Dimension and Facts

By Dr. Ujwala Bharambe

© Dr.Ujwala Bharambe

Structuring/Modeling Issues

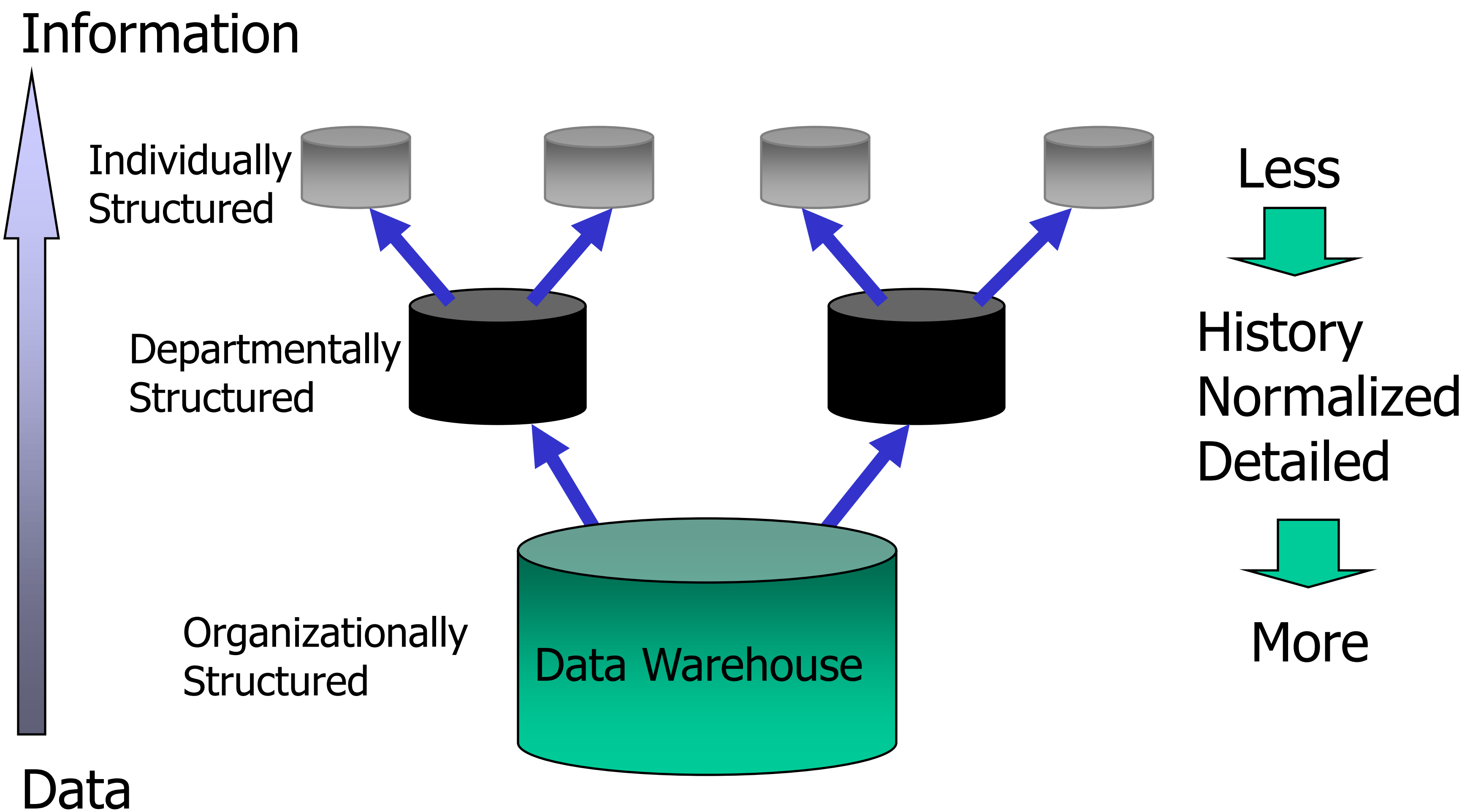
Data -- Heart of the Data Warehouse

- Heart of the data warehouse is the data itself!
- Single version of the truth
- Corporate memory
- Data is organized in a way that represents business -- subject orientation

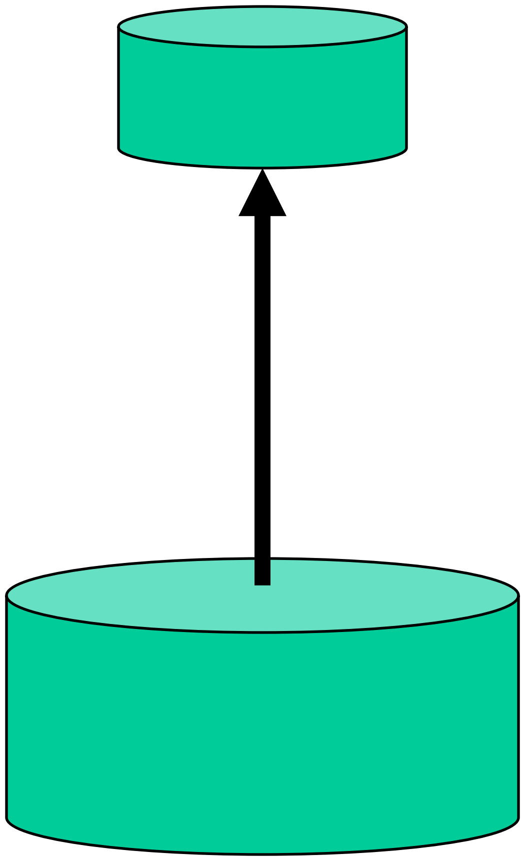
OLAP is an acronym for **Online Analytical Processing**. OLAP performs multidimensional analysis of business data and provides the capability for complex calculations, trend analysis, and sophisticated data modeling.

© Dr.Ujwala Bharambe

From the Data Warehouse to Data Marts



Data Warehouse and Data Marts

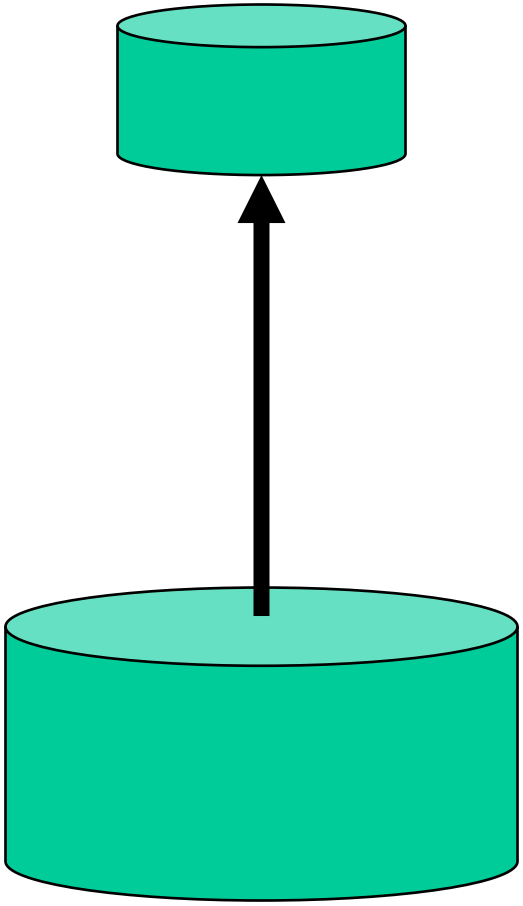


OLAP
Data Mart
Lightly summarized
Departmentally structured

Organizationally structured
Atomic
Detailed Data Warehouse Data

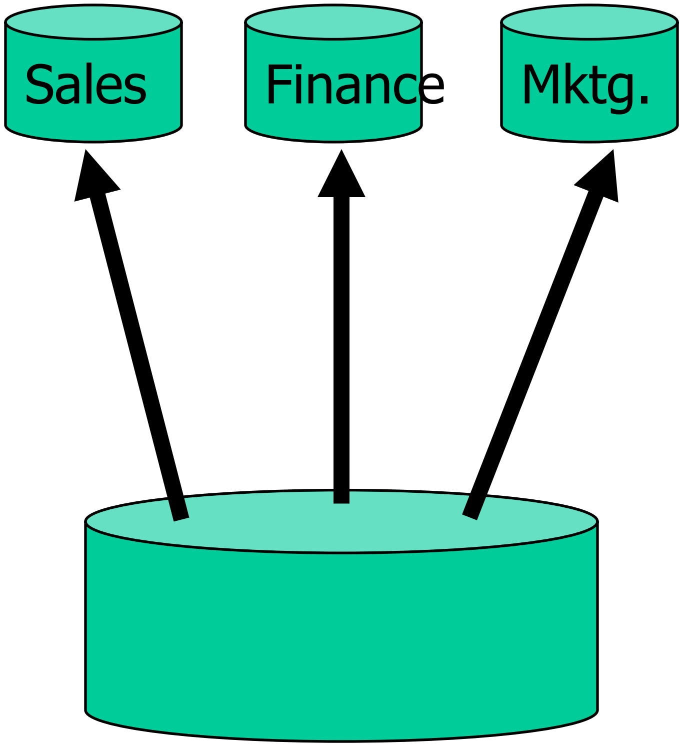
© Dr.Ujwala Bharambe

Characteristics of the Departmental Data Mart



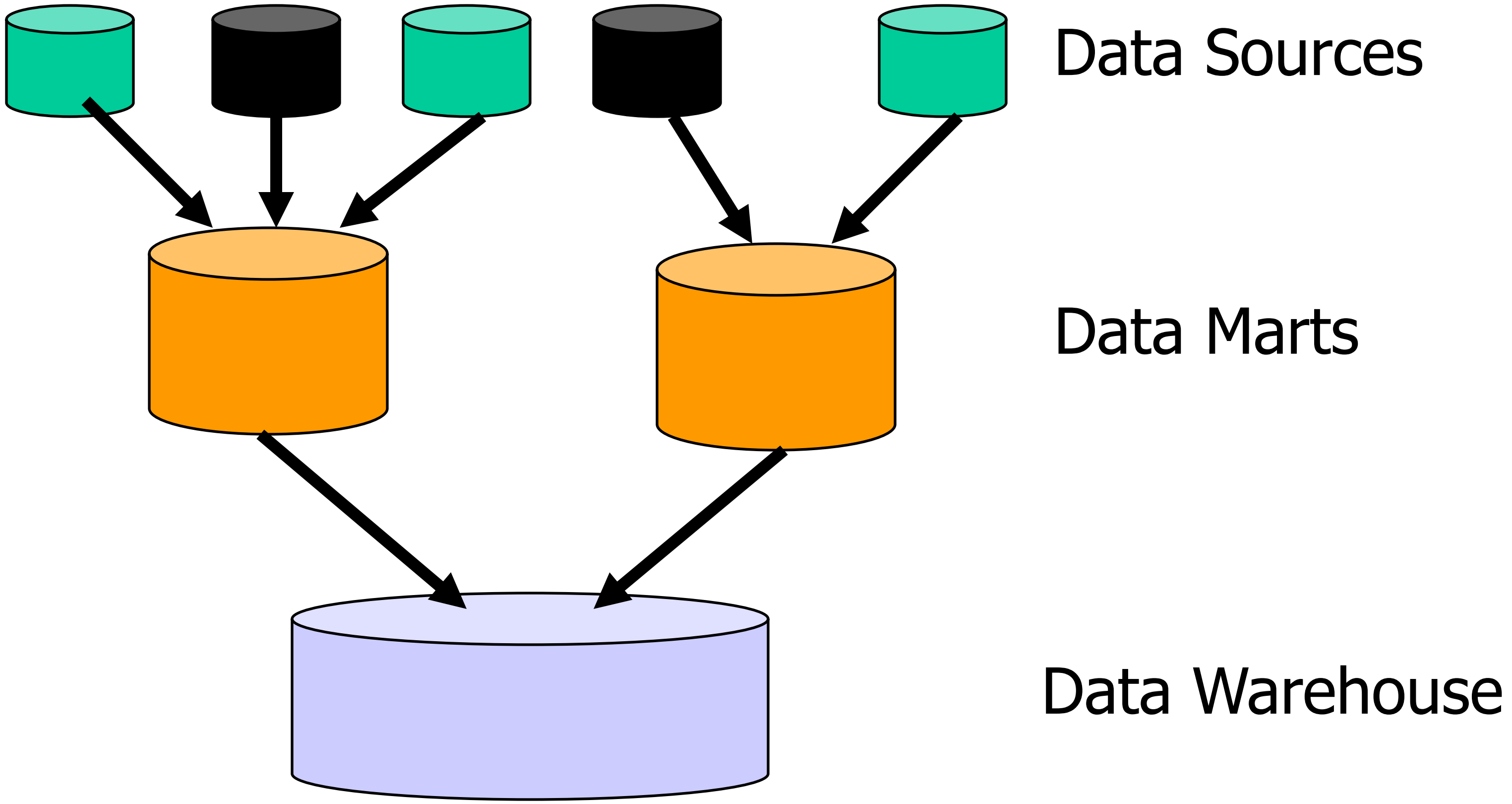
- OLAP
- Small
- Flexible
- Customized by Department
- Source is departmentally structured data warehouse

Techniques for Creating Departmental Data Mart

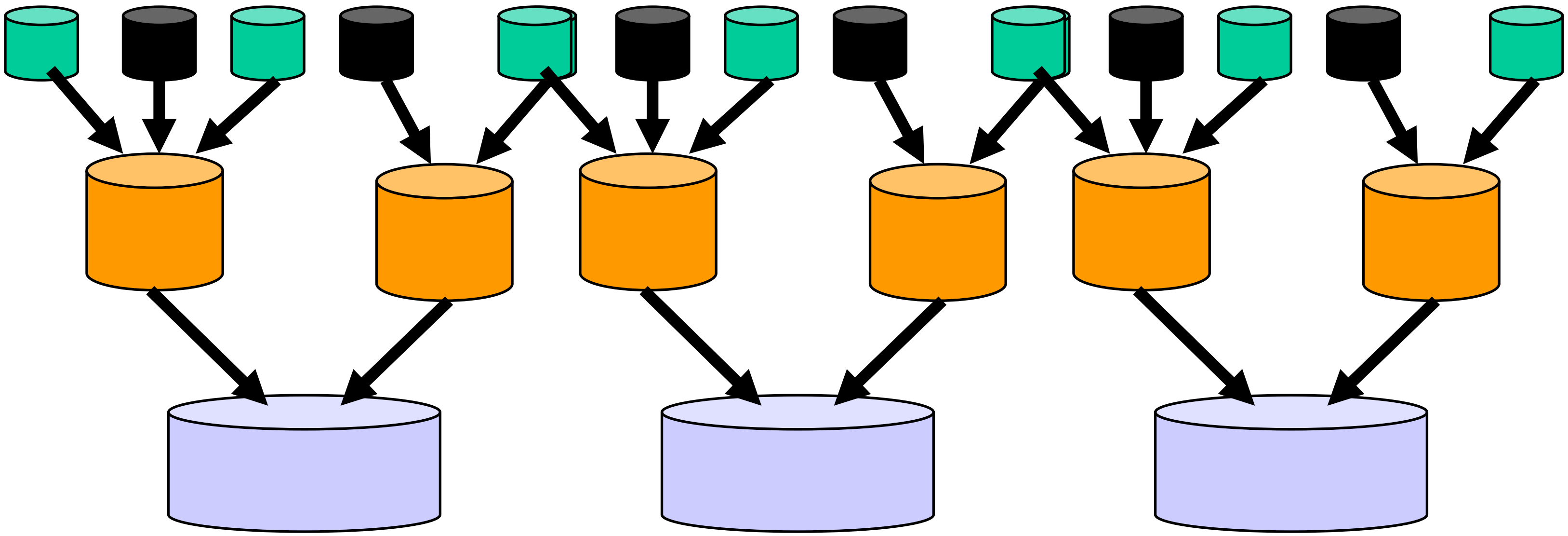


- OLAP
- Subset
- Summarized
- Superset
- Indexed
- Arrayed

Data Mart Centric



Problems with Data Mart Centric Solution



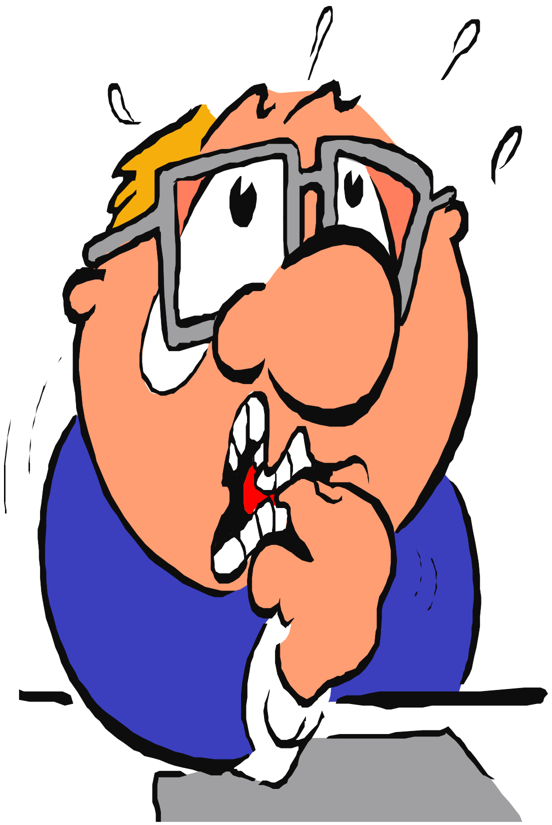
If you end up creating multiple warehouses, integrating them is a problem

© Dr.Ujwala Bharambe

II. On-Line Analytical Processing (OLAP)

Making Decision Support Possible

Limitations of SQL



"A Freshman in Business needs a Ph.D. in SQL"

-- Ralph Kimball

Typical OLAP Queries

- Write a multi-table join to compare sales for each product line YTD this year vs. last year.
- Repeat the above process to find the top 5 product contributors to margin.
- Repeat the above process to find the sales of a product line to new vs. existing customers.
- Repeat the above process to find the customers that have had negative sales growth.

©Dr.Ujwala Bharambe

What Is OLAP?

- Online Analytical Processing - coined by EF Codd in 1994 paper contracted by Arbor Software*
- Generally synonymous with earlier terms such as Decisions Support, Business Intelligence, Executive Information System
- OLAP = Multidimensional Database
- MOLAP: Multidimensional OLAP (Arbor Essbase, Oracle Express)
- ROLAP: Relational OLAP (Informix MetaCube, Microstrategy DSS Agent)

*** Reference: http://www.arborsoft.com/essbase/wht_ppr/coddTOC.html**

Strengths of OLAP

- It is a powerful visualization paradigm
- It provides fast, interactive response times
- It is good for analyzing time series
- It can be useful to find some clusters and outliers
- Many vendors offer OLAP tools

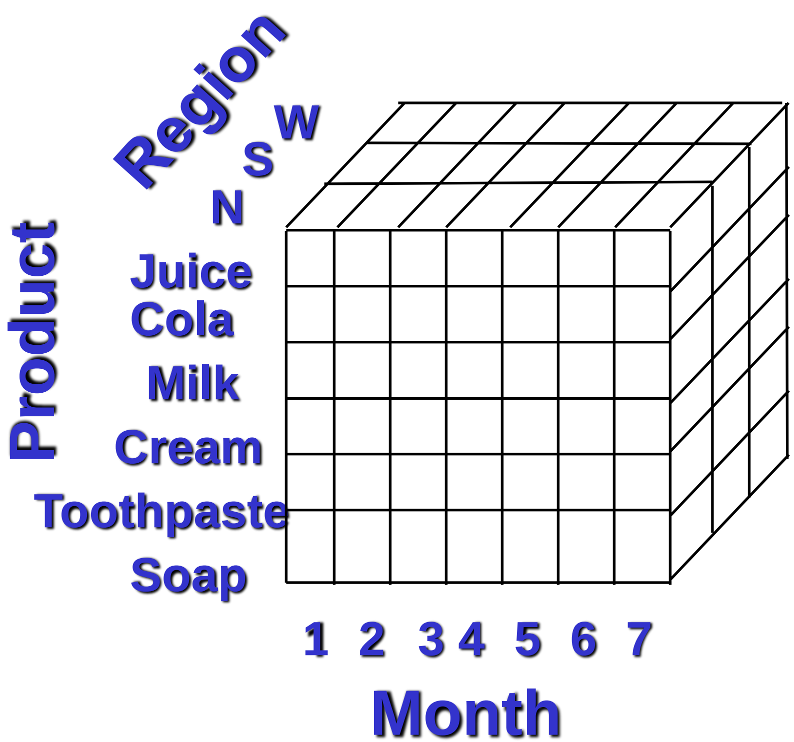
OLAP Is FASMI

- Fast
- Analysis
- Shared
- Multidimensional
- Information

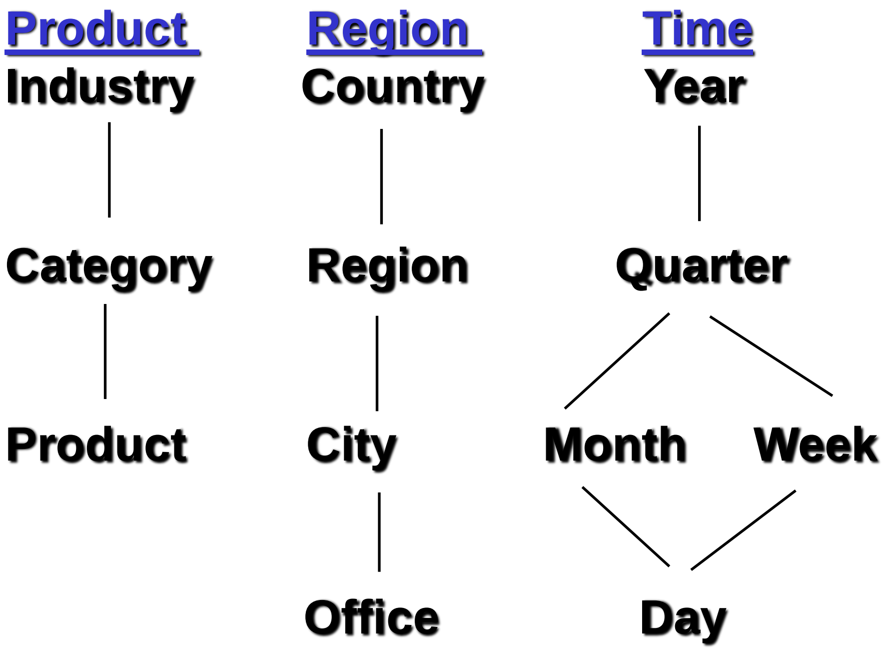
Nigel Pendse, Richard Creath - The OLAP Report

Multi-dimensional Data

- "Hey…I sold $100M worth of goods"



Dimensions: Product, Region, Time
Hierarchical summarization paths



Data Cube Lattice

- Cube lattice
  - ABC
    AB AC BC
    A B C
    none
- Can materialize some groupbys, compute others on demand
- Question: which groupbys to materialze?
- Question: what indices to create
- Question: how to organize data (chunks, etc)

© Dr.Ujwala Bharambe

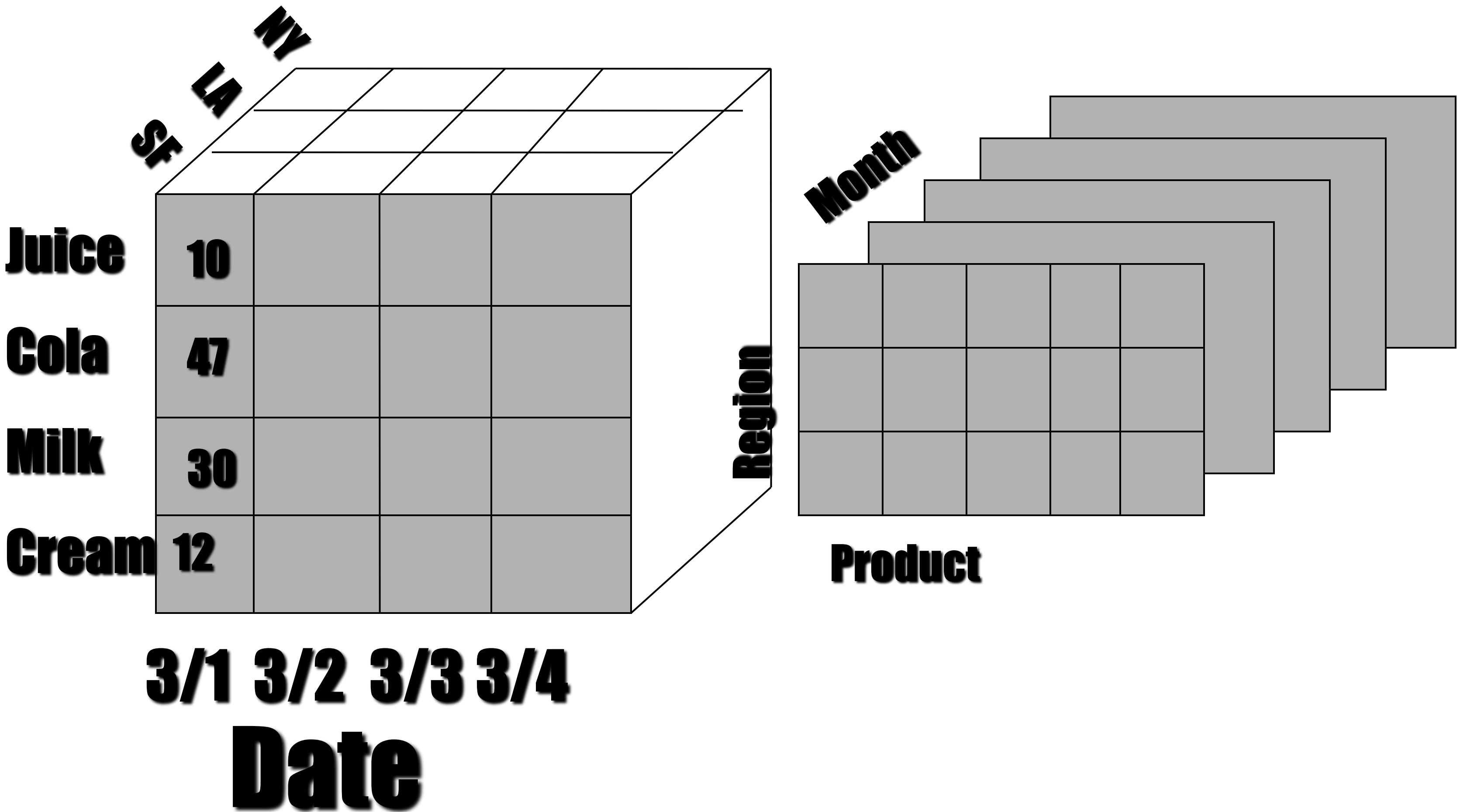
Visualizing Neighbors is simpler

	1	2	3	4	5	6	7	8
Apr								
May								
Jun								
Jul								
Aug								
Sep								
Oct								
Nov								
Dec								
Jan								
Feb								
Mar								

