DATA WAREHOUSE DESIGN – DIMENSIONAL MODELING

Quách Đình Hoàng

Outline

- Dimensional Modeling Concepts
- Dimensional Modeling Process
- Types of Fact Tables
- Types of Facts
- Slowly Changing Dimensions

Dimensional Modeling Concepts

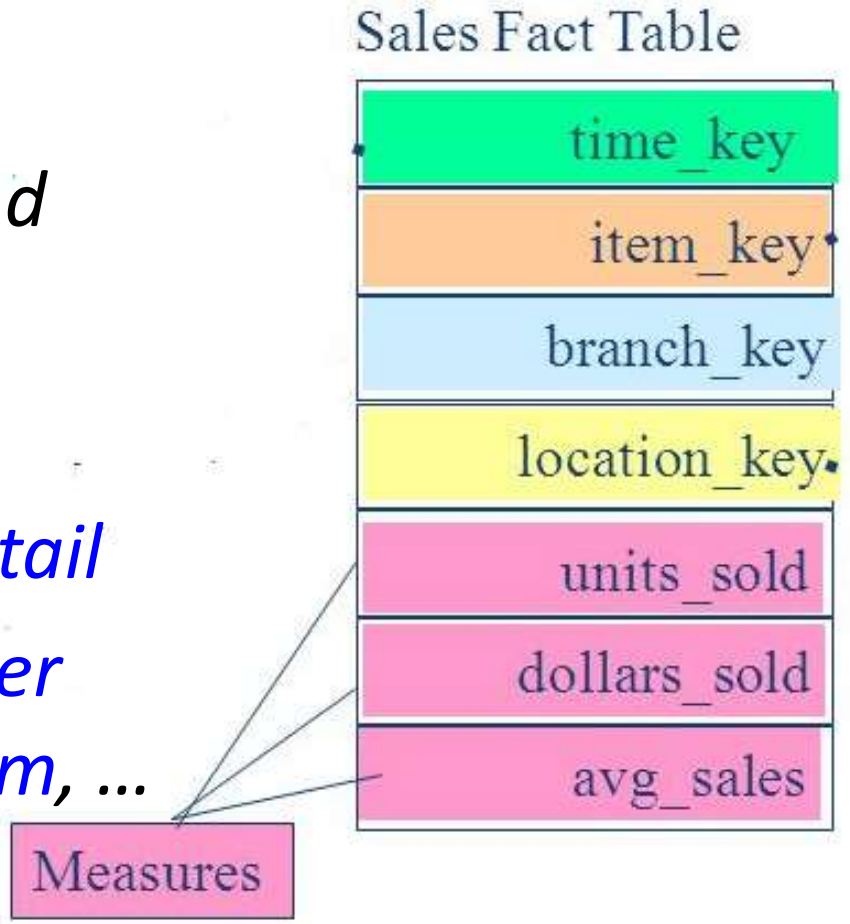
- Dimensions
- Facts
- Dimensional Model or Star Schema
- Conformed Dimensions
- Data Warehouse Bus Matrix

Dimensions

- Set of attributes (columns) related to a subject/object
  - *Who, what, when, where, why, how*
  - *Ex: Product, Customer, Date, Vendor, …*
- Each dimension row is a unique occurrence
  - *One row per product, customer, day, …*
- **Dimension attributes**
  - *Report labels and query constraints*
  - *"By" words and "where" clauses*
  - *Verbose descriptive attributes, in addition to codes*
  - *Hierarchical relationships*

Facts

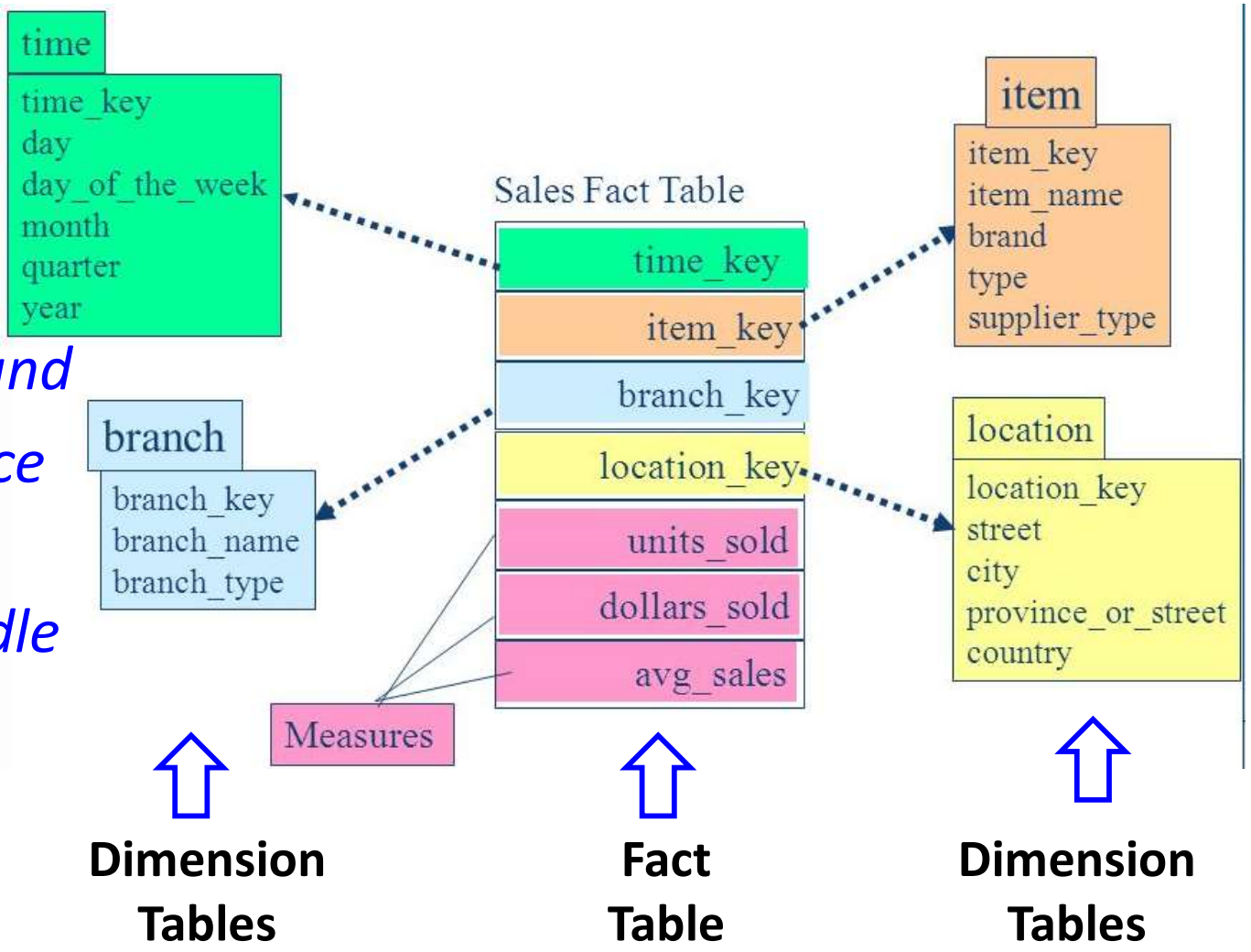
- Result from a business process or business event
  - *Facts are usually numeric and additive*
- Granularity/grain
  - *Identifies the fact level of detail*
  - *One row per sale, one row per service call, one row per claim, …*
  - *Atomic grain is most flexible*



