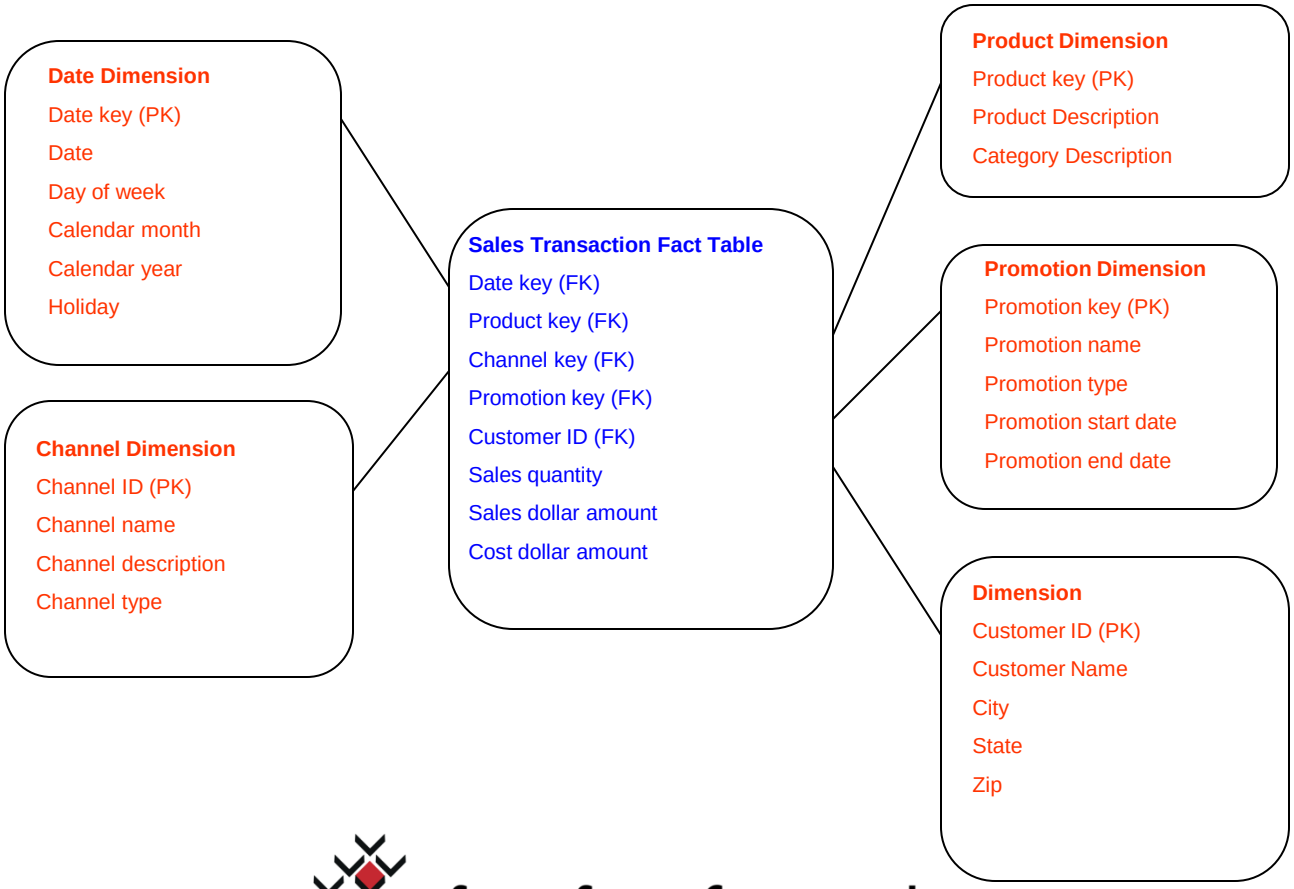
Data Strategies for Efficiency and Growth



SYSTEMS SEMINAR CONSULTANTS, INC.

2997 Yarmouth Greenway Drive Madison, WI 53711

(608) 278-9964 www.sys-seminar.com

Efficiency and Growth are Two sides of the Same Coin

- Scalability

  What happens when data volume increase 10x?

- Consistency

  Does each new system require redesign and relearning for users?

- Flexibility

  Can new data items be integrated?

Issues

- Lack of consistency among systems
- Change is painful
- Backup and rollback (or not)
- Audit and compliance (or not)
- Disaster recovery (or not)
- Steep learning curves for each new system
- The database system is Excel!

What is data modeling?

- The generalized logical relationship among tables
- Usually reflected in the physical structure of the tables
- Not tied to any particular product or DBMS
- A critical design consideration

Why is data modeling important?

- Allows you to optimize performance
- Allows you to minimize costs
- Facilitates system documentation and maintenance

- *The dimensional data model should be the basis for the query side of your business processes.*
- The dimensional data model is the foundation of a well designed data mart or data warehouse

Common data models

- Denormalized or non-normalized
  - Miniminze multiple instances of data…not!
  - Often evolve from informal systems, e.g., spreadsheets
- Normalized
  - Usually the result of a formal data design process
  - Often implement on a RDBS supporting constraints
  - Goal: Transaction efficiency
- Dimensional
  - “Semi normalized”
  - Goal: Query efficiency

De-normalized Data

Sales Transaction Table

Each row represents a sale transaction line.