Month	Store	Sales
Apr	1	
Apr	2	
Apr	3	
Apr	4	
Apr	5	
Apr	6	
Apr	7	
Apr	8	
May	1	
May	2	
May	3	
May	4	
May	5	
May	6	
May	7	
May	8	
Jun	1	
Jun	2	

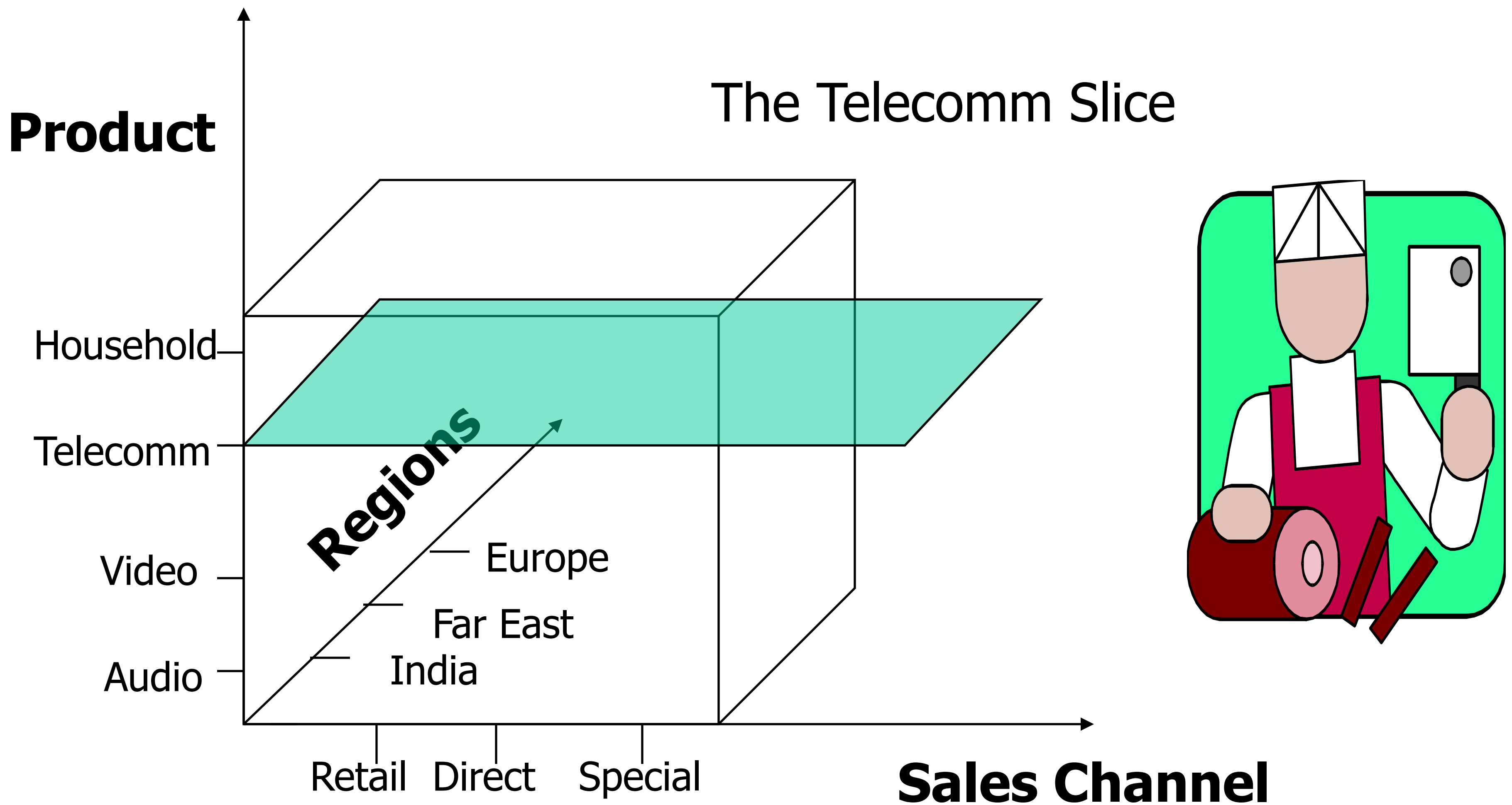
© Dr.Ujwala Bharambe

A Visual Operation: Pivot (Rotate)



©Dr.Ujwala Bharambe

"Slicing and Dicing"



Roll-up and Drill Down

Higher Level of Aggregation

Roll Up

- Sales Channel
- Region
- Country
- State
- Location Address
- Sales Representative

Drill-Down

Low-level Details

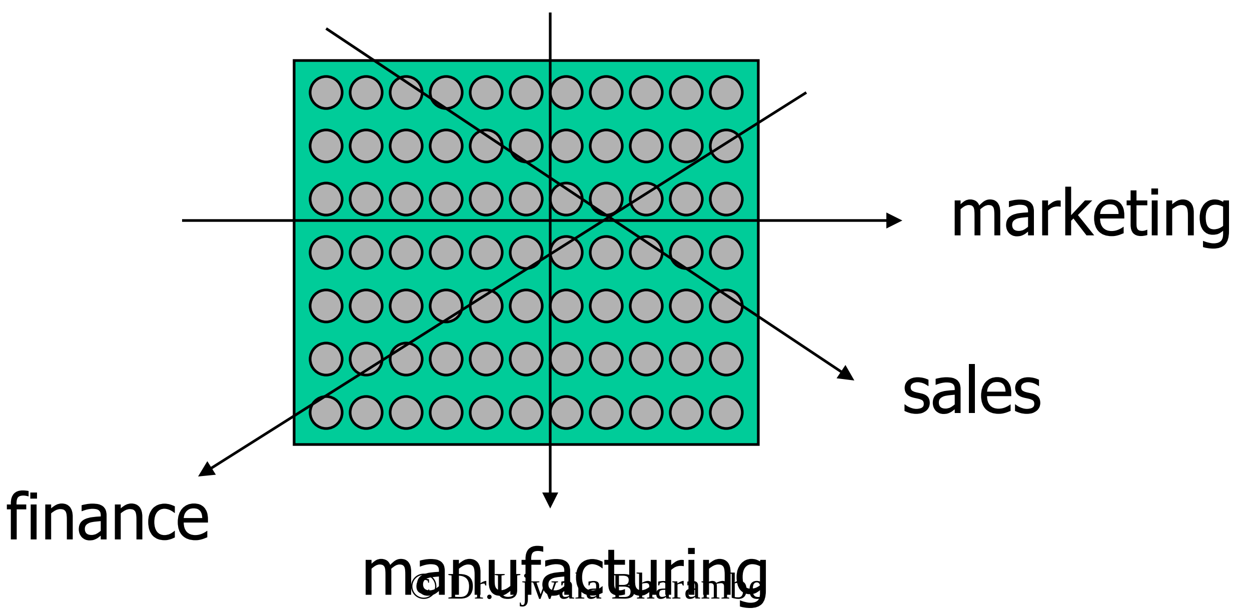
Nature of OLAP Analysis

- Aggregation -- (total sales, percent-to-total)
- Comparison -- Budget vs. Expenses
- Ranking -- Top 10, quartile analysis
- Access to detailed and aggregate data
- Complex criteria specification
- Visualization

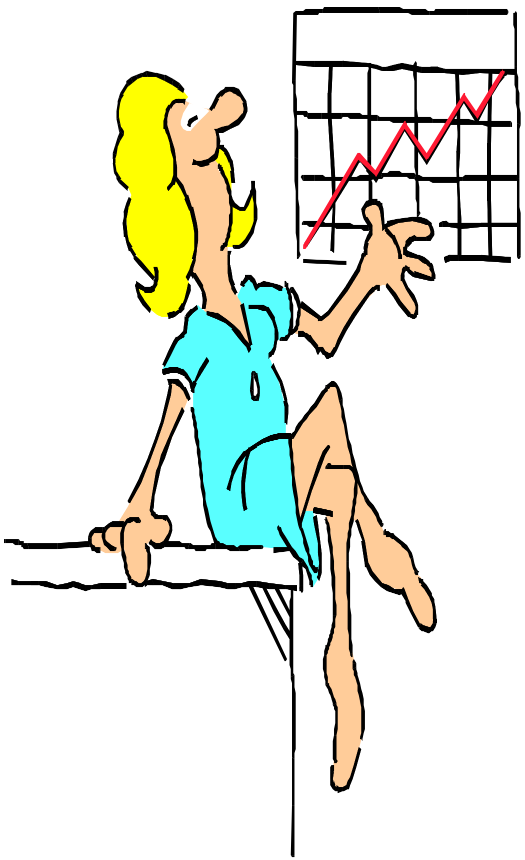
© Dr.Ujwala Bharambe

Organizationally Structured Data

- Different Departments look at the same detailed data in different ways. Without the detailed, organizationally structured data as a foundation, there is no reconcilability of data



Multidimensional Spreadsheets



- Analysts need spreadsheets that support
  - pivot tables (cross-tabs)
  - drill-down and roll-up
  - slice and dice
  - sort
  - selections
  - derived attributes
- Popular in retail domain

OLAP - Data Cube

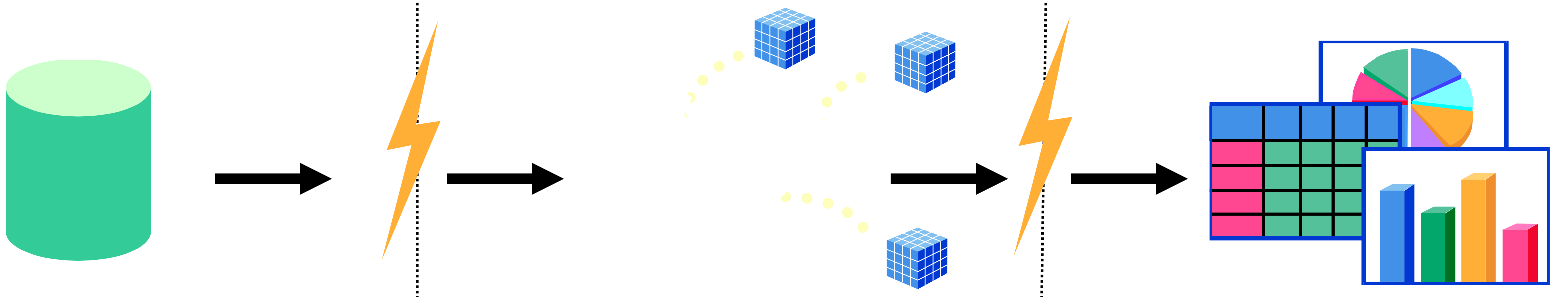
- Idea: analysts need to group data in many different ways
  - eg. Sales(region, product, prodtype, prodstyle, date, saleamount)
  - saleamount is a measure attribute, rest are dimension attributes
  - groupby every subset of the other attributes
    - materialize (precompute and store) groupbys to give online response
  - Also: hierarchies on attributes: date -> weekday, date -> month -> quarter -> year

SQL Extensions

- Front-end tools require
  - Extended Family of Aggregate Functions
    - rank, median, mode
  - Reporting Features
    - running totals, cumulative totals
  - Results of multiple group by
    - total sales by month and total sales by product
  - Data Cube

Relational OLAP: 3 Tier DSS

Data Warehouse | **ROLAP Engine** | **Decision Support Client**



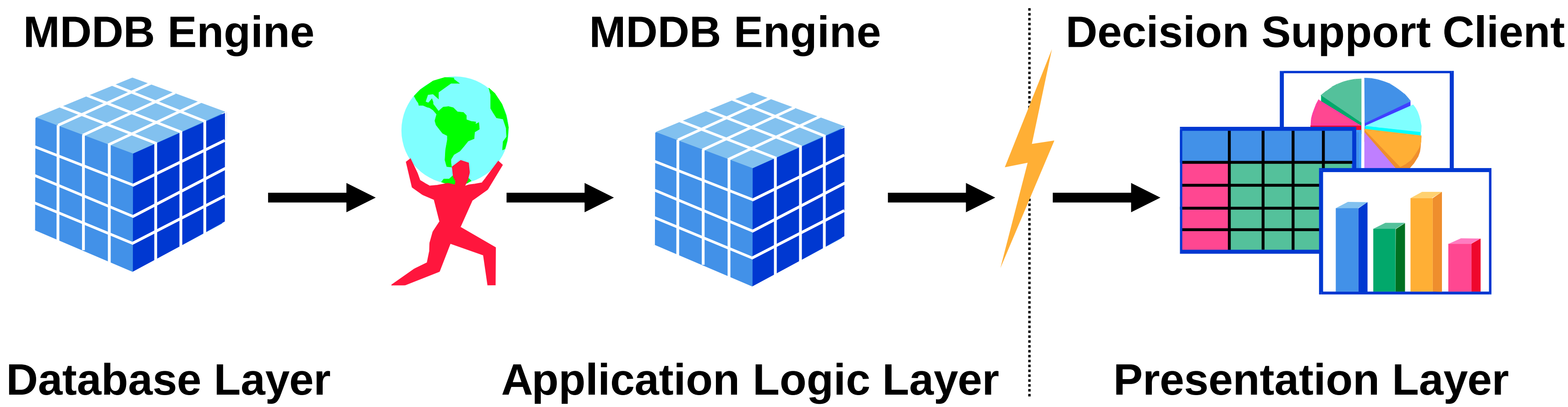
Database Layer | **Application Logic Layer** | **Presentation Layer**

Store atomic data in industry standard RDBMS.

Generate SQL execution plans in the ROLAP engine to obtain OLAP functionality.

Obtain multi-dimensional reports from the DSS Client.

MD-OLAP: 2 Tier DSS

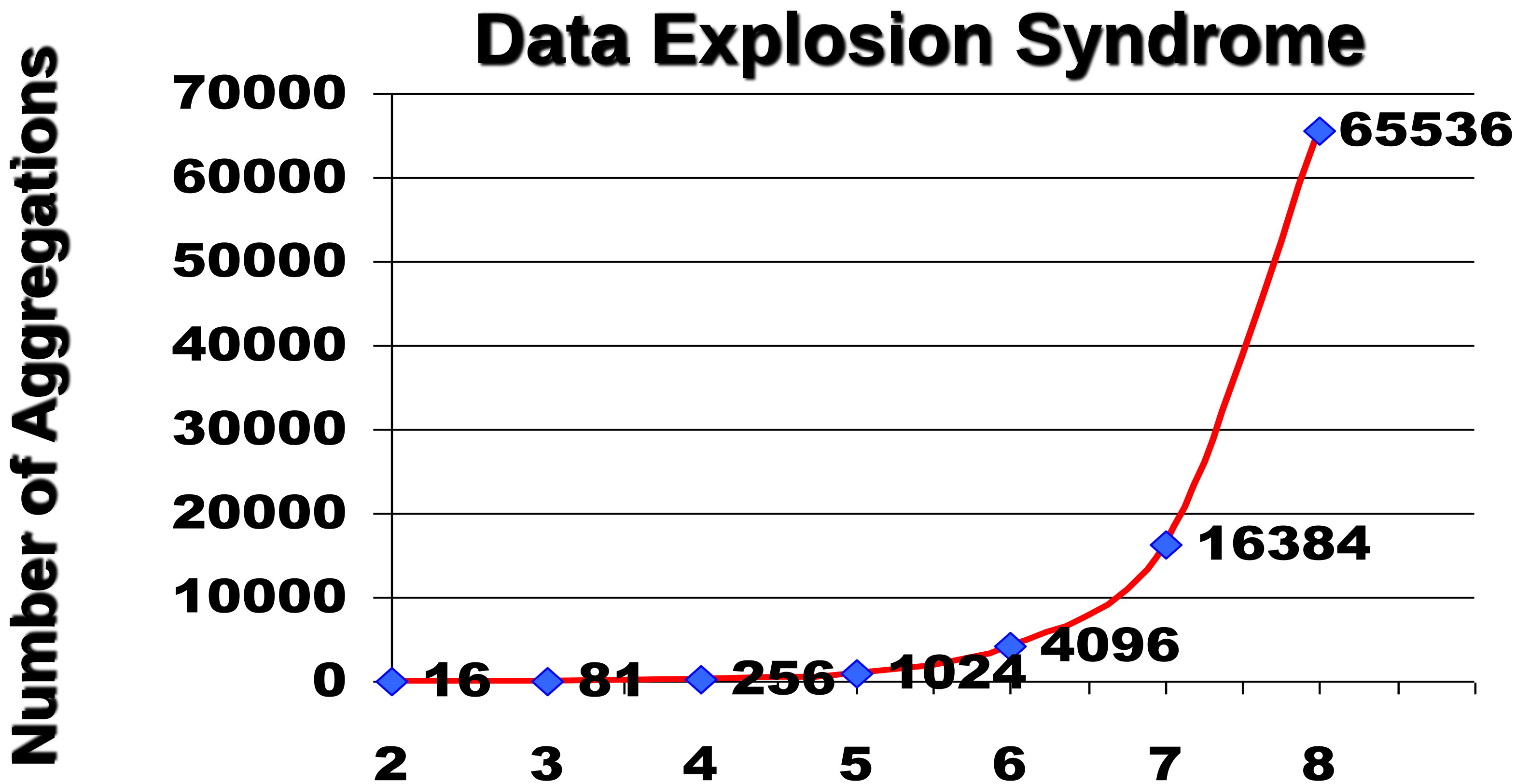


Store atomic data in a proprietary data structure (MDDB), pre-calculate as many outcomes as possible, obtain OLAP functionality via proprietary algorithms running against this data.

Obtain multi-dimensional reports from the DSS Client.

Typical OLAP Problems

Data Explosion



(4 levels in each dimension)

Granularity in Warehouse

- Can not answer some questions with summarized data
  - Did Anand call Seshadri last month? Not possible to answer if total duration of calls by Anand over a month is only maintained and individual call details are not.
- Detailed data too voluminous

Granularity in Warehouse

- Tradeoff is to have dual level of granularity
  - Store summary data on disks
    - 95% of DSS processing done against this data
  - Store detail on tapes
    - 5% of DSS processing against this data

Vertical Partitioning

Acct. No	Name	Balance	Date Opened	Interest Rate	Address

Frequently accessed

Rarely accessed

Acct. No	Balance

Acct. No	Name	Date Opened	Interest Rate	Address

Smaller table and so less I/O

Derived Data

- Introduction of **derived** (calculated data) may often help
- Have seen this in the context of dual levels of granularity
- Can keep **auxiliary views** and indexes to speed up query processing

Schema Design

- Database organization
  - must look like business
  - must be recognizable by business user
  - approachable by business user
  - Must be <u>*simple*</u>
- Schema Types
  - Star Schema
  - Fact Constellation Schema
  - Snowflake schema

Dimensional Modelling

- A dimensional model is a data structure technique optimized for Data warehousing tools. **Facts are the measurements/metrics or facts from your business process. Dimension provides the context surrounding a business process event**. Attributes are the various characteristics of the dimension modelling.

Diamension

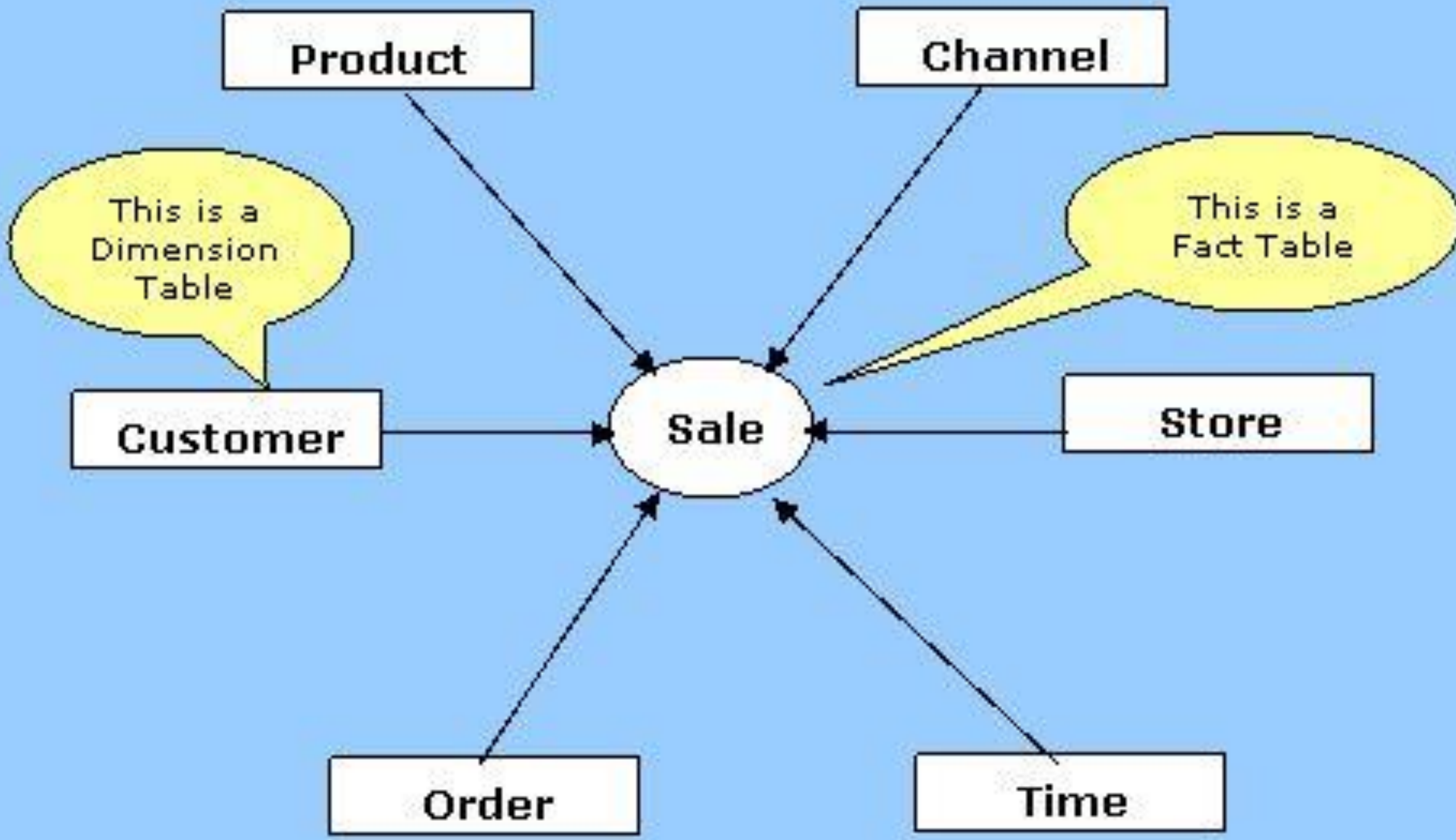
- The dimensions must be defined within the grain from the second step of the 4-step process. Dimensions are **the foundation of the fact table, and is where the data for the fact table is collected**. Typically dimensions are nouns like date, store, inventory etc. These dimensions are where all the data is stored.

