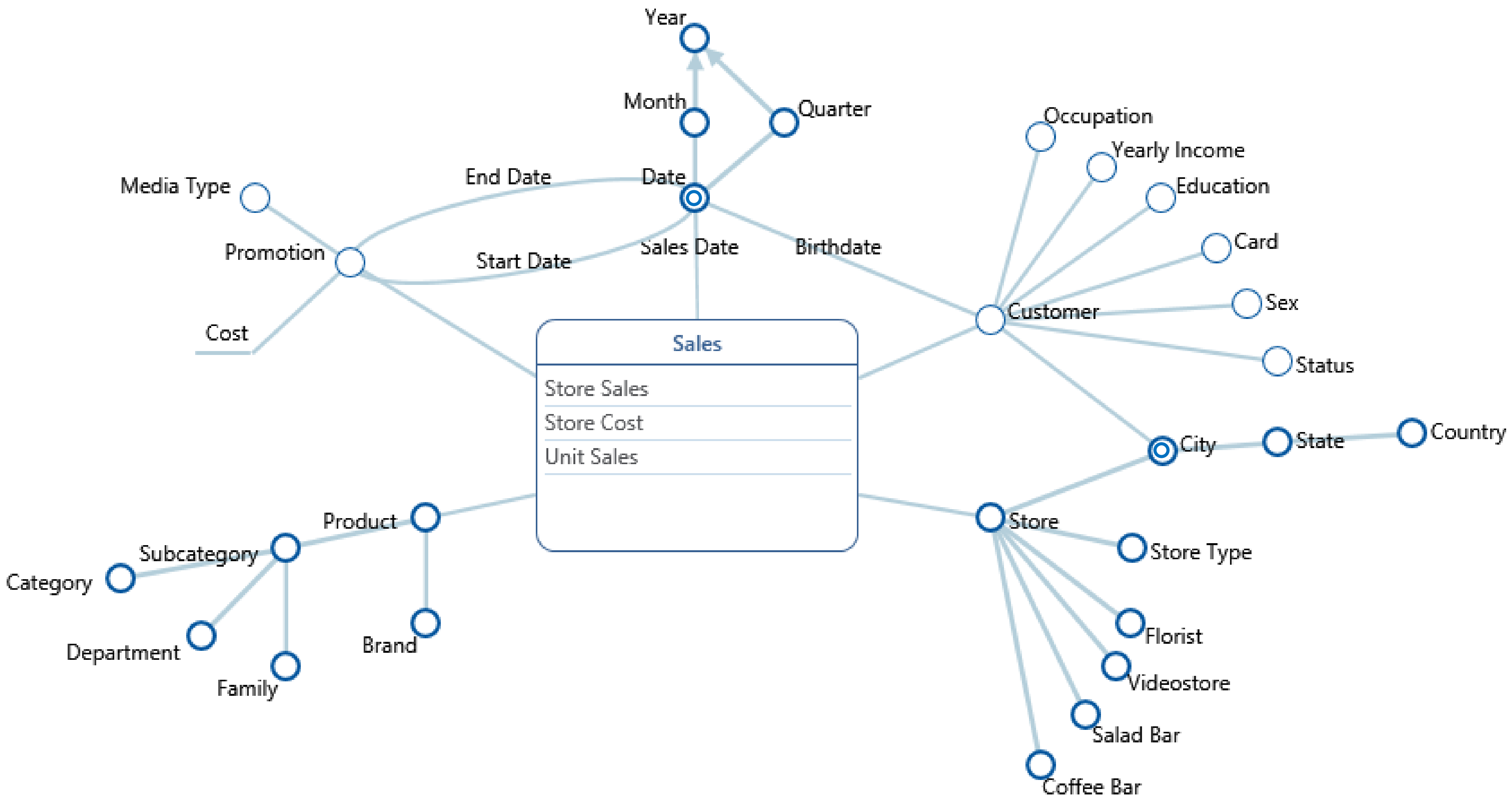
Self-service BI

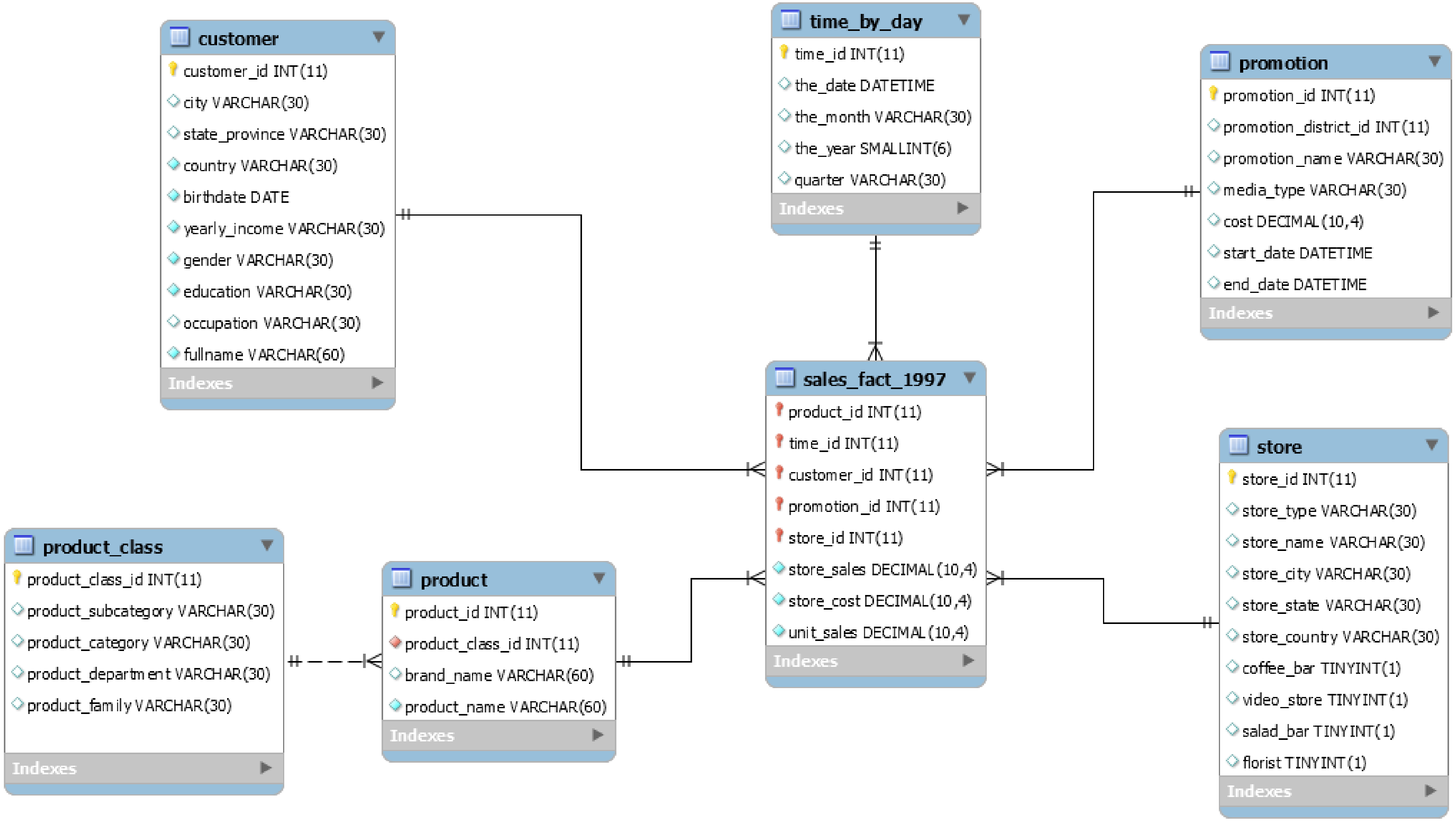
Hands on Tableau

A Brief Recap

DFM - Foodmart



Database Structure – Foodmart (Sales)



customer
customer_id INT(11)
city VARCHAR(30)
state_province VARCHAR(30)
country VARCHAR(30)
birthdate DATE
yearly_income VARCHAR(30)
gender VARCHAR(30)
education VARCHAR(30)
occupation VARCHAR(30)
fullname VARCHAR(60)
Indexes

time_by_day
time_id INT(11)
the_date DATETIME
the_month VARCHAR(30)
the_year SMALLINT(6)
quarter VARCHAR(30)
Indexes

promotion
promotion_id INT(11)
promotion_district_id INT(11)
promotion_name VARCHAR(30)
media_type VARCHAR(30)
cost DECIMAL(10,4)
start_date DATETIME
end_date DATETIME
Indexes

