Data Warehousing

Foundations of Analytical Processing

Database Systems

The Retail Analogy

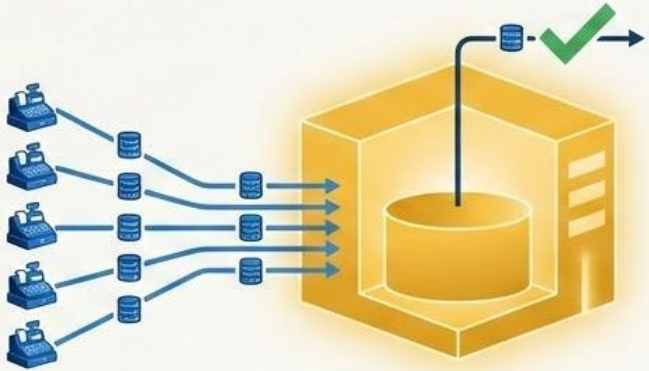
“Imagine a retail company with 1000 stores. Each has its own cash register (operational database).”

The Challenge

To answer “Which products sell best in winter across the Northeast region?”, you can’t query 1000 separate registers.

The Solution

You need a central warehouse that consolidates all register data.



Learning Objectives

Understand the purpose and architecture of data warehouses.

Differentiate between data warehouses, data marts, and data lakes.

Master multidimensional data modeling (cubes).

Design star and snowflake schemas.

Understand OLAP operations for data analysis.

The Need for Warehousing

Operational (OLTP)

Goal: Optimize for INSERT, UPDATE, DELETE.

Example: Recording a sale.

Analytical (OLAP)

Goal: Optimize for SELECT, AGGREGATE.

Example: Sales trend over 5 years.

- Complex queries, large scans.
- Denormalized for speed.
- Fewer users (analysts).

Bill Inmon's Definition

> A data warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision-making process.

Subject-Oriented

Organized by Customer, Product, Sales (not apps).

Integrated

Unified naming and formatting from multiple sources.

Time-Variant

Historical data with explicit time dimensions.

Non-Volatile

Read-only. Data is loaded, not updated.

Architecture & ETL Process

Example: University Data Warehouse

- **Sources:** Student Info System, Library, Finance.
- **Query:** “Retention rate by department over 5 years.”

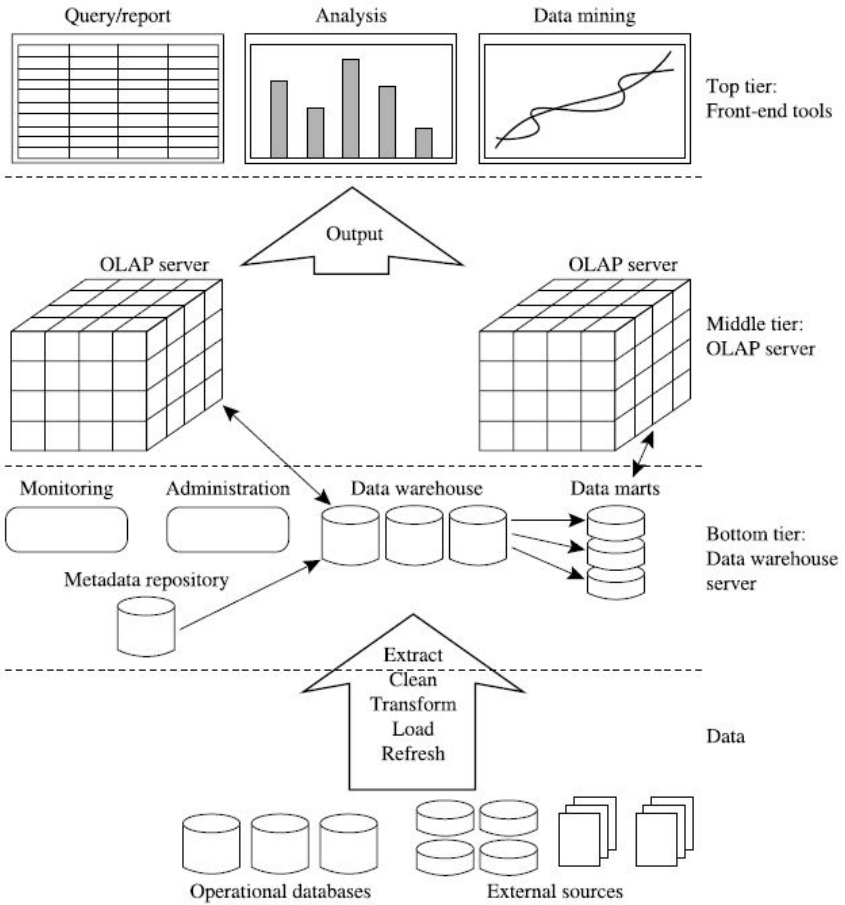


FIGURE 3.1

A three-tier data warehousing architecture.

EDW vs. Data Marts

Enterprise DW

Scope:
Entire Organization

Data Sources:
Many

Cost:
High

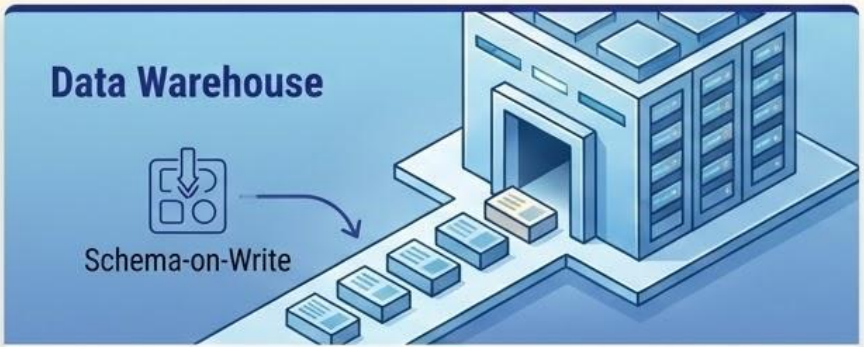
Data Mart

Scope:
Departmental

Data Sources:
Few

Cost:
Lower

The Modern Evolution: Data Lakes



Structured data.

Business Analysts.

Reporting/ Dashboards.

Raw, unstructured (JSON, Images).

Data Scientists/ Engineers.

Machine Learning/Exploration.

Lakehouse

Best of both. Data Lake for raw storage + Data Warehouse for curated analytics.

