Fairfield University
DOLAN
School of Business

Databases for Analytics

Data Warehouse Design and Construction Tips

A Quick Review

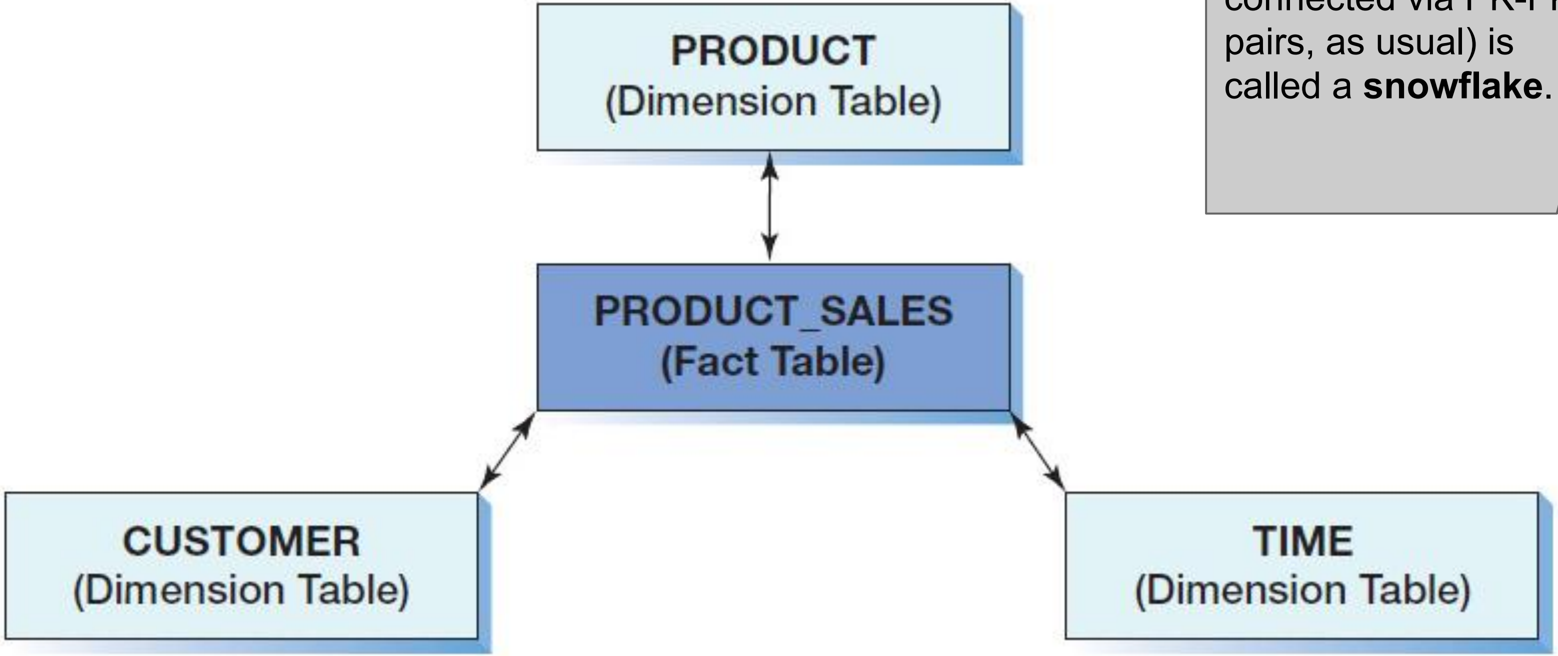
What the textbook says about Star Schema …
once more with feeling

What's a Dimensional Database?

Dimensional DBs are **relational,** with two kinds of tables:

- **Fact tables** with normalized **quantitative** data that can be aggregated as needed
  - tend to be huge, with many rows and columns
- **Dimension tables** that provide attributinal data (tags) that be used to **group** or **enrich/explain** rows in the fact tables
  - tend to be small, with just a few rows and columns

The Star Schema



A dimensional DB with more than one fact table (which can be connected via PK-FK pairs, as usual) is called a **snowflake**.

