A hand in a black glove holds a hammer, poised to strike a baseball. The baseball is covered in binary code (0s and 1s). The background is a blurred green field.

Chapter 3
Introduction to
OLTP and OLAP

**Content of this presentation has been
taken from Book**

**“Fundamentals of Business
Analytics”**

RN Prasad and Seema Acharya

Published by Wiley India Pvt. Ltd.

**and it will always be the copyright of the
authors of the book and publisher only.**

OLTP Understanding

❑ Online Transaction Processing

❑ Consider a point-of-sale (POS) system in a supermarket store. You have picked a bar of chocolate and await your chance in the queue for getting it billed. The cashier scans the chocolate bar's bar code. Consequent to the scanning of the bar code, some activities take place in the background —

- ✓ the database is accessed;
- ✓ the price and product information is retrieved and displayed on the computer screen;
- ✓ the cashier feeds in the quantity purchased;
- ✓ the application then computes the total, generates the bill, and prints it. You pay the cash and leave.

❑ The application has just added a record of your purchase in its database. This was an On-Line Transaction Processing (OLTP) system designed to support on-line transactions and query processing.

❑ In other words, the POS of the supermarket store was an OLTP system.

OLTP Understanding

❑ OLTP systems refer to a class of systems that manage transaction-oriented applications.

❑ These applications are mainly concerned with the entry, storage, and retrieval of data.

❑ They are designed to cover most of the day-to-day operations of an organization such as purchasing, inventory, manufacturing, payroll, accounting, etc.

❑ OLTP systems are characterized by a large number of short on-line transactions such as INSERT (a record of final purchase by a customer was added to the database), UPDATE (the price of a product has been raised from Rs10 to Rs10.5), and DELETE (a product has gone out of demand and therefore the store removes it from the shelf as well as from its database).

❑ Almost all industries today (including airlines, mail-order, supermarkets, banking, etc.) use OLTP systems to record transactional data. The data captured by OLTP systems is usually stored in commercial relational databases. For example, the database of a supermarket store consists of the following tables to store the data about its transactions, products, employees, inventory supplies, Like Transactions, ProductMaster, EmployeeDetails, InventorySupplies, Suppliers, etc.

Online Transactional Processing (OLTP)

- Traditional database application is focused on *Online Transactional Processing (OLTP)*,
 - Short, simple queries and frequent updates involving a relatively small number of tuples e.g., recording sales at cash-registers, selling airline tickets.

OLTP

(ONLINE TRANSACTION PROCESSING SYSTEM)

- Used for transaction oriented applications
- Used by lower level employee
- Quick updates and retrievals
- Many users accessing the same data
- Users are not technical persons
- Response rate is very fast
- Single transaction (one application) at a time

OLTP

(ONLINE TRANSACTION PROCESSING SYSTEM)

- Stores routine data
- Follows client server model
- Applications
 - Banks
 - Retail stores
 - Airline reservation

OLTP

(ONLINE TRANSACTION PROCESSING SYSTEM)



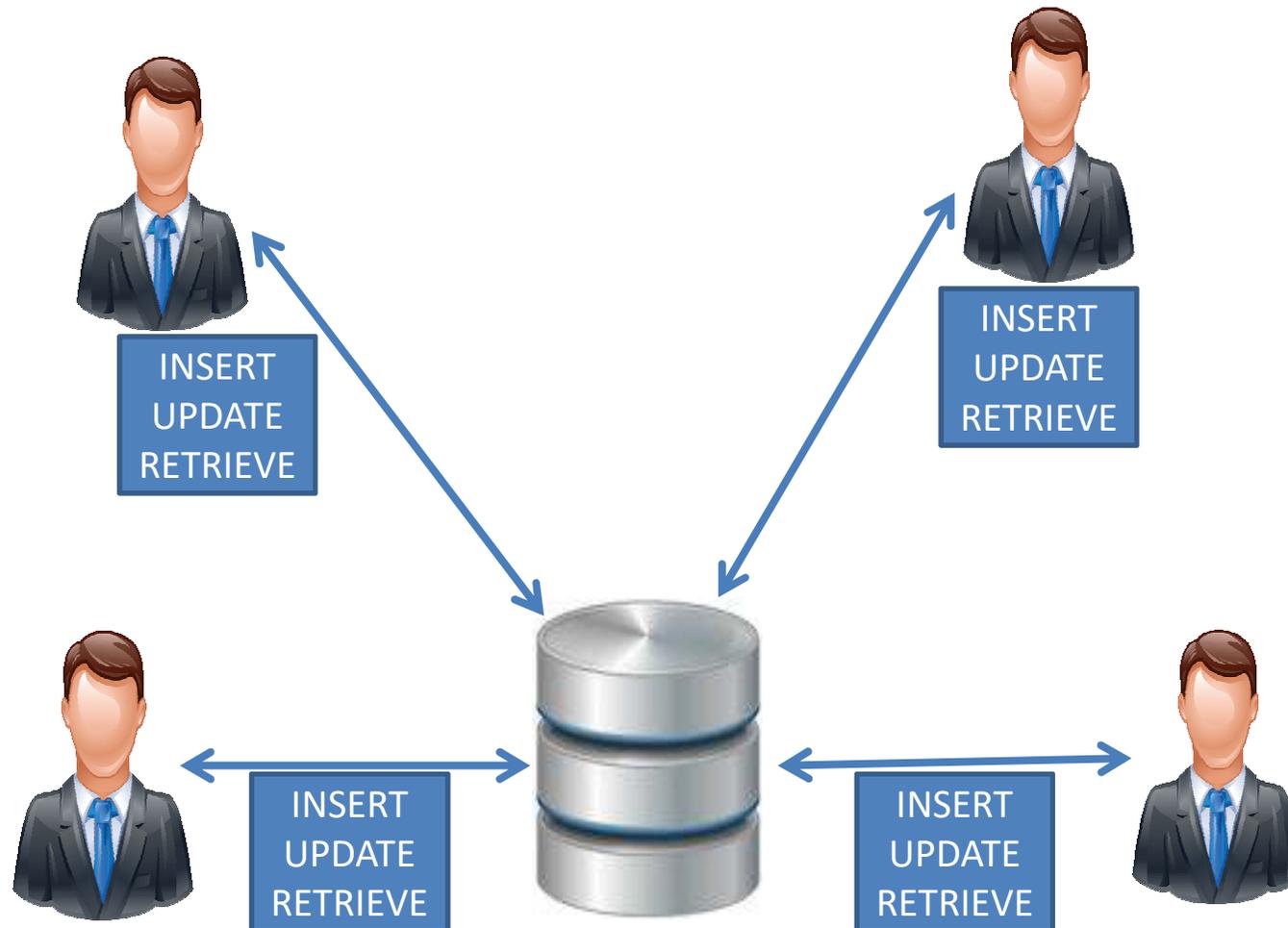
User gets instant update on the account balance after withdrawing the money



