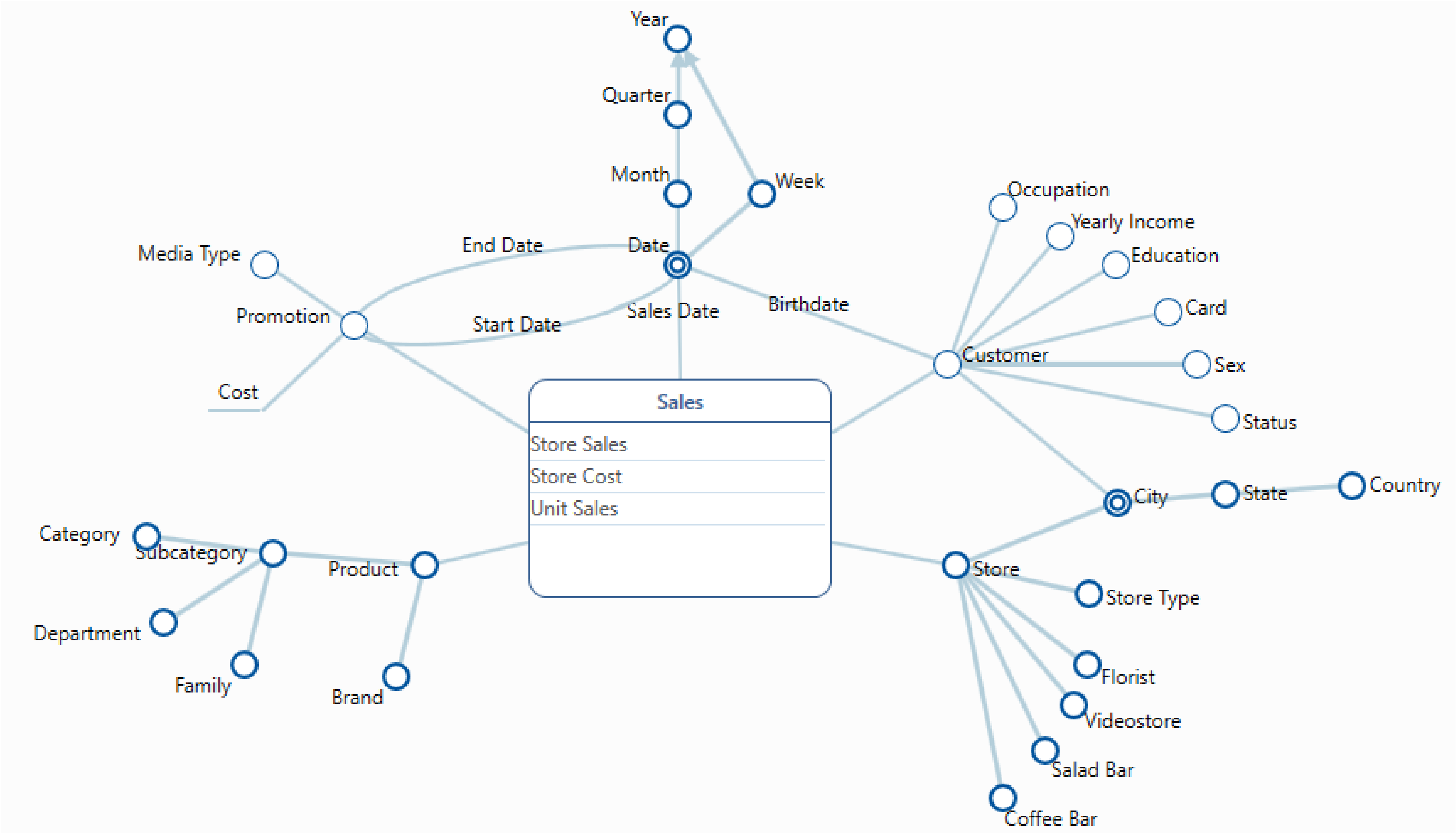
OLAP with Tableau

Enrico Gallinucci <enrico.gallinucci@unibo.it>

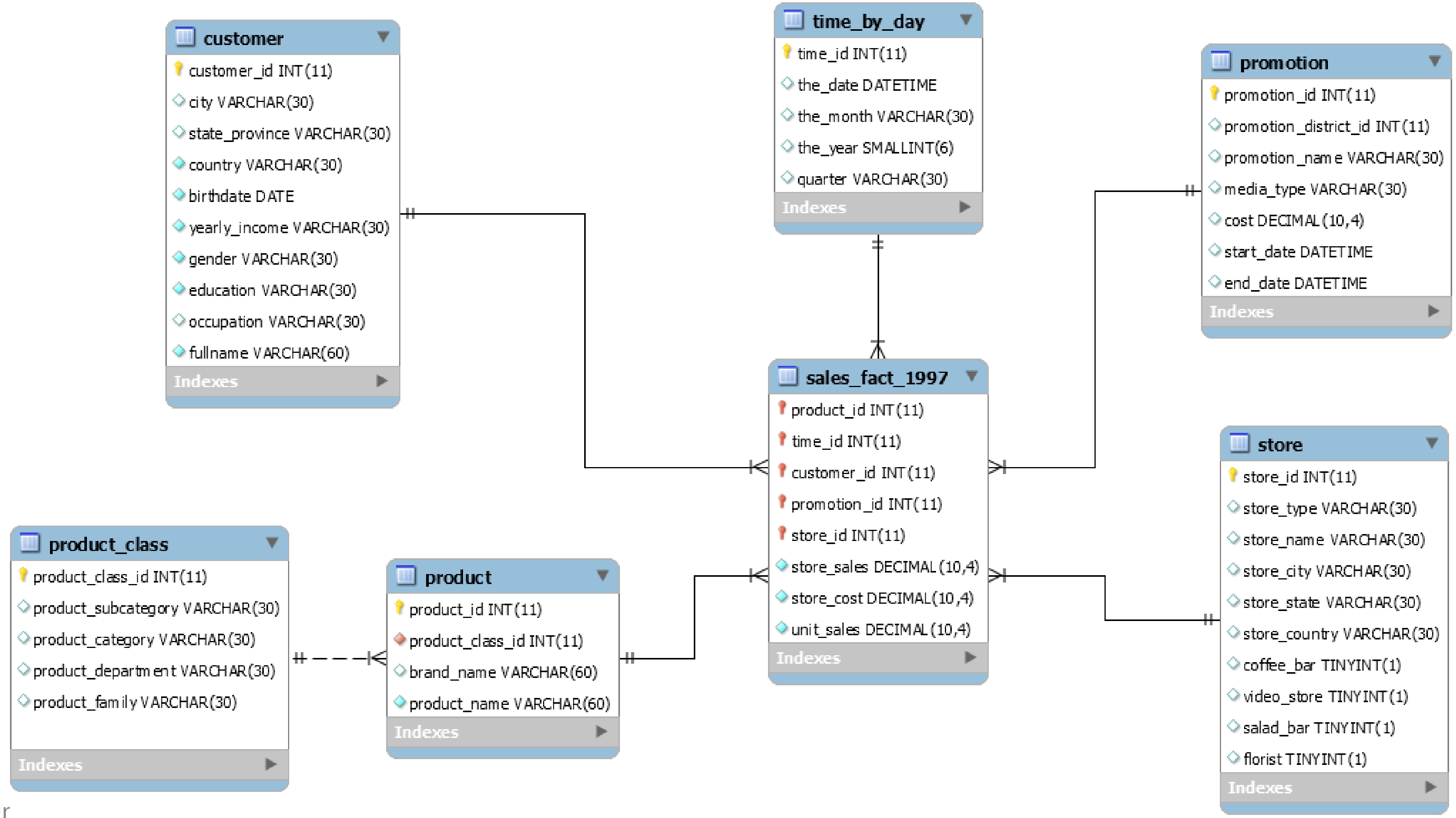
BBS
MULTIPLYING OPPORTUNITIES

BOLOGNA BUSINESS SCHOOL
UNIVERSITÀ DI BOLOGNA
A.D. 1088

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

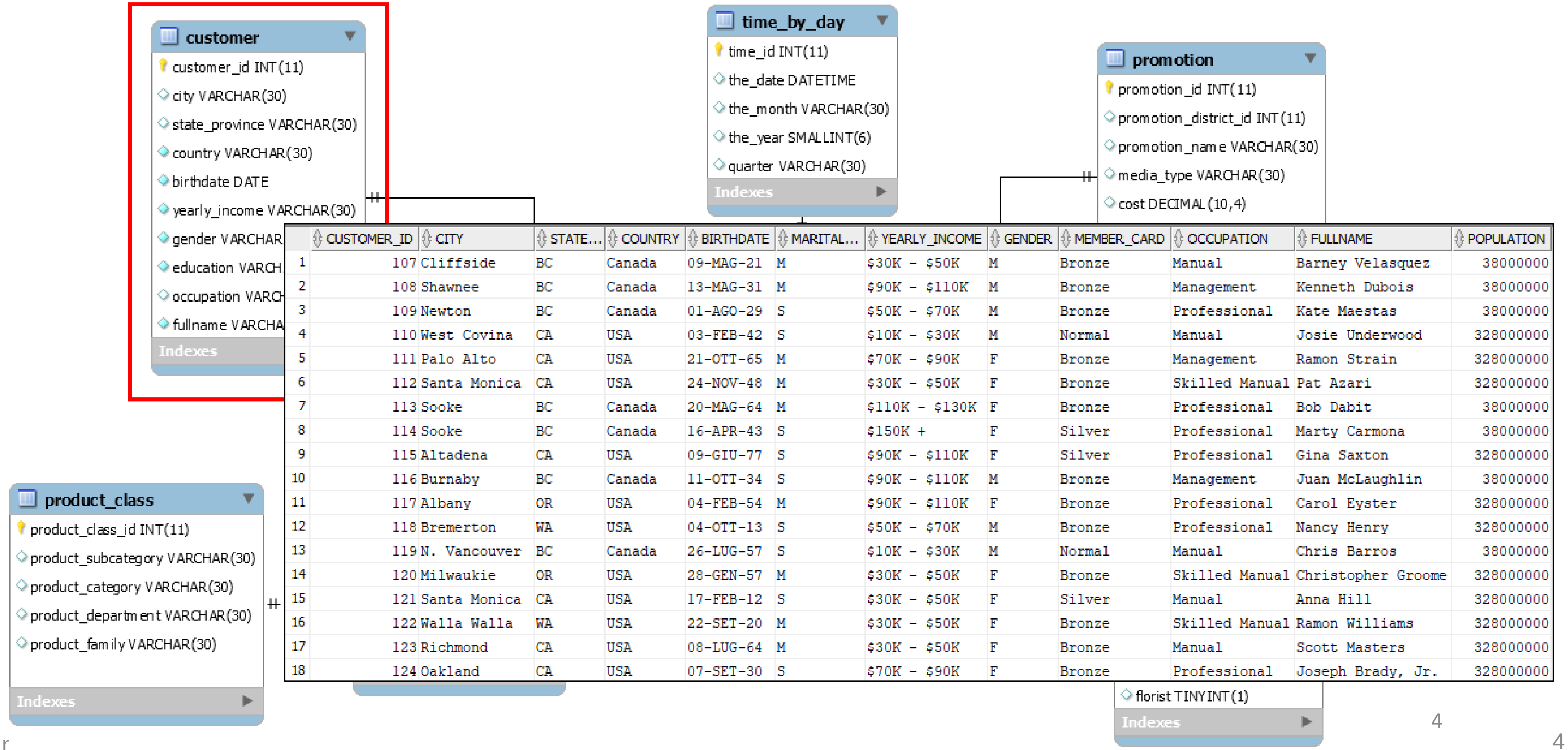
product_class
- product_class_id INT(11)
- product_subcategory VARCHAR(30)
- product_category VARCHAR(30)
- product_department VARCHAR(30)
- product_family VARCHAR(30)

Indexes

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

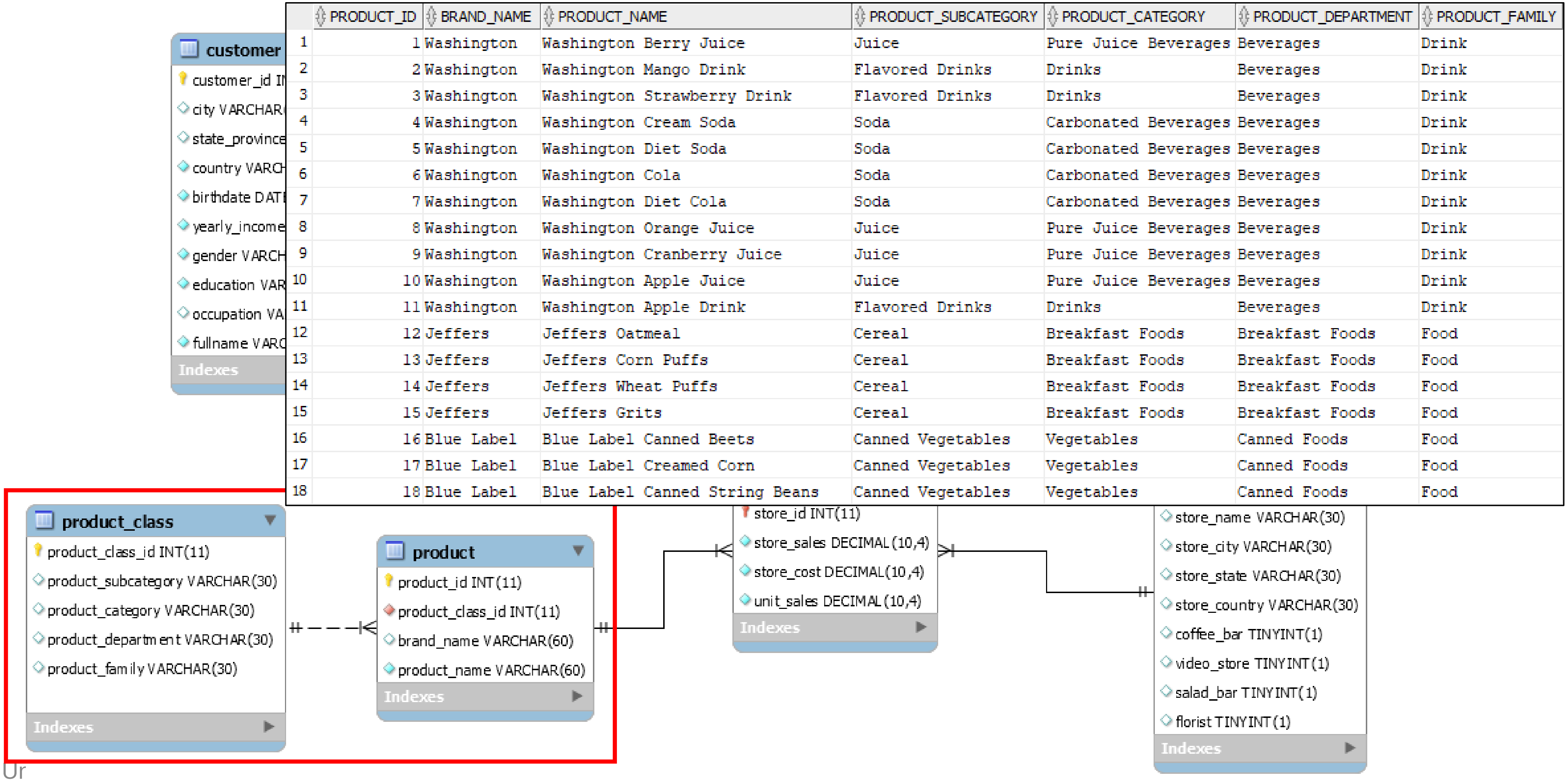
Indexes

Database Structure – Foodmart (Sales)



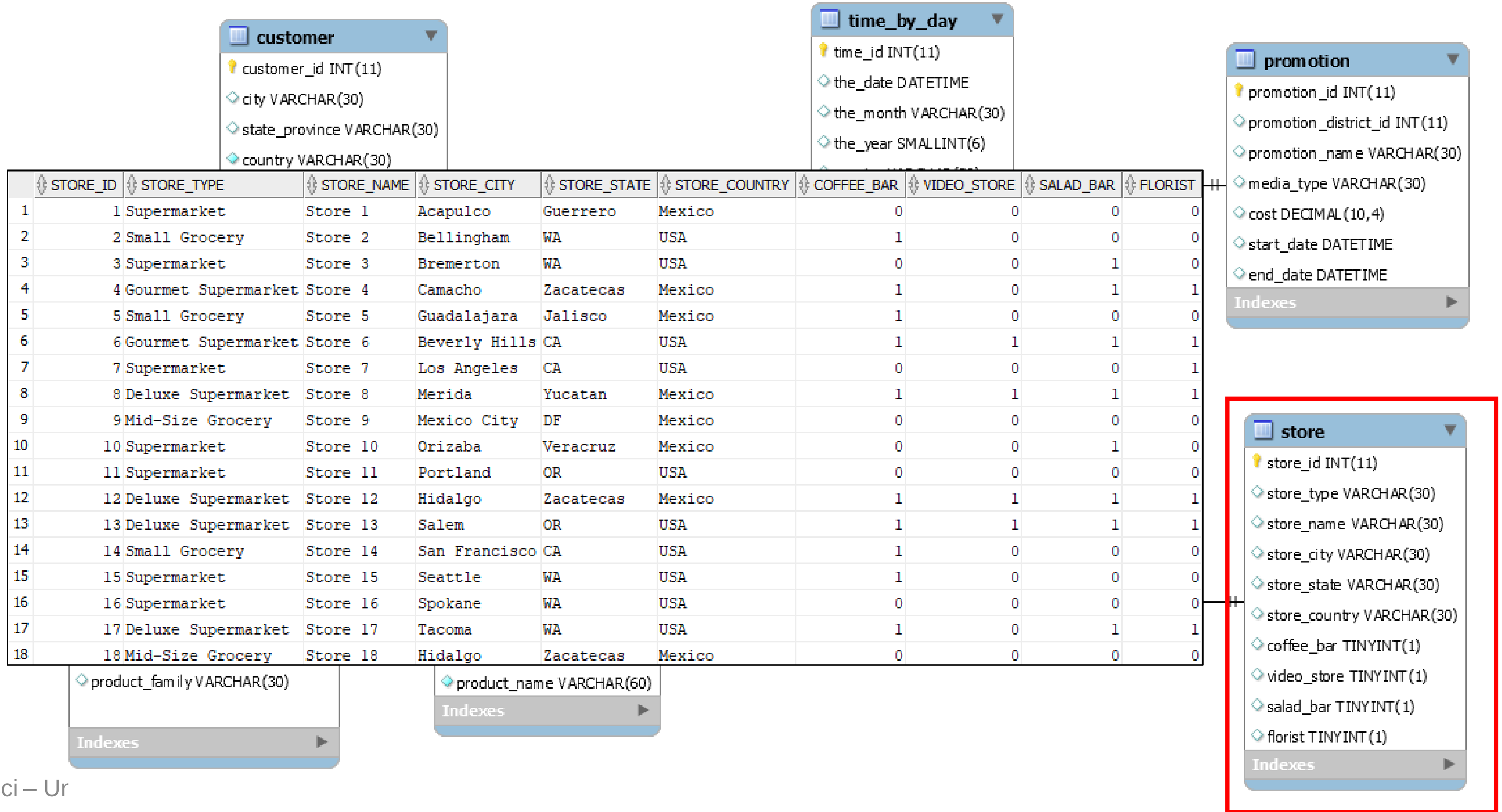
	CUSTOMER_ID	CITY	STATE...	COUNTRY	BIRTHDATE	MARITAL...	YEARLY_INCOME	GENDER	MEMBER_CARD	OCCUPATION	FULLNAME	POPULATION
1	107	Cliffside	BC	Canada	09-MAG-21	M	$30K - $50K	M	Bronze	Manual	Barney Velasquez	38000000
2	108	Shawnee	BC	Canada	13-MAG-31	M	$90K - $110K	M	Bronze	Management	Kenneth Dubois	38000000
3	109	Newton	BC	Canada	01-AGO-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-OTT-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-MAG-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-GIU-77	S	$90K - $110K	F	Silver	Professional	Gina Saxton	328000000
10	116	Burnaby	BC	Canada	11-OTT-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-OTT-13	S	$50K - $70K	M	Bronze	Professional	Nancy Henry	328000000
13	119	N. Vancouver	BC	Canada	26-LUG-57	S	$10K - $30K	M	Normal	Manual	Chris Barros	38000000
14	120	Milwaukie	OR	USA	28-GEN-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-SET-20	M	$30K - $50K	F	Bronze	Skilled Manual	Ramon Williams	328000000
17	123	Richmond	CA	USA	08-LUG-64	M	$30K - $50K	F	Bronze	Manual	Scott Masters	328000000
18	124	Oakland	CA	USA	07-SET-30	S	$70K - $90K	F	Bronze	Professional	Joseph Brady, Jr.	328000000

Database Structure – Foodmart (Sales)



	PRODUCT_ID	BRAND_NAME	PRODUCT_NAME	PRODUCT_SUBCATEGORY	PRODUCT_CATEGORY	PRODUCT_DEPARTMENT	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
18	18	Blue Label	Blue Label Canned String Beans	Canned Vegetables	Vegetables	Canned Foods	Food

Database Structure – Foodmart (Sales)



customer

- customer_id INT(11)
- city VARCHAR(30)
- state_province VARCHAR(30)
- country VARCHAR(30)

time_by_day

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

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

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_family VARCHAR(30)

