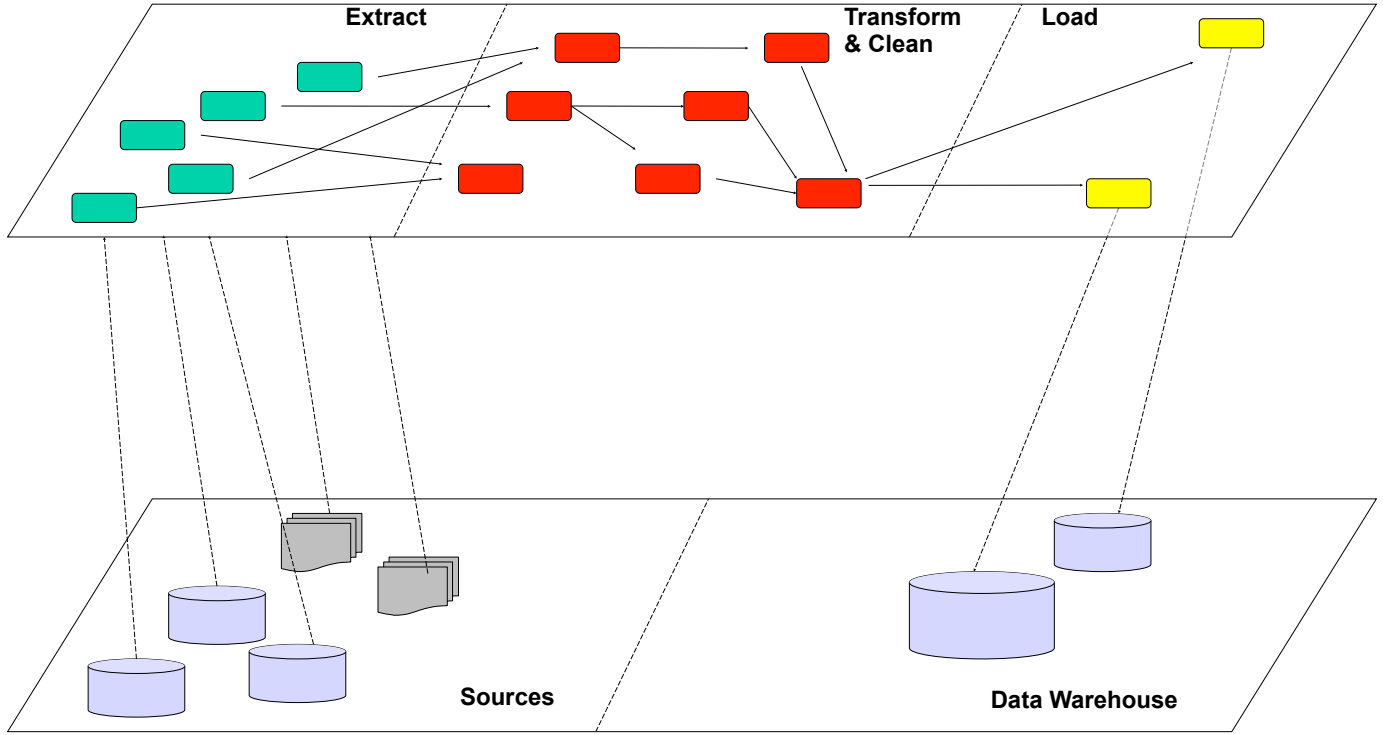
Data Science in the Wild

Lecture 12: Memory-Based Data Warehouses

Eran Toch

CORNELL TECH

Data Engineering

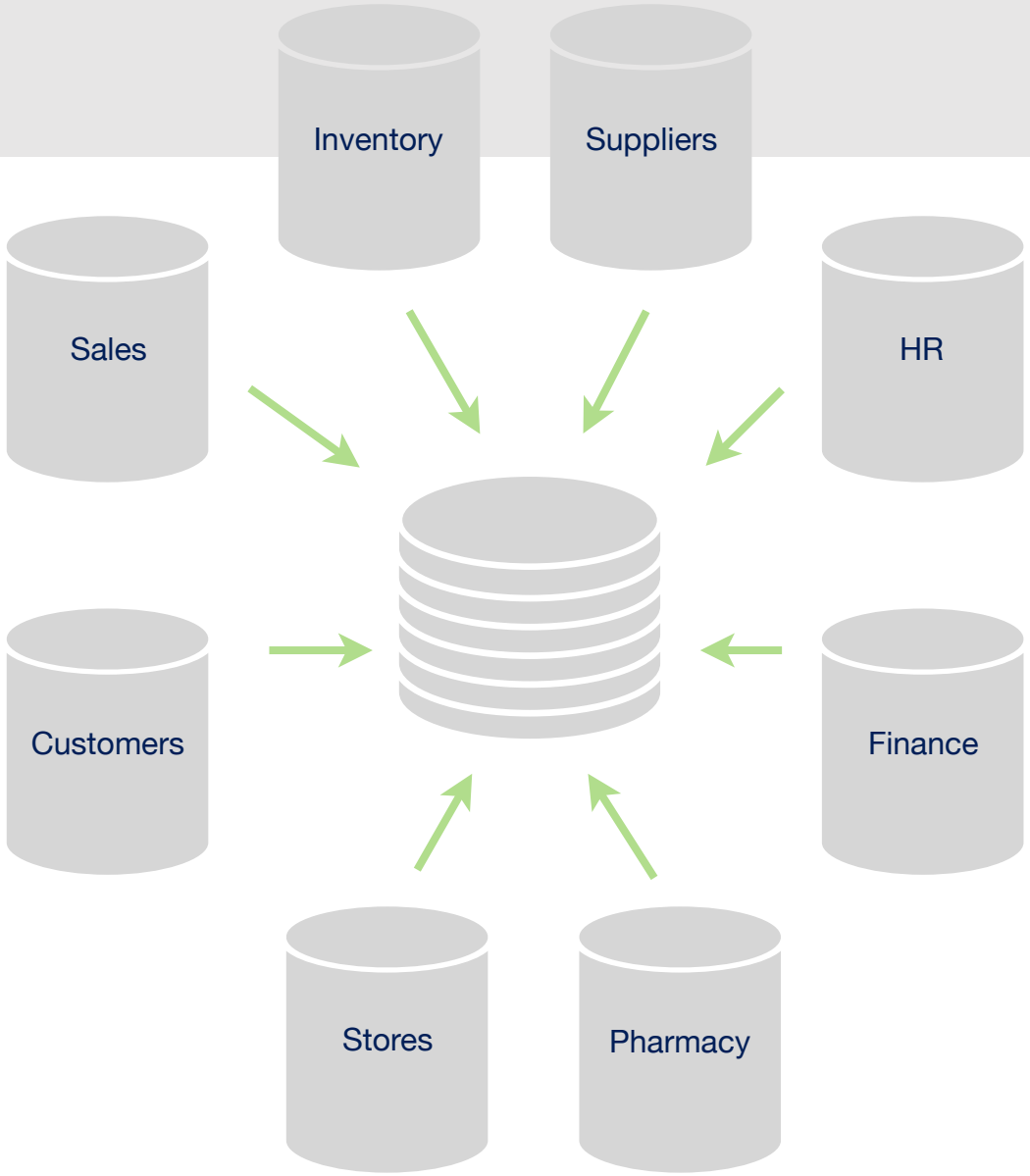


Agenda

1. What are Data Warehouse?
2. Data warehouse architecture
3. The design process
4. Transaction design
5. Periodic snapshot
6. Accumulative transactions