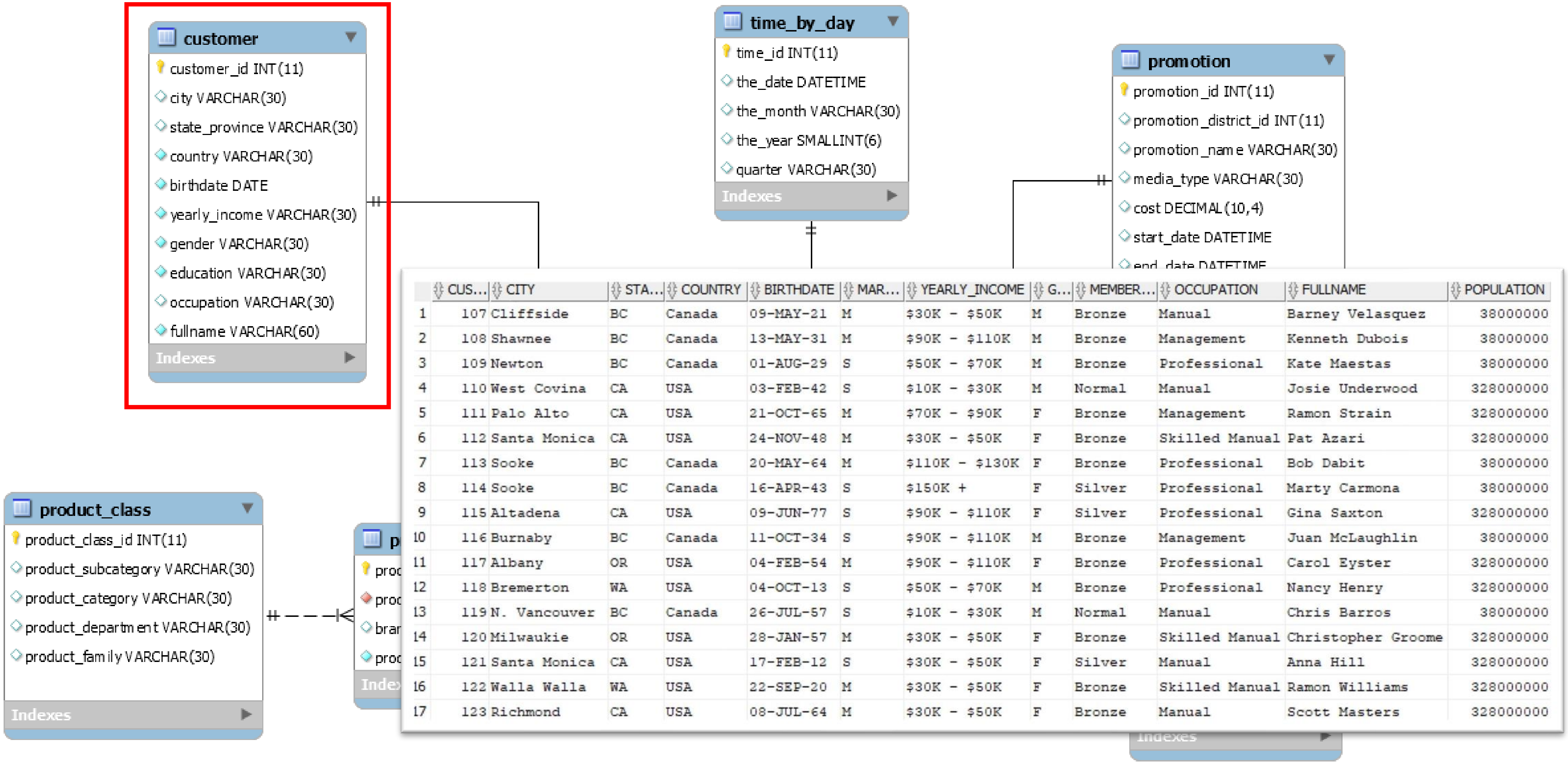
sales_fact_1997
product_id INT(11)
time_id INT(11)
customer_id INT(11)
promotion_id INT(11)
store_id INT(11)
store_sales DECIMAL(10,4)
store_cost DECIMAL(10,4)
unit_sales DECIMAL(10,4)
Indexes

store
store_id INT(11)
store_type VARCHAR(30)
store_name VARCHAR(30)
store_city VARCHAR(30)
store_state VARCHAR(30)
store_country VARCHAR(30)
coffee_bar TINYINT(1)
video_store TINYINT(1)
salad_bar TINYINT(1)
florist TINYINT(1)
Indexes

product_class
product_class_id INT(11)
product_subcategory VARCHAR(30)
product_category VARCHAR(30)
product_department VARCHAR(30)
product_family VARCHAR(30)
Indexes

product
product_id INT(11)
product_class_id INT(11)
brand_name VARCHAR(60)
product_name VARCHAR(60)
Indexes

Database Structure – Foodmart (Sales)



customer
customer_id INT(11)
city VARCHAR(30)
state_province VARCHAR(30)
country VARCHAR(30)
birthdate DATE
yearly_income VARCHAR(30)
gender VARCHAR(30)
education VARCHAR(30)
occupation VARCHAR(30)
fullname VARCHAR(60)
Indexes

time_by_day
time_id INT(11)
the_date DATETIME
the_month VARCHAR(30)
the_year SMALLINT(6)
quarter VARCHAR(30)
Indexes

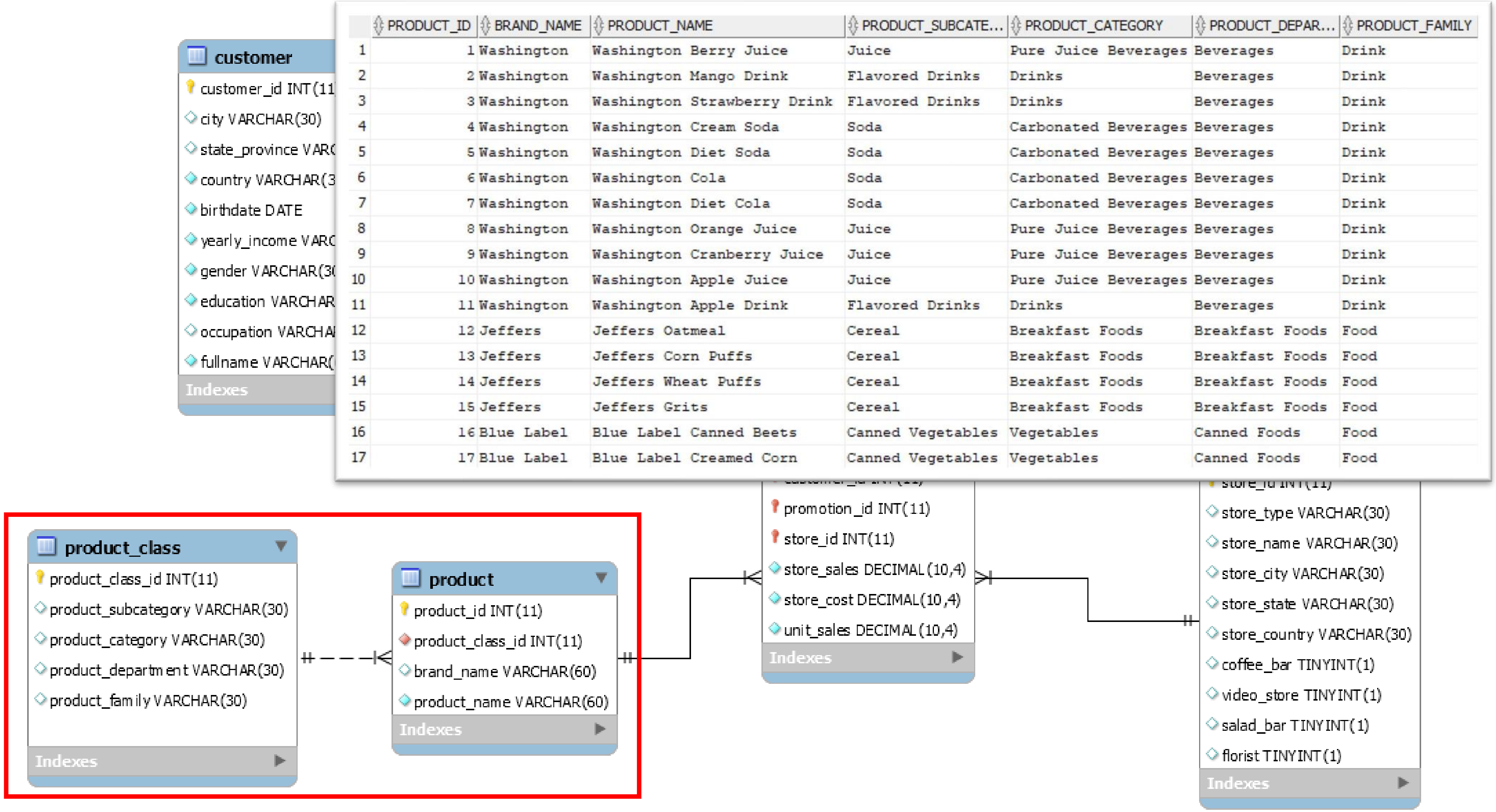
promotion
promotion_id INT(11)
promotion_district_id INT(11)
promotion_name VARCHAR(30)
media_type VARCHAR(30)
cost DECIMAL(10,4)
start_date DATETIME
end_date DATETIME
Indexes

product_class
product_class_id INT(11)
product_subcategory VARCHAR(30)
product_category VARCHAR(30)
product_department VARCHAR(30)
product_family VARCHAR(30)
Indexes

p
proc
proc
brar
proc
Index

	CUS...	CITY	STA...	COUNTRY	BIRTHDATE	MAR...	YEARLY_INCOME	G...	MEMBER...	OCCUPATION	FULLNAME	POPULATION
1	107	Cliffside	BC	Canada	09-MAY-21	M	$30K - $50K	M	Bronze	Manual	Barney Velasquez	38000000
2	108	Shawnee	BC	Canada	13-MAY-31	M	$90K - $110K	M	Bronze	Management	Kenneth Dubois	38000000
3	109	Newton	BC	Canada	01-AUG-29	S	$50K - $70K	M	Bronze	Professional	Kate Maestas	38000000
4	110	West Covina	CA	USA	03-FEB-42	S	$10K - $30K	M	Normal	Manual	Josie Underwood	328000000
5	111	Palo Alto	CA	USA	21-OCT-65	M	$70K - $90K	F	Bronze	Management	Ramon Strain	328000000
6	112	Santa Monica	CA	USA	24-NOV-48	M	$30K - $50K	F	Bronze	Skilled Manual	Pat Azari	328000000
7	113	Sooke	BC	Canada	20-MAY-64	M	$110K - $130K	F	Bronze	Professional	Bob Dabit	38000000
8	114	Sooke	BC	Canada	16-APR-43	S	$150K +	F	Silver	Professional	Marty Carmona	38000000
9	115	Altadena	CA	USA	09-JUN-77	S	$90K - $110K	F	Silver	Professional	Gina Saxton	328000000
10	116	Burnaby	BC	Canada	11-OCT-34	S	$90K - $110K	M	Bronze	Management	Juan McLaughlin	38000000
11	117	Albany	OR	USA	04-FEB-54	M	$90K - $110K	F	Bronze	Professional	Carol Eyster	328000000
12	118	Bremerton	WA	USA	04-OCT-13	S	$50K - $70K	M	Bronze	Professional	Nancy Henry	328000000
13	119	N. Vancouver	BC	Canada	26-JUL-57	S	$10K - $30K	M	Normal	Manual	Chris Barros	38000000
14	120	Milwaukie	OR	USA	28-JAN-57	M	$30K - $50K	F	Bronze	Skilled Manual	Christopher Groome	328000000
15	121	Santa Monica	CA	USA	17-FEB-12	S	$30K - $50K	F	Silver	Manual	Anna Hill	328000000
16	122	Walla Walla	WA	USA	22-SEP-20	M	$30K - $50K	F	Bronze	Skilled Manual	Ramon Williams	328000000
17	123	Richmond	CA	USA	08-JUL-64	M	$30K - $50K	F	Bronze	Manual	Scott Masters	328000000

Database Structure – Foodmart (Sales)