Star Schema

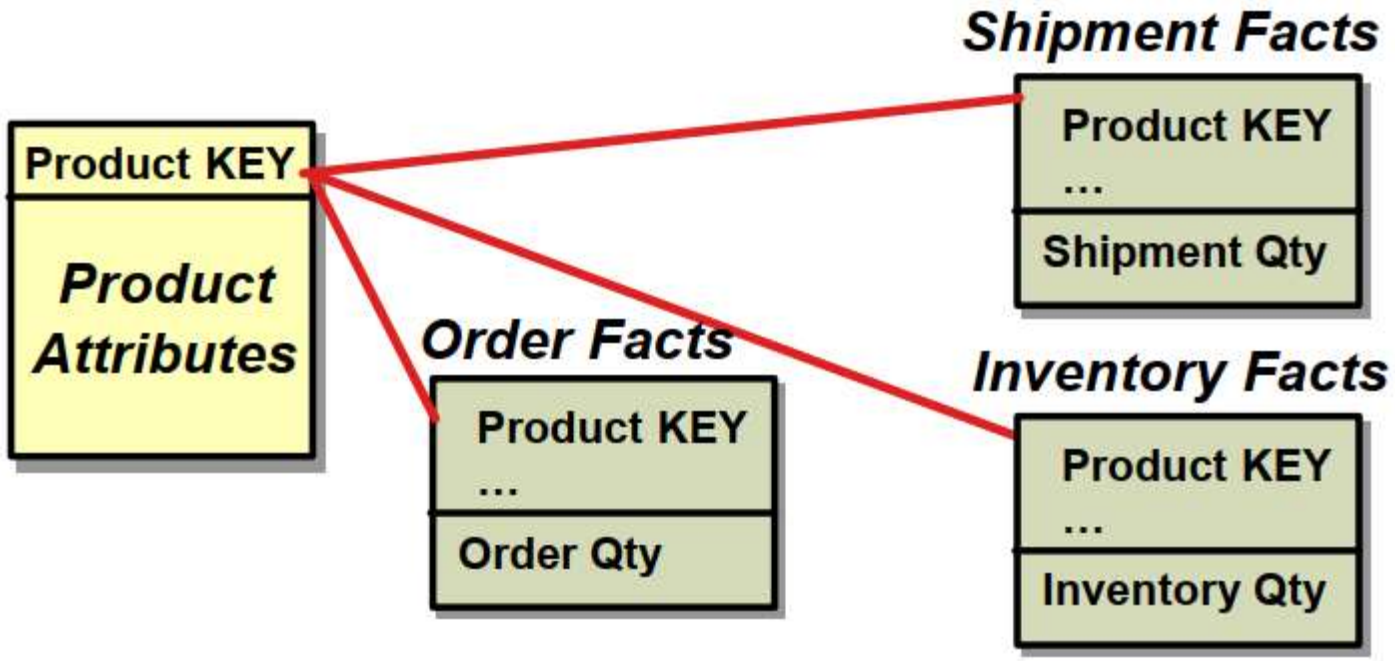
- Fact table per business process/event, plus relevant dimensions

- Benefits
  - *Easier to understand*
  - *Better performance from fewer joins*
  - *Extensible to handle change*

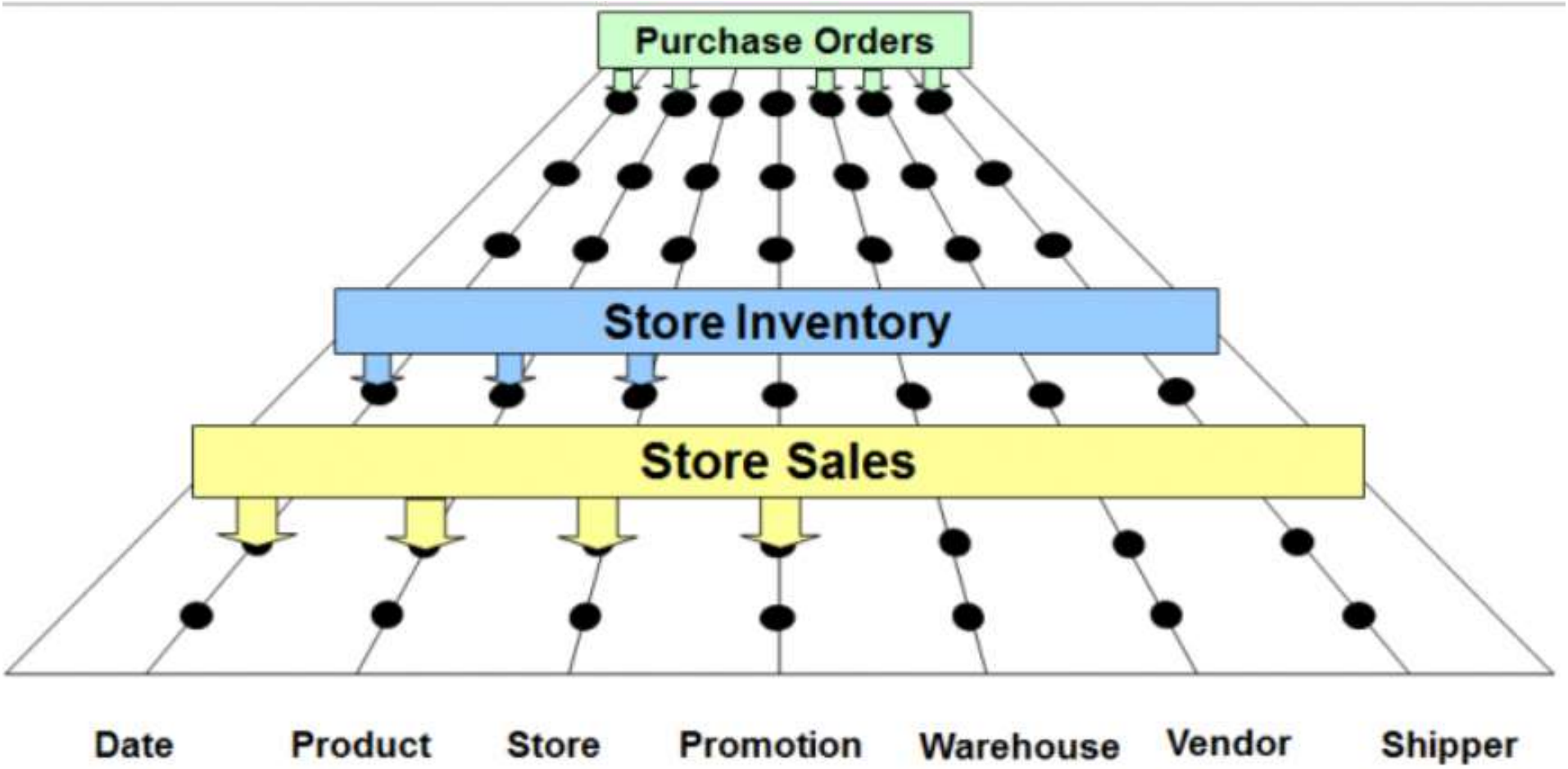


Conformed Dimensions

- Shared across business processes (fact tables) in the DW.
- All fact tables use same standard dimensions
  - *Established via Bus Matrix, enforced in ETL*