Dimension and Fact

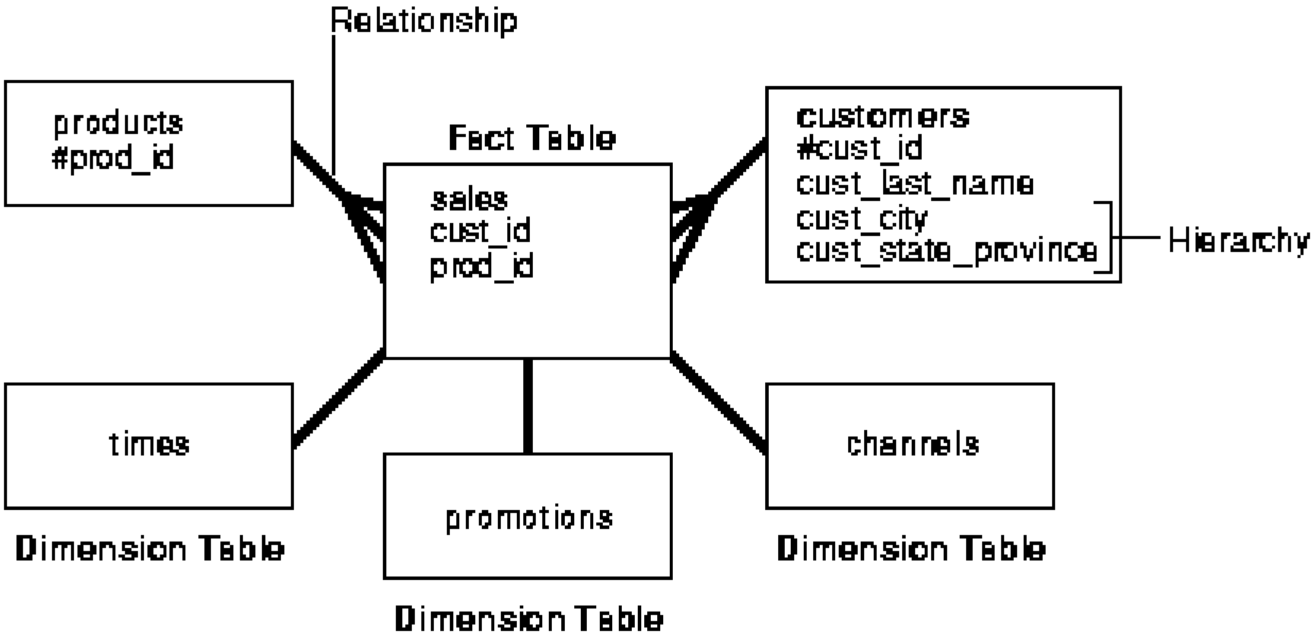
- A dimension is a **structure that categorizes facts and measures** in order to enable users to answer business questions. Commonly used dimensions are people, products, place and time.
- A fact is a **value**, or measurement, which represents a fact about the managed entity or system.

Fact and Diamension

- **Facts and dimensions** are data warehousing terms.
- A fact is a **quantitative piece** of information - such as a sale or a download.
- Facts are stored in fact tables, and have a foreign key relationship with a number of dimension tables.
- Dimensions are **companions to facts**, and describe the objects in a fact table.



© Dr.Ujwala Bharambe


Relationship
products
#prod_id
Fact Table
sales
cust_id
prod_id
customers
#cust_id
cust_last_name
cust_city
cust_state_province
Hierarchy
times
promotions
channels
Dimension Table
Dimension Table
Dimension Table

Dimension Tables

- Dimension tables
  - Define business in terms already familiar to users
  - Wide rows with lots of **descriptive text**
  - Small tables (about a million rows)
  - Joined to **fact table** by a foreign key
  - heavily indexed
  - typical dimensions
    - time periods, geographic region (markets, cities), products, customers, salesperson, etc.

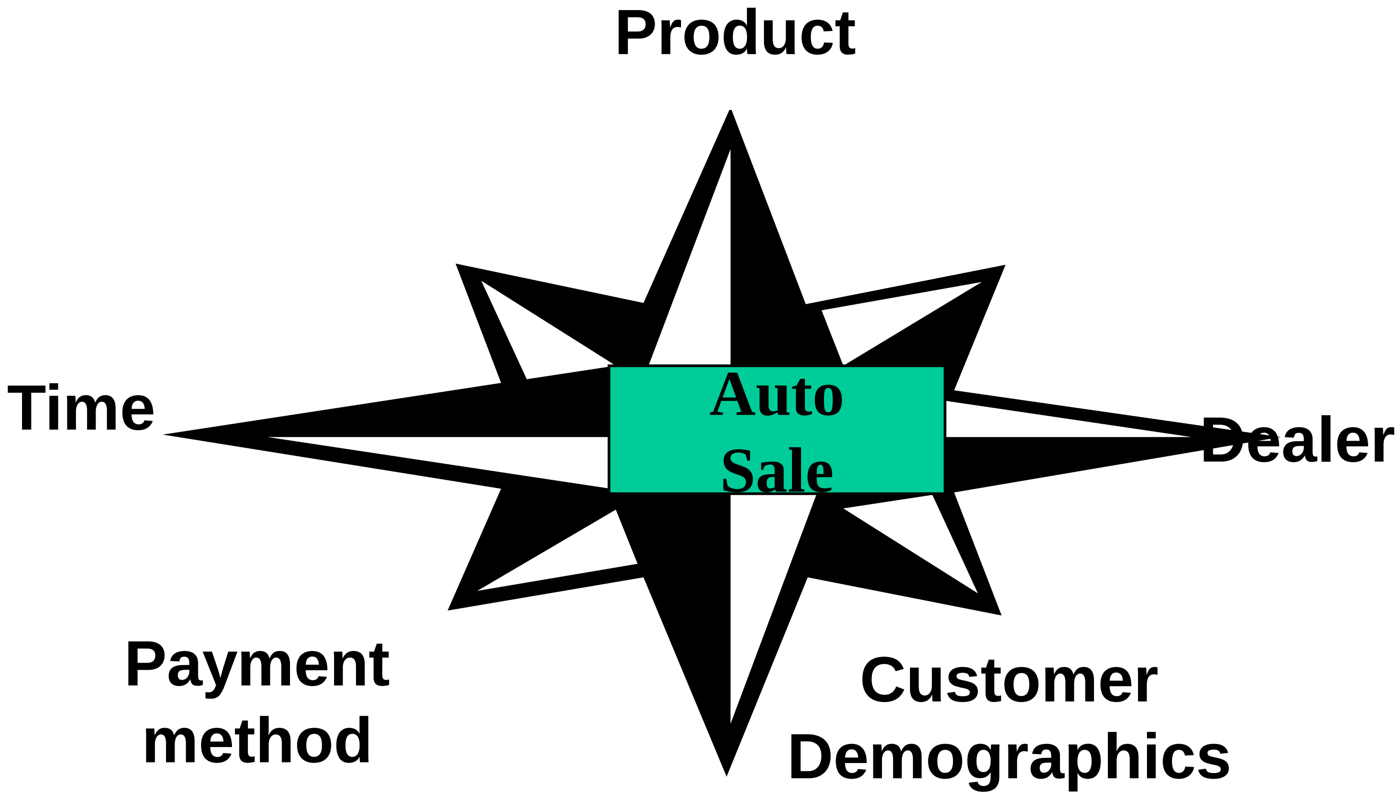
Key difference between Fact and dimension table

- Fact table contains **measurements, metrics, and facts** about a business process while the Dimension table is a companion to the fact table which contains descriptive attributes to be used as query constraining.
- Fact table is located at the **center** of a star or snowflake schema, whereas the Dimension table is located at the edges of the star or snowflake schema.
- Fact table is defined by their grain or its most atomic level whereas Dimension table should be wordy, descriptive, complete, and quality assured.
- Fact table helps to store report labels whereas Dimension table contains detailed data.
- Fact table does not contain a hierarchy whereas the Dimension table contains hierarchies.

Key difference between Fact and dimension table

Parameters	Fact Table	Dimension Table
Definition	Measurements, metrics or facts about a business process.	Companion table to the fact table contains descriptive attributes to be used as query constraining.
Characteristic	Located at the center of a star or snowflake schema and surrounded by dimensions.	Connected to the fact table and located at the edges of the star or snowflake schema
Design	Defined by their grain or its most atomic level.	Should be wordy, descriptive, complete, and quality assured.
Task	Fact table is a measurable event for which dimension table data is collected and is used for analysis and reporting.	Collection of reference information about a business.
Type of Data	Facts tables could contain information like sales against a set of dimensions like Product and Date.	Evert dimension table contains attributes which describe the details of the dimension. E.g., Product dimensions can contain Product ID, Product Category, etc.
Key	Primary Key in fact table is mapped as foreign keys to Dimensions.	Dimension table has a primary key columns that uniquely identifies each dimension.
Storage	Helps to store report labels and filter domain values in dimension tables.	Load detailed atomic data into dimensional structures.
Hierarchy	Does not contain Hierarchy	Contains Hierarchies. For example Location could contain, country, pin code, state, city, etc.

A STAR SCHEMA for Auto Sales



© Dr.Ujwala Bharambe

- Dimensional Modeling:

- Assume this to be the schema for a **manufacturing company** and that the marketing department is interested in determining how they are doing with the orders received by the company.

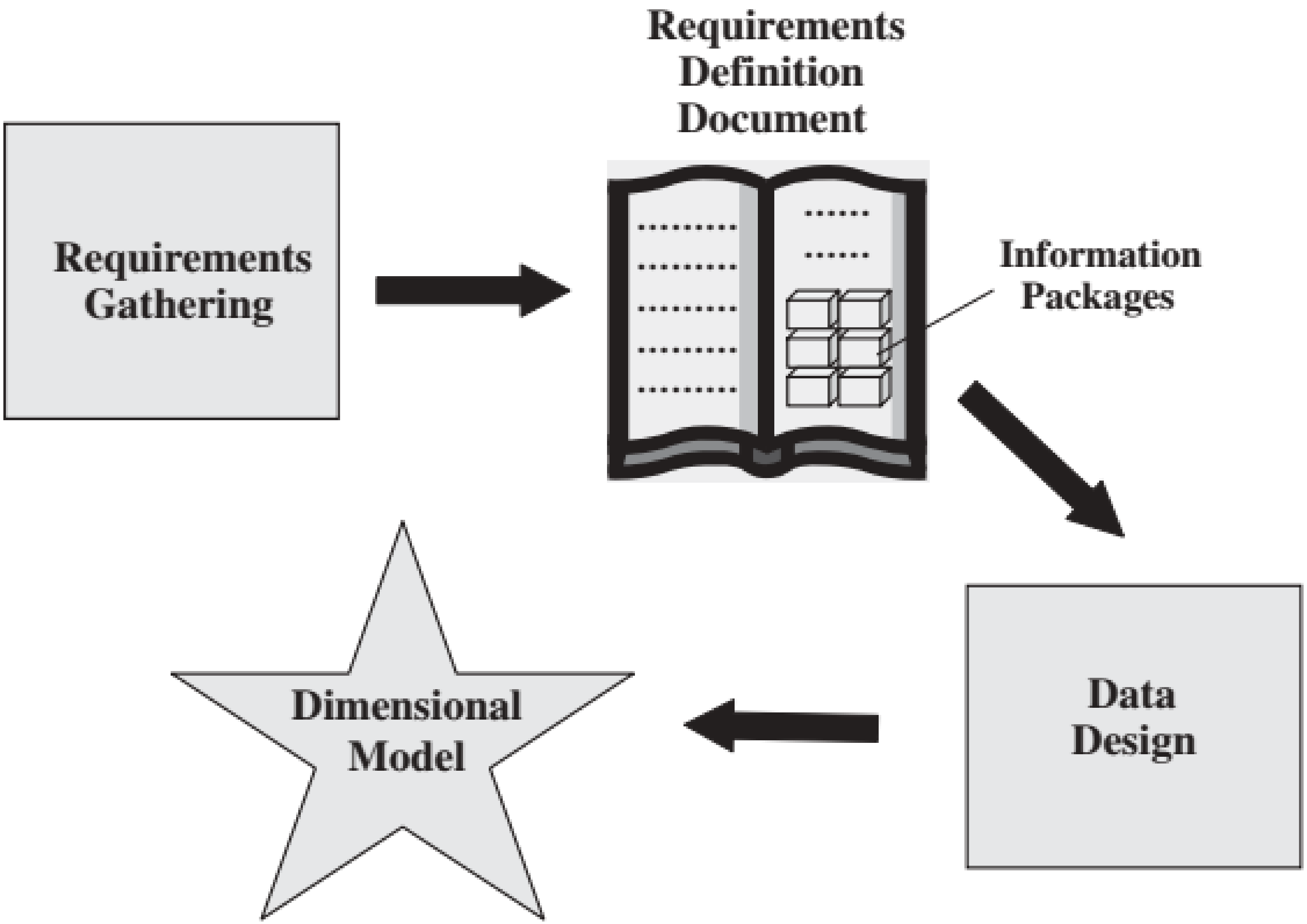


Figure 10-1 From requirements to data design.

Automaker Sales

Fact Table

Actual Sale Price
MSRP
Options Price
Full Price
Dealer Add-ons
Dealer Credits
Dealer Invoice
Down Payment
Proceeds Finance

Dimensions

Time	Product	Payment Method	Customer Demo-graphics	Dealer	
Year	Model Name	Finance Type	Age	Dealer Name	
Quarter	Model Year	Term (Months)	Gender	City	
Month	Package Styling	Interest Rate	Income Range	State	
Date	Product Line	Agent	Marital Status	Single Brand Flag	
Day of Week	Product Category		House-hold Size	Date First Operation	
Day of Month	Exterior Color		Vehicles Owned		
Season	Interior Color		Home Value		
Holiday Flag	First Year		Own or Rent		

Facts: Actual Sale Price, MSRP, Options Price, Full Price, Dealer Add-ons, Dealer Credits, Dealer Invoice, Down Payment, Proceeds, Finance

Figure 10-2 Formation of the automaker sales fact table.

- DW meant to answer questions on overall process
- DW focus is on how managers view the business
- DW reveals business trends
- Information is centered around a business process
- Answers show how the business measures the process
- The measures to be studied in many ways along several business dimensions

Dimensional Modeling

Captures critical measures
Views along dimensions
Intuitive to business users

Figure 10-6 Dimensional modeling for the data warehouse.

Schema Design

- Database organization
  - must look like business
  - must be recognizable by business user
  - approachable by business user
  - Must be *simple*
- Schema Types
  - Star Schema
  - Fact Constellation Schema
  - Snowflake schema

Dimension Tables

- Dimension tables
  - Define business in terms already familiar to users
  - Wide rows with lots of descriptive text
  - Small tables (about a million rows)
  - Joined to fact table by a foreign key
  - heavily indexed
  - typical dimensions
    - time periods, geographic region (markets, cities), products, customers, salesperson, etc.

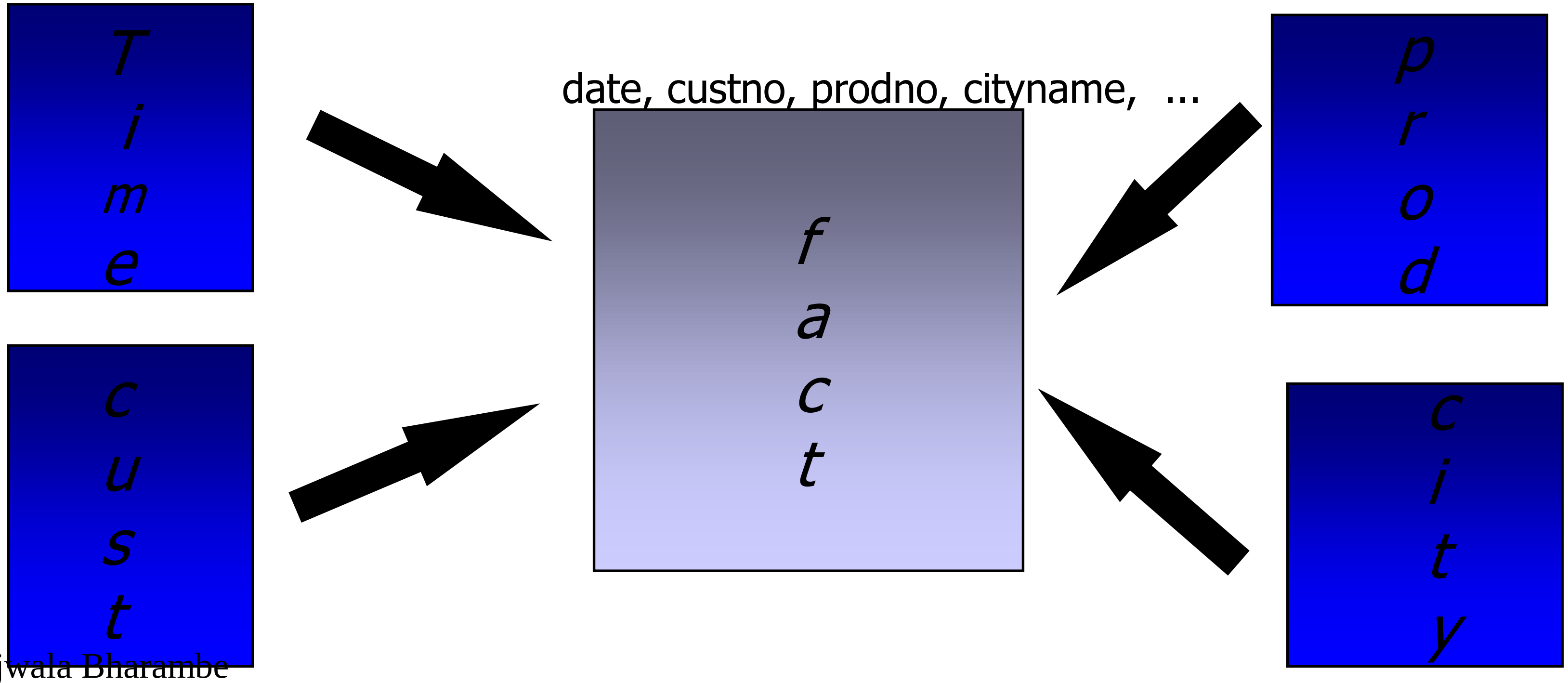
Fact Table

- Central table
  - mostly raw numeric items
  - narrow rows, a few columns at most
  - large number of rows (millions to a billion)
  - Access via dimensions

© Dr.Ujwala Bharambe

Star Schema

- A single fact table and for each dimension one dimension table
- Does not capture hierarchies directly



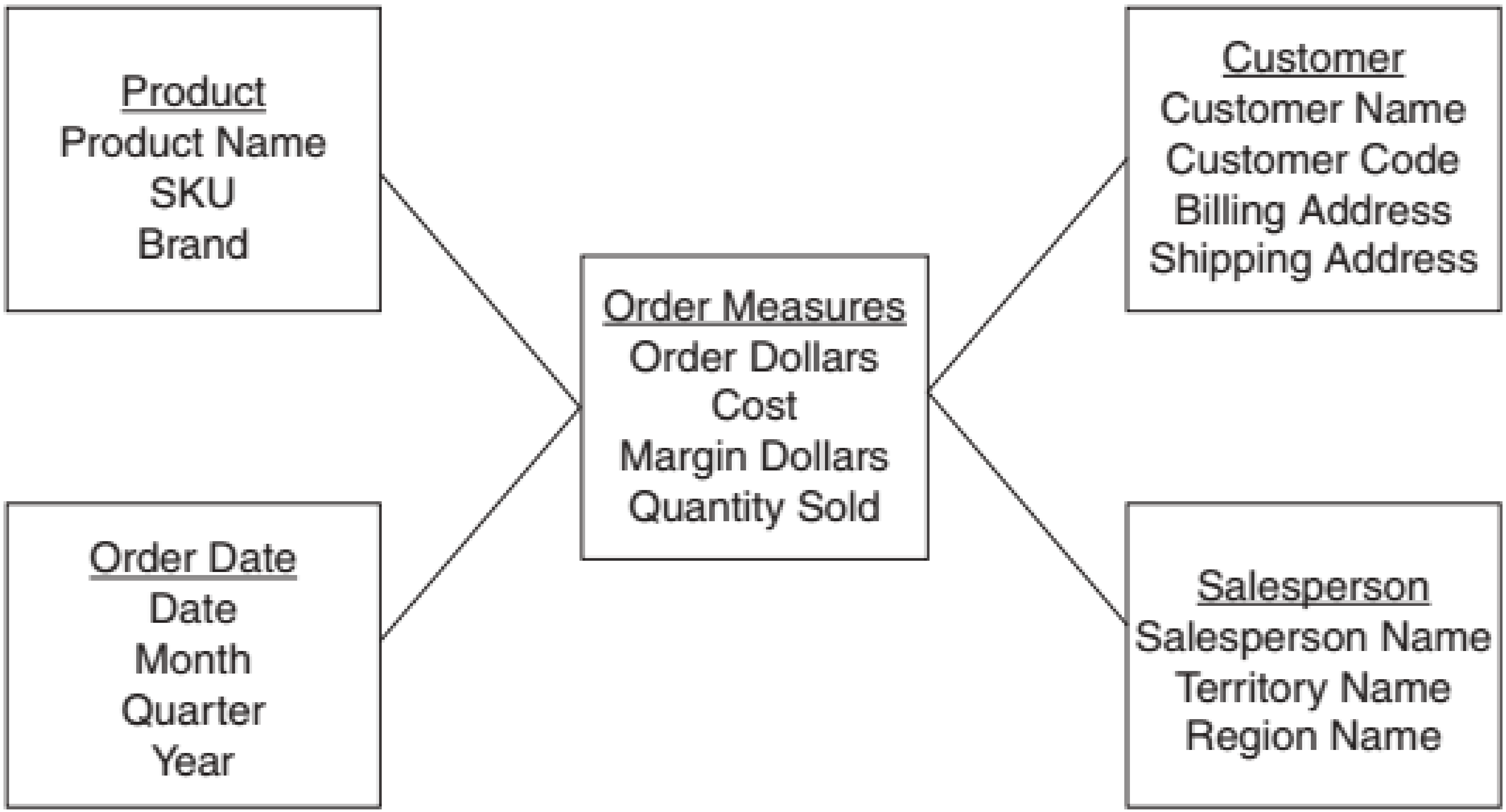
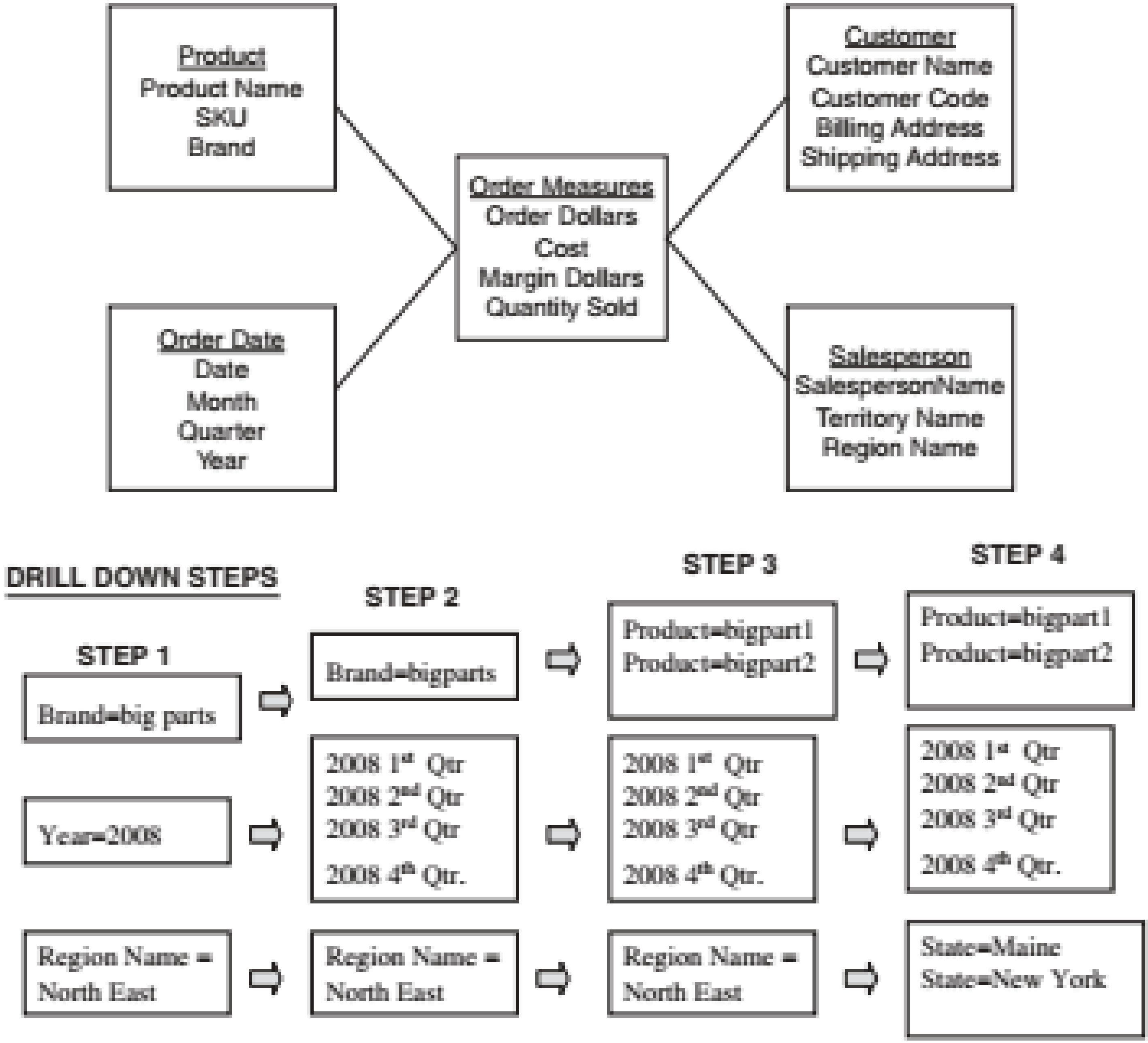


Figure 10-7 Simple STAR schema for orders analysis.