TRANSACTIONS

- Single event that changes something
- Different types of transactions
 - Customer orders
 - Receipts
 - Invoices
 - Payments
- Processing of transactions include storage and editing of data
 - When transaction is completed then the records of an organization are changed

TRANSACTIONS



TRANSACTIONS



Cash at register
gone up

Inventory
of video
game gone
down

Ordering of
new video
game for
the store

OLTP Segmentation

- They can be segmented into:
 - Real-time Transaction Processing
 - Batch Processing

Real-time Transaction processing

- Multiple users can fetch the information
- Very fast response rate
- Transactions processed immediately
- Everything is processed in real time

Batch Processing

- Where information is required in batch
- Offline access to information
- Presorting (sequence) is applied
- Takes time to process information



Characteristics of OLTP Model

- Online connectivity
 - LAN,WAN
- Availability
 - Available 24 hours a day
- Response rate
 - Rapid response rate
 - Load balancing by prioritizing the transactions

Characteristics of OLTP Model

- Cost
 - Cost of transactions is less
- Update facility
 - Less lock periods
 - Instant updates
 - Use the full potential of hardware and software

Limitations of Relational Models

- Create and maintain large number of tables for the voluminous data
- For new functionalities, new tables are added
- Unstructured data cannot be stored in relational databases
- Very difficult to manage the data with common denominator (keys)

Answer a Quick Question

According to your understanding,
what are some of the queries that OLTP systems can process?

Queries that an OLTP System can Process

- Search for a particular customer's record.
- Retrieve the product description and unit price of a particular product.
- Filter all products with a unit price equal to or above Rs. 25.
- Filter all products supplied by a particular supplier.
- Search and display the record of a particular supplier.

Advantages and Challenges of an OLTP System

Advantages of an OLTP System

- Simplicity – It is designed typically for use by clerks, cashiers, clients, etc.
- Efficiency – It allows its users to read, write and delete data quickly.
- Fast query processing – It responds to user actions immediately and also supports transaction processing on demand.

Challenges of an OLTP System

- Security – An OLTP system requires concurrency control (locking) and recovery mechanisms (logging).
- OLTP system data content not suitable for decision making – A typical OLTP system manages the current data within an enterprise/organization. This current data is far too detailed to be easily used for decision making.

The Queries that OLTP Cannot Answer

- The super market store is deciding on introducing a new product. The key questions they are debating are: “Which product should they introduce?” and “Should it be specific to a few customer segments?”
- The super market store is looking at offering some discount on their year-end sale. The questions here are: “How much discount should they offer?” and “Should it be different discounts for different customer segments?”
- The supermarket is looking at rewarding its most consistent salesperson. The question here is: “How to zero in on its most consistent salesperson (consistent on several parameters)? All the queries stated above have more to do with analysis than simple reporting”
- Ideally these queries are not meant to be solved by an OLTP system.