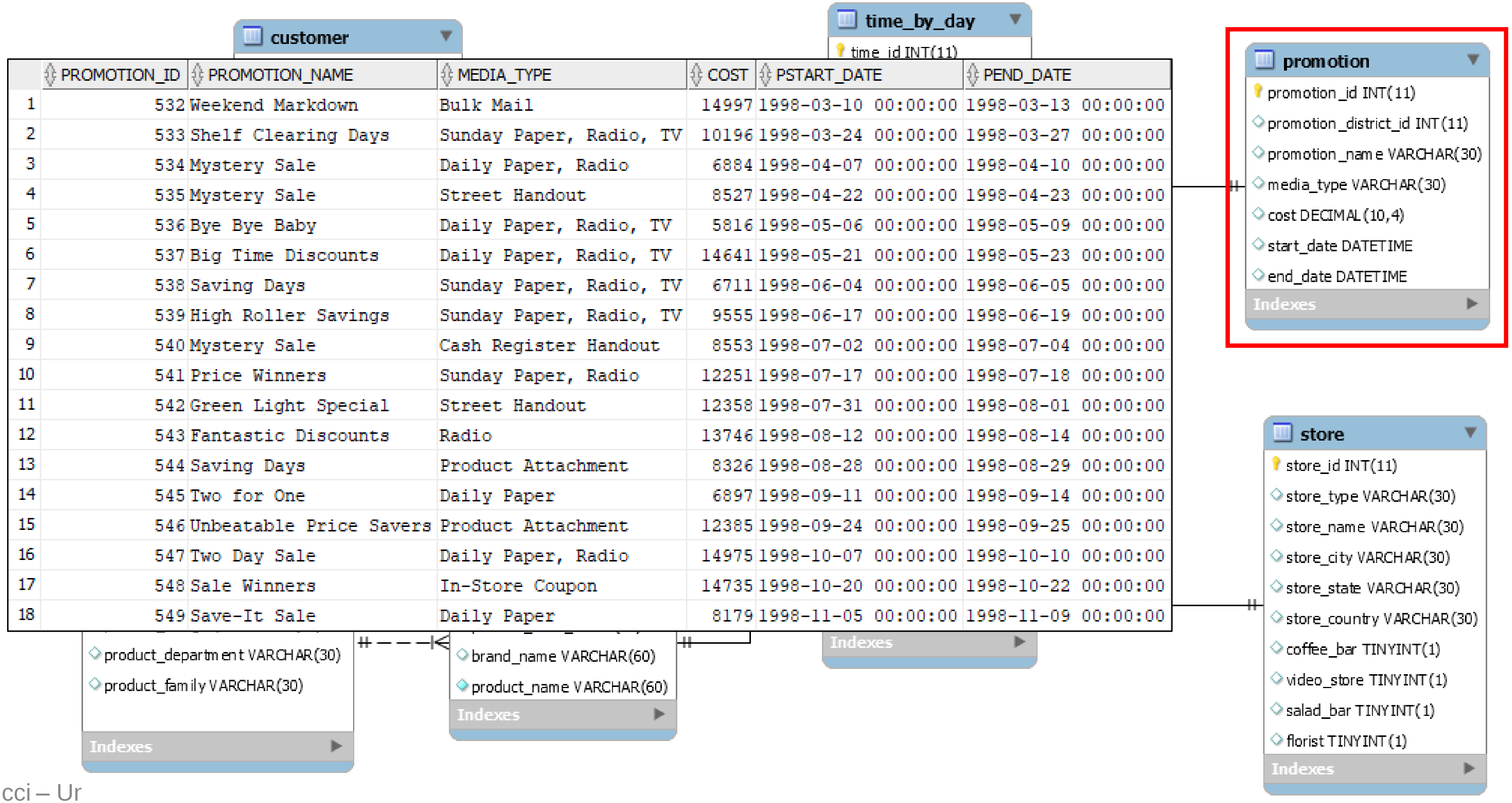
Indexes

- product_name VARCHAR(60)

Indexes

	STORE_ID	STORE_TYPE	STORE_NAME	STORE_CITY	STORE_STATE	STORE_COUNTRY	COFFEE_BAR	VIDEO_STORE	SALAD_BAR	FLORIST
1	1	Supermarket	Store 1	Acapulco	Guerrero	Mexico	0	0	0	0
2	2	Small Grocery	Store 2	Bellingham	WA	USA	1	0	0	0
3	3	Supermarket	Store 3	Bremerton	WA	USA	0	0	1	0
4	4	Gourmet Supermarket	Store 4	Camacho	Zacatecas	Mexico	1	0	1	1
5	5	Small Grocery	Store 5	Guadalajara	Jalisco	Mexico	1	0	0	0
6	6	Gourmet Supermarket	Store 6	Beverly Hills	CA	USA	1	1	1	1
7	7	Supermarket	Store 7	Los Angeles	CA	USA	0	0	0	1
8	8	Deluxe Supermarket	Store 8	Merida	Yucatan	Mexico	1	1	1	1
9	9	Mid-Size Grocery	Store 9	Mexico City	DF	Mexico	0	0	0	0
10	10	Supermarket	Store 10	Orizaba	Veracruz	Mexico	0	0	1	0
11	11	Supermarket	Store 11	Portland	OR	USA	0	0	0	0
12	12	Deluxe Supermarket	Store 12	Hidalgo	Zacatecas	Mexico	1	1	1	1
13	13	Deluxe Supermarket	Store 13	Salem	OR	USA	1	1	1	1
14	14	Small Grocery	Store 14	San Francisco	CA	USA	1	0	0	0
15	15	Supermarket	Store 15	Seattle	WA	USA	1	0	0	0
16	16	Supermarket	Store 16	Spokane	WA	USA	0	0	0	0
17	17	Deluxe Supermarket	Store 17	Tacoma	WA	USA	1	0	1	1
18	18	Mid-Size Grocery	Store 18	Hidalgo	Zacatecas	Mexico	0	0	0	0

Database Structure – Foodmart (Sales)

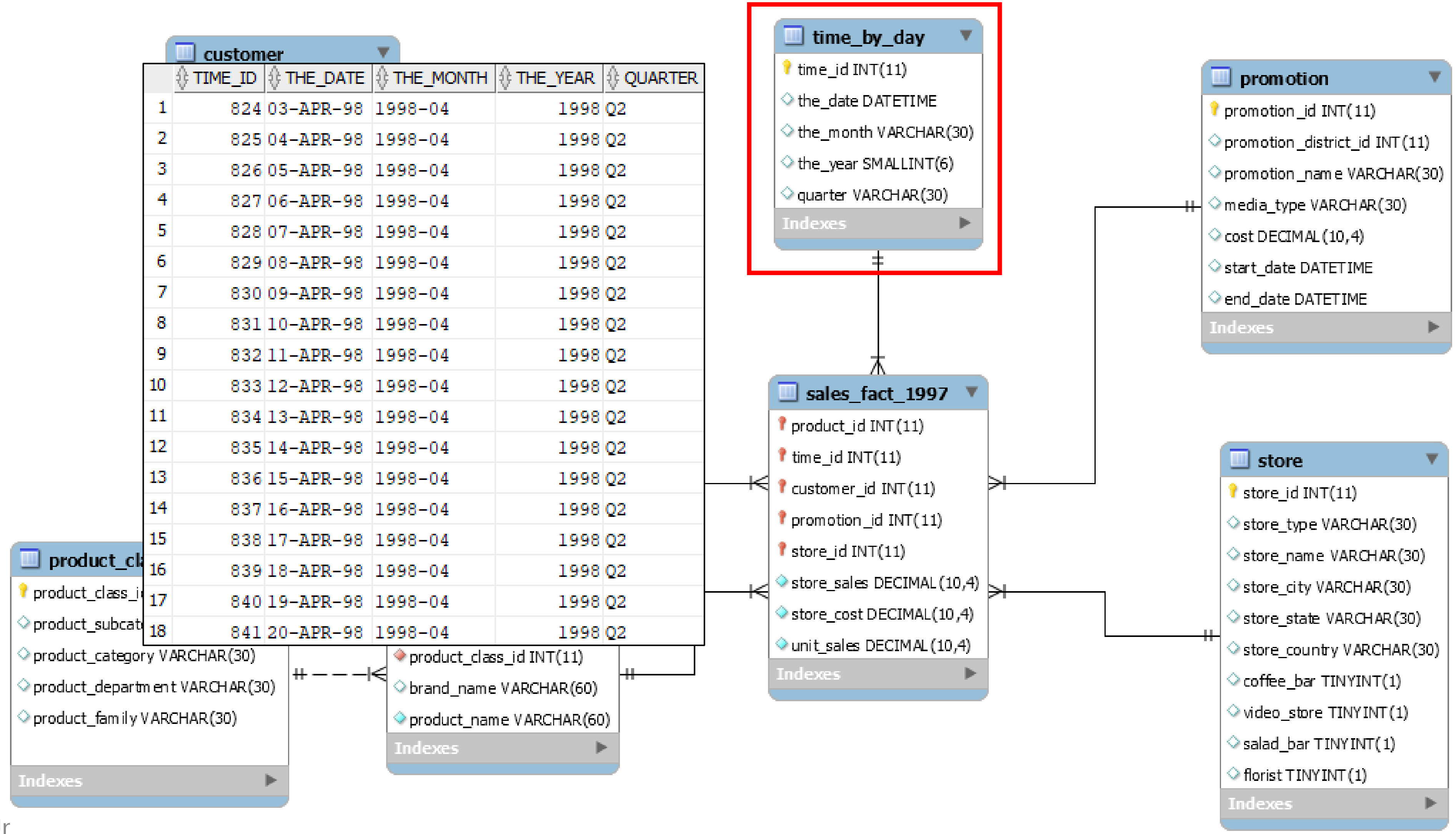


	PROMOTION_ID	PROMOTION_NAME	MEDIA_TYPE	COST	PSTART_DATE	PEND_DATE
1	532	Weekend Markdown	Bulk Mail	14997	1998-03-10 00:00:00	1998-03-13 00:00:00
2	533	Shelf Clearing Days	Sunday Paper, Radio, TV	10196	1998-03-24 00:00:00	1998-03-27 00:00:00
3	534	Mystery Sale	Daily Paper, Radio	6884	1998-04-07 00:00:00	1998-04-10 00:00:00
4	535	Mystery Sale	Street Handout	8527	1998-04-22 00:00:00	1998-04-23 00:00:00
5	536	Bye Bye Baby	Daily Paper, Radio, TV	5816	1998-05-06 00:00:00	1998-05-09 00:00:00
6	537	Big Time Discounts	Daily Paper, Radio, TV	14641	1998-05-21 00:00:00	1998-05-23 00:00:00
7	538	Saving Days	Sunday Paper, Radio, TV	6711	1998-06-04 00:00:00	1998-06-05 00:00:00
8	539	High Roller Savings	Sunday Paper, Radio, TV	9555	1998-06-17 00:00:00	1998-06-19 00:00:00
9	540	Mystery Sale	Cash Register Handout	8553	1998-07-02 00:00:00	1998-07-04 00:00:00
10	541	Price Winners	Sunday Paper, Radio	12251	1998-07-17 00:00:00	1998-07-18 00:00:00
11	542	Green Light Special	Street Handout	12358	1998-07-31 00:00:00	1998-08-01 00:00:00
12	543	Fantastic Discounts	Radio	13746	1998-08-12 00:00:00	1998-08-14 00:00:00
13	544	Saving Days	Product Attachment	8326	1998-08-28 00:00:00	1998-08-29 00:00:00
14	545	Two for One	Daily Paper	6897	1998-09-11 00:00:00	1998-09-14 00:00:00
15	546	Unbeatable Price Savers	Product Attachment	12385	1998-09-24 00:00:00	1998-09-25 00:00:00
16	547	Two Day Sale	Daily Paper, Radio	14975	1998-10-07 00:00:00	1998-10-10 00:00:00
17	548	Sale Winners	In-Store Coupon	14735	1998-10-20 00:00:00	1998-10-22 00:00:00
18	549	Save-It Sale	Daily Paper	8179	1998-11-05 00:00:00	1998-11-09 00:00:00

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

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

Database Structure – Foodmart (Sales)



	TIME_ID	THE_DATE	THE_MONTH	THE_YEAR	QUARTER
1	824	03-APR-98	1998-04	1998	Q2
2	825	04-APR-98	1998-04	1998	Q2
3	826	05-APR-98	1998-04	1998	Q2
4	827	06-APR-98	1998-04	1998	Q2
5	828	07-APR-98	1998-04	1998	Q2
6	829	08-APR-98	1998-04	1998	Q2
7	830	09-APR-98	1998-04	1998	Q2
8	831	10-APR-98	1998-04	1998	Q2
9	832	11-APR-98	1998-04	1998	Q2
10	833	12-APR-98	1998-04	1998	Q2
11	834	13-APR-98	1998-04	1998	Q2
12	835	14-APR-98	1998-04	1998	Q2
13	836	15-APR-98	1998-04	1998	Q2
14	837	16-APR-98	1998-04	1998	Q2
15	838	17-APR-98	1998-04	1998	Q2
16	839	18-APR-98	1998-04	1998	Q2
17	840	19-APR-98	1998-04	1998	Q2
18	841	20-APR-98	1998-04	1998	Q2

customer

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

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

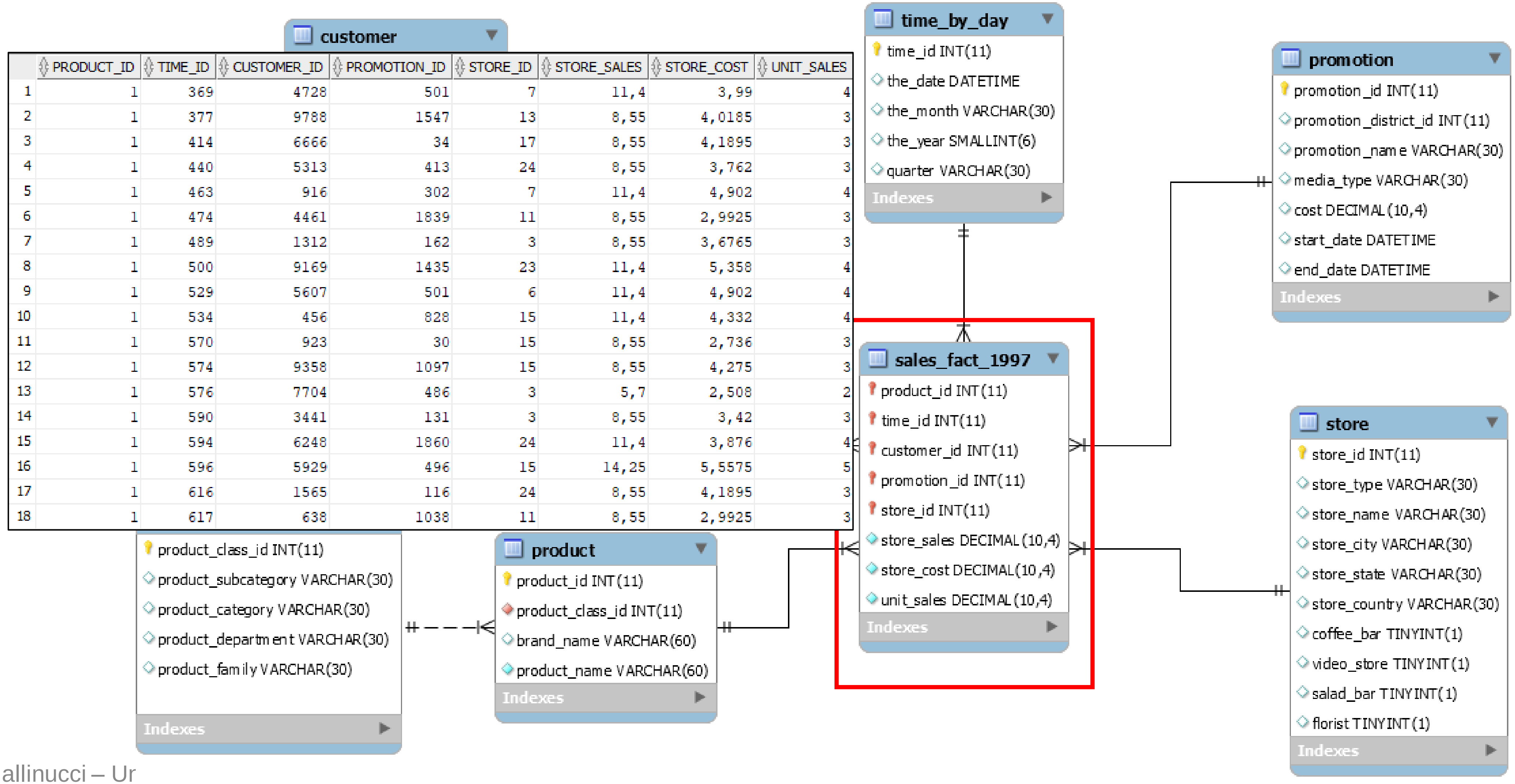
sales_fact_1997
- product_id INT(11)
- time_id INT(11)
- customer_id INT(11)
- promotion_id INT(11)
- store_id INT(11)
- store_sales DECIMAL(10,4)
- store_cost DECIMAL(10,4)
- unit_sales DECIMAL(10,4)

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)

product_cl
- product_class_i
- product_subcat
- product_category VARCHAR(30)
- product_department VARCHAR(30)
- product_family VARCHAR(30)

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

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
18	1	617	638	1038	11	8,55	2,9925	3

customer

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

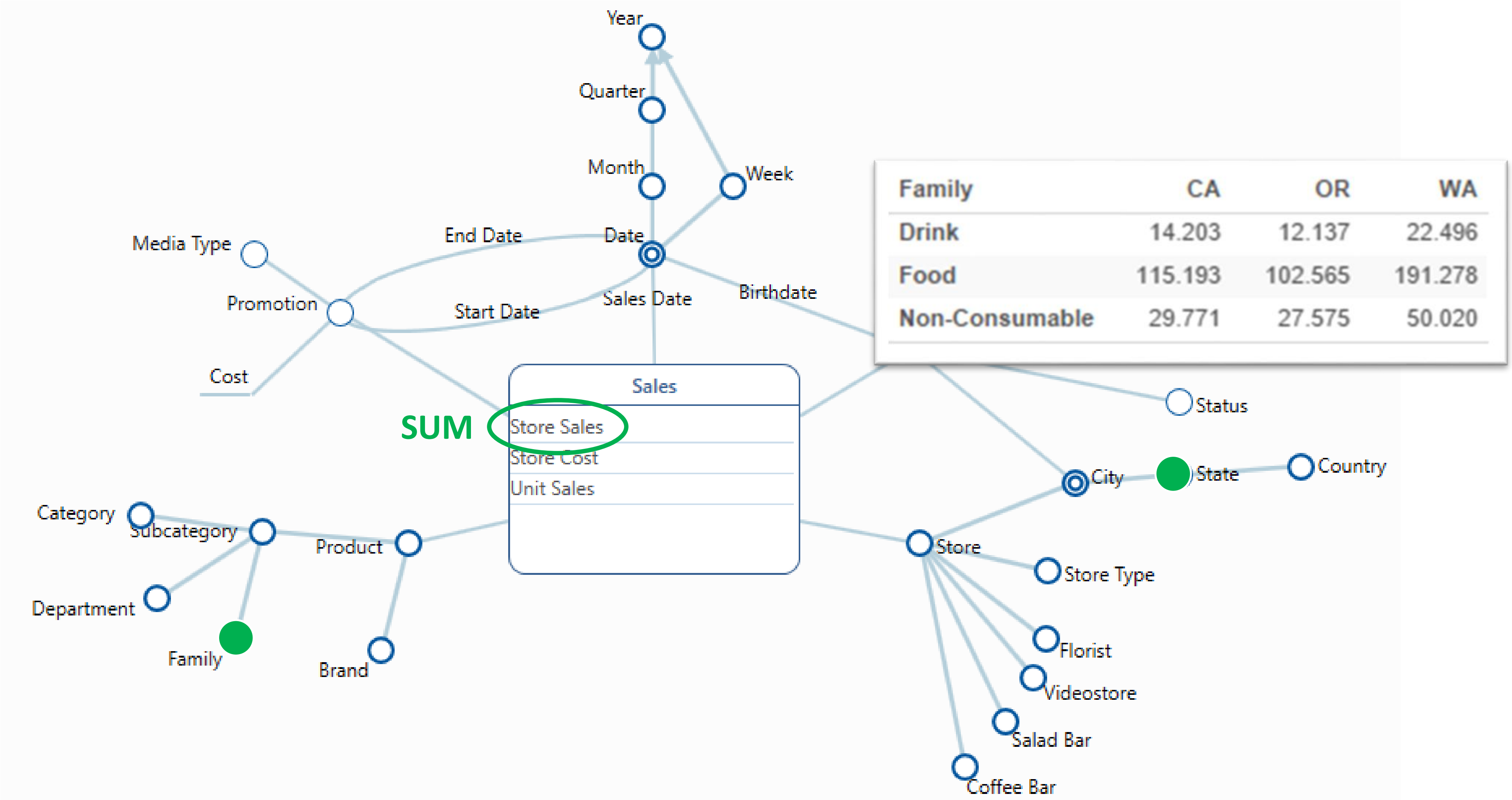
sales_fact_1997
- product_id INT(11)
- time_id INT(11)
- customer_id INT(11)
- promotion_id INT(11)
- store_id INT(11)
- store_sales DECIMAL(10,4)
- store_cost DECIMAL(10,4)
- unit_sales DECIMAL(10,4)
- Indexes

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_id INT(11)
- product_subcategory VARCHAR(30)
- product_category VARCHAR(30)
- product_department VARCHAR(30)
- product_family VARCHAR(30)
- Indexes