Enterprise Data Warehouse Bus Architecture



Data Warehouse Bus Matrix

- Rows = Business processes
- Columns = Conformed dimensions

	COMMON DIMENSIONS						
BUSINESS PROCESSES	Date	Product	Warehouse	Store	Promotion	Customer	Employee
Issue Purchase Orders	X	X	X				
Receive Warehouse Deliveries	X	X	X				X
Warehouse Inventory	X	X	X				
Receive Store Deliveries	X	X	X	X			X
Store Inventory	X	X		X			
Retail Sales	X	X		X	X	X	X
Retail Sales Forecast	X	X		X			
Retail Promotion Tracking	X	X		X	X		
Customer Returns	X	X		X	X	X	X
Returns to Vendor	X	X		X			X
Frequent Shopper Sign-Ups	X			X		X	X

Dimensional Modeling

- A Logical design technique for structuring data with the following objectives:
  1. **Intuitive**: Easy for business users to understand
  2. **Fast**: Excellent query performance

✓ Think of a Dimensional Model as a fact table + the dimensions it requires.

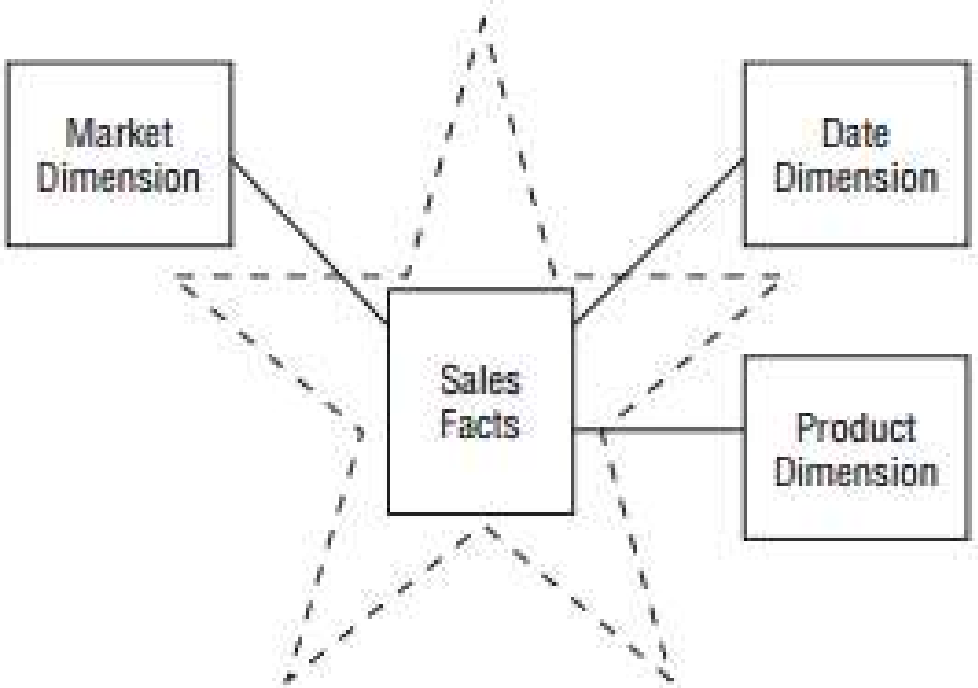
✓ Dimensional Models are implemented in the Relational DBMS as star schemas and in MOLAP databases as cubes.

Components of the Dimensional Model

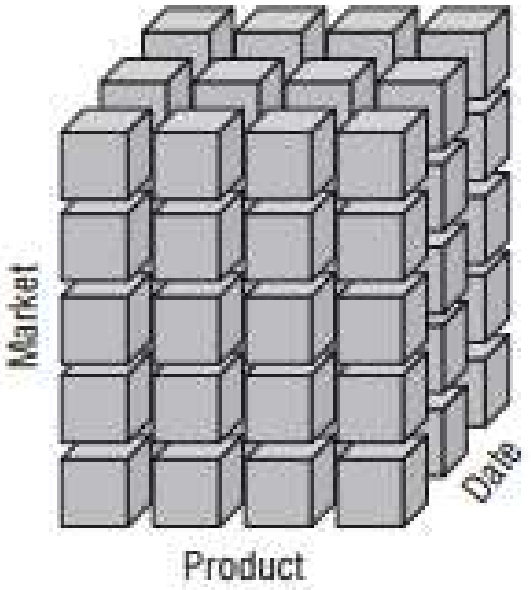
- **Fact Table** – A database table of quantifiable performance measurements (facts).
  - Originate from business processes.
  - Has FK's to each of the dimensions.
  - **Ex.** Sales Amount, Days To Ship, Quantity on Hand.
- **Dimension Table** – A table of contexts for the facts.
  - **Ex.** Date/Time, Location, Customer, Product
- **Attribute** – A characteristic of a dimension.
  - **Ex.** Product: Name, Category, Department
- **Star Schema** – Connections among facts and dimensions which define a business process.
  - **Ex:** Sales, Inventory Management