Transaction line table
Transaction number
Customer Name
Customer Street Address
Customer City
Customer State
Customer Zip
Multi-state region
Product Category
Product Number
Product Name
Calendar day
Day of week
Month
Year
Season
Annual product cycle number
Sale quantity
Sale dollar amount

Attributes of the sale: customer and product info, date, etc.

All attributes of the sale are included with each transaction line row.

Sale facts: number of items and dollars

De-Normalized Data

A single row contains:

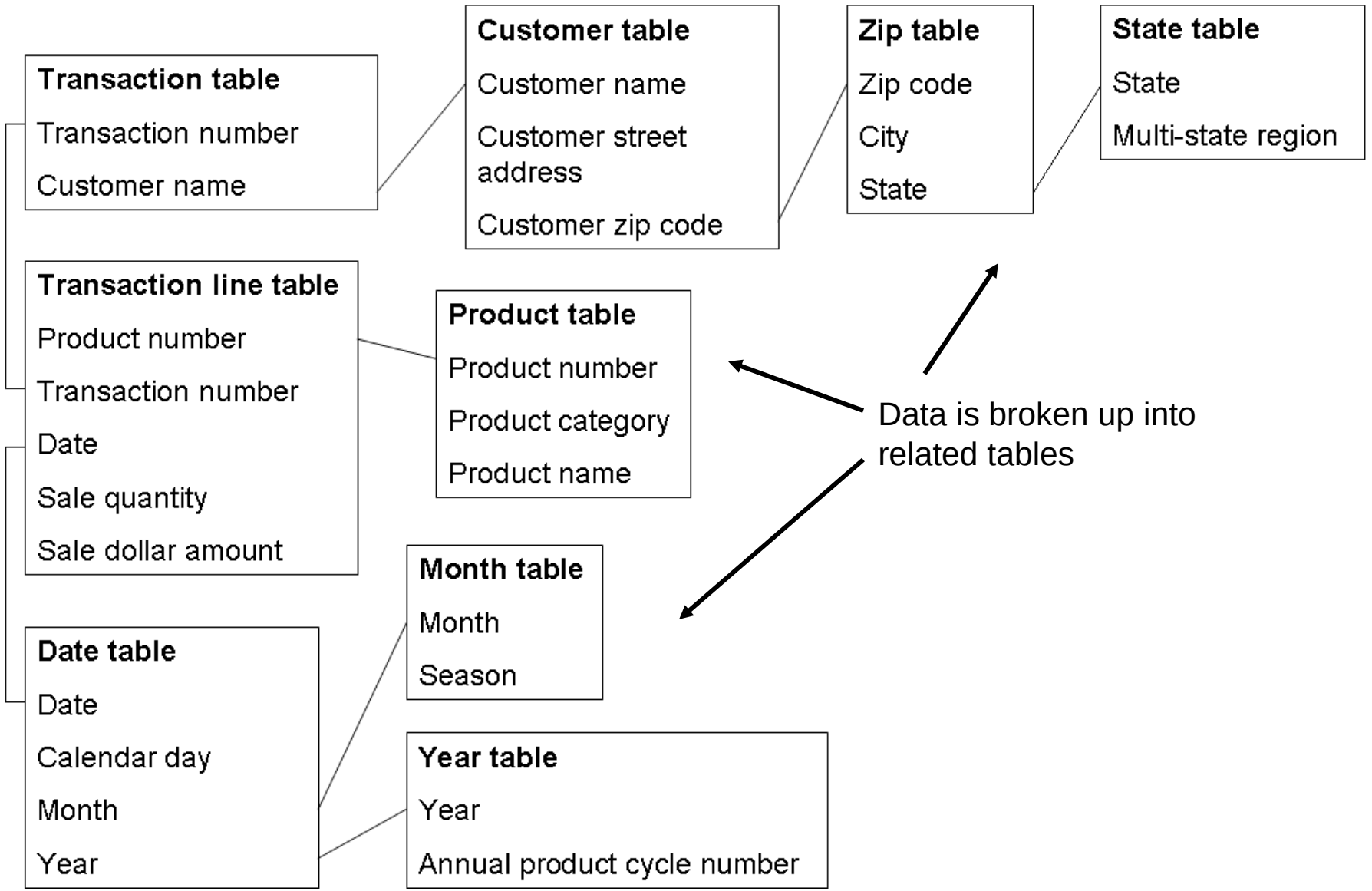
- Numeric measurements and…
- All attributes related to that measurement
- All data is in a single table.

Data redundancy:

Directly correlated attributes, such as product number and product category, are repeated in each row

Sale Number	Product Number	Product Category
1	S3200	Software
2	S3223	Software
3	H7005	Hardware

Normalized Data



Normalized Data

- ***Insert Optimized***

A new transaction line involves gathering only the five data items in the Transaction Line table. No other attribute look up is required.