	PRODUCT_ID	BRAND_NAME	PRODUCT_NAME	PRODUCT_SUBCATE...	PRODUCT_CATEGORY	PRODUCT_DEPAR...	PRODUCT_FAMILY
1	1	Washington	Washington Berry Juice	Juice	Pure Juice Beverages	Beverages	Drink
2	2	Washington	Washington Mango Drink	Flavored Drinks	Drinks	Beverages	Drink
3	3	Washington	Washington Strawberry Drink	Flavored Drinks	Drinks	Beverages	Drink
4	4	Washington	Washington Cream Soda	Soda	Carbonated Beverages	Beverages	Drink
5	5	Washington	Washington Diet Soda	Soda	Carbonated Beverages	Beverages	Drink
6	6	Washington	Washington Cola	Soda	Carbonated Beverages	Beverages	Drink
7	7	Washington	Washington Diet Cola	Soda	Carbonated Beverages	Beverages	Drink
8	8	Washington	Washington Orange Juice	Juice	Pure Juice Beverages	Beverages	Drink
9	9	Washington	Washington Cranberry Juice	Juice	Pure Juice Beverages	Beverages	Drink
10	10	Washington	Washington Apple Juice	Juice	Pure Juice Beverages	Beverages	Drink
11	11	Washington	Washington Apple Drink	Flavored Drinks	Drinks	Beverages	Drink
12	12	Jeffers	Jeffers Oatmeal	Cereal	Breakfast Foods	Breakfast Foods	Food
13	13	Jeffers	Jeffers Corn Puffs	Cereal	Breakfast Foods	Breakfast Foods	Food
14	14	Jeffers	Jeffers Wheat Puffs	Cereal	Breakfast Foods	Breakfast Foods	Food
15	15	Jeffers	Jeffers Grits	Cereal	Breakfast Foods	Breakfast Foods	Food
16	16	Blue Label	Blue Label Canned Beets	Canned Vegetables	Vegetables	Canned Foods	Food
17	17	Blue Label	Blue Label Creamed Corn	Canned Vegetables	Vegetables	Canned Foods	Food

customer
customer_id INT(11
city VARCHAR(30)
state_province VARC
country VARCHAR(3
birthdate DATE
yearly_income VARC
gender VARCHAR(3
education VARCHAR
occupation VARCHA
fullname VARCHAR(
Indexes

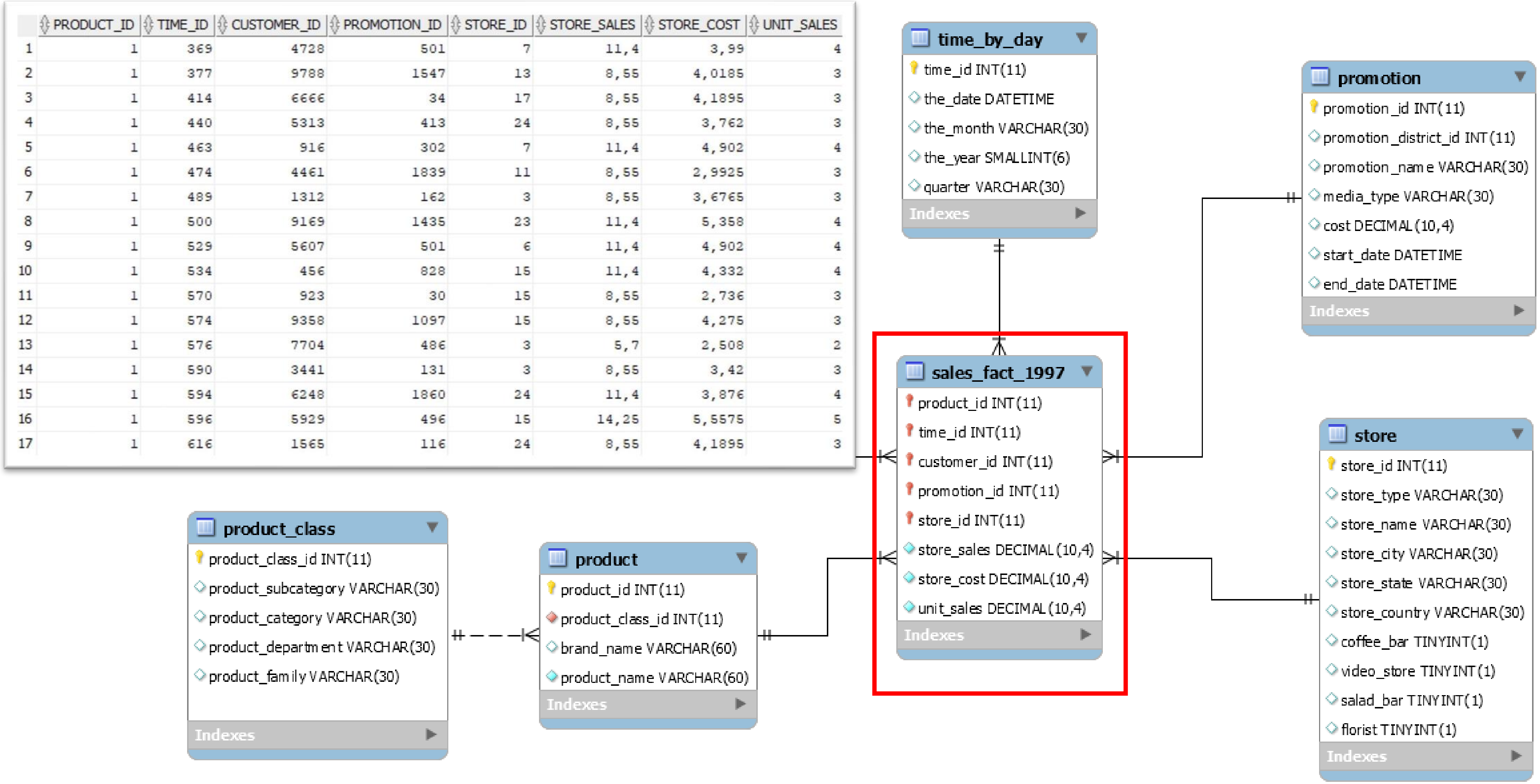
product_class
product_class_id INT(11)
product_subcategory VARCHAR(30)
product_category VARCHAR(30)
product_department VARCHAR(30)
product_family VARCHAR(30)
Indexes

product
product_id INT(11)
product_class_id INT(11)
brand_name VARCHAR(60)
product_name VARCHAR(60)
Indexes

customer_id INT(11)
promotion_id INT(11)
store_id INT(11)
store_sales DECIMAL(10,4)
store_cost DECIMAL(10,4)
unit_sales DECIMAL(10,4)
Indexes

store_id INT(11)
store_type VARCHAR(30)
store_name VARCHAR(30)
store_city VARCHAR(30)
store_state VARCHAR(30)
store_country VARCHAR(30)
coffee_bar TINYINT(1)
video_store TINYINT(1)
salad_bar TINYINT(1)
florist TINYINT(1)
Indexes

Database Structure – Foodmart (Sales)



	PRODUCT_ID	TIME_ID	CUSTOMER_ID	PROMOTION_ID	STORE_ID	STORE_SALES	STORE_COST	UNIT_SALES
1	1	369	4728	501	7	11,4	3,99	4
2	1	377	9788	1547	13	8,55	4,0185	3
3	1	414	6666	34	17	8,55	4,1895	3
4	1	440	5313	413	24	8,55	3,762	3
5	1	463	916	302	7	11,4	4,902	4
6	1	474	4461	1839	11	8,55	2,9925	3
7	1	489	1312	162	3	8,55	3,6765	3
8	1	500	9169	1435	23	11,4	5,358	4
9	1	529	5607	501	6	11,4	4,902	4
10	1	534	456	828	15	11,4	4,332	4
11	1	570	923	30	15	8,55	2,736	3
12	1	574	9358	1097	15	8,55	4,275	3
13	1	576	7704	486	3	5,7	2,508	2
14	1	590	3441	131	3	8,55	3,42	3
15	1	594	6248	1860	24	11,4	3,876	4
16	1	596	5929	496	15	14,25	5,5575	5
17	1	616	1565	116	24	8,55	4,1895	3

time_by_day
- time_id INT(11)
- the_date DATETIME
- the_month VARCHAR(30)
- the_year SMALLINT(6)
- quarter VARCHAR(30)

Indexes

promotion
- promotion_id INT(11)
- promotion_district_id INT(11)
- promotion_name VARCHAR(30)
- media_type VARCHAR(30)
- cost DECIMAL(10,4)
- start_date DATETIME
- end_date DATETIME

Indexes

sales_fact_1997
- product_id INT(11)
- time_id INT(11)
- customer_id INT(11)
- promotion_id INT(11)
- store_id INT(11)
- store_sales DECIMAL(10,4)
- store_cost DECIMAL(10,4)
- unit_sales DECIMAL(10,4)

Indexes

store
- store_id INT(11)
- store_type VARCHAR(30)
- store_name VARCHAR(30)
- store_city VARCHAR(30)
- store_state VARCHAR(30)
- store_country VARCHAR(30)
- coffee_bar TINYINT(1)
- video_store TINYINT(1)
- salad_bar TINYINT(1)
- florist TINYINT(1)

Indexes

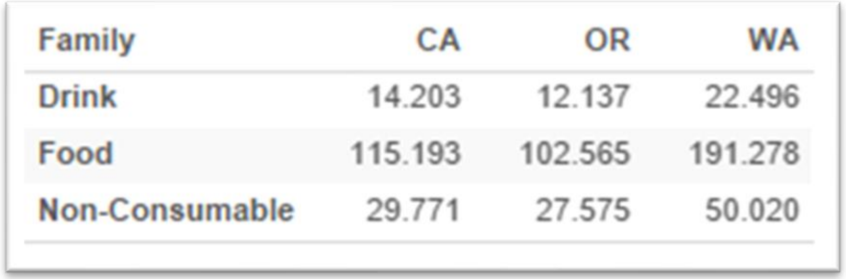
product_class
- product_class_id INT(11)
- product_subcategory VARCHAR(30)
- product_category VARCHAR(30)
- product_department VARCHAR(30)
- product_family VARCHAR(30)

Indexes

product
- product_id INT(11)
- product_class_id INT(11)
- brand_name VARCHAR(60)
- product_name VARCHAR(60)

Indexes

OLAP query



Family	CA	OR	WA
Drink	14.203	12.137	22.496
Food	115.193	102.565	191.278
Non-Consumable	29.771	27.575	50.020