Multidimensional Data Representations

Star schema



OLAP cube



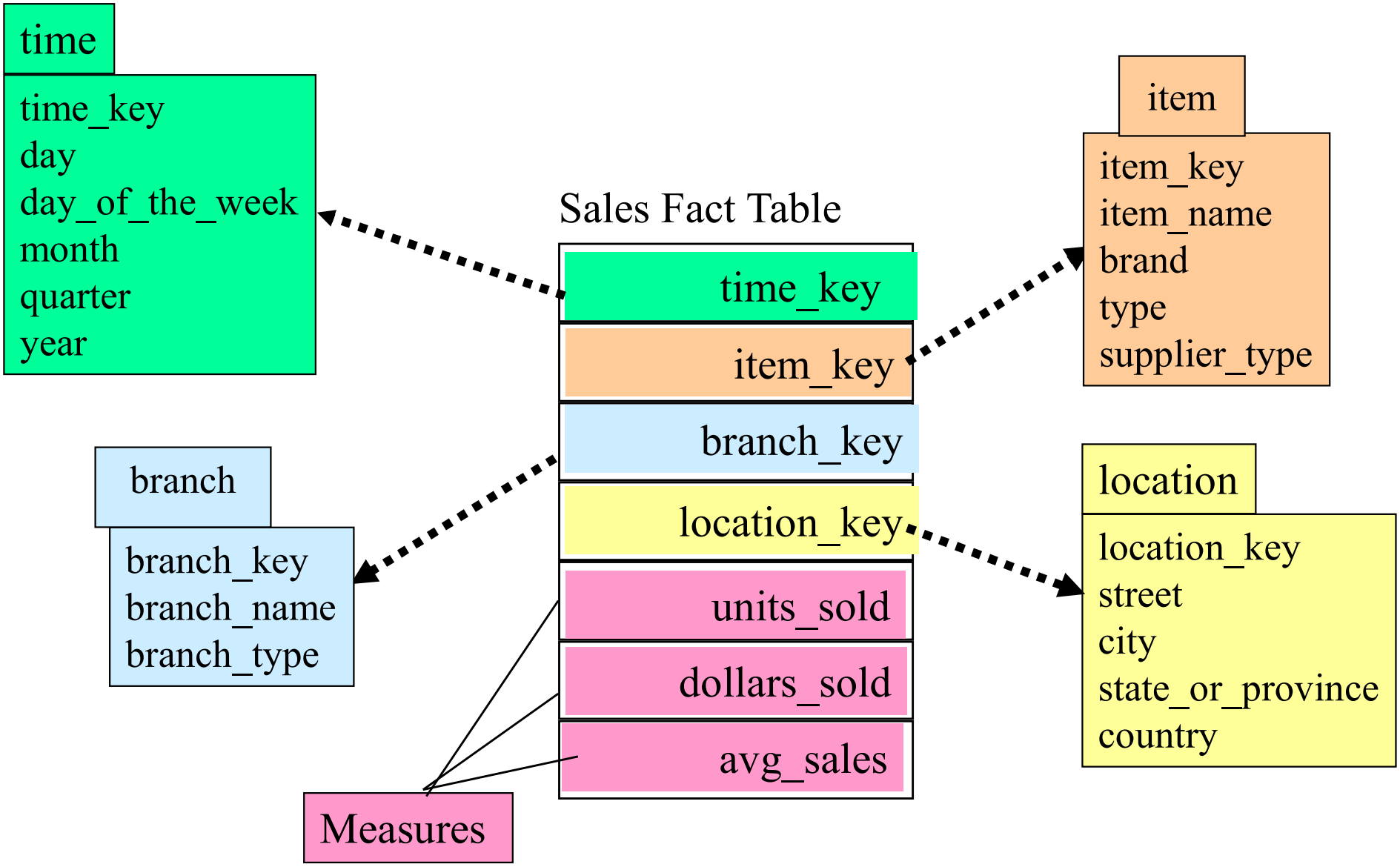
Modeling data warehouses

- A **data warehouse** is based on a multidimensional data model which views data in the form of a data cube
- A **data cube**, such as sales, allows data to be modeled and viewed in multiple dimensions
  - **Dimension tables**, such as item (item_name, brand, type), or time(day, week, month, quarter, year)
  - **Fact table** contains **measures** (such as dollars_sold) and keys to each of the related dimension tables

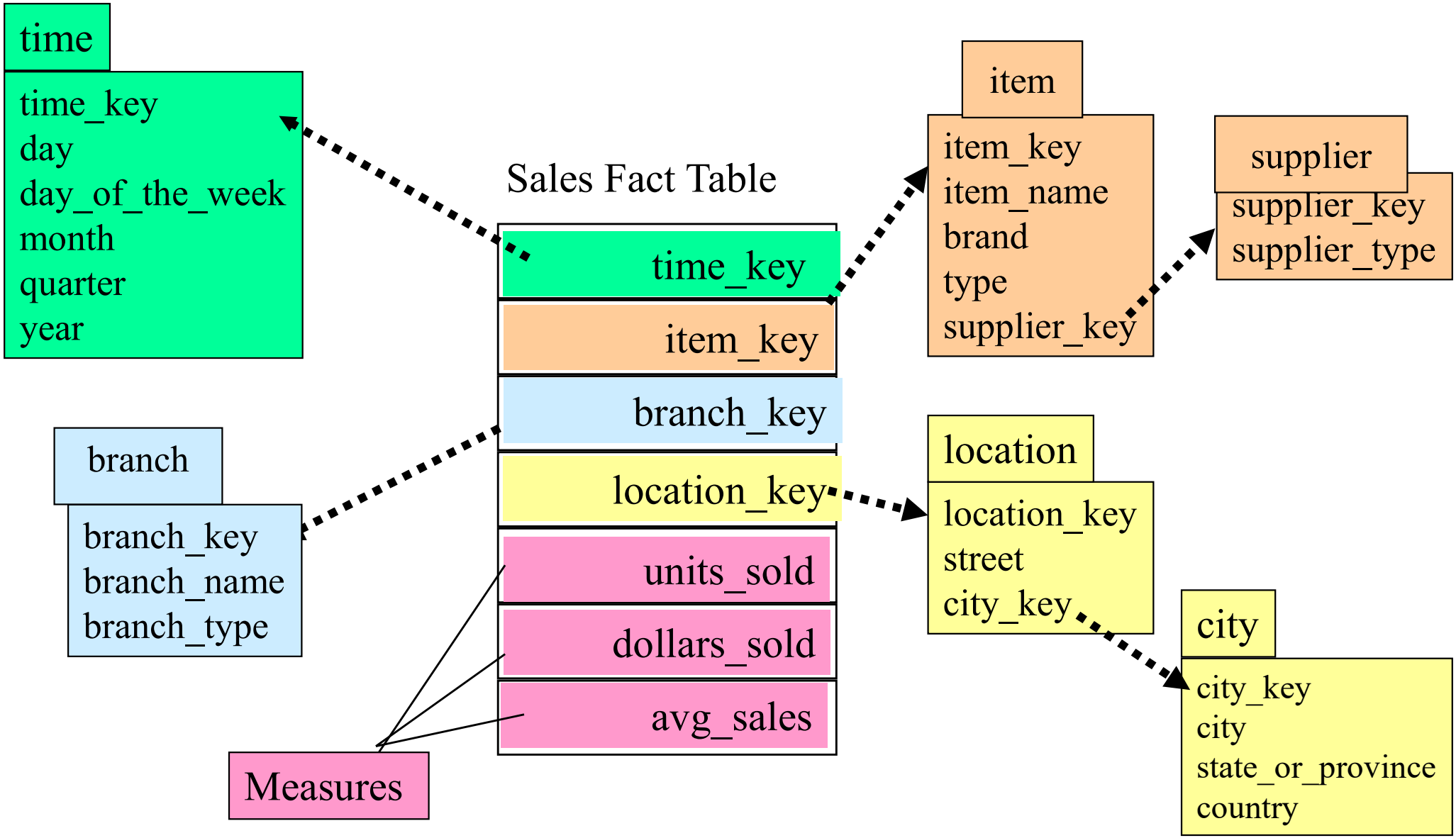
Modeling data warehouses

- Modeling data warehouses: dimensions & measures
  - **Star schema**: A fact table in the middle connected to a set of dimension tables
  - **Snowflake schema**: A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake
  - **Fact constellations**: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation

