Data Modeling

Kristo Raun

Data Engineering 2024 Fall

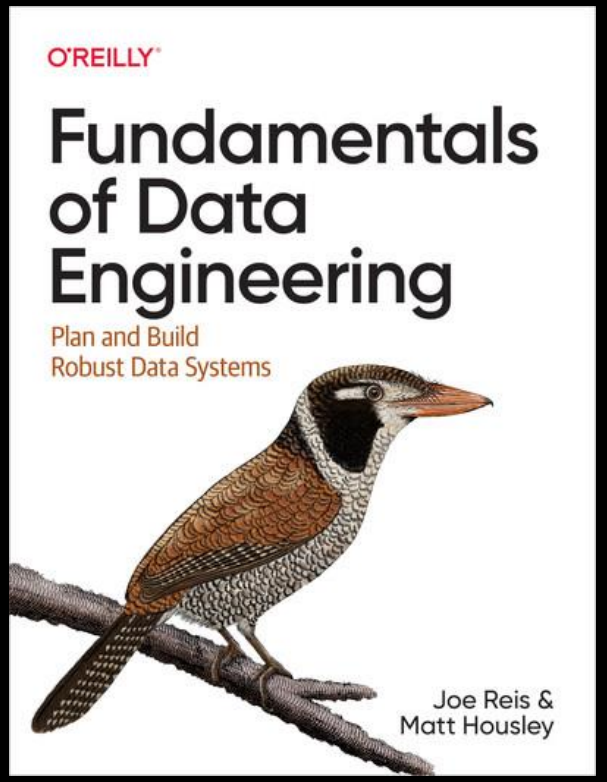
UNIVERSITY OF TARTU

Agenda

- Data architecture
- Data modeling
- Dimensional modeling
- Quiz session

Reading

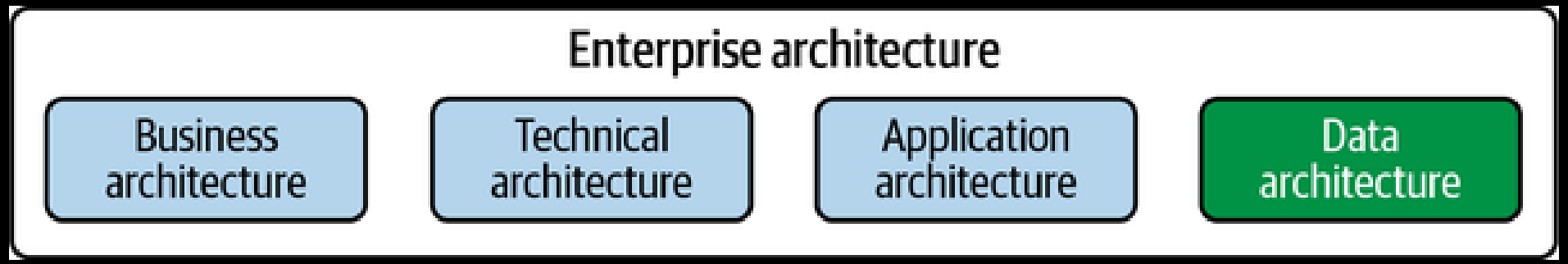
- Chapters III and VIII



Data architecture

Big picture

Technical solutions exist not for their own sake but in support of business goals.



Data architecture

What business processes does the data serve?

How does the organization manage data quality?

