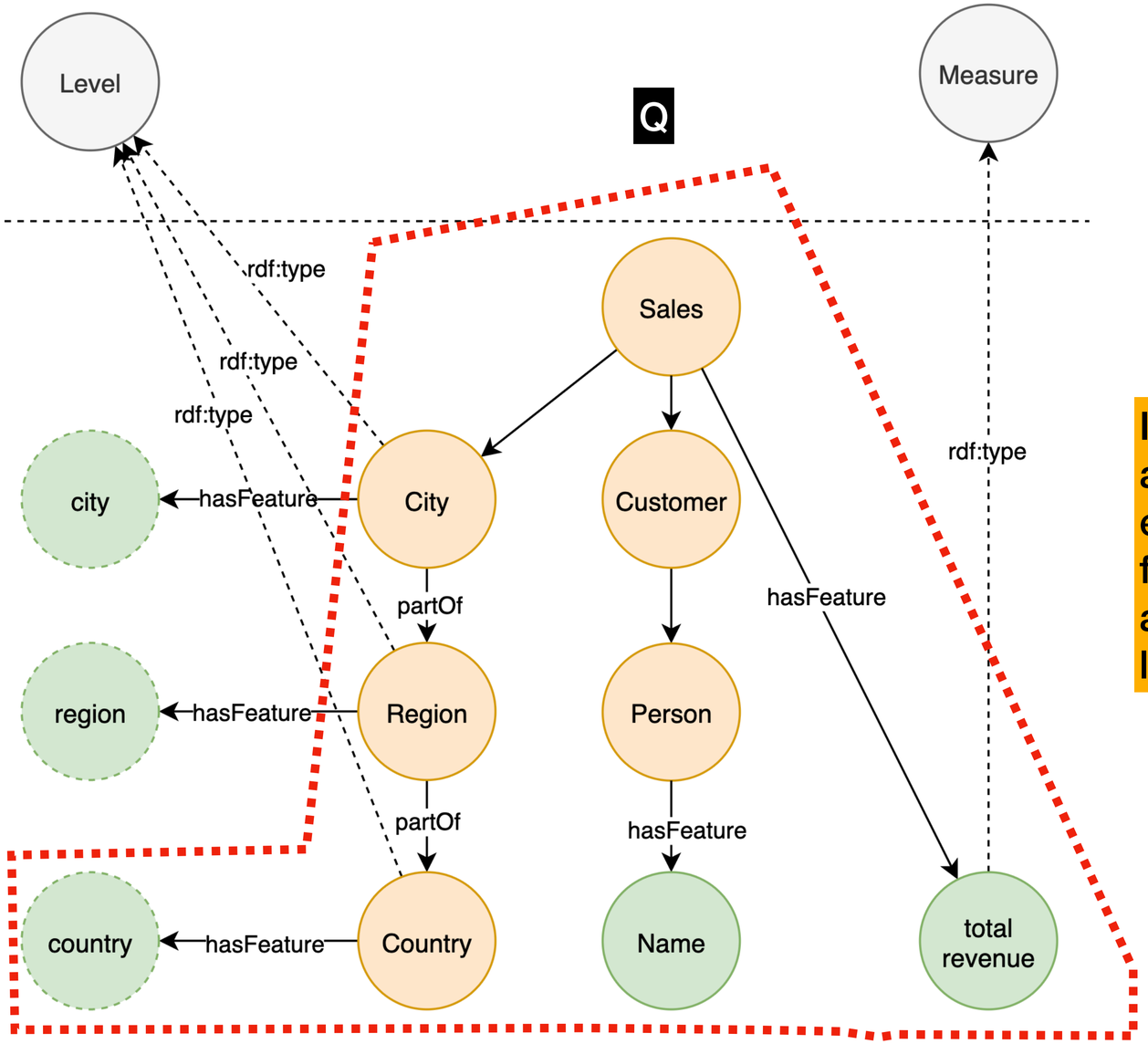
Concepts

Features

Identifiers

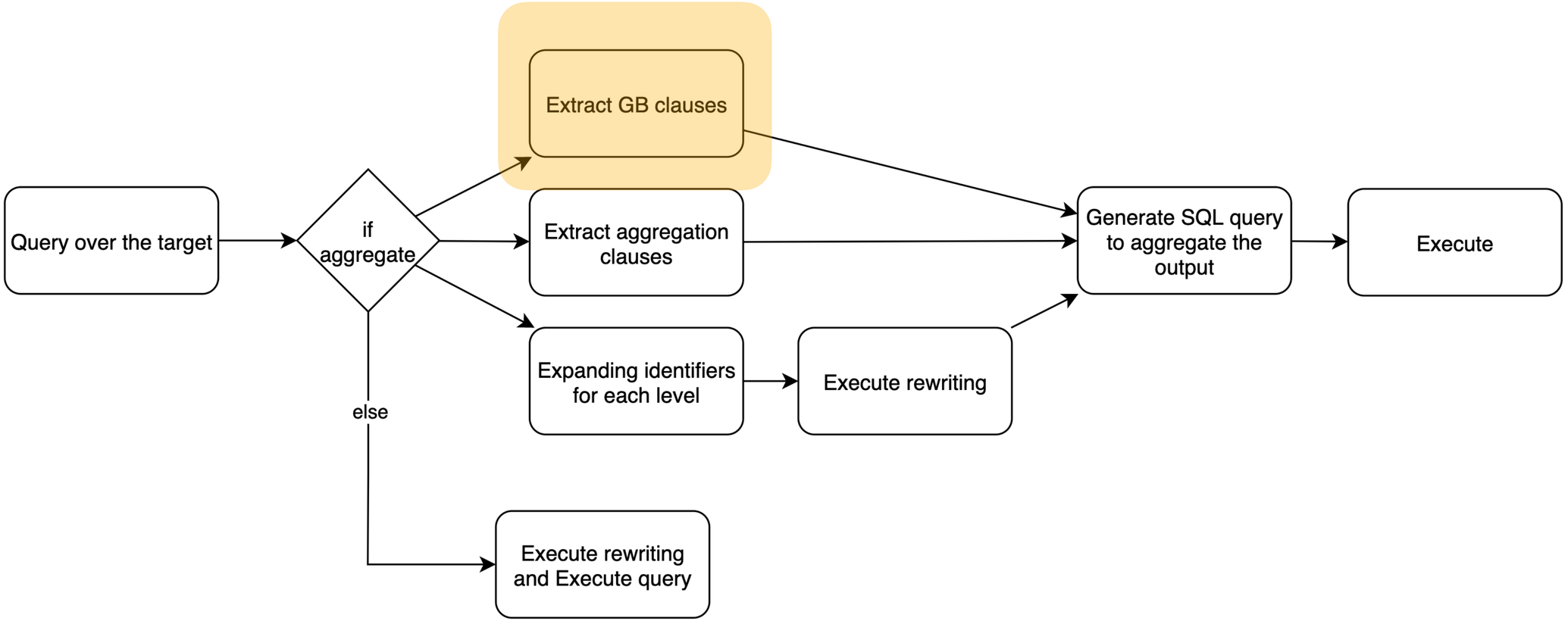
RDFS class

Query



In this case implicit aggregation will not be executed since we have a feature that is not a measure and does not behave to a level