OLAP - Online Analytical Processing

- ❖ OLAP differs from traditional databases in the way data is conceptualized and stored.
- ❖ In OLAP data is held in the **dimensional form** rather than the relational form.
- ❖ OLAP's life blood is **multi-dimensional data**.
- ❖ OLAP tools are based on the multi-dimensional data model. The multi-dimensional data model views data in the form of a data **cube**.
- ❖ Online Analytical Processing (OLAP) is a technology that is used to organize large business databases and support business intelligence.
- ❖ OLAP databases are divided into one or more cubes. The cubes are designed in such a way that creating and viewing reports become easy.
- ❖ OLAP databases are divided into one or more cubes, and each cube is organized and designed by a cube administrator to fit the way that you retrieve and analyze data so that it is easier to create and use the PivotTable reports and PivotChart reports that you need.

OLAP (Online Analytical Processing)

- OLAP is a category of software that allows users to analyze information from multiple database systems at the same time. It is a technology that enables analysts to extract and view business data from different points of view
- Analysts frequently need to group, aggregate and join data. These operations in relational databases are resource intensive. With OLAP, data can be pre-calculated and pre-aggregated, making analysis faster.
- Provides multidimensional **view** of data
- Used for analysis of data
- Data can be viewed from different perspectives
- Determine why data appears the way it does
- **Drill down approach** is used to further dig down deep into the data

OLAP - Example

- ❖ Let us consider the data of a supermarket store, “AllGoods” store, for the year “2001”.
- ❖ This data as captured by the OLTP system is under the following column headings: Section, Product-CategoryName, YearQuarter, and SalesAmount. We have a total of 32 records/rows.
- ❖ The Section column can have one value from amongst “Men”, “Women”, “Kid”, and “Infant”.
- ❖ The ProductCategory Name column can have either the value “Accessories” or the value “Clothing”.
- ❖ The YearQuarter column can have one value from amongst “Q1”, “Q2”, “Q3”, and “Q4”.
- ❖ The SalesAmount column record the sales figures for each Section, ProductCategory Name, and Year Quarter.

OLAP - Example

<i>Section</i>	<i>ProductCategoryName</i>	<i>YearQuarter</i>	<i>SalesAmount</i>
Men	Accessories	Q1	3000.50
Men	Accessories	Q2	1000.50
Men	Accessories	Q3	3500.50
Men	Accessories	Q4	2556.50
Women	Accessories	Q1	1250.50
Women	Accessories	Q2	1000.50
Women	Accessories	Q3	1500.50
Women	Accessories	Q4	1556.50
Kid	Accessories	Q1	1234.50
Kid	Accessories	Q2	5678.50
Kid	Accessories	Q3	1233.50
Kid	Accessories	Q4	1567.50
Infant	Accessories	Q1	1555.50
Infant	Accessories	Q2	2000.50

Characteristics of OLAP

- Multidimensional analysis
- Support for complex queries
- Advanced database support
 - Support large databases
 - Access different data sources
 - Access aggregated data and detailed data

Characteristics of OLAP

- Easy-to-use End-user interface
 - Easy to use graphical interfaces
 - Familiar interfaces with previous data analysis tools
- Client-Server Architecture
 - Provides flexibility
 - Can be used on different computers
 - More machines can be added

One Dimensional

Consider the table shown in the earlier slide - It displays “AllGoods” store’s sales data by Section, which is one-dimensional .

Figure 3.4 shows data in two dimensions (horizontal and vertical), in OLAP it is considered to be one dimension as we are looking at the SalesAmount from one particular perspective, i.e. by Section.

Table 3.4 One-dimensional data by Section

Section	SalesAmount
Infant	22124.00
Kid	34070.00
Men	18313.00
Women	16941.00

Table 3.5 One-dimensional data by ProductCategoryName

ProductCategoryName	SalesAmount
Accessories	33837.00
Clothing	57611.00

Table 3.6 One-dimensional data by YearQuarter

ProductCategoryName	SalesAmount
Q1	16924.00
Q2	22046.00
Q3	26663.00
Q4	25815.00

Table 3.5 presents the sales data of the “AllGoods” stores by ProductCategoryName. This data is again in one dimension as we are looking at the SalesAmount from one particular perspective, ie.ProductCategoryName.

Table 3.6 presents the “AllGoods” sales data by yet another dimension, i.e. YearQuarter. However, this data is yet another example of one-dimensional data as we are looking at the SalesAmount from one particular perspective, i.e. by YearQuarter.

Two Dimensional

One-dimensional data was easy. What if, the requirement was to view Company's data by calendar quarters and product categories? Here, two-dimensional data comes into play. The two-dimensional depiction of data allows one the liberty to think about dimensions as a kind of coordinate system.

Table 3.7 gives you a clear idea of the two-dimensional data. In this table, two dimensions (YearQuarters and ProductCategoryName) have been combined.

Table 3.7 Two-dimensional data by YearQuarter and ProductCategoryName

<i>YearQuarter</i>	<i>Accessories</i>	<i>Clothing</i>	<i>SalesAmount</i>
Q1	7041	9883	16924
Q2	9680	12366	22046
Q3	9660	17003	26663
Q4	7456	18359	25815
Total	33837	57611	91448

In Table 3.7, data has been plotted along two dimensions as we can now look at the SalesAmount from two perspectives, i.e. by YearQuarter and ProductCategoryName. The calendar quarters have been listed along the vertical axis and the product categories have been listed across the horizontal axis. Each unique pair of values of these two dimensions corresponds to a single point of SalesAmount data. For example, the Accessories sales for Q2 add up to \$9680.00 whereas the Clothing sales for the same quarter total up to \$12366.00. Their sales figures correspond to a single point of SalesAmount data, i.e. \$22046.

