OLAP - Querying the DW

With Tableau Desktop

What is Tableau Desktop

Tableau Desktop is a commercial tool that is part of the Tableau suite

It provides a simple GUI to formulate queries on any kind of data source

- Not exactly an OLAP tool
- Besides advanced BI software (e.g., Oracle, SAP), similar commercial alternatives are PowerBI and Qlik
- Open-source alternatives
  - Saiku
  - Apache Superset
- Tableau Desktop is easy to use and provides appealing visualizations
  - Renewable one-year academic license is provided to all students and academic staff

What we are going to do

Guided exercises

- Data source inizialitation
- Multidimensional model setup
- OLAP basics
- Advanced querying options

Individual exercises

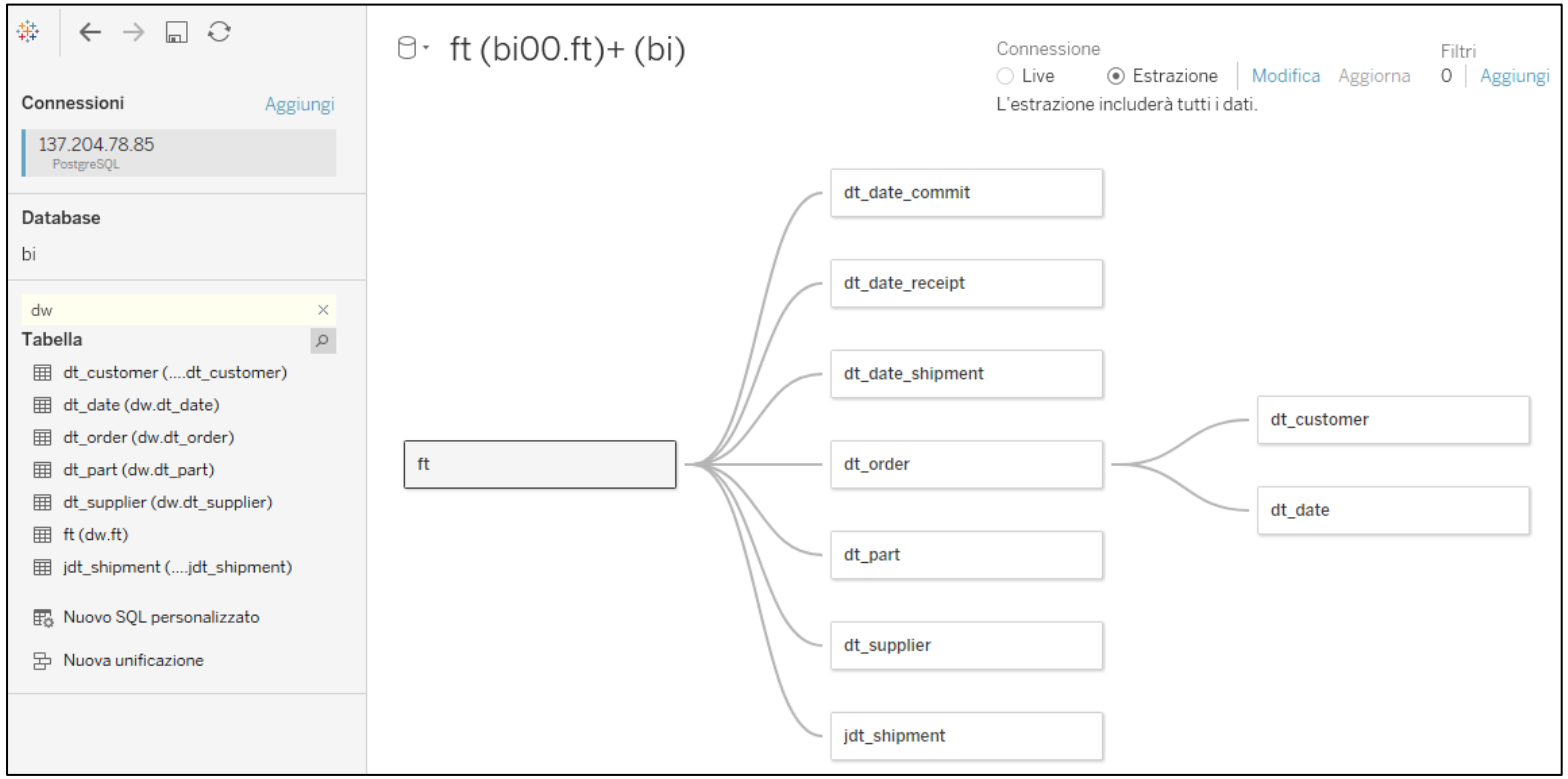
- Define OLAP queries from requirements in natural language

Data source initialization

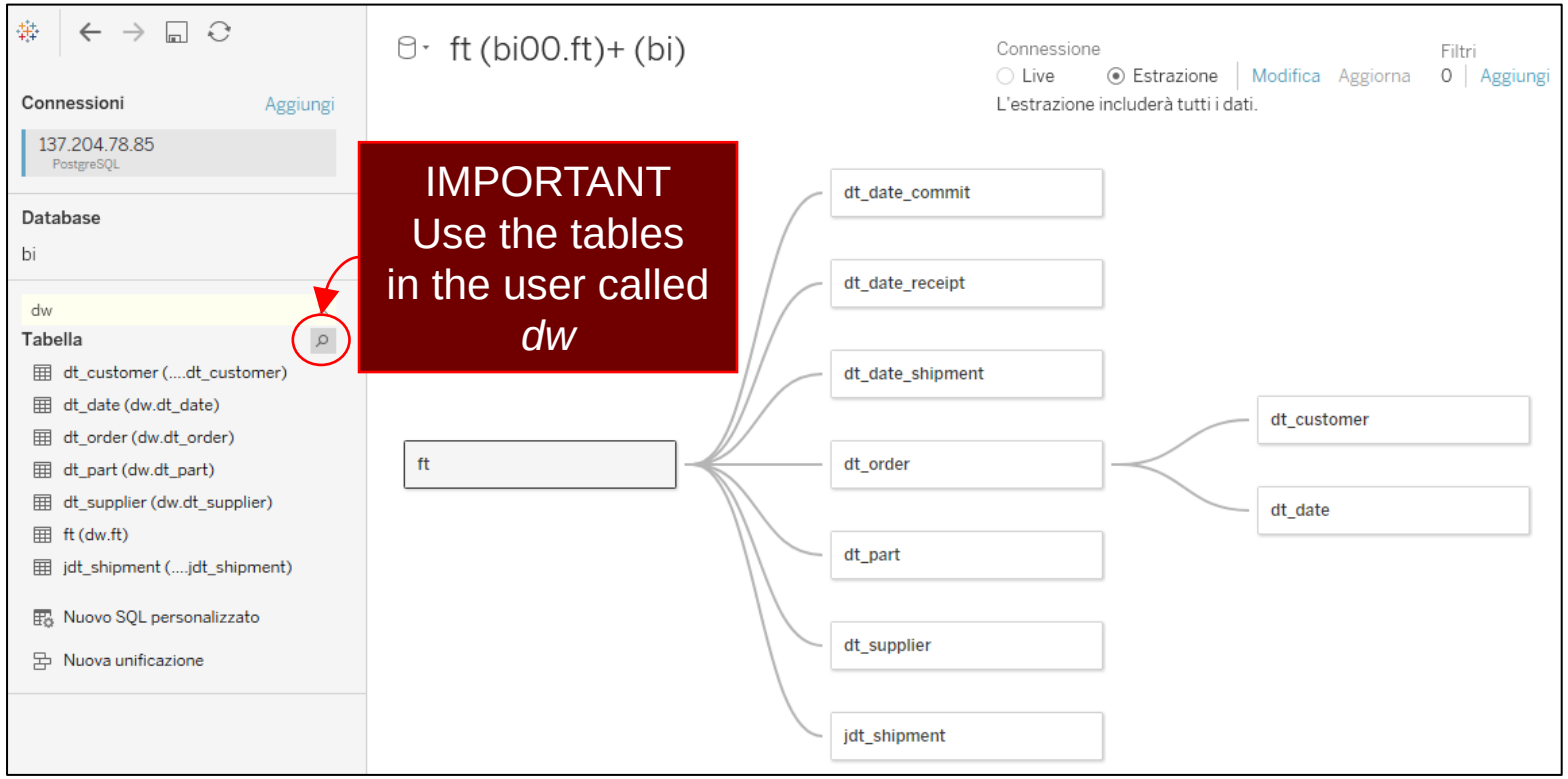
First, we must "tell" Tableau *which* data we want to query

- Connect to the database
- Drag & drop the tables
- Setup join connections
- Check fields
  - Hide unnecessary fields
  - Verify data types