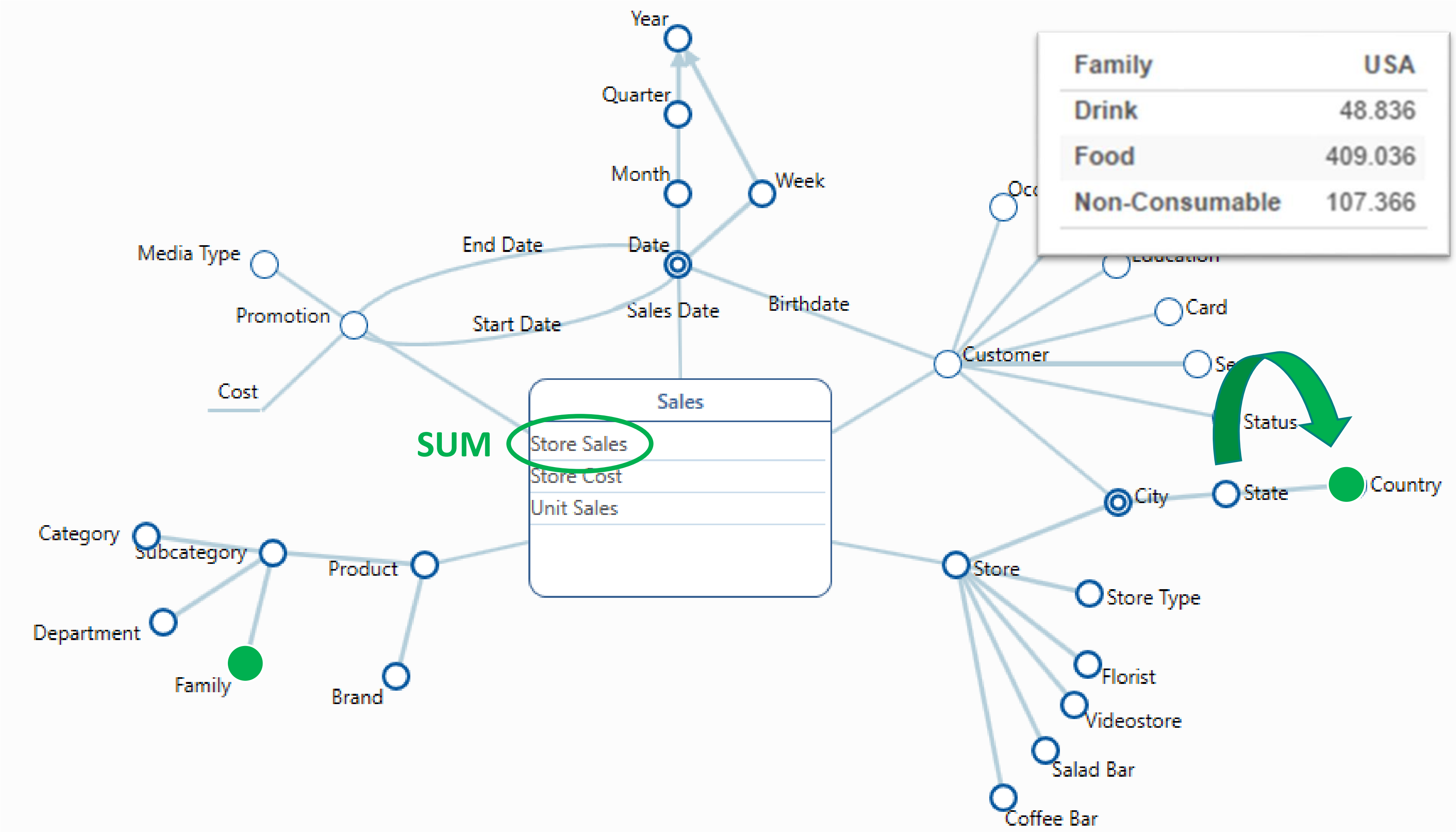
product
- product_id INT(11)
- product_class_id INT(11)
- brand_name VARCHAR(60)
- product_name VARCHAR(60)
- Indexes

OLAP query



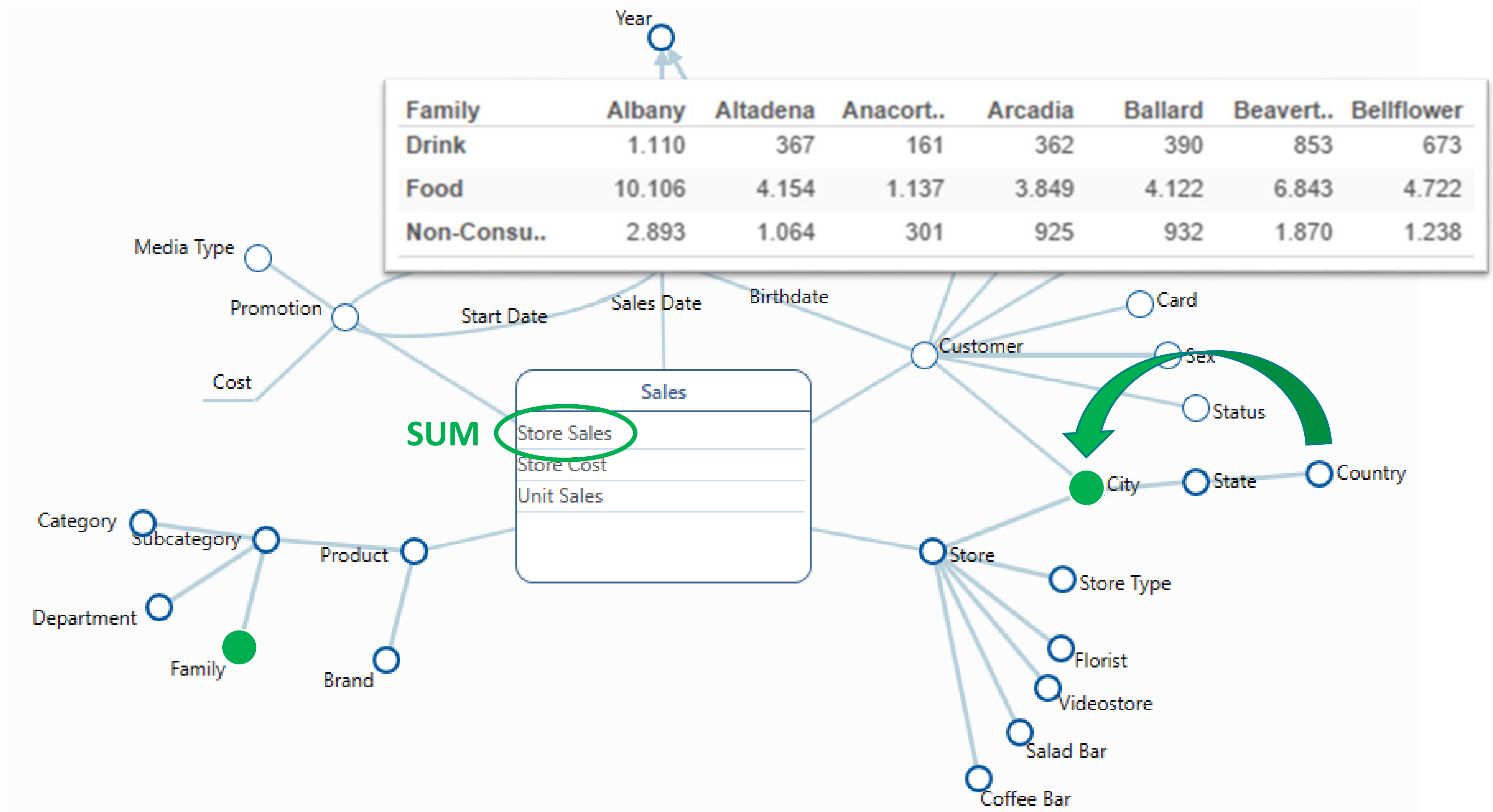
Family	CA	OR	WA
Drink	14.203	12.137	22.496
Food	115.193	102.565	191.278
Non-Consumable	29.771	27.575	50.020

OLAP Operators: Roll-Up



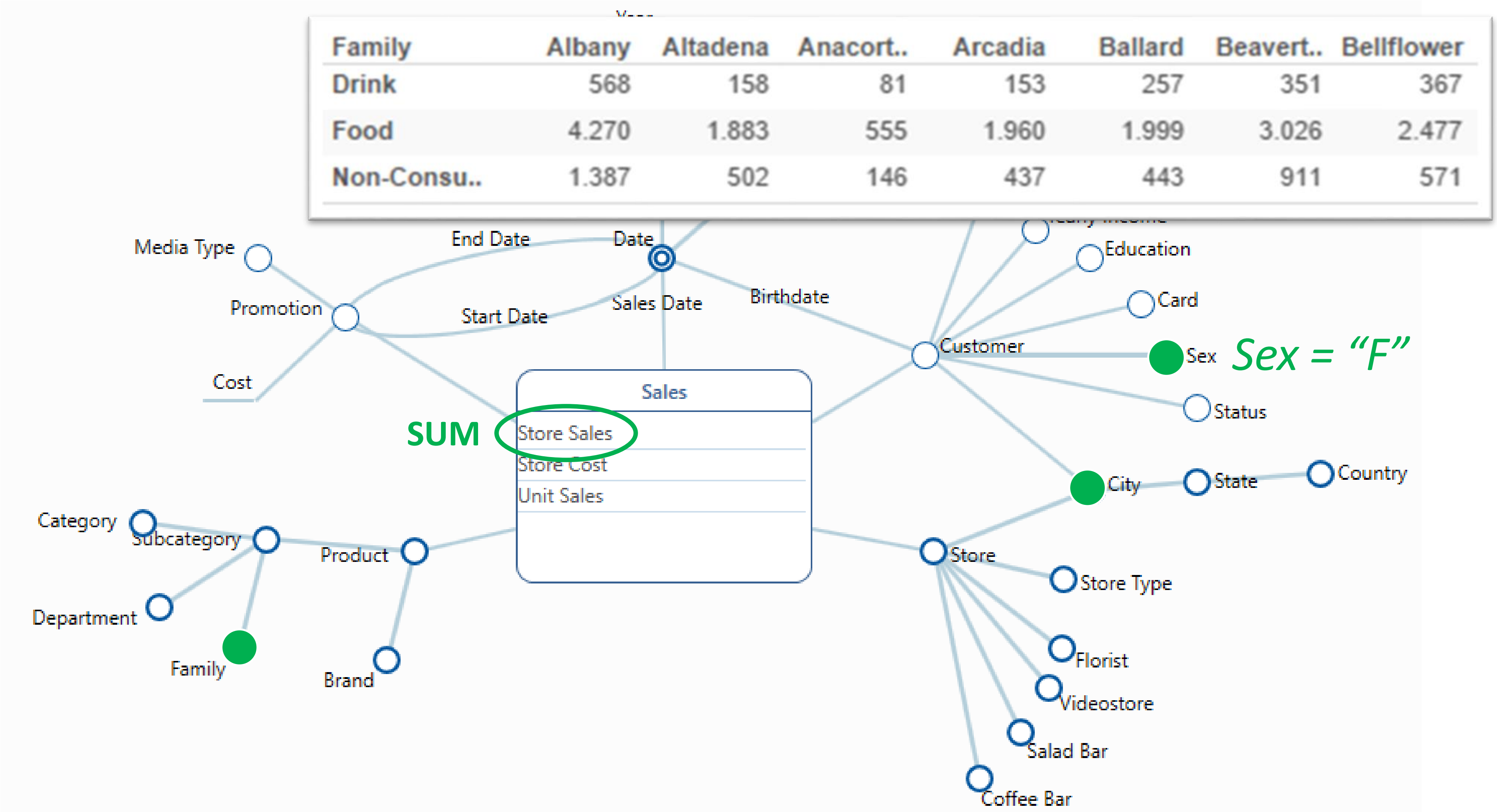
Family	USA
Drink	48.836
Food	409.036
Non-Consumable	107.366

OLAP Operators: Drill-down



Family	Albany	Altadena	Anacort..	Arcadia	Ballard	Beavert..	Bellflower
Drink	1.110	367	161	362	390	853	673
Food	10.106	4.154	1.137	3.849	4.122	6.843	4.722
Non-Consu..	2.893	1.064	301	925	932	1.870	1.238

OLAP Operators: Slice & Dice

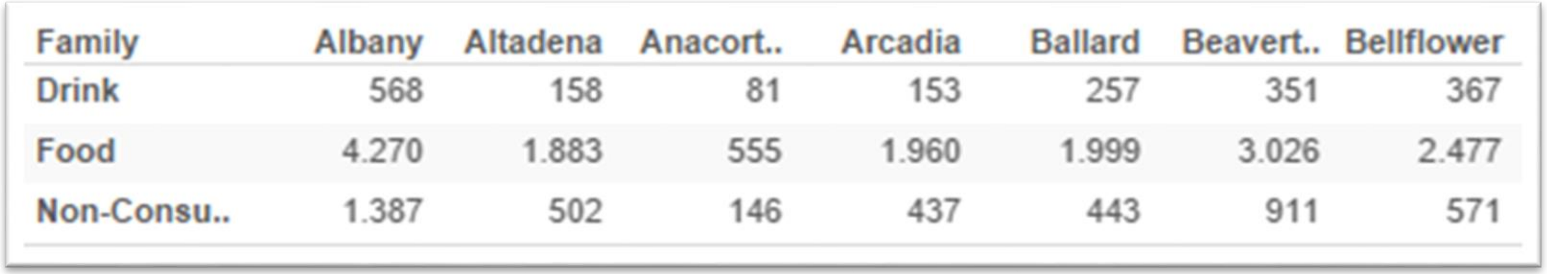


Family	Albany	Altadena	Anacort..	Arcadia	Ballard	Beavert..	Bellflower
Drink	568	158	81	153	257	351	367
Food	4.270	1.883	555	1.960	1.999	3.026	2.477
Non-Consu..	1.387	502	146	437	443	911	571

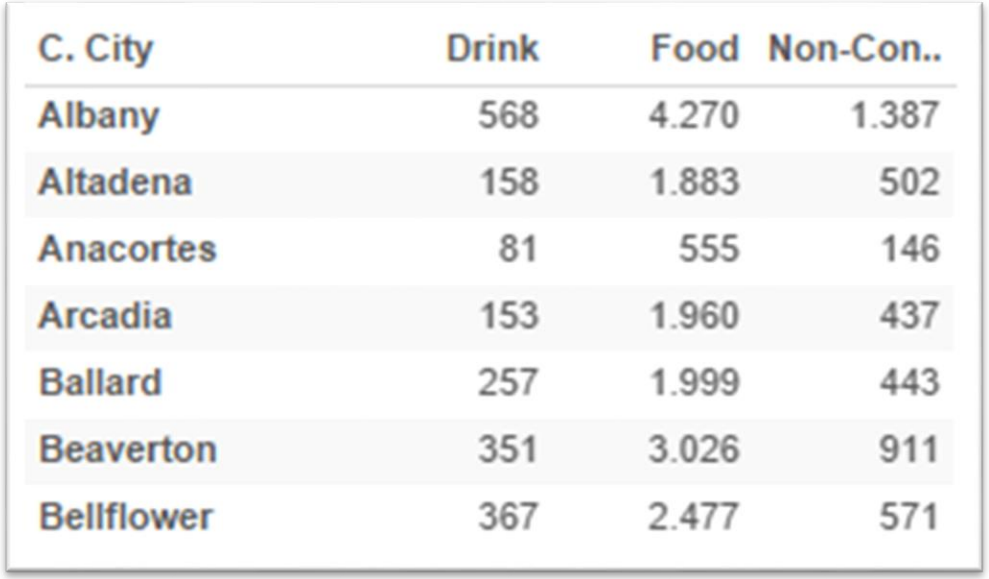
SUM

Sex = “F”

OLAP Operators: Pivoting



Family	Albany	Altadena	Anacort..	Arcadia	Ballard	Beavert..	Bellflower
Drink	568	158	81	153	257	351	367
Food	4.270	1.883	555	1.960	1.999	3.026	2.477
Non-Consu..	1.387	502	146	437	443	911	571



C. City	Drink	Food	Non-Con..
Albany	568	4.270	1.387
Altadena	158	1.883	502
Anacortes	81	555	146
Arcadia	153	1.960	437
Ballard	257	1.999	443
Beaverton	351	3.026	911
Bellflower	367	2.477	571

Tableau: Visual Analytics

Data

Query

MySQL

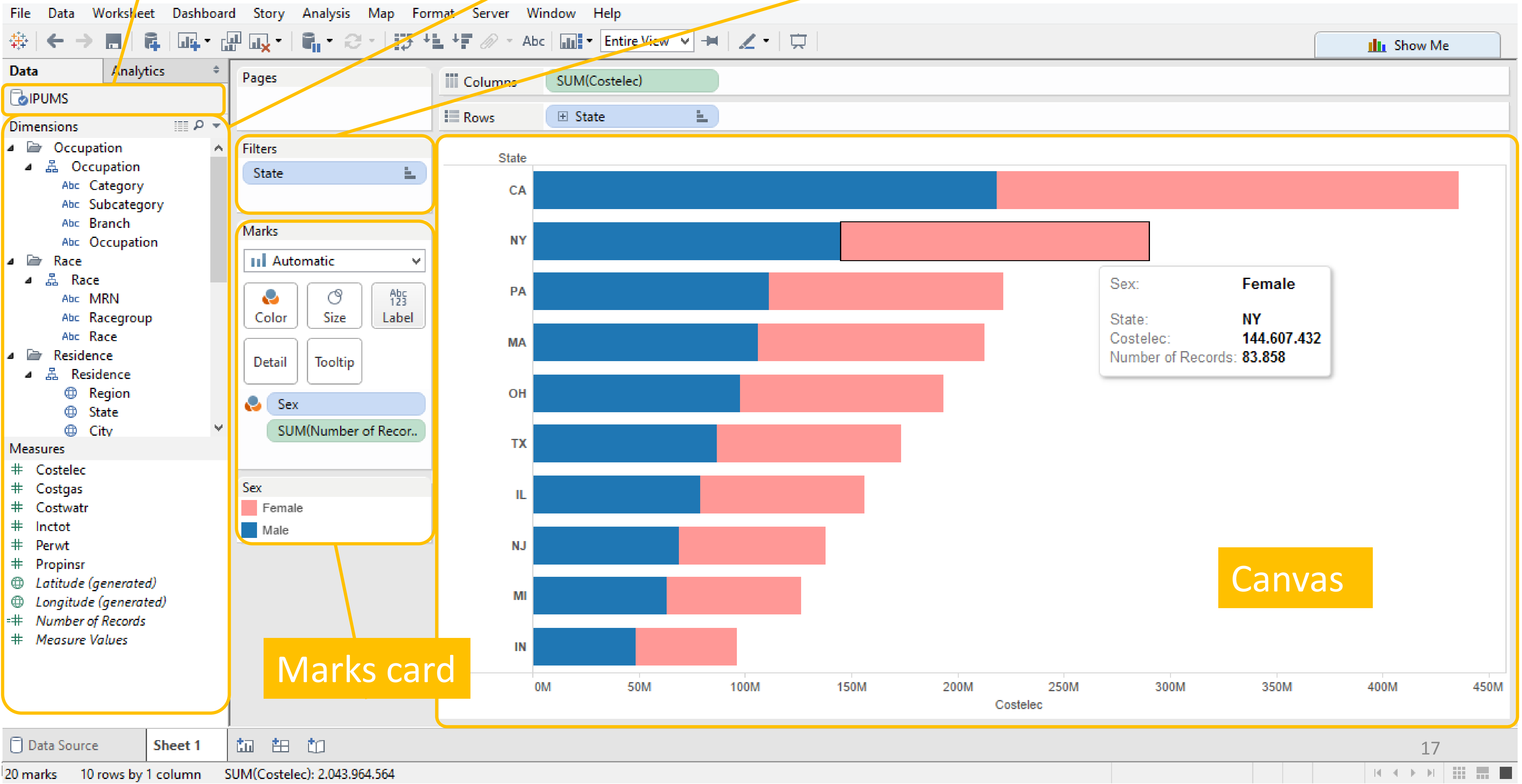
TXT

ODBC

Data sources

Dimensions and measures

Filters



IPUMS

Dimensions
- Occupation
  - Occupation
    - Abc Category
    - Abc Subcategory
    - Abc Branch
    - Abc Occupation
- Race
  - Race
    - Abc MRN
    - Abc Racegroup
    - Abc Race
- Residence
  - Residence
    - Region
    - State
    - City

Measures
- Costelec
- Costgas
- Costwatr
- Inctot
- Perwt
- Propinsr
- *Latitude (generated)*
- *Longitude (generated)*
- *Number of Records*
- *Measure Values*

Columns: SUM(Costelec)

Rows: State

Filters: State

Marks: Sex; SUM(Number of Recor..