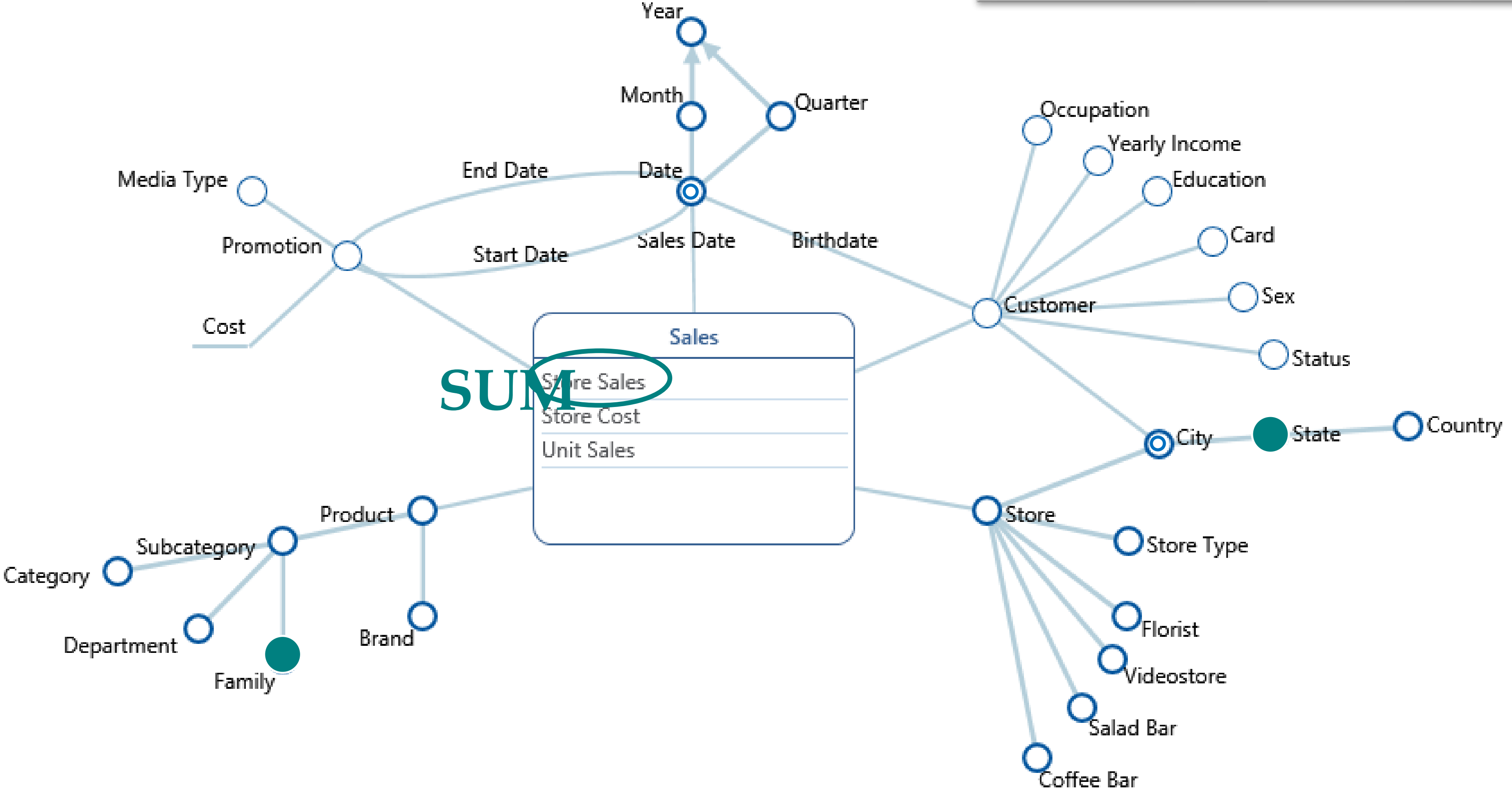
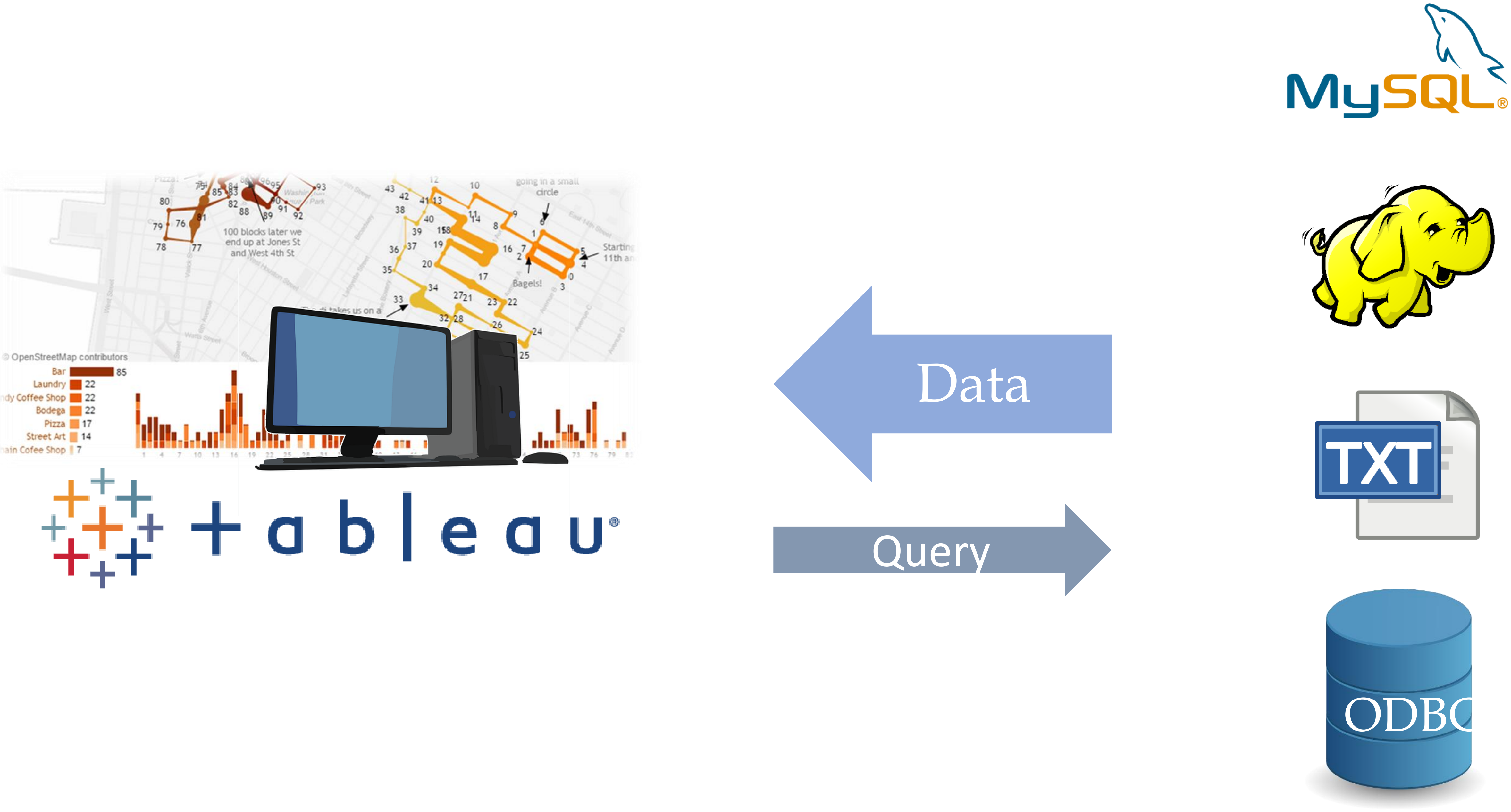
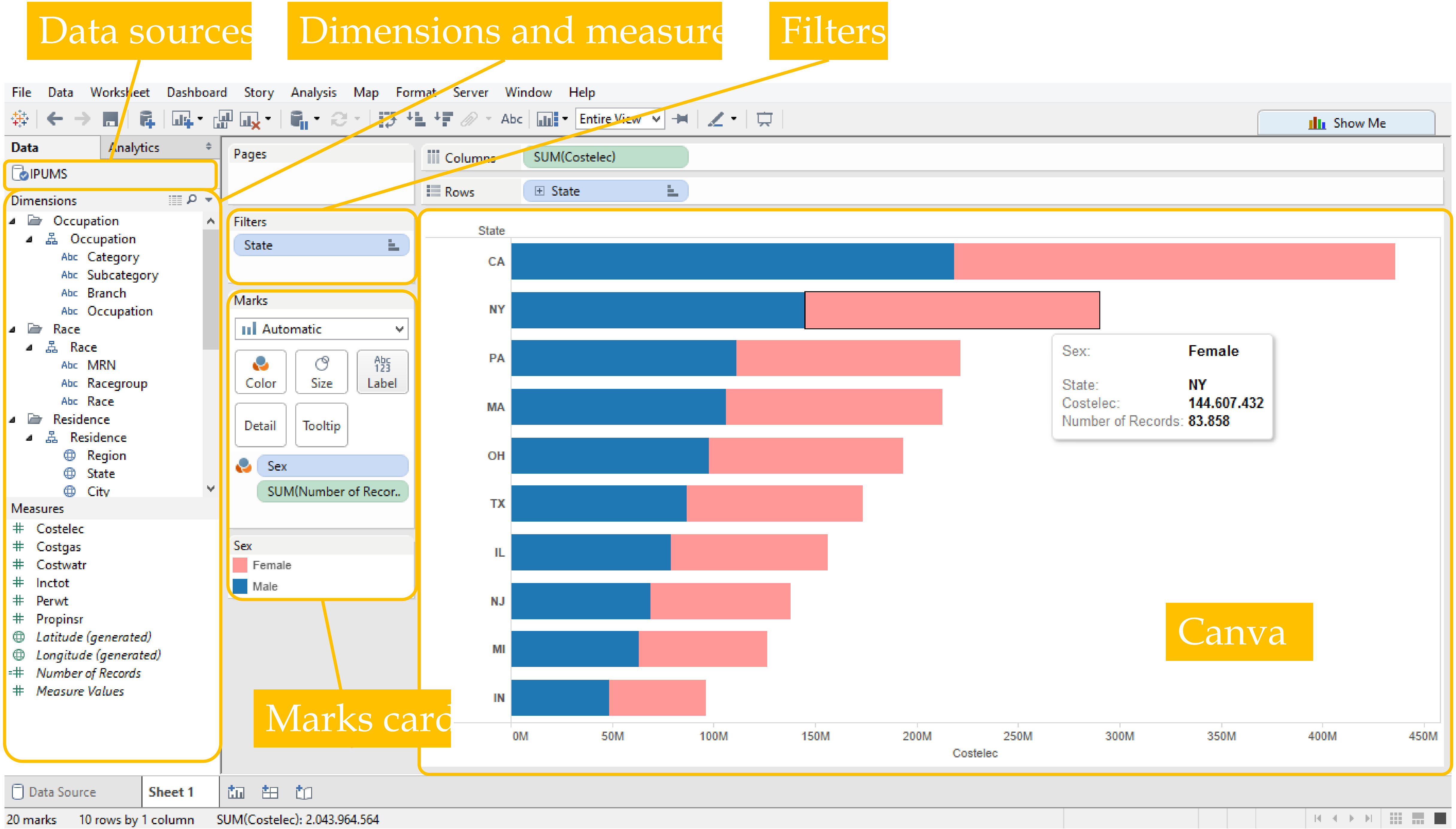