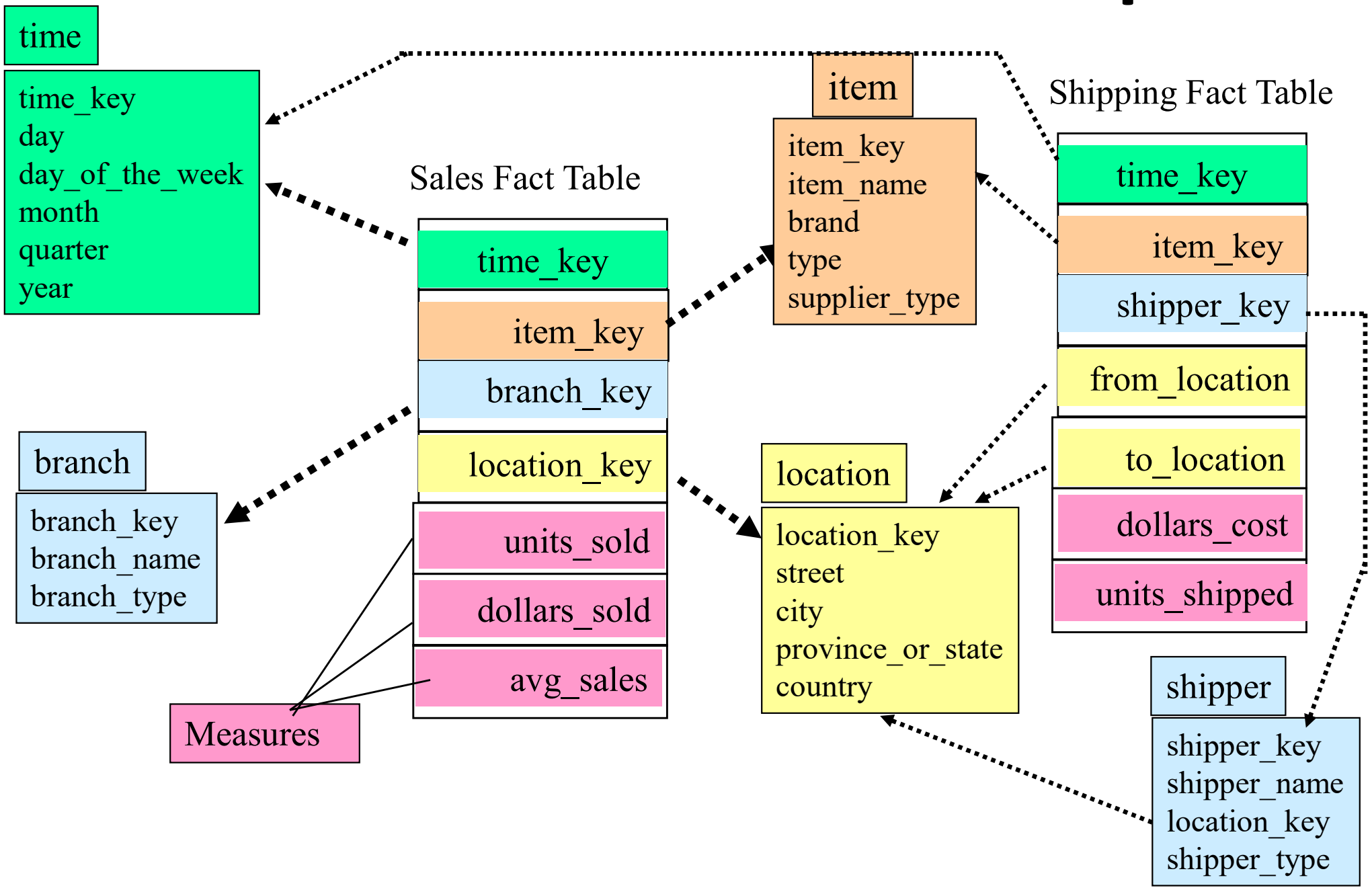
Star Schema Example



Snowflake Schema Example



Constellation Schema Example



Dimensional Modeling Process

- Develop the Data Warehouse Bus matrix
- Follow the 4-step method to define fact and dimension
  - ***Step 1: Identify the business process (matrix row)***
  - ***Step 2: Declare the grain***
  - ***Step 3: Identify the dimensions***
  - ***Step 4: Identify the facts***
- Diagram the dimensional model
- Fill the dimension and fact attributes

#1: Identifying Business Processes

3 type of business processes (fact table)

1. Events or Transactions
2. Workflows a.k.a. Accumulating Snapshots
3. Points in time a.k.a Periodic Snapshots

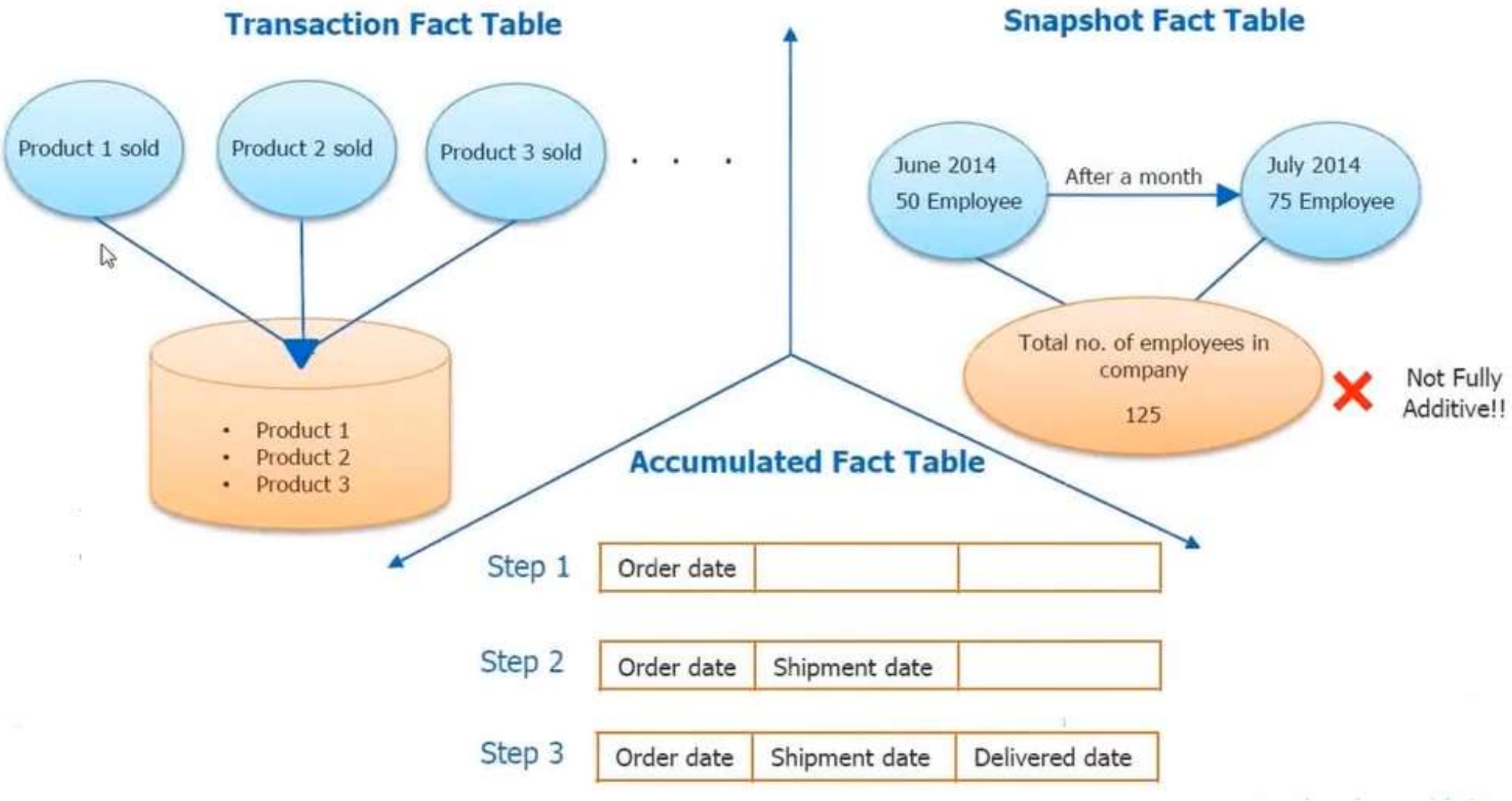
Business processes contain facts which we use end up being the fact tables in our ROLAP star schemas.