Ok, so what?

Ever used pivot tables?
Then you've done this!

When designed properly, a dimensional database reduces virtually all essential SQL queries to

SELECT *a few statistics and dimensional attributes*

FROM *one fact table joined with all applicable dimension tables*

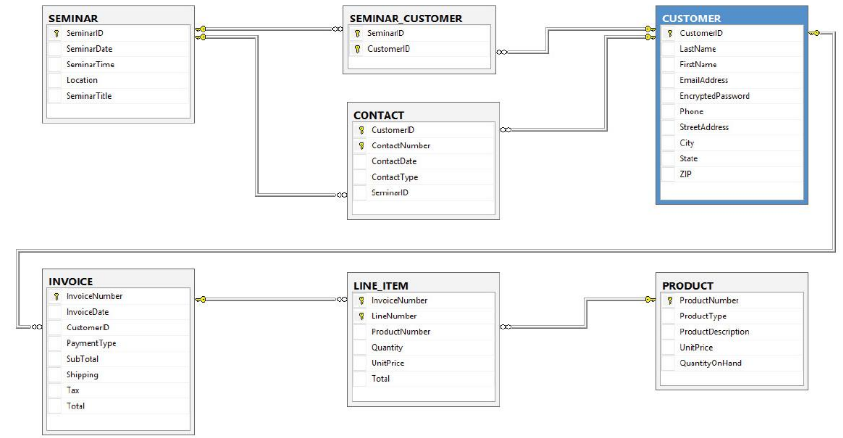
WHERE *the facts have specified attributes or conditions*

GROUP BY *one or more dimensional attributes*

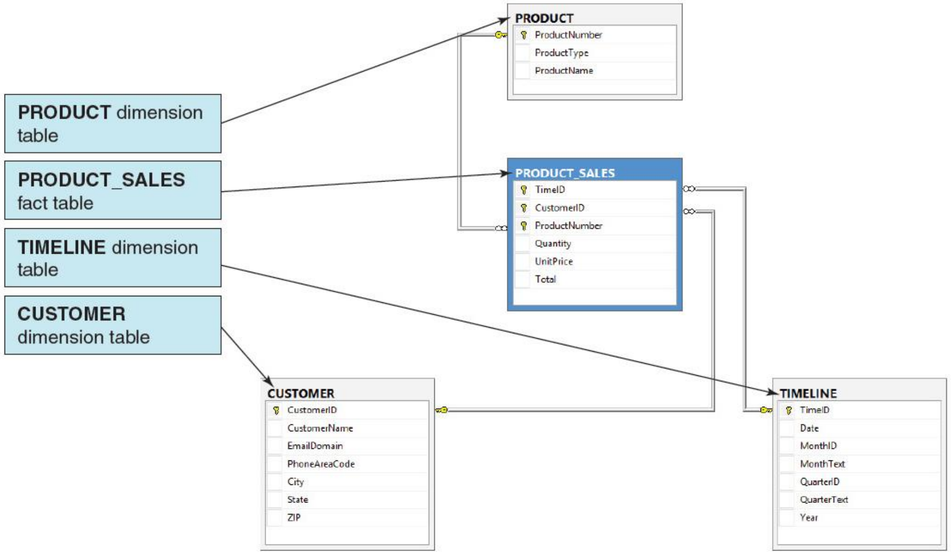
Common queries can be pre-defined (including JOINS), so that the user just has to click on a few attributes (to restrict on or group by) and a few predefined statistics.