© Dr.Ujwala Bharambe **Figure 10-9** Understanding drill-down analysis from the STAR schema.

- Dimension table key
- Large number of attributes (wide)
- Textual attributes
- Attributes not directly related
- Flattened out, not normalized
- Ability to drill down/roll up
- Multiple hierarchies
- Less number of records

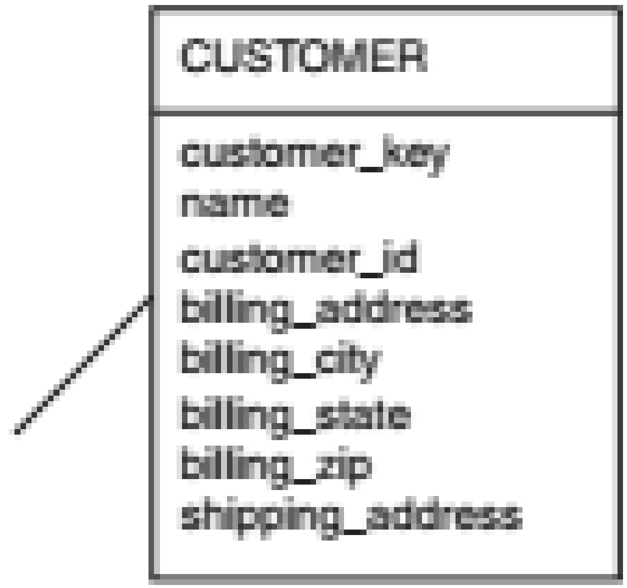


Figure 10-10 Inside a dimension table.

Inside a Dimension Table

- Dimension Table Key. The primary key of the dimension table uniquely identifies each row in the table.
- Table is Wide. Typically, a dimension table has many columns or attributes.
- Textual Attributes. In the dimension table you will seldom find any numerical values used for calculations. The attributes in a dimension table are of textual format.

Inside the Fact Table

- Concatenated Key. A row in the fact table relates to a combination of rows from all the dimension tables
- Data Grain. This is an important characteristic of the fact table. As we know, the data grain is the level of detail for the measurements or metrics

Inside the Fact Table

- Concatenated fact table key
- Grain or level of data identified
- Fully additive measures
- Semi-additive measures
- Large number of records
- Only a few attributes
- Sparsity of data
- Degenerate dimensions

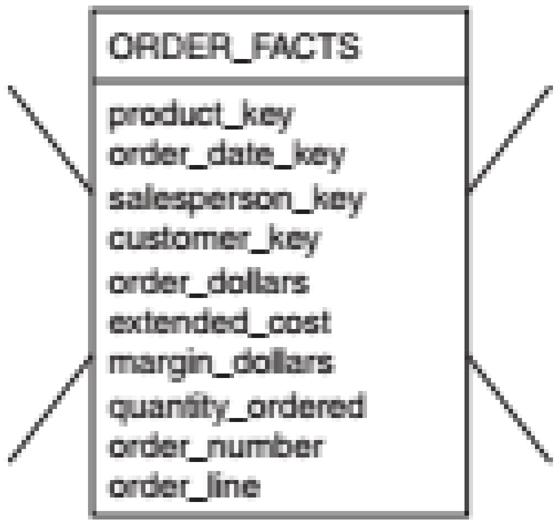


Figure 10-11 Inside a fact table.

Inside the Fact Table

Measures or facts are represented in a fact table. However, there are business events or coverage that could be represented in a fact table, although no measures or facts are associated with these.

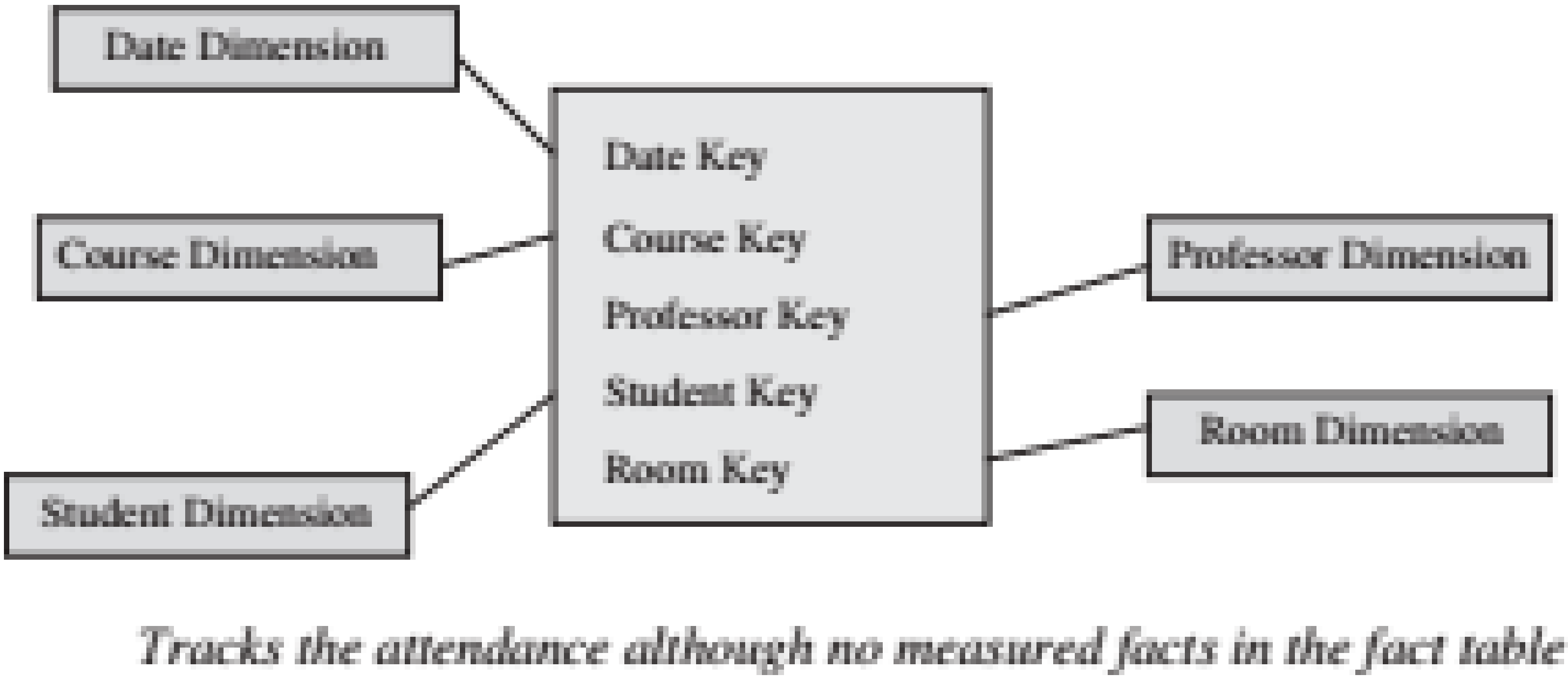


Figure 10-12 A factless fact table.

© Dr.Ujwala Bharambe

STAR SCHEMA KEYS

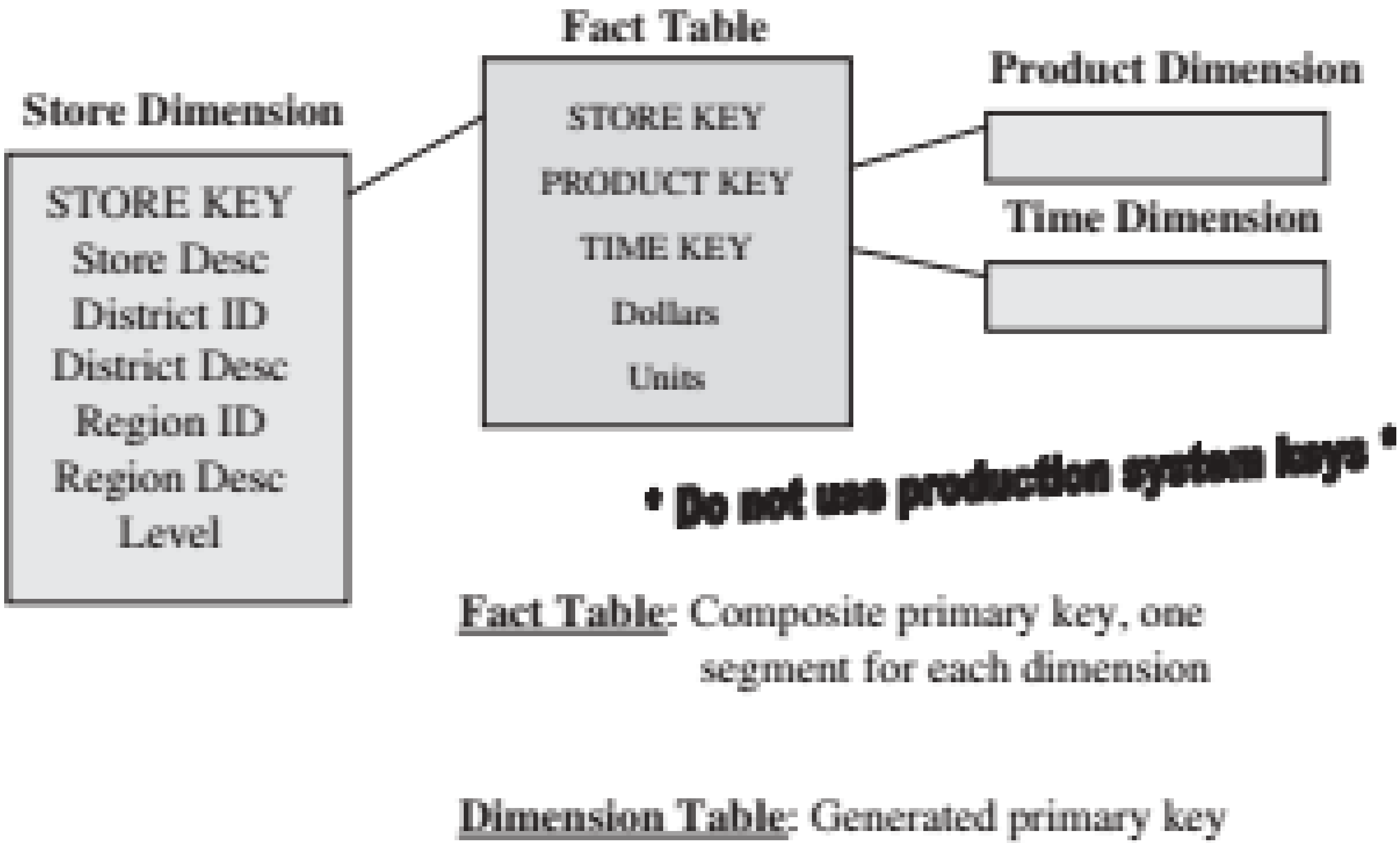


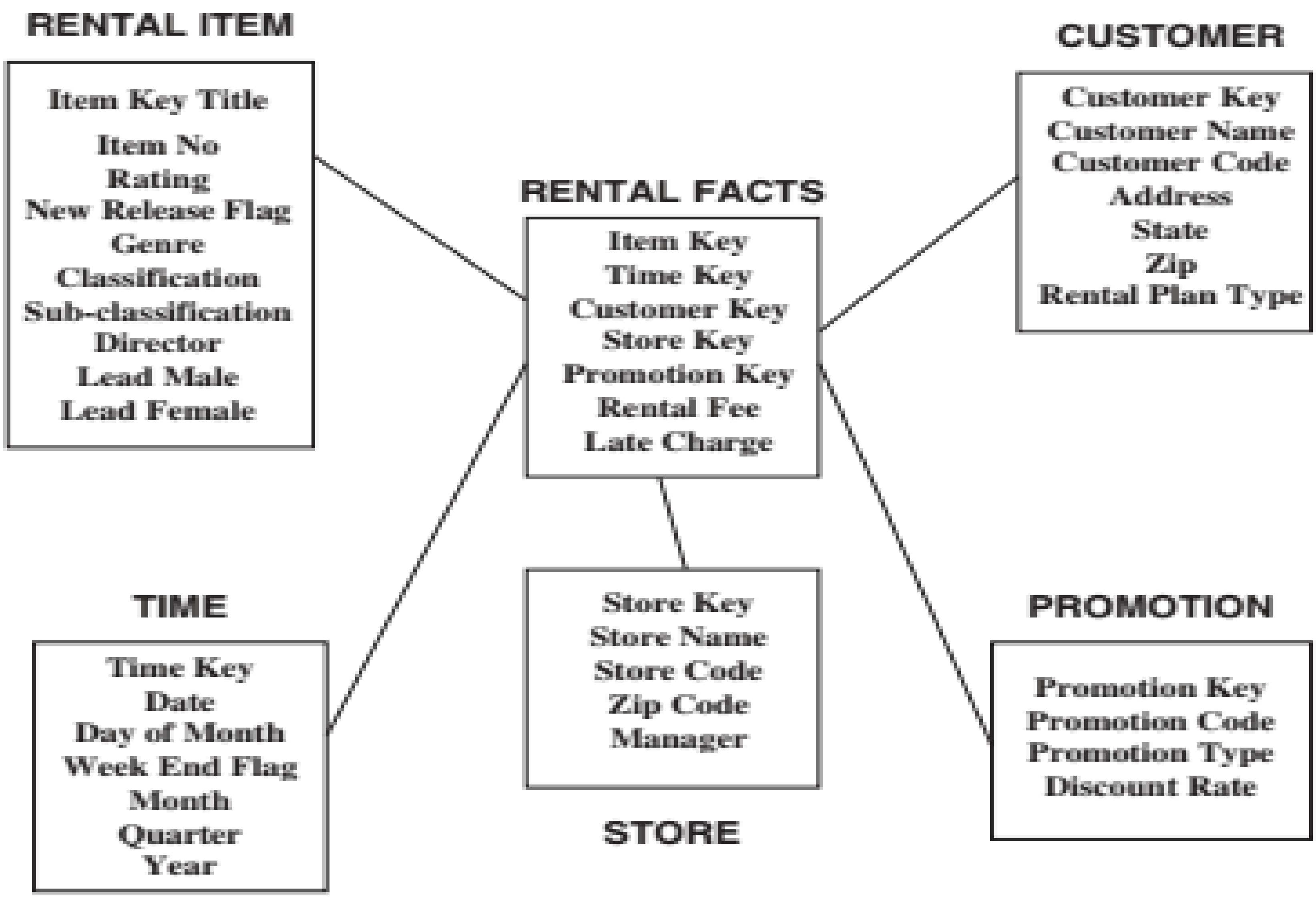
Figure 10-13 The STAR schema keys.

- Primary Keys
  - Each row in a dimension table is identified by a unique value of an attribute designated as the **primary key of the dimension**.
  - In a product dimension table, the primary key identifies each product uniquely.

- There are two general principles to be applied when choosing primary keys for dimension tables.
  - The first principle is derived from the problem caused when the product began to be stored in a different warehouse. In other words, **the product key in the operational system has built-in meanings**.
  - Some positions in the operational system product key indicate the warehouse and some other positions in the key indicate the product category. These are built-in meanings in the key.
  - The first principle to follow is: avoid built-in meanings in the primary key of the dimension tables.

- Let us reexamine the primary keys for the fact tables. There are three options:
  - A single **compound primary key** whose length is the total length of the keys of the individual dimension tables.
  - Under this option, in addition to the compound **primary key**, the foreign keys must also be kept in the fact table as additional attributes. This option increases the size of the fact table.

- A concatenated primary key that is the concatenation of all the primary keys of the dimension tables. Here you need not keep the primary keys of the dimension tables as additional attributes to serve as foreign keys. The individual parts of the primary keys themselves will serve as the foreign keys.
- A generated primary key independent of the keys of the dimension tables**. In addition to the generated primary key, the foreign keys must also be kept in the fact table as additional attributes**. This option also increases the size of the fact table.



© Dr.Ujwala Bharambe

Figure 10-15 STAR schema example: video rental.

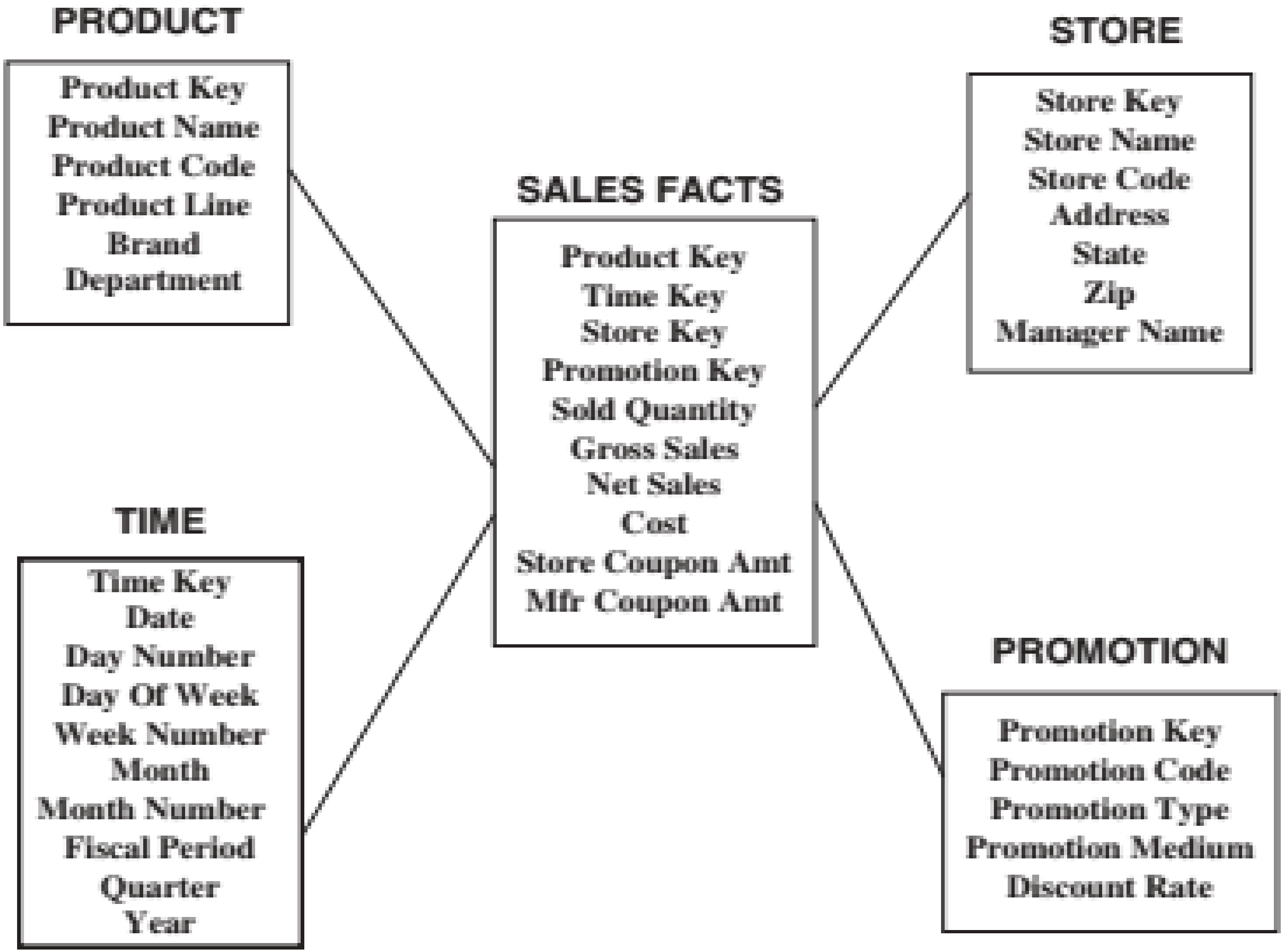


Figure 10-16 STAR schema example: supermarket.

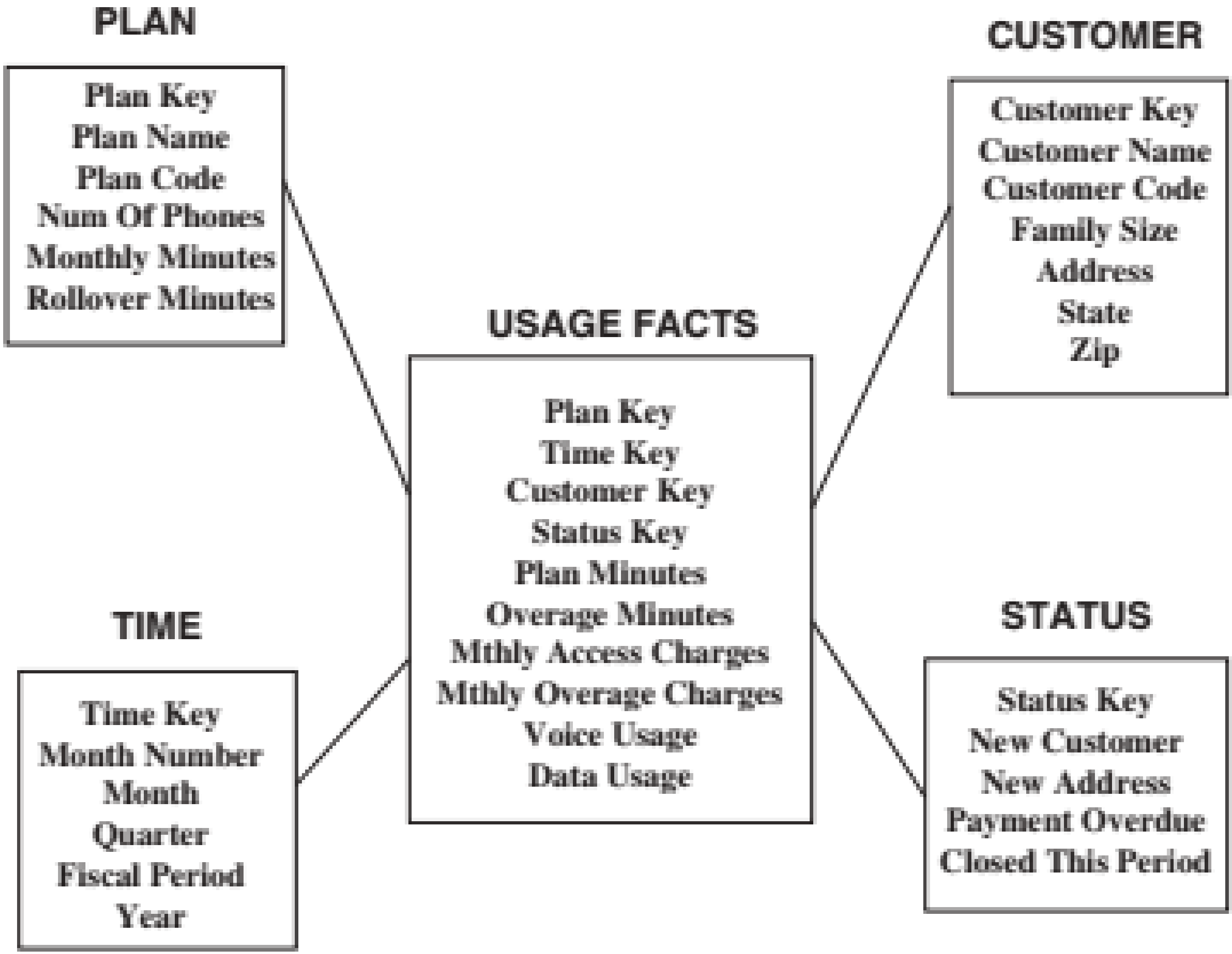


Figure 10-17 STAR schema example: wireless phone service.

Summary

- The **entity-relationship modeling technique is not suitable** for data warehouses; the dimensional modeling technique is appropriate.
- The STAR schema used for **data design** is a relational model consisting of fact and dimension tables.
- The fact table contains the **business metrics or measurements**; the dimension tables contain the business dimensions. Hierarchies within each dimension table are used for drilling down to lower levels of data.
- STAR schema advantages are that it is easy for users to understand optimizes navigation, is most suitable for query processing, and enables specific performance schemes.