- ***Redundancy is reduced:***

For example, Product Category is not repeated for each transaction

- ***Changes have less impact on the database:***

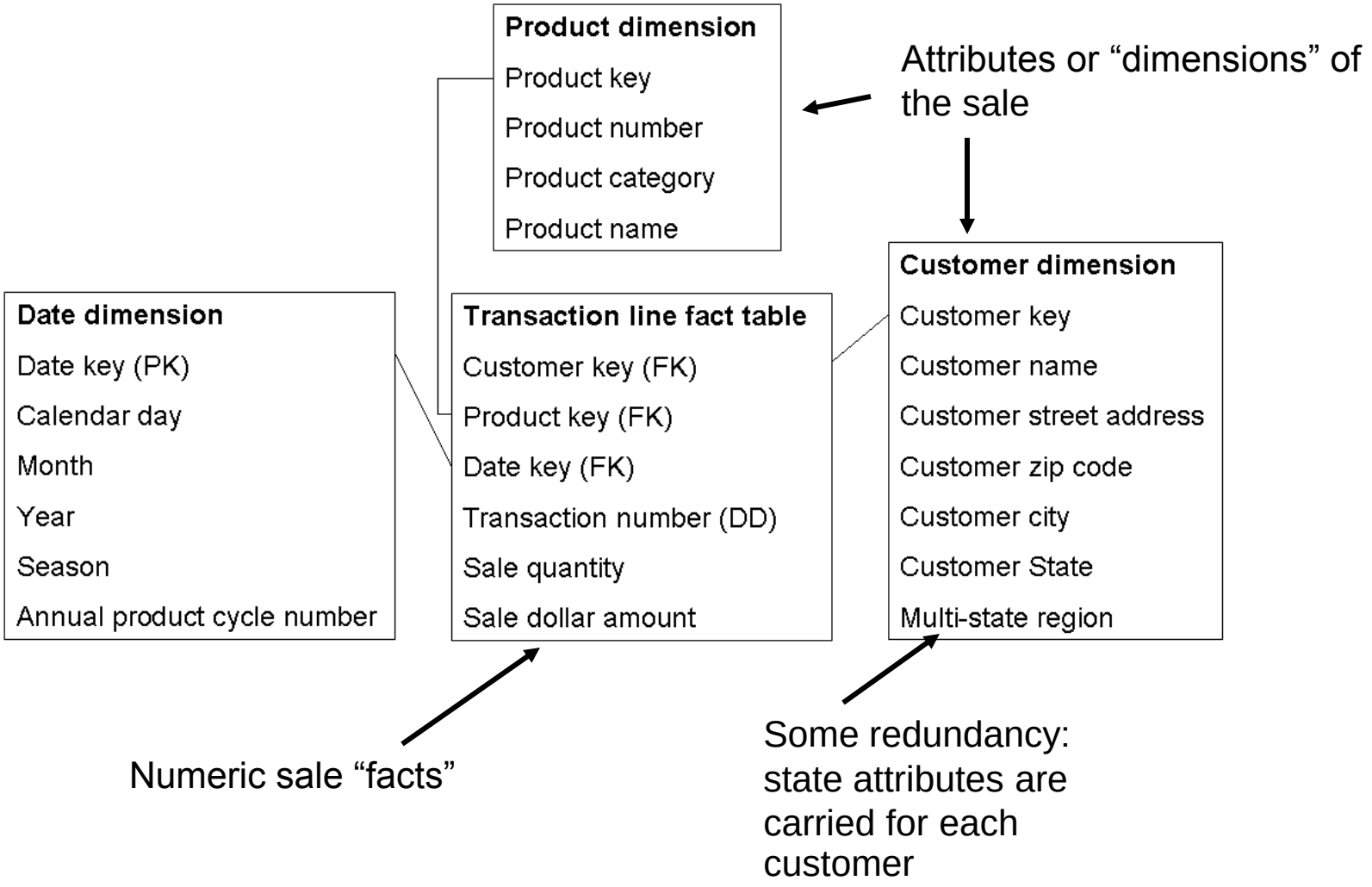
If a product category changes, only the Product table needs to be changed

- ***Query complexity is increased:***

Several tables must be related to each other in order to answer simple questions.

What is the sum of sale amount for each state?

Dimensional Data



Dimensional Data

- *The central fact table is surrounded by dimension tables*

"Star schema"

- *Table relationships are only one level deep*
- No more than two tables need to be joined together for common business questions and aggregations

What is the sum of sale amount for each state?

Facts and Dimensions

•Key terms: Fact and Dimension

•*Fact:*

High cardinality, numeric measure of some event such dollars for a sale.

Typically many rows, one per business event.

•*Dimension:*

Low cardinality, typically non-numeric, attribute of a fact.

Typically many columns, one per attribute of interest.

The dimensional model is made up of facts and dimensions

Why dimensional modeling?

- Bring together data from many different sources and create a single, consistent user view.
- Support the ad hoc queries that arise from novel business questions and situations.
- Maximize flexibility and scalability.
- Improve the end-user experience because...
  - Query performance is optimized
  - Learning curve is flattened

Fact and Dimensions

Dimensional modeling implies two distinct types of data:

1. Facts
2. Dimensions

These data are stored two types of tables:

1. Fact tables
2. Dimension tables

Facts and Dimensions

A fact is…

- A business measurement, amount, or event
- Typically numeric, continuously valued, and additive
- Something we analyze: “What were total sales by state?”