Tableau: Visual Analytics



Data

Query



Data sources

Dimensions and measure

Filters

Canva

Marks card

Columns: SUM(Costelec)

Rows: State

Filters: State

Marks: Automatic

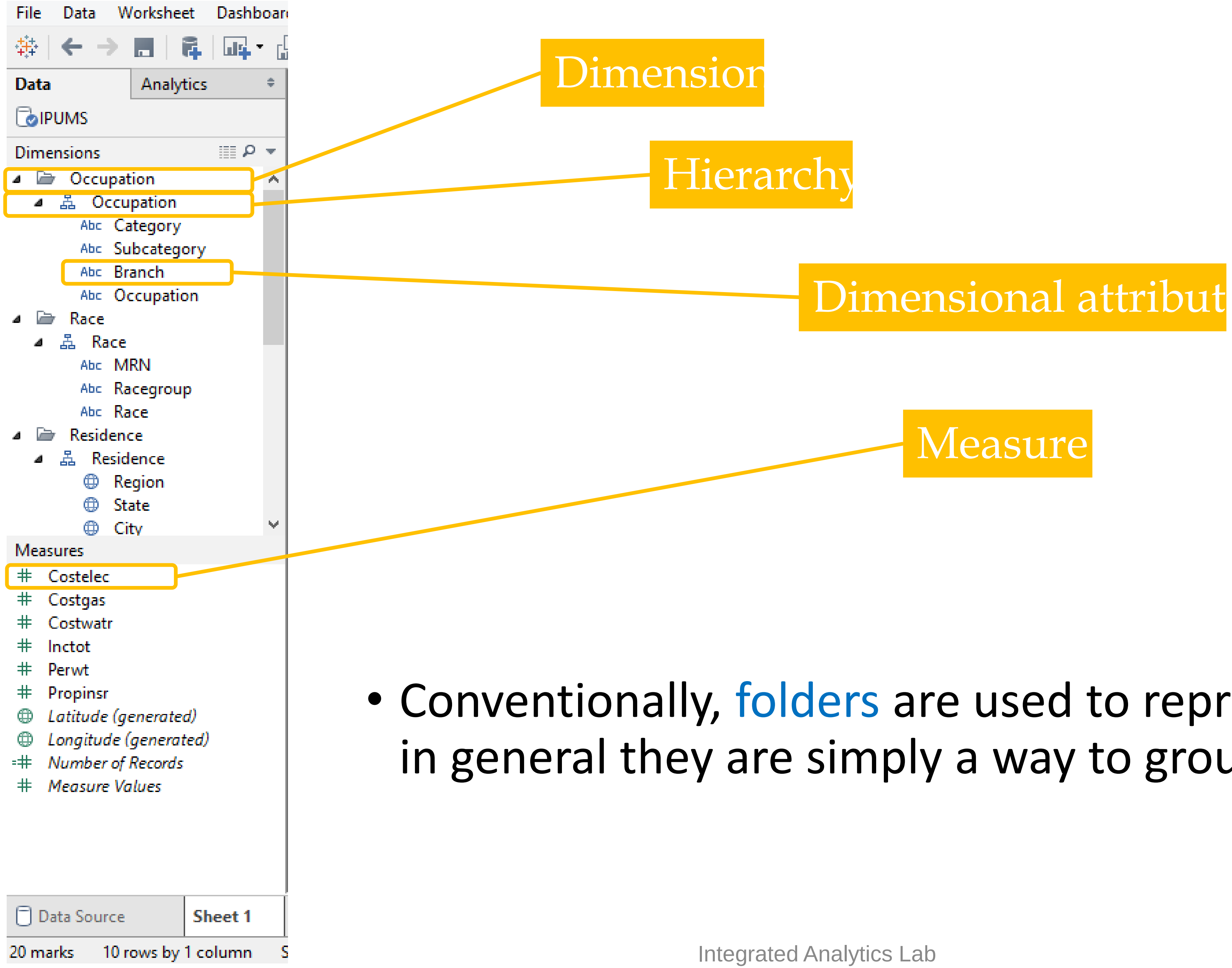
Sex

SUM(Number of Recor..

Sex: Female

State: NY
Costelec: 144.607.432
Number of Records: 83.858

20 marks 10 rows by 1 column SUM(Costelec): 2.043.964.564



Dimension

Hierarchy

Dimensional attribut

Measure

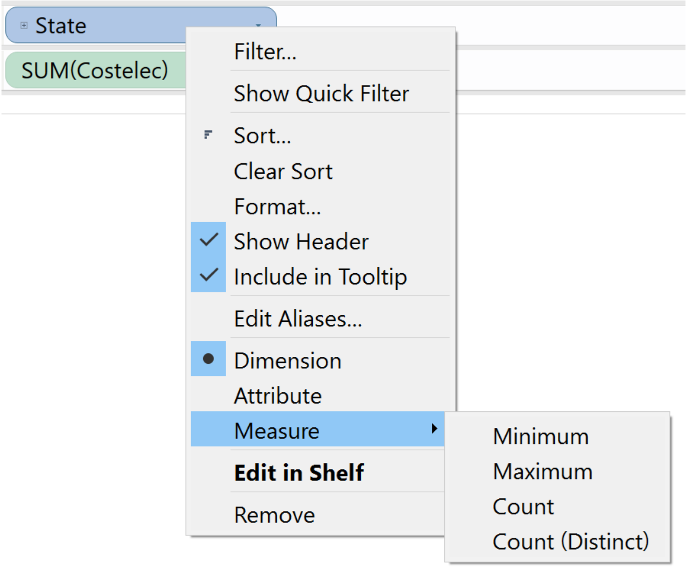
- Conventionally, folders are used to represent dimensions; in general they are simply a way to group elements

Dimension VS Measure

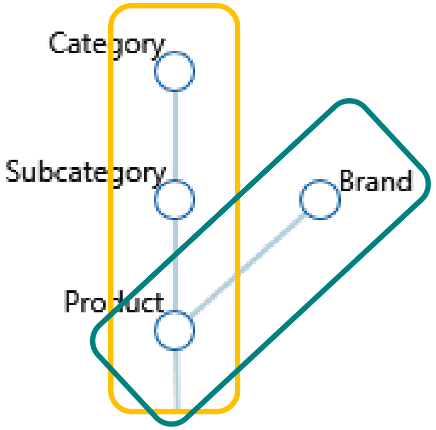
With Tableau the definitions of dimension and measure are quite loose. Indeed, every field can be used both as a dimension and as a measure

Regardless, it is a good idea to give an initial classification to the available fields by following these guidelines

- A dimension is any *independent* field, such as *city*, *product*, etc.
- A measure is any field that is *dependent* on other fields, such as *profit*
- Usually measures are numerical while dimensions are categorical (but not always!)

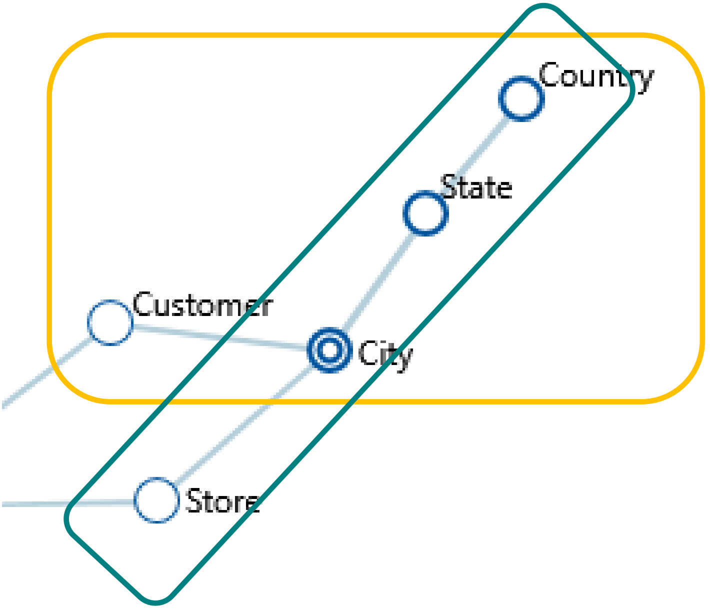


Hierarchies



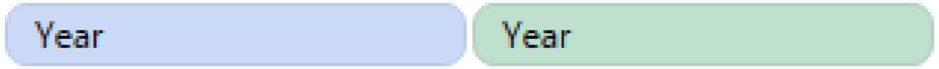
- Each path from leaves to root becomes a different hierarchy; shared attributes are duplicated

- Shared hierarchies are duplicated



Green VS Blue

With Tableau, the greed colour is associated to *continuous* fields, while the blue colour refers to *discrete* ones



Often (but not always) measures are continuous fields, while dimensions are discrete

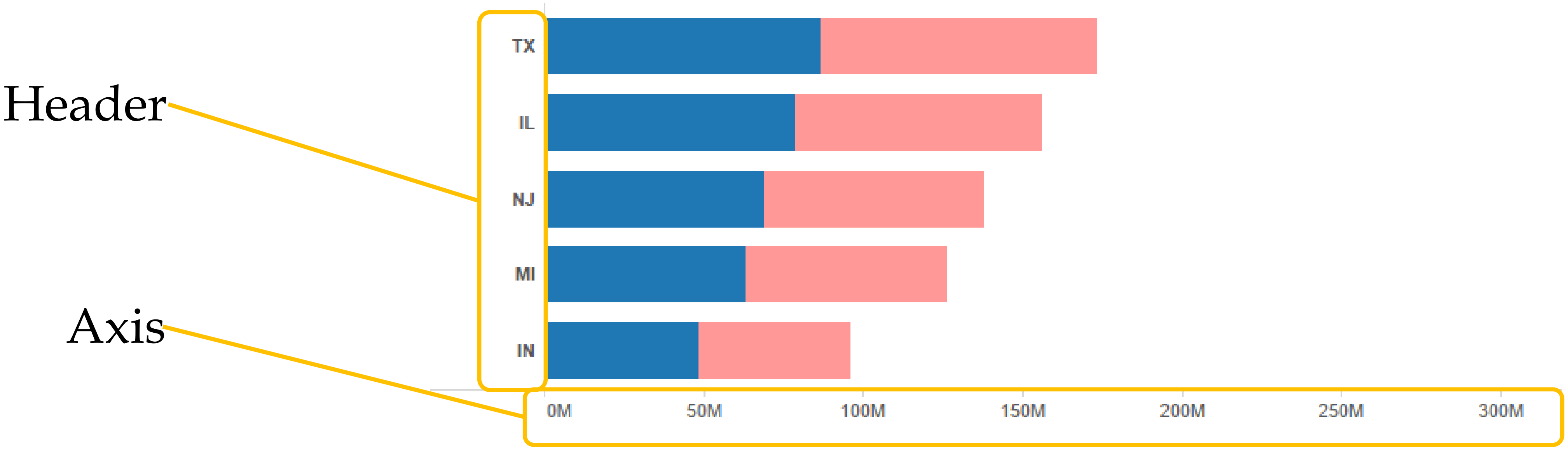
Continuous and discrete fields behave in different ways