Warehouse Models & Operators

- Data Models
  - relations
  - stars & snowflakes
  - cubes
- Operators
  - slice & dice
  - roll-up, drill down
  - pivoting
  - other

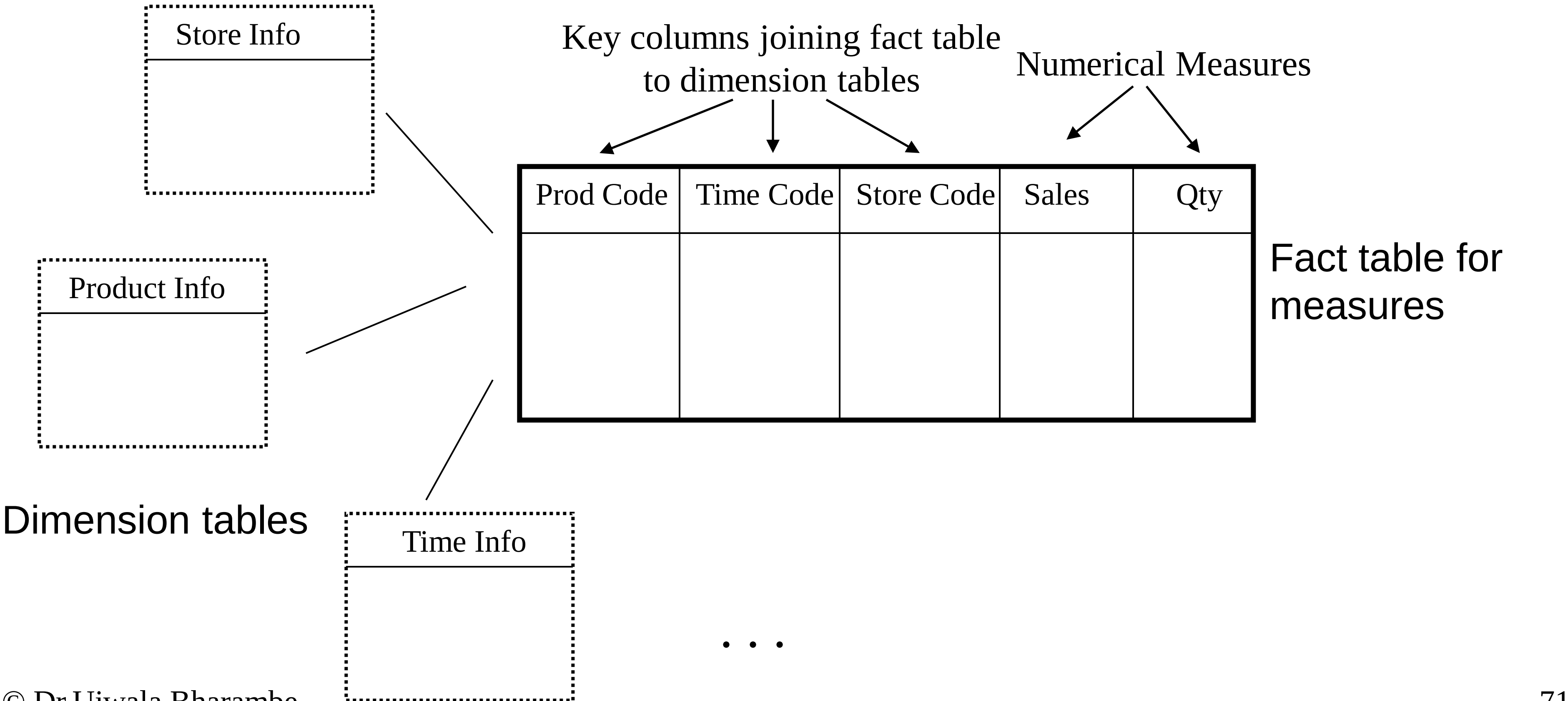
Multi-Dimensional Data

- Measures - numerical (and additive) data being tracked in business, can be analyzed and examined
- Dimensions - business parameters that define a transaction, relatively static data such as lookup or reference tables
- Example: Analyst may want to view <u>***sales***</u> data (measure) by <u>*geography*</u>, by <u>*time*</u>, and by <u>*product*</u> (dimensions)

The Multi-Dimensional Model

“Sales by product line over the past six months”

“Sales by store between 1990 and 1995”



Multidimensional Modeling

- Multidimensional modeling is a technique for structuring data around the business concepts
- ER models describe "entities" and "relationships"
- Multidimensional models describe "measures" and "dimensions"

Dimensional Modeling

- Dimensions are organized into hierarchies
  - E.g., Time dimension: days → weeks → quarters
  - E.g., Product dimension: product → product line → brand
- Dimensions have attributes

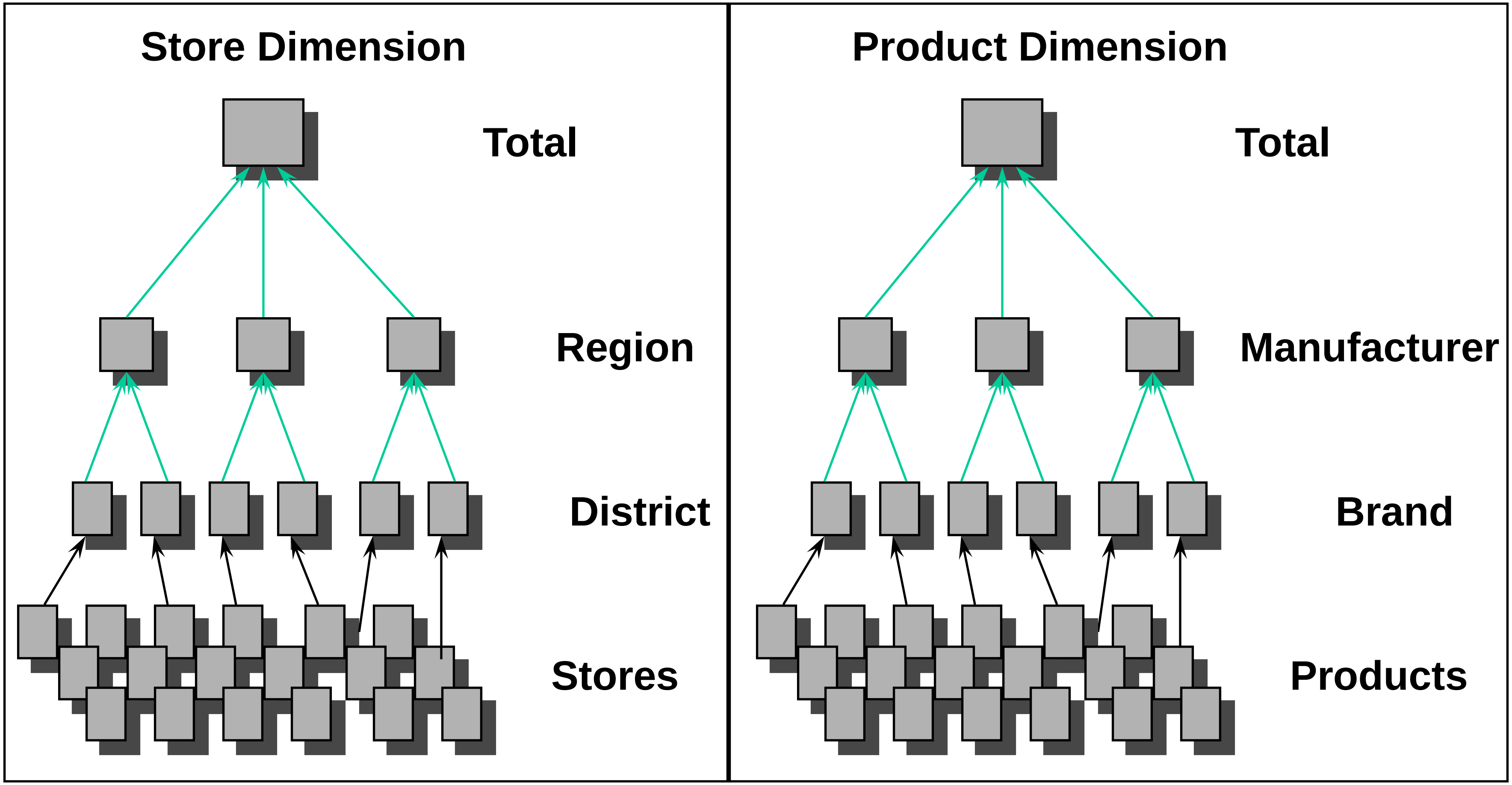
Time

Date
Month
Year

Store

StoreID
City
State
Country
Region

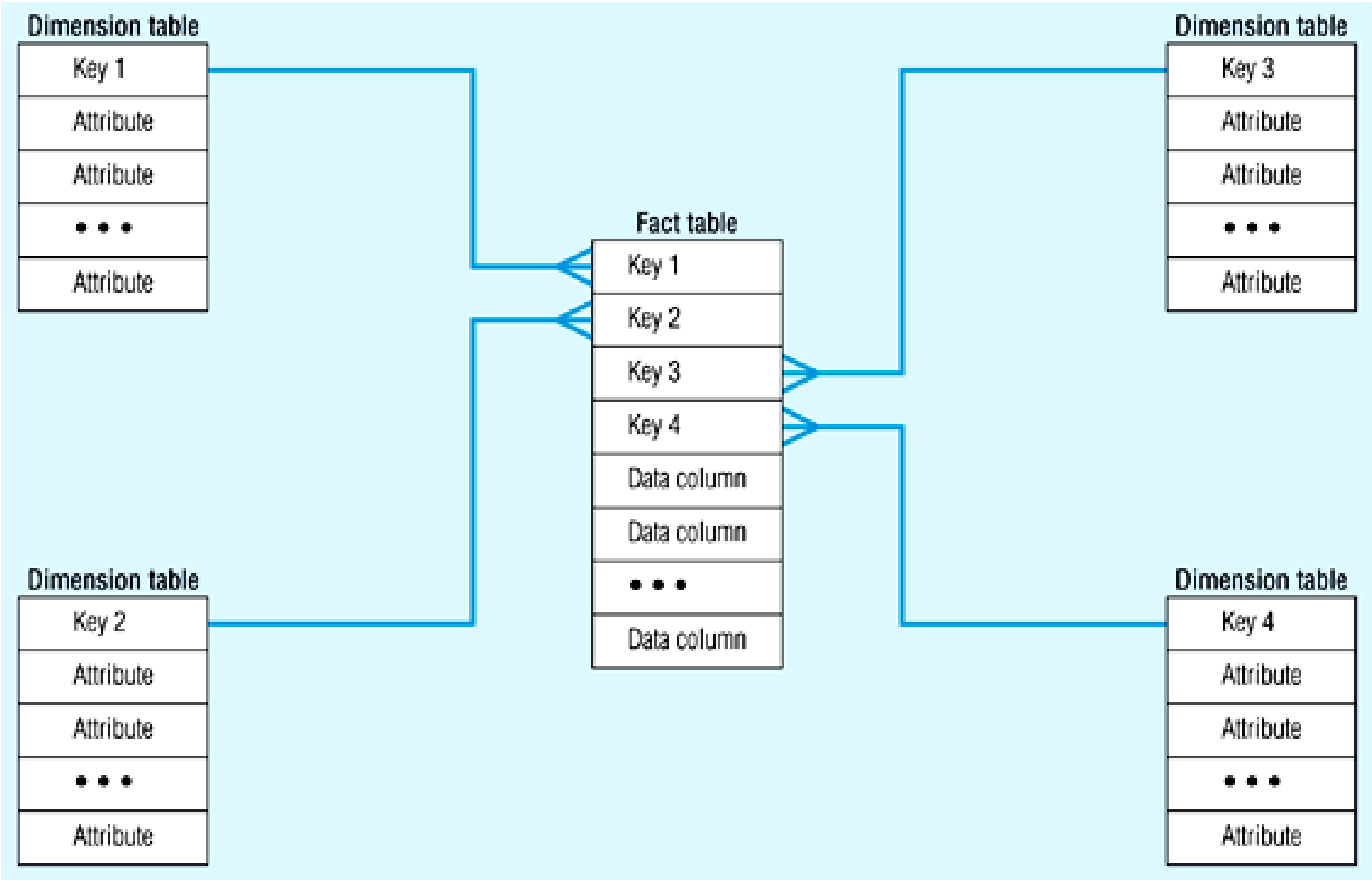
Dimension Hierarchies



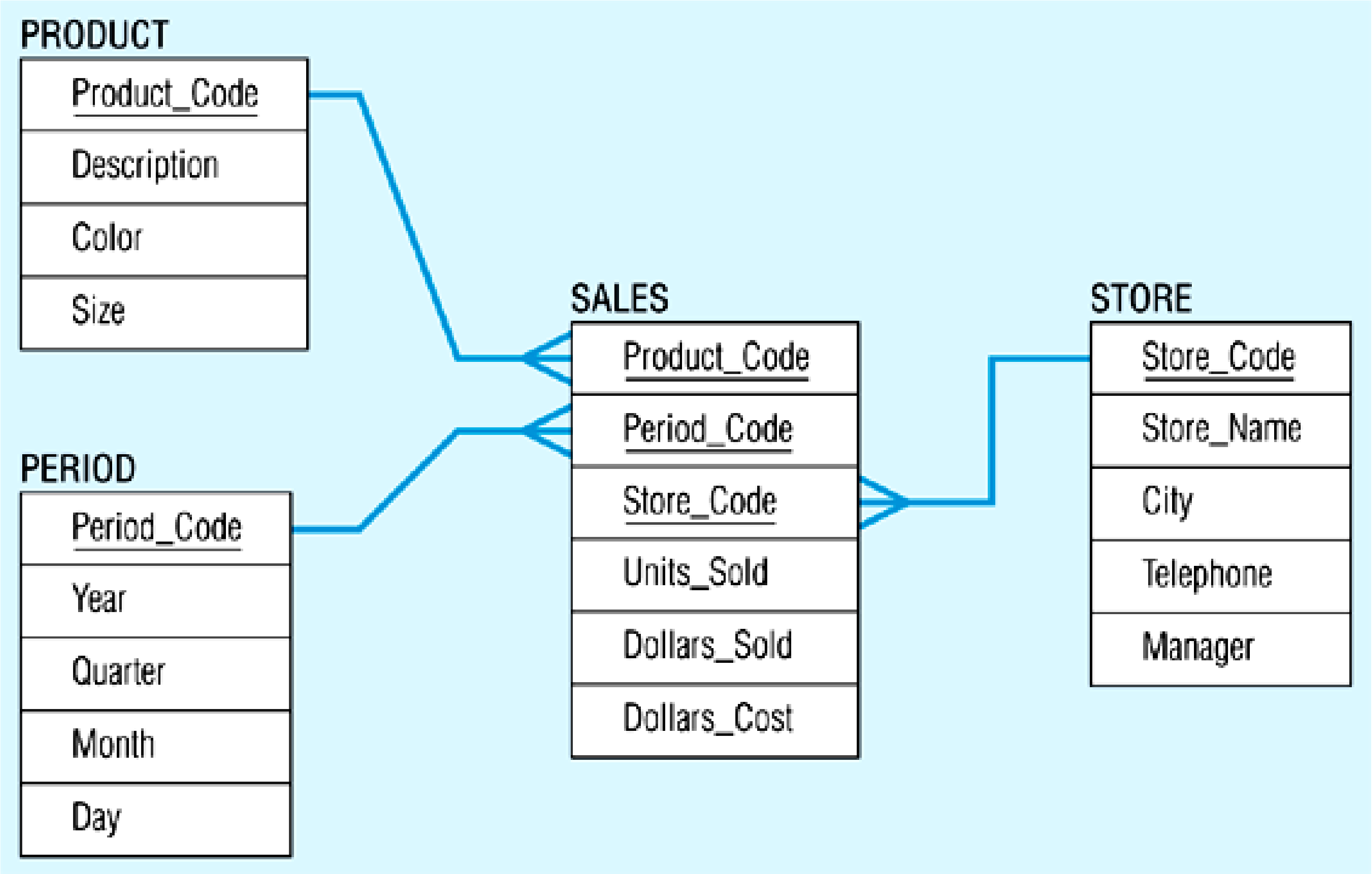
Schema Design

- Most data warehouses use a star schema to represent multi-dimensional model.
- Each dimension is represented by a **dimension table** that describes it.
- A **fact table** connects to all dimension tables with a multiple join. Each tuple in the fact table consists of a pointer to each of the dimension tables that provide its multi-dimensional coordinates and stores measures for those coordinates.
- The links between the fact table in the center and the dimension tables in the extremities form a shape like a star.

Star Schema (in RDBMS)



Star Schema Example



Star Schema with Sample Data

Product

Product_Code	Description	Color	Size
100	Sweater	Blue	40
110	Shoes	Brown	10 1/2
125	Gloves	Tan	M
• • •			

Period

Period_Code	Year	Quarter	Month
001	1999	1	4
002	1999	1	5
003	1999	1	6
• • •			

Sales

Product_Code	Period_Code	Store_Code	Units_Sold	Dollars_Sold	Dollars_Cost
110	002	S1	30	1500	1200
125	003	S2	50	1000	600
100	001	S1	40	1600	1000
110	002	S3	40	2000	1200
100	003	S2	30	1200	750
• • •					

Store

Store_Code	Store_Name	City	Telephone	Manager
S1	Jan's	San Antonio	683-192-1400	Burgess
S2	Bill's	Portland	943-681-2135	Thomas
S3	Ed's	Boulder	417-196-8037	Perry
• • •				

The “Classic” Star Schema

- A relational model with a **one-to-many relationship** between dimension table and fact table.
- A single fact table, with detail and summary data
- Fact table primary key has only one key column per dimension
- Each dimension is a single table, highly denormalized
- **Benefits**: Easy to understand, intuitive mapping between the business entities, easy to define hierarchies, reduces # of physical joins, low maintenance, very simple metadata
- **Drawbacks**: Summary data in the fact table yields poorer performance for summary levels, huge dimension tables a problem

Slowly Changing Dimensions

- Most dimensions are generally constant over time.
- Many dimensions, though not constant over time, change slowly.
- The product key of the source record does not change.
- The description and other attributes change slowly over time.

Slowly Changing Dimensions

- In the source OLTP systems, the new values overwrite the old ones.
- Overwriting of dimension table attributes is not always the appropriate option in a data warehouse.
- The ways changes are made to the dimension tables depend on the types of changes and what information must be preserved in the data warehouse.

© Dr.Ujwala Bharambe

Type1 Changes

- Usually, the changes relate to correction of errors in source systems.
- Sometimes the change in the source system has no significance.
- The old value in the source system needs to be discarded.
- The change in the source system need not be preserved in the data warehouse.

Type1 Changes

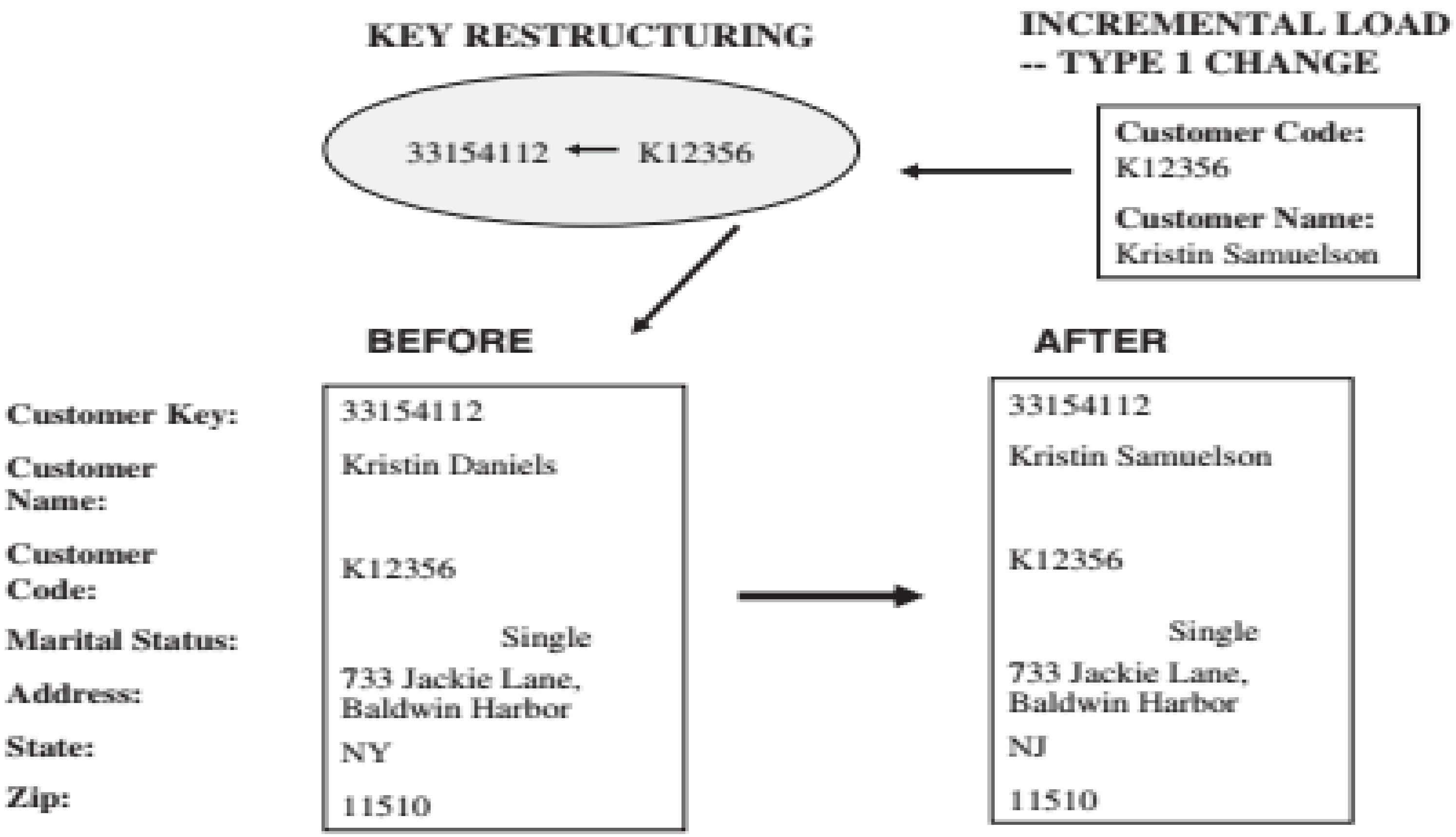


Figure 11-2 The method for applying type 1 changes.

Type 2 Changes

- Add a new dimension table row with the new value of the changed attribute.
- An effective date field may be included in the dimension table.
- There are no changes to the original row in the dimension table.
- The key of the original row is not affected.
- The new row is inserted with a new surrogate key.

A surrogate represents an *entity* in the outside world. The surrogate is internally generated by the system but is nevertheless visible to the user or application

© Dr.Ujwala Bharambe

Type 2 Changes

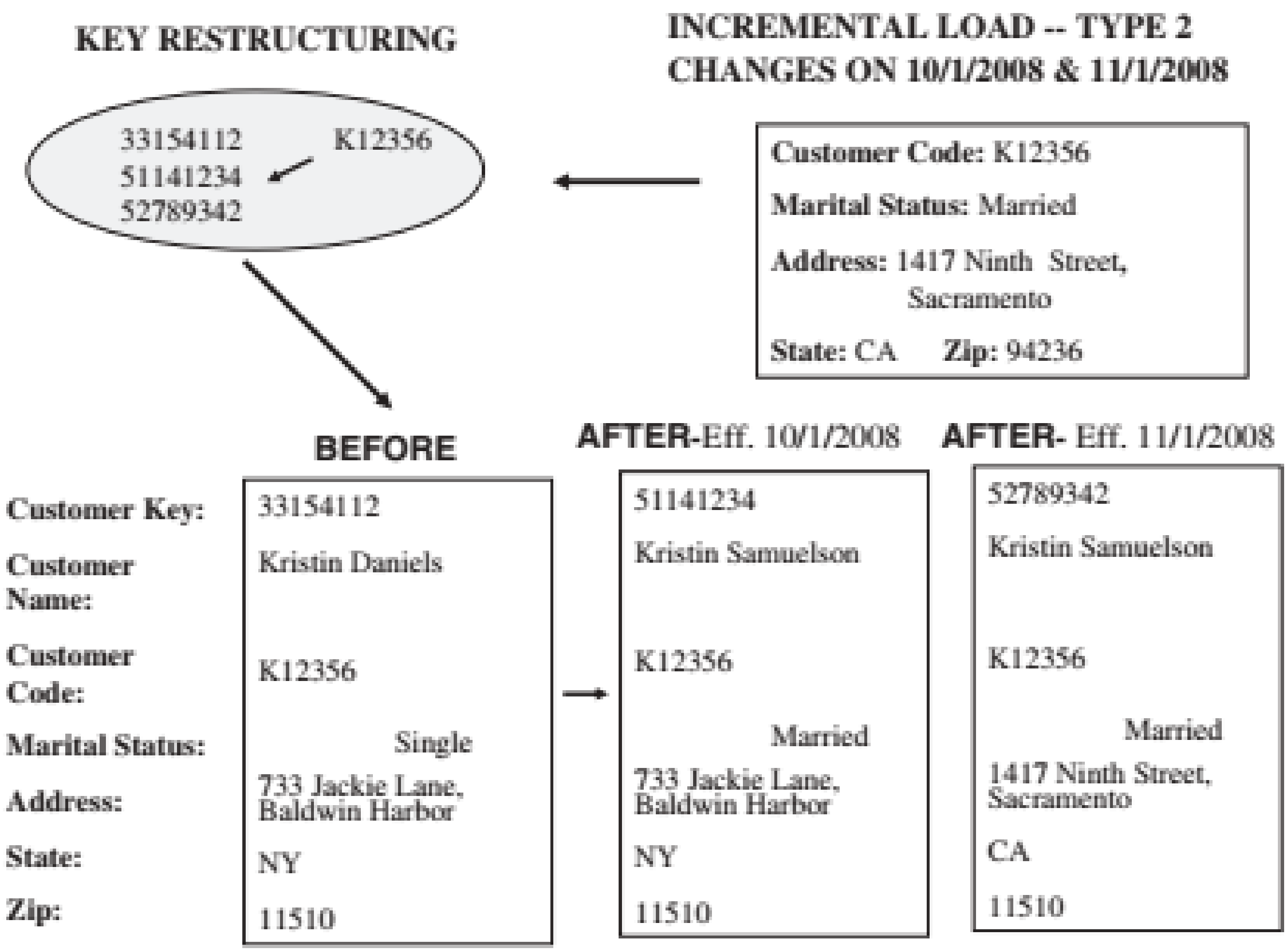


Figure 11-3 The method for applying type 2 changes.