Three Dimensional

What if the company's analyst wishes to view the data — all of it — along all the three dimensions (Year-Quarter, ProductCategoryName, and Section) and all on the same table at the same time? For this the analyst needs a three-dimensional view of data as arranged in Table 3.8. In this table, one can now look at the data by all the three dimensions/perspectives, i.e. Section, ProductCategoryName, YearQuarter. If the analyst wants to look for the section which recorded maximum Accessories sales in Q2, then by giving a quick glance to Table 3.8, he can conclude that it is the Kid section.

Table 3.8 Three-dimensional data by Section, ProductCategoryName, and YearQuarter

<i>ProductCategoryName</i>	<i>YearQuarter</i>	<i>Men</i>	<i>Women</i>	<i>Kid</i>	<i>Infant</i>	<i>Total</i>
Accessories	Q1	3000.5	1250.5	1234.5	1555.5	7041
	Q2	1000.5	1000.5	5678.5	2000.5	9680
	Q3	3500.5	1500.5	1233.5	3425.5	9660
	Q4	2556.5	1556.5	1567.5	1775.5	7456
Clothing	Q1	2000.5	4536.5	1000.5	2345.5	9883
	Q2	1230.5	2345.5	6789.5	2000.5	12366
	Q3	1456.5	3200.5	8889.5	3456.5	17003
	Q4	3567.5	1550.5	7676.5	5564.5	18359
Total		18313	16941	34070	22124	91448

Can we go beyond Three Dimensional?

✓ Well, if the question is “Can you go beyond the third dimension?” the answer is YES!

✓ If at all there is any constraint, it is because of the limits of your software. But if the question is “Should you go beyond the third dimension?” we will say it is entirely on what data has been captured by your operational transactional systems and what kind of queries you wish your OLAP system to respond to.

✓ Now that we understand multi-dimensional data, it is time to look at the functionalities and characteristics of an OLAP system. OLAP systems are characterized by a low volume of transactions that involve very complex queries. Some typical applications of OLAP are: budgeting, sales forecasting, sales reporting, business process manage

✓ Example: Assume a financial analyst reports that the sales by the company have gone up. The next question is “Which Section is most responsible for this increase?” The answer to this question is usually followed by a barrage of questions such as “Which store in this Section is most responsible for the increase?” or “Which particular product category or categories registered the maximum increase?” The answers to these are provided by multidimensional analysis or OLAP;

Can we go beyond Three Dimensional?

✓ Let us go back to our example of a company's ("AllGoods") sales data viewed along three dimensions: Section, ProductCategoryName, and YearQuarter.

✓ Given below are a set of queries, related to example, that a typical OLAP system is capable of responding to:

- What will be the future sales trend for "Accessories" in the "Kid's" Section?
- Given the customers buying pattern, will it be profitable to launch product "XYZ" in the "Kid's" Section?
- What impact will a 5% increase in the price of produces have on the customers?

Advantages of an OLAP System

- Multi-dimensional data representation.
- Consistency of information.
- “What if” analysis.
- Provides a single platform for all information and business needs – planning, budgeting, forecasting, reporting and analysis.
- Fast and interactive ad hoc exploration.

Answer a Quick Question

According to your understanding,
what are some of the queries that OLAP systems can process?

OLTP vs. OLAP

	OLTP	OLAP
	Online Transaction Processing	Online Analytical Processing
Focus	Data in	Data out
Source of data	Operational/Transactional Data	Data extracted from various operational data sources, transformed and loaded into the data warehouse
Purpose of data	Manage (control and execute) basic business tasks	Assists in planning, budgeting, forecasting and decision making
Data contents	Current data. Far too detailed – not suitable for decision making	Historical data. Has support for summarization and aggregation. Stores and manages data at various levels of granularity, thereby suitable for decision making
Inserts and updates	Very frequent updates and inserts	Periodic updates to refresh the data warehouse
Queries	Simple queries, often returning fewer records	Often complex queries involving aggregations
Processing speed	Usually returns fast	Queries usually take a long time (several hours) to execute and return
Access	Field level access	Typically aggregated access to data of business interest