A Side-by-Side Comparison

Operational DB



Dimensional DB


PRODUCT dimension table

PRODUCT_SALES fact table

TIMELINE dimension table

CUSTOMER dimension table

Star Schema Essentials

A Slightly More In-Depth Review

Star Schema Structure

Fact tables have two kinds of columns:

- Facts that capture **numerical data** (only) regarding some entity of interest
- Foreign Keys that connect to **dimensions**

Dimension tables tag the facts with descriptive detail:

- Who, what, where, when, how, and why
- Any non-numerical data belongs in the dimensions
- Dimensions are **the labels** we apply to facts so we can search, filter, and group the facts

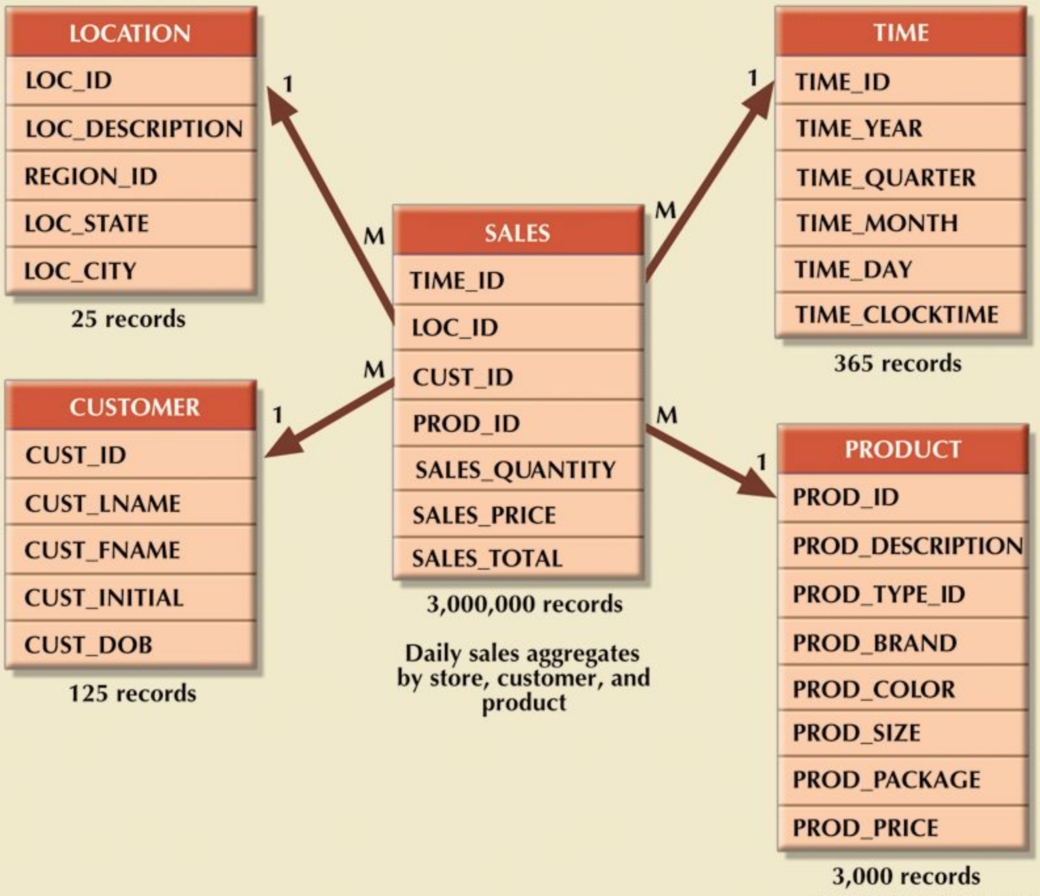
Dimensions

- If it can **classify a fact** then it is a dimension
- Each dimension has one or more **attributes**, which can provide related details on the same theme
- Each dimension record may have several levels of ***granularity***; an address might have ...
  - Street
  - Town
  - State
  - Region

Another Textbook Example

Notice how the dimensions are almost the same as the ones from our textbook?