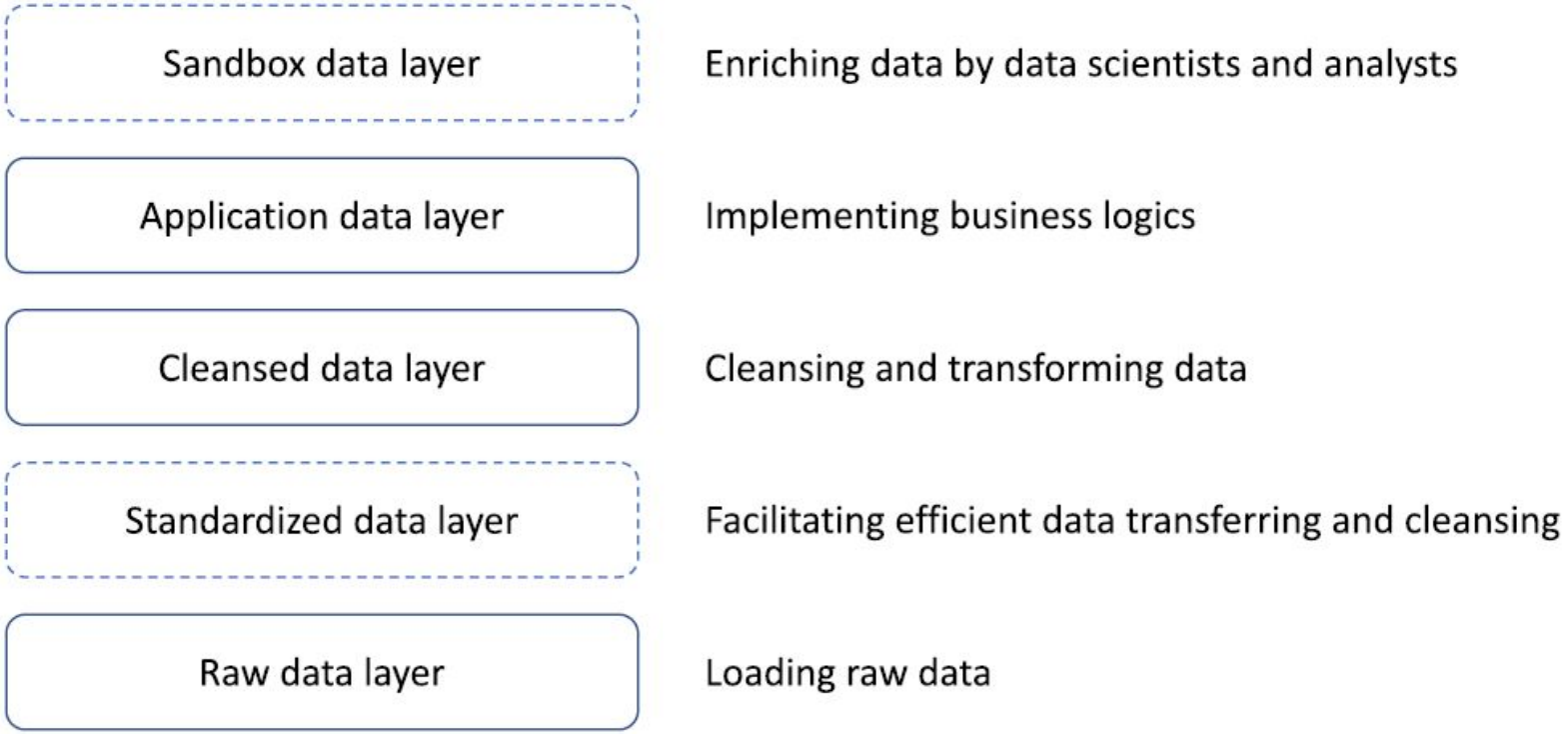


FIGURE 3.2

The layers of data storage in data lakes.

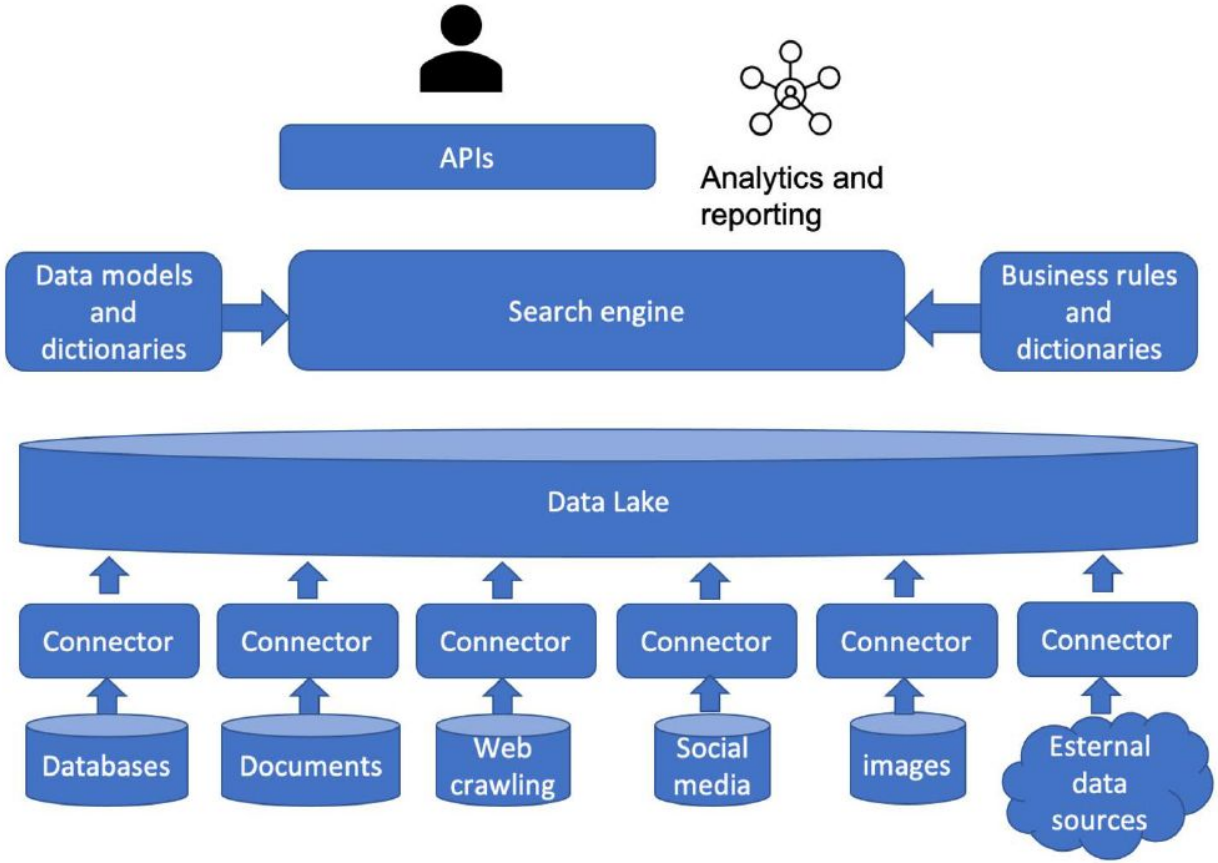


FIGURE 3.3

The conceptual architecture of data lakes.