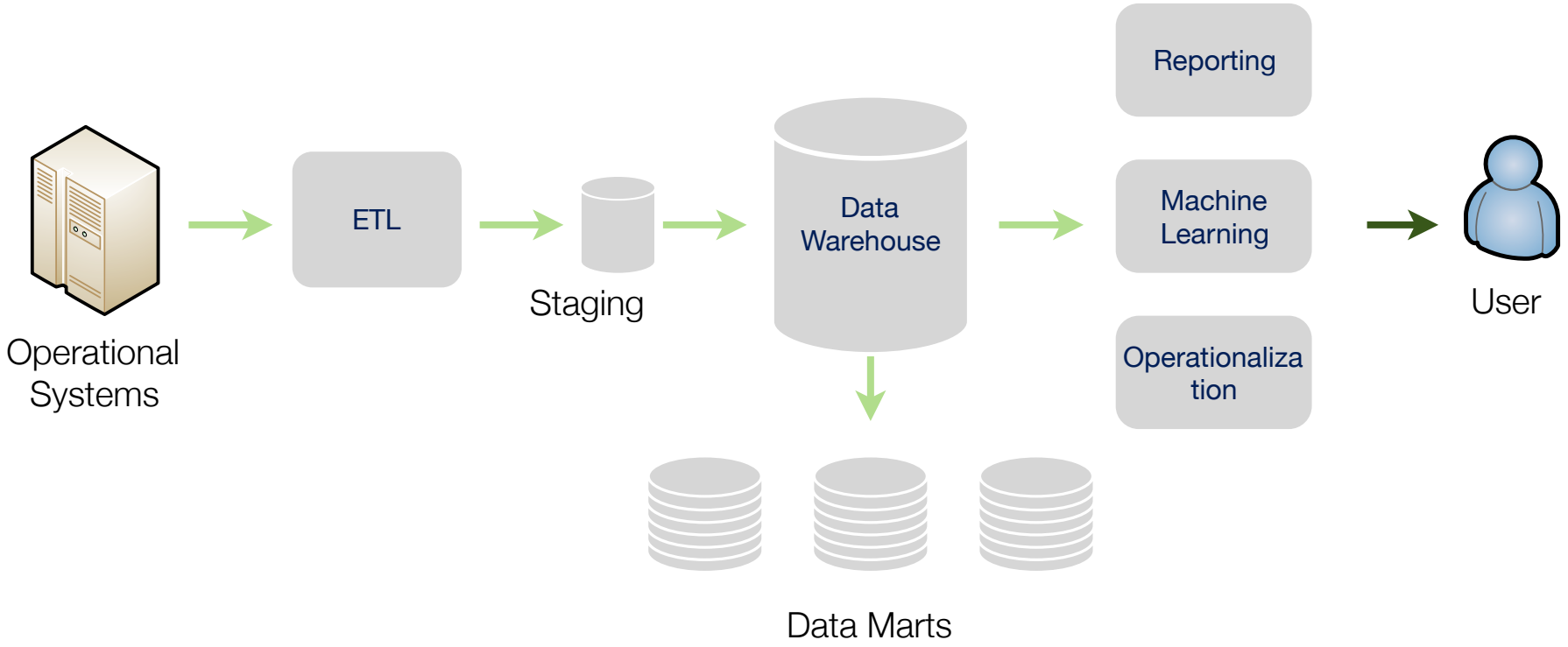
Data Warehouse: A definition

- A Data Warehouse is a central repository of integrated data from one or more disparate sources
- Data warehouses don't aim to solve a **single** problem
- Instead, they provide the infrastructure for an organizational data science process



Data Warehouse: Basic Architecture

- It stores current and historical data in one single place
- Data marts represent repositories for specific subjects (sales, orders, website navigation)



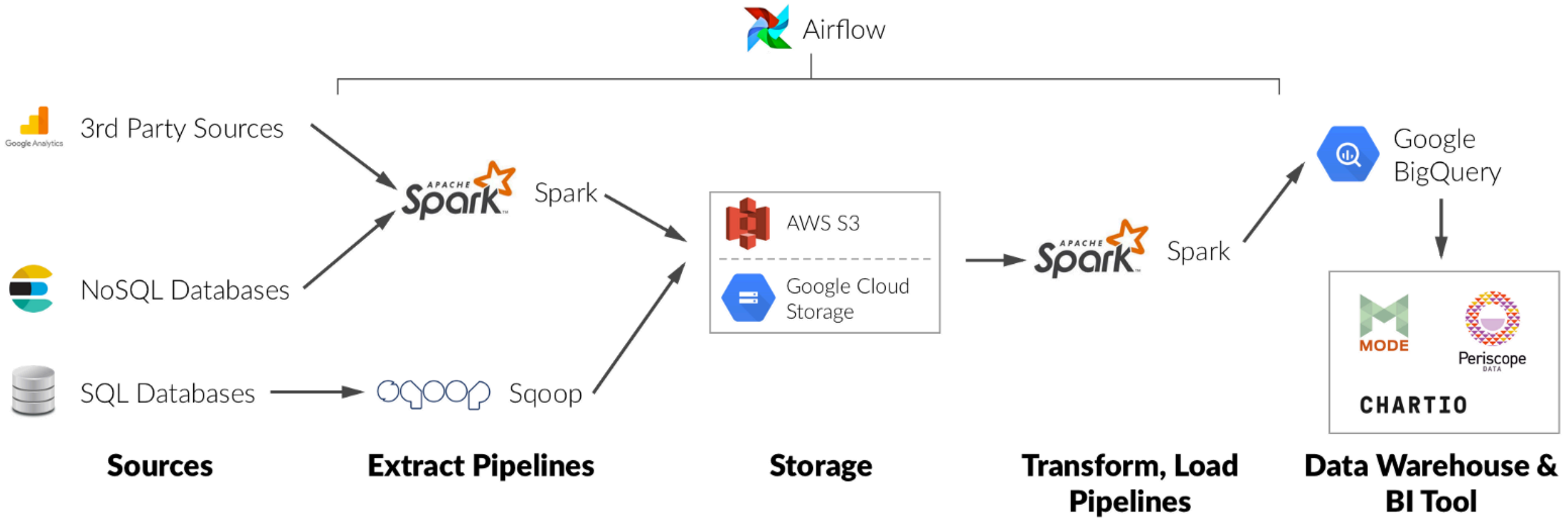
Technologies for Data Warehouse

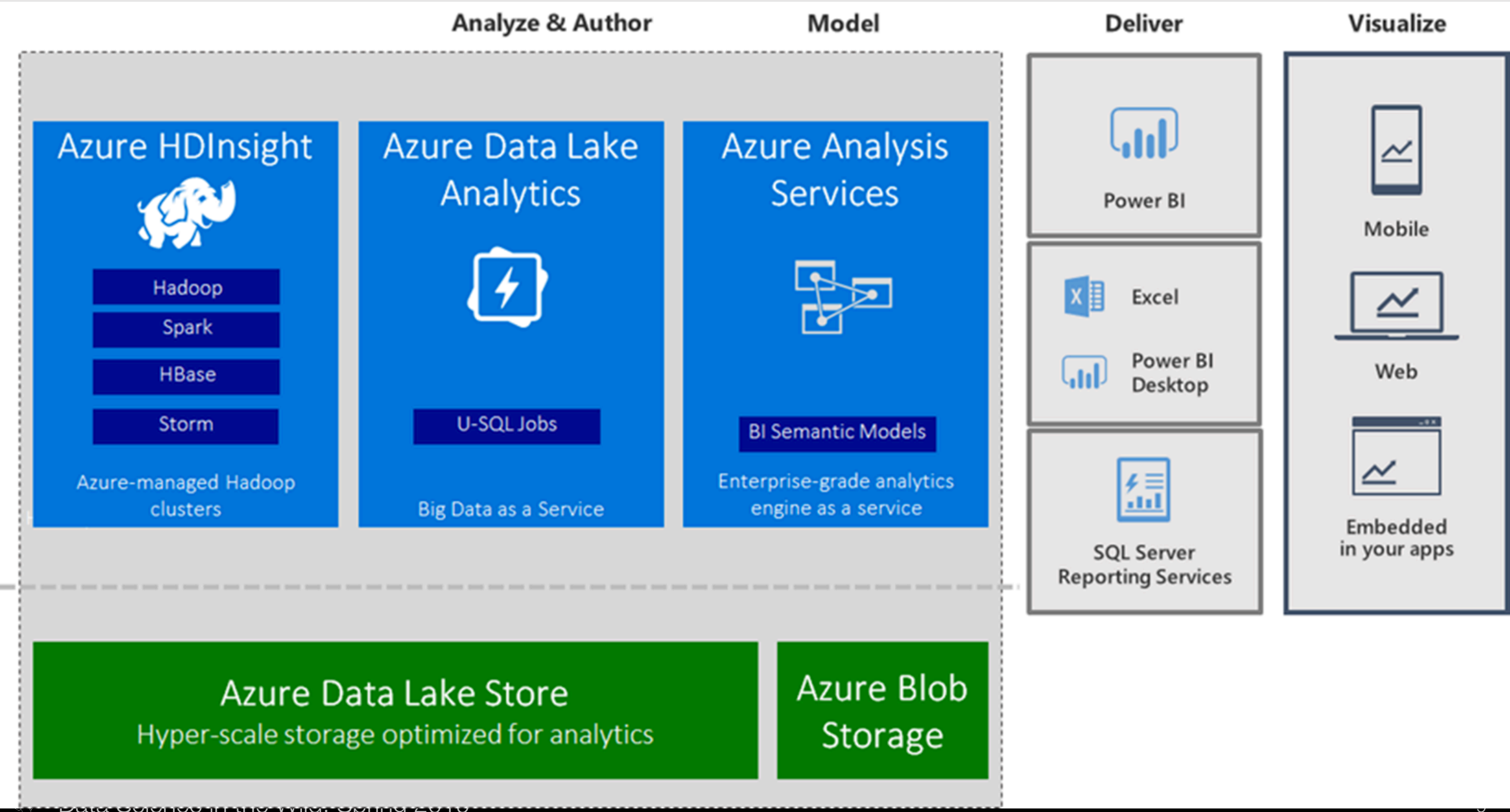
- RDBMS - Relational Database Management Systems
- Hadoop
- Hadoop / Spark
- And many other variations

Figure 1. Magic Quadrant for Data Management Solutions for Analytics

Source: Gartner (January 2019)