Extract GB clauses



Extract GB clauses

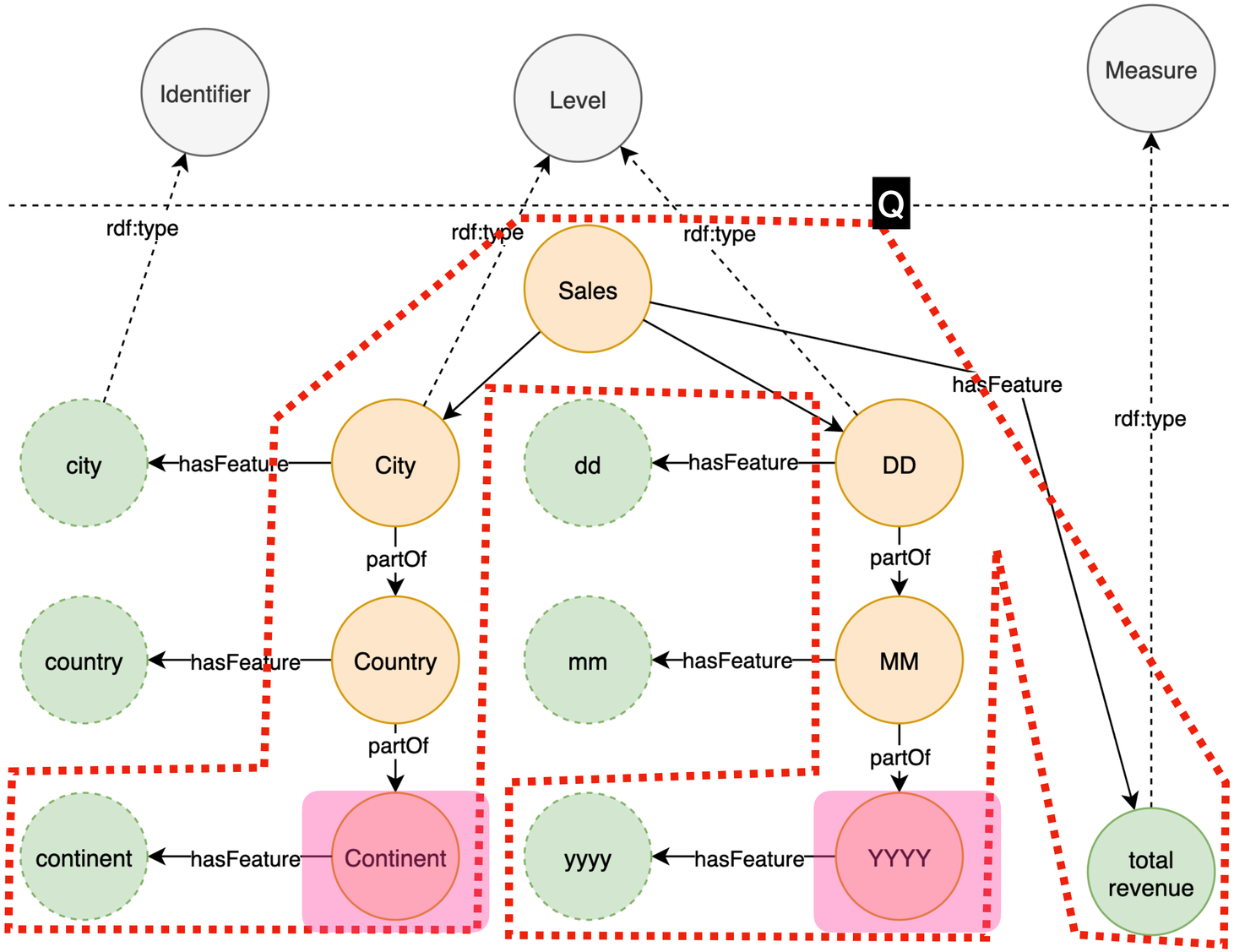
Concepts

Features

Identifiers

RDFS class

Query



We need in Q Concepts that are instances of Level and have at least a feature that is an Identifier.