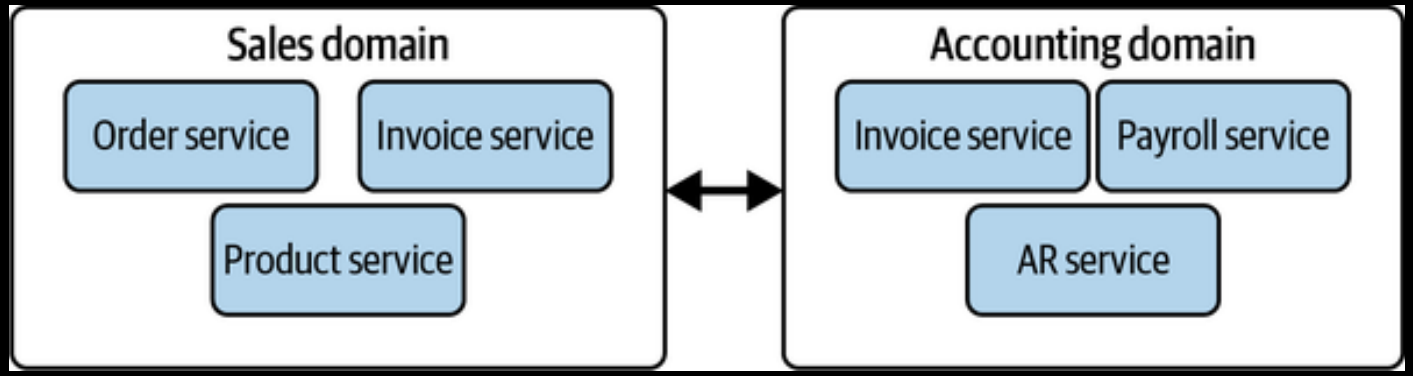
What is the latency requirement from when the data is produced to when it becomes available to query?

How is data ingested, stored, transformed, and served along the data engineering lifecycle?
How will you move 10 TB of data every hour from a source database to your data lake?

Data architecture

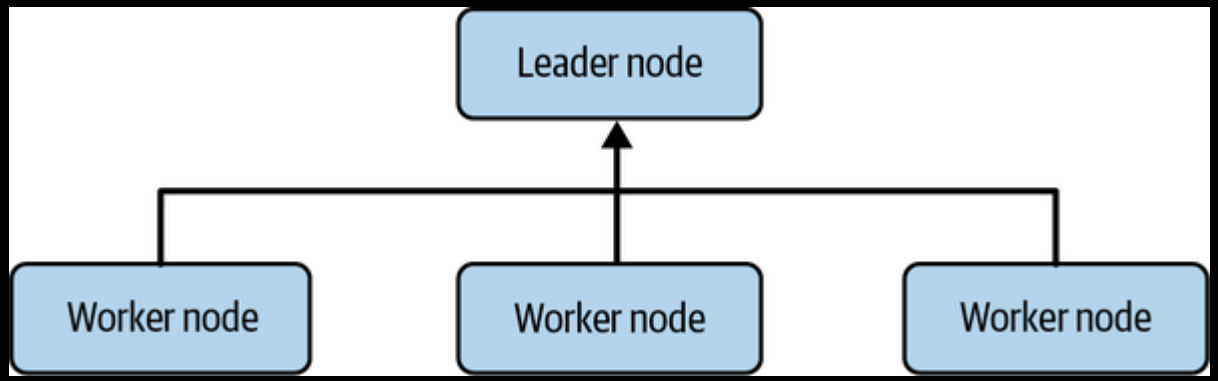
Domain: real world subject area

Service: a set of functionality for accomplishing a task



Data architecture

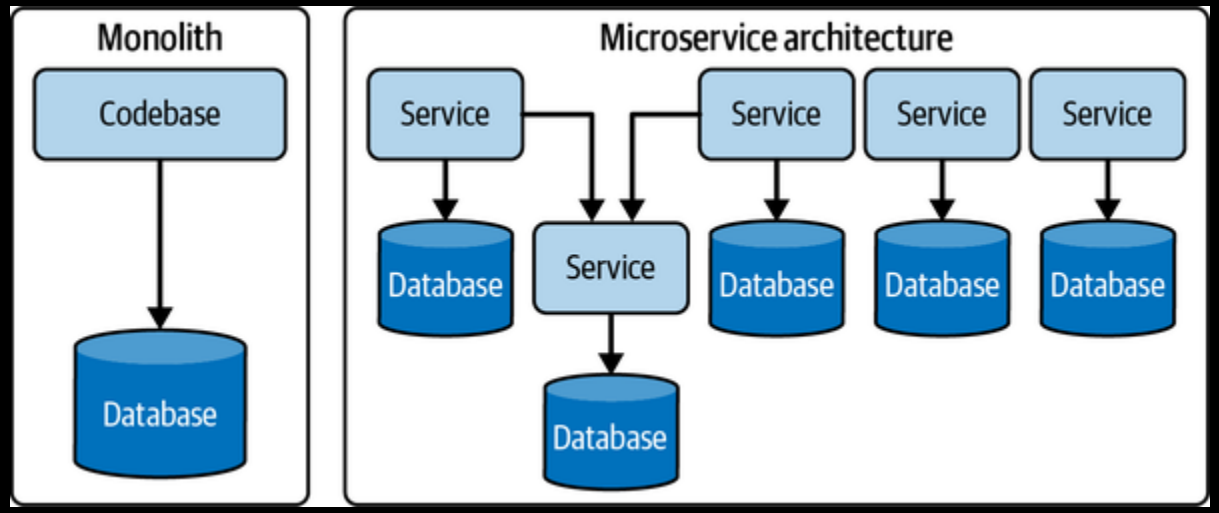
Distributed systems



Partitioning strategy?
Scalability (and elasticity)?