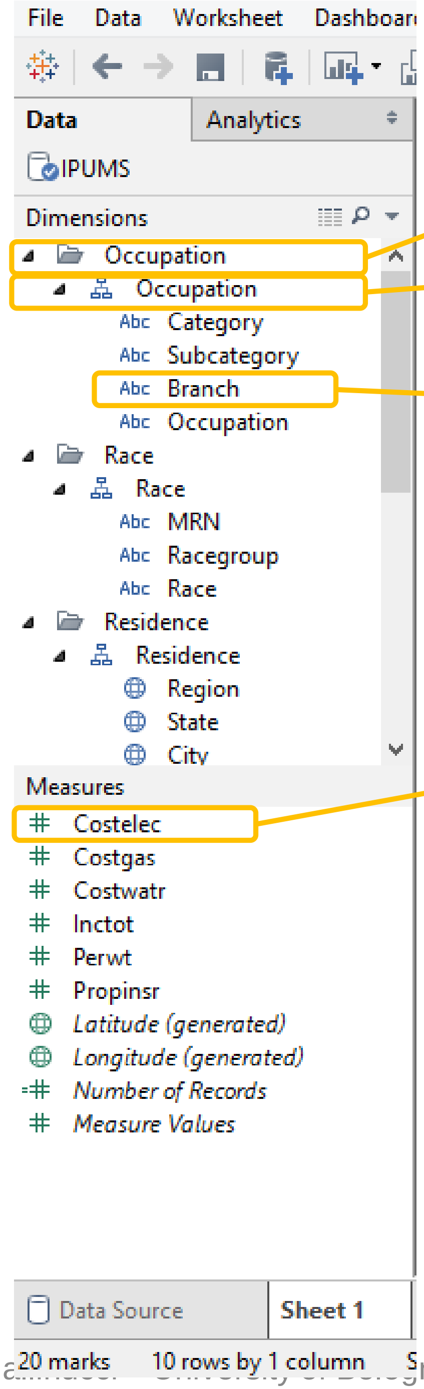
Sex
- 🟥 Female
- 🟦 Male

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

Canvas

Marks card



IPUMS

Dimensions

- Occupation
  - Occupation
    - Abc Category
    - Abc Subcategory
    - Abc Branch
    - Abc Occupation
- Race
  - Race
    - Abc MRN
    - Abc Racegroup
    - Abc Race
- Residence
  - Residence
    - Region
    - State
    - City

Measures

- # Costelec
- # Costgas
- # Costwatr
- # Inctot
- # Perwt
- # Propinsr
- *Latitude (generated)*
- *Longitude (generated)*
- *Number of Records*
- *Measure Values*

Dimension

Hierarchy

Dimensional attribute

Measure

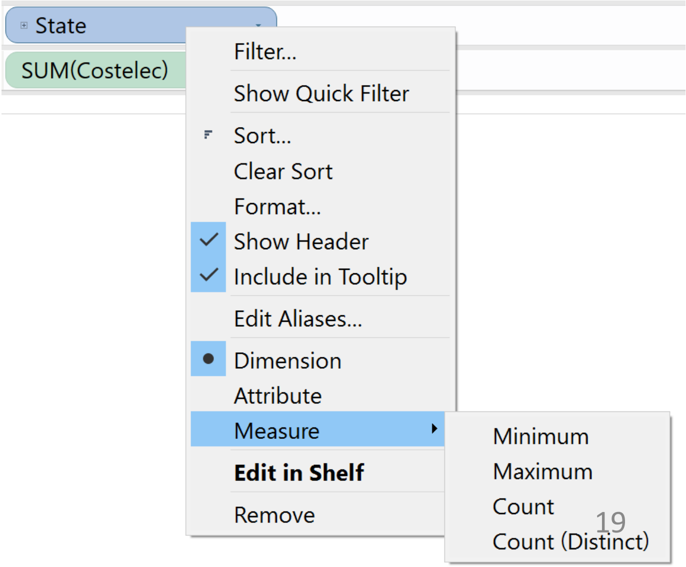
- Conventionally, folders are used to represent dimensions; in general they are simply a way to group elements

Dimension VS Measure

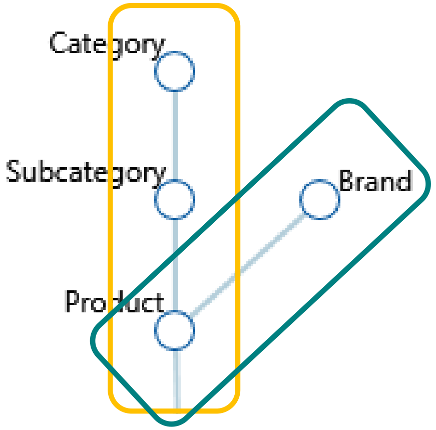
With Tableau the definitions of dimension and measure are quite loose. Indeed, every field can be used both as a dimension and as a measure

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!)

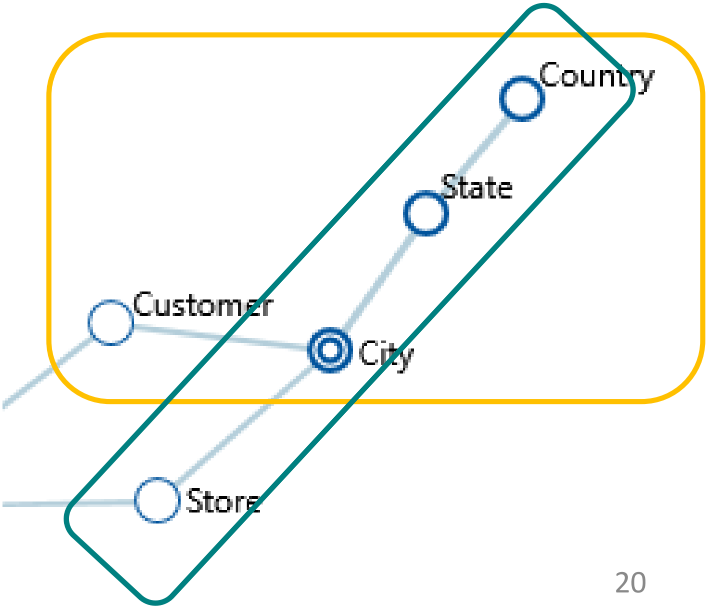


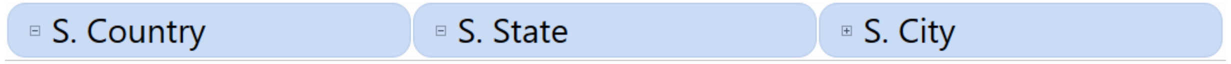
Hierarchies



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

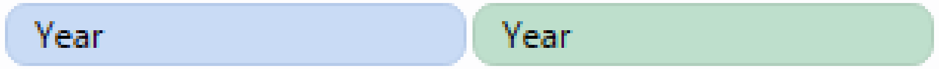
- Shared hierarchies are duplicated





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

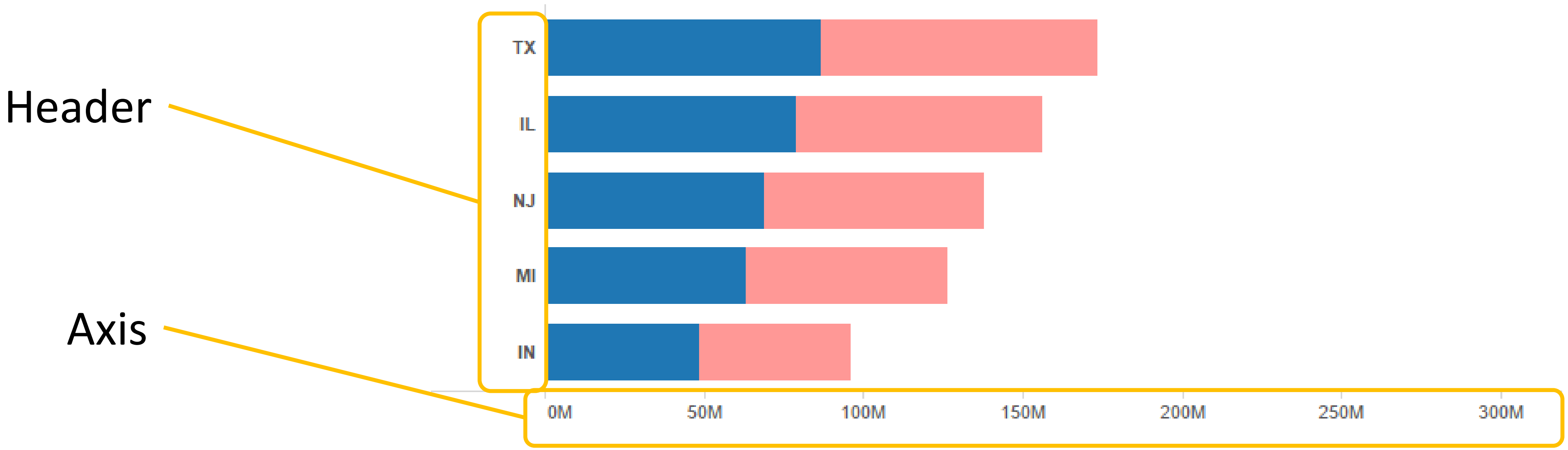
Continuous and discrete fields behave in different ways

- 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)

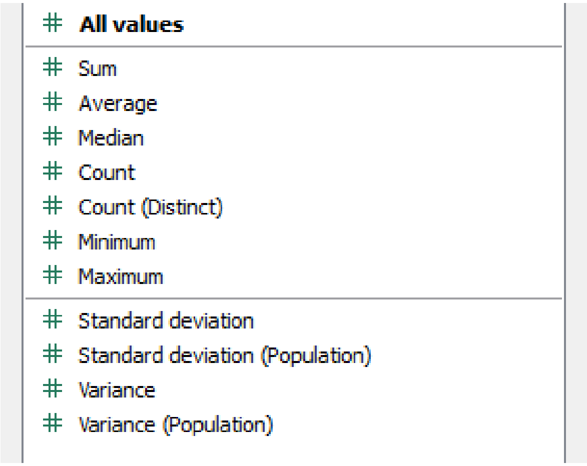
When they are used on rows and columns

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



Green VS Blue (Filters)

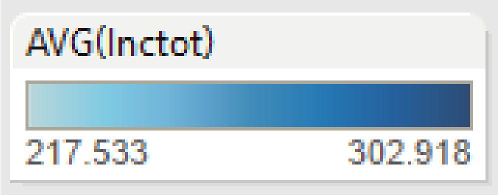
- With a continuous field it is possible to specify a *range*
  - On the values at the most detailed level (i.e., not aggregated)
  - Or at a particular aggregation level of the base values



- With a discrete field it is also possible to select specific values one by one

Green VS Blue (Colours)

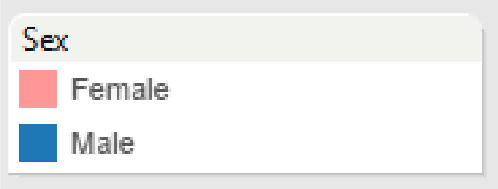
- A continuous field is associated to a sequence of related colours; e.g., different shades of the same base colour: darker and brighter shades may refer to respectively higher and lower values



AVG(Inctot)

217.533 302.918

- With a discrete field each value is associated to a single arbitrary colour without any constraint

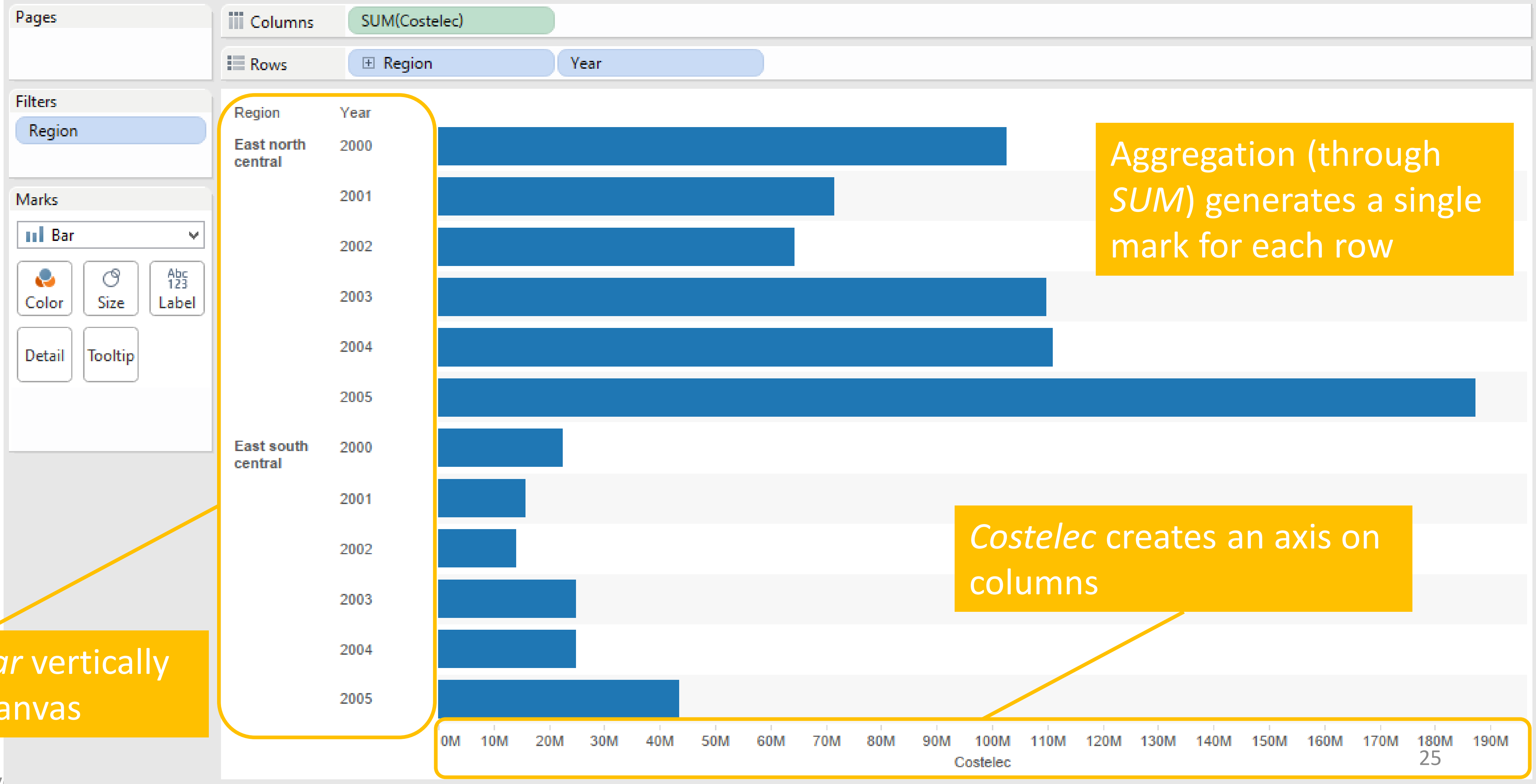


Sex

🟥 Female

🟦 Male

Canvas



Aggregation (through *SUM*) generates a single mark for each row

Costelec creates an axis on columns

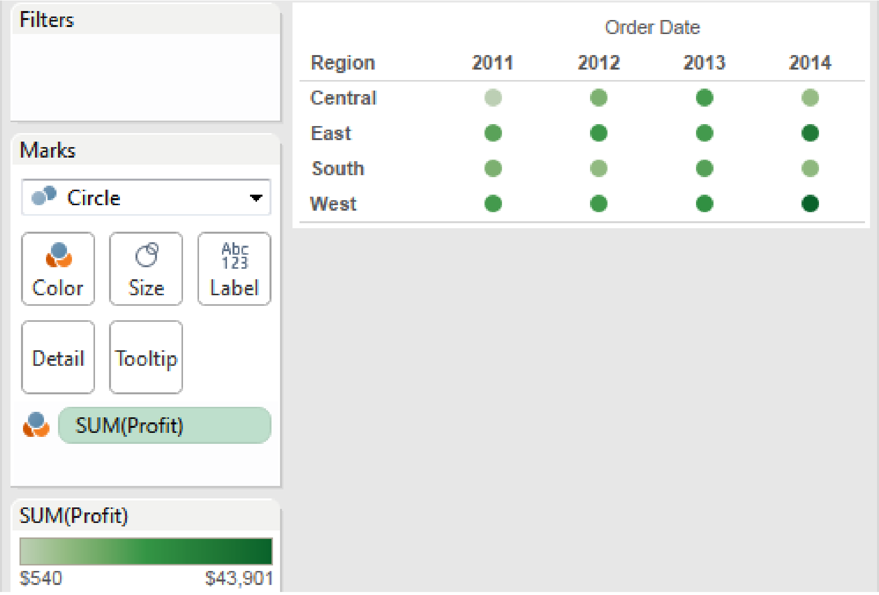
Region and *Year* vertically partition the canvas

Mark

Data are visualised inside the canvas through the use of *marks*

There are different types of marks (*bar*, *line*, *text*, etc.)

- Each mark type has differe
- At each property can be a pend on whether the field is continuous or discrete and



	Order Date			
Region	2011	2012	2013	2014
Central	●	●	●	●
East	●	●	●	●
South	●	●	●	●
West	●	●	●	●

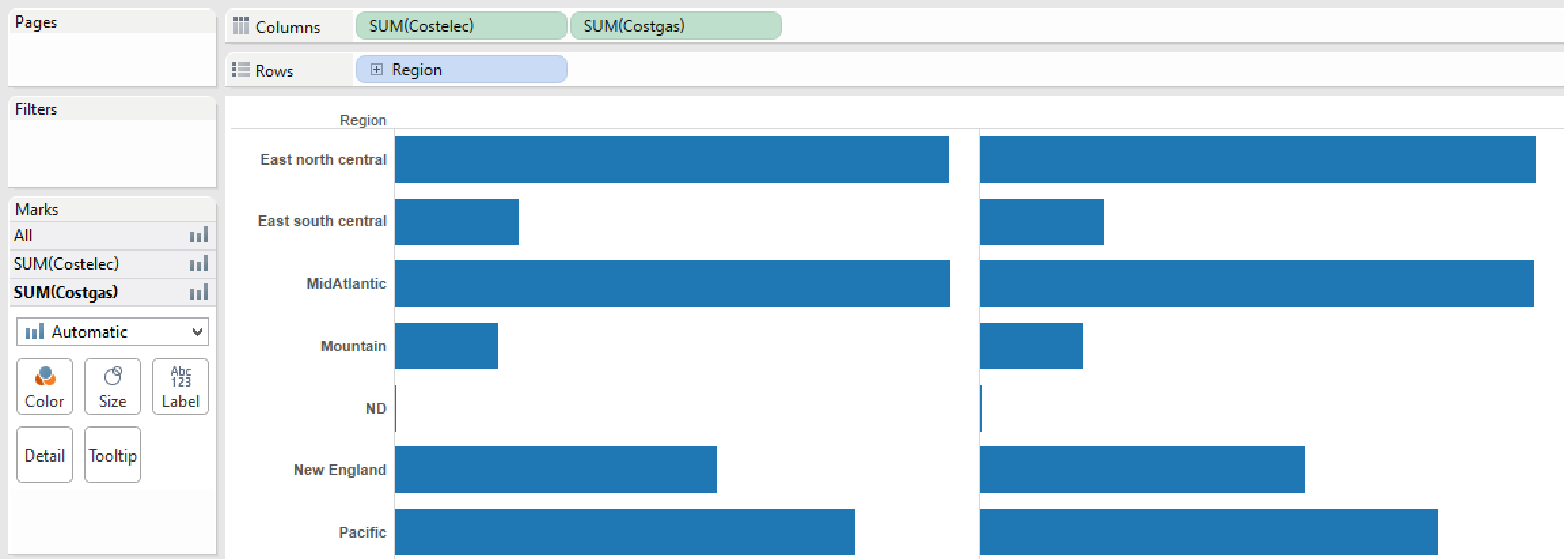
SUM(Profit)

$540 $43,901

Mark (2)

A

D



Columns: SUM(Costelec) SUM(Costgas)

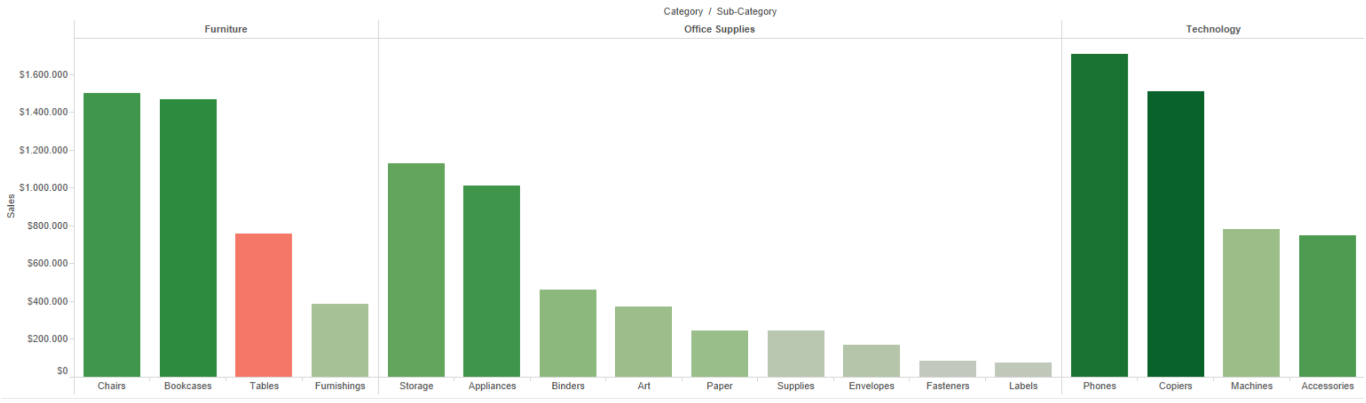
Rows: Region

Marks: All, SUM(Costelec), **SUM(Costgas)**

Automatic

Ordering

Ordering does not break hierarchies (although there exist workarounds)

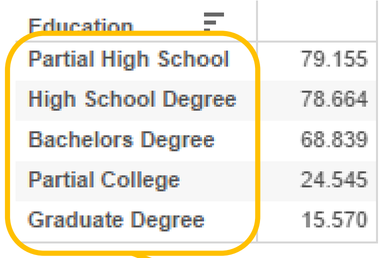


There are different types of ordering

- *Alphabetical*
- *Manual*: the order is fixed and manually chosen by the user
- *Computed*: the order is based on a computation (e.g., the sum of specific measure)
- *Data Source Order*: the order is the same as on the datasource
- *Nested: useful* in the presence of two (or more) independent fields

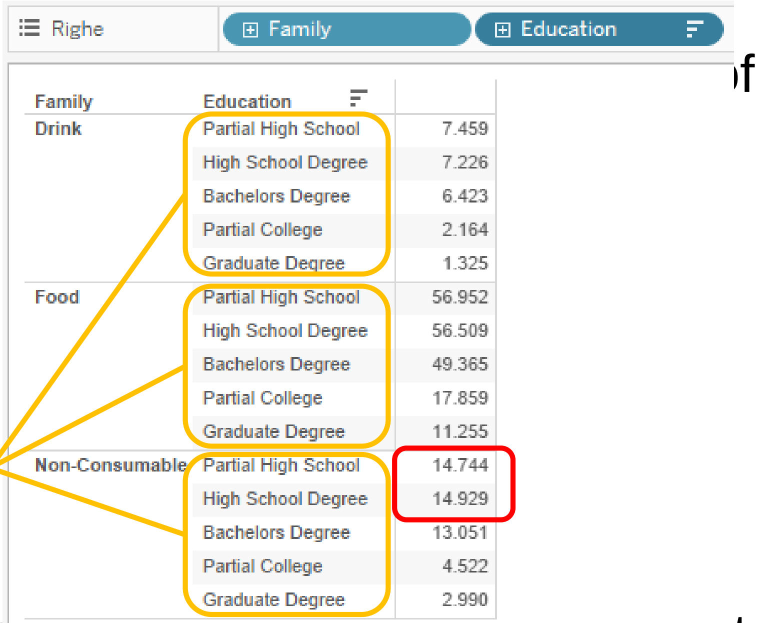
Ordering (2)

on ordered … f unit sales)

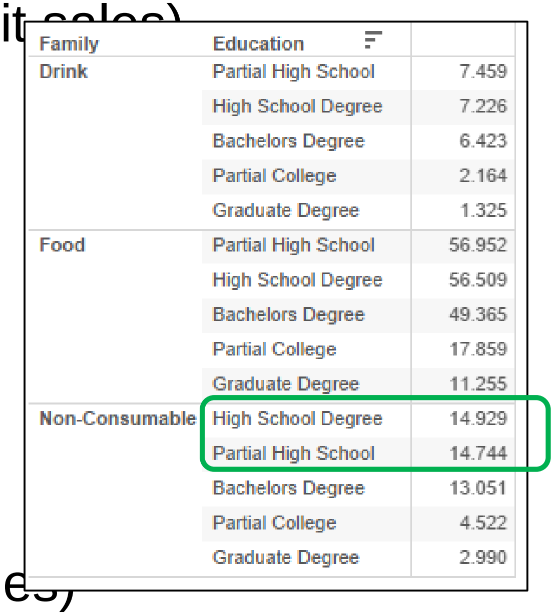


Education	
Partial High School	79.155
High School Degree	78.664
Bachelors Degree	68.839
Partial College	24.545
Graduate Degree	15.570

The order used for Education is the one obtained by evaluating Education independently!