The Data Cube

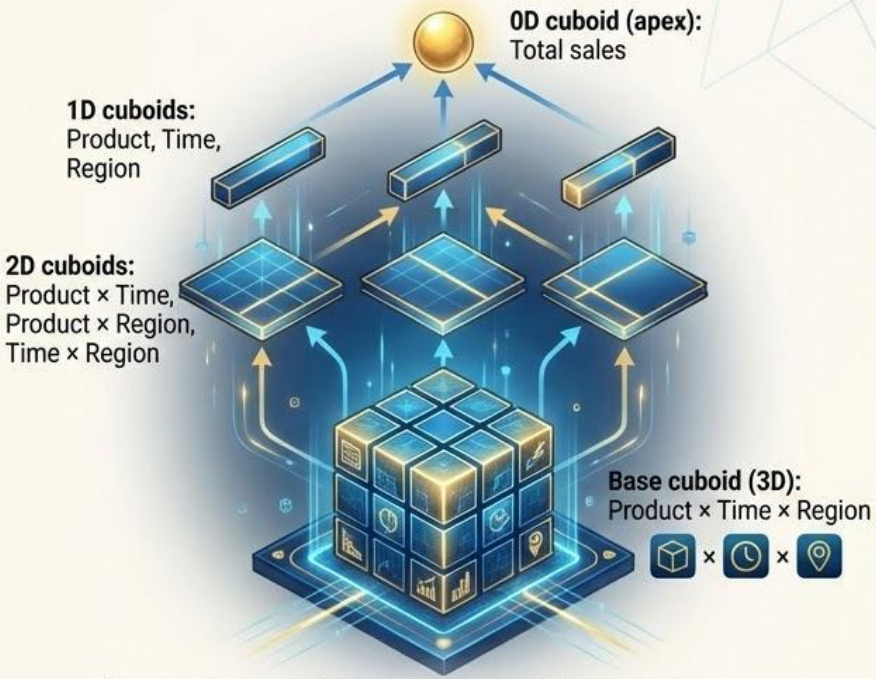
Components

Dimensions: Perspectives (Time, Product, Region).

Measures: Numerical values (Sales, Profit).

Cells: Intersection of dimensions.

Lattice of Cuboids (Power of Multidimensionality)



- For *n* dimensions: 2*n* possible cuboids

Table 3.1 2-D view of sales data according to *time* and *item*.

	location = "Vancouver"			
time (quarter)	**item (type)**			
	home entertainment	**computer**	**phone**	**security**
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

Note: The sales are from branches located in the city of Vancouver. The measure displayed is dollars_sold *(in thousands).*

Table 3.2 3-D view of sales data according to *time*, *item*, and *location*.

	location = "Chicago"				location = "New York"				location = "Toronto"				location = "Vancouver"			
time	item				item				item				item			
	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

Note: The measure displayed is dollars_sold *(in thousands).*

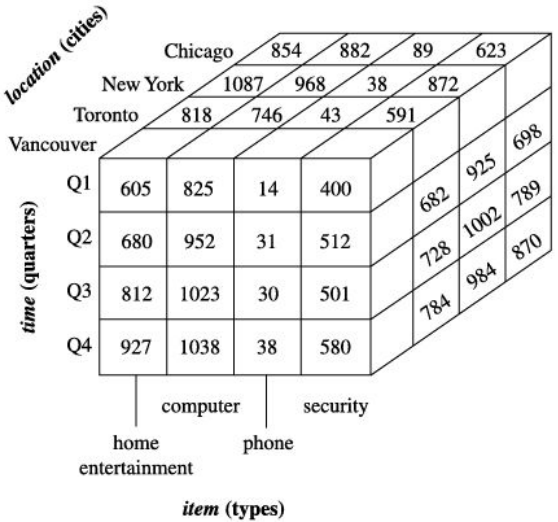


FIGURE 3.4

A 3-D data cube representation of the data in Table 3.2, according to *time*, *item*, and *location*. The measure displayed is *dollars_sold* (in thousands).

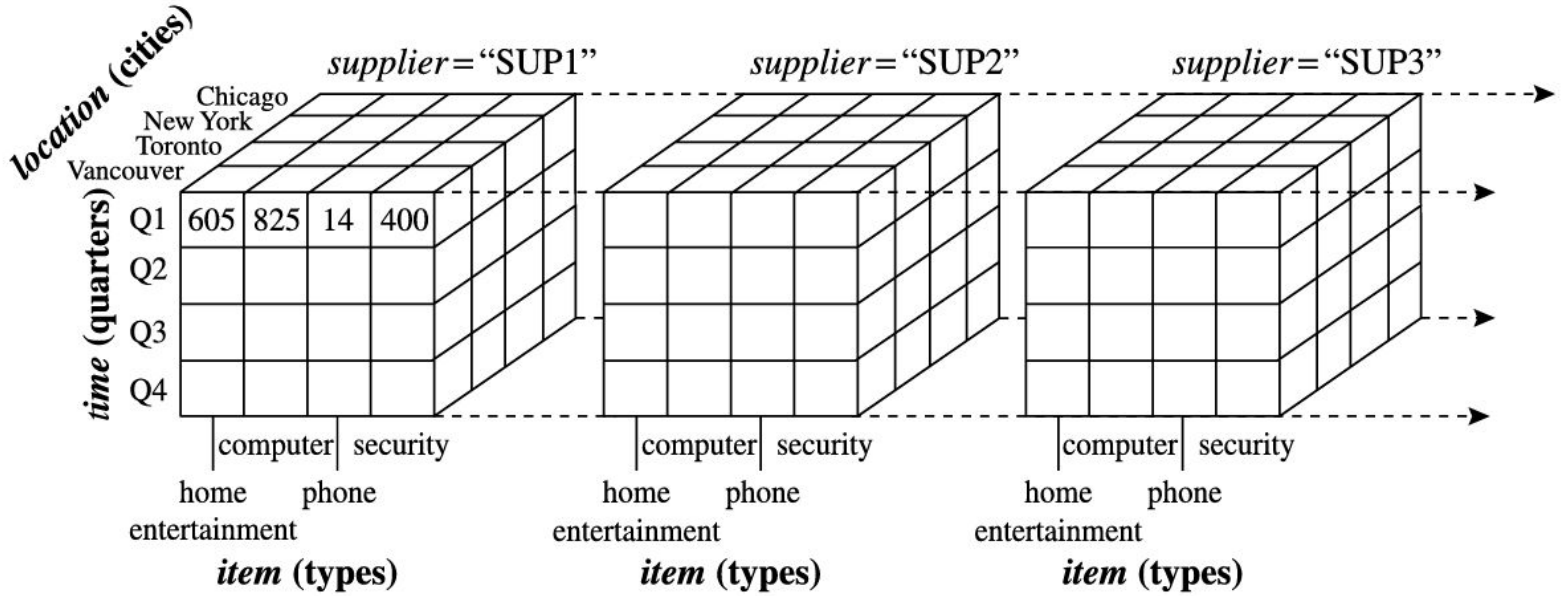


FIGURE 3.5

A 4-D data cube representation of sales data, according to *time*, *item*, *location*, and *supplier*. The measure displayed is *dollars_sold* (in thousands). For improved readability, only some of the cube values are shown.