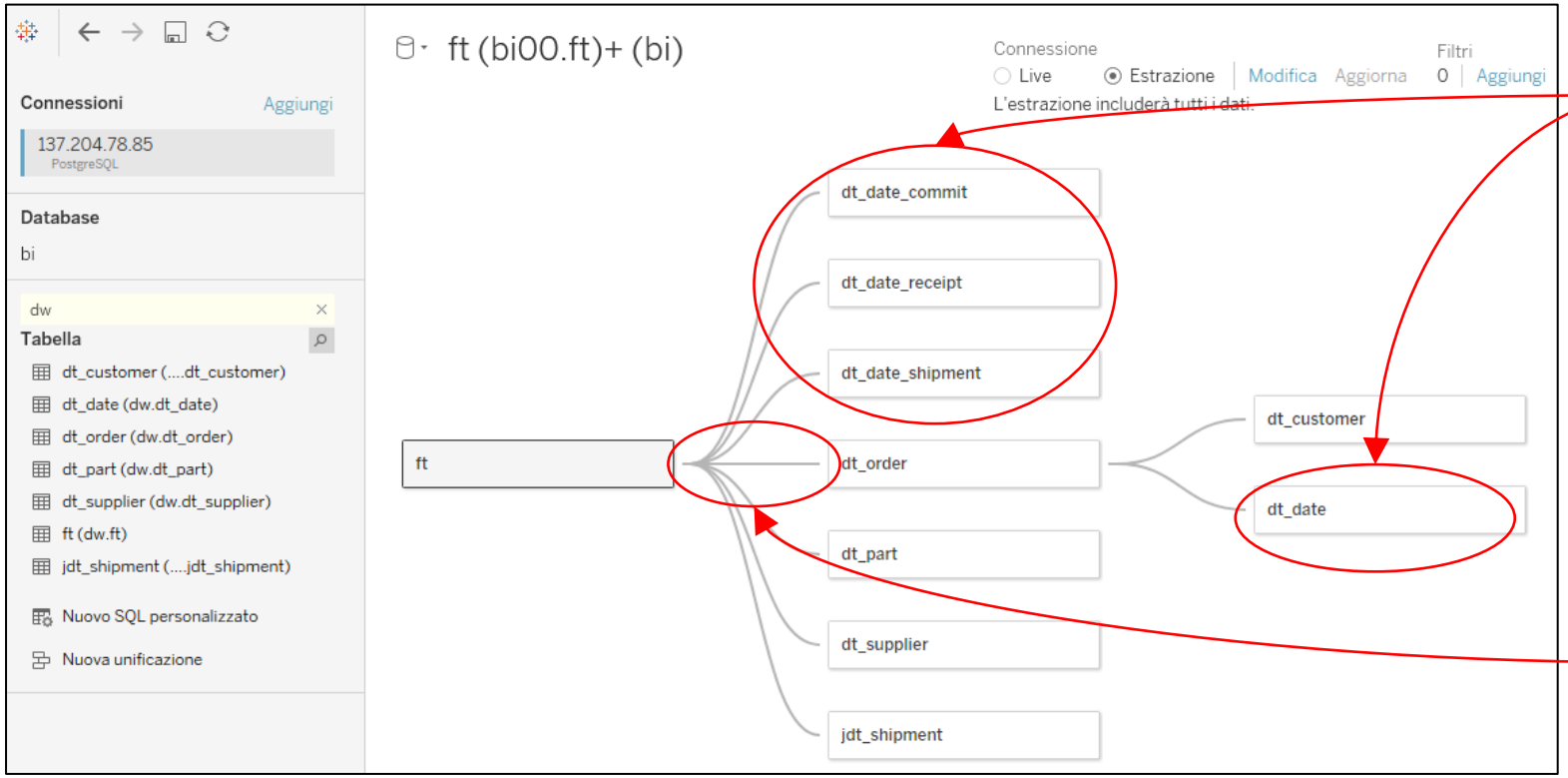
Data source initialization



Data source initialization



Data source initialization

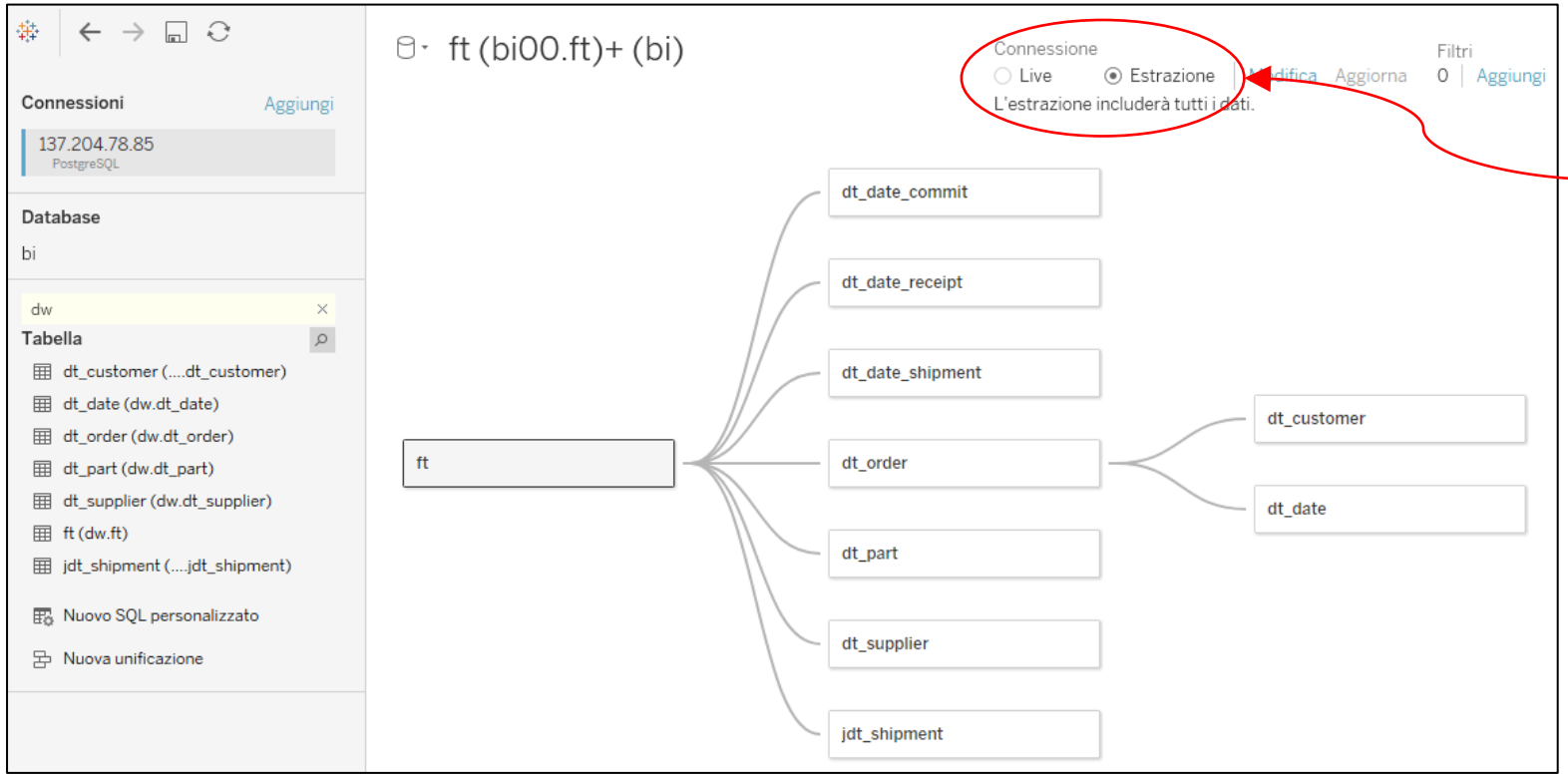


Shared hierarchy must be repeated

Links must be in the -to-one direction from left right

Click links to setup join predicates

Data source initialization



Extract the data (and save a snapshot on your machine) to improve efficiency

Multidimensional model setup

Tableau infers only few things

- Numeric fields are interpreted as measures, String/Date fields as dimensional attributes
- Hierarchies are not recognized (if not for few exceptions)

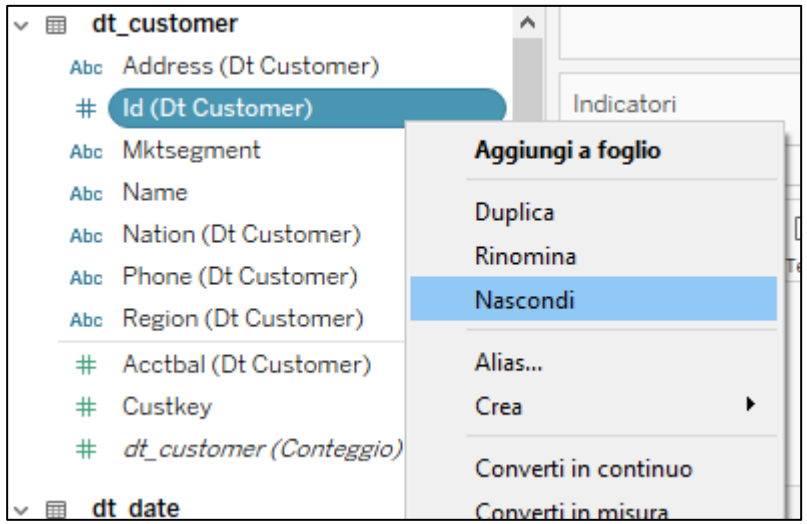
Most of the setup is manual

- Remove unnecessary fields (e.g., IDs)
- Correct measures and attributes
- Define hierarchies
- Extend hierarchies with new fields
  - E.g., binning, groups