Data architecture

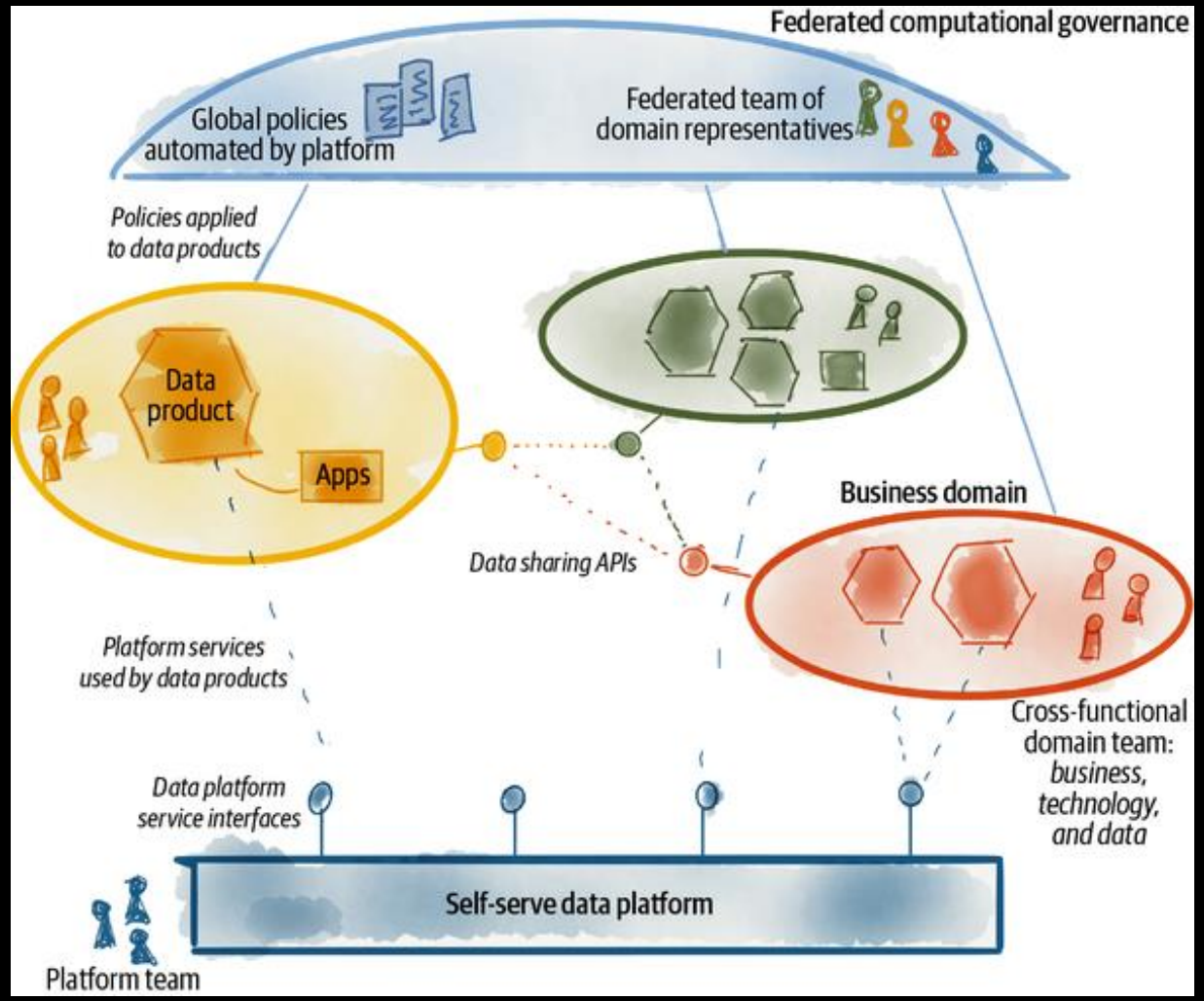
Is a central data warehouse usually a monolith or following the microservices architecture?