Type 3 Changes

- They usually relate to "soft" or tentative changes in the source systems.
- There is a need **to keep track of history** with old and new values of the changed attribute.
- They are used to **compare performances** across the transition.
- They provide the **ability to track** forward and backward.

© Dr.Ujwala Bharambe

Type 3 Changes

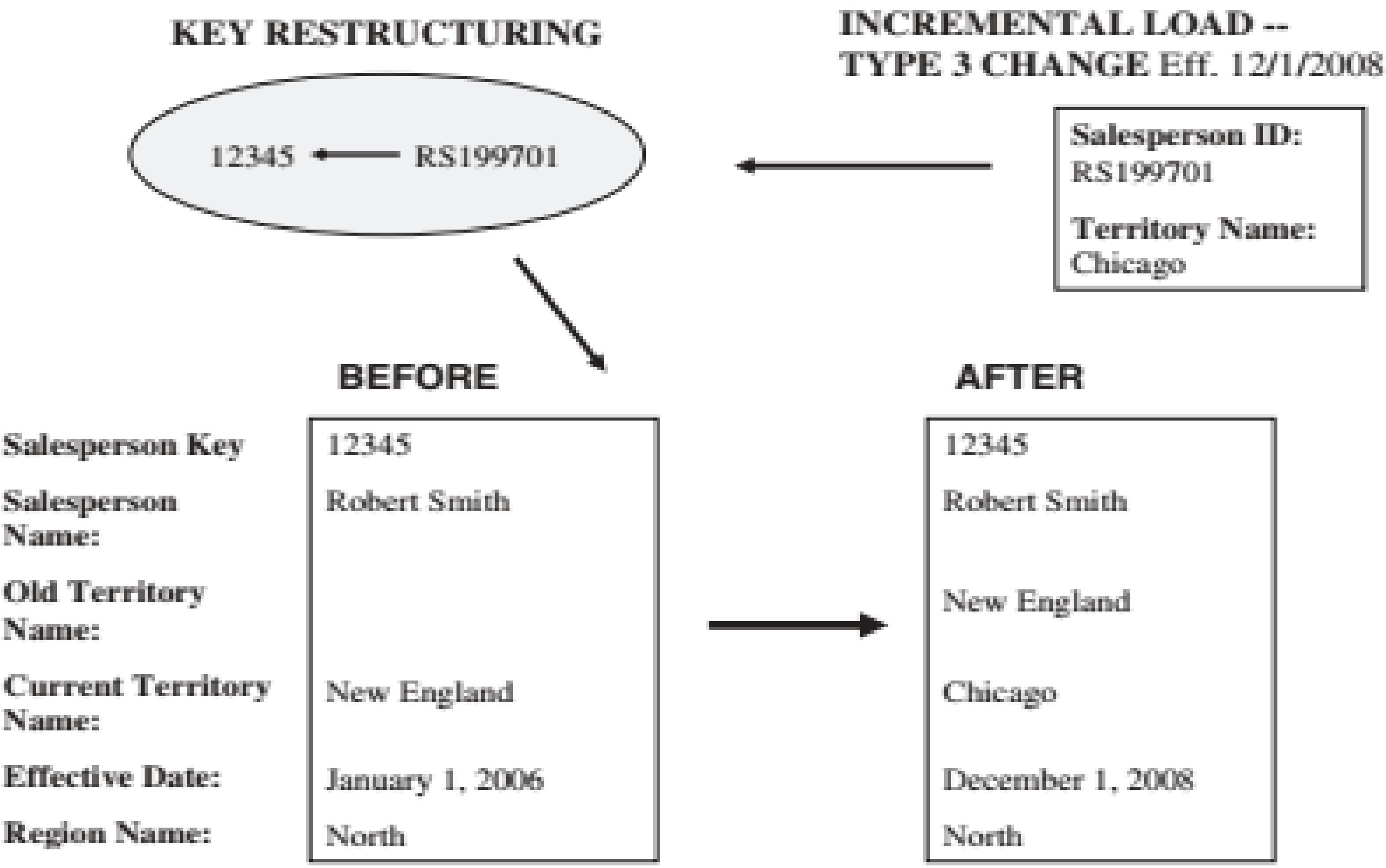


Figure 11-4 Applying type 3 changes.

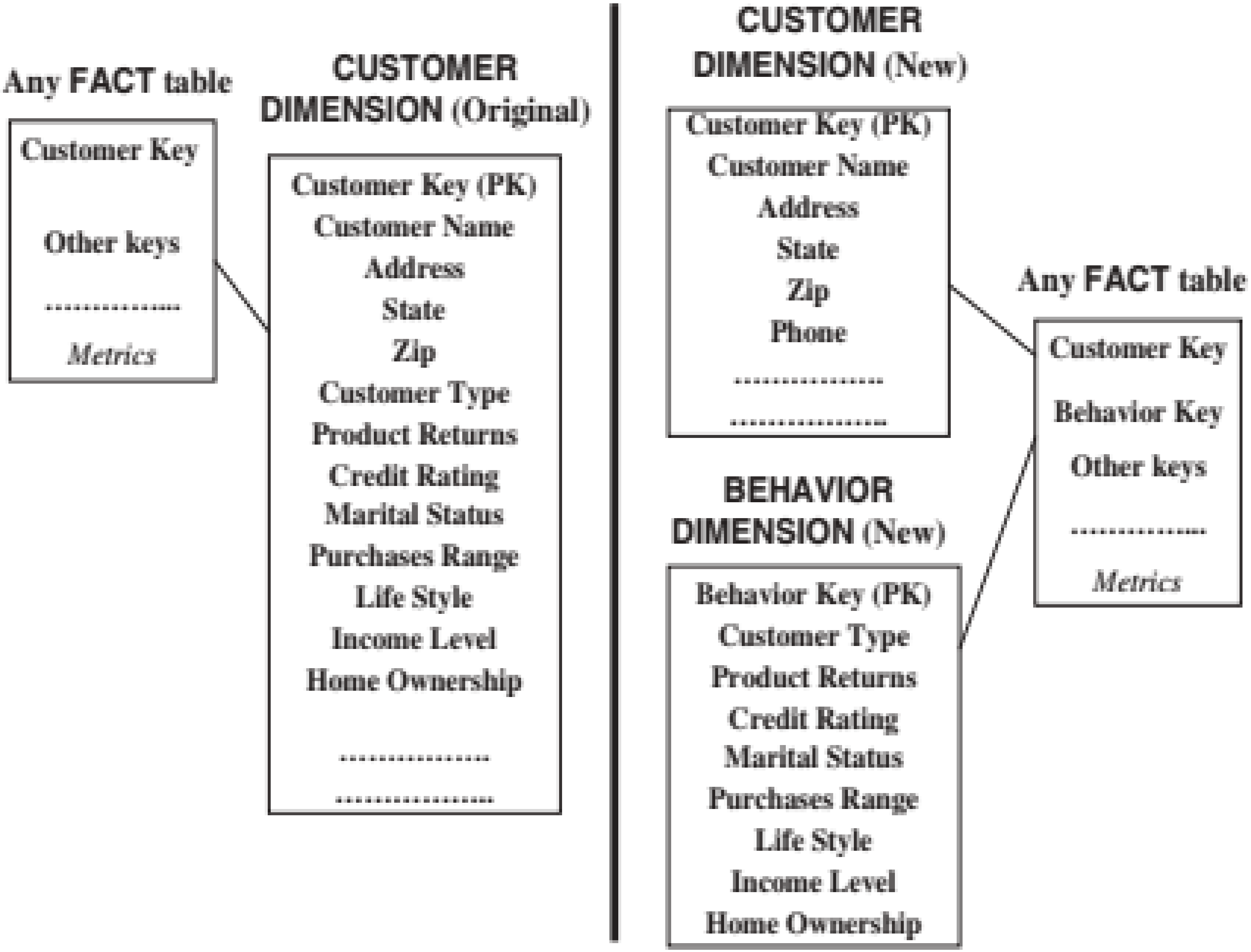
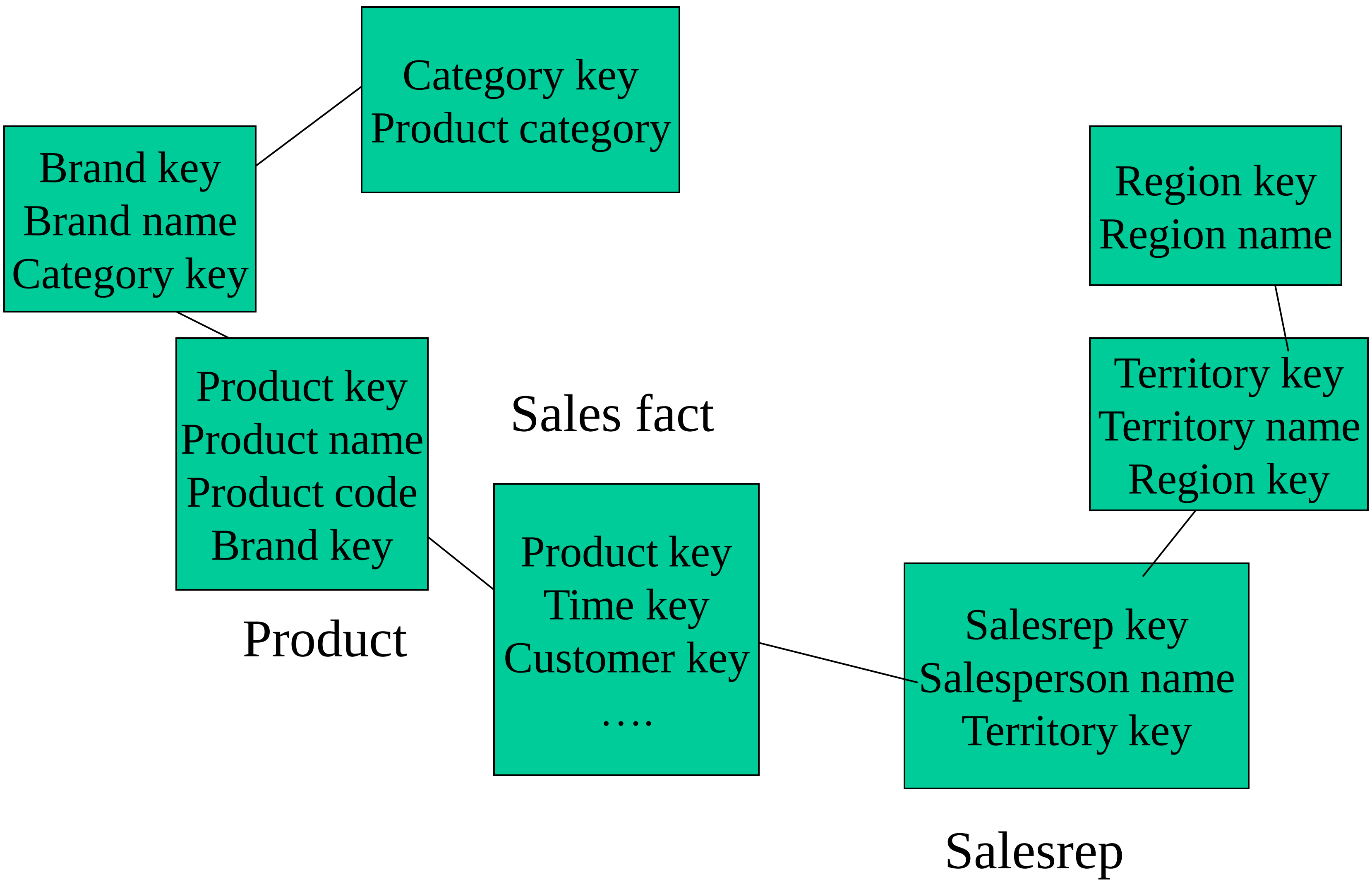


 Figure 11-6 Dividing a large, rapidly changing dimension table.

Snowflake Schema

- Snowflake schema is a type of star schema but a more complex model.
- “Snowflaking” is a method of normalizing the dimension tables in a star schema.
- The normalization eliminates redundancy.
- The result is more complex queries and reduced query performance.

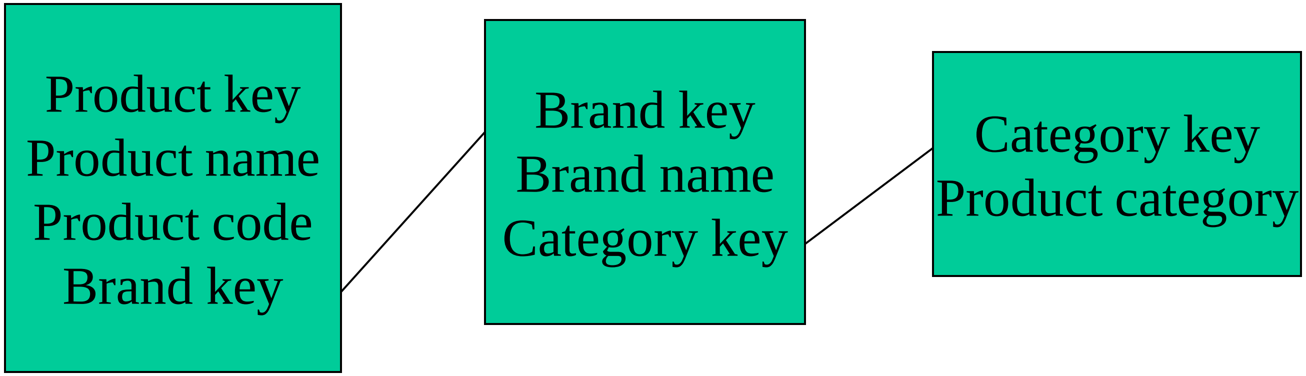
Sales: Snowflake Schema



© Dr.Ujwala Bharambe

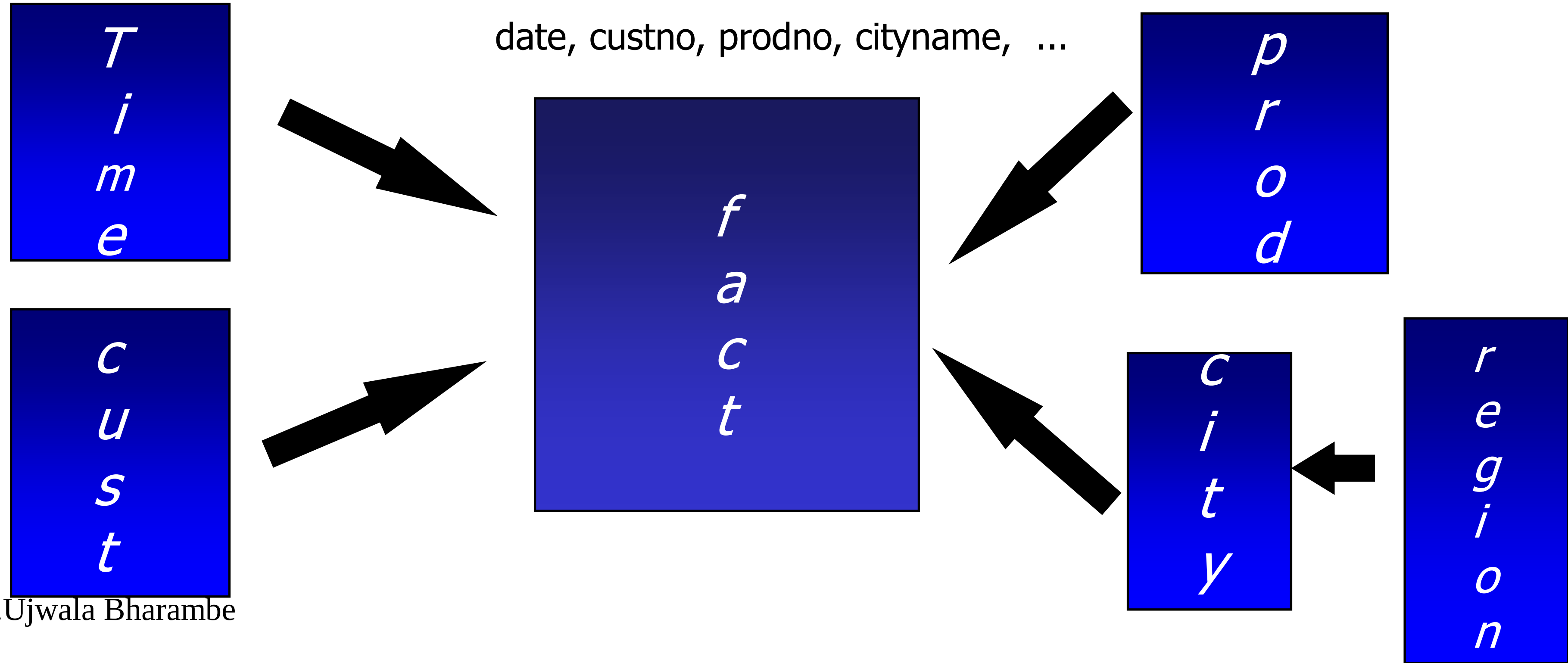
Snowflaking

- The attributes with low cardinality in each original dimension table are removed to form separate tables. These new tables are linked back to the original dimension table through artificial keys.



Snowflake schema

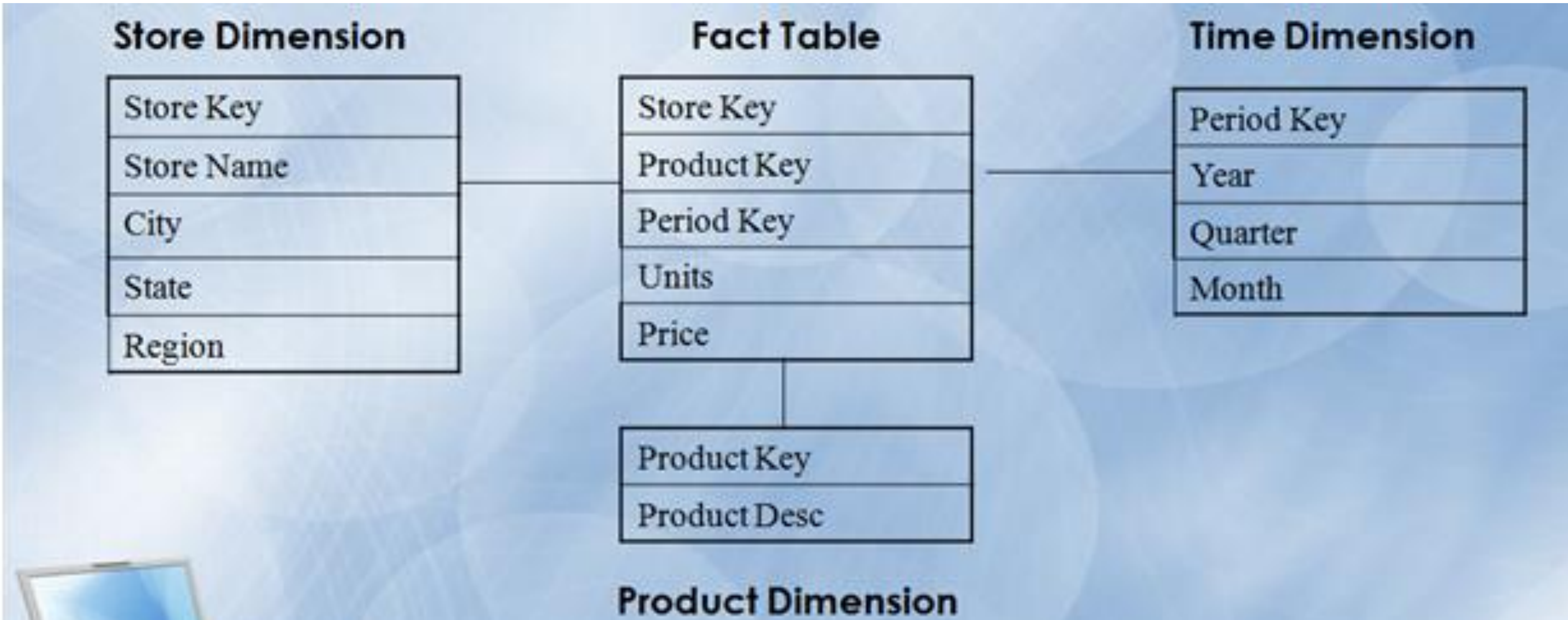
- Represent dimensional hierarchy directly by normalizing tables.
- Easy to maintain and saves storage



Snowflake Schema

- Advantages:
  - Small saving in storage space
  - Normalized structures are easier to update and maintain
- Disadvantages:
  - Schema less intuitive and end-users are put off by the complexity
  - Ability to browse through the contents difficult
  - Degrade query performance because of additional joins

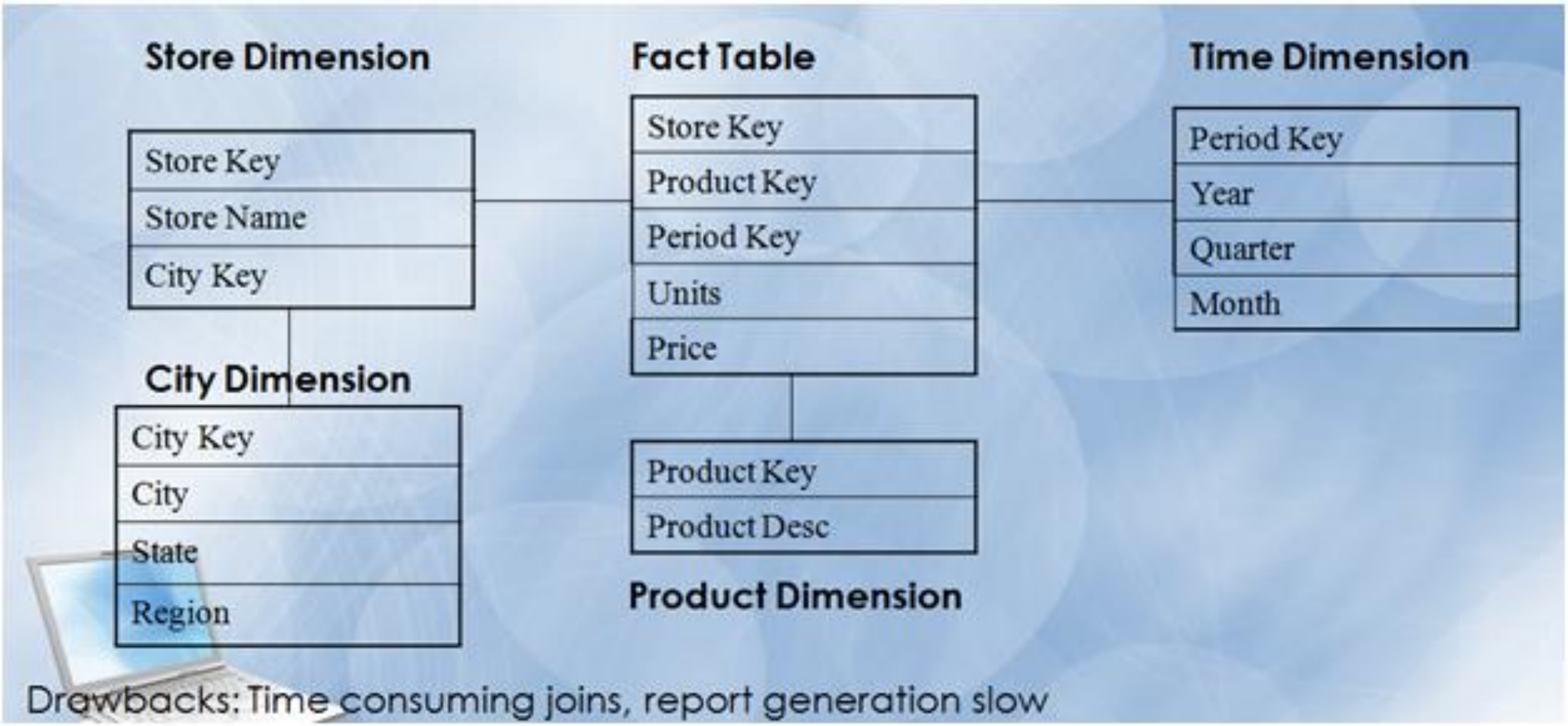
- Star Schema
- A single large **central fact table** and one table for each dimension
- Every fact points to one tuple in each of the dimensions and has additional attributes
- Does not capture hierarchies directly.