Multidimensional model setup

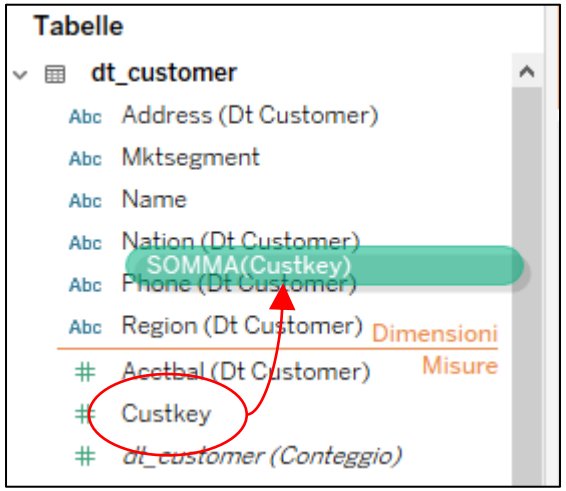
Remove unnecessary fields

- IDs and measures to count rows *(Conteggio)*



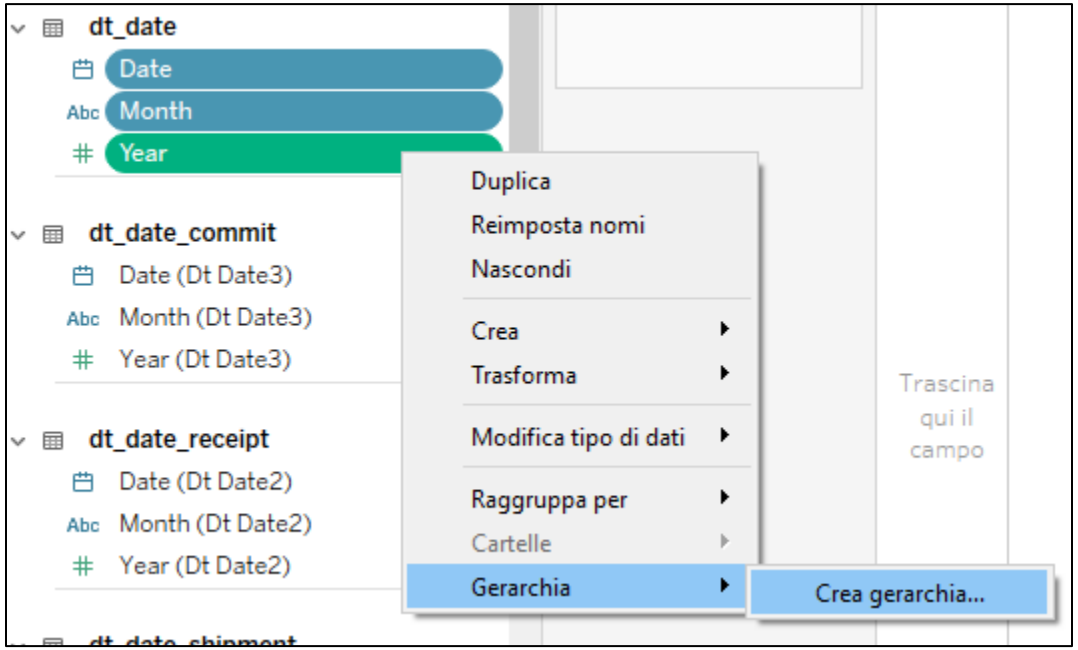
Fix measures and attributes

- Natural keys (e.g., *custkey*) are attributes
- Some numeric fields (e.g., *year*) are attributes



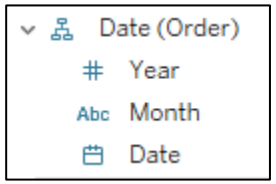
Multidimensional model setup

Define hierarchies



- Attributes in hierarchies must be listed always from the coarsest (top) to the finest (bottom) to ensure drill-down correctness

Wrong

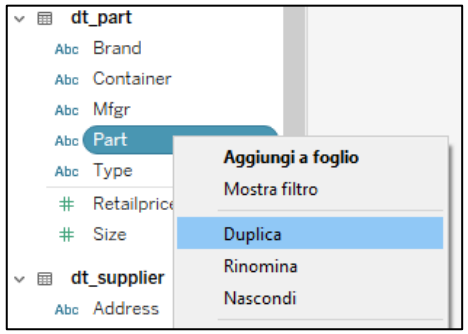


Correct

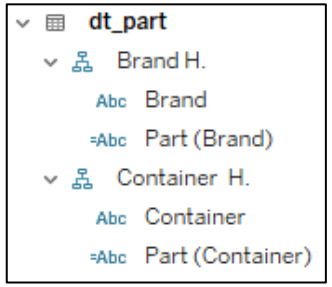
Multidimensional model setup

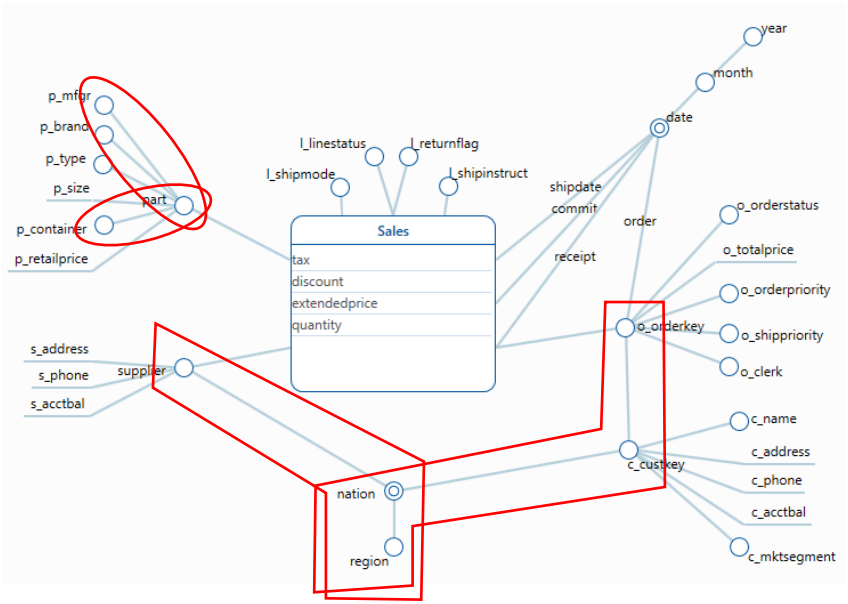
A hierarchy must be defined for *every* linear root-to-leaf path

- Attributes involved in branches or shared hierarchies must be replicated for every existing linear path



- Name hierarchies after the coarsest attribute
- For replicated attributes, put the hierarchy name between parenthesis

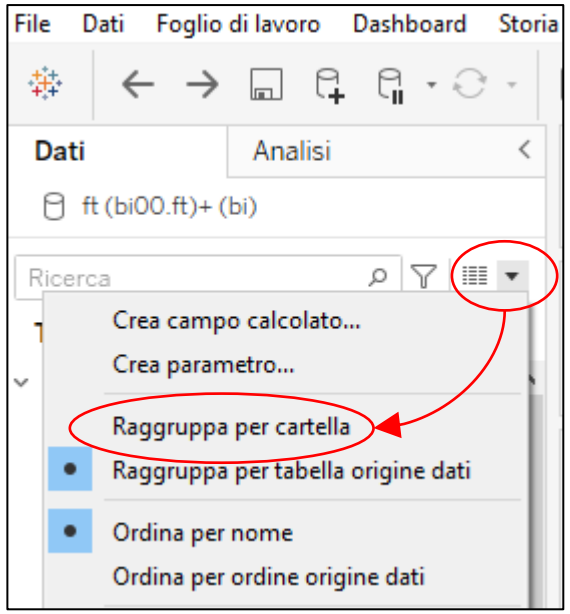


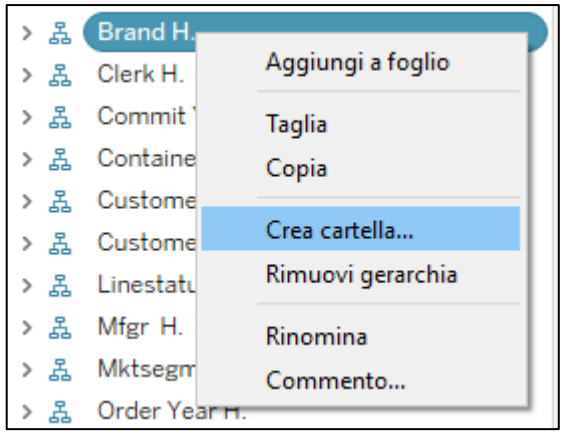


Multidimensional model setup

Organize hierarchies into folders

- One folder per dimension

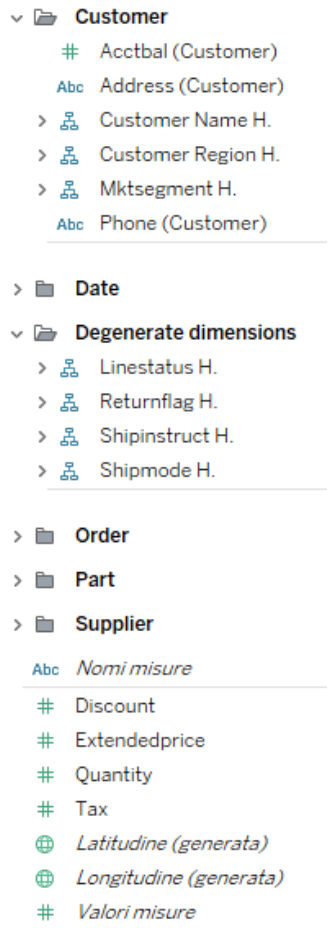




Multidimensional model setup

Organize hierarchies into folders

- One folder per dimension
- Include descriptive attributes in the folders
  - Descriptive attributes must NOT be in hierarchies
  - Numeric descriptive attributes should be continuous (green icon)
- Degenerate dimensions can be collapsed into a single folder
- Verify which measures are left at the bottom