Family	Education	
Drink	Partial High School	7.459
	High School Degree	7.226
	Bachelors Degree	6.423
	Partial College	2.164
	Graduate Degree	1.325
Food	Partial High School	56.952
	High School Degree	56.509
	Bachelors Degree	49.365
	Partial College	17.859
	Graduate Degree	11.255
Non-Consumable	Partial High School	14.744
	High School Degree	14.929
	Bachelors Degree	13.051
	Partial College	4.522
	Graduate Degree	2.990



Family	Education	
Drink	Partial High School	7.459
	High School Degree	7.226
	Bachelors Degree	6.423
	Partial College	2.164
	Graduate Degree	1.325
Food	Partial High School	56.952
	High School Degree	56.509
	Bachelors Degree	49.365
	Partial College	17.859
	Graduate Degree	11.255
Non-Consumable	High School Degree	14.929
	Partial High School	14.744
	Bachelors Degree	13.051
	Partial College	4.522
	Graduate Degree	2.990

Right: Education ordered by nested field (desc sum of unit sales)

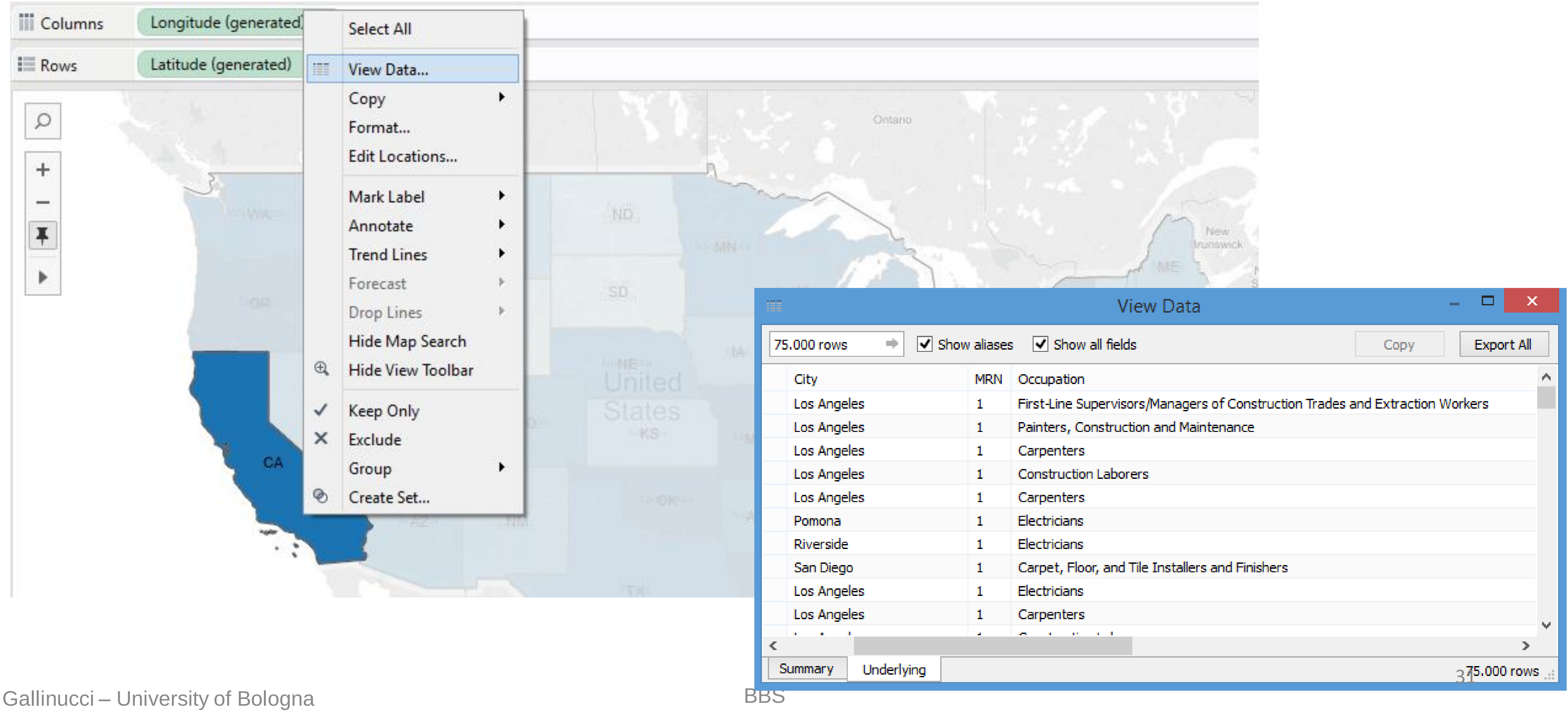
View Data

Through the *View Data* option it is possible to visualise the set of records (i.e., the data at the most detailed level of aggregation) used to compute a certain mark

View Data can be used as a *light* version of the *Drill Through* operation

It is particularly useful for testing and debugging complex visualisations

View Data (2)



75.000 rows ☑ Show aliases ☑ Show all fields

City	MRN	Occupation
Los Angeles	1	First-Line Supervisors/Managers of Construction Trades and Extraction Workers
Los Angeles	1	Painters, Construction and Maintenance
Los Angeles	1	Carpenters
Los Angeles	1	Construction Laborers
Los Angeles	1	Carpenters
Pomona	1	Electricians
Riverside	1	Electricians
San Diego	1	Carpet, Floor, and Tile Installers and Finishers
Los Angeles	1	Electricians
Los Angeles	1	Carpenters

75.000 rows

Filters

It is possible to apply a filter (i.e., *Slice & Dice*)

- To specific worksheets: the filter is applied only to the specified worksheets
- To a specific data source: the filter is applied to all the worksheets that use data from that source

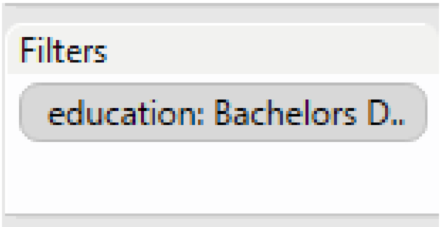
It is possible to filter

- At *record level*: the view is computed using only the records that satisfy the filter; each filter is computed independently from the others
  - Eg. *Sales* > 100.00$
- At *aggregation level*: after that the view has been computed, all cells containing marks that do not satisfy the aggregation level filter are removed
  - Eg. SUM(*Sales*) > 100.00$

Filters: Context Filter

A *Context Filter* is a particular type of filter that is applied *before* record and aggregation level filters (i.e., the other filters depend on the result of context filters)

A context filter cannot be applied to aggregations (i.e., record level only)

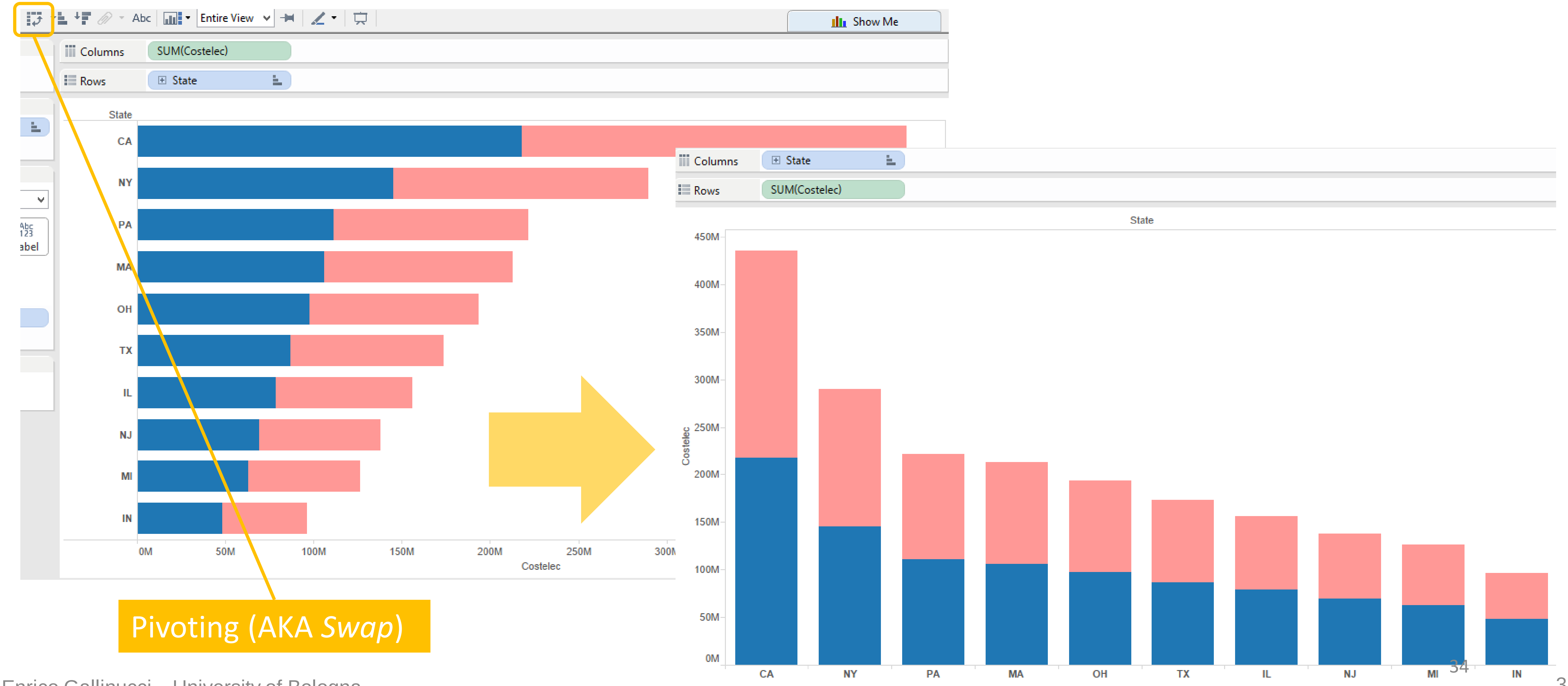


Filters

education: Bachelors D..

Context filters can be distinguished by their gray colour (for both discrete and continuous fields)

Pivoting



Columns: SUM(Costelec)

Rows: State

Columns: State

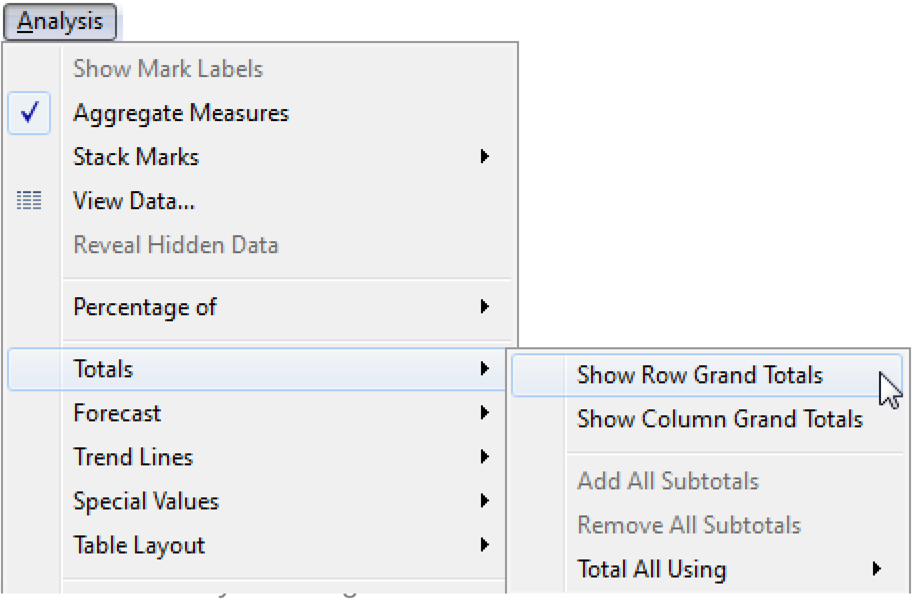
Rows: SUM(Costelec)

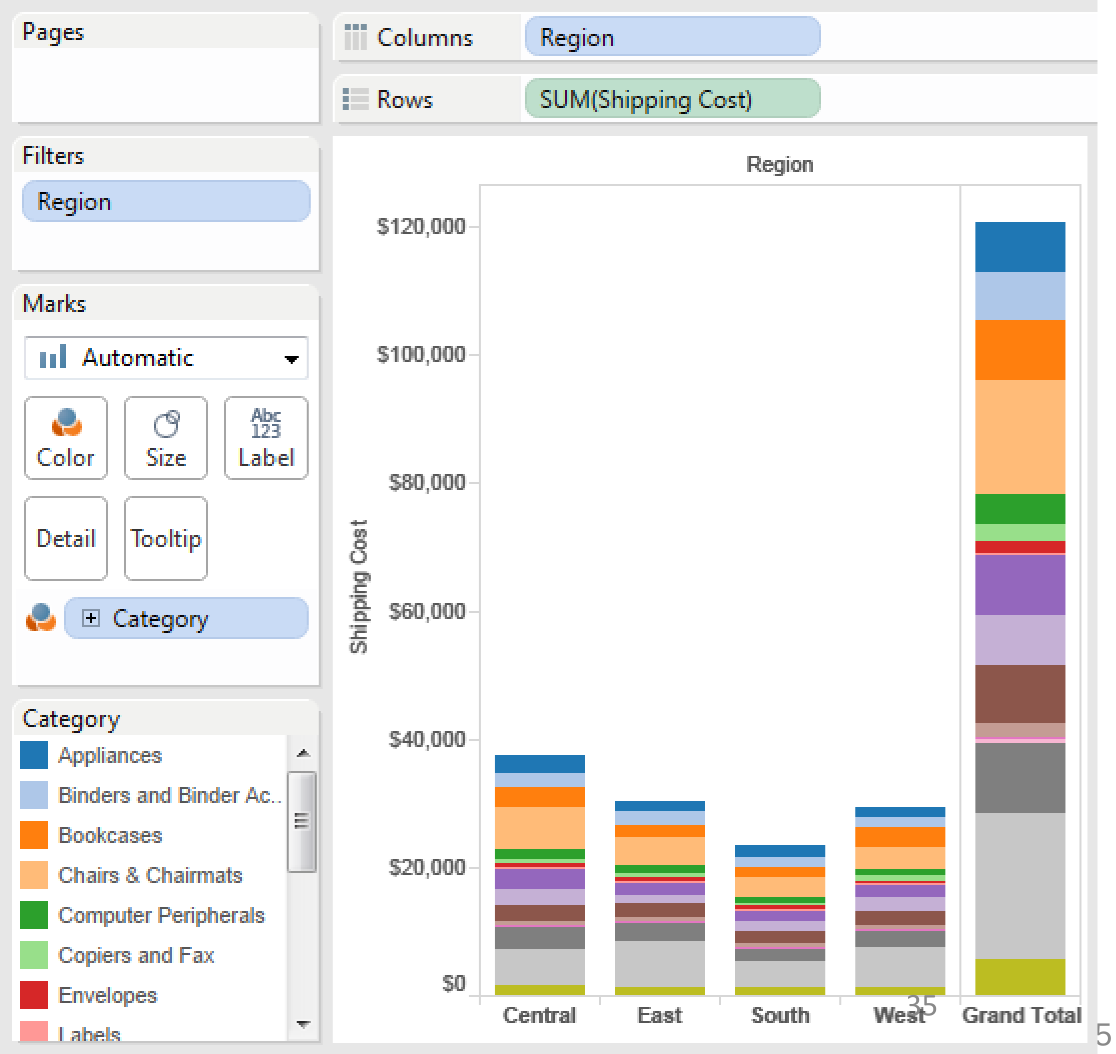
Pivoting (AKA *Swap*)

Grand Totals and Sub Totals

Totals are useful to show different aggregation levels in the same view

They can be applied only to discrete fields



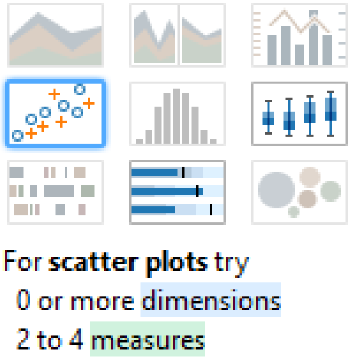


Show Me

The *Show Me* window contains shortcuts to create advanced visualisations starting from a set of dimensions and measures

Each visualisation type in the Show Me window has specific requirements: e.g., a scatter plot can be created only if there are at least continuous fields

Some types of visualisation can be quite difficult to build from scratch, for instance maps and box-plots



For **scatter plots** try
0 or more dimensions
2 to 4 measures

Top N

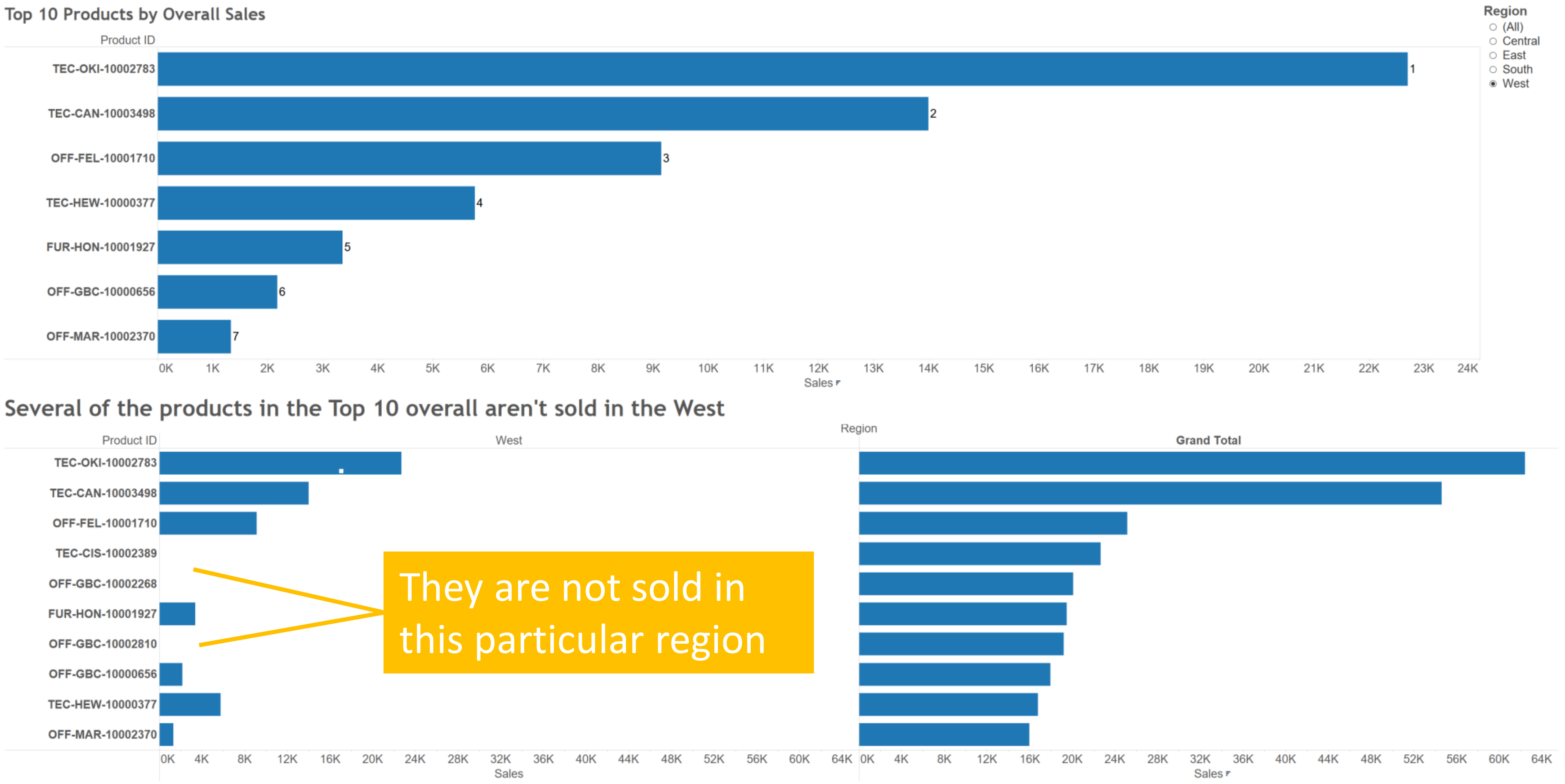
It is also possible to keep only the first (or last) *N* elements based on a given ordering

- E.g., the first 10 categories by sold amount

Attention: top / bottom filters are applied independently from other filters and from the view

- E.g., selecting a region while using a top 10 filter on categories would result in a visualisation that shows the categories whose sales are globally (i.e., across all the regions) the top 10!