OLTP vs. OLAP

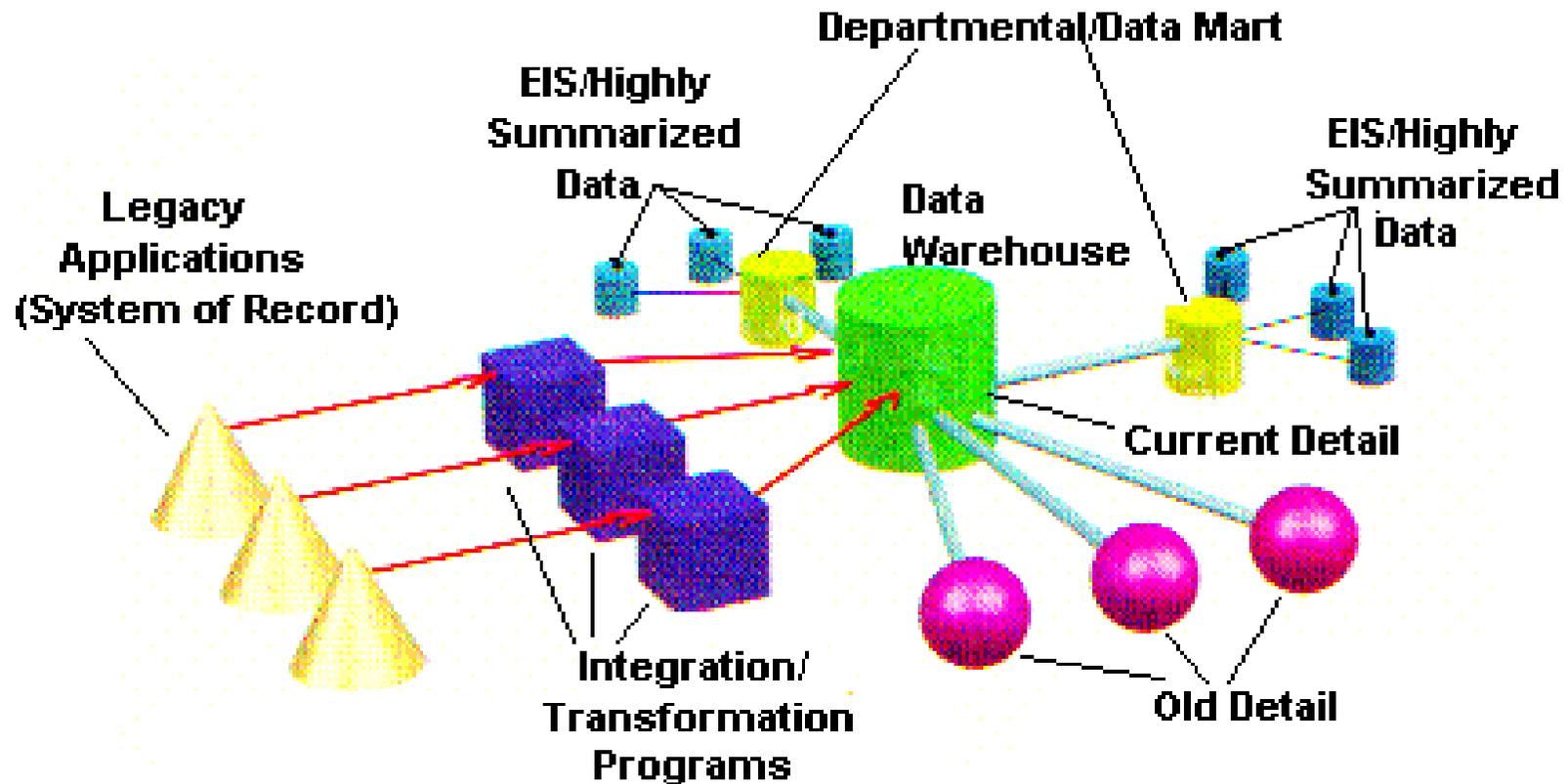
	OLTP	OLAP
	Online Transaction Processing	Online Analytical Processing
Database Design	Typically normalized tables. OLTP system adopts ER (Entity Relationship) model	Typically de-normalized tables; uses star or snowflake schema
Operations	Read/Write	Mostly read
Backup and Recovery	Regular backups of operational data are mandatory. Requires concurrency control (locking) and recovery mechanisms (logging)	Instead of regular backups, data warehouse is refreshed periodically using data from operational data sources
Joins	Many	Few
Derived data and aggregates	Rare	Common
Data Structures	Complex	Multi-dimensional
Few Sample Queries	<ul style="list-style-type: none"> • Search & locate student(s) • Print student scores • Filter students above 90% marks 	<ul style="list-style-type: none"> • Which courses have productivity impact on-the-job? • How much training is needed on future technologies for non-linear growth in BI? • Why consider investing in DSS experience lab?

Sr.No.	Data Warehouse (OLAP)	Operational Database (OLTP)
1	Involves historical processing of information.	Involves day-to-day processing.
2	OLAP systems are used by knowledge workers such as executives, managers and analysts.	OLTP systems are used by clerks, DBAs, or database professionals.
3	Useful in analyzing the business.	Useful in running the business.
4	It focuses on Information out.	It focuses on Data in.
5	Based on Star Schema, Snowflake, Schema and Fact Constellation Schema.	Based on Entity Relationship Model.
7	Provides summarized and consolidated data.	Provides primitive and highly detailed data.
8	Provides summarized and multidimensional view of data.	Provides detailed and flat relational view of data.
9	Number of users is in hundreds.	Number of users is in thousands.
10	Number of records accessed is in millions.	Number of records accessed is in tens.
11	Database size is from 100 GB to 1 TB	Database size is from 100 MB to 1 GB.
12	Highly flexible.	Provides high performance.