Some facts:

a sale dollar amount, an order quantity, banner ad click

A dimension is…

- Context surrounding a fact: who the fact applies to; when, where, and under what conditions the fact was measured
- Usually a discrete character or numeric value
- Static or slowly changing
- Something we use to identify or group data:
  “What were total sales by state?”

Some dimensions:

customer, date, time, location

Fact Table

Elements of a fact table:

- Fact: **the measure(s) of interest**
- Dimension foreign key: **Key to a row in a dimension table**

Sales Transaction Fact Table

Date key (FK)

Product key (FK)

Channel key (FK)

Promotion key (FK)

Customer ID (FK)

Sales quantity

Sales dollar amount

Cost dollar amount

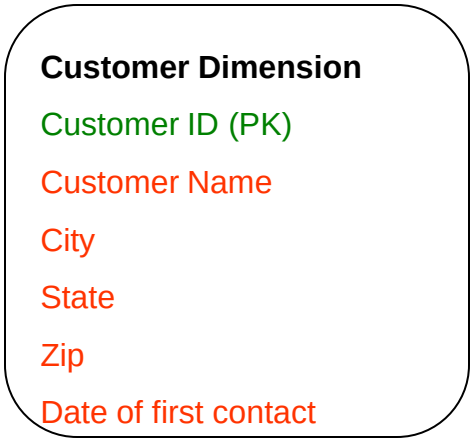
Dimension Table

The dimension table represents an entity of interest to the business:

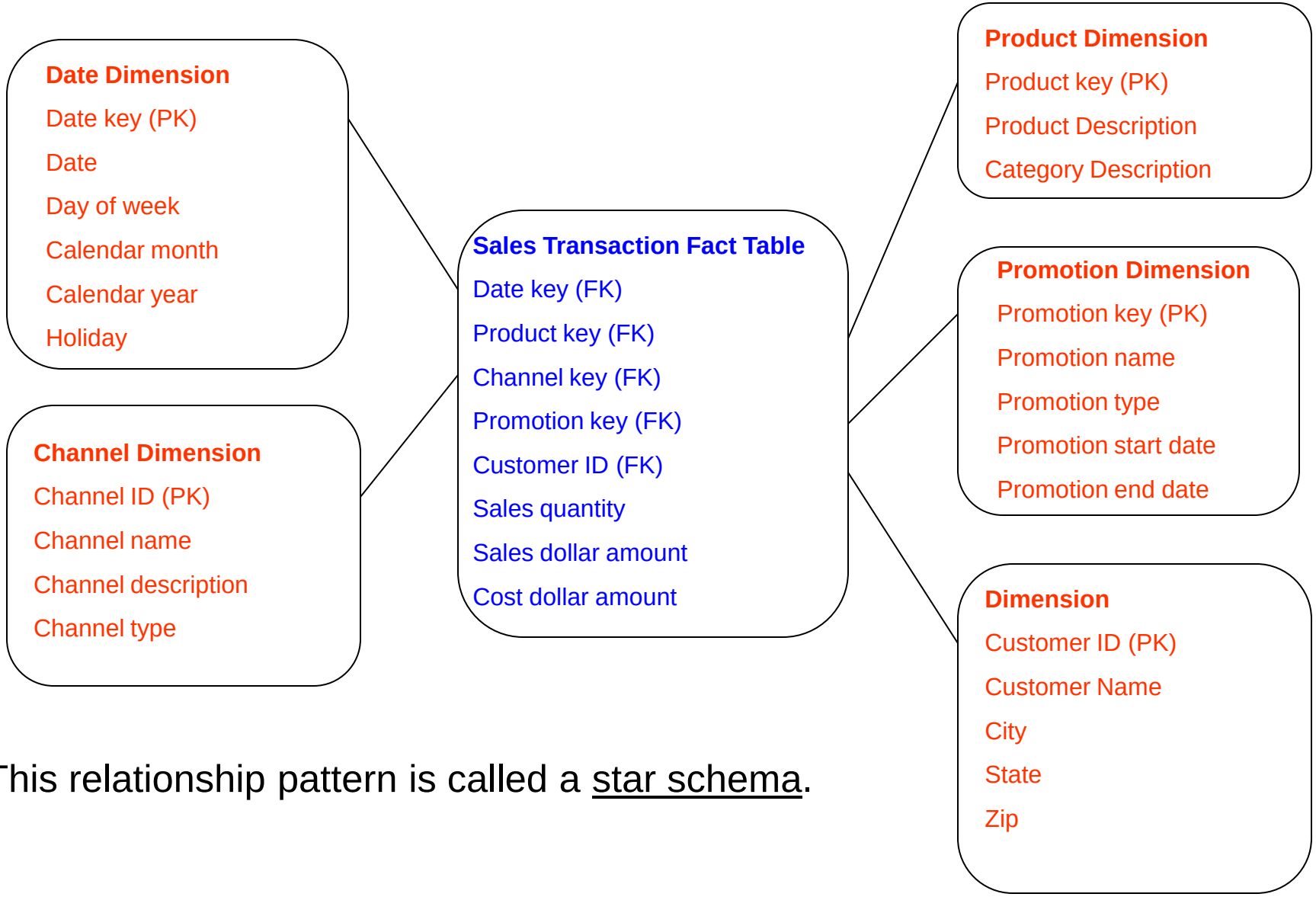
Customer, product, vendor, promotion, location, etc.