Top N (2)



They are not sold in this particular region

Binning

Through binning it is possible to create *bins* (or *buckets*), which are numerical ranges over the values of a numerical attribute

- In Tableau, ranges are left-inclusive and right-exclusive: [*start*, *end*)

Binning is useful to *discretize* continuous attributes while controlling the number of shown values; it is also possible to further group discrete numerical attributes to reduce their number (i.e., show fewer labels)

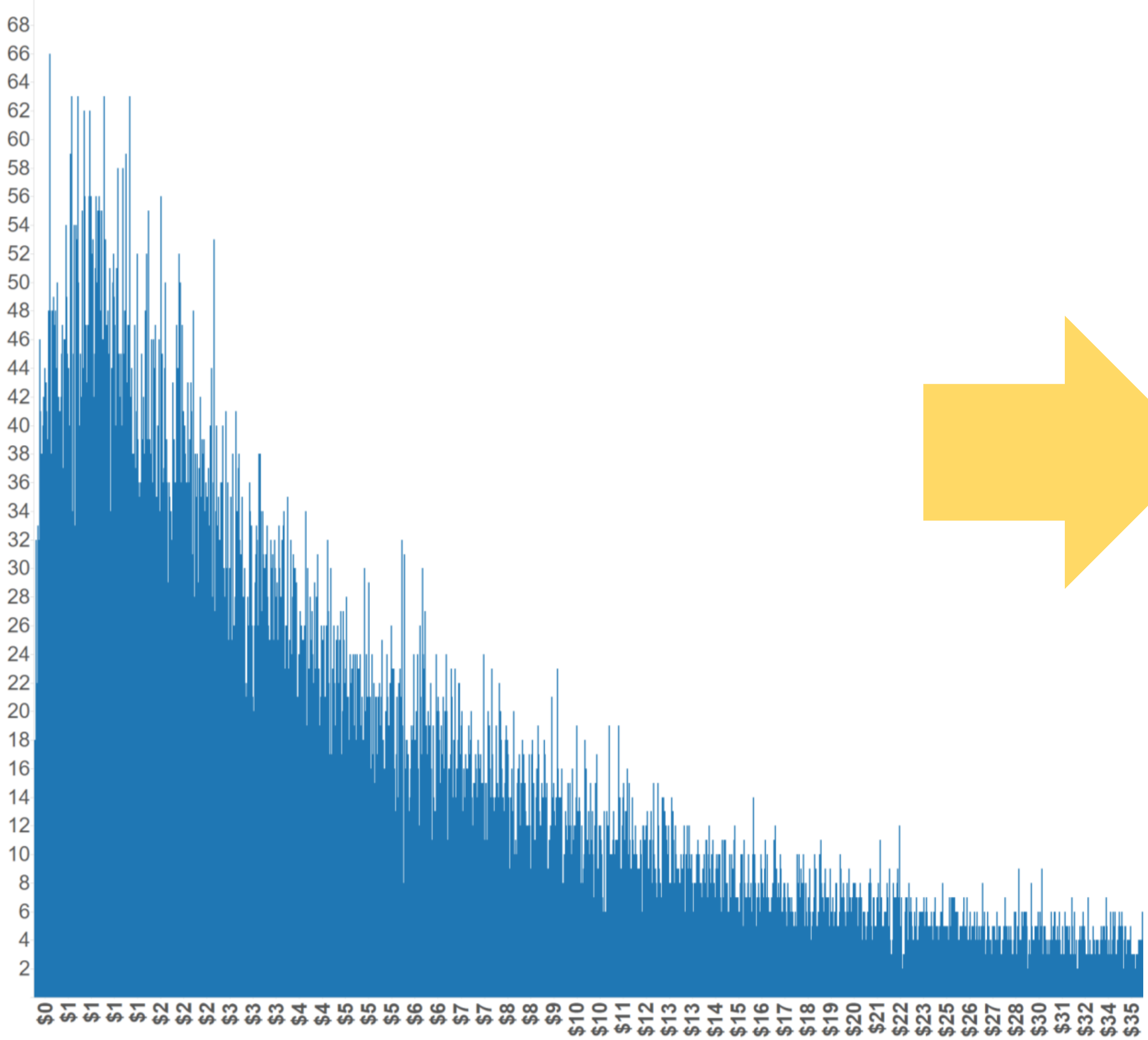
Binning can also be used as a *smoothing* mechanism, that is, to reduce the effect small variations considered noise

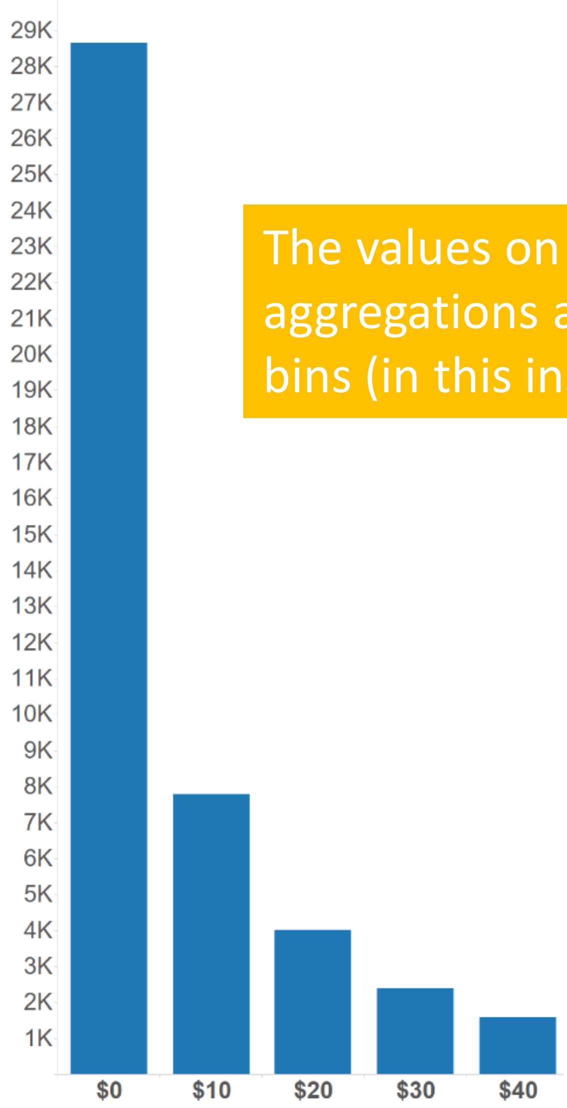
Histograms are created by first binning and then counting the number of elements belonging to each bin

Measures are aggregated by bin

Tableau only supports *equi-width* bins, i.e., each bin has the same range length

Binning (2)



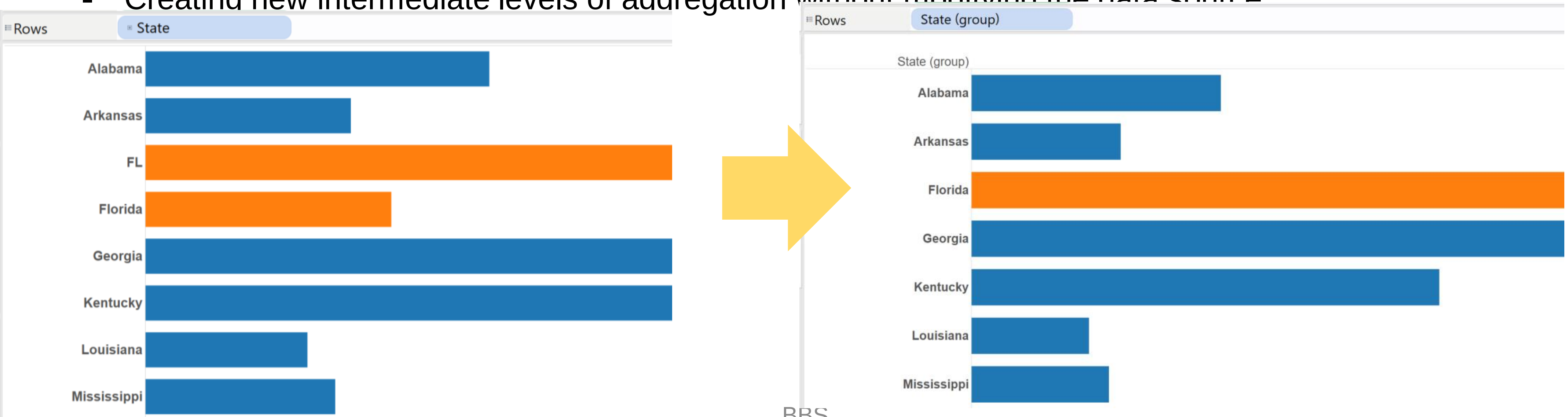


The values on the y-axis are aggregations associated to the bins (in this instance, *sum*)

Group

It is possible to create new attributes by (manually) grouping values of already existing ones; especially useful for

- Grouping values that have the same semantic but are labeled differently
- Creating new intermediate levels of aggregation without modifying the data source



Set

Sets are custom fields that allow to create subsets of attribute values (cannot be used with measures) that satisfy a certain condition. Based on the defined condition, sets can be

- *Constant*: the elements of the set must be manually selected; the set remains the same even when underlying data change
- *Computed*: the set is created by defining a condition (e.g., sales > 100) that the elements must satisfy

Sets can be used as filters, hierarchy levels, inside calculated fields, …

Different sets can be *combined* to create new ones

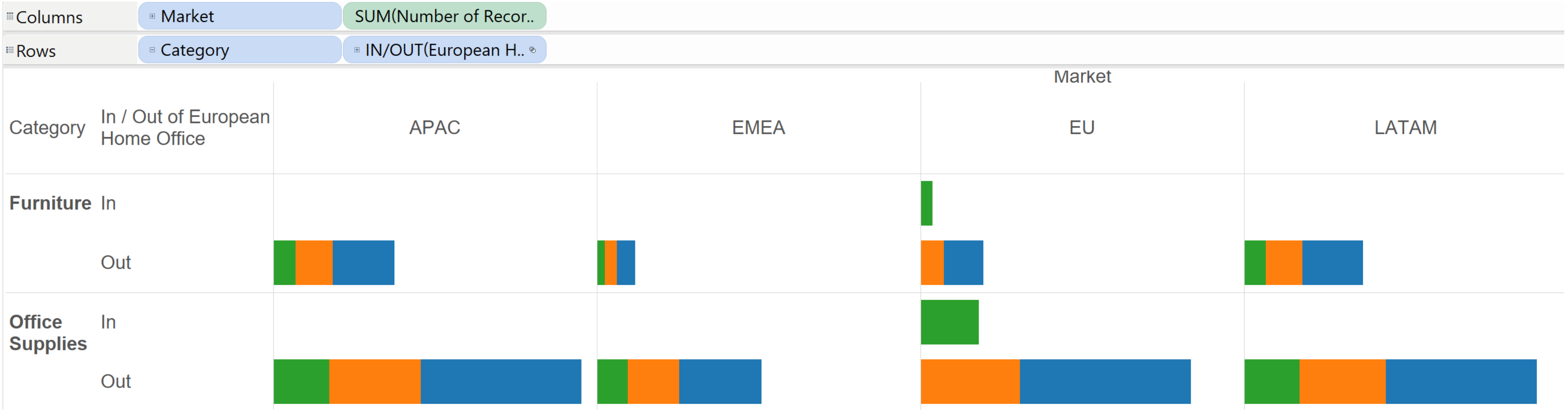
- For instance, it is possible to create the union of two sets

Set (2)

When used as filters, all elements that do not belong to the set are discarded

When used as a dimension, a set creates a new header composed by values *In* and *Out*

Similarly, when assigned to the colour property, markers are split into two (one colour for *In* and one for *Out*)



Columns: Market | SUM(Number of Recor..

Rows: Category | IN/OUT(European H..

Calculated Field

A *calculated field* is a field defined by a formula that can make use of other fields and various functions such as arithmetic and logical operations

- E.g., the *Profit* field can be defined as *Sales* - *Cost*

Through a calculated field it is possible to create new fields without modifying the data source

A calculated field can be defined at different granularity levels

- *Line Granularity*: the field is computed tuple by tuple (e.g., profit)
- *Aggregated Granularity*: the is computed over other aggregations

A calculated field can be used as any other field; an exception are calculated fields with aggregated granularity, which can be used as filters only if they are continuous fields

Calculated Field: Syntax

To use an existing field inside a formula it is sufficient to type its name surrounded by square brackets

- e.g., `[Sales]`

Conditional statements

```
if [Profit] > 0 then
   'Profitable'
else
   'Not profitable'
end
```

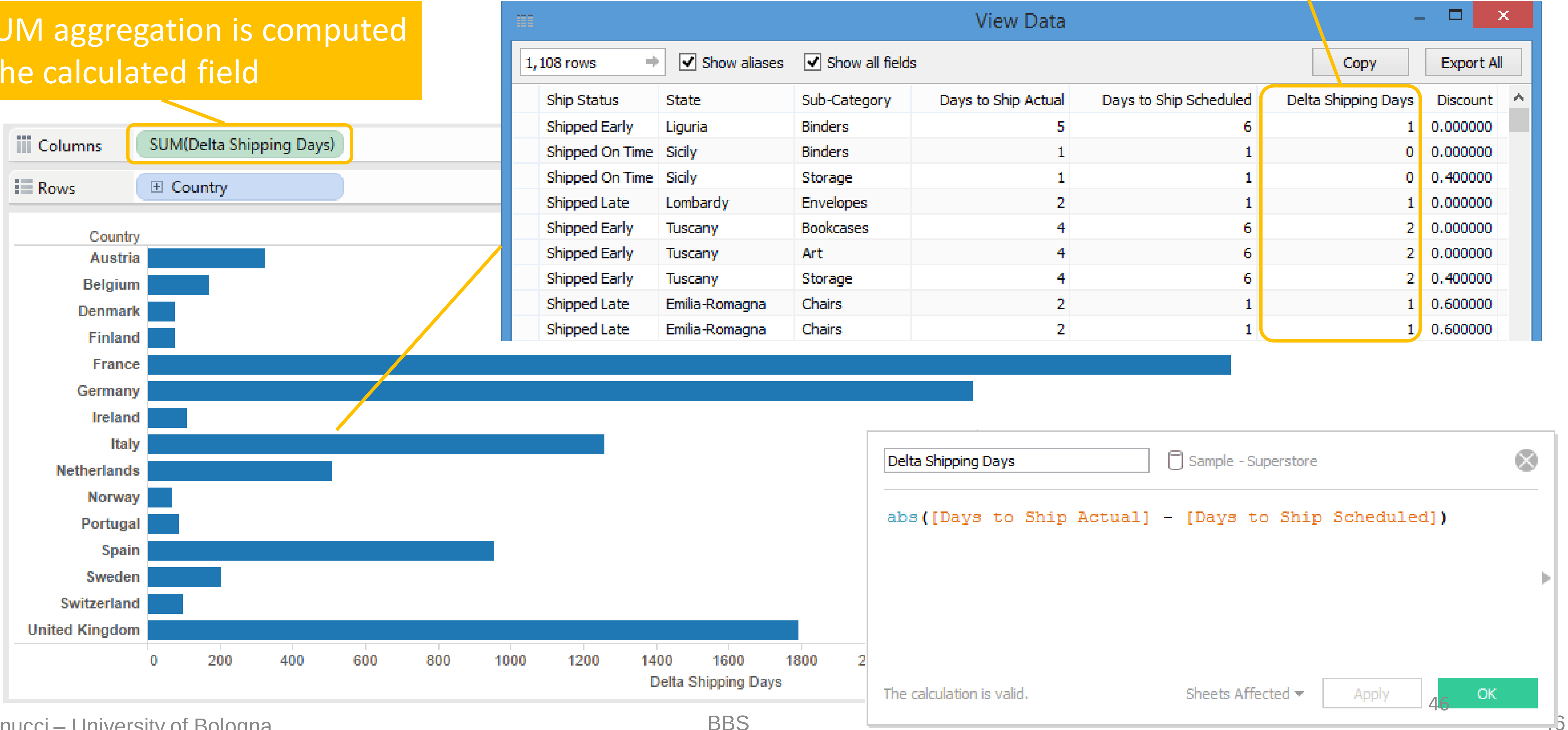
A function can be called by simply typing its name followed by its parameters inside parentheses; the parameters must be separated by commas

- e.g., `MAX([Sales], [Cost])`

Calculated Field: Line Gran

Before computing the view, a new value is computed for each tuple

The SUM aggregation is computed over the calculated field



Ship Status	State	Sub-Category	Days to Ship Actual	Days to Ship Scheduled	Delta Shipping Days	Discount
Shipped Early	Liguria	Binders	5	6	1	0.000000
Shipped On Time	Sicily	Binders	1	1	0	0.000000
Shipped On Time	Sicily	Storage	1	1	0	0.400000
Shipped Late	Lombardy	Envelopes	2	1	1	0.000000
Shipped Early	Tuscany	Bookcases	4	6	2	0.000000
Shipped Early	Tuscany	Art	4	6	2	0.000000
Shipped Early	Tuscany	Storage	4	6	2	0.400000
Shipped Late	Emilia-Romagna	Chairs	2	1	1	0.600000
Shipped Late	Emilia-Romagna	Chairs	2	1	1	0.600000

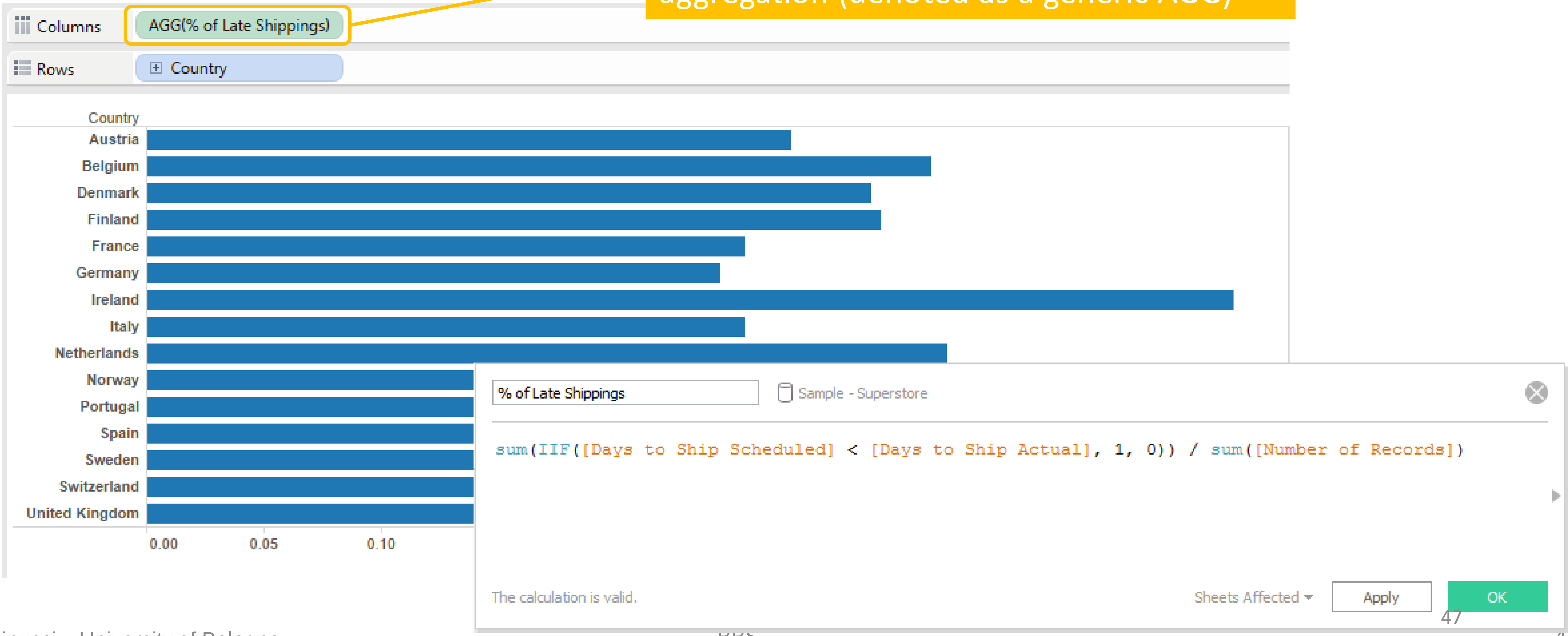
Delta Shipping Days

```
abs([Days to Ship Actual] - [Days to Ship Scheduled])
```

The calculation is valid.