Data architecture

Data mesh

In order to decentralize the monolithic data platform, we need to reverse how we think about data, its locality, and ownership. Instead of flowing the data from domains into a centrally owned data lake or platform, ***domains need to host and serve their domain datasets in an easily consumable way****.*



Simplified example of a data mesh architecture. Source: From Data Mesh, by Zhamak Dehghani

Agenda

- ~~Data architecture~~
- Data modeling
- Dimensional modeling
- Quiz session

Data modeling

A **data model** represents the way data relates to the real world.

It reflects how the data must be structured and standardized to best reflect your organization's processes, definitions, workflows, and logic.

A good data model captures how communication and work naturally flow within your organization

Data modeling

Conceptual data model (CDM)	Logical data model (LDM)	Physical data model (PDM)
Includes high-level data constructs	Includes entities (tables), attributes (columns/fields) and relationships (keys)	Includes tables, columns, keys, data types, validation rules, database triggers, stored procedures, domains, and access constraints
Non-technical names, so that executives and managers at all levels can understand the data basis of Architectural Description	Uses business names for entities & attributes	Uses more defined and less generic specific names for tables and columns, such as abbreviated column names, limited by the database management system (DBMS) and any company defined standards
Uses general high-level data constructs from which Architectural Descriptions are created in non-technical terms	Is independent of technology (platform, DBMS)	Includes primary keys and indices for fast data access.

https://en.wikipedia.org/wiki/Logical_schema

Data modeling

Normal forms

Denormalization

- Storage is cheap
- Faster read

Constraint (informal description in parentheses)	UNF (1970)	1NF (1970)	2NF (1971)	3NF (1971)	EKNF (1982)	BCNF (1974)	4NF (1977)	ETNF (2012)	5NF (1979)	DKNF (1981)	6NF (2003)
Unique rows (no duplicate records)[4]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Scalar columns (columns cannot contain relations or composite values)[5]	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Every non-prime attribute has a full functional dependency on each candidate key (attributes depend on the *whole* of every key)[5]	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
Every non-trivial functional dependency either begins with a superkey or ends with a prime attribute (attributes depend *only* on candidate keys)[5]	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
Every non-trivial functional dependency either begins with a superkey or ends with an elementary prime attribute (a stricter form of 3NF)	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	—
Every non-trivial functional dependency begins with a superkey (a stricter form of 3NF)	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	—
Every non-trivial multivalued dependency begins with a superkey	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	—
Every join dependency has a superkey component[8]	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	—
Every join dependency has only superkey components	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	—
Every constraint is a consequence of domain constraints and key constraints	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Every join dependency is trivial	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

https://en.wikipedia.org/wiki/Database_normalization

Data modeling

- Bill Inmon
  - Top-down approach
  - Corporate Information Factory (CIF)
  - Enterprise Data Warehouse (EDW)
- Ralph Kimball
  - Bottom-up approach
  - Dimensional modeling
  - Star schema (snowflake schema)
  - Facts and dimensions
- Data Vault (Dan Linstedt)
  - Hub-and-Spoke
  - EDW 2.0
- One Big Table (OBT)
  - Wide tables

What is dimensional modeling?

- Dimensional modeling is widely accepted as the preferred technique for presenting analytic data because it addresses two simultaneous requirements:
  - Deliver data that's understandable to the business users.
  - Deliver fast query performance.
- Dimensional modeling is a longstanding technique for making databases simple.