- When they are used on rows and columns
- When a filter is applied on them
- When they are associated to colours (i.e., with a colour marker)

Green VS Blue (Rows and Columns)

When they are used on rows and columns

- A discrete field generates a *header* where each value has a its own label
- A *continuous* field generates an *axis*



Self-service BI

Connectors

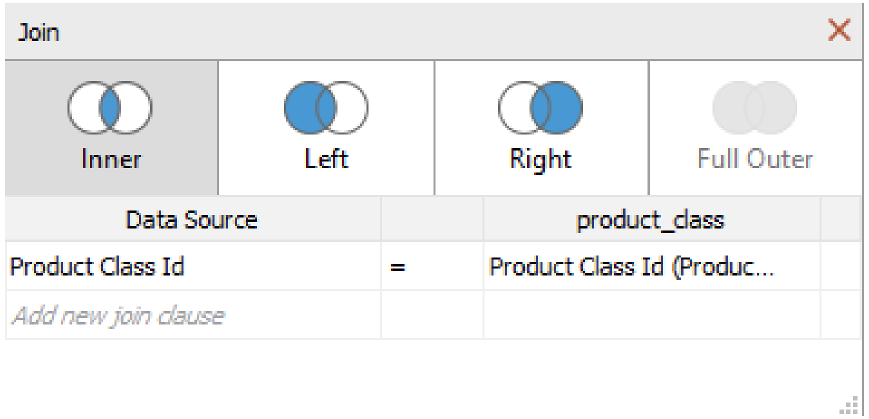
Tableau is able to connect to different data sources

- Text files (DSV) and Excel
- Relational DBMSs (e.g., MySQL, Oracle and SQL Server)
- Big Data (e.g., Hive, Spark SQL and Impala)
- Etc.

Tableau needs a *tabular view* of the data (i.e., a set of tables)

- Different tables from the same source can be linked through joins
- A special exception arises when using JSON data; in this case Tableau (with the help of the user) converts the JSON structure into a table

Join



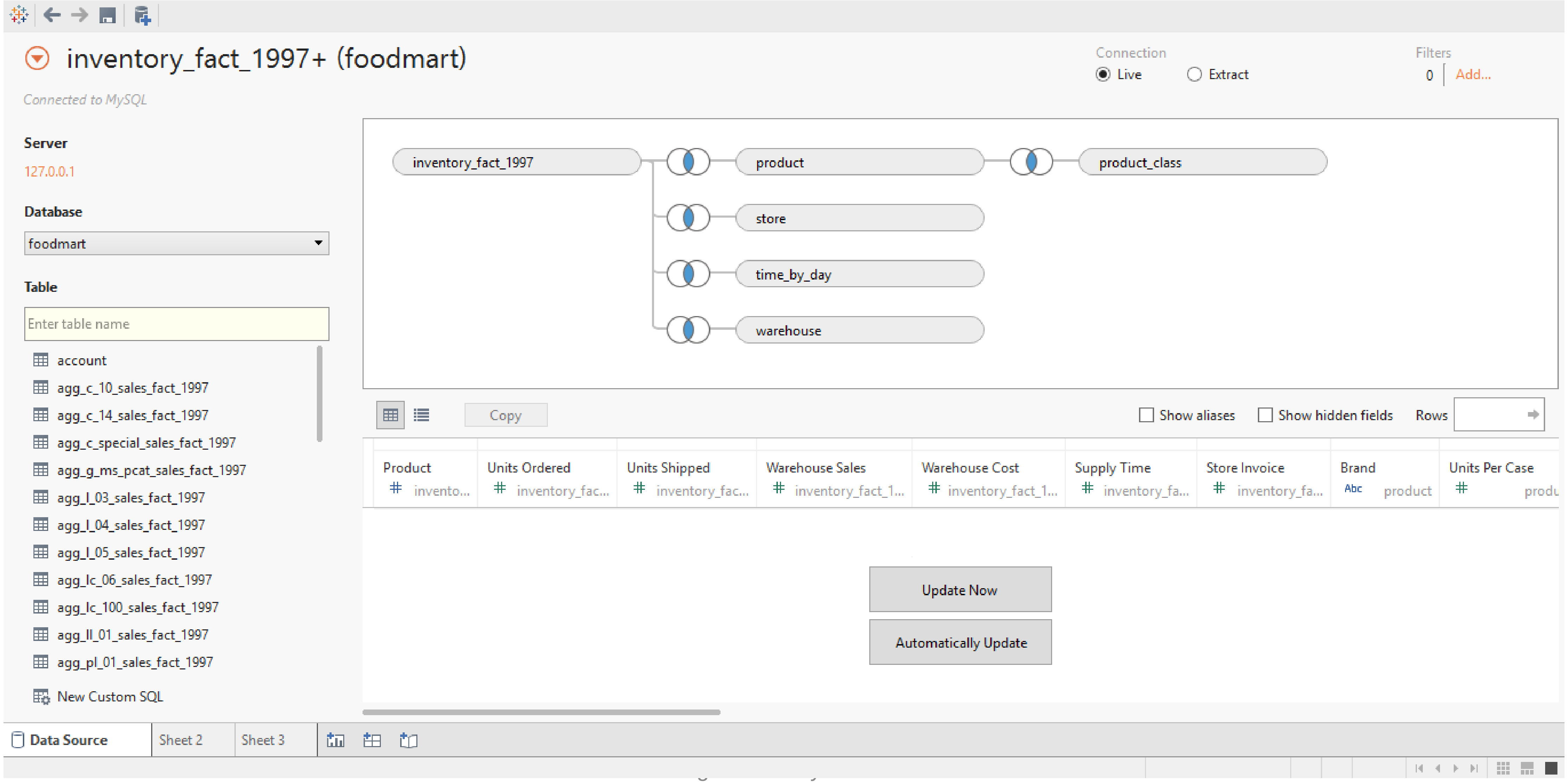
Data Source		product_class
Product Class Id	=	Product Class Id (Produc...
Add new join clause		

Tableau supports different types of join

- *Inner*: only the records that satisfy the given condition are returned
- *Left* (*Right*): corresponds to an inner join where there are also the tuples belonging to the left (right) table that do not satisfy the given condition
- *Full Outer*: corresponds to the union between the results of left and right joins

The join condition can also be specified (*theta*-join)

Example – MySQL



inventory_fact_1997+ (foodmart)

Connected to MySQL

Connection

● Live ○ Extract

Filters

0 | Add...

Server

127.0.0.1

Database

foodmart

Table

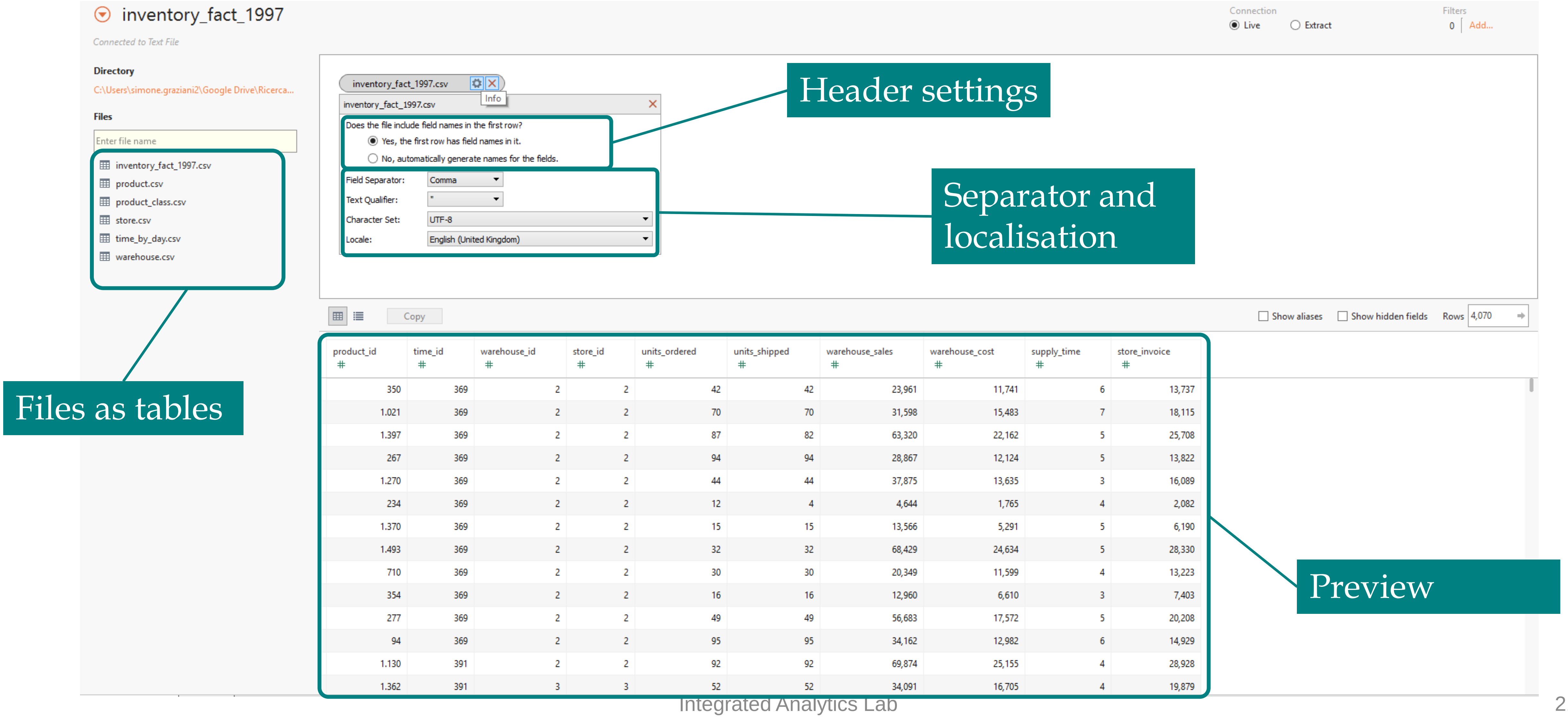
Enter table name

- account
- agg_c_10_sales_fact_1997
- agg_c_14_sales_fact_1997
- agg_c_special_sales_fact_1997
- agg_g_ms_pcat_sales_fact_1997
- agg_l_03_sales_fact_1997
- agg_l_04_sales_fact_1997
- agg_l_05_sales_fact_1997
- agg_lc_06_sales_fact_1997
- agg_lc_100_sales_fact_1997
- agg_ll_01_sales_fact_1997
- agg_pl_01_sales_fact_1997
- New Custom SQL

□ Show aliases □ Show hidden fields Rows

Product	Units Ordered	Units Shipped	Warehouse Sales	Warehouse Cost	Supply Time	Store Invoice	Brand	Units Per Case
# invento...	# inventory_fac...	# inventory_fac...	# inventory_fact_1...	# inventory_fact_1...	# inventory_fa...	# inventory_fa...	Abc product	# produ

Example – Text File (CSV)



inventory_fact_1997

Connected to Text File

Connection: ● Live ○ Extract

Filters: 0 Add...