Exercise 3

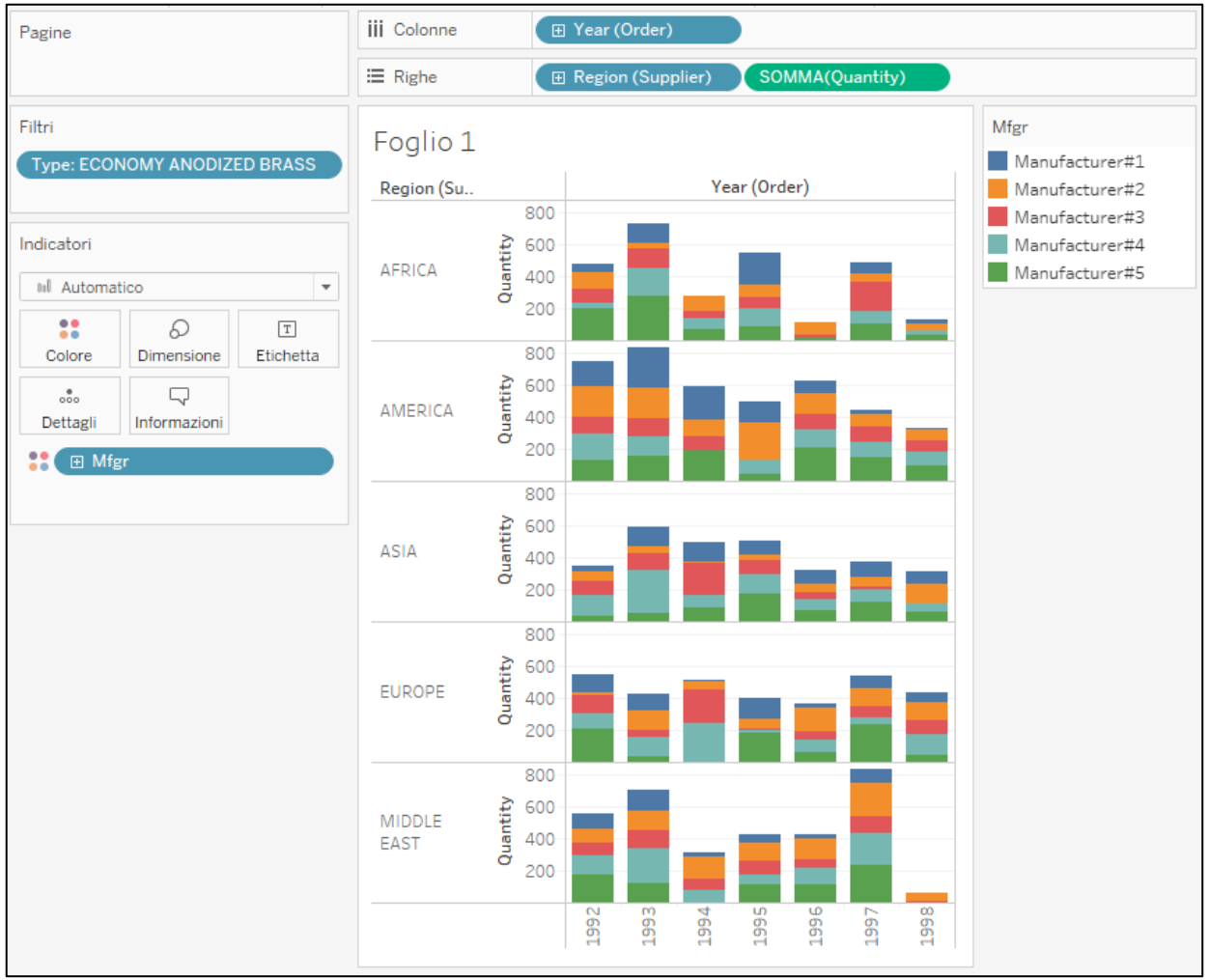
Complete the multidimensional setup for the Sales cube

Do the same for the Orders cube

OLAP basics

Queries are formulated by drag&dropping attributes and measures onto rows, columns, filters, and marks

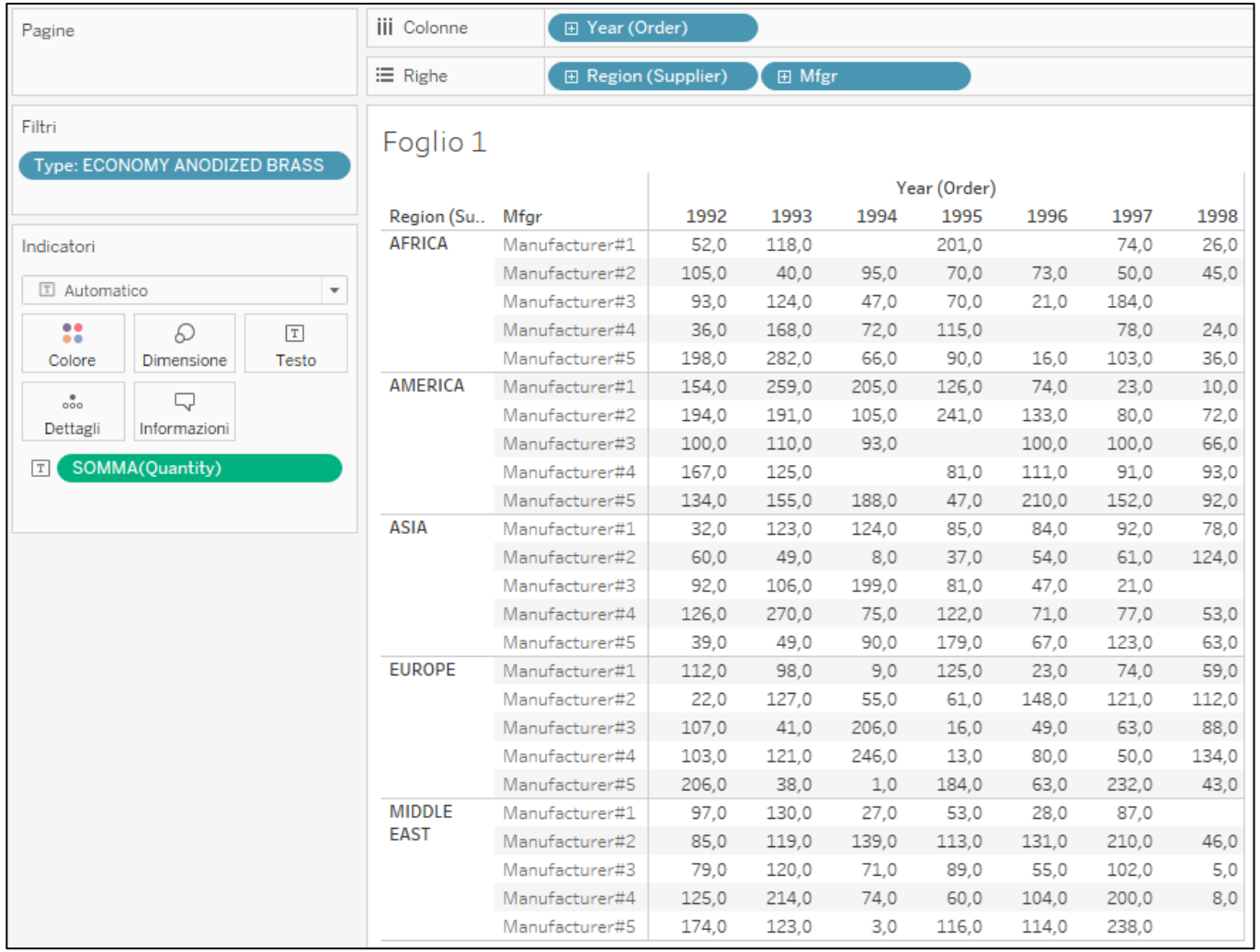
- SELECT d.year, s.region, p.mfgr, sum(ft.quantity)
  FROM ft, dt_part p, dt_supp s, dt_order o, dt_date d
  WHERE ft.idpart = p.id
  AND ft.idsupp = s.id
  AND ft.idorder = o.id
  AND o.iddate = d.id
  AND p.type = 'ECONOMY ...'
  GROUP BY d.year, s.region, p.mfgr



OLAP basics

The same query can be issued in different ways, where only the visualization changes (but the data is the same)

- SELECT d.year, s.region, p.mfgr, sum(ft.quantity)
  FROM ft, dt_part p, dt_supp s, dt_order o, dt_date d
  WHERE ft.idpart = p.id
  AND ft.idsupp = s.id
  AND ft.idorder = o.id
  AND o.iddate = d.id
  AND p.type = 'ECONOMY ...'
  GROUP BY d.year, s.region, p.mfgr