OLAP Cube

Data Warehouse Models and OLAP Operations

Data Warehouse Architecture



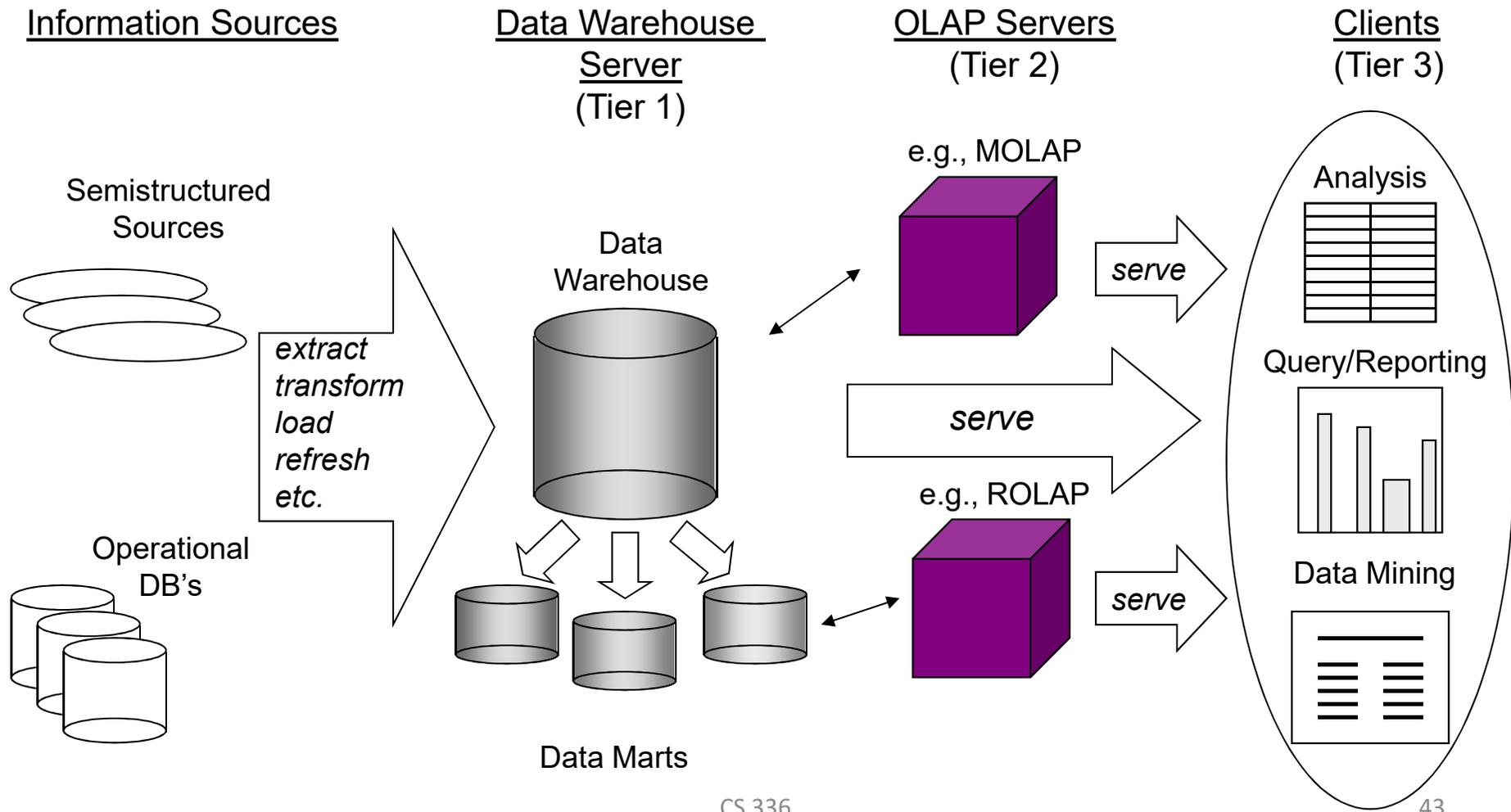
Decision Support

- Information technology to help the knowledge worker (executive, manager, analyst) make faster & better decisions
 - *“What were the sales volumes by region and product category for the last year?”*
 - *“How did the share price of comp. manufacturers correlate with quarterly profits over the past 10 years?”*
 - *“Which orders should we fill to maximize revenues?”*
- On-line analytical processing (OLAP) is an element of decision support systems (DSS)

Three-Tier Decision Support Systems

- Warehouse database server
 - Almost always a relational DBMS, rarely flat files
- OLAP servers
 - Relational OLAP (ROLAP): extended relational DBMS that maps operations on multidimensional data to standard relational operators
 - Multidimensional OLAP (MOLAP): special-purpose server that directly implements multidimensional data and operations
- Clients
 - Query and reporting tools
 - Analysis tools
 - Data mining tools