Benefits: Easy to understand, easy to define hierarchies, reduces no. of physical joins.

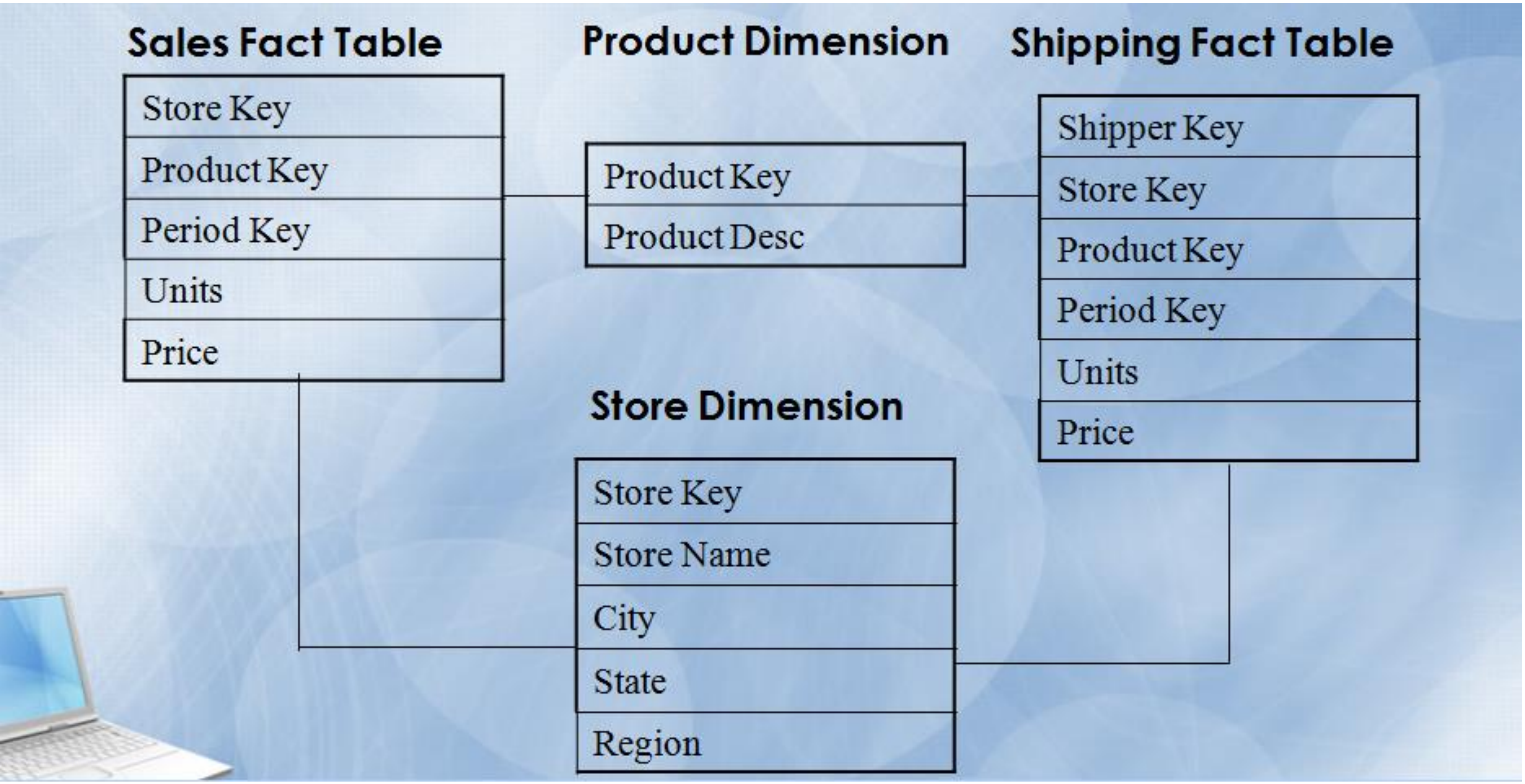
- Snowflake Schema
  - Variant of star schema model.
  - A single, large and central fact table and one or more tables for each dimension.
  - Dimension tables are normalized split dimension table data into additional tables.

Snowflake Schema



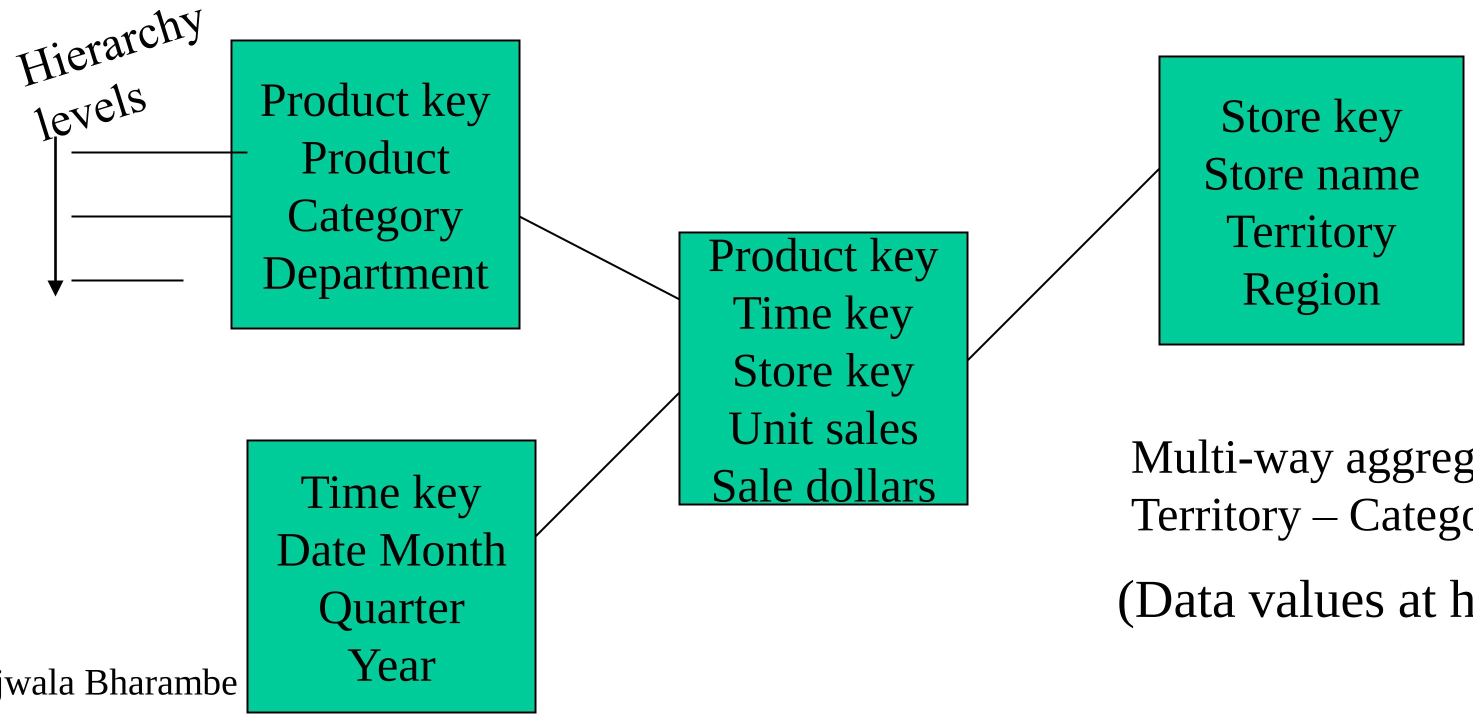
- Fact Constellation:
  - – Multiple fact tables share dimension tables.
  - – This schema is viewed as collection of stars hence called galaxy schema or fact constellation.
  - – Sophisticated application requires such schema.

Fact Constellation



Aggregating Fact Tables

- Aggregate fact tables are summaries of the most granular data at higher levels along the dimension hierarchies.

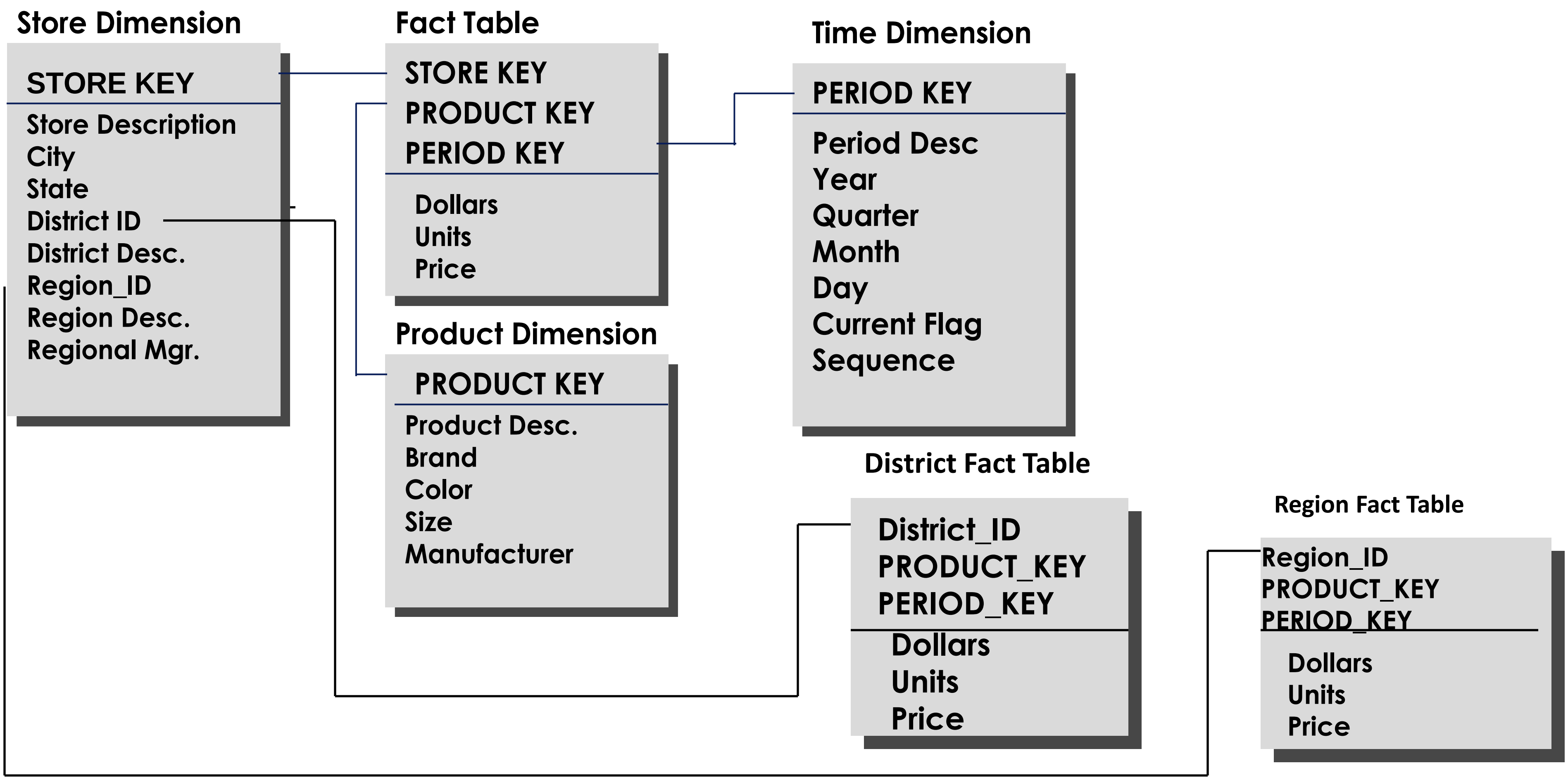


Multi-way aggregates:
Territory – Category – Month

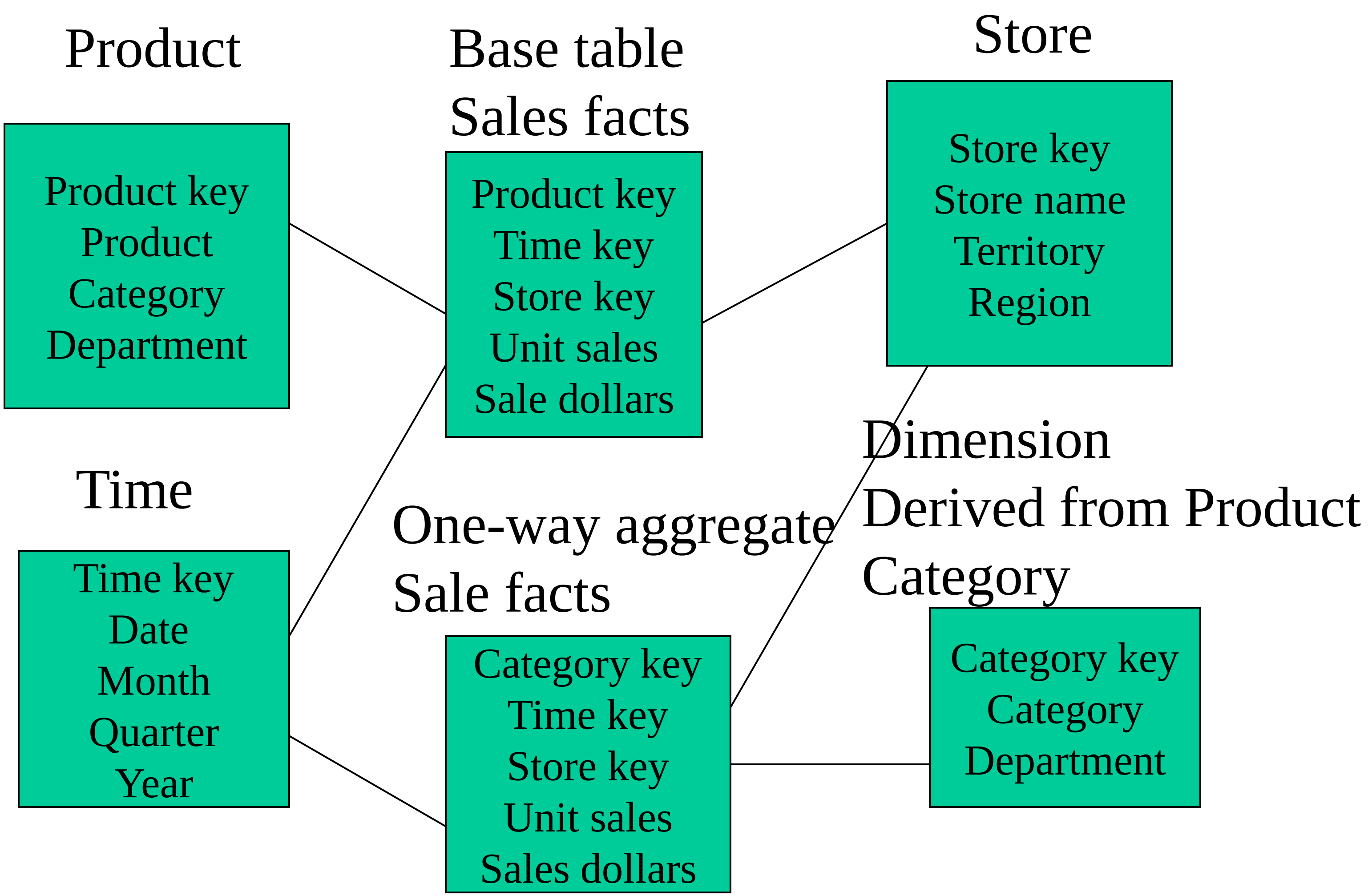
(Data values at higher level)