FIGURE 13.10 Star schema for SALES



LOCATION
- LOC_ID
- LOC_DESCRIPTION
- REGION_ID
- LOC_STATE
- LOC_CITY

25 records

CUSTOMER
- CUST_ID
- CUST_LNAME
- CUST_FNAME
- CUST_INITIAL
- CUST_DOB

125 records

SALES
- TIME_ID
- LOC_ID
- CUST_ID
- PROD_ID
- SALES_QUANTITY
- SALES_PRICE
- SALES_TOTAL

3,000,000 records

Daily sales aggregates by store, customer, and product

TIME
- TIME_ID
- TIME_YEAR
- TIME_QUARTER
- TIME_MONTH
- TIME_DAY
- TIME_CLOCKTIME

365 records

PRODUCT
- PROD_ID
- PROD_DESCRIPTION
- PROD_TYPE_ID
- PROD_BRAND
- PROD_COLOR
- PROD_SIZE
- PROD_PACKAGE
- PROD_PRICE

3,000 records

SOURCE: Course Technology/Cengage Learning

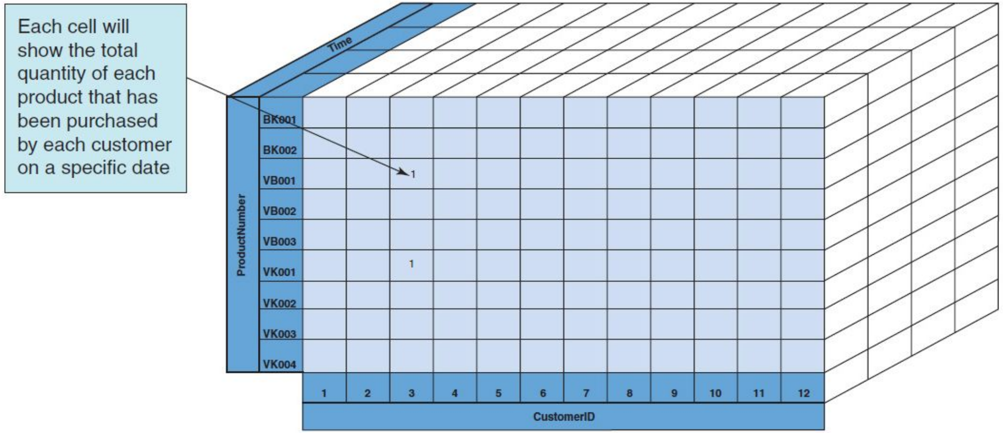
Some Common Dimensions

- Time Slices (note: not just points in time)
- Places / Locations
- People
  - Customers
  - Salespeople
  - Employees
- Event Types
- Products
- Organizational Units (departments, etc.)

Any of these could be at several levels of granularity. Each level would be a different column of the dimension table

Facts are Granular Too!

- Facts exist at the intersection of the dimensions
- The scope of each fact is determined by the granularity of its dimensions