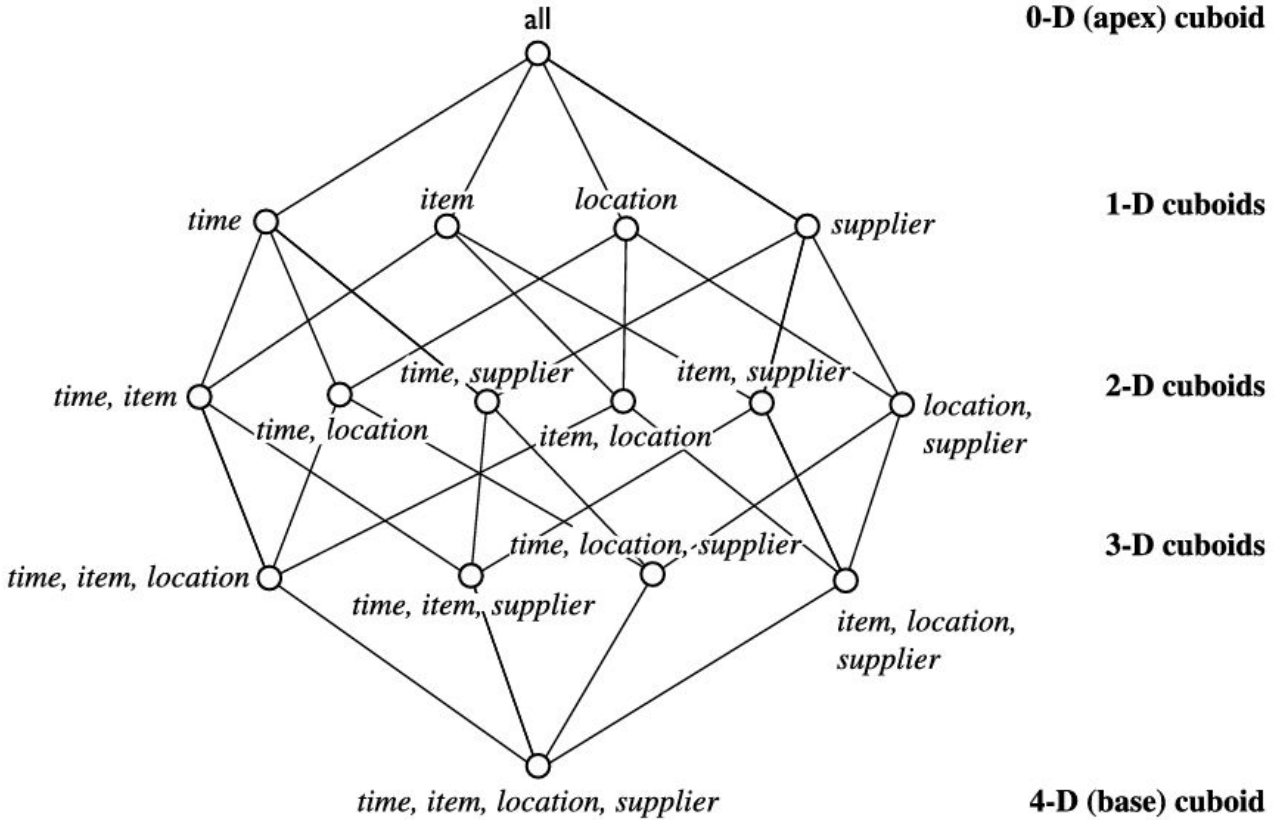


FIGURE 3.6

Lattice of cuboids, making up a 4-D data cube for *time*, *item*, *location*, and *supplier*. Each cuboid represents a different degree of summarization.

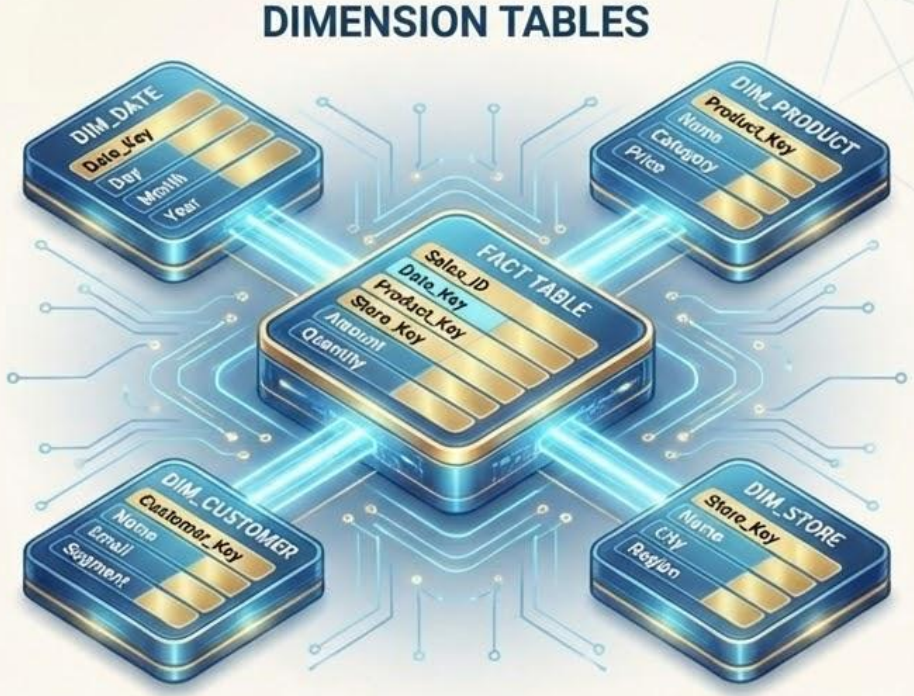
Star Schema

Architecture

The simplest and most common schema.

- **Fact Table (Center):** Contains keys and measures. Denormalized.
- **Dimension Tables (Points):** Contain descriptive attributes.

Pros: Fast performance, easy to understand.