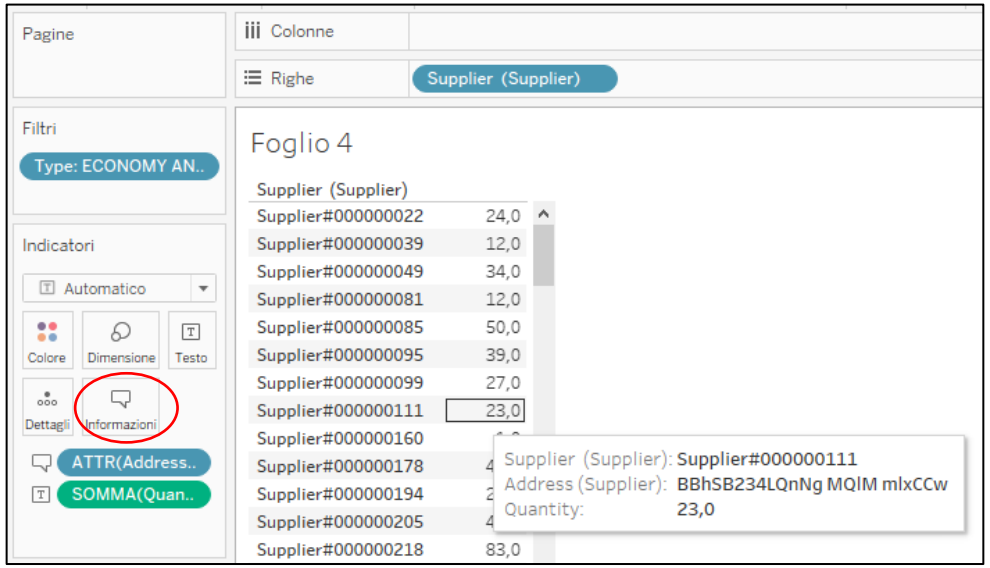
Foglio 1

Region (Su..	Mfgr	Year (Order) 1992	1993	1994	1995	1996	1997	1998
AFRICA	Manufacturer#1	52,0	118,0		201,0		74,0	26,0
	Manufacturer#2	105,0	40,0	95,0	70,0	73,0	50,0	45,0
	Manufacturer#3	93,0	124,0	47,0	70,0	21,0	184,0	
	Manufacturer#4	36,0	168,0	72,0	115,0		78,0	24,0
	Manufacturer#5	198,0	282,0	66,0	90,0	16,0	103,0	36,0
AMERICA	Manufacturer#1	154,0	259,0	205,0	126,0	74,0	23,0	10,0
	Manufacturer#2	194,0	191,0	105,0	241,0	133,0	80,0	72,0
	Manufacturer#3	100,0	110,0	93,0		100,0	100,0	66,0
	Manufacturer#4	167,0	125,0		81,0	111,0	91,0	93,0
	Manufacturer#5	134,0	155,0	188,0	47,0	210,0	152,0	92,0
ASIA	Manufacturer#1	32,0	123,0	124,0	85,0	84,0	92,0	78,0
	Manufacturer#2	60,0	49,0	8,0	37,0	54,0	61,0	124,0
	Manufacturer#3	92,0	106,0	199,0	81,0	47,0	21,0	
	Manufacturer#4	126,0	270,0	75,0	122,0	71,0	77,0	53,0
	Manufacturer#5	39,0	49,0	90,0	179,0	67,0	123,0	63,0
EUROPE	Manufacturer#1	112,0	98,0	9,0	125,0	23,0	74,0	59,0
	Manufacturer#2	22,0	127,0	55,0	61,0	148,0	121,0	112,0
	Manufacturer#3	107,0	41,0	206,0	16,0	49,0	63,0	88,0
	Manufacturer#4	103,0	121,0	246,0	13,0	80,0	50,0	134,0
	Manufacturer#5	206,0	38,0	1,0	184,0	63,0	232,0	43,0
MIDDLE EAST	Manufacturer#1	97,0	130,0	27,0	53,0	28,0	87,0	
	Manufacturer#2	85,0	119,0	139,0	113,0	131,0	210,0	46,0
	Manufacturer#3	79,0	120,0	71,0	89,0	55,0	102,0	5,0
	Manufacturer#4	125,0	214,0	74,0	60,0	104,0	200,0	8,0
	Manufacturer#5	174,0	123,0	3,0	116,0	114,0	238,0	

OLAP basics

Descriptive attributes should never be put on rows, columns

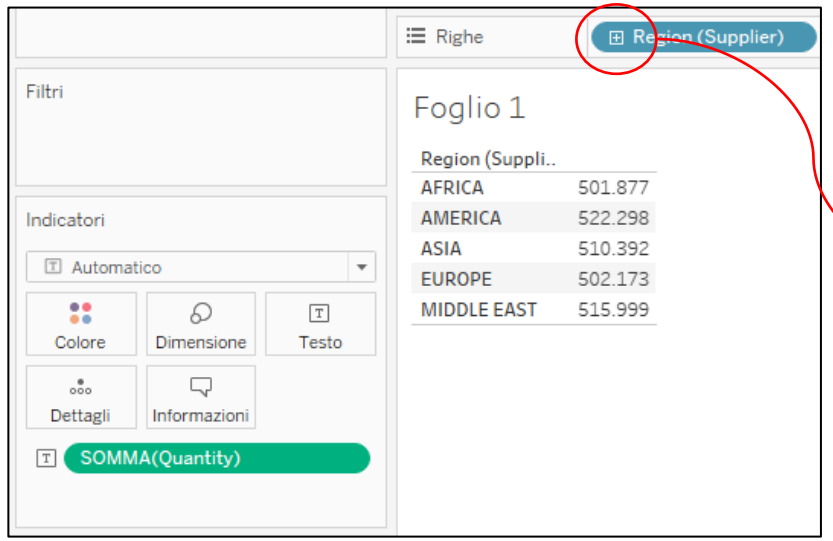
- Only as additional information, that becomes visible upon cell hovering

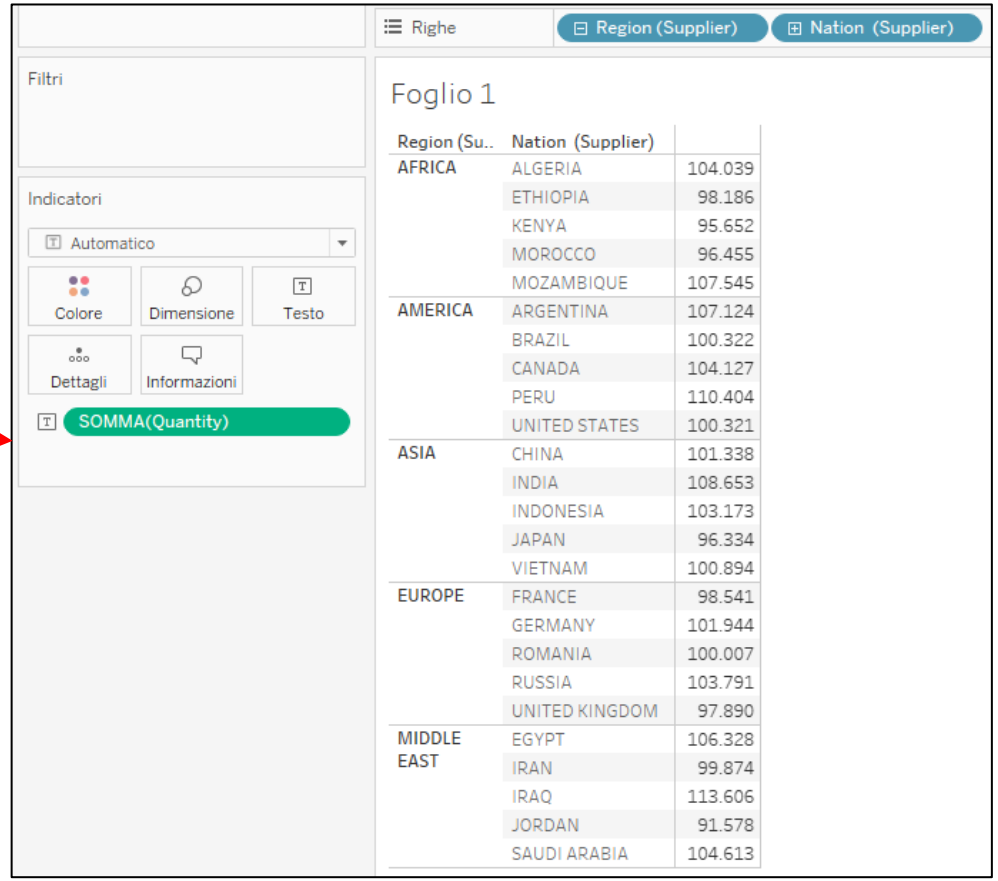


OLAP basics

Drill-down: just hit the (+) button

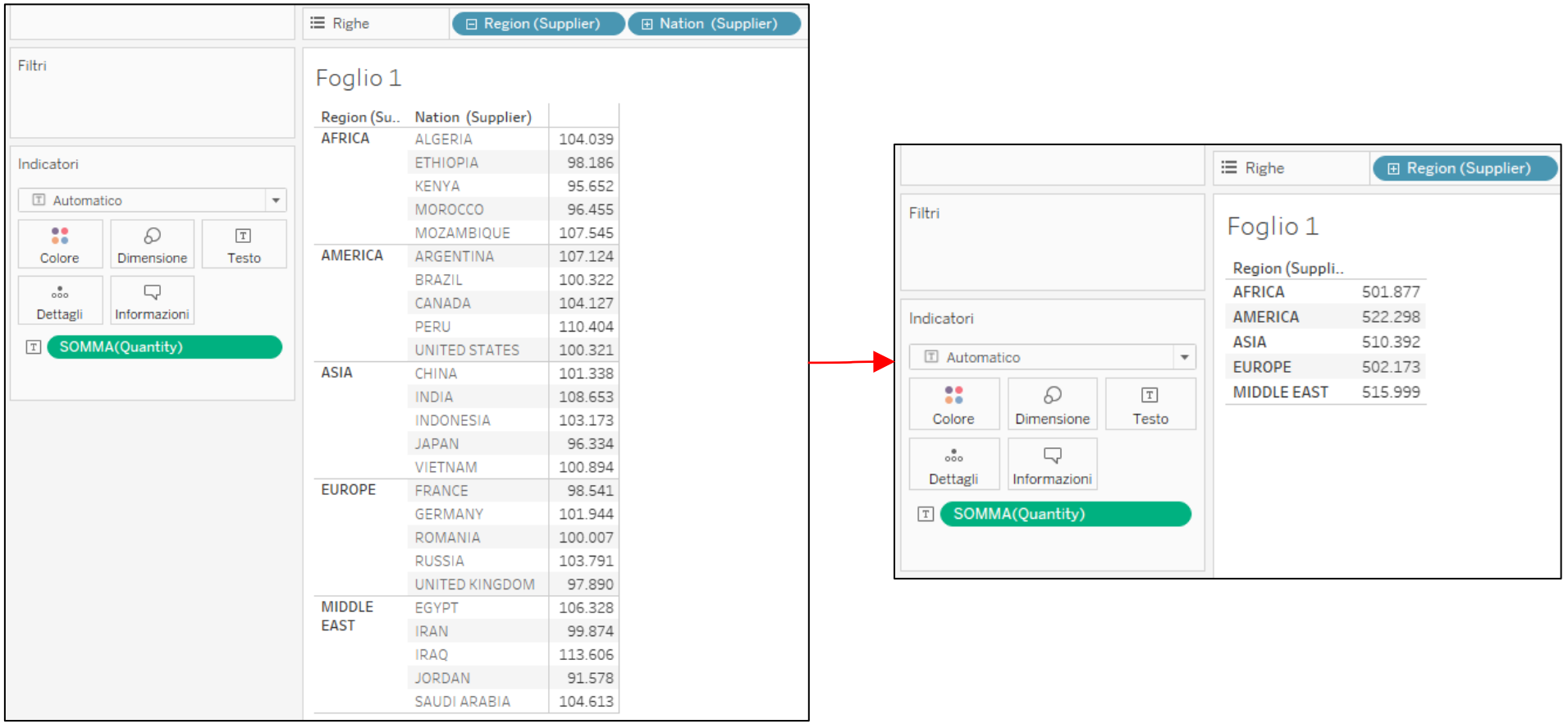
- Beware: a wrong hierarchy setup leads to hardly readable results