Calculated Field: Aggregated Granularity

The calculated field implicitly defines an aggregation (denoted as a generic AGG)



Columns: AGG(% of Late Shippings)

Rows: Country

% of Late Shippings

```
sum(IIF([Days to Ship Scheduled] < [Days to Ship Actual], 1, 0)) / sum([Number of Records])
```

The calculation is valid.

More functionalities: calculated fields

Can be defined at different granularities

- Line granularity: e.g., *[Extendedprice]*[Quantity]*
- Aggregated granularity: e.g., *SUM(IF ([Tax]>0) THEN 1 ELSE 0 END)/COUNT([Quantity])*

Categorical fields can be defined as well

- E.g., *IF ([Tax]>0) THEN 'Taxed' ELSE 'Not taxed' END*

Level-of-detail (LOD) expressions: fix the reference group-by attributes

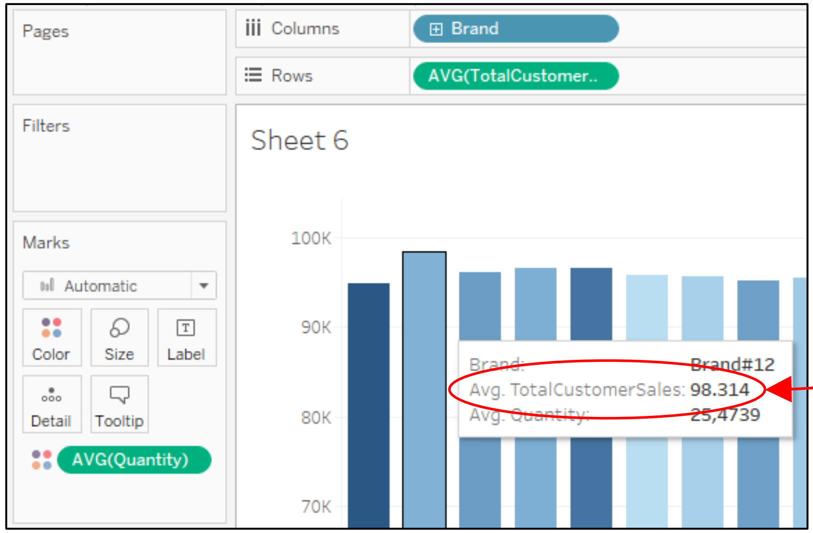
- Define reference aggregated values
- Useful to create more advanced queries combining more aggregations

More functionalities: LOD fields

Nested aggregation

- Further aggregate the results of a query
- For instance:
  - Calculate the sum(Quantity) by Brand and Nation
  - Calculate the average of the result by Brand
  - TotalCustomerSales = {INCLUDE [Nation (Customer)]: SUM([Quantity])}

TotalCustomerSales is defined at a *finer* aggregation level



Pages

Columns: Brand

Rows: AVG(TotalCustomer..

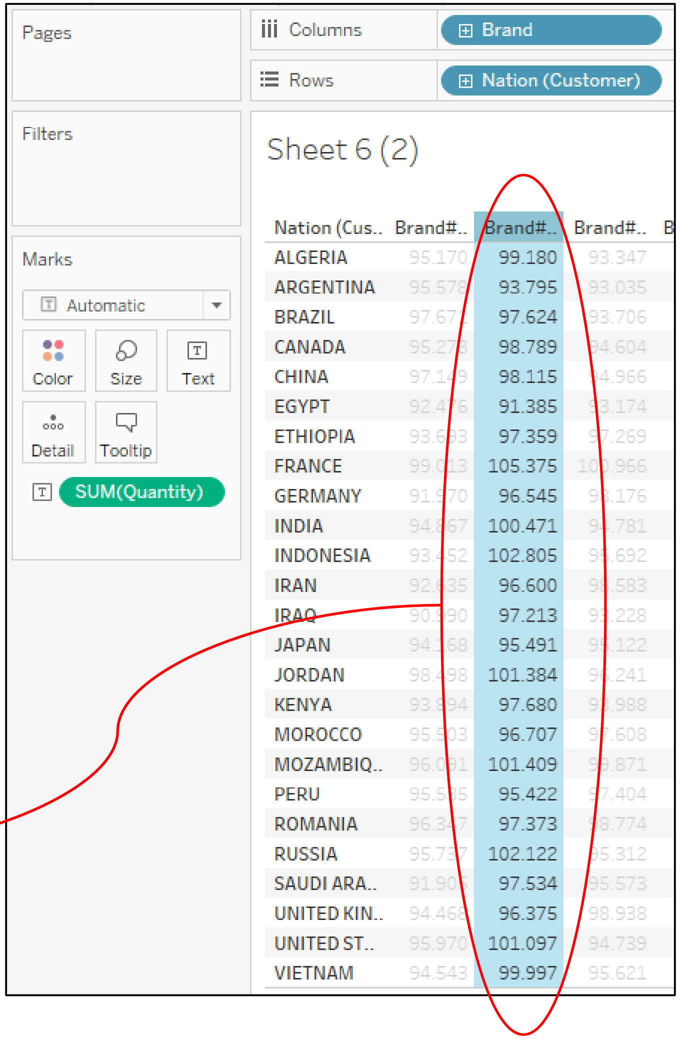
Filters

Marks: Automatic

AVG(Quantity)

Sheet 6

Brand: Brand#12
Avg. TotalCustomerSales: 98.314
Avg. Quantity: 25,4739



Pages

Columns: Brand

Rows: Nation (Customer)

Filters

Marks: Automatic

SUM(Quantity)

Sheet 6 (2)

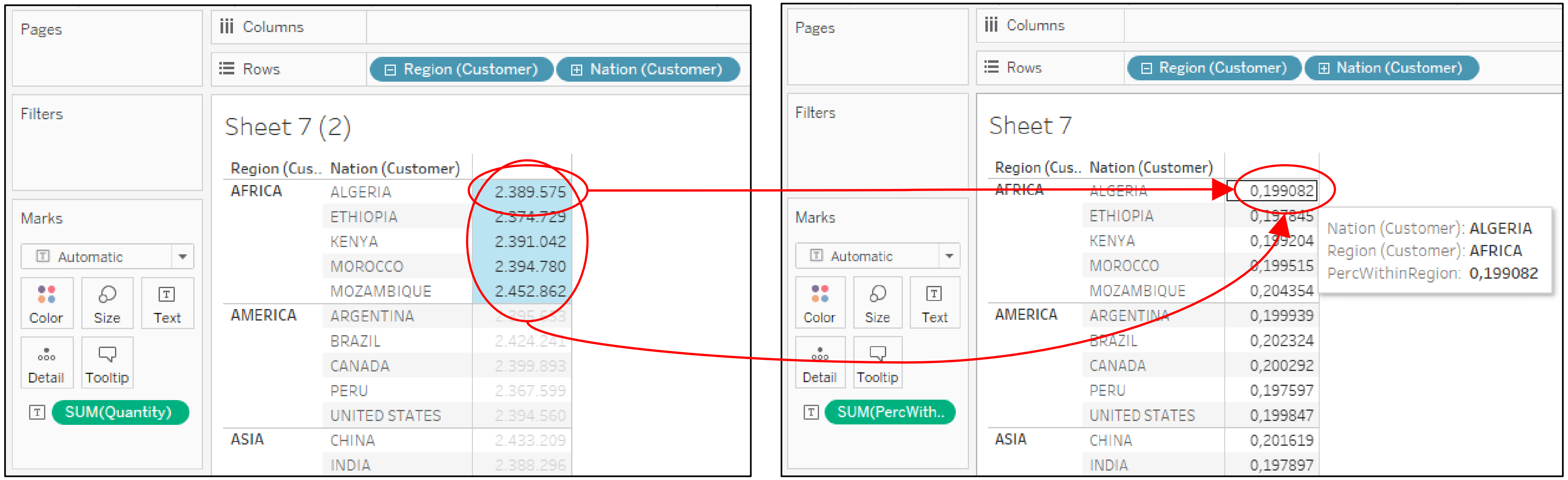
Nation (Cus..	Brand#..	Brand#..	Brand#..	B
ALGERIA	95.170	99.180	93.347	
ARGENTINA	95.578	93.795	93.035	
BRAZIL	97.671	97.624	93.706	
CANADA	95.273	98.789	94.604	
CHINA	97.149	98.115	94.966	
EGYPT	92.476	91.385	93.174	
ETHIOPIA	93.693	97.359	97.269	
FRANCE	99.013	105.375	100.966	
GERMANY	91.970	96.545	98.176	
INDIA	94.867	100.471	94.781	
INDONESIA	93.452	102.805	95.692	
IRAN	92.635	96.600	98.583	
IRAQ	90.990	97.213	93.228	
JAPAN	94.168	95.491	95.122	
JORDAN	98.498	101.384	96.241	
KENYA	93.894	97.680	93.988	
MOROCCO	95.503	96.707	97.608	
MOZAMBIQ..	96.091	101.409	99.871	
PERU	95.535	95.422	97.404	
ROMANIA	96.347	97.373	98.774	
RUSSIA	95.737	102.122	95.312	
SAUDI ARA..	91.905	97.534	95.573	
UNITED KIN..	94.468	96.375	98.938	
UNITED ST..	95.970	101.097	94.739	
VIETNAM	94.543	99.997	95.621	

More functionalities: LOD fields

Use coarser data at a finer aggregation level

- For instance:
  - Calculate the sum(Quantity) by Nation (and Region)
  - Divide the result by the sum(Quantity) by Region
  - PercWithinRegion =
    {FIXED [Nation (Customer)]: SUM([Quantity])} / {FIXED [Region (Customer)]: SUM([Quantity])}

PercWithinRegion embeds a calculation at a *coarser* aggregation level



Sheet 7 (2)

Region (Cus..	Nation (Customer)	
AFRICA	ALGERIA	2.389.575
	ETHIOPIA	2.374.729
	KENYA	2.391.042
	MOROCCO	2.394.780
	MOZAMBIQUE	2.452.862
AMERICA	ARGENTINA	2.395.653
	BRAZIL	2.424.241
	CANADA	2.399.893
	PERU	2.367.599
	UNITED STATES	2.394.560
ASIA	CHINA	2.433.209
	INDIA	2.388.296

Sheet 7

Region (Cus..	Nation (Customer)	
AFRICA	ALGERIA	0,199082
	ETHIOPIA	0,197845
	KENYA	0,199204
	MOROCCO	0,199515
	MOZAMBIQUE	0,204354
AMERICA	ARGENTINA	0,199939
	BRAZIL	0,202324
	CANADA	0,200292
	PERU	0,197597
	UNITED STATES	0,199847
ASIA	CHINA	0,201619
	INDIA	0,197897

Nation (Customer): **ALGERIA**
Region (Customer): **AFRICA**
PercWithinRegion: **0,199082**

References

Practical
Tableau: 10…
Ryan Sleep…

Learning
Tableau 20…
Joshua N. …

Learning
Tableau
Joshua N. …

Visual
Analytics wi…
Alexander …

Tableau Your
Data! Fast …
Daniel G. M…

Mastering
Tableau 20…
Marleen Me…

Tableau For
Dummies
Pavol Socháň

Communic…
Data with T…
Ben Jones, …

Innovative
Tableau: 10…
Ryan Sleep…

Mastering
Tableau
David Bald…

The Big Book
of Dashboa…
Steve Wexl…

Mastering
Tableau 20…
Marleen Me…

Storytelling
With Data: …
Cole Nussb…

Exercises - Part I

Connecting to a source

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.

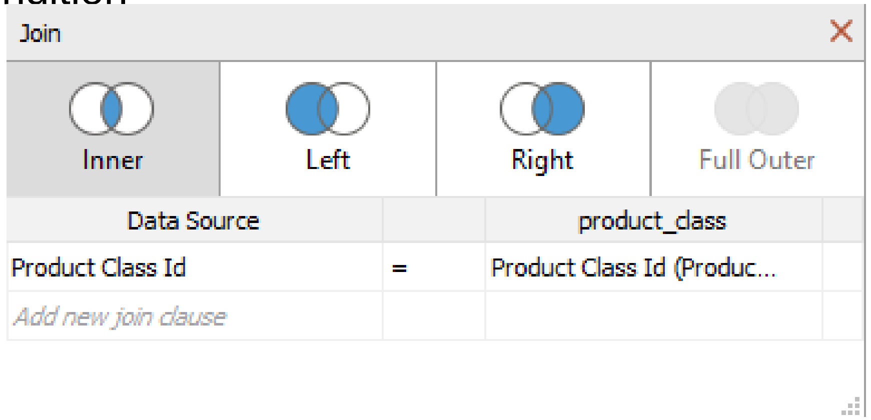
Tableau needs a *tabular view* of the data (i.e., a set of tables)

- Different tables from the same source can be linked through joins
- JSON data can be converted into a tablar view as well

Join

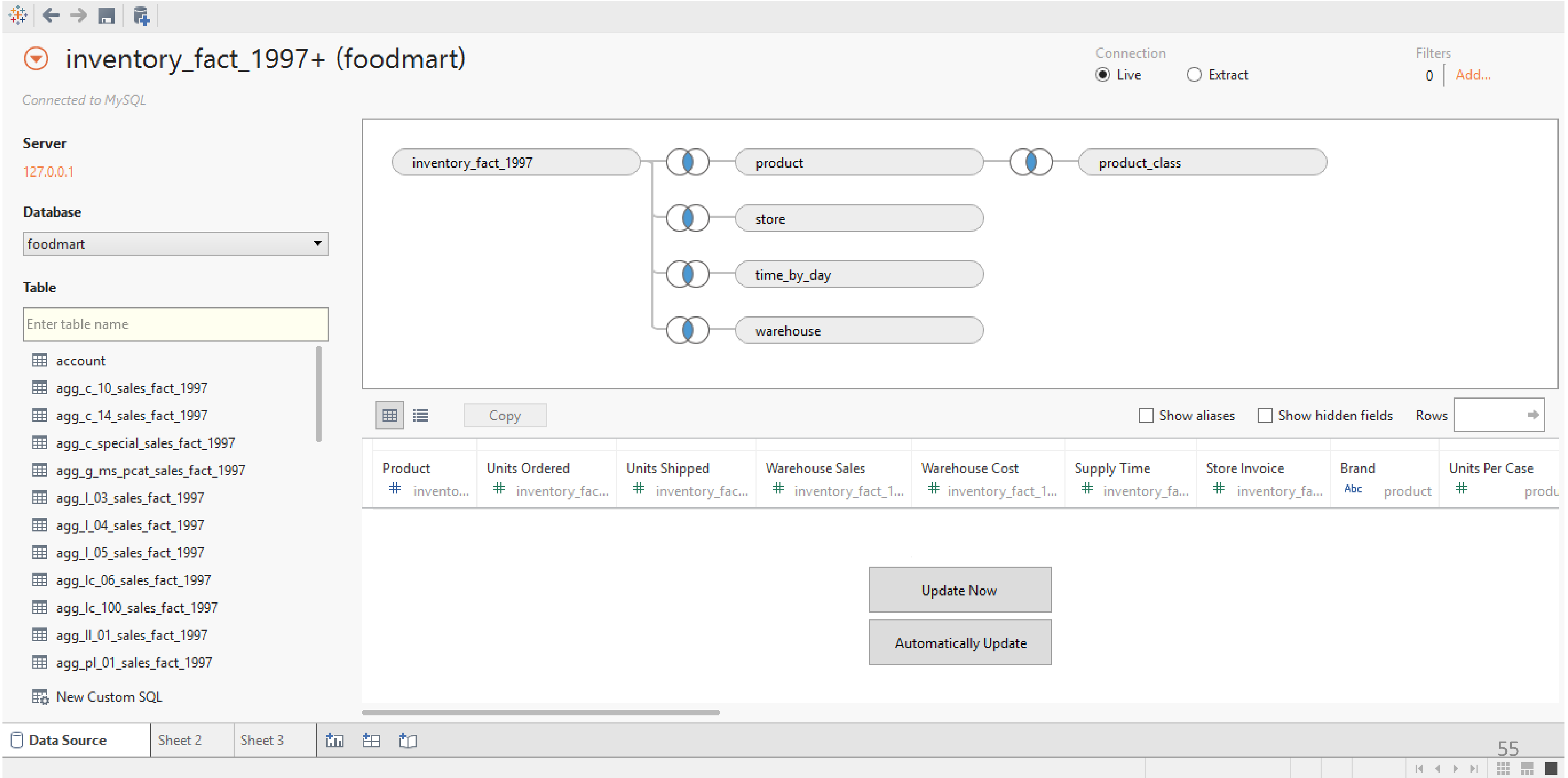
Tableau supports different types of join

- *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 joins



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

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

Copy

□ 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

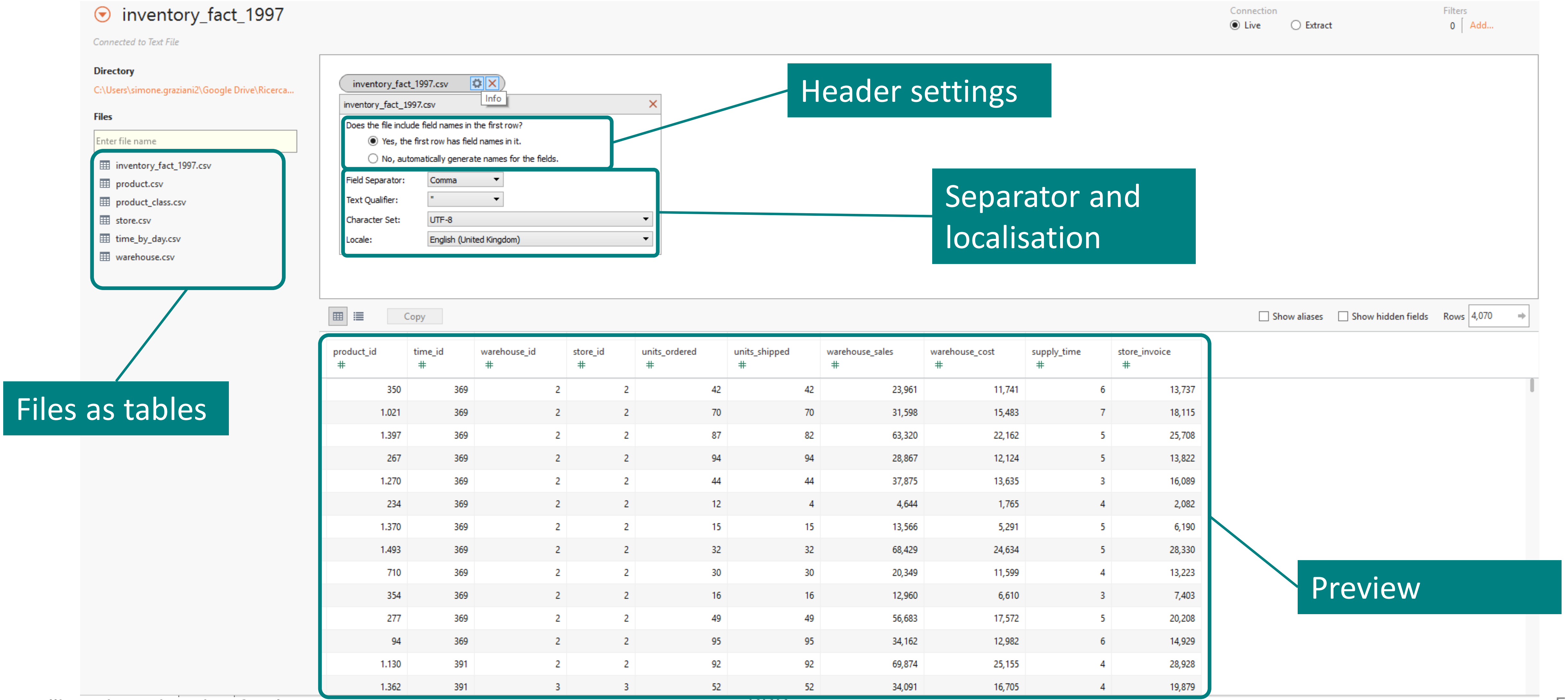
Update Now

Automatically Update

Data Source | Sheet 2 | Sheet 3

55

Example – Text File (DSV)