Directory

C:\Users\simone.graziani2\Google Drive\Ricerca...

Files

Enter file name

- inventory_fact_1997.csv
- product.csv
- product_class.csv
- store.csv
- time_by_day.csv
- warehouse.csv

inventory_fact_1997.csv

Does the file include field names in the first row?

● Yes, the first row has field names in it.

○ No, automatically generate names for the fields.

Field Separator: Comma

Text Qualifier: "

Character Set: UTF-8

Locale: English (United Kingdom)

□ Show aliases □ Show hidden fields Rows 4,070

product_id	time_id	warehouse_id	store_id	units_ordered	units_shipped	warehouse_sales	warehouse_cost	supply_time	store_invoice
#	#	#	#	#	#	#	#	#	#
350	369	2	2	42	42	23,961	11,741	6	13,737
1.021	369	2	2	70	70	31,598	15,483	7	18,115
1.397	369	2	2	87	82	63,320	22,162	5	25,708
267	369	2	2	94	94	28,867	12,124	5	13,822
1.270	369	2	2	44	44	37,875	13,635	3	16,089
234	369	2	2	12	4	4,644	1,765	4	2,082
1.370	369	2	2	15	15	13,566	5,291	5	6,190
1.493	369	2	2	32	32	68,429	24,634	5	28,330
710	369	2	2	30	30	20,349	11,599	4	13,223
354	369	2	2	16	16	12,960	6,610	3	7,403
277	369	2	2	49	49	56,683	17,572	5	20,208
94	369	2	2	95	95	34,162	12,982	6	14,929
1.130	391	2	2	92	92	69,874	25,155	4	28,928
1.362	391	3	3	52	52	34,091	16,705	4	19,879

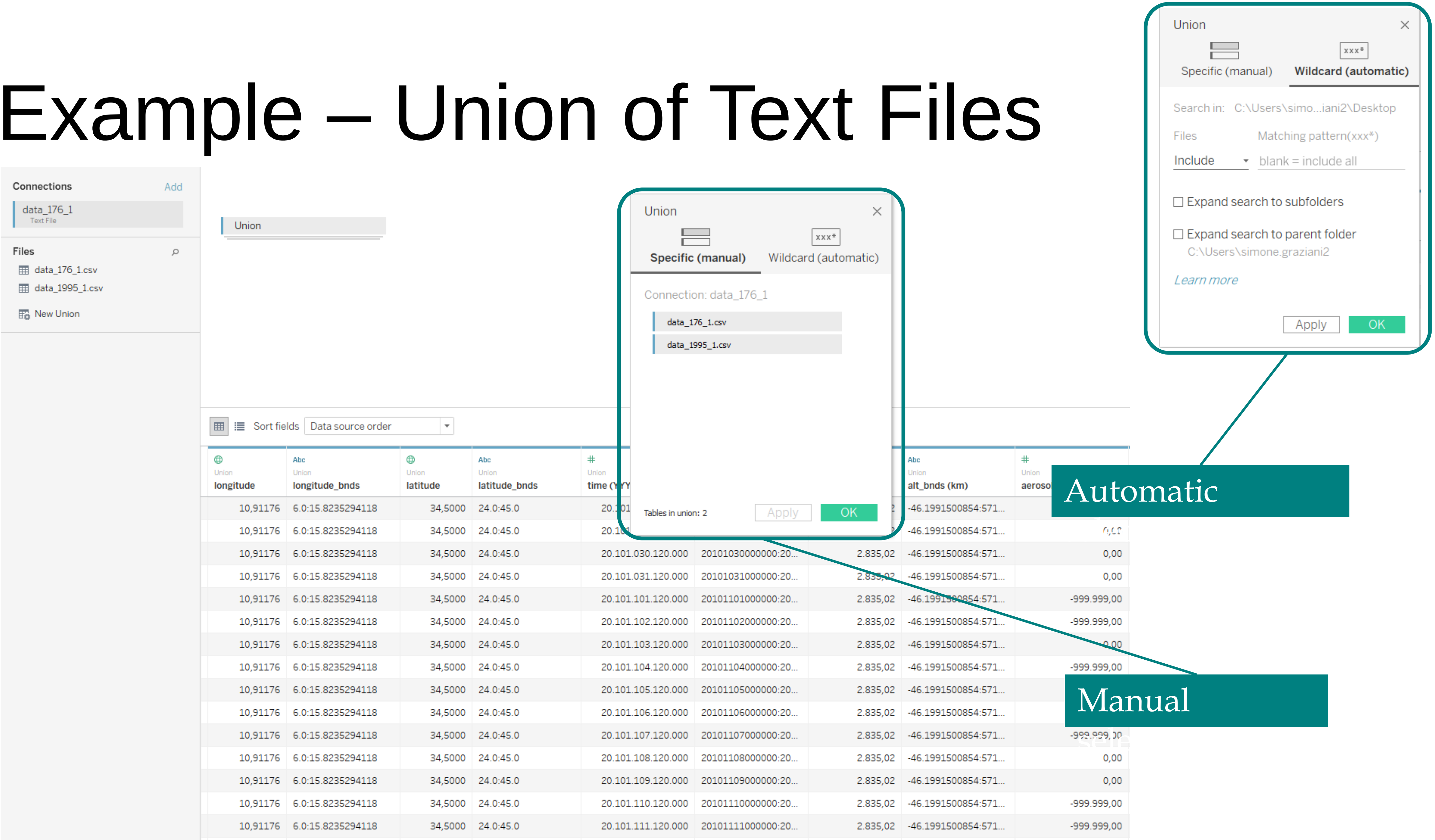
Header settings

Separator and localisation

Files as tables

Preview

Example – Union of Text Files



Connections Add

data_176_1
Text File

Files

- data_176_1.csv
- data_1995_1.csv
- New Union

Union

Sort fields: Data source order

longitude	longitude_bnds	latitude	latitude_bnds	time (YYY			alt_bnds (km)	aeroso
10,91176	6.0:15.8235294118	34,5000	24.0:45.0	20.101		2	-46.1991500854:571...	
10,91176	6.0:15.8235294118	34,5000	24.0:45.0	20.10		2	-46.1991500854:571...	0,00
10,91176	6.0:15.8235294118	34,5000	24.0:45.0	20.101.030.120.000	20101030000000:20...	2.835,02	-46.1991500854:571...	0,00
10,91176	6.0:15.8235294118	34,5000	24.0:45.0	20.101.031.120.000	20101031000000:20...	2.835,02	-46.1991500854:571...	0,00
10,91176	6.0:15.8235294118	34,5000	24.0:45.0	20.101.101.120.000	20101101000000:20...	2.835,02	-46.1991500854:571...	-999.999,00
10,91176	6.0:15.8235294118	34,5000	24.0:45.0	20.101.102.120.000	20101102000000:20...	2.835,02	-46.1991500854:571...	-999.999,00
10,91176	6.0:15.8235294118	34,5000	24.0:45.0	20.101.103.120.000	20101103000000:20...	2.835,02	-46.1991500854:571...	0,00
10,91176	6.0:15.8235294118	34,5000	24.0:45.0	20.101.104.120.000	20101104000000:20...	2.835,02	-46.1991500854:571...	-999.999,00
10,91176	6.0:15.8235294118	34,5000	24.0:45.0	20.101.105.120.000	20101105000000:20...	2.835,02	-46.1991500854:571...	
10,91176	6.0:15.8235294118	34,5000	24.0:45.0	20.101.106.120.000	20101106000000:20...	2.835,02	-46.1991500854:571...	
10,91176	6.0:15.8235294118	34,5000	24.0:45.0	20.101.107.120.000	20101107000000:20...	2.835,02	-46.1991500854:571...	-999.999,00
10,91176	6.0:15.8235294118	34,5000	24.0:45.0	20.101.108.120.000	20101108000000:20...	2.835,02	-46.1991500854:571...	0,00
10,91176	6.0:15.8235294118	34,5000	24.0:45.0	20.101.109.120.000	20101109000000:20...	2.835,02	-46.1991500854:571...	0,00
10,91176	6.0:15.8235294118	34,5000	24.0:45.0	20.101.110.120.000	20101110000000:20...	2.835,02	-46.1991500854:571...	-999.999,00
10,91176	6.0:15.8235294118	34,5000	24.0:45.0	20.101.111.120.000	20101111000000:20...	2.835,02	-46.1991500854:571...	-999.999,00

Union ✕

Specific (manual) | Wildcard (automatic)