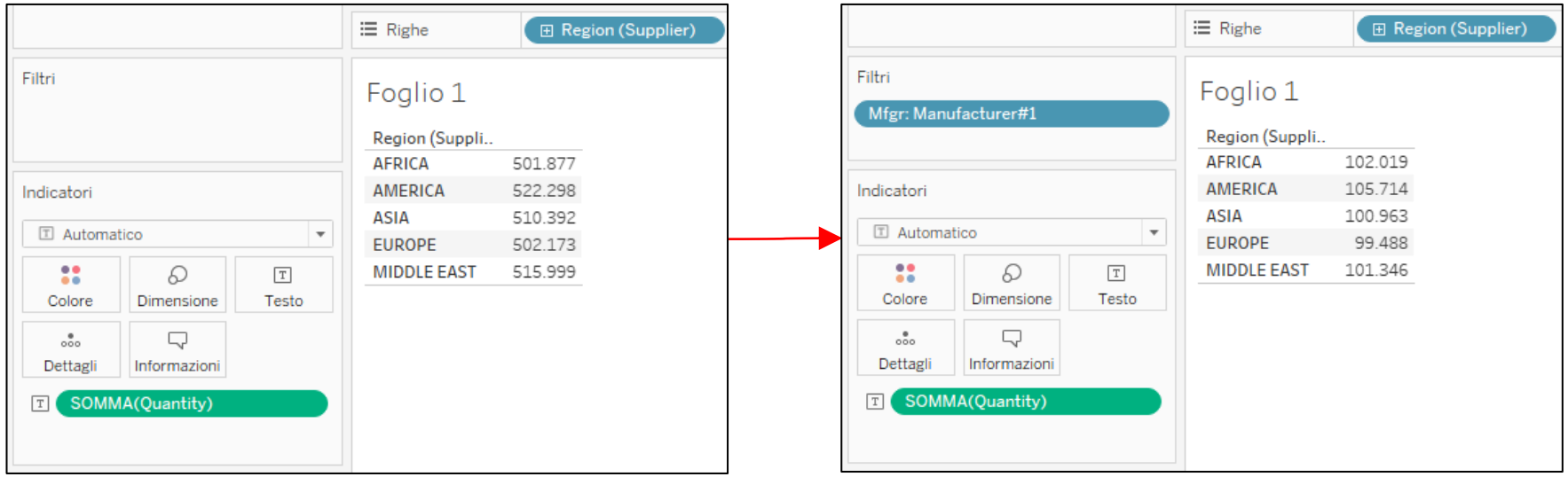
OLAP basics

Roll-up: either remove the finer attribute or replace it with a coarser one



OLAP basics

Slice: either add an attribute/measure in the filters panel or right-click a pre-existing attribute/measure to add a filter



OLAP basics

Slice: either add an attribute/measure in the filters panel or right-click a pre-existing attribute/measure to add a filter

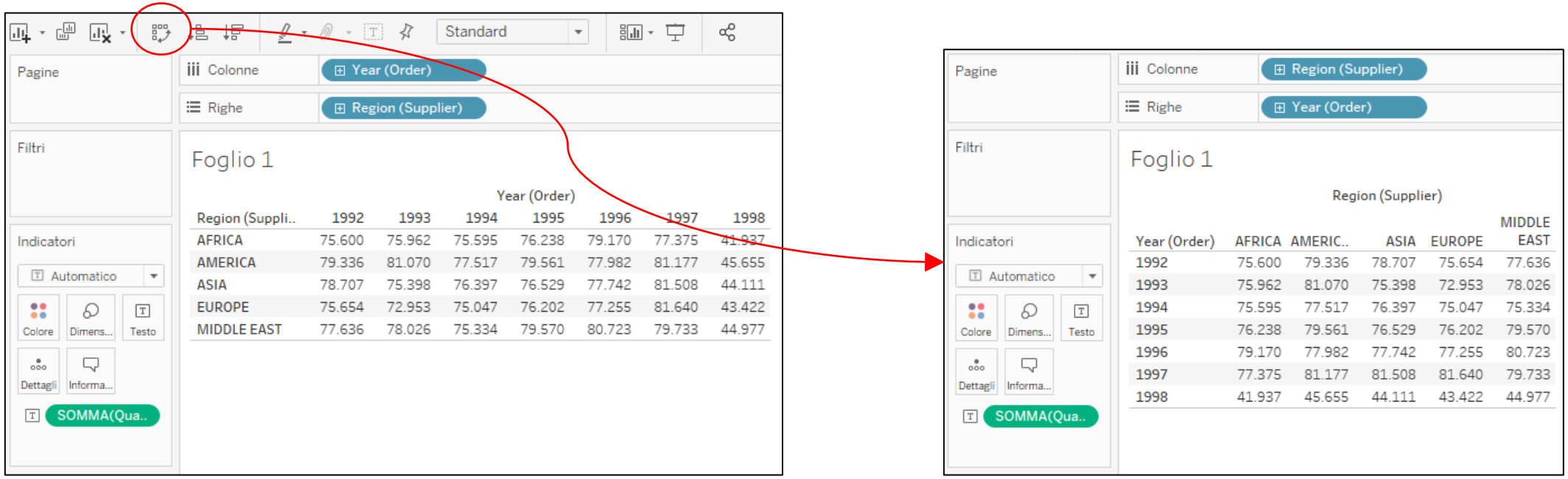
Filters depend on the field type

- Discrete fields: one or more categorical values can be chosen
- Continuous fields (either attributes ore measures), a range can be specified and a different filtering level can be chosen
  - Pre-aggregation; e.g., "SELECT [..] *WHERE retailprice > 1000* [..]"

  - Post-aggregation; e.g., "SELECT [..] GROUP BY [..] *HAVING AVG(retailprice) > 1000*"

OLAP basics

Pivot: just hit the pivot button



OLAP basics

Ordering: right click on an attribute > Order

- In case of multiple criteria, each attribute must be set independently
- There are different ordering types
  - Alphabetical (intuitive)
  - Manual (intuitive)
  - Data Source Order: the order is the same as on the datasource
  - Computed: the order is based on a computation (e.g., the sum of specific measure)
  - Nested: useful in the presence of two (or more) independent fields

Region (S..	
AMERICA	442.962
MIDDLE EAST	438.363
ASIA	431.685
EUROPE	426.519
AFRICA	426.277

Computed order
Regions are ordered based on their global evaluation
(the same ordering applies in every year)

Year (Order)	Region (S..	
1993	AMERICA	81.070
	MIDDLE EAST	78.026
	ASIA	75.398
	EUROPE	72.953
	AFRICA	75.962
1994	AMERICA	77.517
	MIDDLE EAST	75.334
	ASIA	76.397
	EUROPE	75.047
	AFRICA	75.595

Year (Order)	Region (S..	
1993	AMERICA	81.070
	MIDDLE EAST	78.026
	AFRICA	75.962
	ASIA	75.398
	EUROPE	72.953
1994	AMERICA	77.517
	ASIA	76.397
	AFRICA	75.595
	MIDDLE EAST	75.334
	EUROPE	75.047

Nested order
Each year has its own ordering, based on actual cell values

More functionalities: visualization

Marks

- Play around with marks to understand how they work
- Each chart can have multiple marks

Show me panel (top-right corner)

- Useful to get suggestions and to generate special charts (e.g., boxplots)

Analysis > View data

- Useful to export results and to check the original values before the aggregation

Analysis > Grand totals

- Activates totals by rows and/or columns

More functionalities: new fields

Right-click on an attribute > Create > ...

- > Group
  - Manually take some members from an attribute and put them in a new attribute
- > Set
  - Like groups, but based on dynamic criteria
- > Set > TopN
  - A special *set* to select the top/bottom members based on some criteria
  - E.g., the customers who spent the most
- > Bin
  - Useful to discretize numerical descriptive attributes (e.g., the account balance of customers)
  - Support only for equi-width binning
- > Calculated fields
  - Create a new field based on a custom formula

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

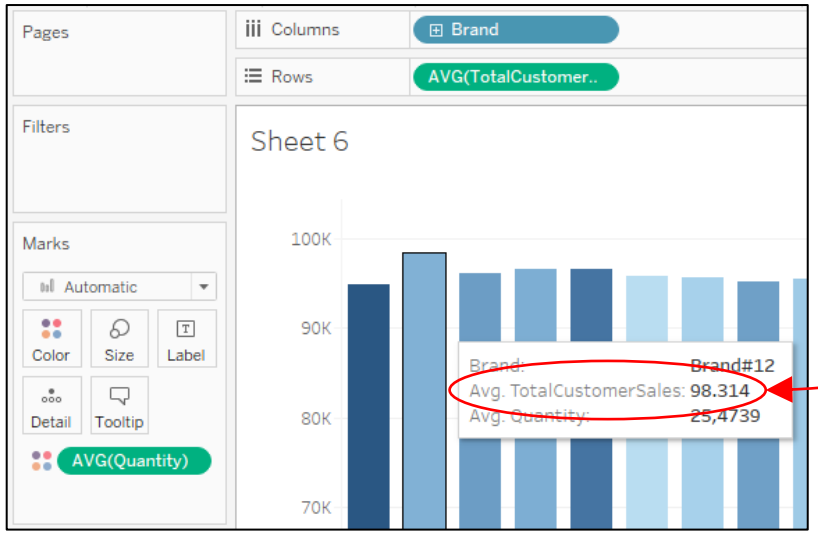
- Define reference aggregated values
- Useful to create more advanced queries combining more aggregations