The GB clauses are *Continent* and *YYYY*

Extract GB clauses

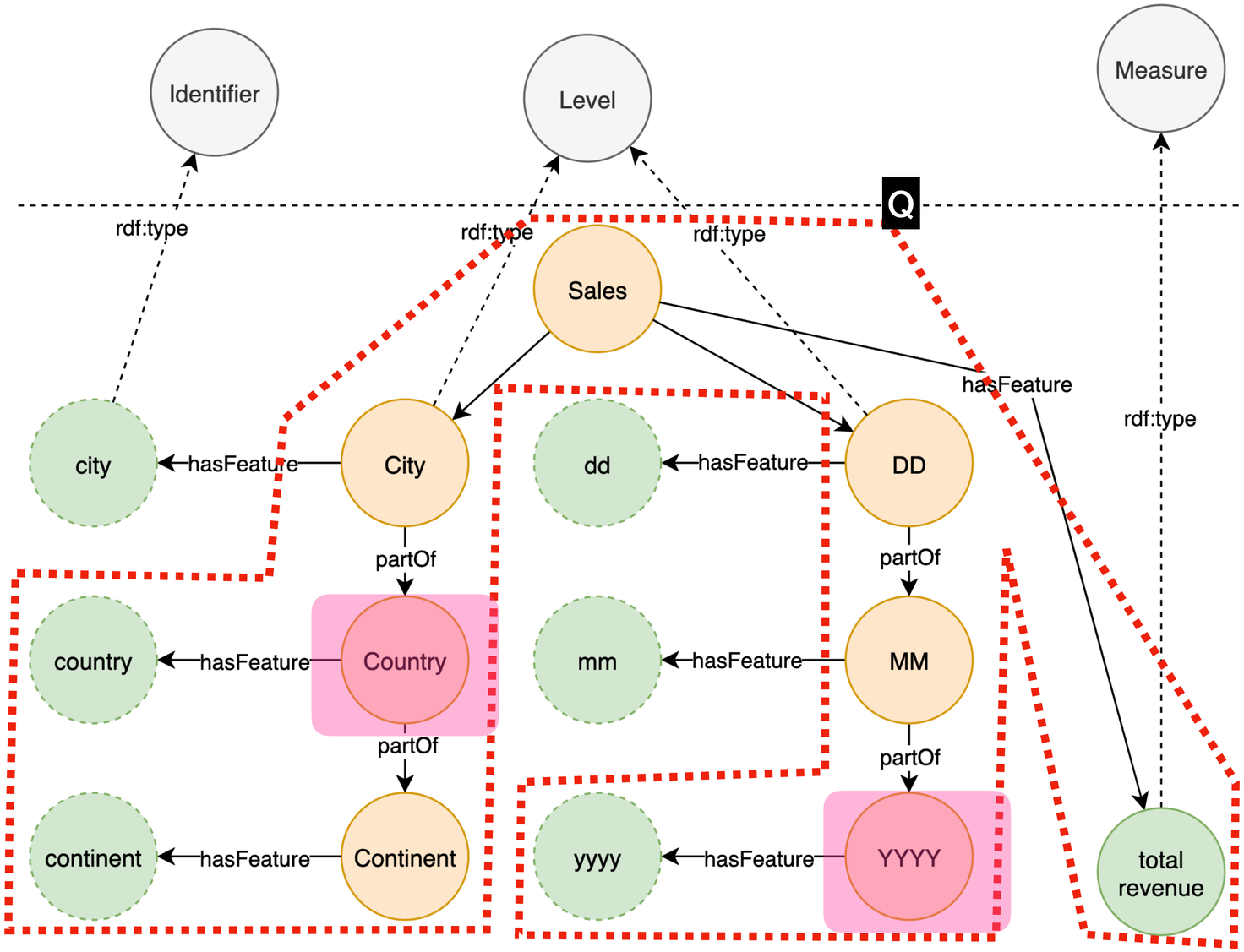
Concepts

Features

Identifiers

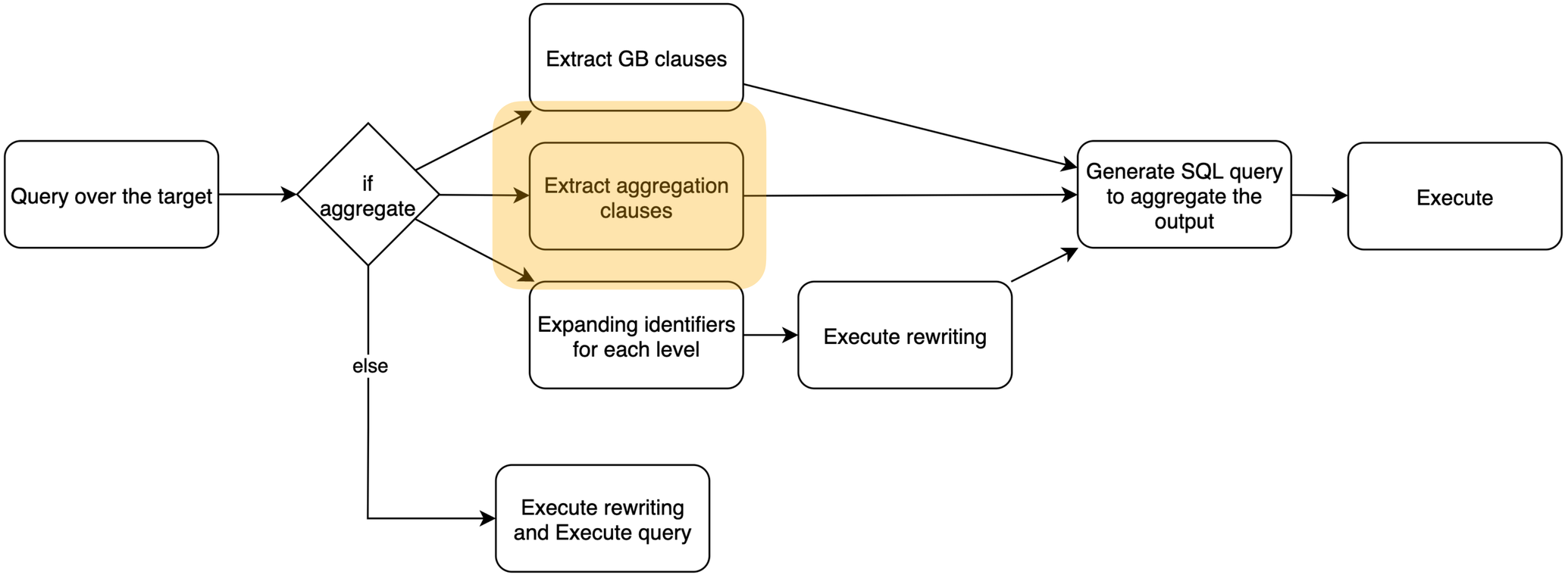
RDFS class

Query



If multiple concept of the same dimension the GB clause is the level with the finest granularity.