© Dr.Ujwala Bharambe

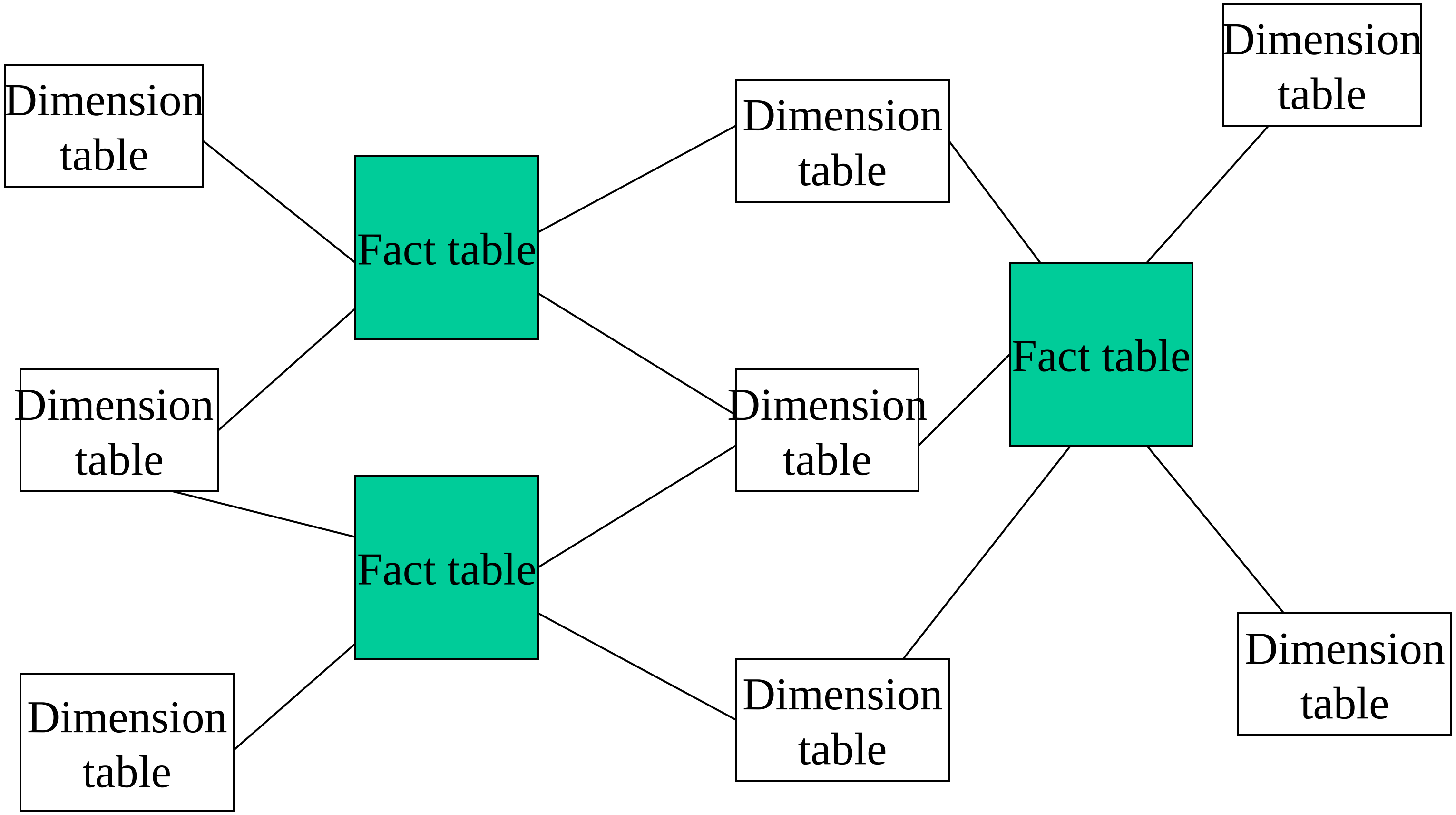
The “Fact Constellation” Schema



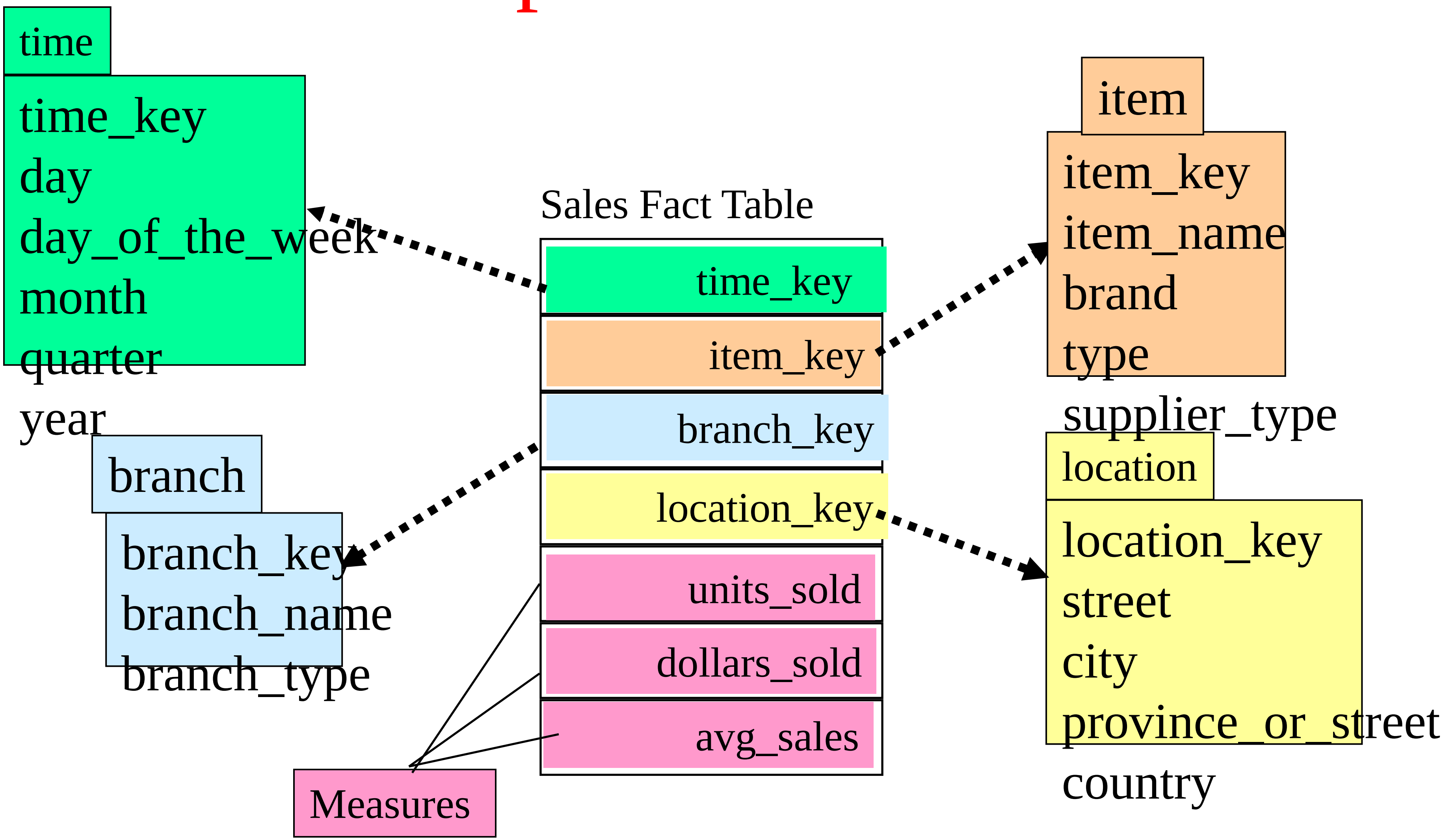
Aggregate Fact Tables



Families of Stars

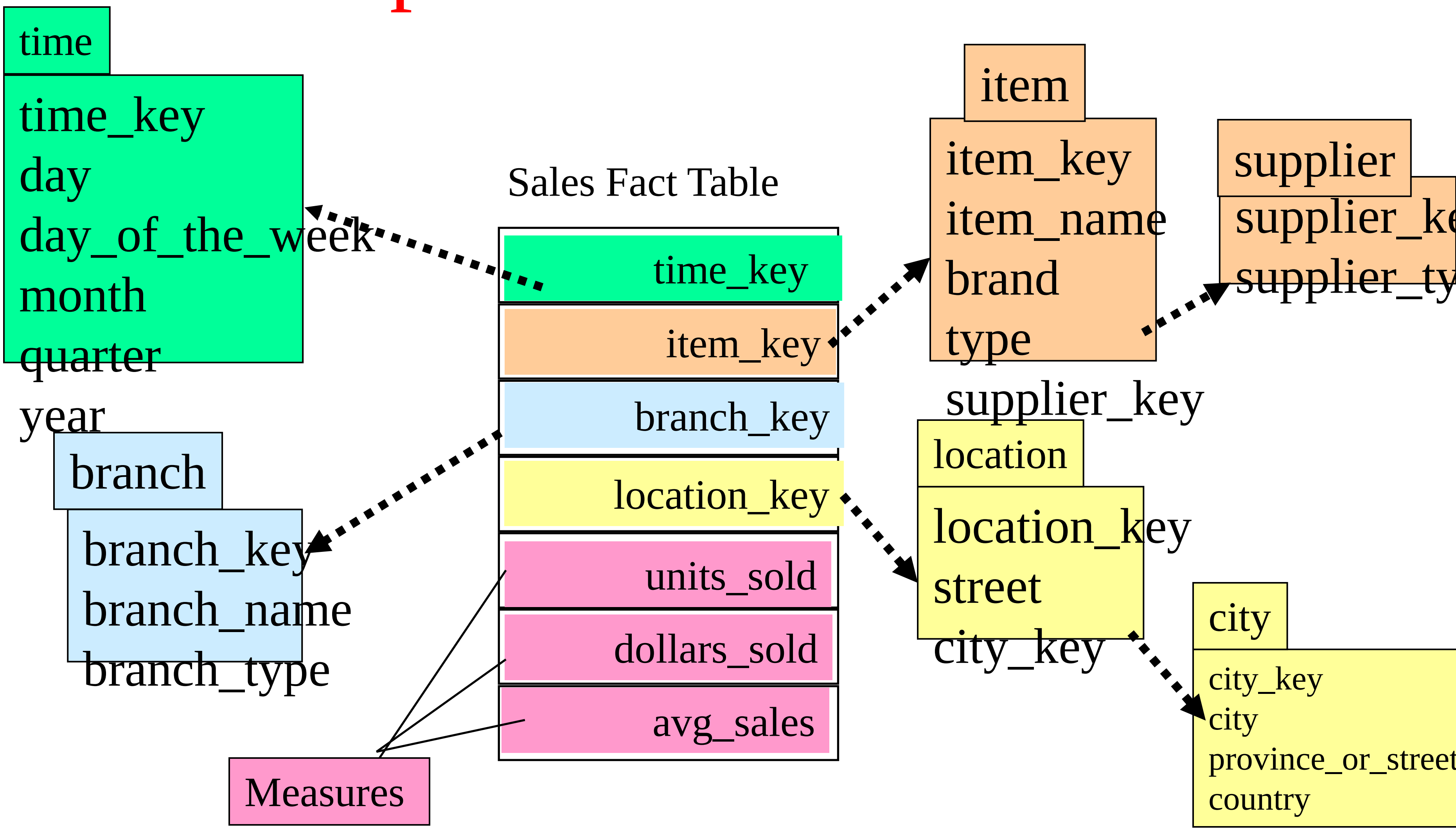


Example of Star Schema



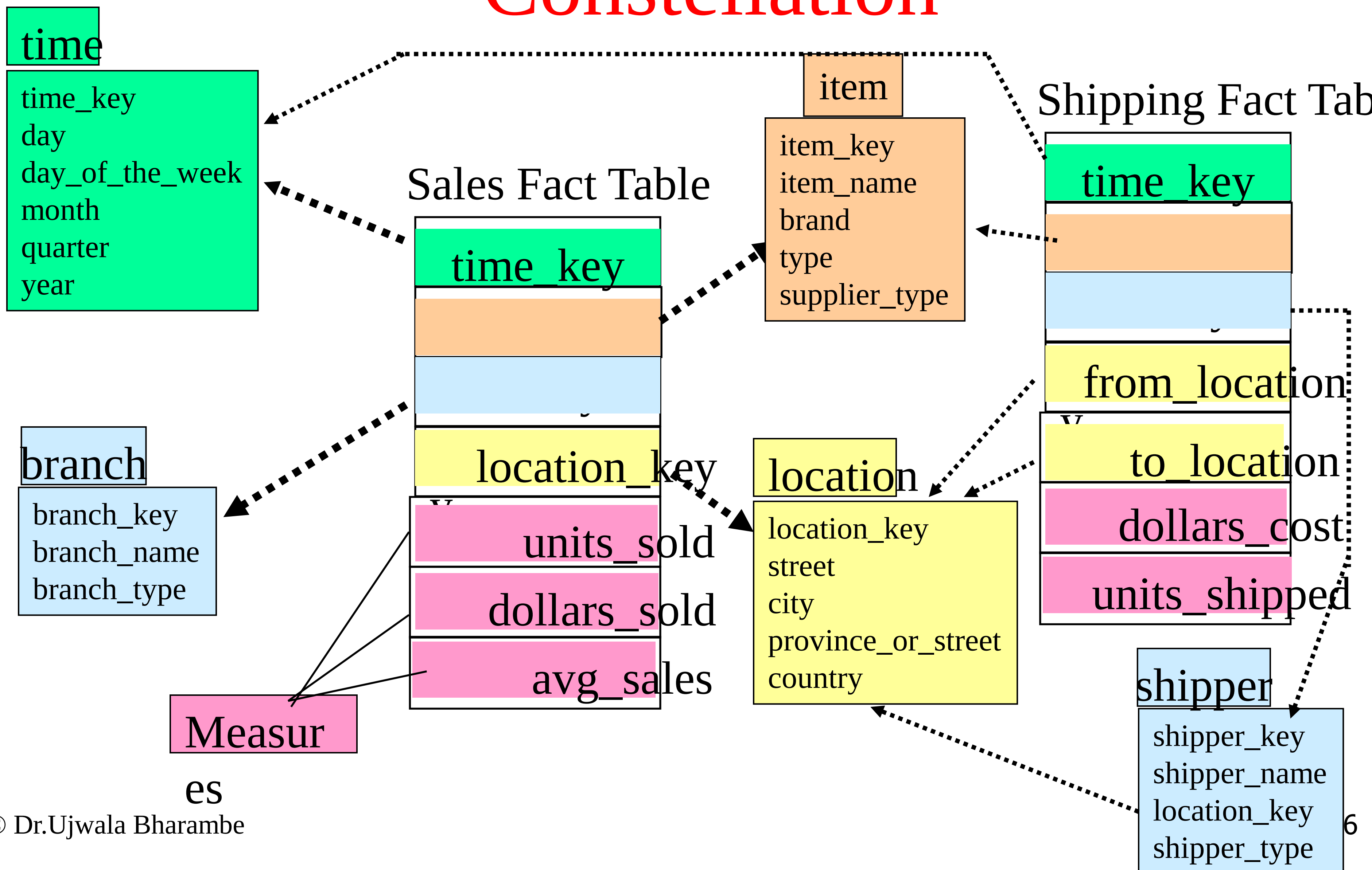
© Dr.Ujwala Bharambe

Example of Snowflake Schema



© Dr.Ujwala Bharambe

Example of Fact Constellation



What is the Best Design?

- Performance benchmarking can be used to determine what is the best design.
- Snowflake schema: easier to maintain dimension tables when dimension tables are very large (reduce overall space). It is not generally recommended in a data warehouse environment.
- Star schema: more effective for data cube browsing (less joins): can affect performance.

© Dr.Ujwala Bharambe

Aggregates

- Add up amounts for day 1
- In SQL: SELECT sum(amt) FROM SALE
  WHERE date = 1

sale	prodId	storeId	date	amt
	p1	s1	1	12
	p2	s1	1	11
	p1	s3	1	50
	p2	s2	1	8
	p1	s1	2	44
	p1	s2	2	4

⇨ 81

Aggregates

- Add up amounts by day
- In SQL: SELECT date, sum(amt) FROM SALE
  GROUP BY date

sale	prodId	storeId	date	amt
	p1	s1	1	12
	p2	s1	1	11
	p1	s3	1	50
	p2	s2	1	8
	p1	s1	2	44
	p1	s2	2	4

ans	date	sum
	1	81
	2	48

Another Example

- Add up amounts by day, product
- In SQL: SELECT date, sum(amt) FROM SALE
  GROUP BY date, prodId

sale	prodId	storeId	date	amt
	p1	s1	1	12
	p2	s1	1	11
	p1	s3	1	50
	p2	s2	1	8
	p1	s1	2	44
	p1	s2	2	4

sale	prodId	date	amt
	p1	1	62
	p2	1	19
	p1	2	48

rollup →

← drill-down

Aggregates

- Operators: sum, count, max, min, median, ave
- "Having" clause
- Using dimension hierarchy
  - average by region (within store)
  - maximum by month (within date)

© Dr.Ujwala Bharambe

Data Cube

Fact table view:

sale	prodId	storeId	amt
	p1	s1	12
	p2	s1	11
	p1	s3	50
	p2	s2	8

Multi-dimensional cube:

	s1	s2	s3
p1	12		50
p2	11	8	

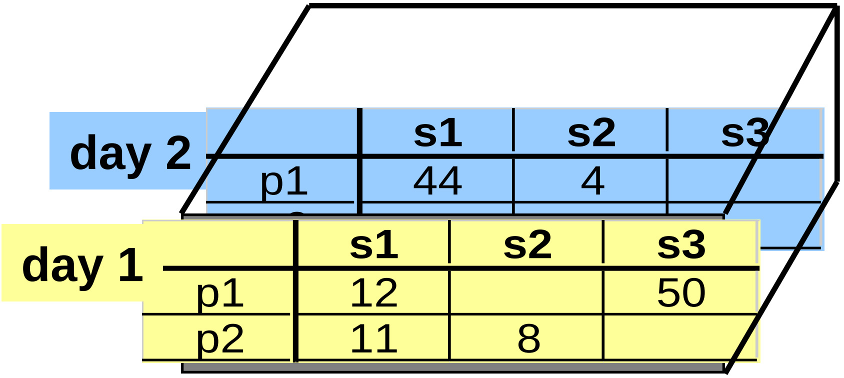
dimensions = 2

3-D Cube

Fact table view:

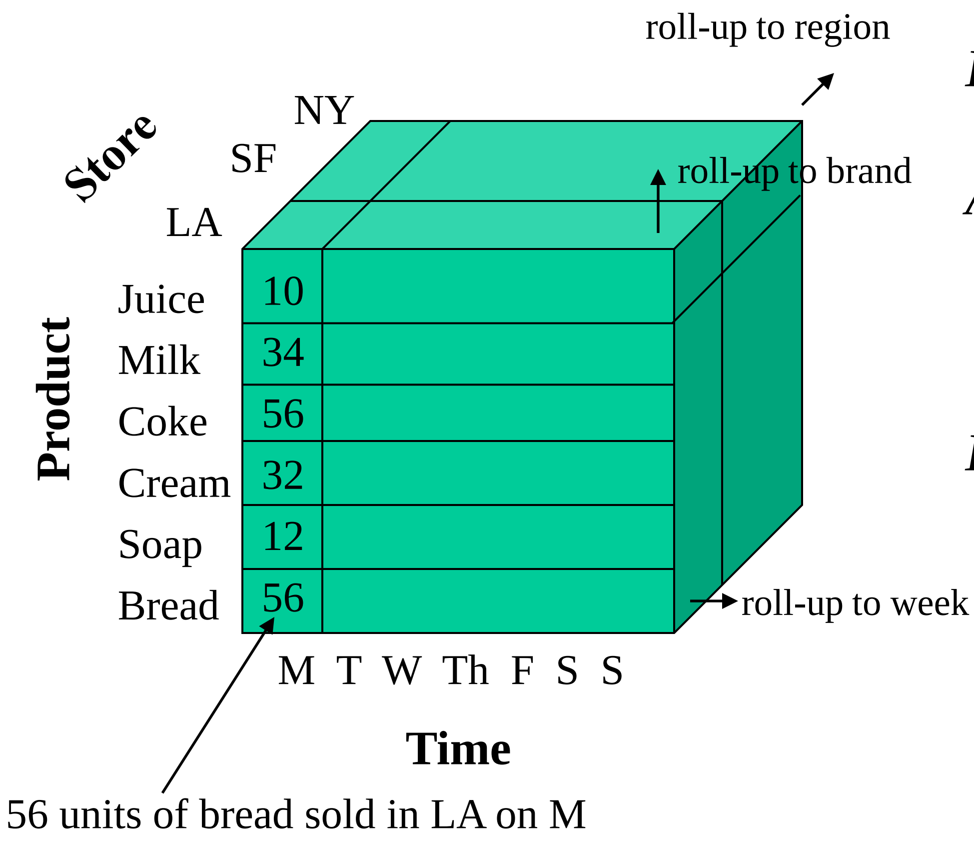
sale	prodId	storeId	date	amt
	p1	s1	1	12
	p2	s1	1	11
	p1	s3	1	50
	p2	s2	1	8
	p1	s1	2	44
	p1	s2	2	4

Multi-dimensional cube:



dimensions = 3

Example



Dimensions:

Time, Product, Store

Attributes:

Product (upc, price, …)

Store …

…

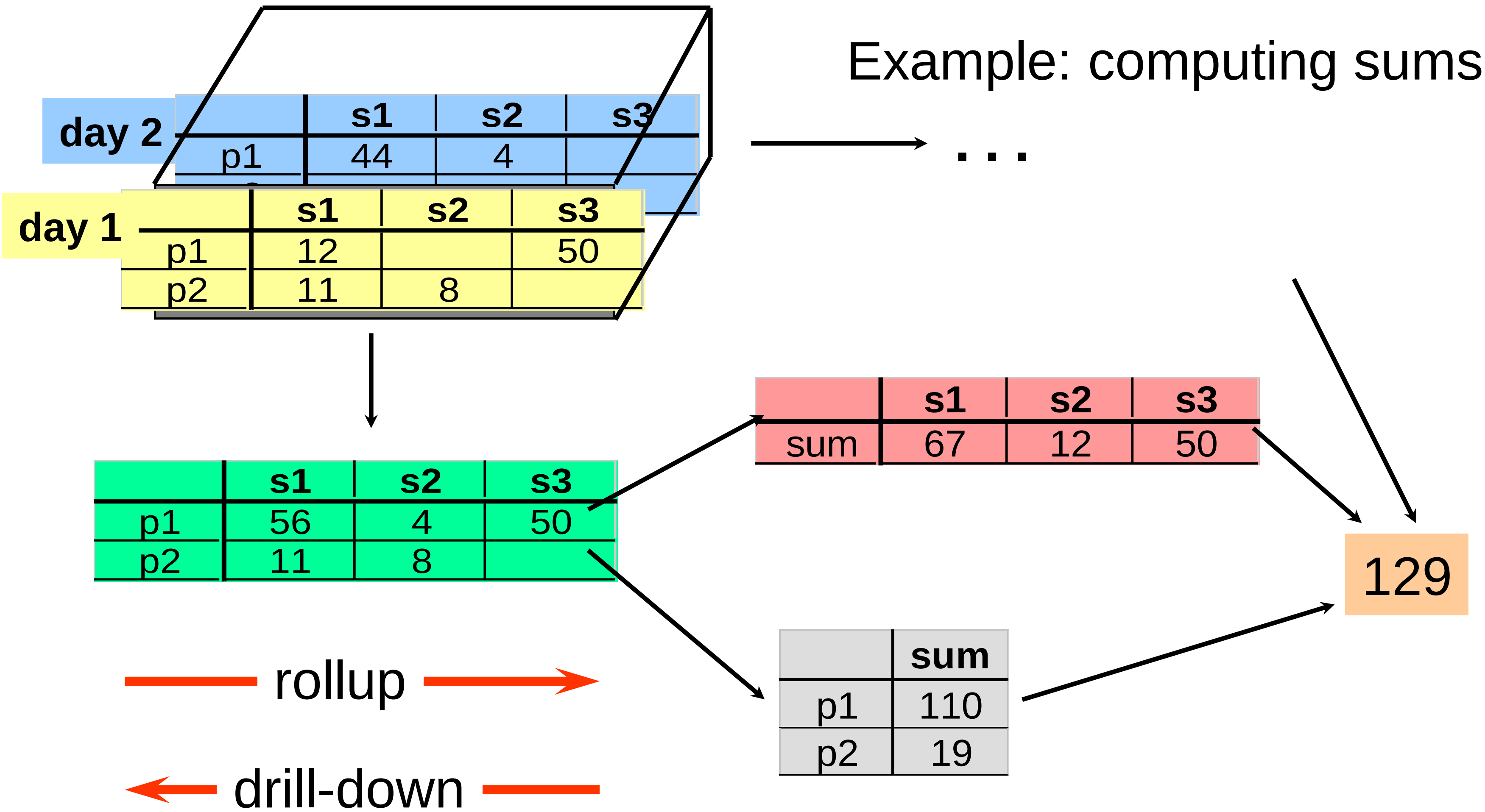
Hierarchies:

Product → Brand → …

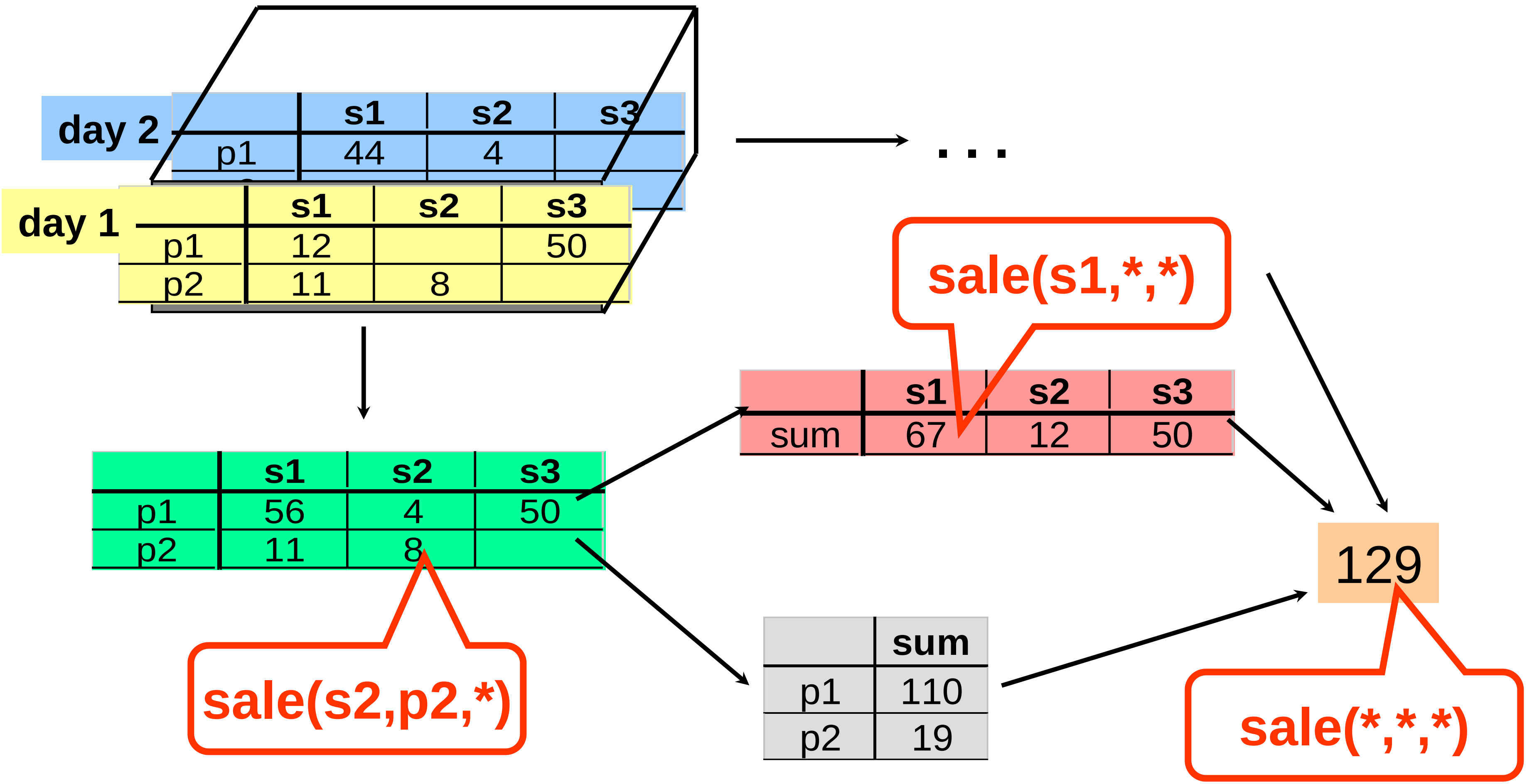
Day → Week → Quarter

Store → Region → Country

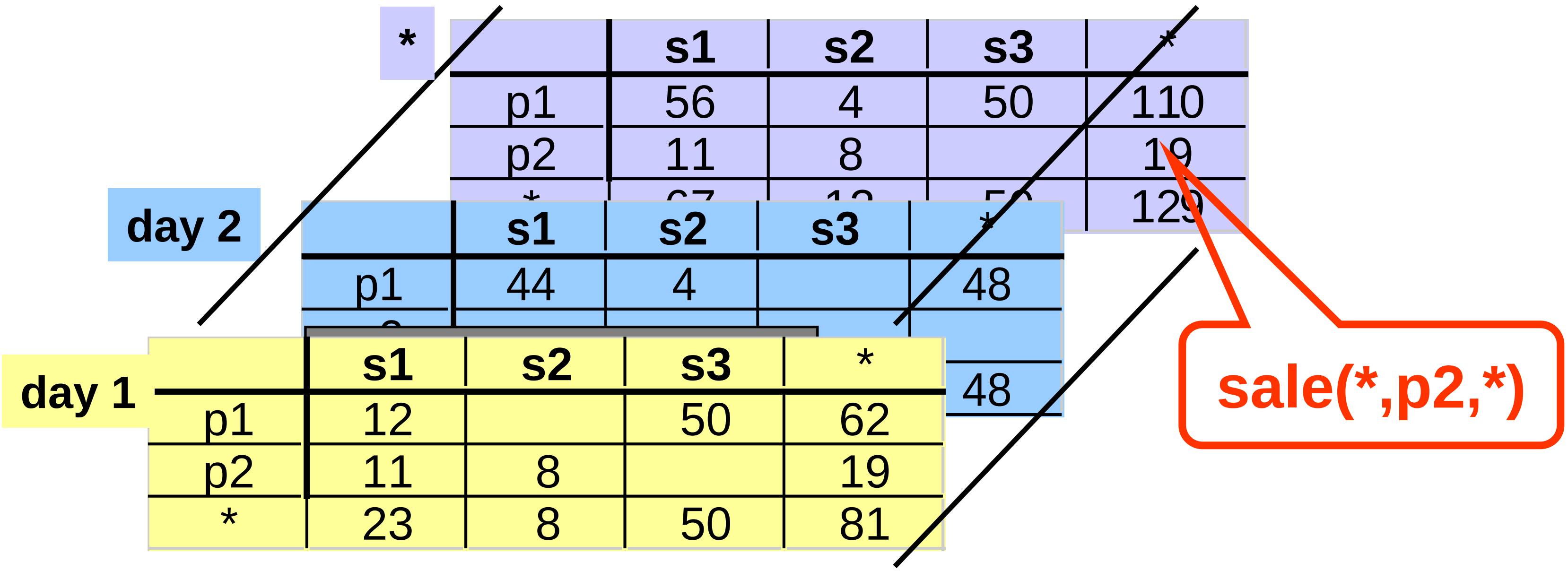
Cube Aggregation: Roll-up



Cube Operators for Roll-up



Extended Cube



Aggregation Using Hierarchies

day 2

	s1	s2	s3
p1	44	4	

day 1

	s1	s2	s3
p1	12		50
p2	11	8	

	region A	region B
p1	56	54
p2	11	8

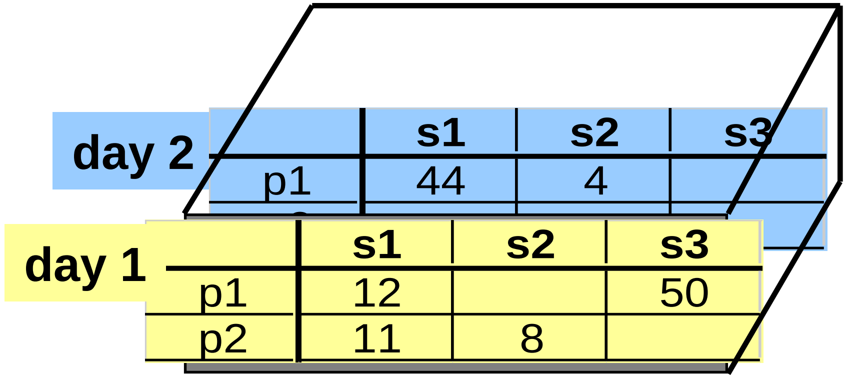
store

region

country

(store s1 in Region A;
stores s2, s3 in Region B)

Slicing



TIME = day 1

	s1	s2	s3
p1	12		50
p2	11	8	

Slicing & Pivoting

	Sales ($ millions)			
	Products	Time		
		d1	d2	
Store s1	Electronics	$5.2		
	Toys	$1.9		
	Clothing	$2.3		
	Cosmetics	$1.1		
Store s2	Electronics	$8.9		
	Toys	$0.75		
	Clothing	$4.6		
	Cosmetics	$1.5		

	Sales ($ millions)			
	Products	d1		
		Store s1	Store s2	
Store s1	Electronics	$5.2	$8.9	
	Toys	$1.9	$0.75	
	Clothing	$2.3	$4.6	
	Cosmetics	$1.1	$1.5	
Store s2	Electronics			
	Toys			
	Clothing			

Summary of Operations

- Aggregation (roll-up)
  - aggregate (summarize) data to the next higher dimension element
  - e.g., total sales by city, year $\rightarrow$ total sales by region, year
- Navigation to detailed data (drill-down)
- Selection (slice) defines a subcube
  - e.g., sales where city ='Gainesville' and date = '1/15/90'
- Calculation and ranking
  - e.g., top 3% of cities by average income
- Visualization operations (e.g., Pivot)
- Time functions
  - e.g., time average