Transaction

Accumulating Snapshot

Periodic Snapshot

Types of Fact Table



Transaction Fact

- The most **basic** fact grain
- **One row per line** in a transaction
- Corresponds to a point in space and time
- Once inserted, it is not revisited for update
- Rows inserted into fact table when transaction or event occurs
- **Examples**
  - Sales, Returns, Telemarketing, Registration Events

Accumulating Snapshot Fact

- Less frequently used, application specific.
- Used to capture a **business process workflow**.
- Fact row is initially inserted, then **updated** as milestones occur
- Fact table has **multiple date FK** that **correspond to each milestone**
- **Special facts**: milestone counters and lag facts for length of time between milestones
- **Examples:**
  - Order fulfillment, Job Applicant tracking, Rental Cars

Periodic Snapshot Fact

- At **predetermined intervals snapshots** of the same level of details are taken and stacked consecutively in the fact table
- Snapshots can be taken daily, weekly, monthly, hourly, etc…
- Complements detailed transaction facts but does not replace them
- Share the same conformed dimensions but has less dimensions
- **Examples:**
  - Financial reports, Bank account values, Semester class schedules, Daily classroom Lab Logins, Student GPAs

Which Fact Table?

1. Concert ticket purchases?
2. Voter exit polls in an election?
3. Mortgage loan application and approval?
4. Auditing software use in a computer lab?
5. Daily summaries of visitors to websites?
6. Tracking Law School applications?
7. Attendance at sporting events?
8. Admissions to sporting events at 15 minute intervals?

Transaction

Accumulating Snapshot

Periodic Snapshot

#2: Declare the grain

- **Grain** is the **level of detail** stored in the data warehouse
  - Do we store all products or just product categories
  - Each month, week, day, or hour
- Grain **impact on the size** of the data warehouse
- Typically implement the lowest possible dimension grain
  - We can aggregate in many different ways

#3: Identify the Dimensions

- Dimensions provide **context** for our facts.
- We can easily identify dimensions because of the "***by***" and/or "***for***" words.
  - **Ex.** Total accounts receivables ***for*** the IT Department ***by*** Month.
- Dimensions have **attributes** which describe and categorize their values.
  - **Ex.** Student: Major, Year, Dormitory, Gender.
- The attributes help **constrain** and **summarize facts**.

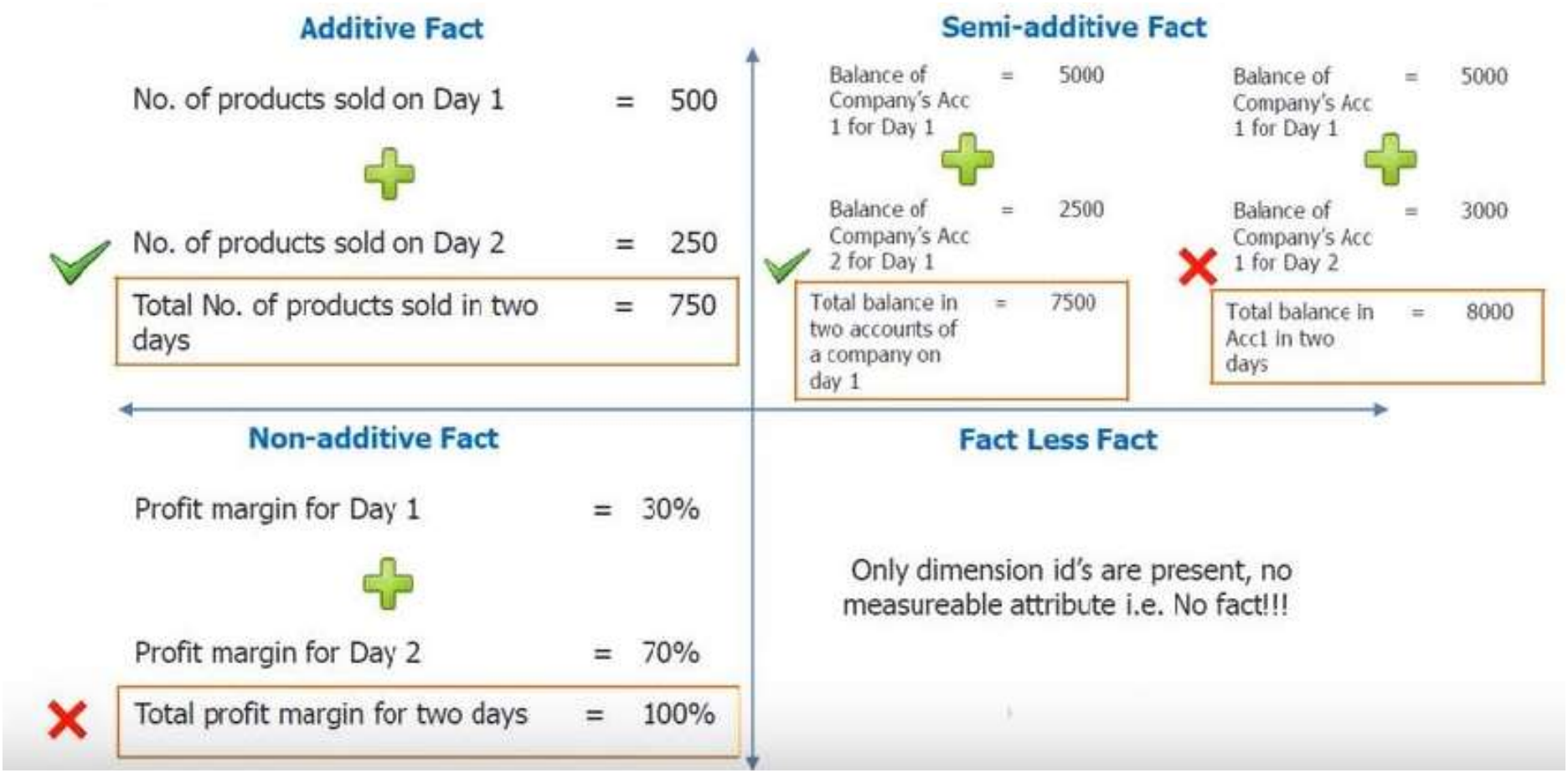
#4 Identify the Facts

- **Facts** are **quantifiable numerical values** associated with the **business process**.
  - How much?
  - How many?
  - How long?
  - How often?
- If its not tied to the business process, its not a fact.
- For example:
  - Points Scored == Fact