Spark-based Architecture



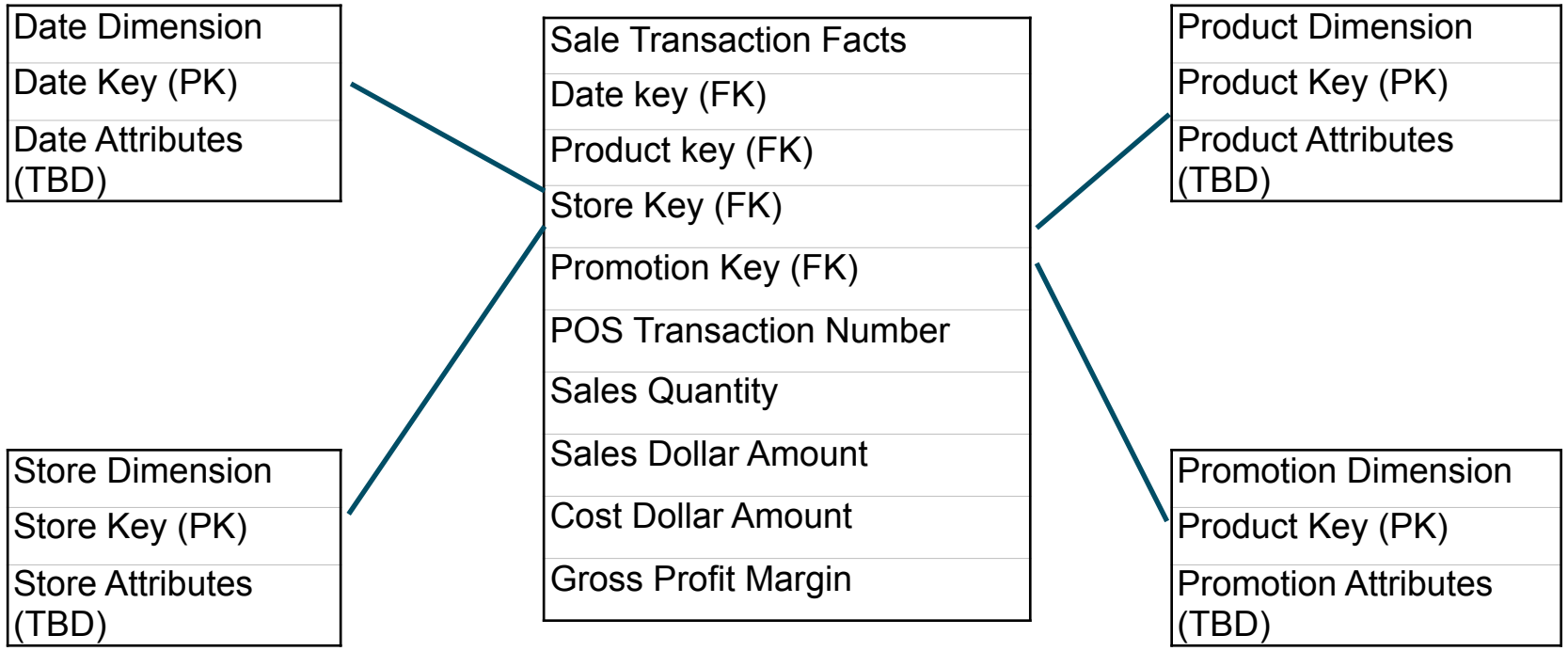

Analyze & Author
Model
Deliver
Visualize
Azure HDInsight
Hadoop
Spark
HBase
Storm
Azure-managed Hadoop clusters
Azure Data Lake Analytics
U-SQL Jobs
Big Data as a Service
Azure Analysis Services
BI Semantic Models
Enterprise-grade analytics engine as a service
Power BI
Excel
Power BI Desktop
SQL Server Reporting Services
Mobile
Web
Embedded in your apps
Azure Data Lake Store
Hyper-scale storage optimized for analytics
Azure Blob Storage

Comparison

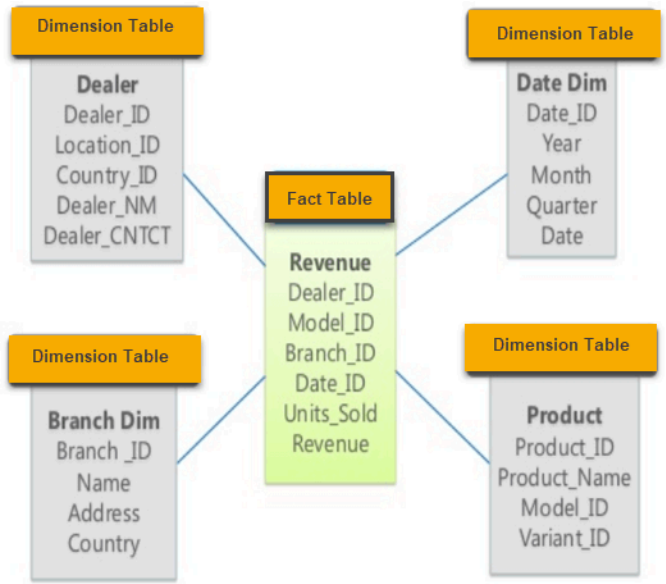
Operational Databases	Data Warehouses
Process Oriented	Subject Oriented
Add, Modify, Remove single rows	Bulk load, rarely modify, never remove
Online human / sensors entry	ETL jobs
Queries for small sets of rows with all their details	Scan large sets for aggregates
Using trained models	Training models

Designing the Data Warehouse

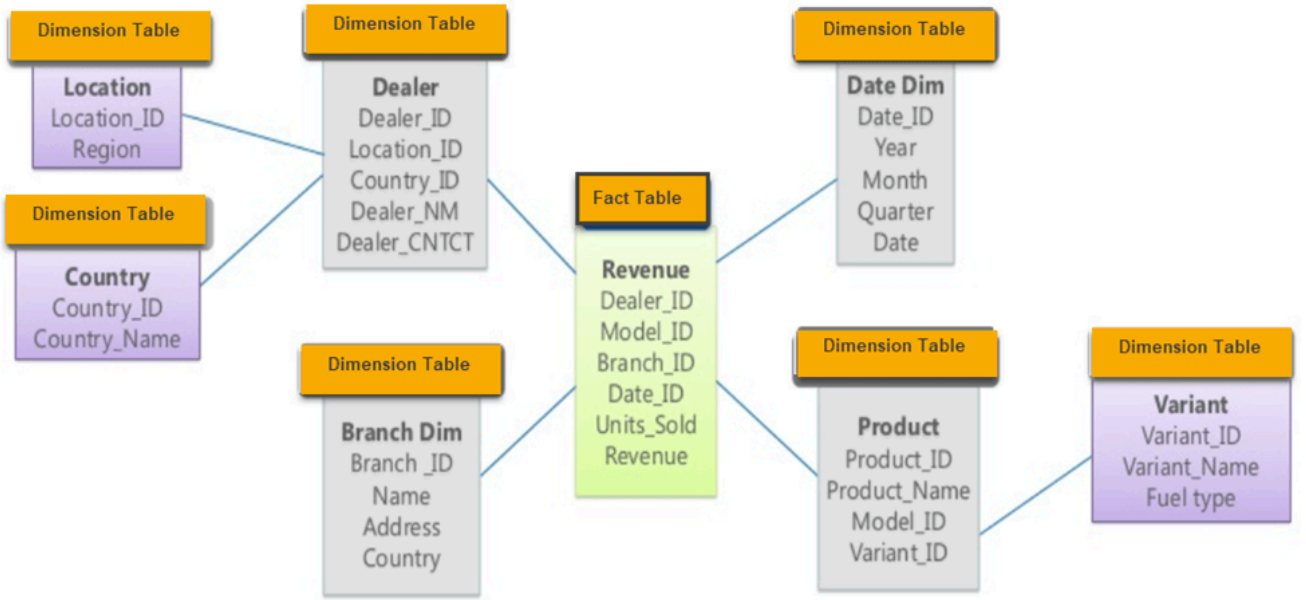
- Dimensional model of a business process:
  - The **facts** we want to analyze
  - The **dimensions** we analyze the facts