Extract aggregation clauses



Extract aggregation clauses

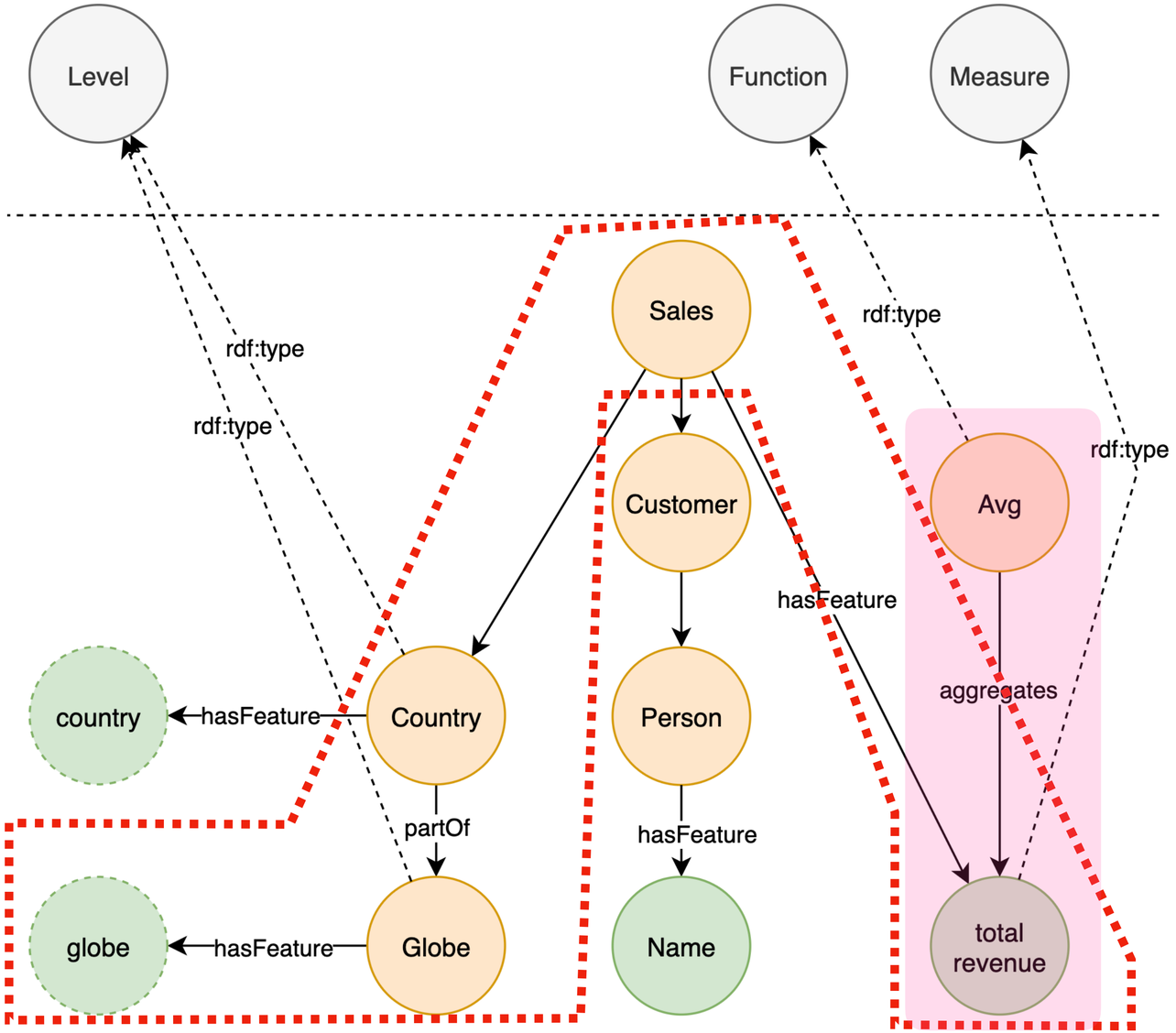
Concepts

Features

Identifiers

RDFS class

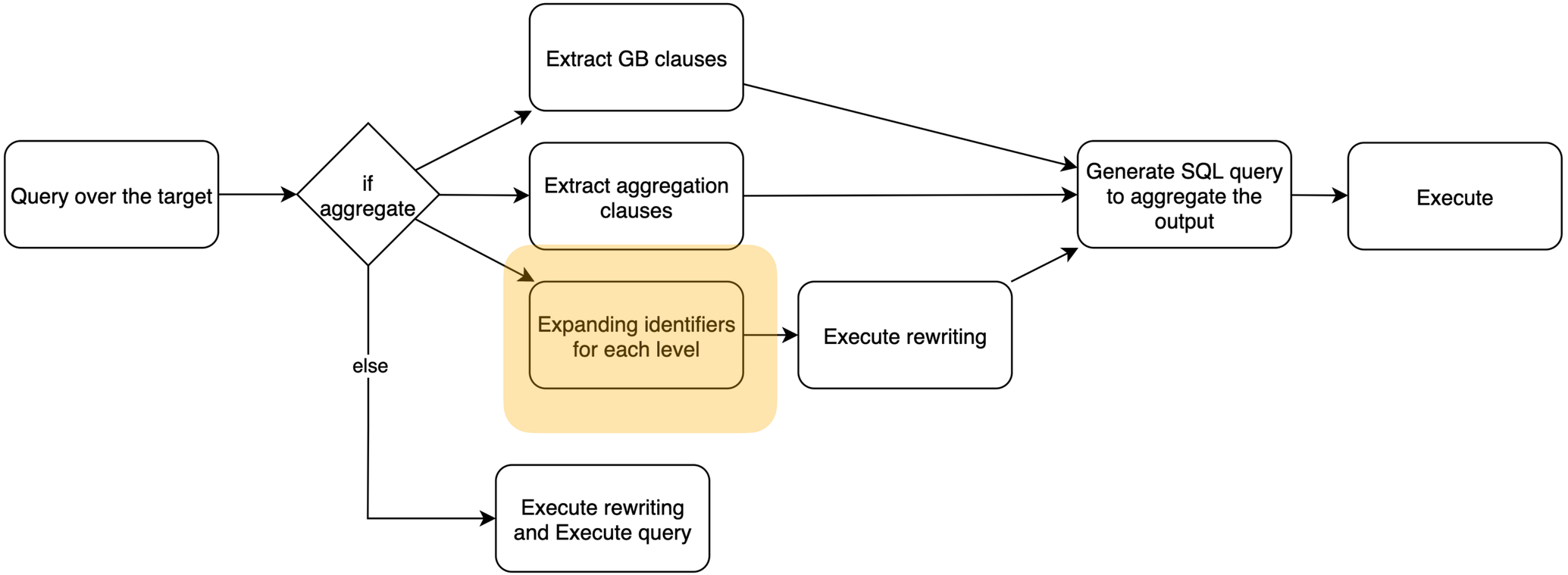
Query



All the features that are instance of Measure and are also linked to an aggregating function