The Complete Decision Support System



Data Warehouse vs. Data Marts

- *Enterprise warehouse*: collects all information about subjects (*customers, products, sales, assets, personnel*) that span the entire organization
 - Requires extensive business modeling (may take years to design and build)
- *Data Marts*: Departmental subsets that focus on selected subjects
 - Marketing data mart: customer, product, sales
 - Faster roll out, but complex integration in the long run
- *Virtual warehouse*: views over operational dbs
 - Materialize sel. summary views for efficient query processing
 - Easy to build but require excess capability on operat. db servers

Approaches to OLAP Servers

- Relational DBMS as Warehouse Servers
- Two possibilities for OLAP servers
- (1) Relational OLAP (ROLAP)
 - Relational and specialized relational DBMS to store and manage warehouse data
 - OLAP middleware to support missing pieces
- (2) Multidimensional OLAP (MOLAP)
 - Array-based storage structures
 - Direct access to array data structures

OLAP Server: Query Engine Requirements

- Aggregates (maintenance and querying)
 - Decide what to precompute and when
- Query language to support multidimensional operations
 - Standard SQL falls short
- Scalable query processing
 - Data intensive and data selective queries

OLAP for Decision Support

- OLAP = Online Analytical Processing
- Support (almost) ad-hoc querying for business analyst
- Think in terms of spreadsheets
 - View sales data by geography, time, or product
- Extend spreadsheet analysis model to work with warehouse data
 - Large data sets
 - Semantically enriched to understand business terms
 - Combine interactive queries with reporting functions
- Multidimensional view of data is the foundation of OLAP
 - Data model, operations, etc.

Warehouse Models & Operators

- Data Models
 - relations
 - stars & snowflakes
 - cubes
- Operators
 - slice & dice
 - roll-up, drill down
 - pivoting
 - other

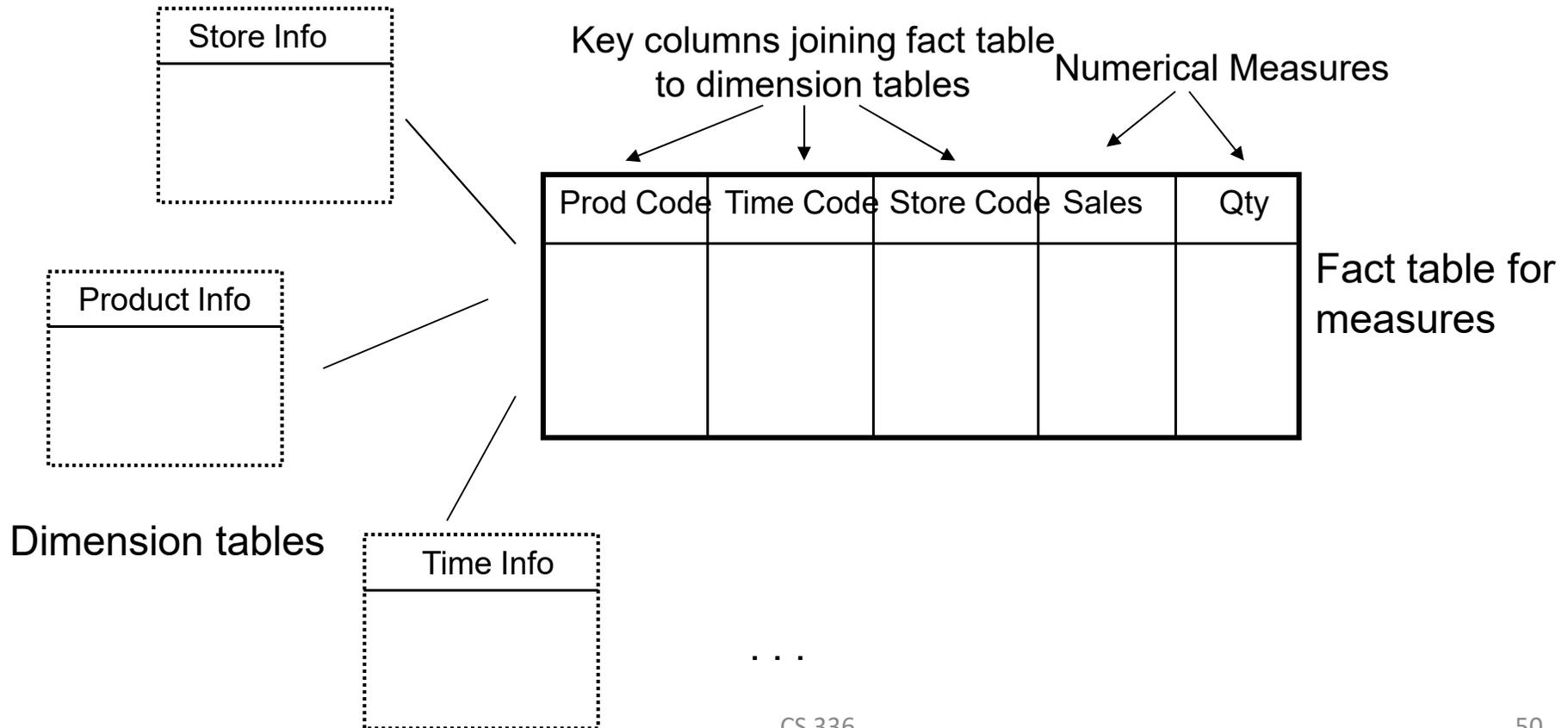
Multi-Dimensional Data

- Measures - numerical data being tracked
- Dimensions - business parameters that define a transaction
- Example: Analyst may want to view sales data (measure) by geography, by time, and by product (dimensions)
- Dimensional modeling is a technique for structuring data around the business concepts
- ER models describe “entities” and “relationships”
- Dimensional models describe “measures” and “dimensions”