Two Super Architectures



Star Schema

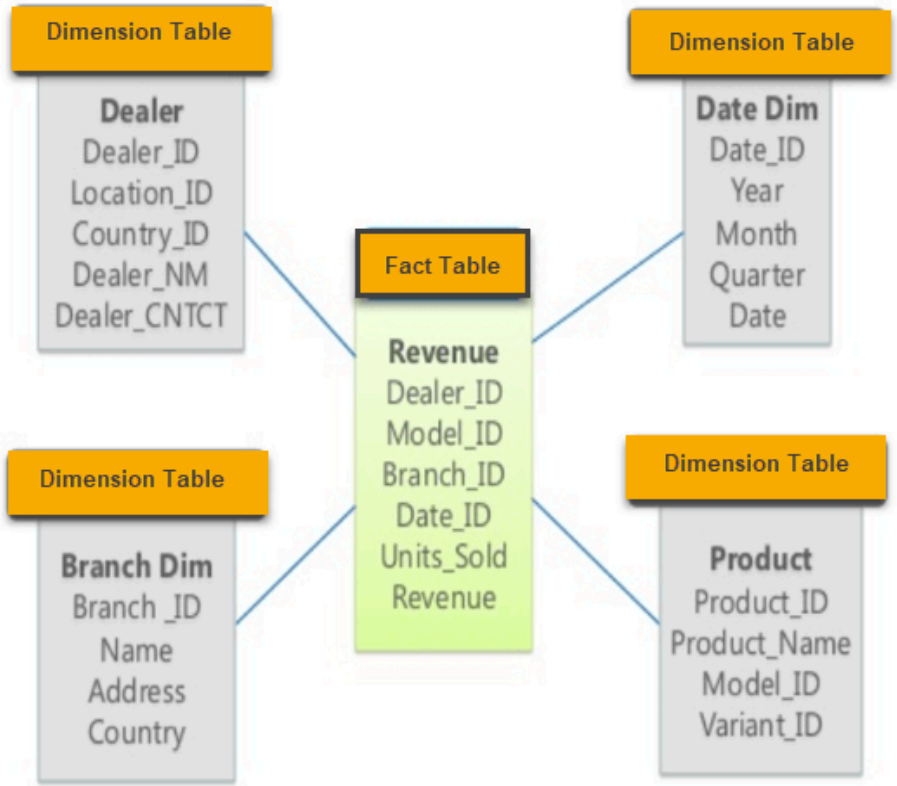


Snowflake

https://www.guru99.com/star-snowflake-data-warehousing.html

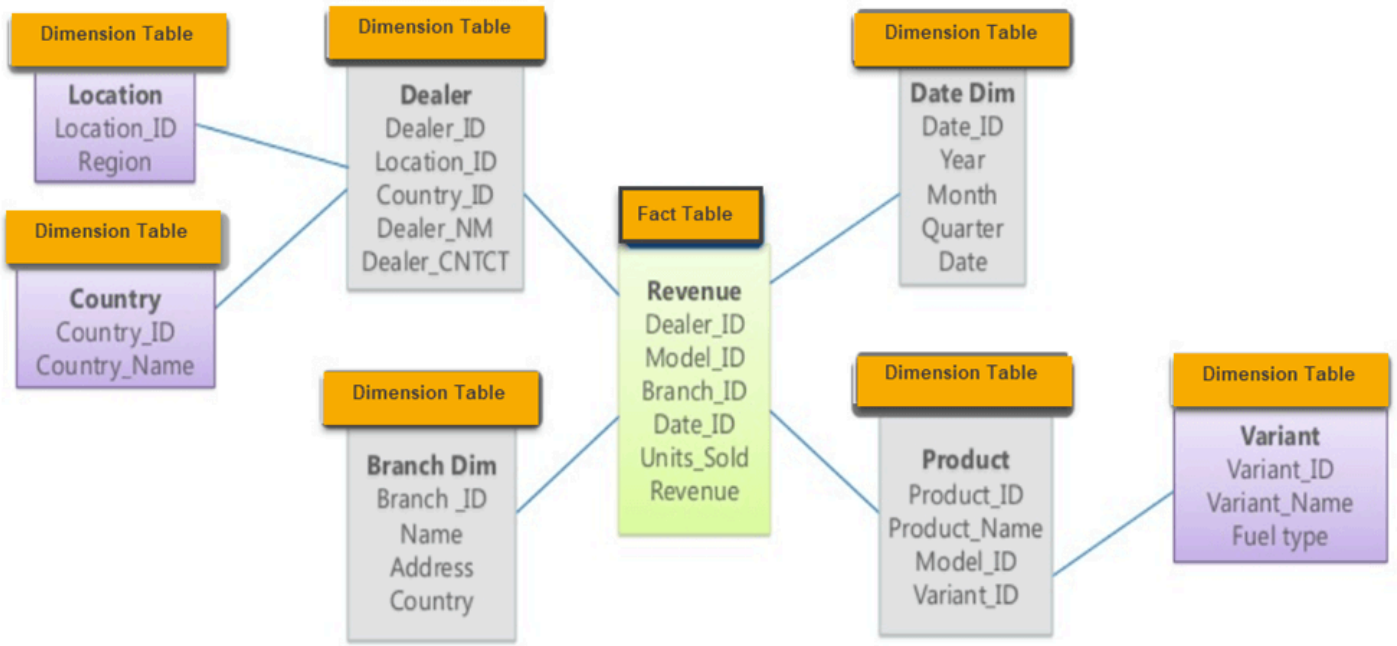
Star Schema

- Fact table contains a key and measure
- Every dimension in a star schema is represented with the only one-dimension table
- The dimension table is joined to the fact table
- Dimension tables are not joined to each other
- The dimension tables are not normalized
- Main advantages
  - Queries are simpler
  - Optimizes number of joins



Snowflake

- The dimension tables are normalized which splits data into additional tables
- Main advantages:
  - Optimizes storage
  - Easier to understand
  - Easier to engineer the dimensions (adding, removing etc)



4 Step Design Process

1. Identify the business process
2. Identify the facts
3. Declare the grain
4. Choose the dimensions

Case Study: A Retail Sales Operation

- ~2000 Stores
- ~$75.17B yearly revenue
- Typical 80K individual products (SKU's)
  - In a store
  - In any given moment
- ~10 departments
  - Food, medicine, cosmetics, nature, kids...

Step 2: Identify the Facts

- Identifying the numeric facts for analysis
- Facts are determined by answering the question, "What are we measuring?"
- All candidate facts in a design must be true to the grain defined in step 2
- Facts that clearly belong to a different grain must be in a separate fact table

- **Examples**
  - Sales quantity
  - Sales dollar amount
  - Cost dollar amount
  - Gross profit margin

Step 3: Find the Grain

- Declaring the grain means specifying exactly what an individual fact table row represents.
- The grain conveys the level of detail associated with the fact table measurements.
- It provides the answer to the question, "How do you describe a single row in the fact table?"

THE SMALL GOLDEN DISK IS A PIECE OF PURE GOLD WEIGHING ONE TROY GRAIN.

Examples

- Total sale for each customer
- A record for an individual line item on a customer's retail sales ticket as measured by a scanner device
- A record for each item

Grain Design

- What are the options?
  - For example, summary of sales per day per store... (what's the problem?)
- The grain of data should support
  - The ability to drill down
  - The ability to support independent dimensions
- Selected option:
  - Individual line item on a POS

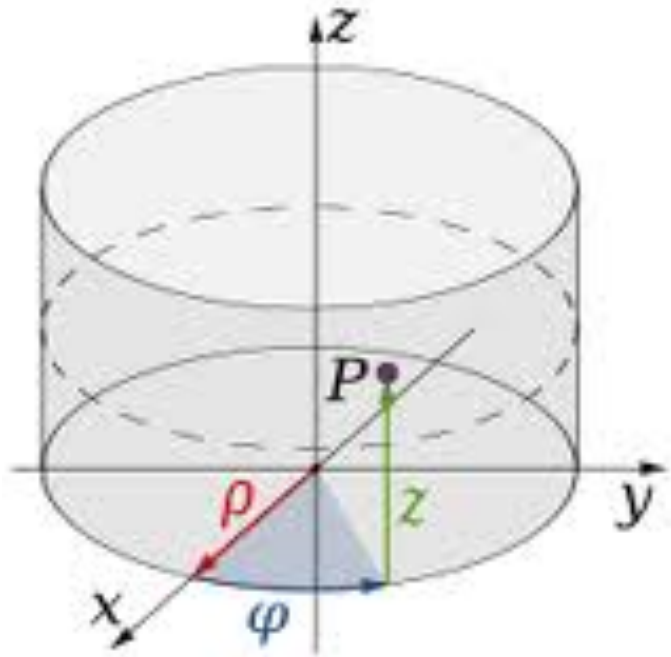
POS Sale Transaction Facts
POS Transaction Number
Sales Quantity
Sales Dollar Amount
Price reduction

Good Grains

- How should we judge a design?
- Preferably you should develop dimensional models for the most atomic information captured by a business process
- Atomic data is the most detailed information collected; such data cannot be subdivided further

Step 4: Choose Dimensions

- Dimensions can be designed by thinking about:
  - "How do business describe the data that results from the business process?"
- We want a robust set of dimensions representing all possible descriptions that take on single values in the context of each measurement