The measure are *total revenue* and *number of unit*

Expanding identifiers



Expanding identifiers into Q

Concepts

Features

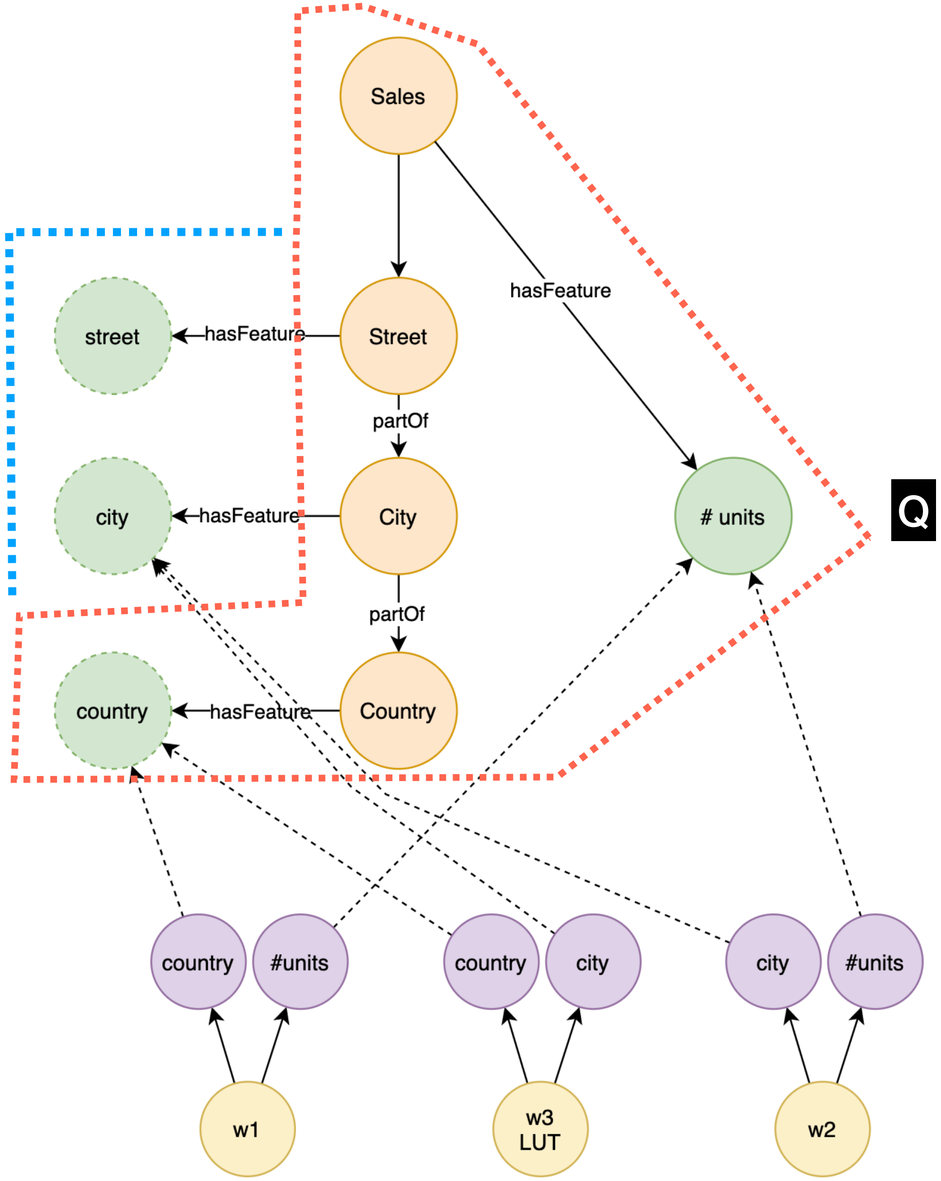
Identifiers

Attributes

Wrappers

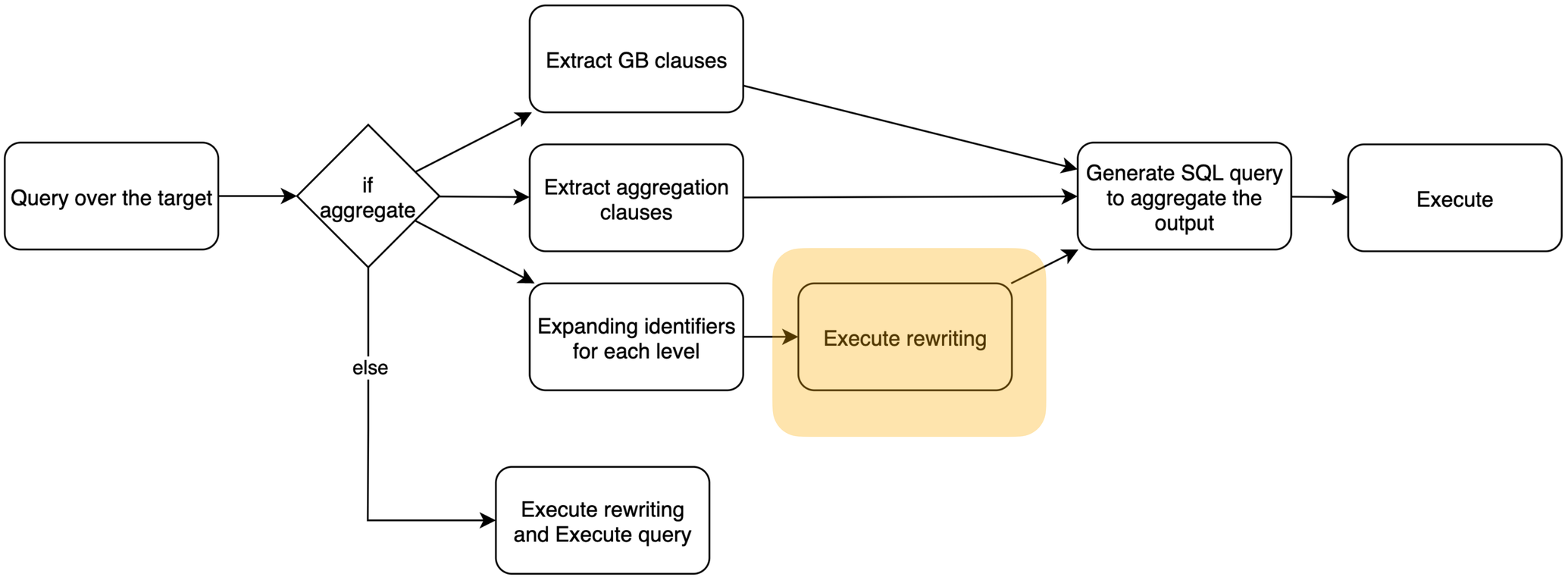
Query

Query extension



Since Q would not actually join dimensional data since the city id is not included in query we have to include all the ids of each level of a queried dimension in Q.