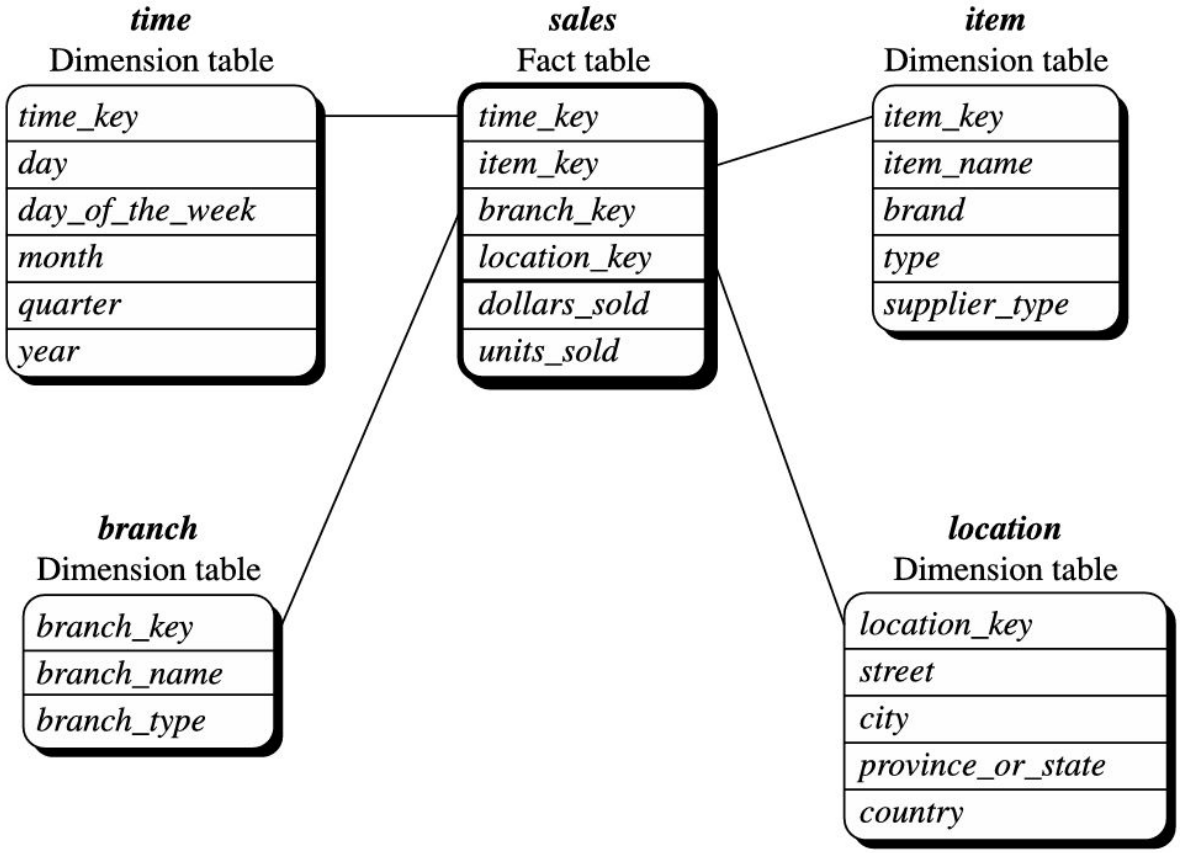


FIGURE 3.7

Star schema of *sales* data warehouse.

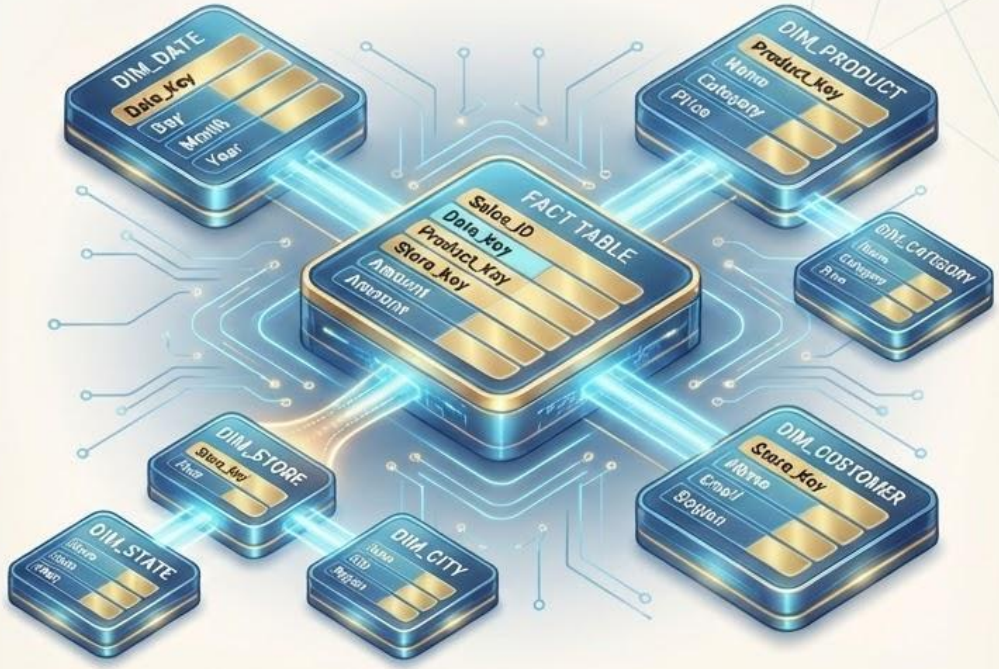
Snowflake Schema

Architecture

An extension of the Star Schema.

- **Normalized Dimensions:** Dimension tables are split into further tables.
- Example: Dim_Store splits into Dim_City, Dim_State.

⚠**Cons: More joins = Slower queries.**



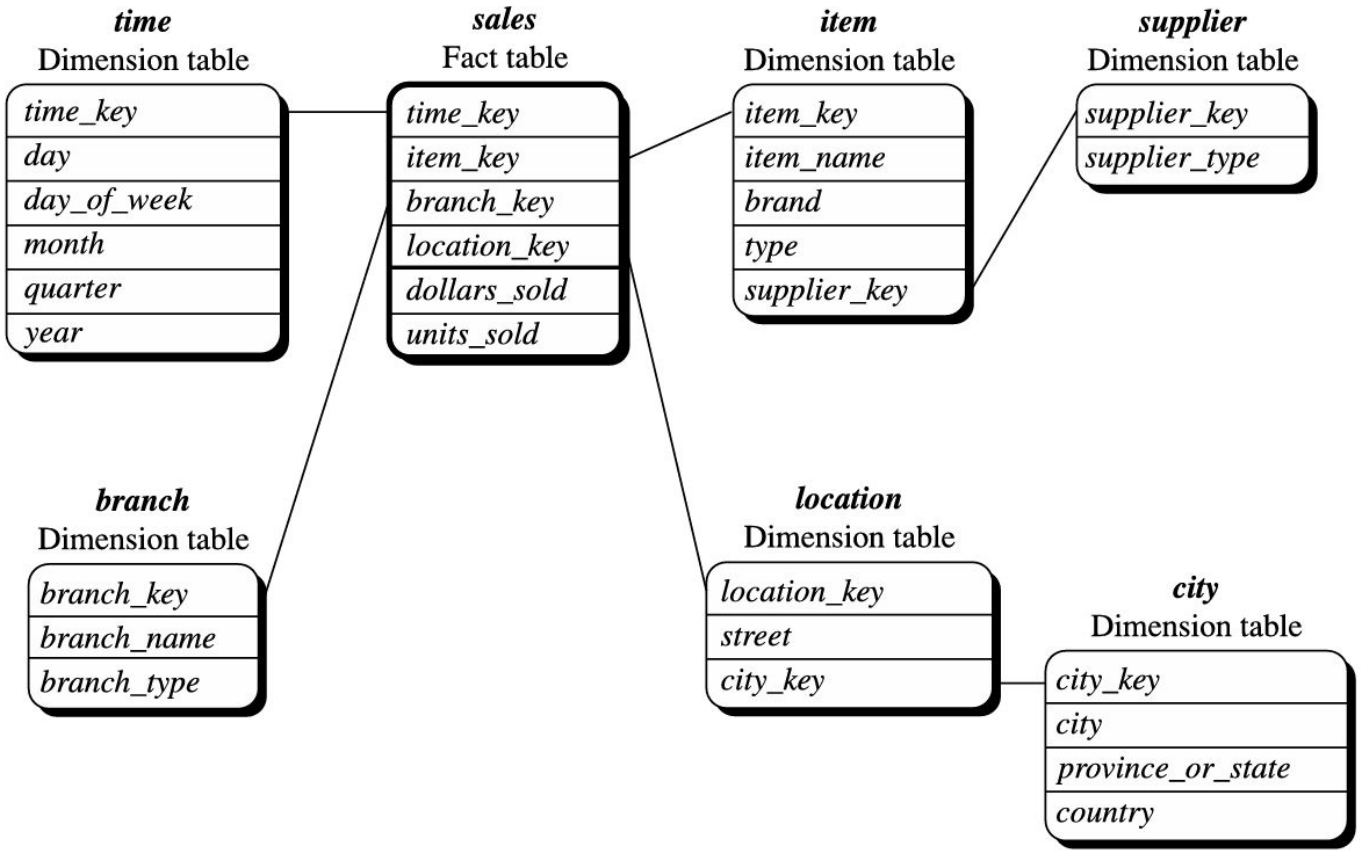


FIGURE 3.8

Snowflake schema of a *sales* data warehouse.