Summary

1. Identify the business process
2. Identify the facts
3. Declare the grain
4. Choose the dimensions

Grain Design Pattern

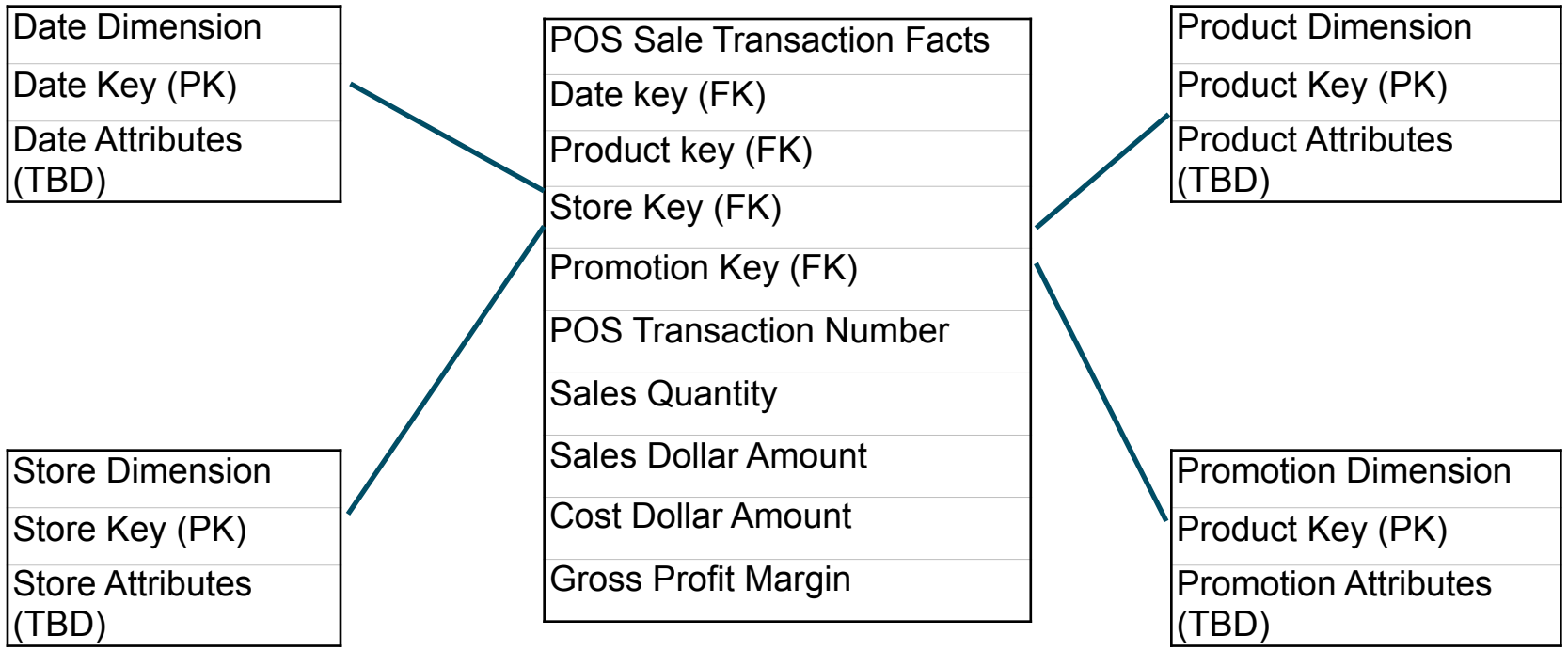
- **Transactions**
  - e.g., POS transaction, financial transaction, medical action
- **Periodic snapshot**
  - Inventory state in a given day, closing stock price
- **Accumulative transactions**
  - Order management, hospitalization process
- Other types…

<4> Transaction design

Case Study Data

- Point of Sale (POS):
  - Individual product purchase (as scanned at the POS)
- Supply:
  - the purchase price of each product
- Promotions
  - Temporary price reductions
  - Ads
  - Inserts
  - Coupons

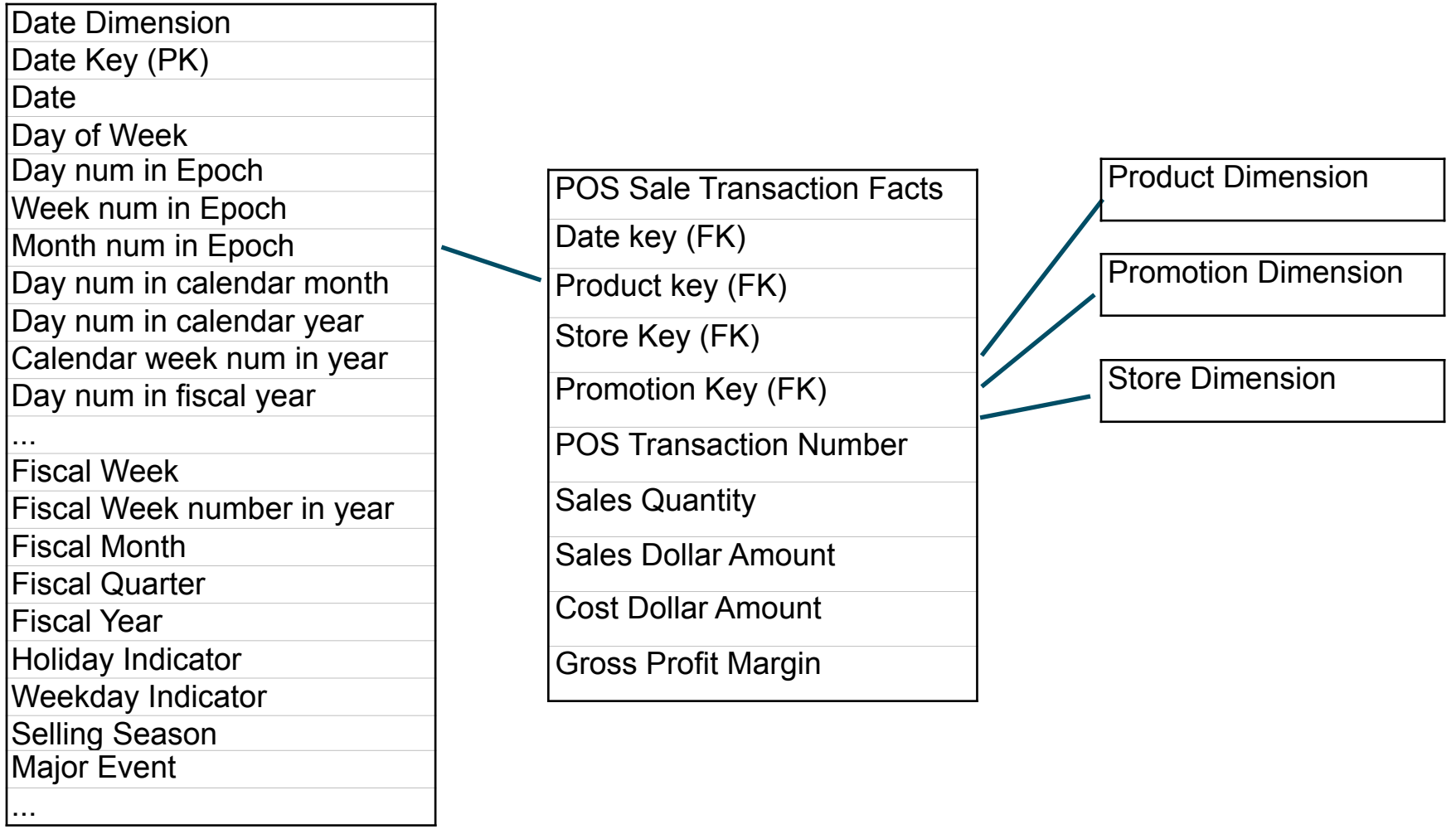
Preliminary Star Schema



Date Dimension

- Almost always exists in DW systems.
- Can be built and populated in advance
  - Why is this important?
  - Each row represents one day (or one time unit)
  - 10 years = 3,650 rows

Date Dimension



Date Example

- There are many date attributes not supported by the SparkSQL date function, including fiscal periods, seasons, holidays, and weekends.
- Rather than attempting to determine these non-standard calendar calculations in a query, we should look them up in a date dimension table

Date Key	Date	Full Date Description	Day of Week	Calendar Month	Calendar Year	Fiscal Year-Month	Holiday Indicator	Weekday Indicator
1	01/01/2002	January 1, 2002	Tuesday	January	2002	F2002-01	Holiday	Weekday
2	01/02/2002	January 2, 2002	Wednesday	January	2002	F2002-01	Non-Holiday	Weekday
3	01/03/2002	January 3, 2002	Thursday	January	2002	F2002-01	Non-Holiday	Weekday
4	01 /04/2002	January 4, 2002	Friday	January	2002	F2002-01	Non-Holiday	Weekday
5	01/05/2002	January 5, 2002	Saturday	January	2002	F2002-01	Non-Holiday	Weekend
6	01/06/2002	January 6, 2002	Sunday	January	2002	F2002-01	Non-Holiday	Weekend
7	01/07/2002	January 7, 2002	Monday	January	2002	F2002-01	Non-Holiday	Weekday
8	01/08/2002	January 8, 2002	Tuesday	January	2002	F2002-01	Non-Holiday	Weekday