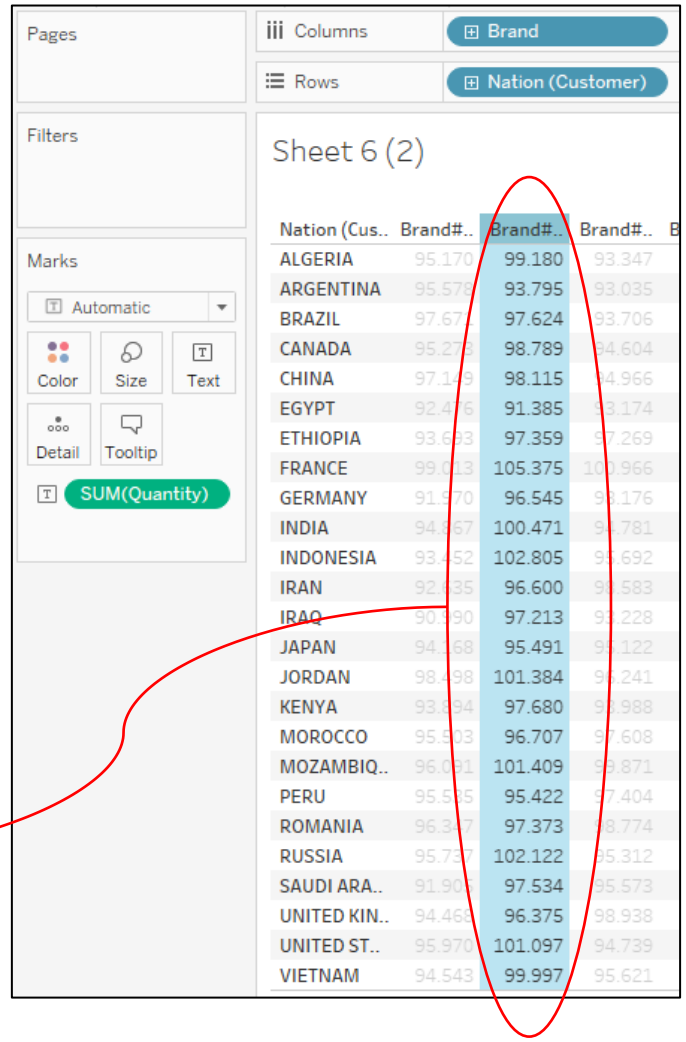
More functionalities: LOD fields

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



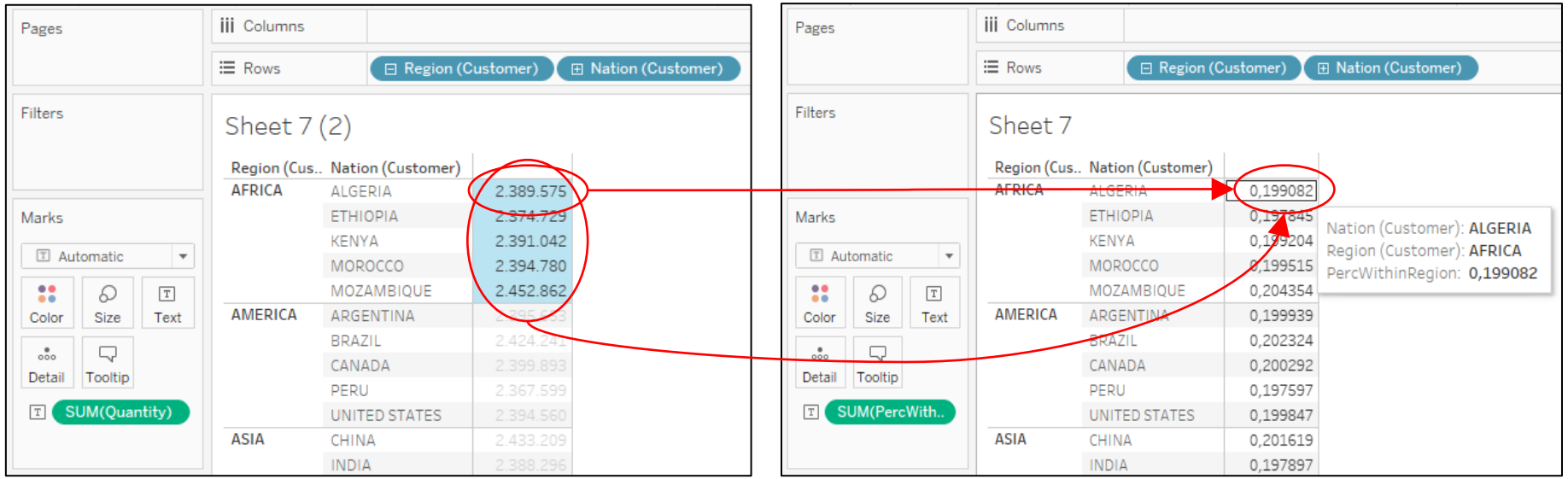


More functionalities: LOD fields

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



More functionalities: LOD fields

Level keywords

- FIXED: aggregates by all-and-only the specified attributes
- INCLUDE: aggregates by the specified attributes + those in columns/rows/marks
- EXCLUDE: aggregates by the attributes in columns/rows/marks - the specified ones
- (no keyword): aggregates all facts

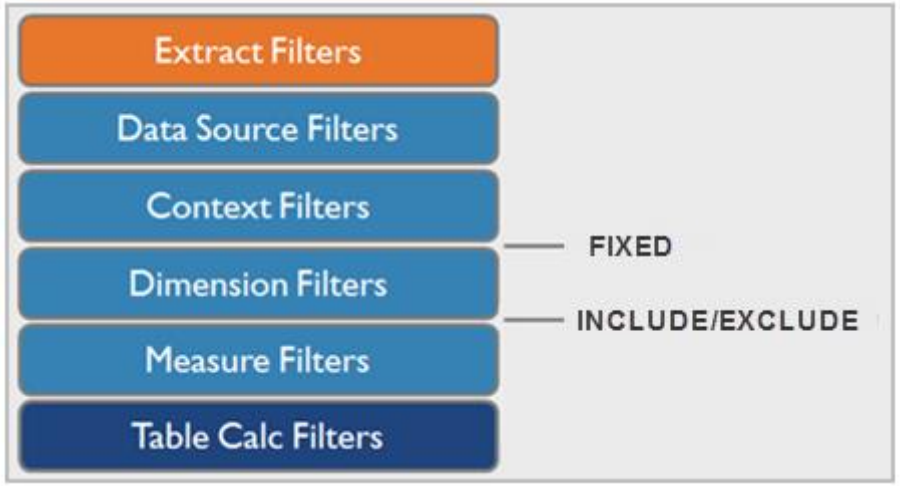
Check the documentation for further info:

- https://help.tableau.com/current/pro/desktop/en-gb/calculations_calculatedfields_lod.htm#include

Order of filter applications

Different types of filter are applied in different order (top-down)

- Use this table as a reference



Exercise 4

Simulate on Tableau the following OLAP sessions

- Produce a new worksheet for every bullet point marked with [W]

Exercise 4.1

- [W] Check the quantity of items sold by supplier nation
  - Are there some nations performing better/worse than the others?
- [W] Drill-down on the suppliers
  - Nations ordered alphabetically, suppliers by decreasing quantities sold
- Isolate the worst 100 suppliers
- [W] Show how the worst 100 are distributed across nations

Exercise 4.2

- Define a *sales* measure as follows
  - Sales = Quantity * Extendedprice * (1-Discount) * (1+Tax)
- [W] Visualise yearly sales trend
  - [W] Explain the drop in 1998 with a slice and drill
- [W] Visualise monthly sales trends for every year
  - Use the Tableau's date functions on the Date attribute instead of the Month attribute
  - Help yourself with the "Show me" panel to produce trend lines
  - Is there any interesting pattern?

Exercise 4.3

- [W] Visualize the sum of Sales by Mktsegment in July 1997 through a bar chart
- [W] Add the average Sales as color
  - Average sales in Furniture are higher despite the sum being lower
  - [W] Add a label that explains this

Exercise 5

Produce OLAP queries using advanced functionalities

- Main goal: answering the given query
- Secondary goal: reproducing the given chart

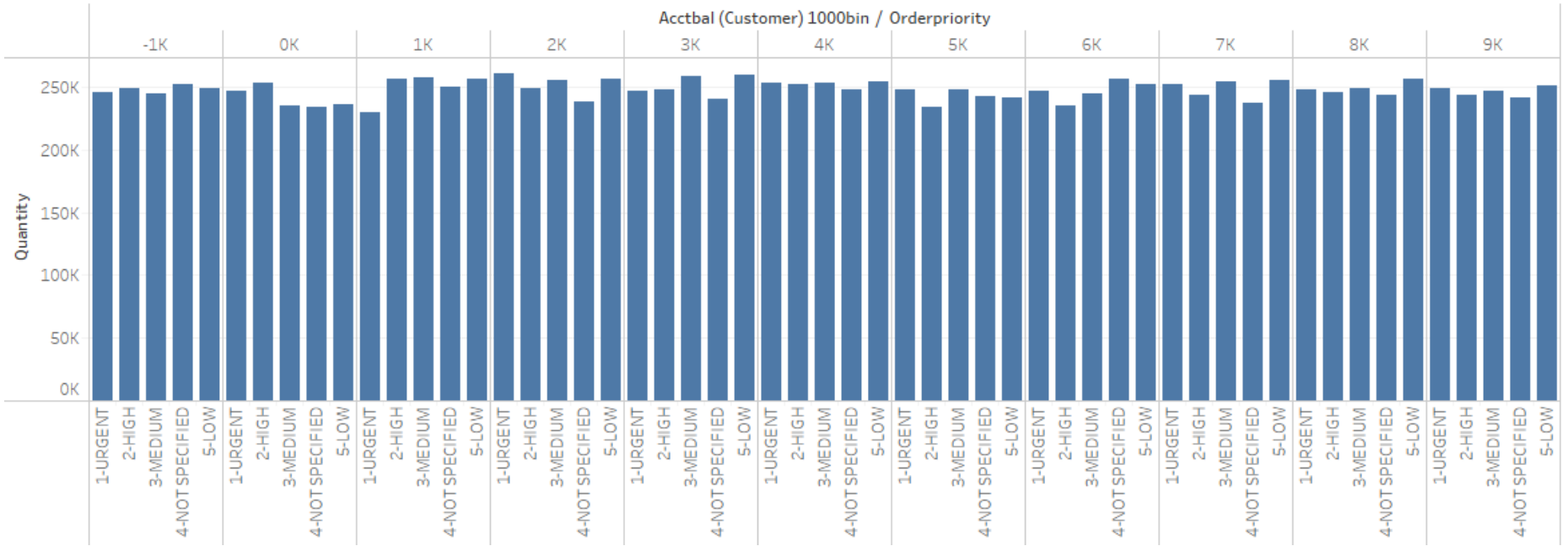
The hints in grey will *not* be given during the exam