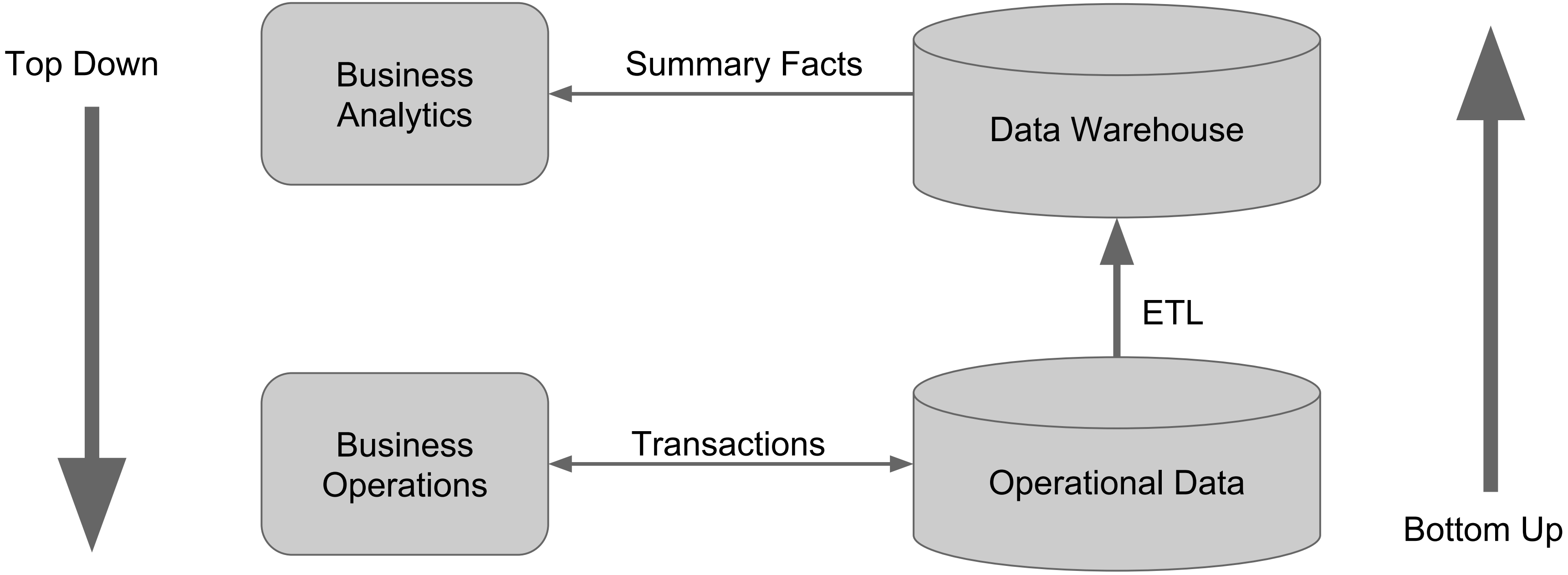


Each cell will show the total quantity of each product that has been purchased by each customer on a specific date

	1	2	3	4	5	6	7	8	9	10	11	12
BK001												
BK002												
VB001			1									
VB002												
VB003												
VK001			1									
VK002												
VK003												
VK004												

Bottom Up vs Top Down

Sometimes where you end up
depends on how you start


Top Down
Business Analytics
Summary Facts
Data Warehouse
ETL
Business Operations
Transactions
Operational Data
Bottom Up

Bottom Up Design (Dimension-First)

This approach builds upward from the operational database design to the data warehouse design.

1. Each strong entity in the database is a dimension
2. The fact table **rolls up** the data between the dimension tables by counting, summing, data from the intervening **transaction** tables
3. The granularity of the facts is determined by the dimensions. Conversely, add more dimensions as the facts allow.

Top Down Design (Facts-First)

This approach starts with a set of desired analyses to guide the design of the operational database.

1. Create a list of quantities (facts) needed to answer specific analytical questions.
2. Define dimensions based on desired grouping and filtering criteria.
3. Design an ETL process to capture and tag the facts with the dimensions from the operational data.

Bottom Up → Top Down → ...

In practice, you will find yourself doing a little of both approaches, perhaps oscillating from one to the other:

- BU: Selecting dimensions based on existing data
- TD: Adding facts based on your analytical needs
- BU: Adding levels to the dimensions to control the granularity
- TD: Add data sources to expand the available facts …

Stop when the design supports the desired analyses or you run out of data sources.

(Re-) Building the DW

A data warehouse will get rebuilt/refreshed many, many, many times …
so try to be as efficient as you can

Top Secret Tips ...

- "Automate the Boring Stuff …" (DRY)
- Sometimes the dimensions are defined by the analysts by hand.
  - They don't have to exist in the operational database.
- Dimensions can sometimes have dimensional facts
  - A room dimension may include capacities
- Facts can sometimes relate to other facts. One fact acts like a dimension to another fact.
  - Think of households and bank accounts, for example.

Keep things simple

- Your data warehouse does not need to answer every question one might every want to ask.
- Focus on one potential user at a time: what does that user need to get his/her job done.
- You Aren't Gonna Need It (YAGNI)
- Let the fact table get as wide as you need it to be, even if that means violating normalization rules.
- Keep the dimension tables small and independent; they exist before you collect the first fact!

Fairfield University
DOLAN
School of Business

Databases for Analytics

Data Warehouse Design
and Construction Tips