Connection: data_176_1

- data_176_1.csv
- data_1995_1.csv

Tables in union: 2

Apply OK

Union ✕

Specific (manual) | **Wildcard (automatic)**

Search in: C:\Users\simo...iani2\Desktop

Files: Include — Matching pattern(xxx*): blank = include all

☐ Expand search to subfolders

☐ Expand search to parent folder
C:\Users\simone.graziani2

Learn more

Apply OK

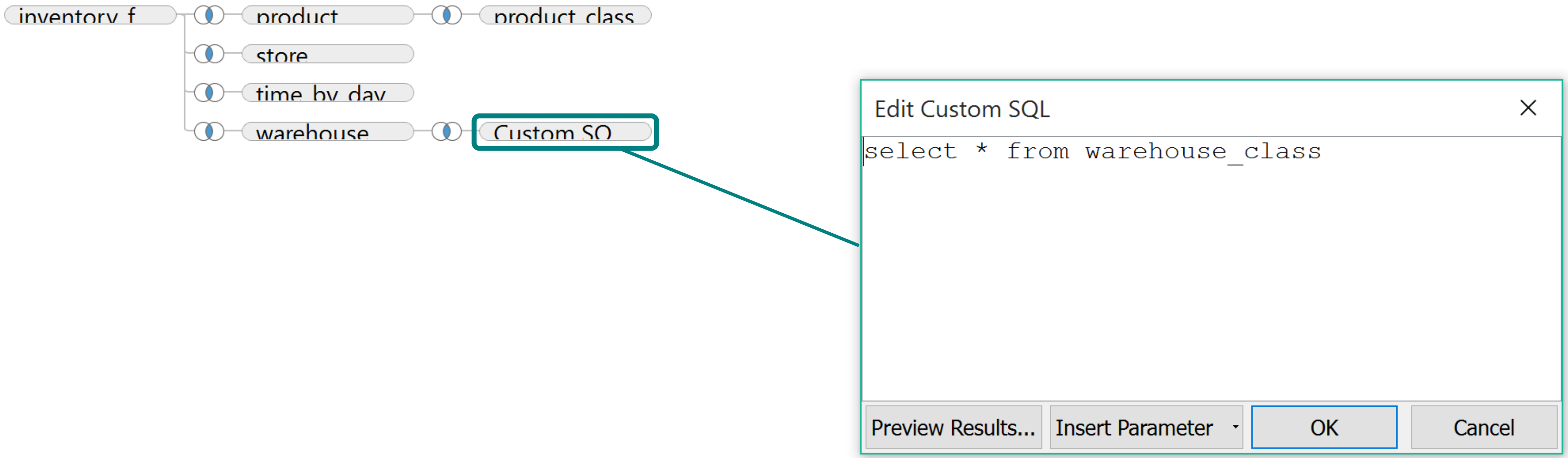
Automatic

Manual

Custom SQL

With data sources that support SQL it is also possible to define a table through a *custom query*

Tables created through custom query can also be joined with other tables

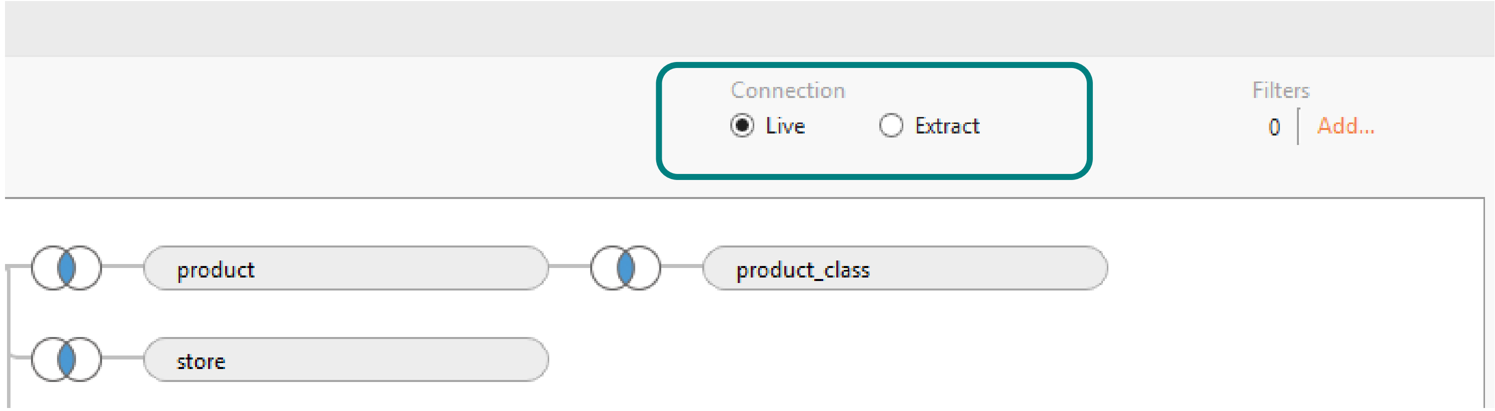


```sql
select * from warehouse_class
```

Live VS Extract

There are two different types of connection

- *Live*: each time that a visualisation is created (or edited) Tableau performs a query to the data source
- *Extract*: Tableau performs a single (big) query at the beginning to pull all the required data locally; by exploiting the (locally) stored data, there is no need to issue additional queries the data source



Connection

● Live ○ Extract

Filters

0 | Add...

Live VS Extract (2)

A live connection is advised when

- The data source offers high performances (e.g., a typical DW)
- Data freshness is a must
- The amount of data to be analysed is too high to be stored in local

An extract connection is instead advised when

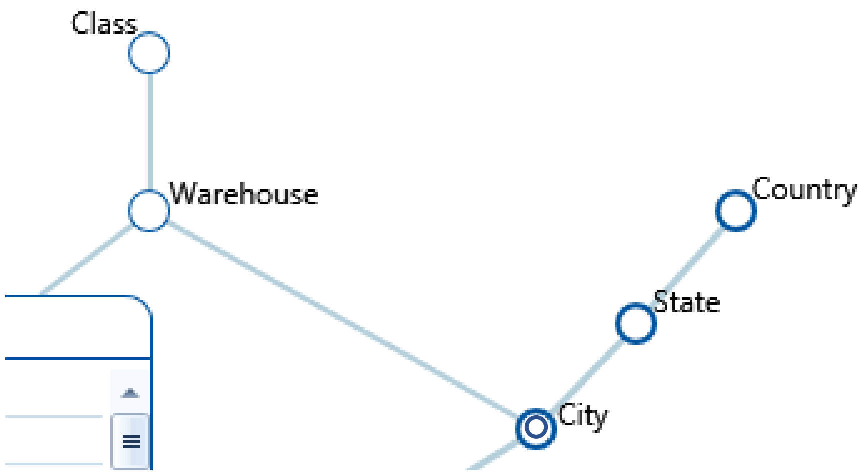
- The data source does not offer fast responses to analytical queries
- The data source must not be overloaded with analytical queries (e.g., an operational data store)
- Off-line data is needed (e.g., performing analyses without a connection to the data source)

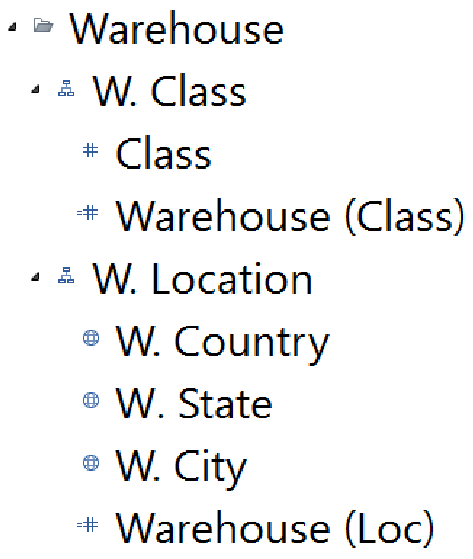
From the DFM to Tableau

Tableau offers limited support to directly represent DFM structures. Specifically, Tableau cannot directly represent

- Hierarchies with branches, shared hierarchies and convergences
  - *Workaround*: linearise with attributes duplication
- Descriptive attributes
  - *Workaround*: use a standard field placed outside of hierarchies
- Multiple edges and cross-dimensional attributes
  - *Workaround*: calculated fields (complex and ad-hoc solutions that are often difficult to use)
- Aggregation constraints
  - It is not possible to force the user to only use sensible aggregation operations (however, proper documentation is usually enough to guide the user)

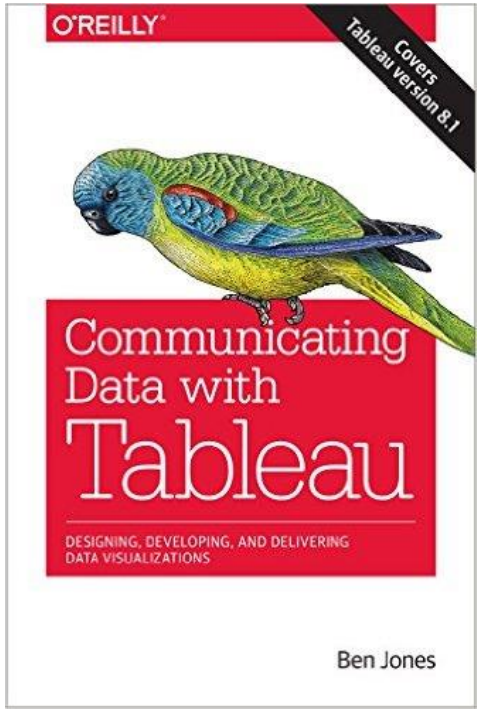
From the DFM to Tableau – Example

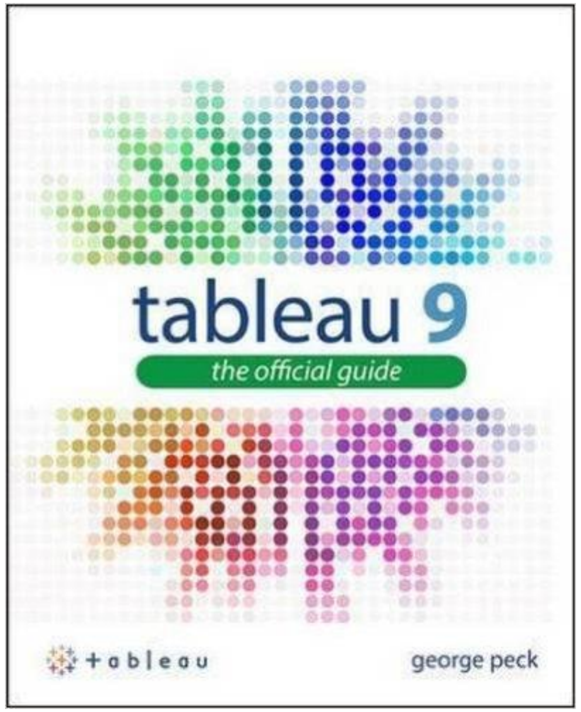




- Warehouse
  - W. Class
    - Class
    - Warehouse (Class)
  - W. Location
    - W. Country
    - W. State
    - W. City
    - Warehouse (Loc)

References


O'REILLY
Covers Tableau version 8.1
Communicating Data with Tableau
DESIGNING, DEVELOPING, AND DELIVERING DATA VISUALIZATIONS
Ben Jones



Tutorial: http://www.tableau.com/learn/training

Knowledge Base: http://kb.tableau.com/