Fact Constellation (Galaxy Schema)

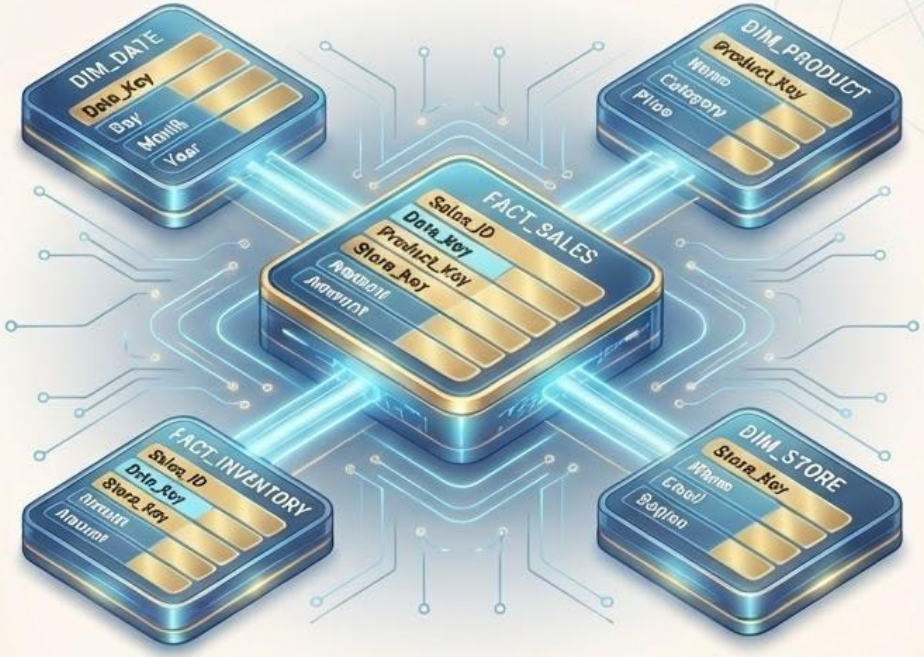
Multiple Fact Tables

Complex schema where multiple fact tables share dimension tables.

Real Example: Retail Chain

- **Fact_Sales**: Daily transactions
- **Fact_Inventory**: Daily stock levels
- **Fact_Returns**: Product returns

Shared Dimensions: Product, Date, Store



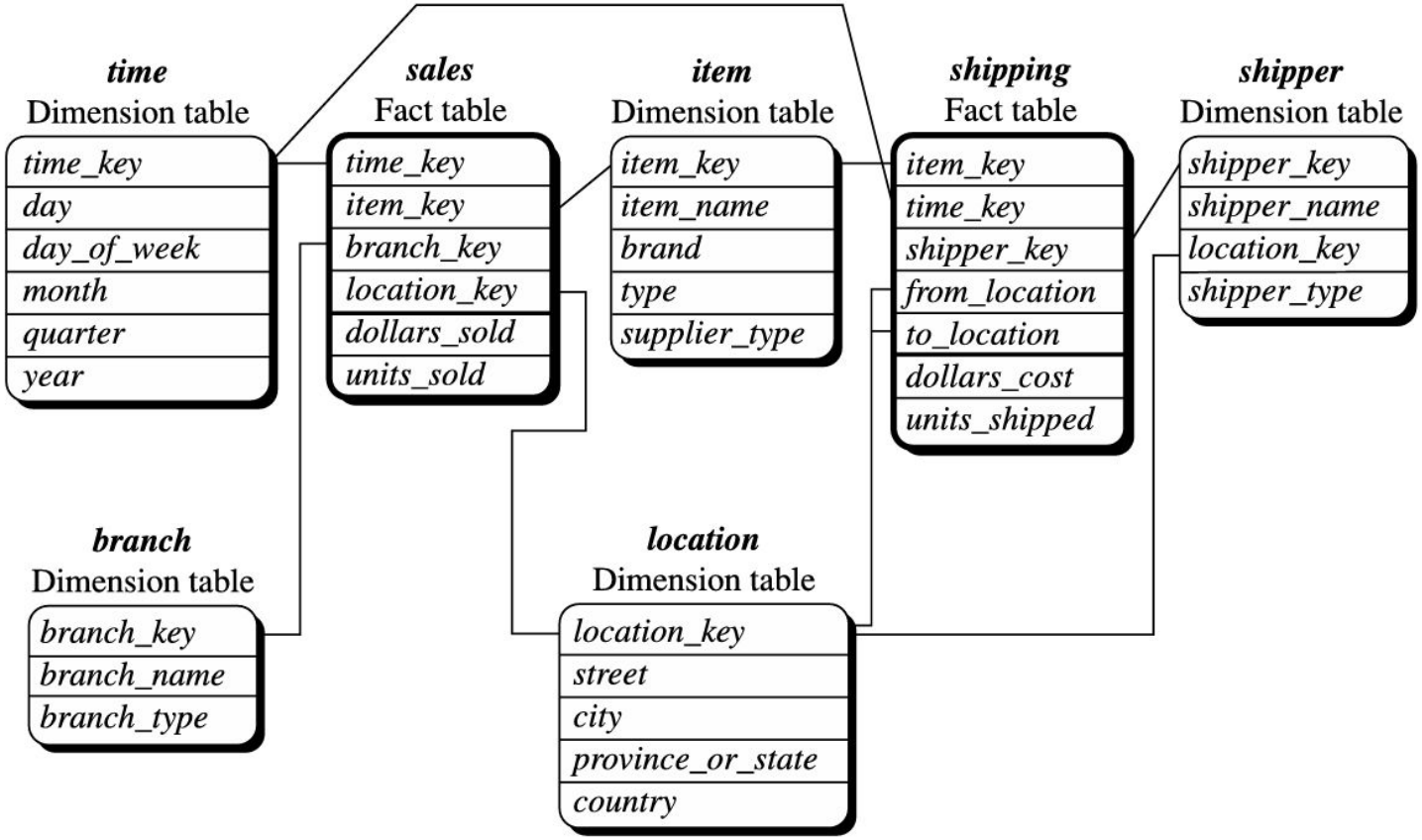


FIGURE 3.9

Fact constellation schema of a sales and shipping data warehouse.