Elements:

- Primary key (PK): Unique for each row in the table. It should be a surrogate key, i.e., have no inherent meaning. The value of the dimension key is what's stored in the fact table.
- Dimension attributes: A set of variables that encompass what is known about the business entity.



Fact-Dimension Data Model



This relationship pattern is called a star schema.

Granularity

- Granularity is the level of detail in a fact table
- Granularity is the combination of all dimensions. The more dimensions the finer the grain.

The grain of the previous table is:

Date

Product

Channel

Promotion

Customer

- Only facts with the same grain (i.e. described by the same dimensions) can coexist in a fact table.

- Granularity can always be reduced through aggregation, but can never be increased after the table is built. So: *This is a critical design factor!*

Granularity

- The fact tables represent two different business processes.
- The fact tables each have a unique set of foreign keys, though some foreign keys match (red).

Sales Transaction Fact Table

Date key (FK)

Product key (FK)

Channel key (FK)

Promotion key (FK)

Customer ID (FK)

Sales quantity

Sales dollar amount

Cost dollar amount

Each line is a sales transaction– one customer buying some quantity of one product.

Promotion Event Fact Table

Date key (FK)

Promotion key (FK)

Medium key (FK)

Customer ID (FK)

Count variable

Each line is a promotion event– one customer being offered one promotion.

Surrogate Key

Each row in a dimension table should be identified by a surrogate primary key. A surrogate key has no inherent meaning.

Surrogate key

Natural key

Channel ID (PK)	Channel Name	Channel Description	Channel Type
1042	**Store #0720**	**St. Louis Retail Store**	**Retail Store**
1043	**Store #0721**	**Albuquerque Street Kiosk**	**Kiosk**
1044	**Store #0722**	**Scranton Retail Store**	**Retail Store**
1045	**Store #0720**	**St. Louis Outlet Store**	**Outlet Store**

- The two records for Store #0720 (natural key) can coexist without conflict because each has a unique surrogate key.

Surrogate Key

Benefits of using a surrogate key:

- Surrogate keys make it possible to integrate data from sources that use different forms of a natural key.

- Allow the use of legitimate unknown and null natural keys, or natural keys with special meanings.

- Natural keys may be reused. For example, transaction numbers may be recycled six months after the transaction. A unique surrogate key value distinguishes between two like-numbered transactions.

Drill Down and Up

Drill down means displaying facts at a finer level of granularity.

When you drill down you add dimensional attributes.

Example:

I am viewing sales by state and I want to drill down to the zip code level within state.

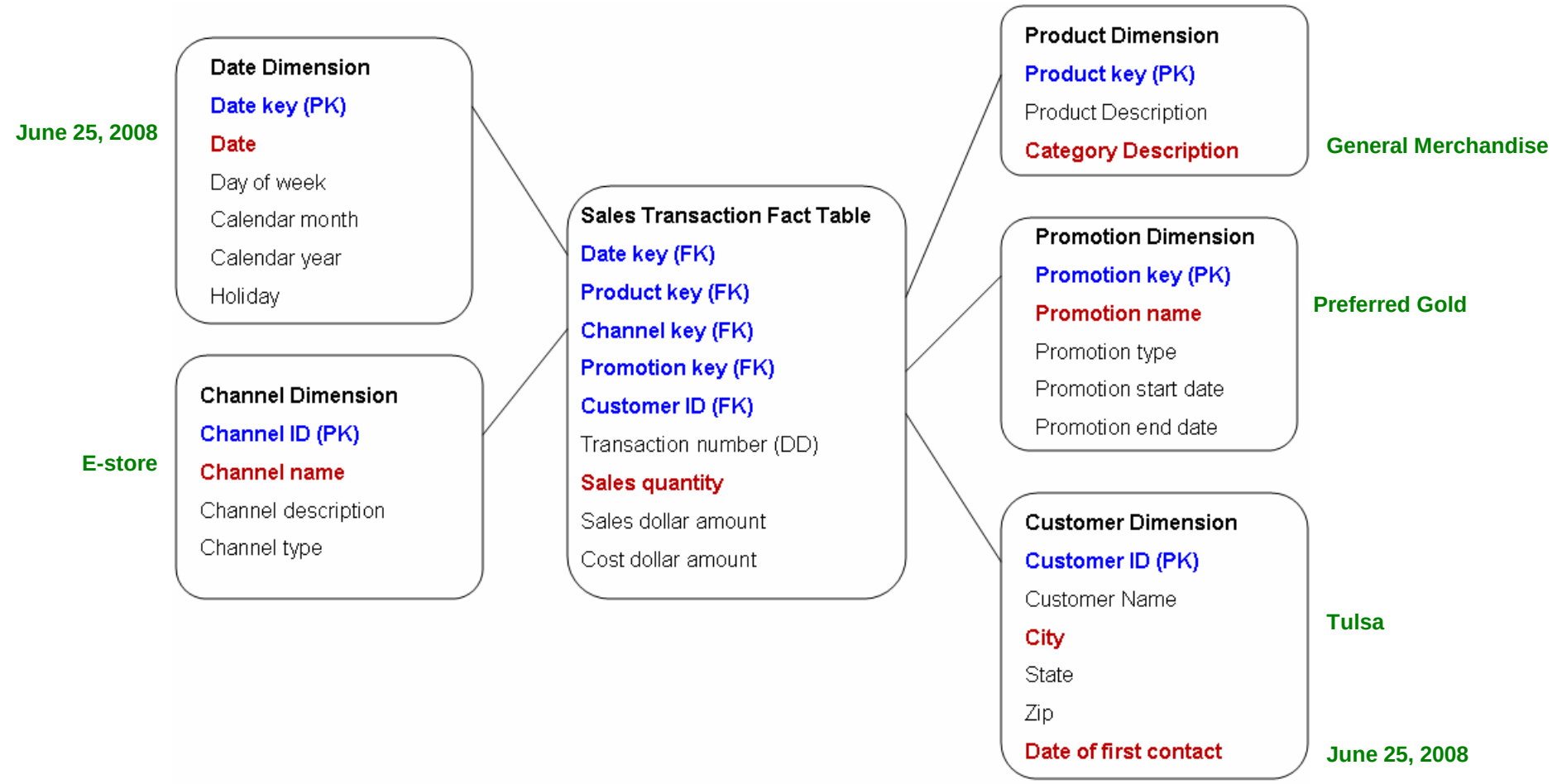
Drill up is the reverse. Drill up reduces the number of dimensional attributes.