Types of Facts

- **Additive -** Fact can be summed across all dimensions
  - The most useful kind of fact.
  - Quantity sold, hours billed.
- **Semi-Additive -** Fact cannot be summed across all dimensions, such as time periods.
  - Sometime these are *averaged* across the time dimension.
  - Quantity on Hand, Time logged on to computer.
- **Non-Additive -** Cannot be summed across any dimension.
  - These do not belong in the fact table, but with the dimension.
  - Basketball player height, Retail Price

Types of Fact



Addictive Fact

- Airline Industry Fact
  - Dimensions: Date and Branch
  - Measure: Number of ticket sold

Airline Industry		
Date	Branch	No. of tickets sold → Measure / Fact
1st Jan 2000	New York	1000
1st Jan 2000	Chicago	1500
2nd Jan 2000	New York	2000
2nd Jan 2000	Chicago	1000

Addictive Fact

Sum across Date Dimension

Airline Industry	
Date	No. of tickets sold
1st Jan 2000	1000
1st Jan 2000	1500
Total No. of tickets sold	2500

Sum across Branch Dimension

Airline Industry	
Branch	No. of tickets sold
New York	1000
New York	2000
Total No. of tickets sold	3000

Semi-addictive Fact

- Bank Industry Fact
  - Dimensions: Date and Account
  - Measure: Current Balance

Bank Industry		
Date	Account	Current Balance → Measure / Fact
1st Jan 2000	1234	1000
1st Jan 2000	4567	1000
2nd Jan 2000	1234	2000
2nd Jan 2000	4567	1000

Semi-addictive Fact

Sum across Date Dimension

Bank Industry		
Date	Account	Current Balance
1st Jan 2000	1234	1000
1st Jan 2000	4567	1000
Total current balance across all accounts on 1st Jan 2000		2000 ✓

Sum across Account Dimension

Bank Industry		
Date	Account	Current Balance
1st Jan 2000	1234	1000
2nd Jan 2000	1234	2000
Total current balance for acc no 1234		3000 ✗

Non-addictive Fact

- Supper Market Chain Fact
  - Dimensions: Date and Store
  - Measure: Profit Margin

Super Market Chain		
Date	Store Name	Profit Margin %
1st Jan 2000	ABC	30
1st Jan 2000	XYZ	40
2nd Jan 2000	ABC	50
2nd Jan 2000	XYZ	60

Measure / Fact

Non-addictive Fact

Sum across Date Dimension

Super Market Chain		
Date	Store	Profit Margin %
1st Jan 2000	XYZ	40
2nd Jan 2000	XYZ	60
Total profit margin		100 ✗

Sum across Store Dimension

Super Market Chain		
Date	Store	Profit Margin %
1st Jan 2000	XYZ	40
1st Jan 2000	ABC	30
Total profit margin		70 ✗

Which Fact? Additive? Semi? Non?

1. Number of page views on a website?
2. The amount of taxes withheld on an employee's monthly paycheck?
3. Credit card balance.
4. Pants waist size? 32, 34, etc…
5. Tracking when a student attends class?
6. Product Retail Price?
7. Vehicle's MPG rating?
8. The number of minutes late employees arrive to work each day.

Enterprise Bus Matrix

- A key deliverable from requirements gathering, the **bus matrix** documents your **business processes**, **grain**, **dimensions**, and **facts** across all projects in your program.

Business Process Name	Fact Table	Fact Grain Type	Granulairty	Facts	Product	Customer	Employee	Order Date	Shipped Date
Sales Reporting	FactSales	Transaction	one row per order detail / line item.	Quantity, Unit Price , Discount Amount, Sold Amount, Freight Amount	x	x	x	x	x

Build A Bus Matrix

- Identify business processes
  - Transaction, Periodic Snapshot or Accumulating Snapshot
- Declare the grain
  - The level of detail stored in the data warehouse
- Identify the dimensions
  - The context for our facts
- Identify facts
  - Should be Additive, or at least Semi-Additive

The Dimension Table Key

- **Surrogate keys** (identities, sequences e.g. 1,2,3,…) are used for the **primary key constraint**.
- They yield best performance for the Star Schema
  - most efficient joins,
  - smaller indexes in fact table,
  - more rows per block in the fact table
- They have no dependency on primary key in operational source data
  - Makes it easier to deal with changes to the source data
- Dimension table requires a **natural key** or **business key** to identify a unique row
  - Ex: Customer's email address, Employee's ID number.

Date and Time Dimensions

- Just about every fact table as a date and/or time dimension.
- This is the most common of **conformed dimensions**
- Usually **generated programmatically** during the ETL process or **imported** from a spreadsheet.
- Acceptable to use **PK** in the form **YYYYMMDD (int)**
- In you need time of day, use a separate dimension.
- Time of day should only be used if there are **meaningful textual descriptions** of time
  - **Ex.** Lunch, Dinner, 1st shift, 2nd Shift, Etc…
- Elapsed times intervals are **facts**, not attributes.
  - **Ex**. Minutes between when order was received and shipped

Date Dimension Example

DateKey	FullDate	DateName	DayOfWeek	DayNameOfWeek	DayOfMonth	DayOfYear	WeekdayWeekend	WeekOfYear	MonthName	MonthOfYear	IsLastDayOfMonth	Calen
20130720	7/20/2013	2013/07/20	7	Saturday	20	201	Weekend	29	July	7	N	
20130721	7/21/2013	2013/07/21	1	Sunday	21	202	Weekday	30	July	7	N	
20130722	7/22/2013	2013/07/22	2	Monday	22	203	Weekday	30	July	7	N	
20130723	7/23/2013	2013/07/23	3	Tuesday	23	204	Weekday	30	July	7	N	
20130724	7/24/2013	2013/07/24	4	Wednesday	24	205	Weekday	30	July	7	N	
20130725	7/25/2013	2013/07/25	5	Thursday	25	206	Weekday	30	July	7	N	
20130726	7/26/2013	2013/07/26	6	Friday	26	207	Weekend	30	July	7	N	
20130727	7/27/2013	2013/07/27	7	Saturday	27	208	Weekend	30	July	7	N	
20130728	7/28/2013	2013/07/28	1	Sunday	28	209	Weekday	31	July	7	N	
20130729	7/29/2013	2013/07/29	2	Monday	29	210	Weekday	31	July	7	N	
20130730	7/30/2013	2013/07/30	3	Tuesday	30	211	Weekday	31	July	7	N	
20130731	7/31/2013	2013/07/31	4	Wednesday	31	212	Weekday	31	July	7	Y	
20130801	8/1/2013	2013/08/01	5	Thursday	1	213	Weekday	31	August	8	N	
20130802	8/2/2013	2013/08/02	6	Friday	2	214	Weekend	31	August	8	N	
20130803	8/3/2013	2013/08/03	7	Saturday	3	215	Weekend	31	August	8	N	
20130804	8/4/2013	2013/08/04	1	Sunday	4	216	Weekday	32	August	8	N	
20130805	8/5/2013	2013/08/05	2	Monday	5	217	Weekday	32	August	8	N	
20130806	8/6/2013	2013/08/06	3	Tuesday	6	218	Weekday	32	August	8	N	
20130807	8/7/2013	2013/08/07	4	Wednesday	7	219	Weekday	32	August	8	N	
20130808	8/8/2013	2013/08/08	5	Thursday	8	220	Weekday	32	August	8	N	
20130809	8/9/2013	2013/08/09	6	Friday	9	221	Weekend	32	August	8	N	
20130810	8/10/2013	2013/08/10	7	Saturday	10	222	Weekend	32	August	8	N	
20130811	8/11/2013	2013/08/11	1	Sunday	11	223	Weekday	33	August	8	N	