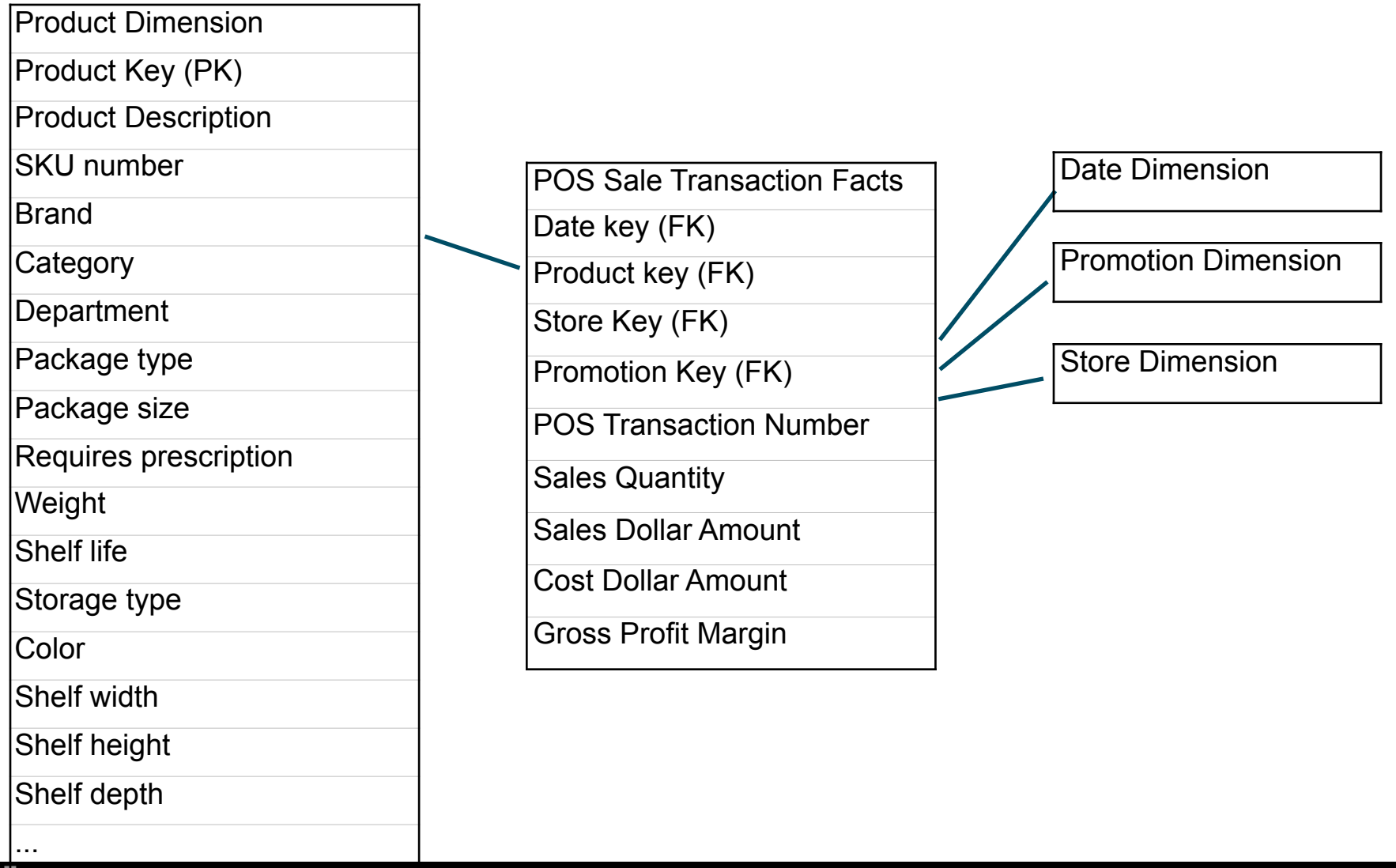
Advantages of Date Dimension

- Indexing integer-based key is faster than date key
- Simplifies calculations
- Simpler comparisons
  - Month vs. Month
  - Year vs. Year
  - Day of week, day of month
  - Special events

Product Dimension

- 60K current products -> 150K distinct products (SKUs)
- Contains hierarchy:
  - Department
    - Category
      - Brand
        - SKU

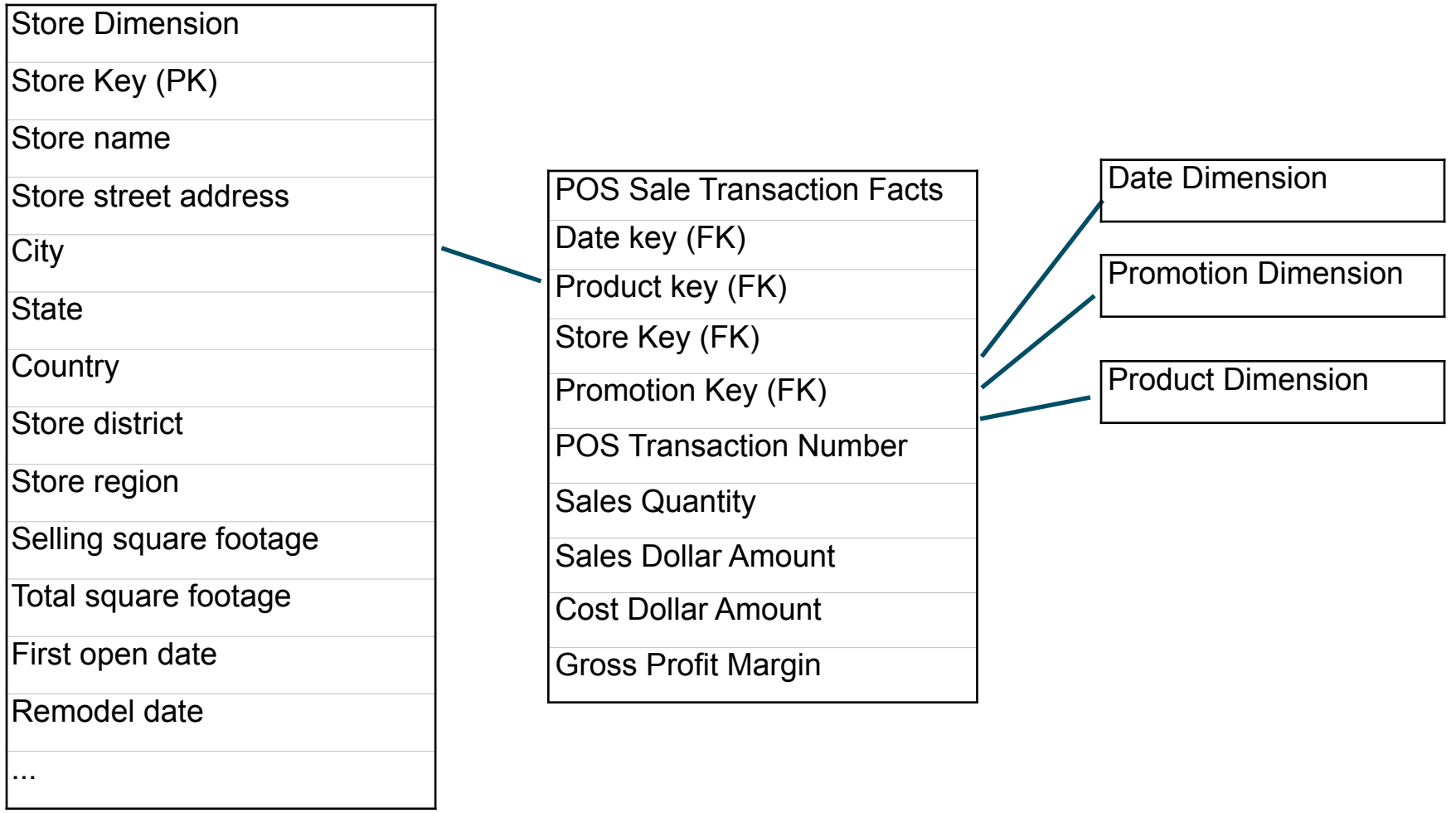
Product Dimension



Store Dimension

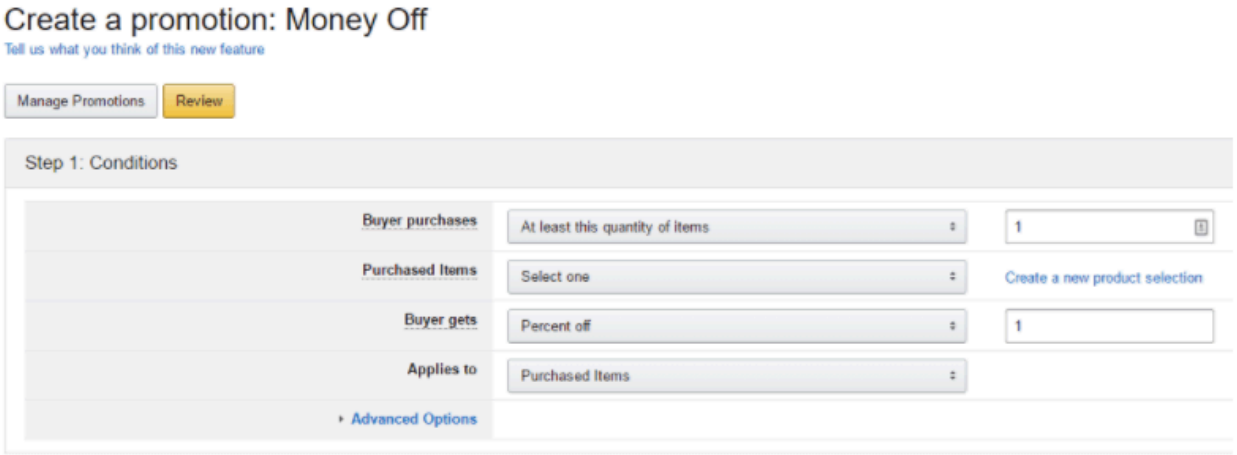
- Business questions:
  - Store performance
  - Geographical questions
  - Floor plan (crucial for retail)
  - Size and diversity