Drill up is aggregation.

Example:

I am viewing sales by state but want sales aggregated by multi-state region.

Drill Up and Down



Drill Across

Drill Across means:

Join two or more facts that share the same dimensions.

Consider the question…

"How many customers who purchased products this December were notified of the Year End Clearance promotion by e-mail?"

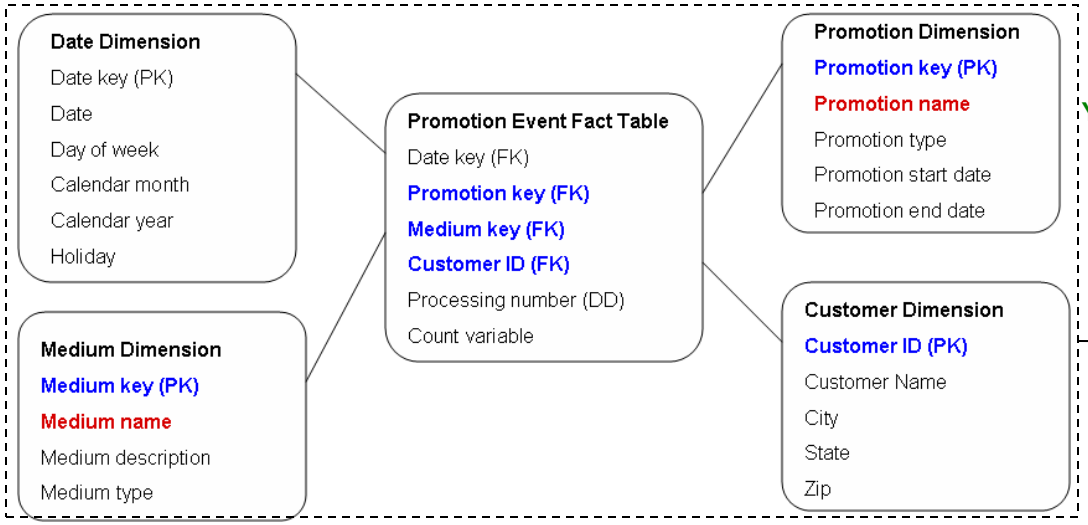
The answer involves two different facts:

1. Sale events
2. Promotion events

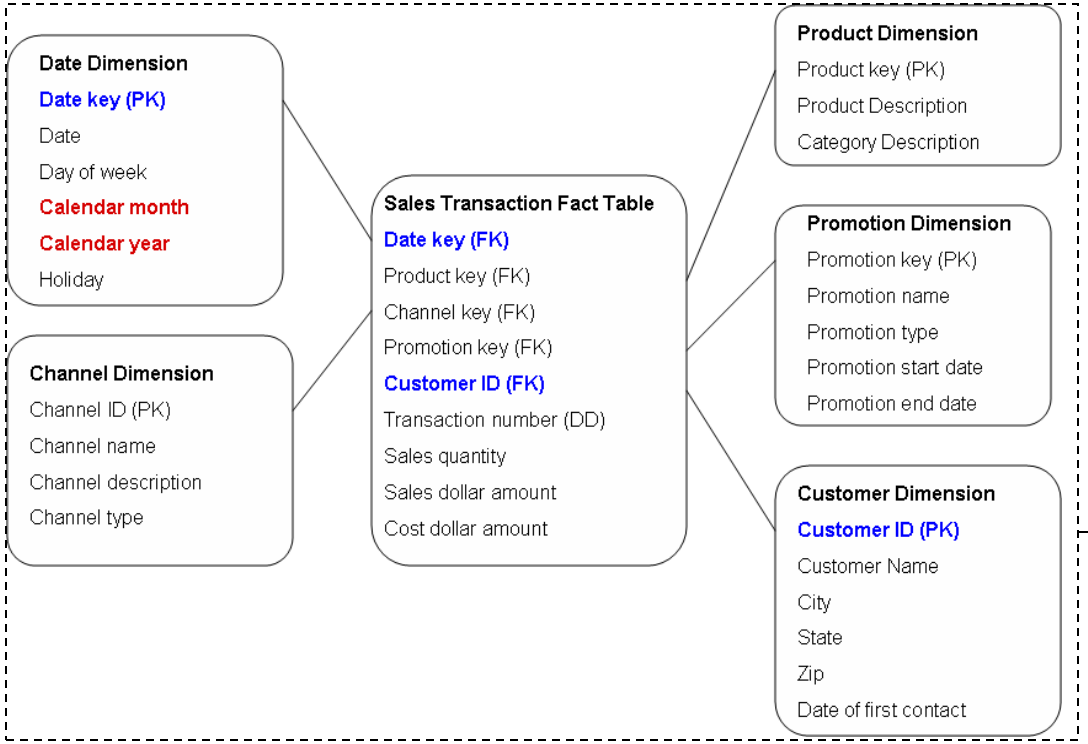
Sales facts and promotion facts can be joined on their common dimension: customer