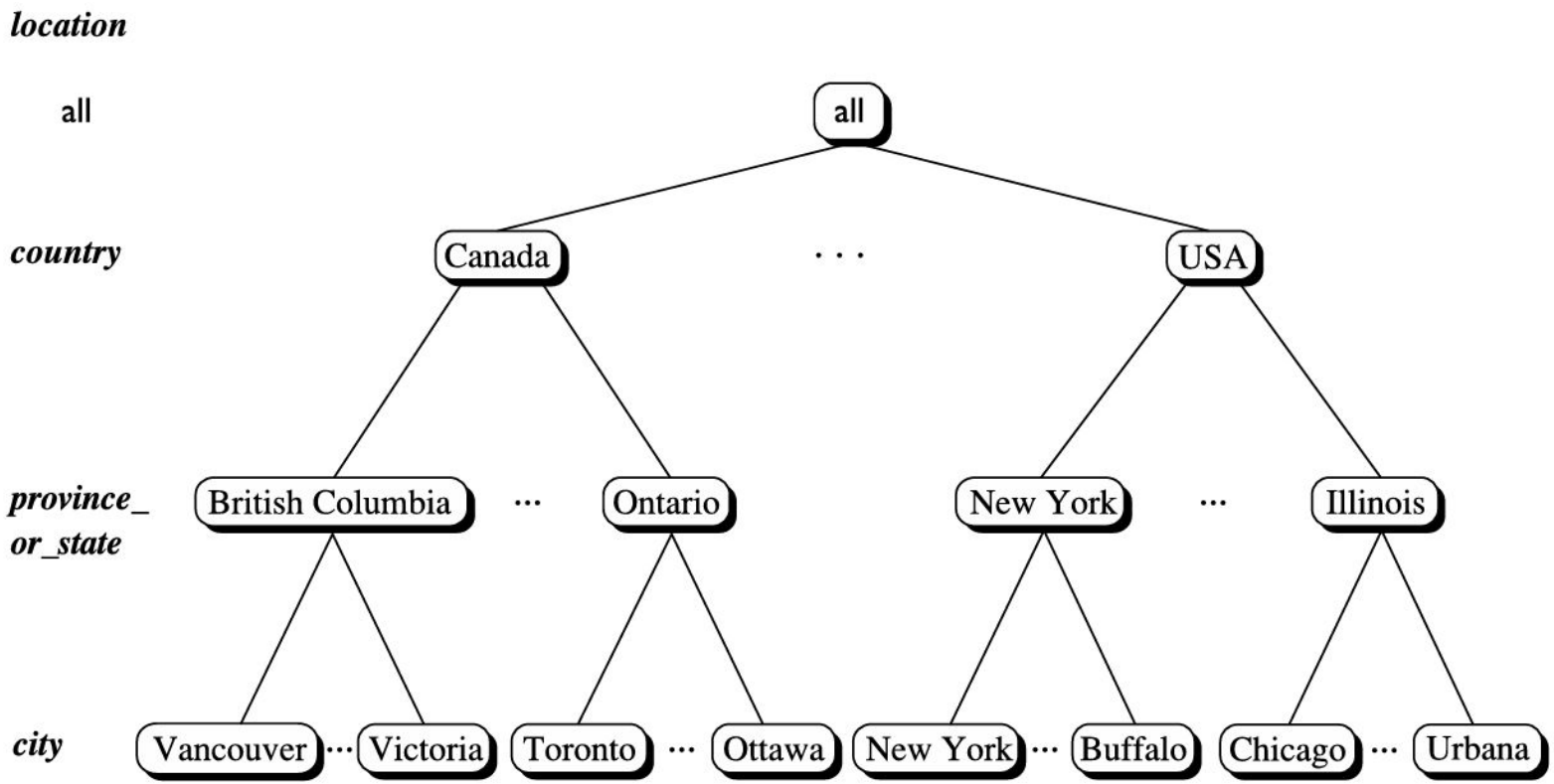


FIGURE 3.10

A concept hierarchy for *location*. Due to space limitations, not all of the hierarchy nodes are shown, indicated by ellipses between nodes.

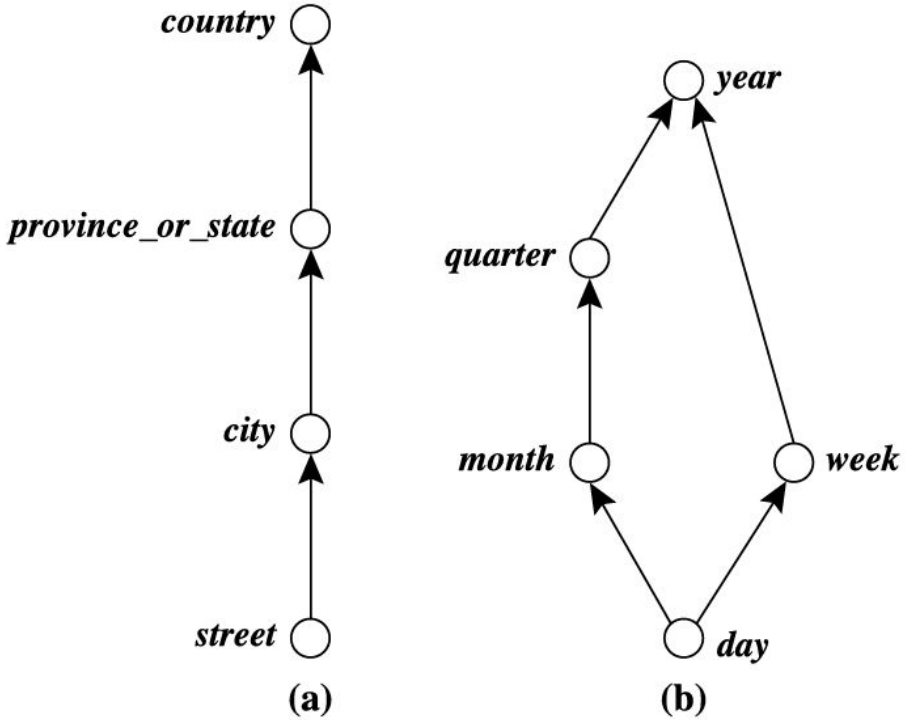


FIGURE 3.11

Hierarchical and lattice structures of attributes in warehouse dimensions: (a) a hierarchy for *location* and (b) a lattice for *time*.