Time Dimension Example

Time_SK	12_hr	24_hr	am_pm	Minute_in_Hour	Time_Value
0	12	0	am	0	12:00 AM
1	12	0	am	1	12:01 AM
2	12	0	am	2	12:02 AM
3	12	0	am	3	12:03 AM
4	12	0	am	4	12:04 AM
5	12	0	am	5	12:05 AM
59	12	0	am	59	12:59 AM
60	1	1	am	1	1:00 AM
719	1	12	am	59	11:59 AM
720	2	13	pm	0	12:00 PM
1439	11	23	pm	59	11:59 PM

Degenerate Dimensions

- **Dimensions we store in the fact table**, because there's too many of them for their own a dimension
  - For example a 1-1 relationship from fact to dimension
- These occur in transaction fact tables that have a parent child (One to Many) structure
  - **Ex.** Order → Order Detail,
  - Airline Ticket → Flights
- Allow us to **drill-through** to operational data, in the ODS.
- Usually ends up as part of the **primary key** of the fact table.

Slowly Changing Dimensions

- Dimensional data changes infrequently but when it does you need a strategy for addressing the change.
  - **Ex:** What happens when a customer has a new address, or an Employee has a name change?
- **4 Popular strategies**
  - **Type 1**: Overwrite the existing attribute
  - **Type 2**: Add a new Dimension row
  - **Type 3**: Add a new Dimension attribute -
  - **Mini-Dimension**: Add a new Dimension
- These strategies are not mutually exclusive, and can be combined.

Type 1: Overwrite

- Appropriate for
  - correcting mistakes or errors in data
  - changes where historical associations do not matter
  - the old value has no significance
- If the **previous value matters**, **don't use this strategy**.
  - You are rewriting history
- Problems will occur with data aggregated on old values.
- **Ex.** Employee Name Changes, Corrections, Natural Key Edits.

Type 2: Add New Dimension Row

- Most popular strategy, as it preserves history
- Natural key is repeated
- Old and new values are stored along with effective dates and indicator of which row is "**current**"

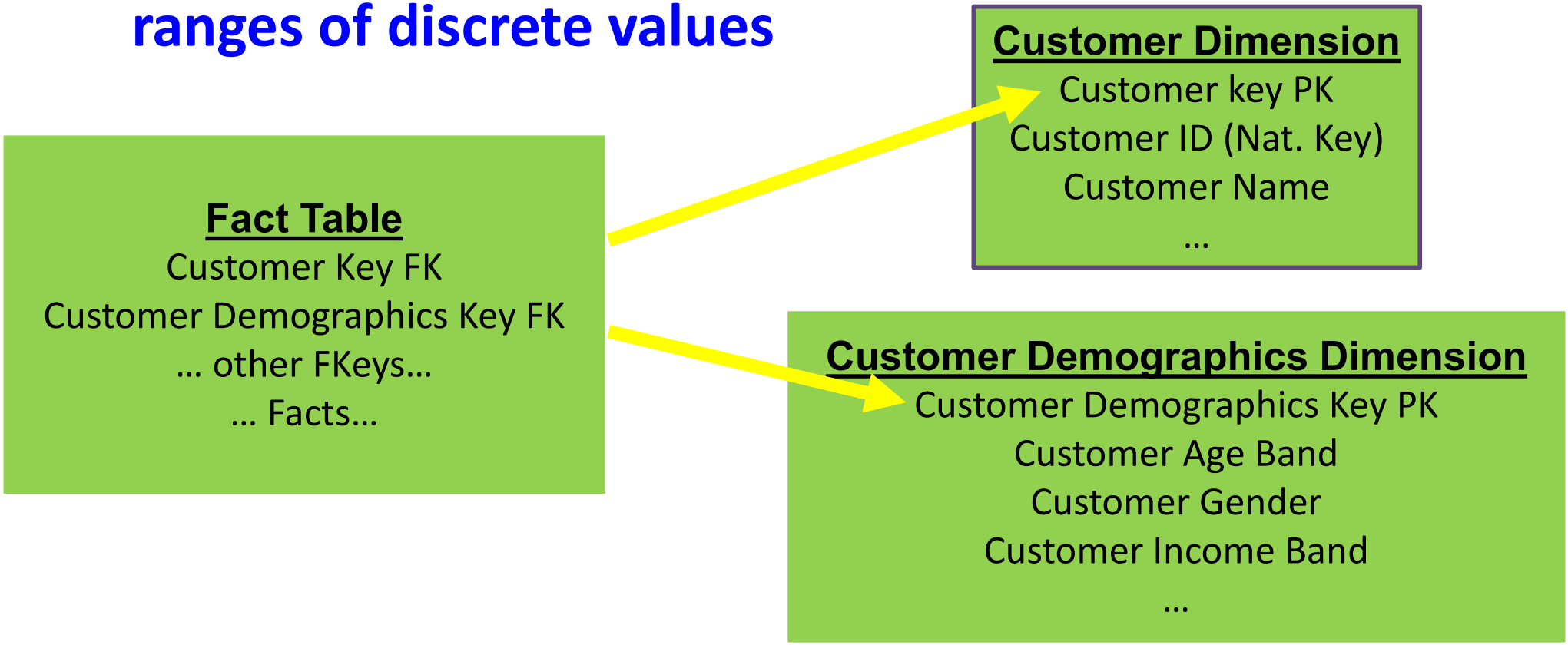
Product Key	Product Descr.	Product Code	Department	Effective Date	Expiration Date	Current Row
11981	Stapler, Red	ST901	Accessories	4/7/2010	9/1/2011	N
20344	Stapler, Red	ST901	Supplies	9/2/2011	3/31/2013	N
45393	Stapler, Red	ST901	Office Supplies	4/1/2013	12/31/9999	Y

Type 3: Add A New Dimension Attribute

- Infrequently used, preserves history
- Useful for "**soft**" changes where users might want to choose between the old and new attribute, or need to access both values for a time
- The new value is written to the existing column, the old value is stored in a new column
- This way queries do not have to be re-written to access the new attribute
- **Ex.** Redistricting sales territories. Re-charting accounting codes

Mini-Dimensions: Add a new Dimension

- If **attributes change frequently** consider placing them in their own "**mini-dimensions**"
- Most effective when you have **banded values**, or **ranges of discrete values**



Summary

- Dimensional Modeling Concepts
- Dimensional Modeling Process
- Types of Fact Tables
- Types of Facts
- Slowly Changing Dimensions

Dimensional Modeling Practice

- Identify: Business Process, Fact and Dimensions

1. What's the total amount of product shipped by sales region for 2010-2014?
2. What's the average time in days for a student's application to be processed?
3. How many employees wait more than 15 minutes for a bus to the Manley parking lot?