Drill Across Query

Promotion Event Schema



Sales Transaction Schema



The shared customer dimension allows for a join on Customer ID

Conformed Dimensions

Criteria for conformed dimensions:

- Like-entities represented in different tables have the same primary key
- Like-named attributes are equivalent– they have the same meaning and the same range of values.

Customer Dimension
Customer ID (PK)
Customer Name
City
State
Zip
Date of first contact

Customer dimension from the Sales Transaction schema

Customer Dimension
Customer ID (PK)
Customer Name
City
State
Zip

Customer dimension from the Promotion Event schema.

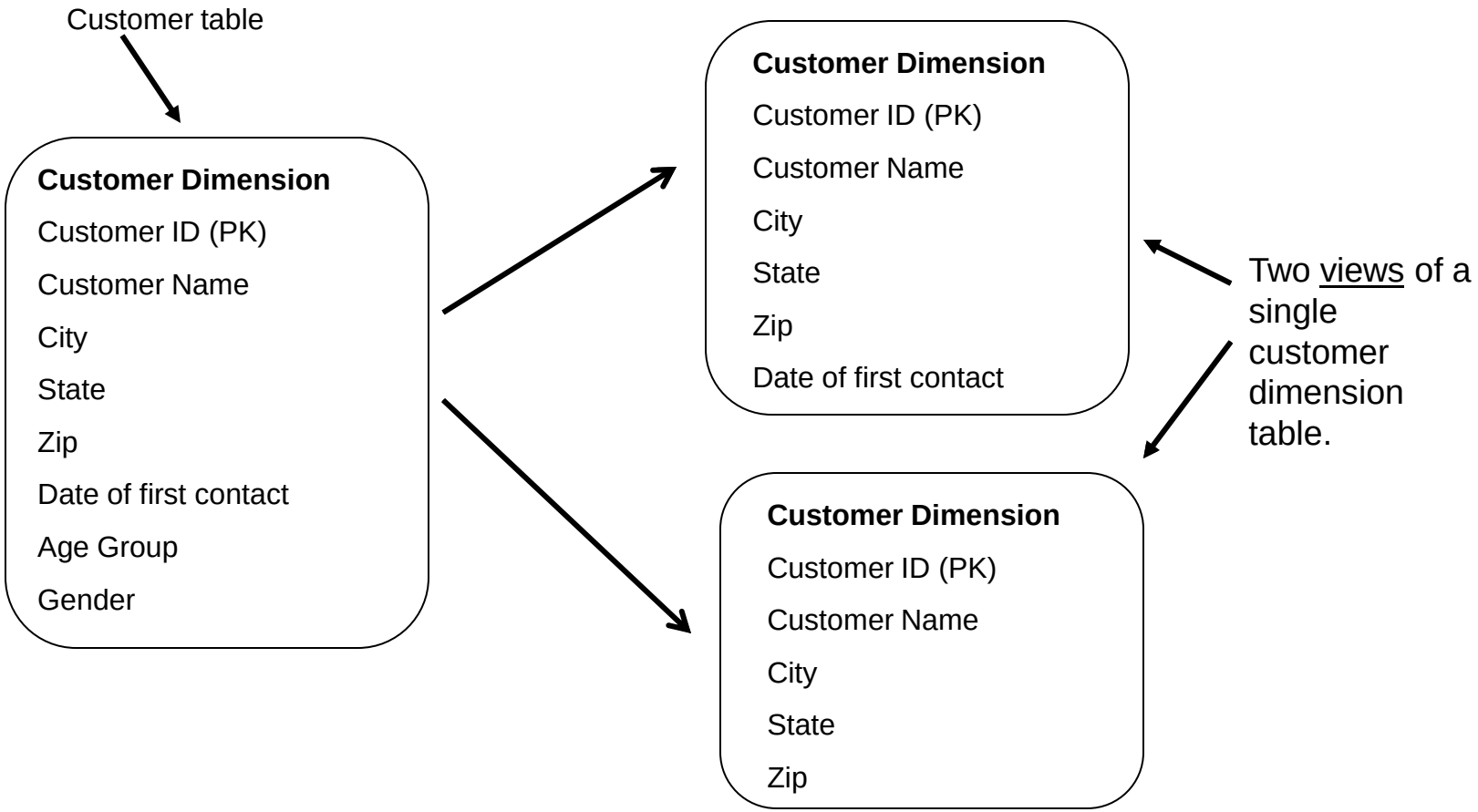
A subset of transaction customer

One set of dimension attributes (table) may be a subset of the another

Conformed Dimensions

"Conformed dimension" is a logical concept.