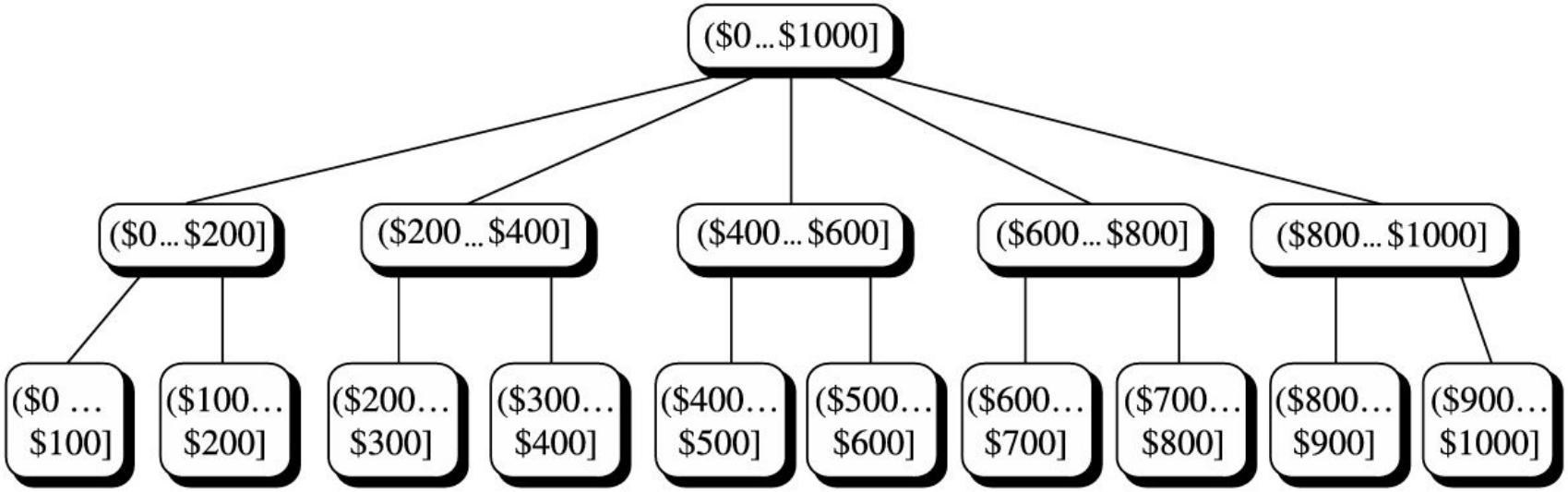
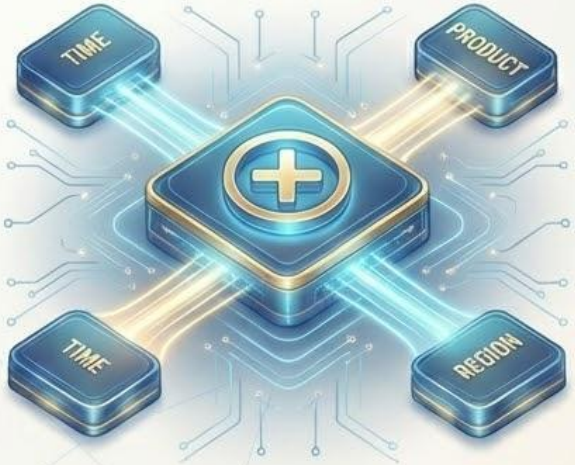


FIGURE 3.12

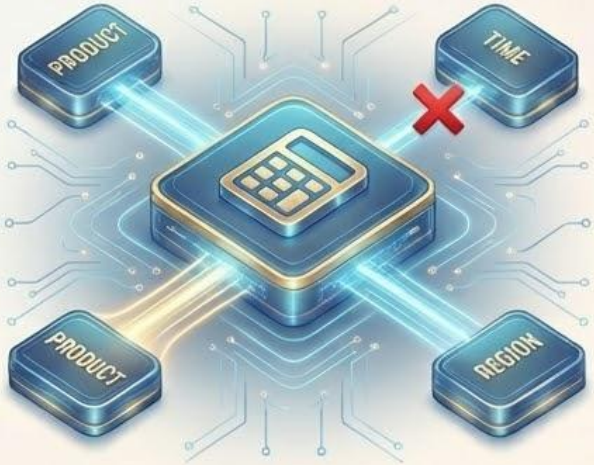
A concept hierarchy for *price*.

Types of Measures



Additive

Can be summed across ALL dimensions.
Ex: Sales Amount, Quantity.



Semi-Additive

Summable across SOME dimensions (not Time).
Ex: Account Balance (Cannot sum balances across days).

Non-Additive

Cannot be summed. Need Avg, Min, Max.
Ex: Unit Price, Profit Margin, Temperature.

OLAP Operations

Roll-up (Drill-up)
Summarize slonth
Summarize data
(Day -> Month).

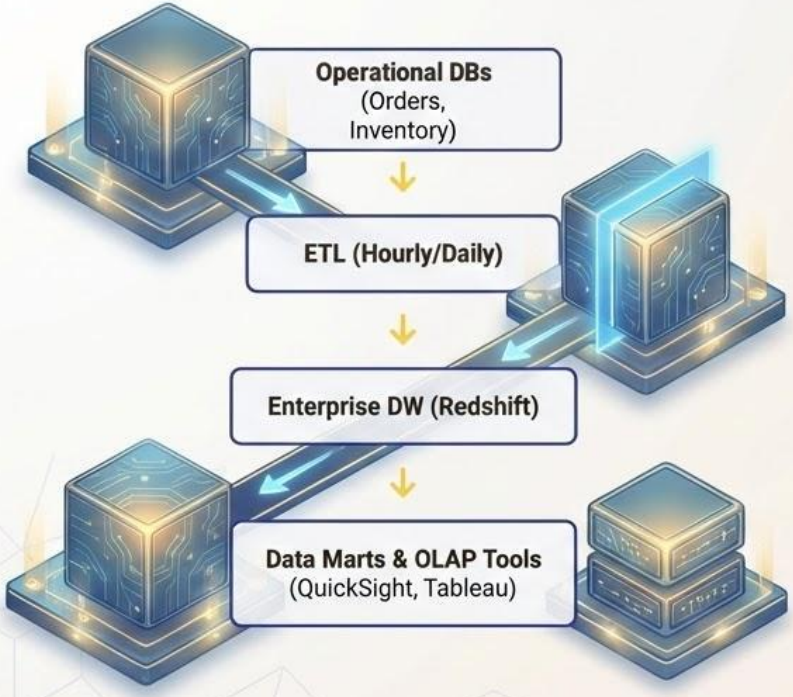
Drill-down
Size > data
Go into detail
(Country -> City).

Slice
Select sectice
Select one dimension
member (Region="East").

Dice
Select sub-cube
Select sub-cube
(Product="PC", Time="Q1")

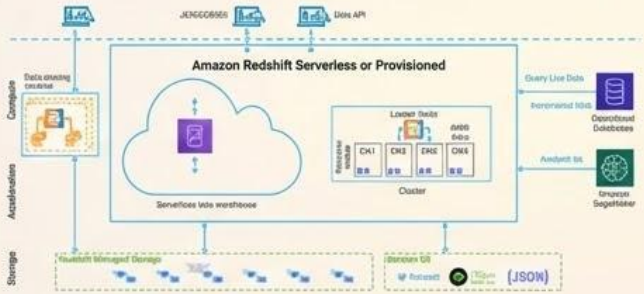
Pivot
Rotate axes for
Rotate axes for
different perspective.

Case Study: Amazon Data Warehouse



Sample Queries Enabled

- "Top 10 products by sales growth QoQ"
- "Customer lifetime value by acquisition channel"
- "Inventory turnover by warehouse"
- "Regional demand forecasting"



Summary

Decision Framework

Data Warehouse: For structured business reporting.

Data Lake: For raw, unstructured exploration.

Star Schema: For simplicity and speed.

Snowflake Schema: For storage optimization.

Pitfalls to Avoid

Ignoring data quality (Garbage in = Garbage out).

Over-normalizing schemas.

Underestimating ETL complexity (70% of effort).