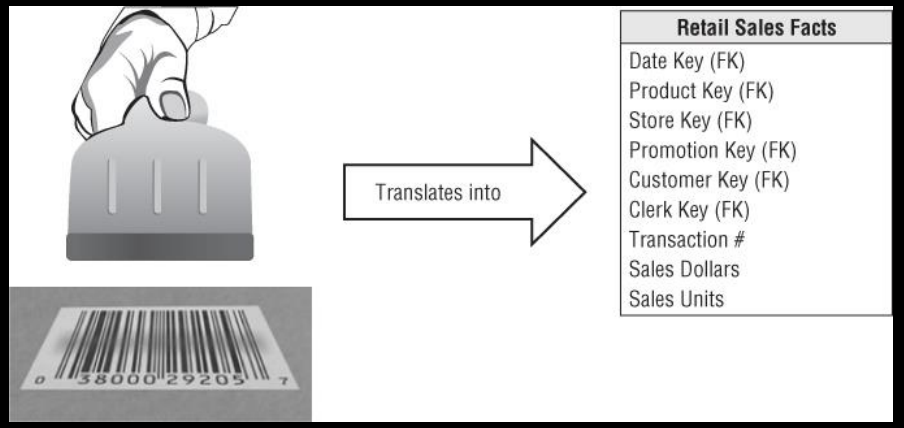
◈ Source: Ross, Margy, and Ralph Kimball. The data warehouse toolkit: the definitive guide to dimensional modeling. John Wiley & Sons, 2013.

Main flow of dimensional modeling

- Start from **business requirements**
  - What needs to be done? Why?
- Design facts and dimensions

Facts and dimensions

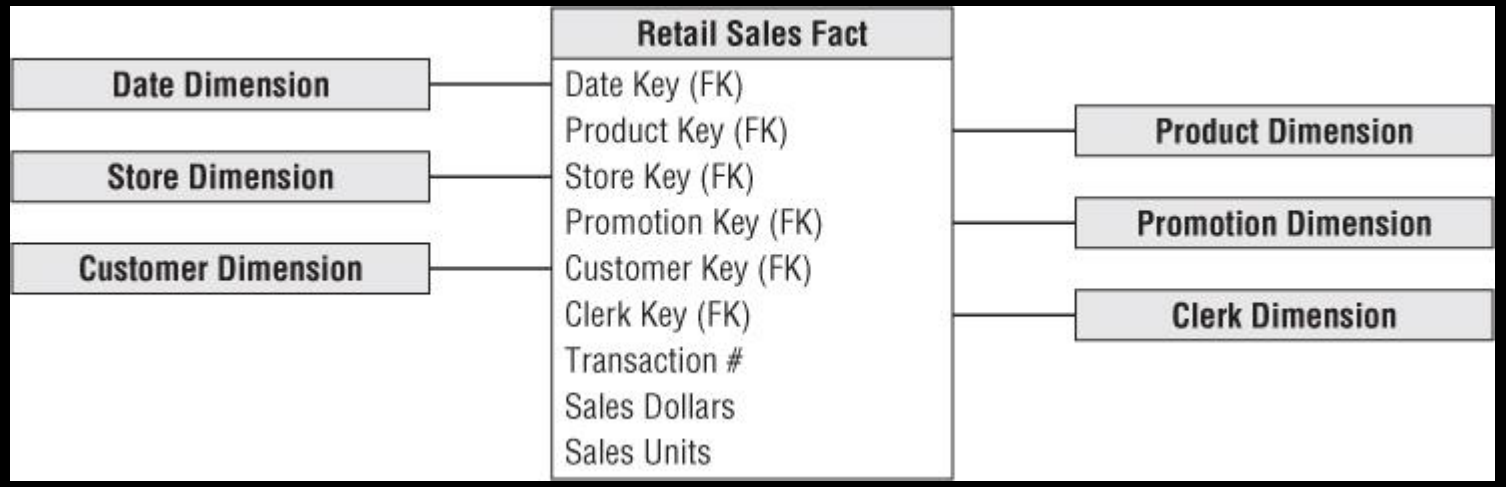
- Fact tables are for measurements
- Dimension tables are for descriptive context