Store Dimension

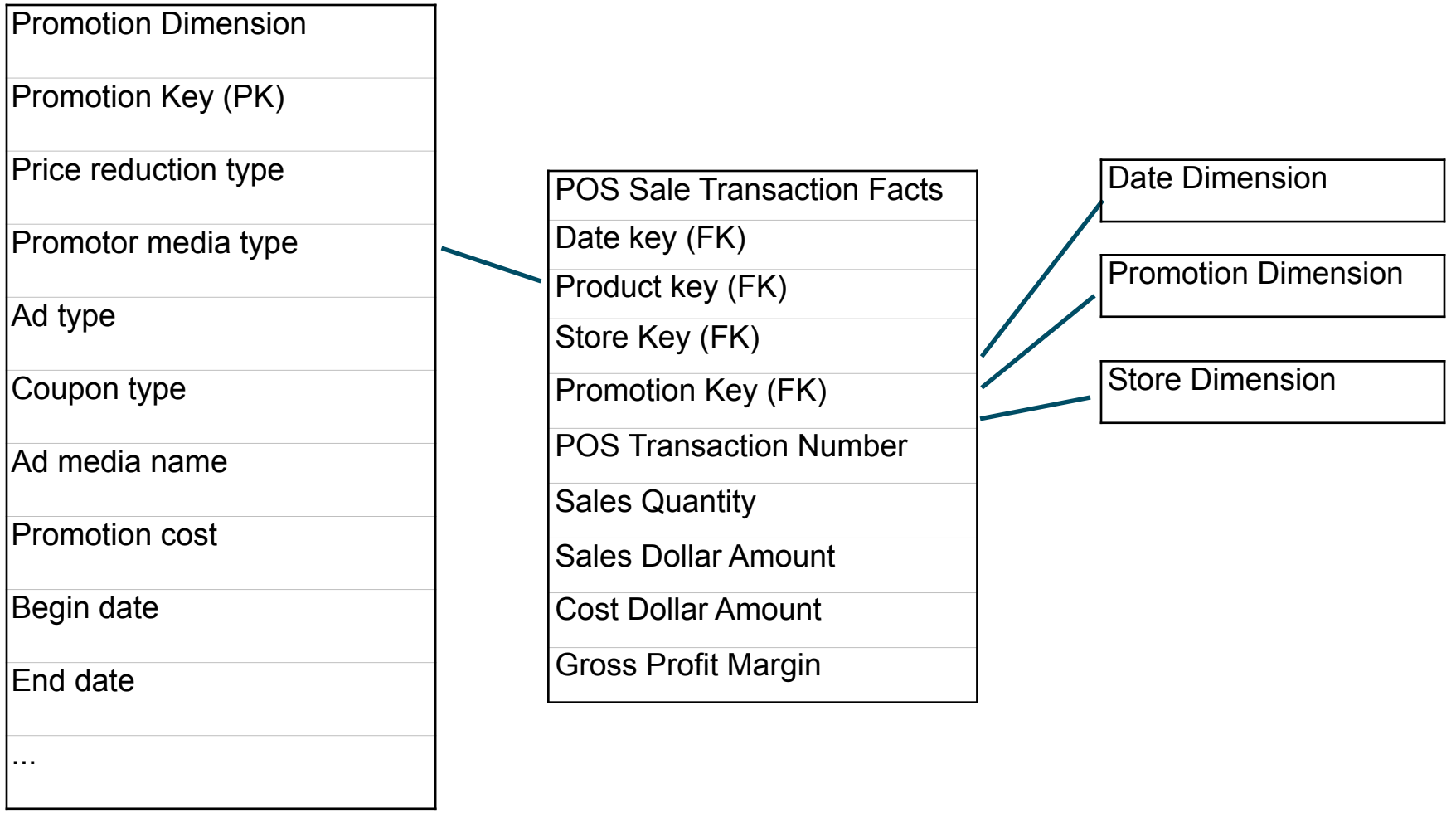


Promotion Dimension

- Business analysis:
  - Sales lift, compared with baseline
  - Cannibalization
  - Profit
  - Market growth
- Not all promotions appear in POS
  - An example of how manual data enrichment can improve business analysis



Promotion Dimension



<5> Periodic Snapshot

Case Study: Store Inventory

- Our story:
  - Inventory of different products is managed in different stores.
  - The business process we are interested in analyzing is the retail store inventory.
- What would be the grain and main dimensions of the data warehouse?

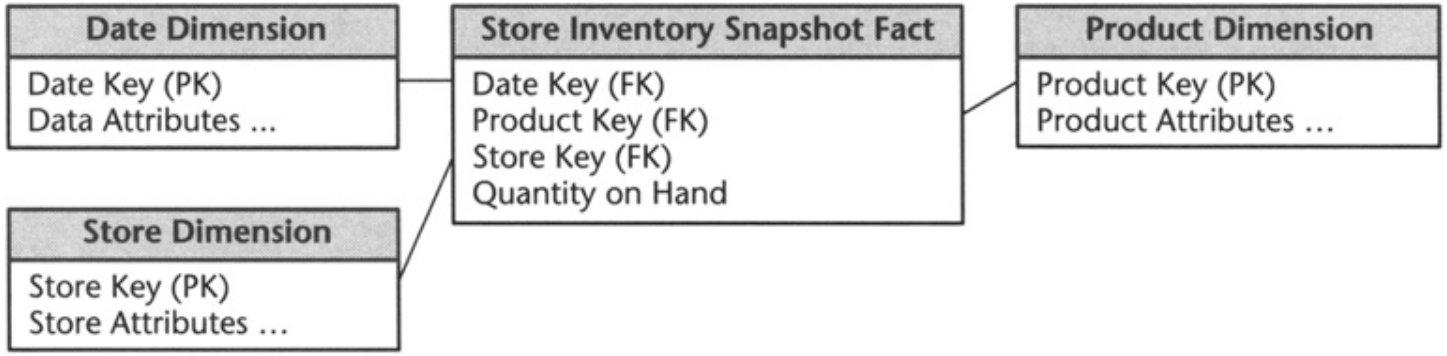
Analyzing Inventory Levels

- Optimized inventory levels in the stores can have a major impact on profitability
- Minimizing out-of-stocks and reducing overall inventory carrying costs
- The retailer needs the ability to analyze daily quantity-on-hand inventory levels by product and store
- Which questions would a business ask to better manage inventory?
    - Inventory value
    - Inventory size
    - Inventory turnover
    - Process timing
    - Planning procurement
    - physical planning
    - ...

Periodic Snapshot

Grain:

- We want to see daily inventory by product at each individual store, which we assume is the atomic level of detail provided by the operational inventory system.



Additivity

- Quantity is additive?
  - Yes for store and product
  - No for date
- Therefore it is semi-additive
- Is there another function that logically summarize quantity over time?