The conformed dimension that is shared between two dimensional models may be a single physical table.



Slowly Changing Dimensions

The value of a dimension attribute may change.

For example, the Channel Type for Store #0720 may change:

Channel ID (PK)	Channel Name	Channel Description	Channel Type
1042	**Store #0720**	**St. Louis Retail Store**	**Retail Store**

Channel ID (PK)	Channel Name	Channel Description	Channel Type
1042	**Store #0720**	**St. Louis Outlet Store**	**Outlet Store**

Slowly Changing Dimensions

There are three generally accepted ways to handle slowly changing dimensions (SCD):

Type 1 – simply replace the value old attribute value with the new.

Type 2 – insert a new dimension row, with a new key, representing the changed attribute. The old version of the dimension, with its original key, remains.

Type 3 – Design the dimension table with columns that hold previous values of the attribute anticipated to change.

SCD Type 1

Channel ID (PK)	Channel Name	Channel Description	Channel Type
1042	**Store #0720**	**St. Louis Retail Store**	**Retail Store**

Channel ID (PK)	Channel Name	Channel Description	Channel Type
1042	**Store #0720**	**St. Louis Outlet Store**	**Outlet Store**

Note the same primary key: 1042

SCD Type 2

Channel ID (PK)	Channel Name	Channel Description	Channel Type
1042	**Store #0720**	**St. Louis Retail Store**	**Retail Store**

Channel ID (PK)	Channel Name	Channel Description	Channel Type
1042	**Store #0720**	**St. Louis Retail Store**	**Retail Store**
1099	**Store #0720**	**St. Louis Outlet Store**	**Outlet Store**

SCD Type 3

Channel ID (PK)	Channel Name	Current Channel Type	Previous Channel Type
1042	Store #0720	Outlet Store	Retail Store

- SCD Type 3 tables often include columns indicating when a change happened.
- SCD Type 3 designers must anticipate the number of change events to store.

What can dimensional modeling do for you...

- **Single version of the truth**

  The dimensional model enforces business rules so the same fact or dimensional attribute always has the same definition.

- **Data integration**

  The dimensional model is built around data integration. The dimensional modeling process reveals inconsistencies and allows (or forces) them to be reconciled.

Bring together data from many different sources and create a **single**, **consistent** user view **.**

What can dimensional modeling do for you...

- **Analyze on the fly**

  The dimensional model facilitates ad hoc queries and unanticipated business questions because it is generic and not tied to any specific report structure or view of the data.

- **Drill up or drill down to any level of detail contained in the data**

  The dimensional model is a natural for summary reports and drill down applications. Dimensions are added for drill down, removed for summaries. Commonly used summaries may be pre-aggregated for improved performance.

Supports the query side of the business process: both standardized reporting and ad hoc queries.

What can dimensional modeling do for you...

- **Enterprise-wide data warehouse or specialized data mart**

  The dimensional model works equally well with generalized corporate data warehouse schemes or “data marts” focused on specific departments or user groups. Small scale data marts can be expanded and large warehouse structures can be sub-setted to change your project scale in either direction.

- **Tool agnostic**

  Almost any BI tool supports dimensional models. You can use your favorite query tool while someone in the next department accesses the same data with a modeling application. SQL queries against a dimensional model all have the same general structure.

- **The data warehouse evolves with the organization**

  Adding new data sources and adapting to changes in current data sources is handled in a consistent, reproducible manner.

Maximize **flexibility** and **scalability.**

What can dimensional modeling do for you...

- **The dimensional model is all about queries**

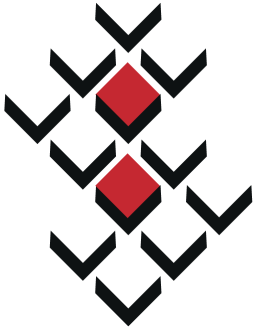
  The dimensional model is designed to make queries consistent, understandable, and fast. The dimensional model makes business data available to more users because query structure is less of a mystery.

- **Understandable**

  In the dimensional model, data relationships are consistent and typically no more than one level deep. This makes the data structure more understandable for experts and casual users alike. It also facilitates documentation and meta-data set up.

Optimize the **end-user** experience.

Contact Us With Questions



SYSTEMS SEMINAR CONSULTANTS, INC.

SAS® Training, Consulting, & Help Desk Services

(608) 278-9964

train@sys-seminar.com

www.sys-seminar.com

2997 Yarmouth Greenway Drive

Madison, WI 53711

sas® | SAS Alliance *Affiliate Member*