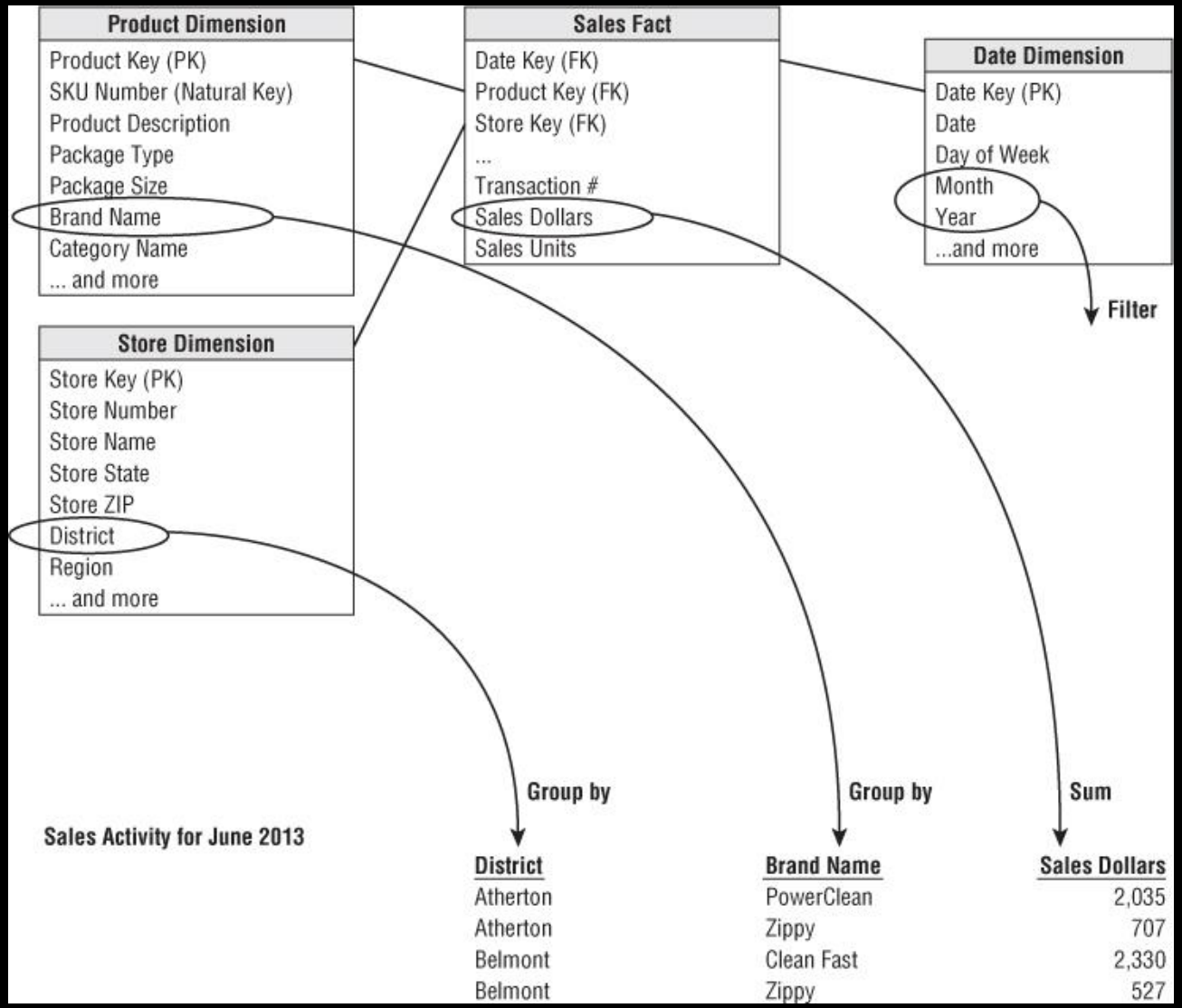
Product Dimension
Product Key (PK)
SKU Number (Natural Key)
Product Description
Brand Name
Category Name
Department Name
Package Type
Package Size
Abrasive Indicator
Weight
Weight Unit of Measure
Storage Type
Shelf Life Type
Shelf Width
Shelf Height
Shelf Depth
...

Facts and dimensions

- Fact tables are for measurements
- Dimension tables are for descriptive context