Executing rewriting



Concepts

Features

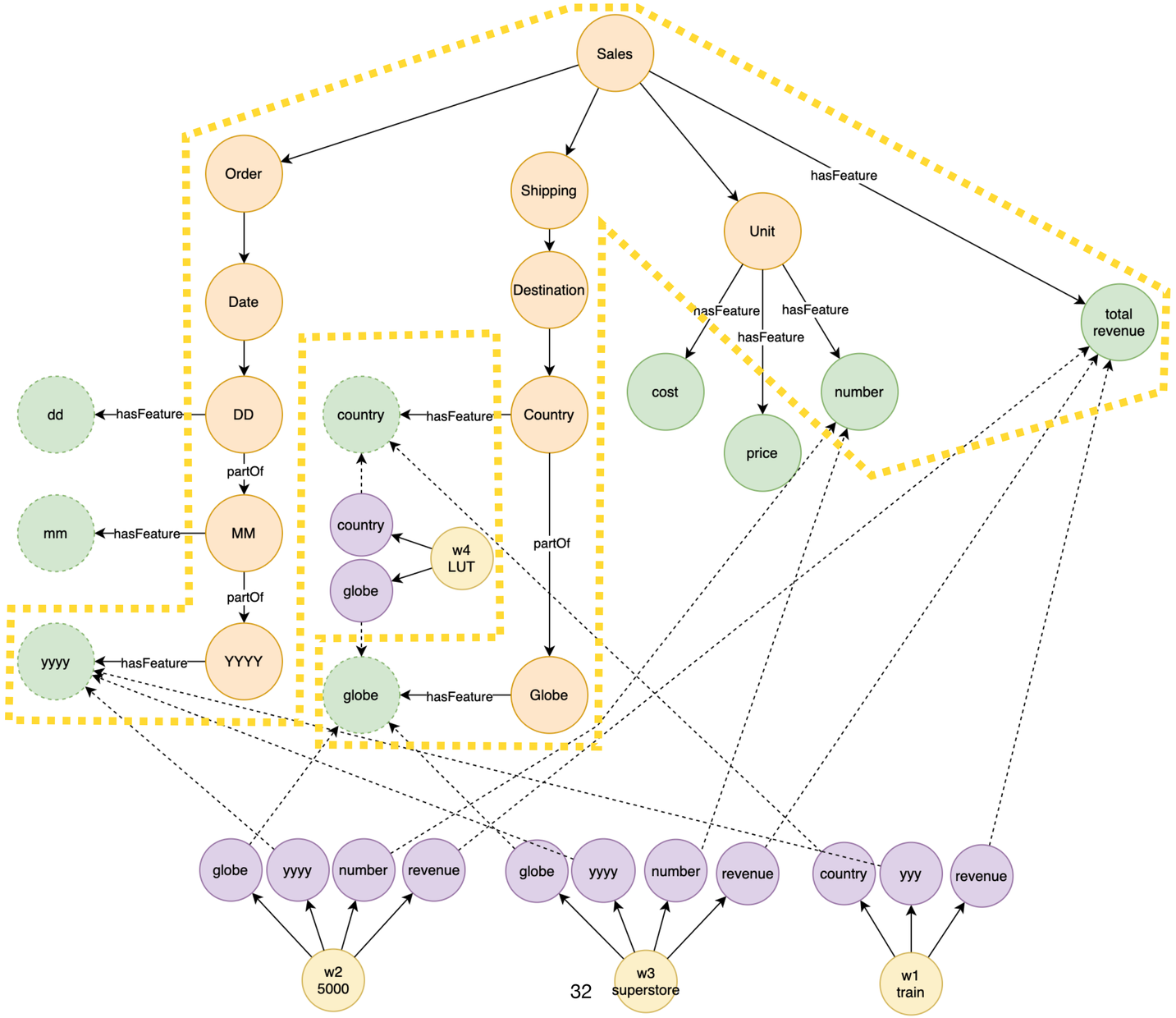
Identifiers

Attributes

Wrappers

Mappings

Executing rewriting



w2 mapping

Concepts

Features

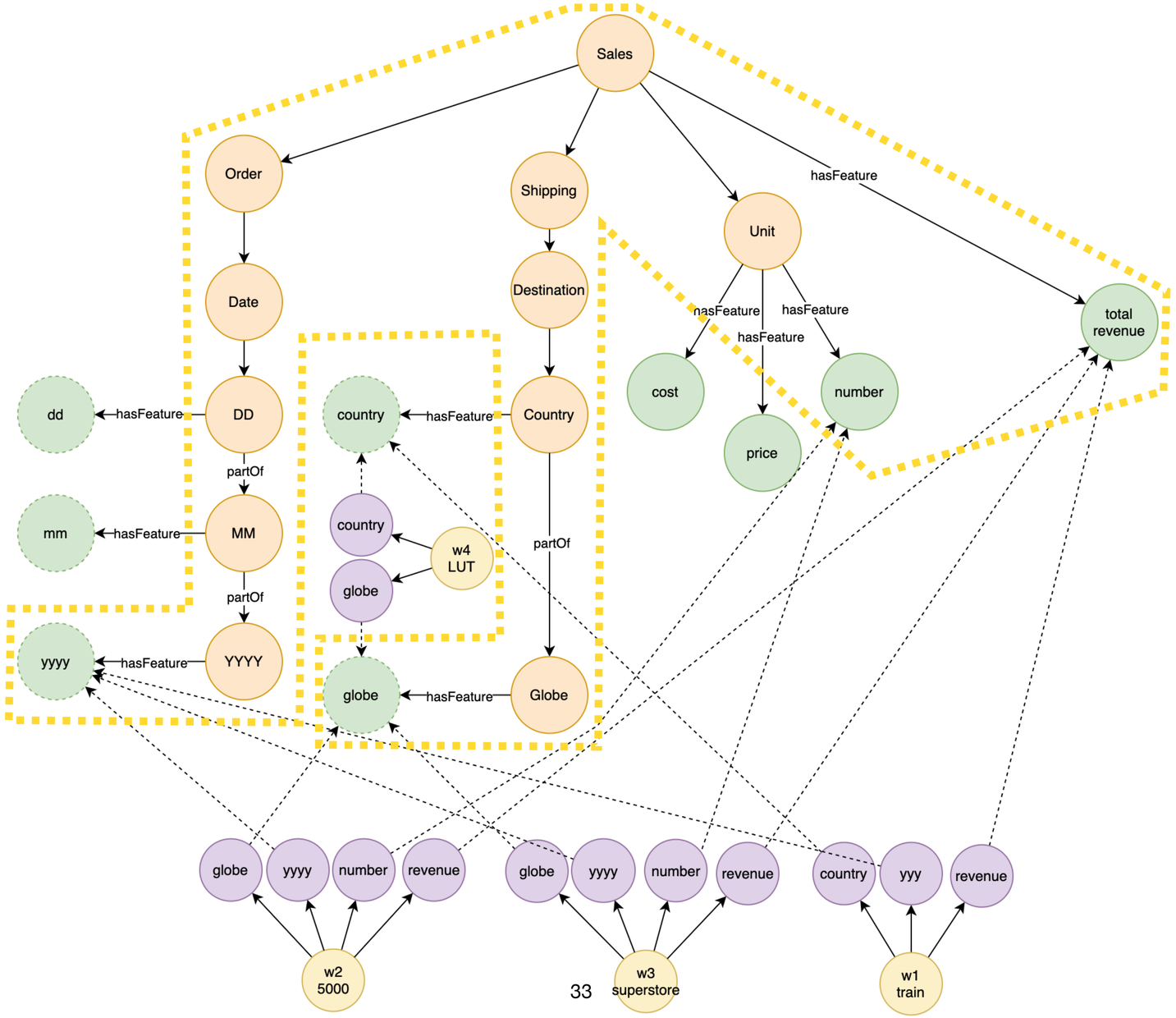
Identifiers

Attributes

Wrappers

Mappings

Executing rewriting



w3 mapping

Concepts

Features

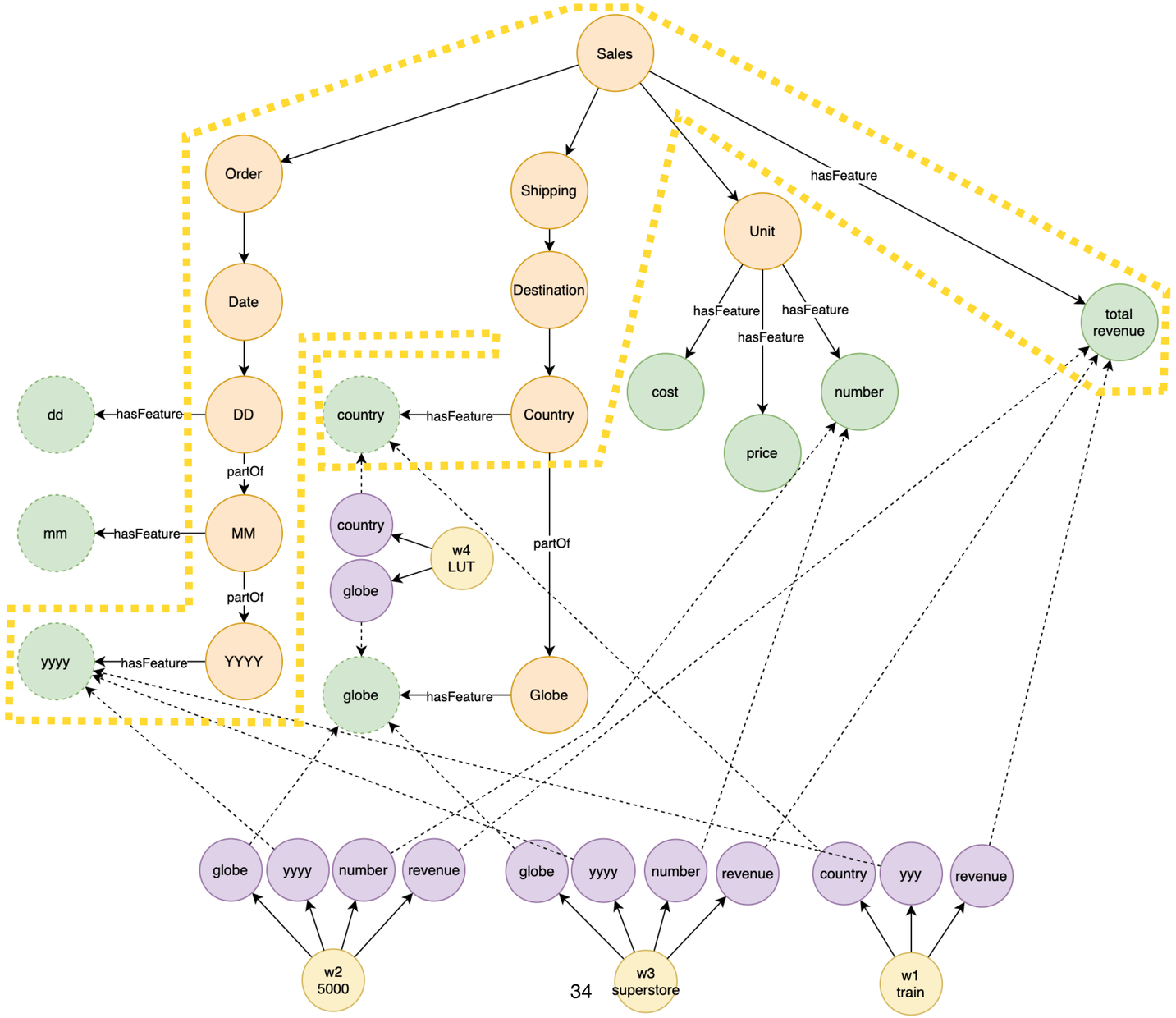
Identifiers

Attributes

Wrappers

Mappings

Executing rewriting



w1 mapping

Concepts

Features