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

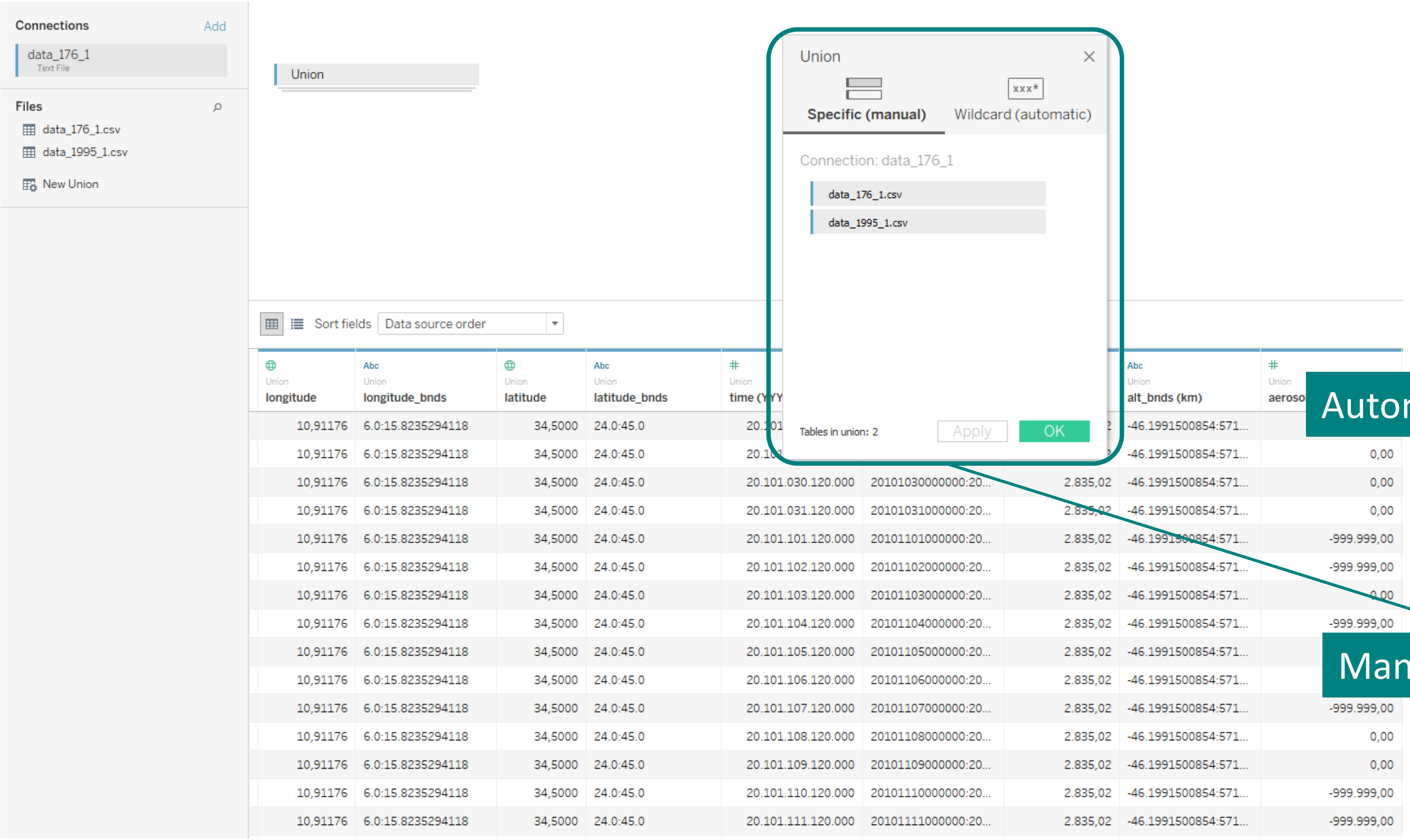
Header settings

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

Union	Union	Union	Union	Union			Union	Union
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

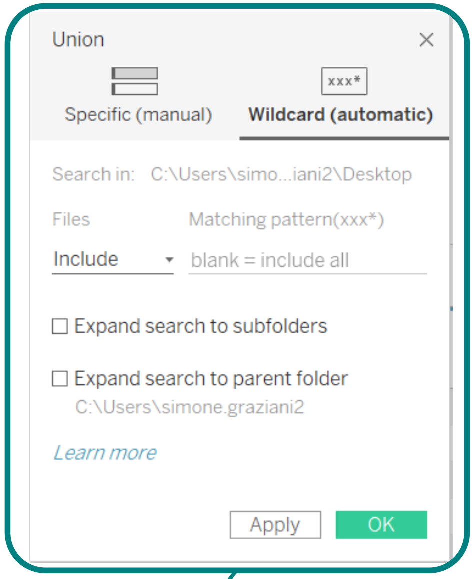
Connection: data_176_1

data_176_1.csv

data_1995_1.csv

Tables in union: 2

Manual selection



Union

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

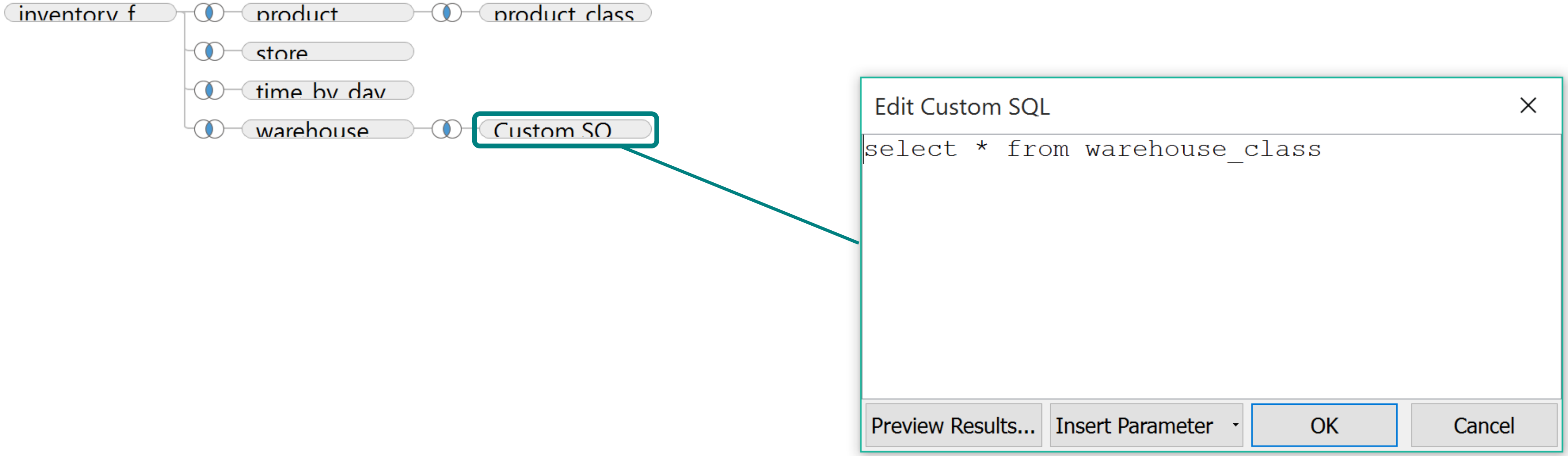
Learn more

Automatic selection

Custom SQL

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

Table

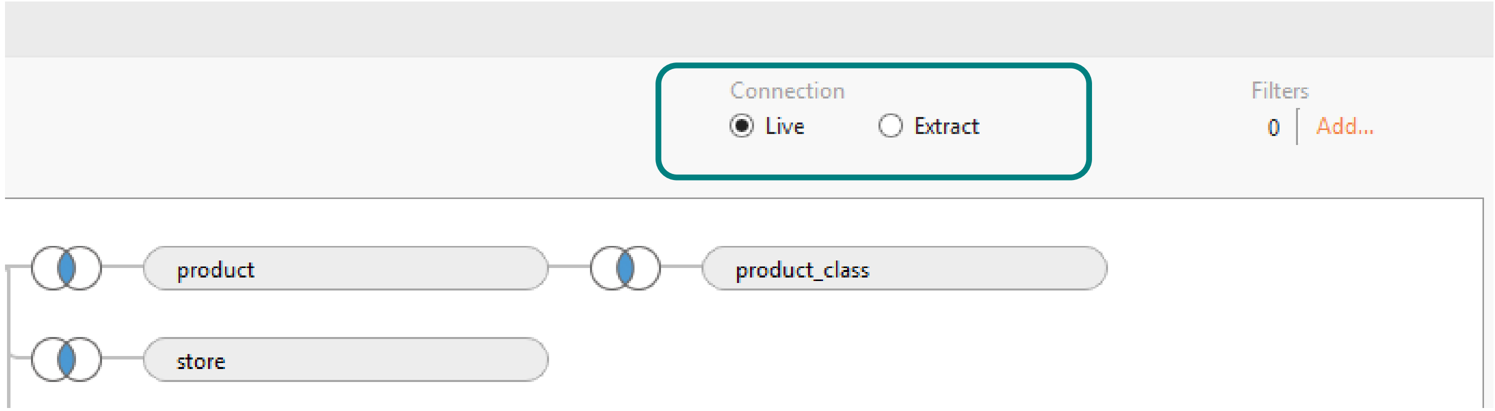


```
select * from warehouse_class
```

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

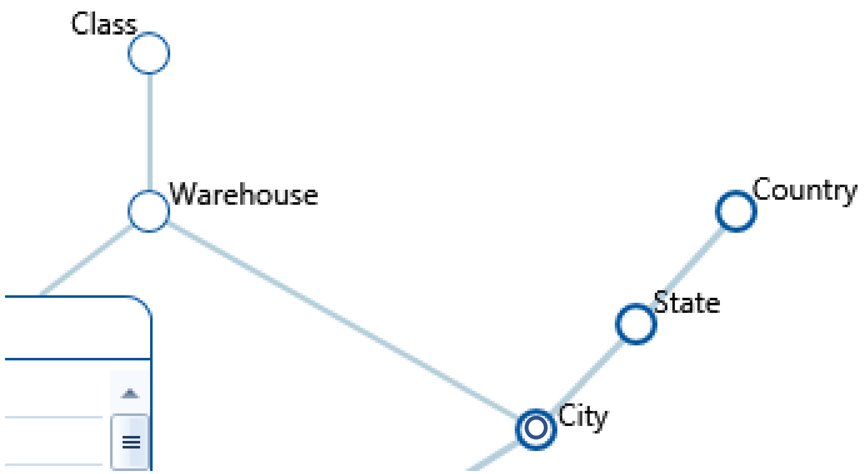
- The data source does not offer fast responses to analytical queries
- The data source must not be overloaded with analytical queries (e.g., an operational data store)
- Off-line data is needed (e.g., performing analyses without a connection to the data source)

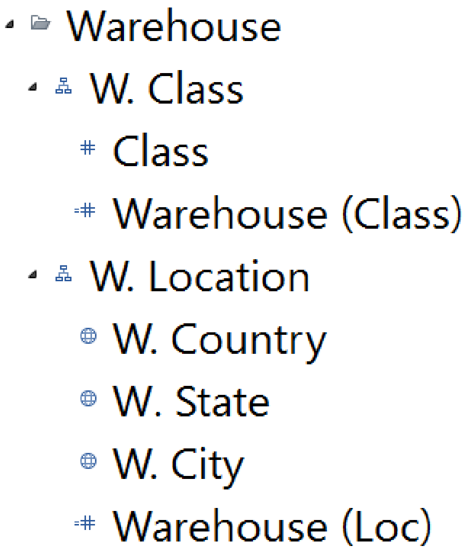
From the DFM to Tableau

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
- 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)

Exercises - Part II

First queries

Exercise 1

Visualise, through a bar chart, the total sum of *Store Sales* for each *S. State*

- Which one is the state with the highest sales?

Apply a drill-down operation to show the sales at the *S. City* level

- Are there cities whose sales are much lower than the others'?

How many stores (*Store*) are there in each *S. State*? In each *S. City*?

- Tip: Use the *COUNTD* aggregation
- Would it be reasonable to say that cities with fewer stores also have lower total sales?

Visualise the sales for each city (as done before) and represent the number of stores through the color property

Exercise 2

Given the last visualisation of Exercise 1, assign the *Type* field to the color property

- Can you notice any interesting pattern?

Visualise the total (SUM) sales for each *Type*

- Is the result surprising / expected?

Assign the number of stores to the properties color and label

Exercise 3

Visualise, through a line chart, the monthly sales trend

- Any interesting pattern?

Split the previous chart by *S. State* (i.e., an axis for each state)

- Does the previous pattern hold for each state?
- Tip: by default all axes have the same range, if you wish to change this behaviour: right click on any axis > *Edit Axis* > Select *Independent axis*…

Given the previous result, visualise the impact of each *Family* on the total sales while still showing the monthly trends

- Which type of visualisation is best suited?
- Tip: try to assign the *Family* field to one (or more) mark properties; try also different types of marks
- Tip: to change the type of mark use the drop-down menu on the *Marks* panel

Exercise 4

Visualise all stores and sort them in descending order by their total sales

Add the *Type* attribute to the previous visualisation

Add the number of customers and assign it to the color property

Visualise, in descending order, the total sales for each *Type* and *S. State* (put both fields on the rows)

- Verify that the correct order (by descending total sales) is applied

Exercise 5

Visualise the sales for each *Occupation* (*Customer* dimension) and exclude (i.e., filter out) the tuples whose *Store Sales* value is less than 5

- Tip: use a filter on *Store Sales*

Given the previous visualisation, apply another filter (in addition to the previous one) to exclude all occupations whose total sales is less than 80K

- There are still some occupations with sales less than 80K… Why?
  - Tip: to convert a regular filter into a *context filter*, right click > *Apply to Context*

Exercises - Part III

More queries

Exercise 6

Visualise the top ten customers by total sales

- Tip: a *Top N* filter can be appplied by drag-and-dropping a field into the filters shelf (as any other filter) and by using the *Top* tab (right-click > *Edit Filter…* > *Top*)

Given the previous visualisation, add the *Occupation* field (next to the *Customer* one)

Given the previous visualisation, filter by *Occupation* selecting the *Professional* value

- How many customers are there? Can you show the Top N customers whose occupation is *Professional*?
    - Tip: see Exercise 5

Exercise 7

Visualise the distribution of *Store Sales* values (i.e., how many receipts are there such that the total bill is between 0$ and 1$, between 1$ and 2$, etc.?)

- Tip: a histogram can be easily created from the *Show Me* window
- How can you describe the resulting distribution? Normal, uniform, skewed, …
- Which sales does the bin labeled as 0 contain? Does it include only sales with value = 0?
  - Tip: you can use the *View Data* feature

Without using the *Show Me* window, create a bar chart such that: the average *Store Cost* is on the rows shelf, while on the columns shelf there are the bins of *Store Sales* (bin size = 2)

- Tip: to create a bin, from the *Measures* pane, right click on *Store Sales* > *Create* > *Bins…*
- Does this chart clearly show a correlation between sales and cost? Why?
- Edit the chart so that the correlation (if it indeed exists) between sales and cost is more apparent

Exercise 8

Create a set with the Top 500 customers by total *Store Sales*

- Tip: to create a set, right click on *Customer* > *Create* > *Set*

Put the sum of sales on the columns shelf and the set of customers (created at the previous point) on the rows shelf

- What does this visualisation represent?

Edit the previous visualisation by moving the set of customers on the color property and adding *S. State* on the rows shelf

Exercise 9

Visualise the monthly trend of the total profits (Profit = Sales - Costs) for each *Type*

- Tip: to create a *Calculated Field*, from the main menu (on top), *Analysis* > *Create Calculated Field…*

Create a histogram of the number of sales by customers' age (create age bins of size = 5)

- Tip: function *DATEDIFF* returns the difference between two dates (for each function tableau has a brief description accessible directly from the calculated field editor)
- How can you describe the age distribution of the customers?

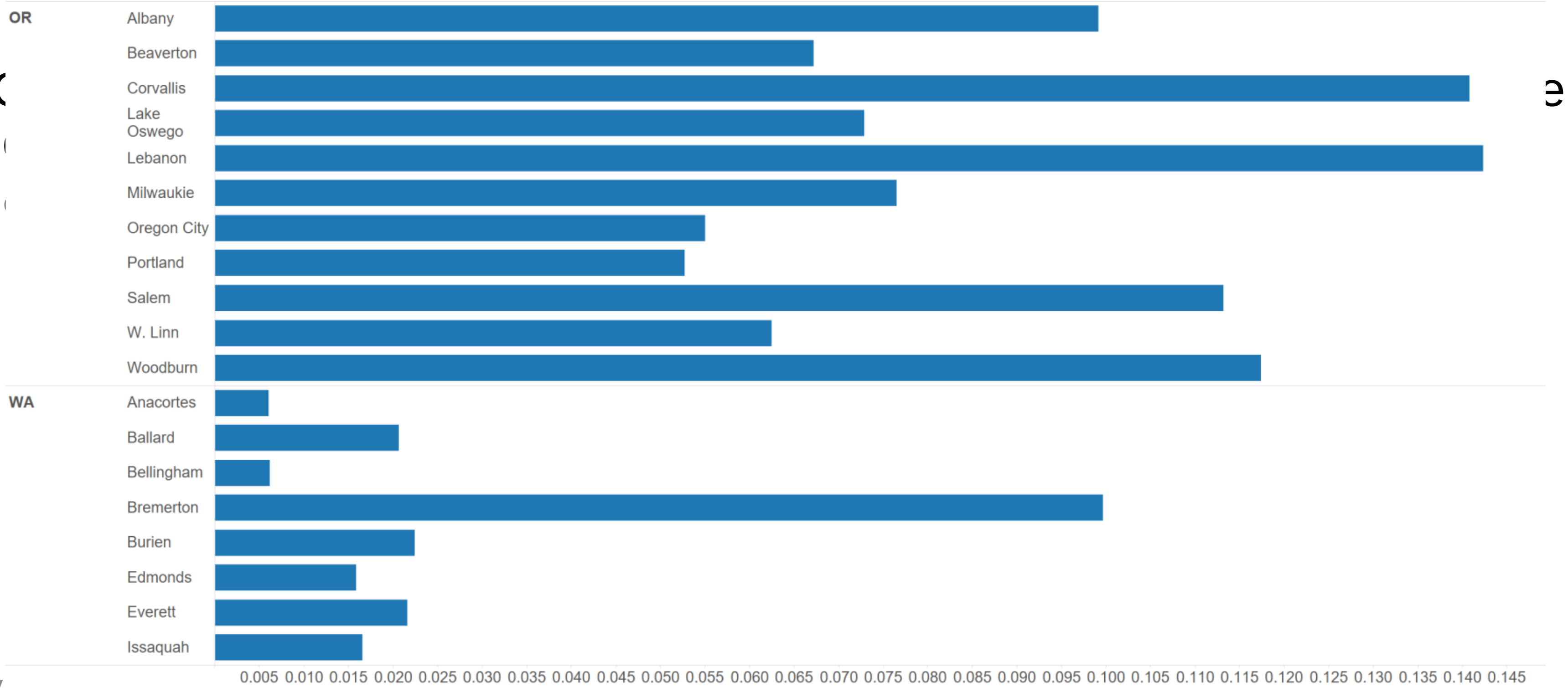
Given the previous visualisation, instead of the number of sales, visualise the number of sales over the number of customers (i.e., # sales / # customers)

- Any significant difference with the previous chart?

Exercise 10

For each (eir
correspon

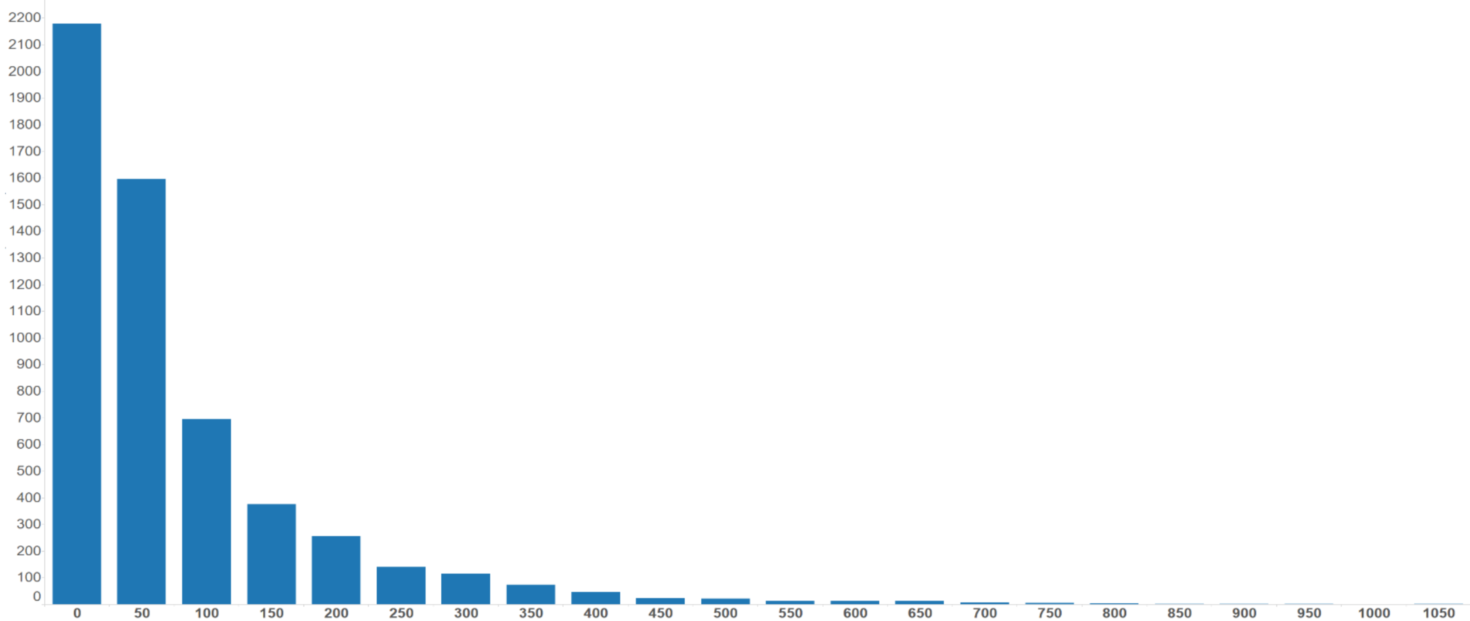
- Tip:



Exercise 11

Compute the distribution of customers over total sales, i.e., show how many customers are there such that the total sales of each one of them is in the ranges 0 – 49, 50 – 99, 100 – 149, …

- The result



2200
2100
2000
1900
1800
1700
1600
1500
1400
1300
1200
1100
1000
900
800
700
600
500
400
300
200
100
0
0
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
800
850
900
950
1000
1050