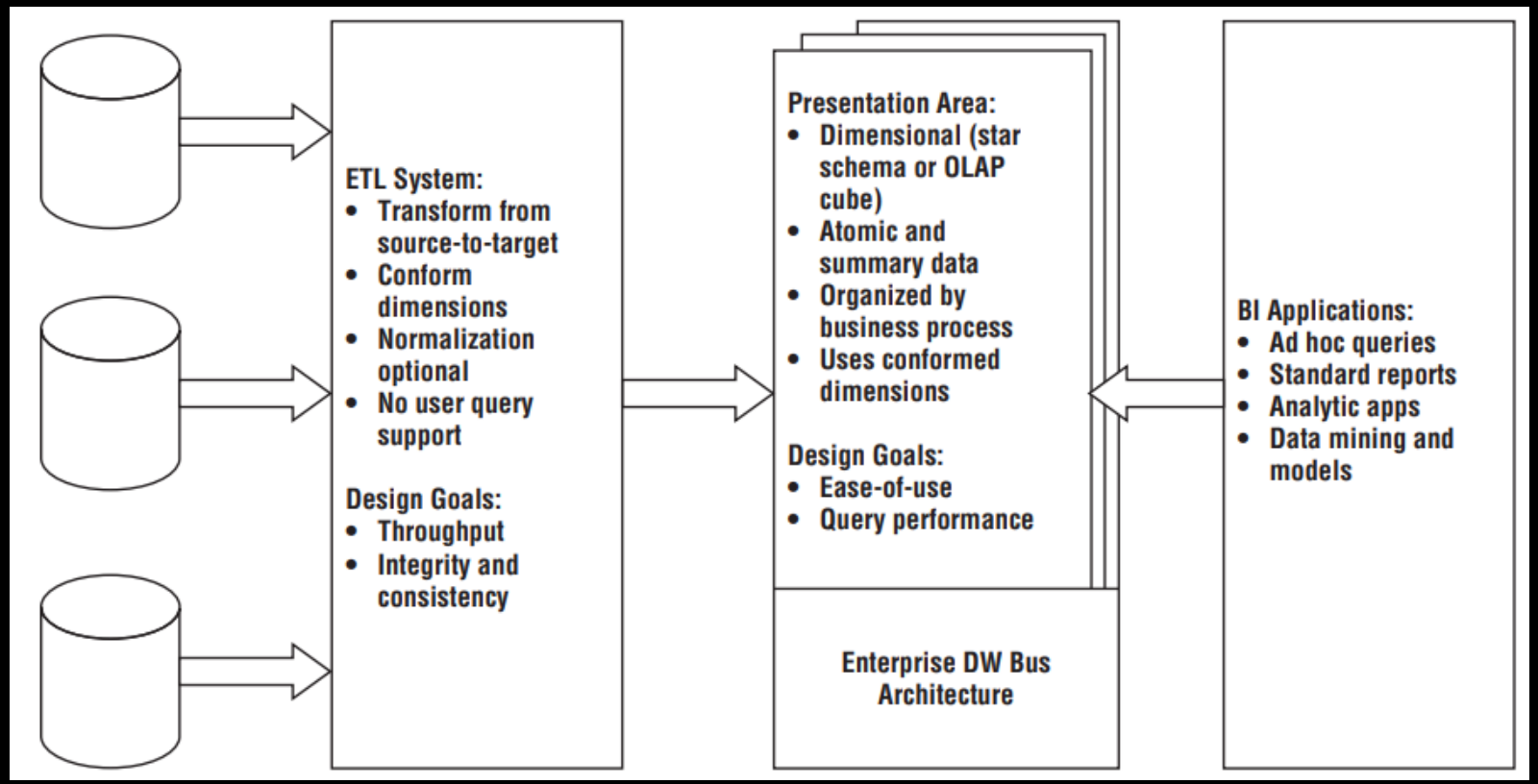
Facts and dimensions




```
SELECT
    store.district_name,
    product.brand,
    sum(sales_facts.sales_dollars) AS "Sales Dollars"
FROM
    store,
    product,
    date,
    sales_facts
WHERE
    date.month_name="January" AND
    date.year=2013 AND
    store.store_key = sales_facts.store_key AND
    product.product_key = sales_facts.product_key AND
    date.date_key = sales_facts.date_key
GROUP BY
    store.district_name,
    product.brand
```

General architecture

Sources Staging area Presentation layer BI/ML



- Source: Ross, Margy, and Ralph Kimball. The data warehouse toolkit: the definitive guide to dimensional modeling. John Wiley & Sons, 2013.

Further reading

- Chapter III
  - Principles of designing good data architecture
  - Lambda, Kappa architectures
  - IoT
- Chapter VIII
  - Modeling streaming data
- https://www.youtube.com/watch?v=IdCmMkQLvGA (Kahan Data Solutions)
- The Data Warehouse Toolkit, 3rd edition
  - https://learning.oreilly.com/library/view/the-data-warehouse/9781118530801/