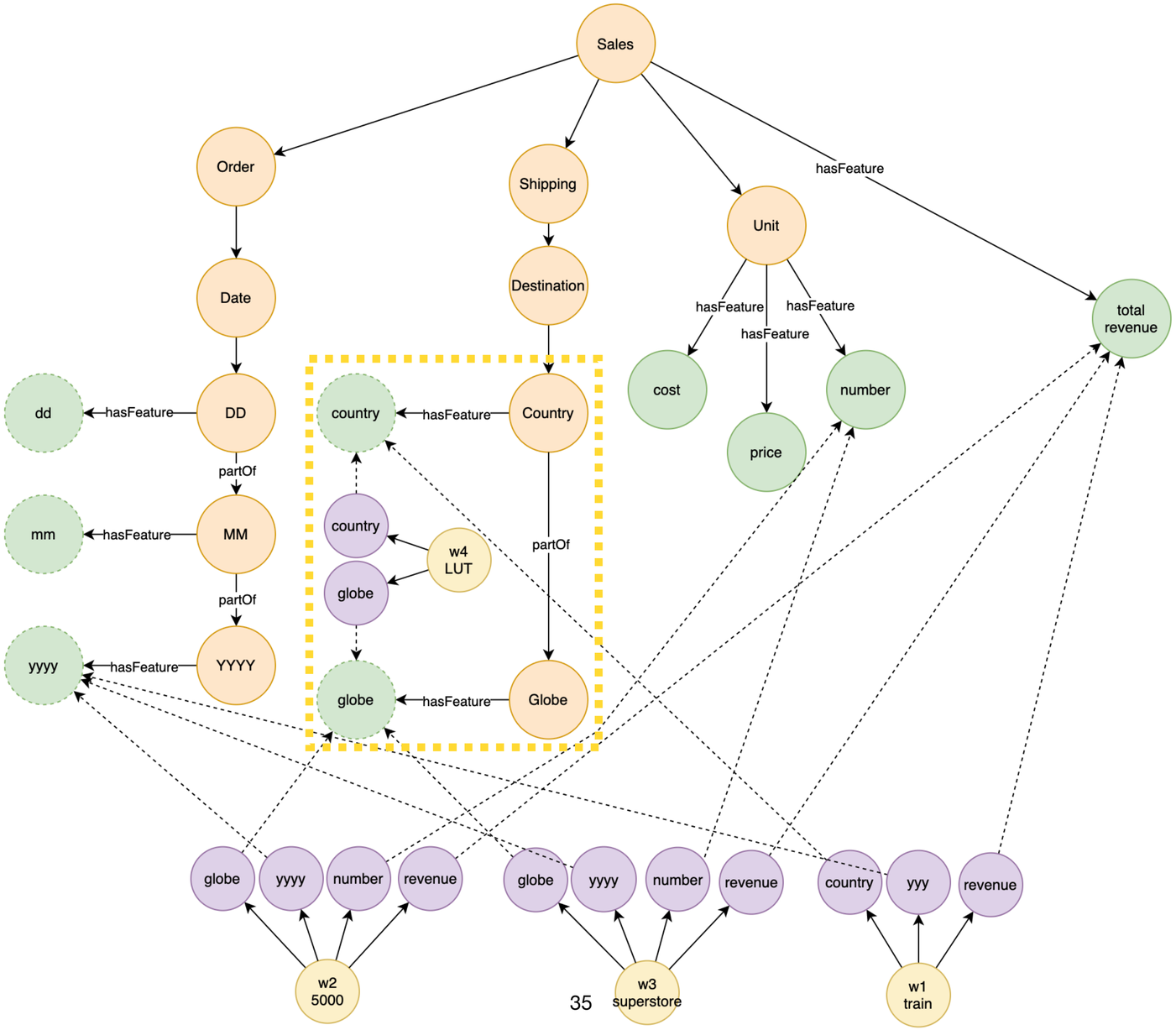
Identifiers

Attributes

Wrappers

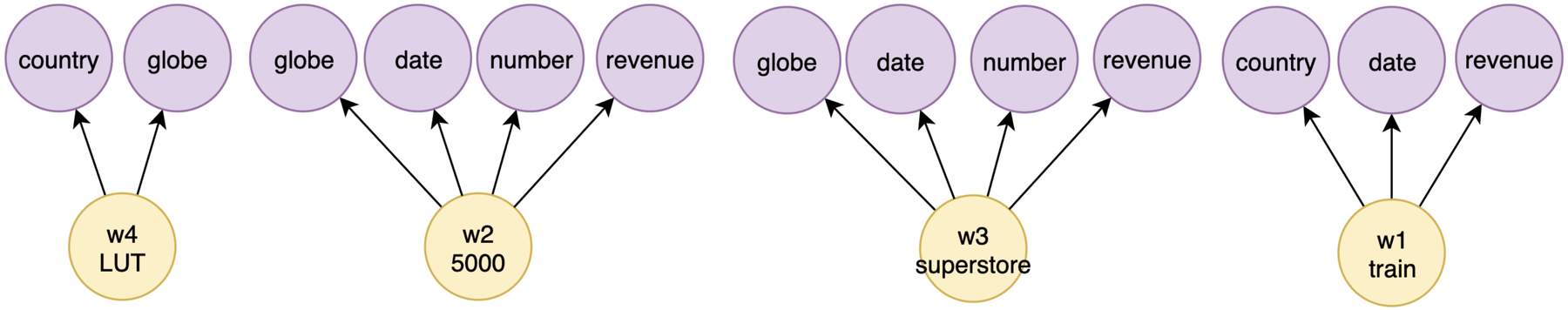
Mappings

Executing rewriting



W4 mapping

Executing rewriting



W1

W_id	Country	Year	Number	Revenue
1	US	2021	50	20000
1	US	2018	50	900
1	US	2019	50	7009
1	US	2021	50	180

⋈

W4

Country	Globe
US	North America

⋈

W3

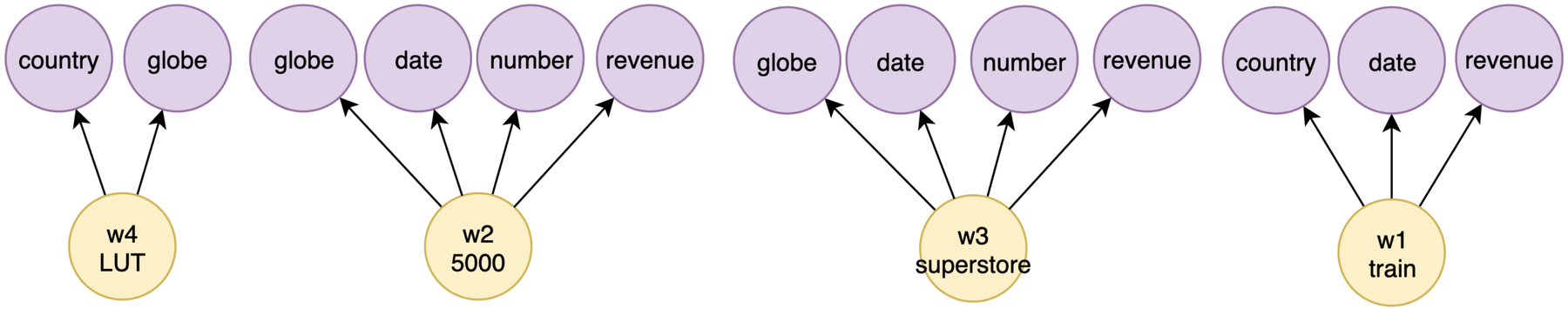
W_id	Globe	Year	Number	Revenue
3	EU	2019	50	200
3	North America	2020	90	2500
3	Asia	2019	800	3000
3	EU	2021	230	9000