Exercise 5.1

Query:

- Measure: sum(Quantity)
- Filter: [Order] Year = 1998
- Group by: [Order] Order priority, [Customer] Account balance
  - [Customer] Account balance must be binned; set bin size to 1000

Chart:



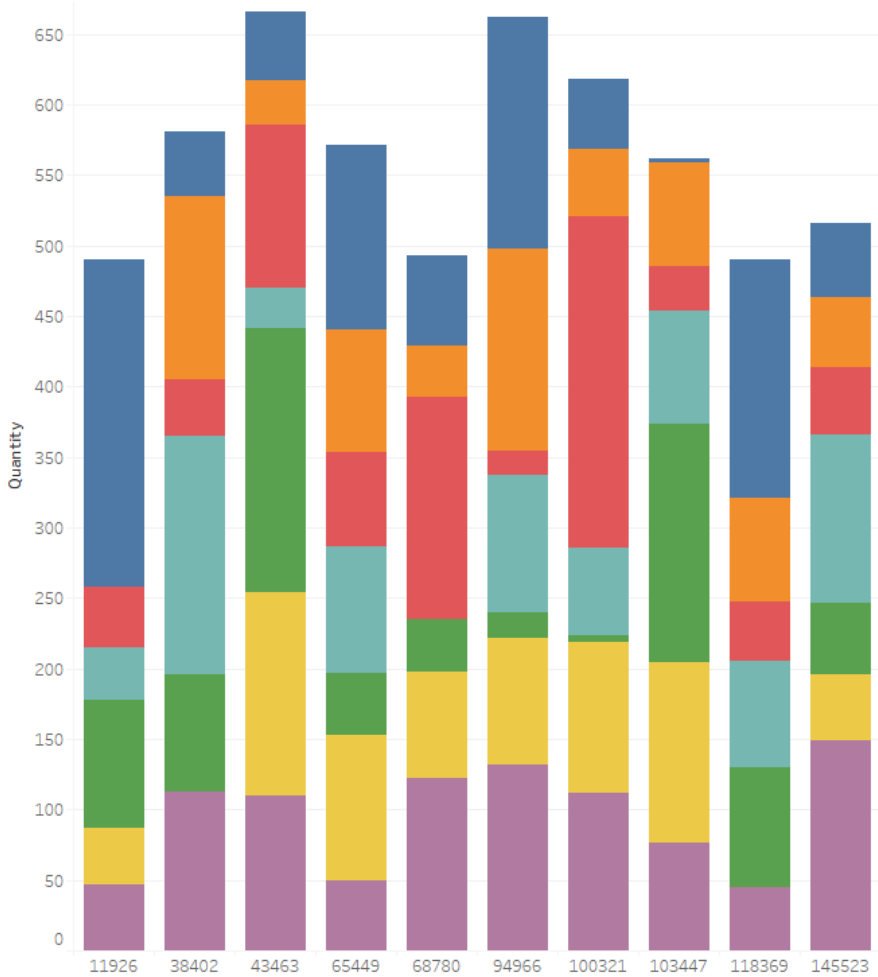
Exercise 5.2

Query:

- Measure: sum(Quantity)
- Filters:
  - [Part] Mfgr = ‘Manufacturer#1’
  - Top 10 customers that maximize sum(Sales) considering only the facts associated with the previous filter
  - The filter on [Part] Mfgr must be "*applied to the context*" to make the Top 10 filter depend on its results
- Group by: [JDT] Shipmode, [Customer] Custkey (Top 10 by sum(Sales))

Chart:

- [Customer] Name to be shown on hover
- [JDT] Shipmode to "break down" the bars



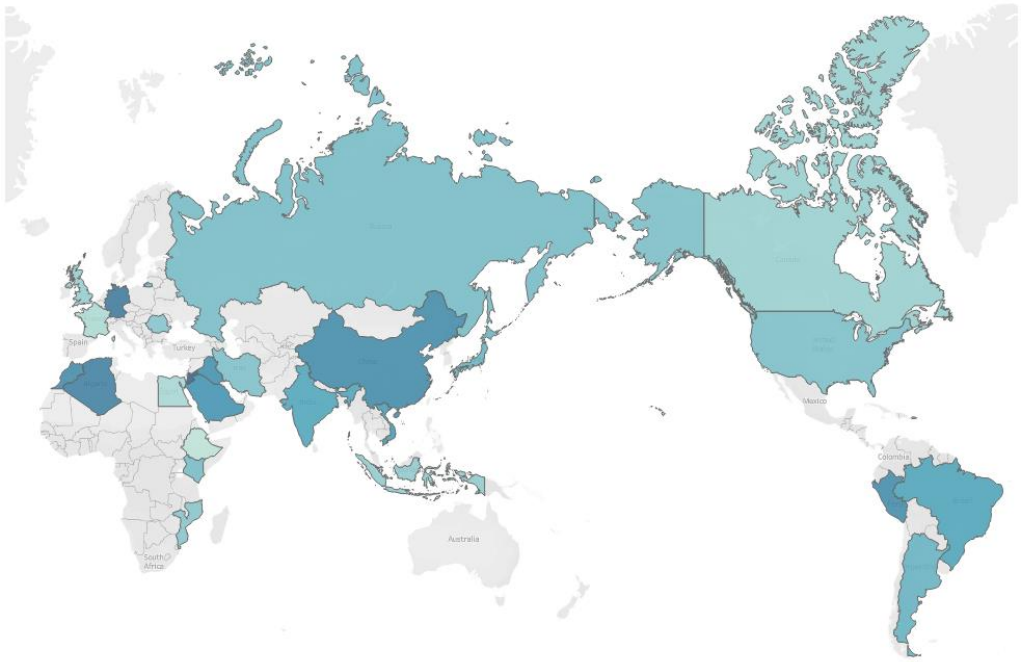
Exercise 5.3

Query:

- Measure: avg(TotSalesPerCustomer)
  - TotSalesPerCustomer is the sum(Sales) calculated by Custkey
    {FIXED [Custkey (Customer Region)]: SUM([Sales])}
- Filter: -
- Group by: [Customer] Nation

Chart:

- You may need to set the data type of [Customer] Nation to a geographic role

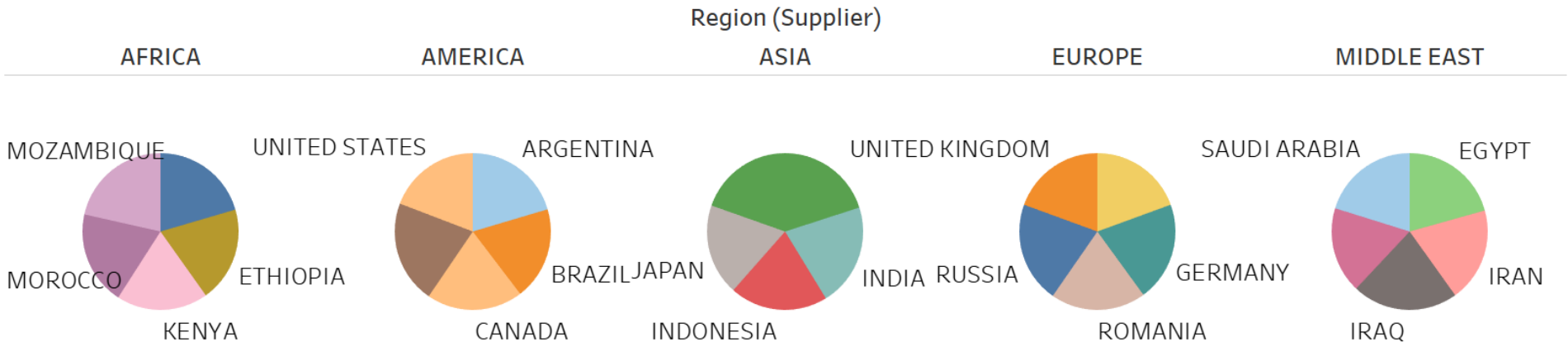


Exercise 5.4

Query:

- Measure: PercentageOfSupplierNationSalesByRegion
  - I.e., the sum(Sales) by [Supplier] Nation divided by the sum(Sales) by [Supplier] Region
  - {FIXED [Nation (Supplier)]: SUM([Sales])} / {FIXED [Region (Supplier)]: SUM([Sales])}
- Filter: [Order] Orderpriority = ‘Urgent’
  - The measure must be calculated on the results of the filter
  - I.e., the filter must be "*applied to the context*"
- Group by: [Supplier] Region, [Supplier] Nation

Chart:



Exercise 5.5

Query:

- Measure: countDistinct(Orderkey), countDistinct(Custkey)
- Filter: [JDT] Shipinstruct != null AND [JDT] Shipmode != null
- Group by: [JDT] Shipinstruct, [JDT] Shipmode

Chart:

- countDistinct(Orderkey) as size
- countDistinct(Custkey) as color
- Use the "Show Me" panel to display the heatmap