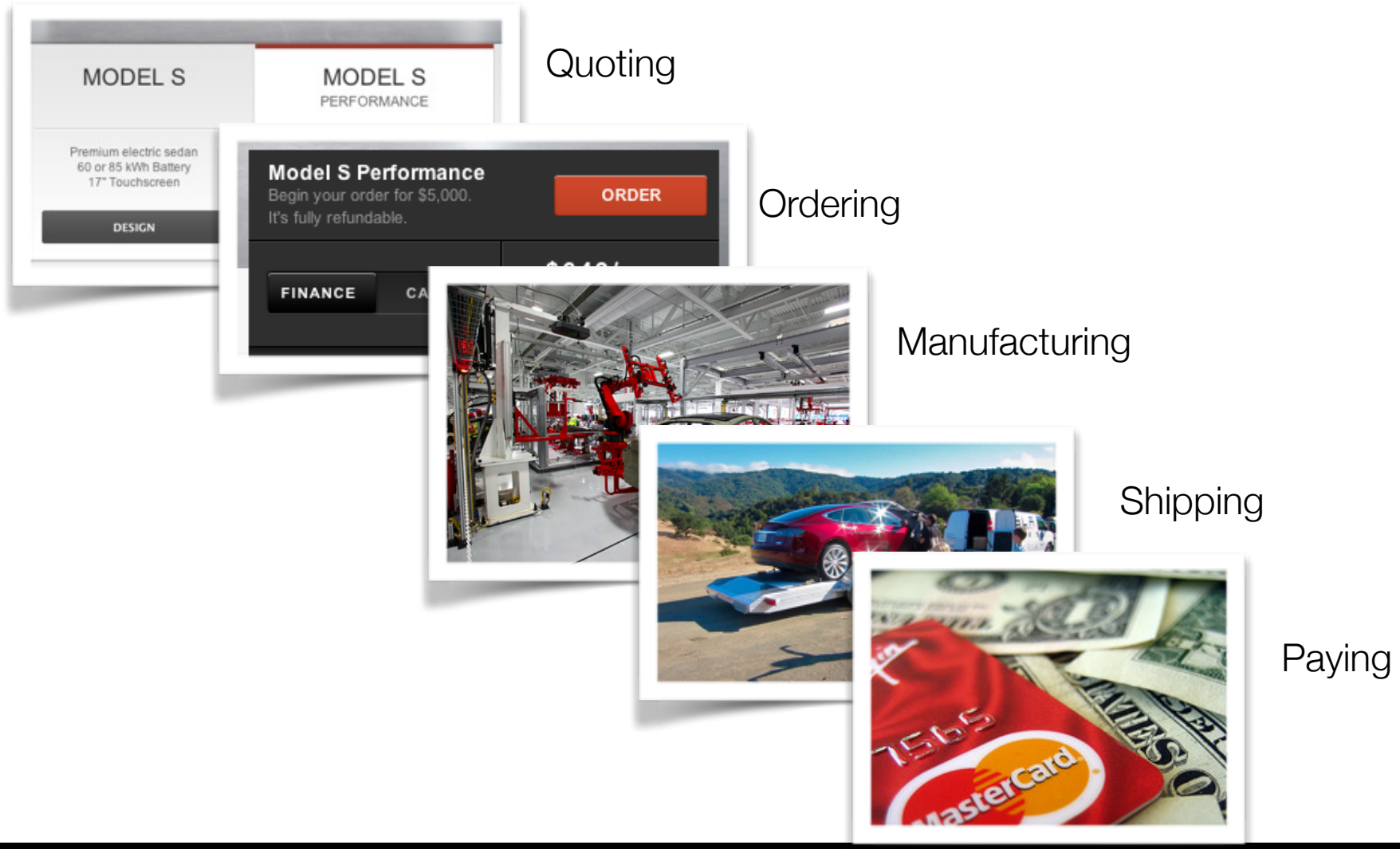
What is Still Missing

- The periodic snapshot is the most common inventory schema
- However, it does not teach us everything we need to know about inventory
- What is missing?

Table Size

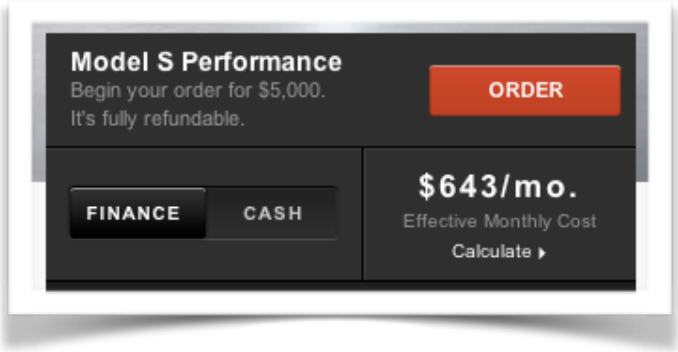
- Unlike the transaction table, where there is a record only if a transaction occurred...
- This table has a record for each product/store/date, even if the inventory did not change (or if the quantity is 0)

Example: Order Management



Types of Business Processes

- We have seen single stage processes:
  - Purchasing
  - Procurement
- But what about multi-stage processes, like:
  - Order management
  - Patient hospitalization
  - Sales process
  - Manufacturing process



Ordering

Business Process Bus Matrix

The business process bus matrix shows which business processes require what types of data.

	Date	Product	Customer	Deal	Sales Rep	Ship From	Shipper
Quotes	X	X	X	X	X		
Orders	X	X	X	X	X		
Shipments	X	X	X	X	X	X	X
Invoicing	X	X	X	X	X	X	X

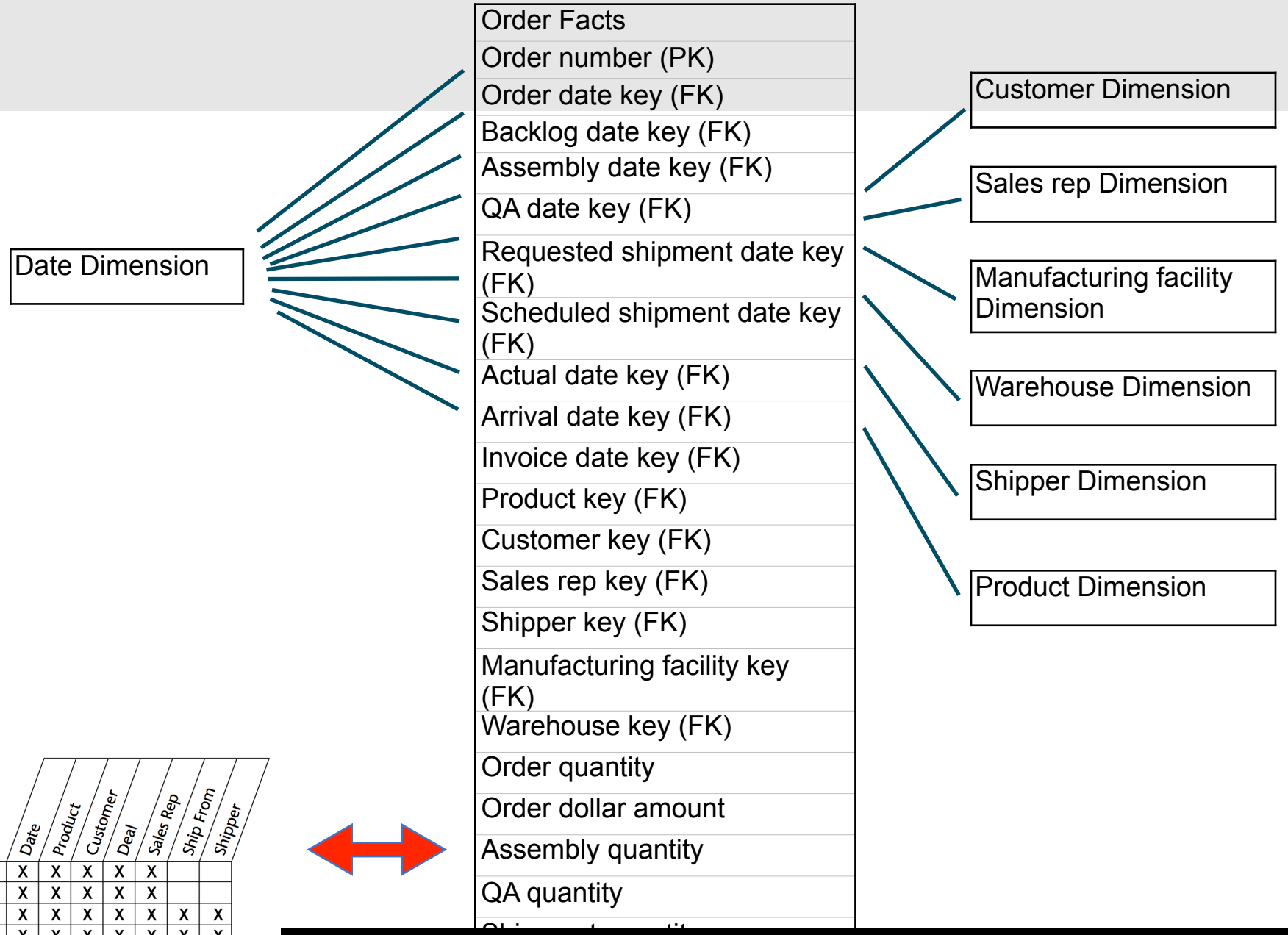
The data warehouse bus matrix closely corresponds to the organization's value chain, refocussing the relations between data and business processes.

Design Solution

- Accumulative transactions table: provides multi-stages time tracking
- Each record represents a single line item on the shipment invoice
- The Fact table has multiple date foreign keys, each assigned to a different process

	Date	Product	Customer	Deal	Sales Rep	Ship From	Shipper
Quotes	X	X	X	X	X		
Orders	X	X	X	X	X		
Shipments	X	X	X	X	X	X	X
Invoicing	X	X	X	X	X	X	X

Example



	Date	Product	Customer	Deal	Sales Rep	Ship From	Shipper
Quotes	X	X	X	X	X		
Orders	X	X	X	X	X		
Shipments	X	X	X	X	X	X	X
Invoicing	X	X	X	X	X	X	X

Summary: How Patterns Compare?

Characteristic	Transaction	Periodic Snapshot	Accumulative transactions
Time period represented?	Point in time	Regular, set intervals	Changing time spans
Grain	One row per transaction	One row per period	One row per lifecycle
Fact table loads	Insert	Insert	Insert and update
Date dimension	Transaction date	End-of-period date	Multiple dates for milestones
Facts	Transaction activity	Performance for predefined time interval	Performance over lifetime

Summary