U

W2

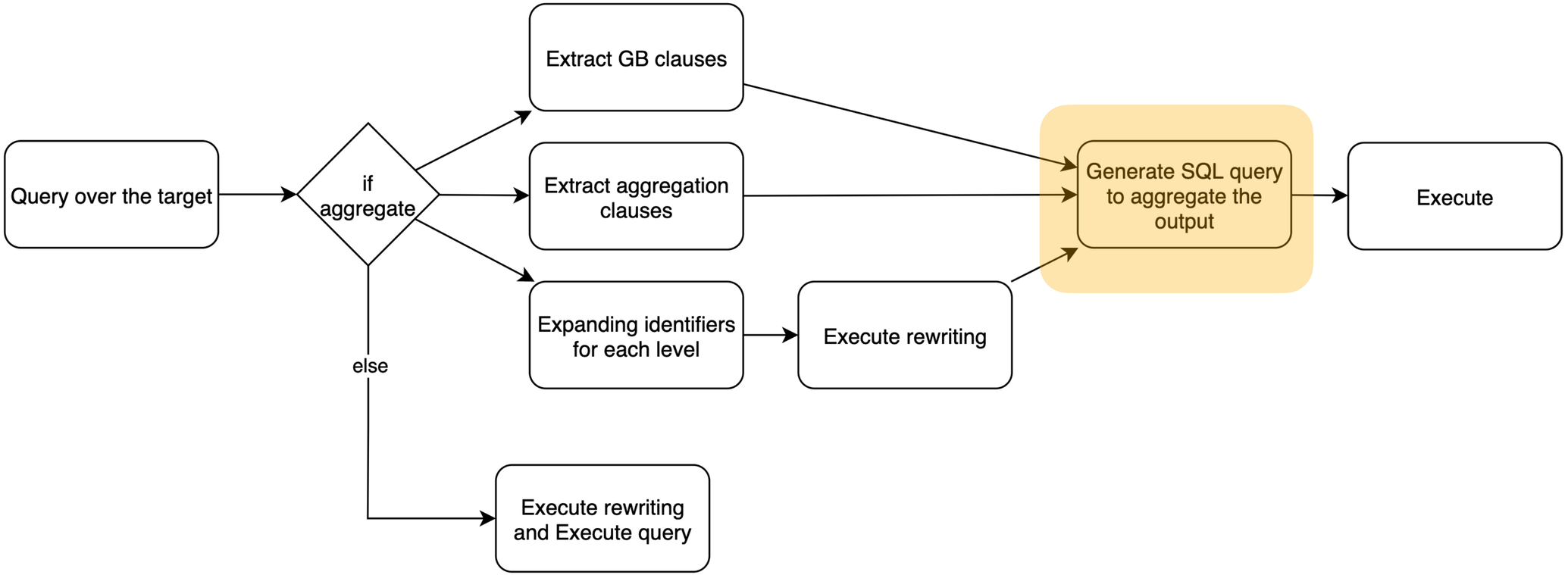
W_id	Globe	Year	Number	Revenue
2	EU	2021	100	3000
2	North America	2020	109	2500
2	North America	2020	90	340
2	Africa	2021	80	930

Executing rewriting



`country, globe, YYYY, number, revenue((w1 ⋈ w4 ⋈ w3) U w2)` → `View V (SQL query)`

Generating SQL query



Generating SQL query

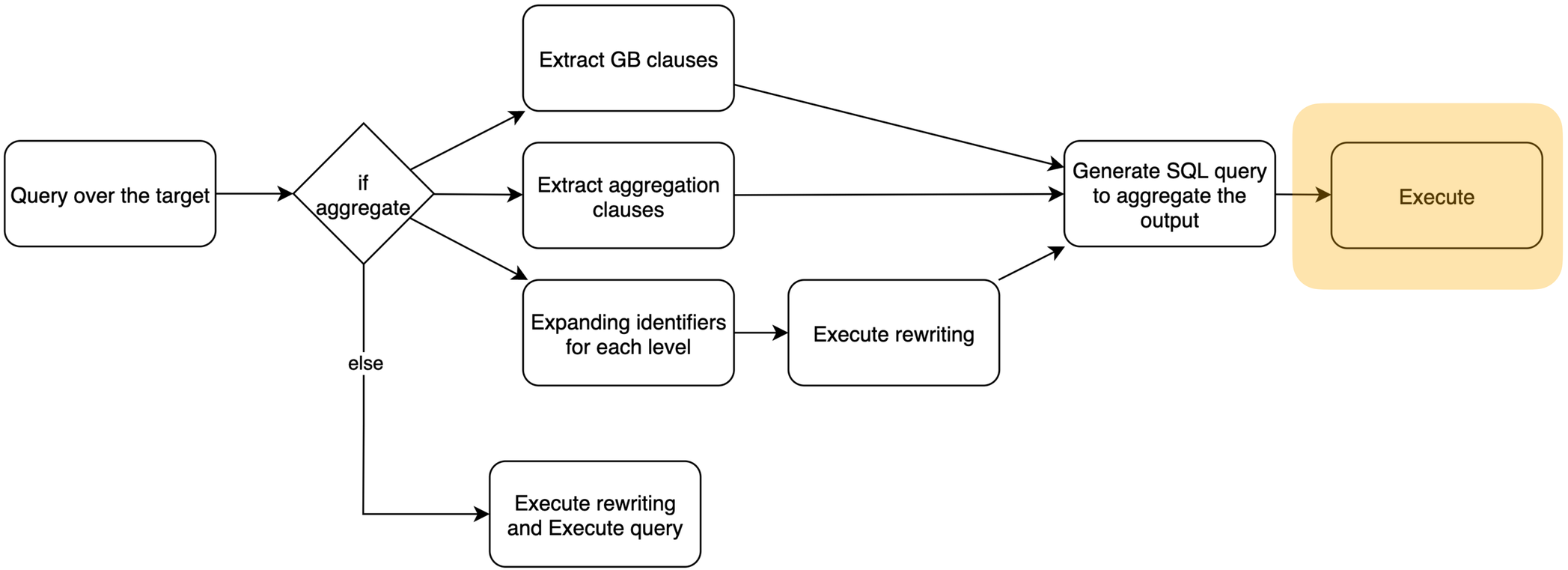
```
SELECT _gb_,_ag_
FROM (V)
GROUP BY _gb_
```

The query template.

Replace the fields:
_gb_ => group by clauses
_ag_ => aggregation clauses
V => the view given by the Rewriting Algorithm

```
SELECT Year, Globe, sum(number) as number, sum(revenue) as revenue
FROM V
GROUP BY Year, Globe, [WrapperID]
```

Executing the SQL query



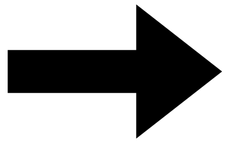
Executing the SQL query for the View (In the FROM clause)

`exec(country, globe, YYYY, number, revenue((w1 ⋈ w4 ⋈ w3) U w2))`

W_id	Country	Globe	Year	Number	Revenue
2		EU	2019	50	200
2		North America	2020	90	2500
2		Asia	2019	800	3000
2		EU	2021	230	9000
1	US	North America	2021	50	20000
1	US	North America	2018	50	900
1	US	North America	2019	50	7009
1	US	North America	2021	50	180
3		EU	2021	100	3000
3		North America	2020	109	2500
3		North America	2020	90	340
3		Africa	2021	80	930

Executing the SQL query

Globe	Year	Number	Revenue
EU	2019	50	200
North America	2020	90	2500
Asia	2019	800	3000
EU	2021	230	9000
North America	2021	50	20000
North America	2018	50	900
North America	2019	50	7009
North America	2021	50	180
EU	2021	100	3000
North America	2020	109	2500
North America	2020	90	340
Africa	2021	80	930



Globe	Year	Number	Revenue
EU	2019	50	200
EU	2021	330	12000
North America	2021	100	20180
North America	2020	289	5340
North America	2019	50	7009
North America	2018	50	900
Africa	2021	80	930
Asia	2019	800	3000

```
SELECT Year, Globe, sum(number) as number, sum(revenue) as revenue
FROM V
GROUP BY Year, Globe, [WrapperID]
```