The Multi-Dimensional Model

“Sales by product line over the past six months”

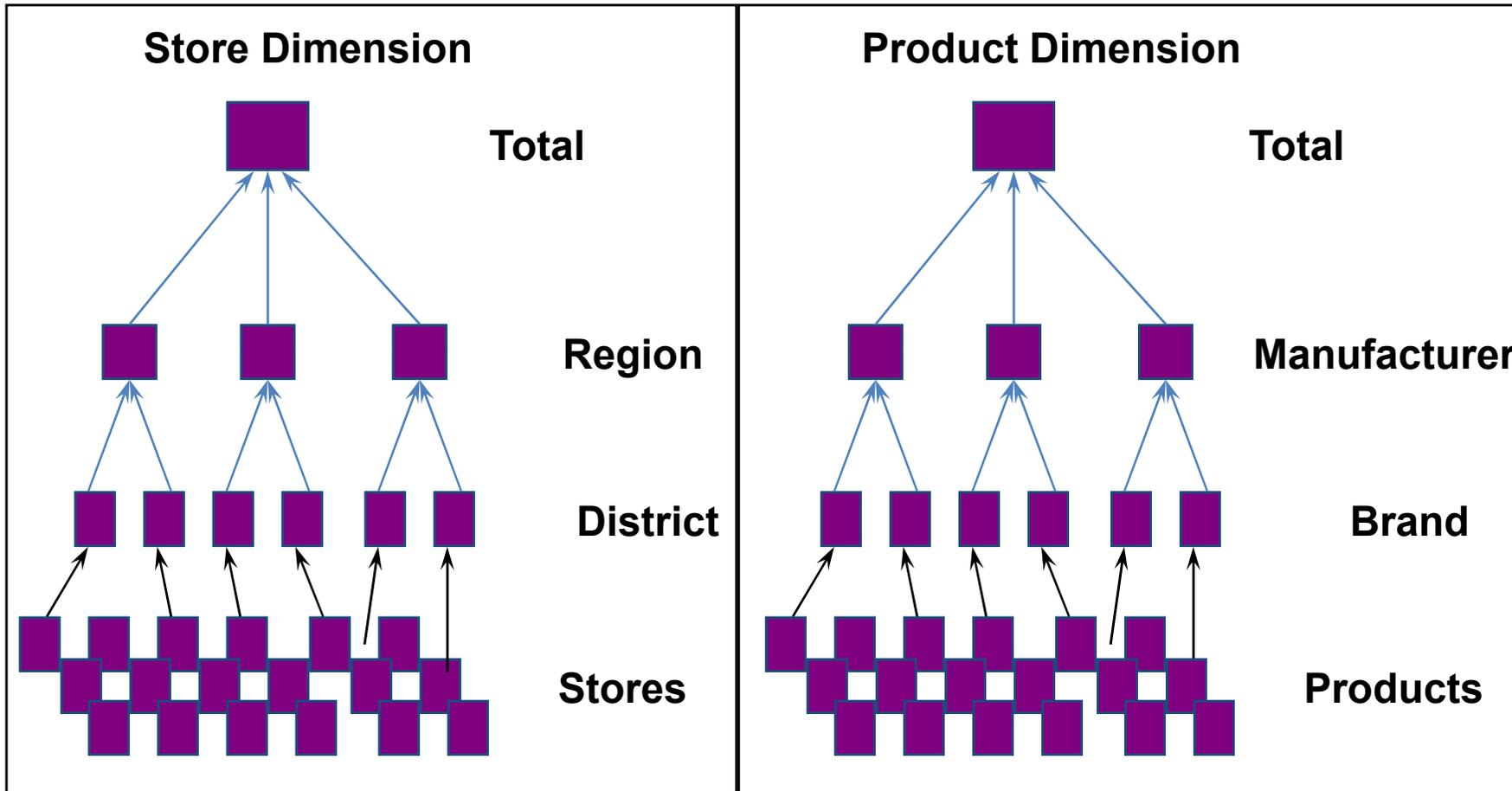
“Sales by store between 1990 and 1995”



Dimensional Modeling

- Dimensions are organized into hierarchies
 - E.g., Time dimension: days → weeks → quarters
 - E.g., Product dimension: product → product line → brand
- Dimensions have attributes

Dimension Hierarchies



ROLAP: Dimensional Modeling

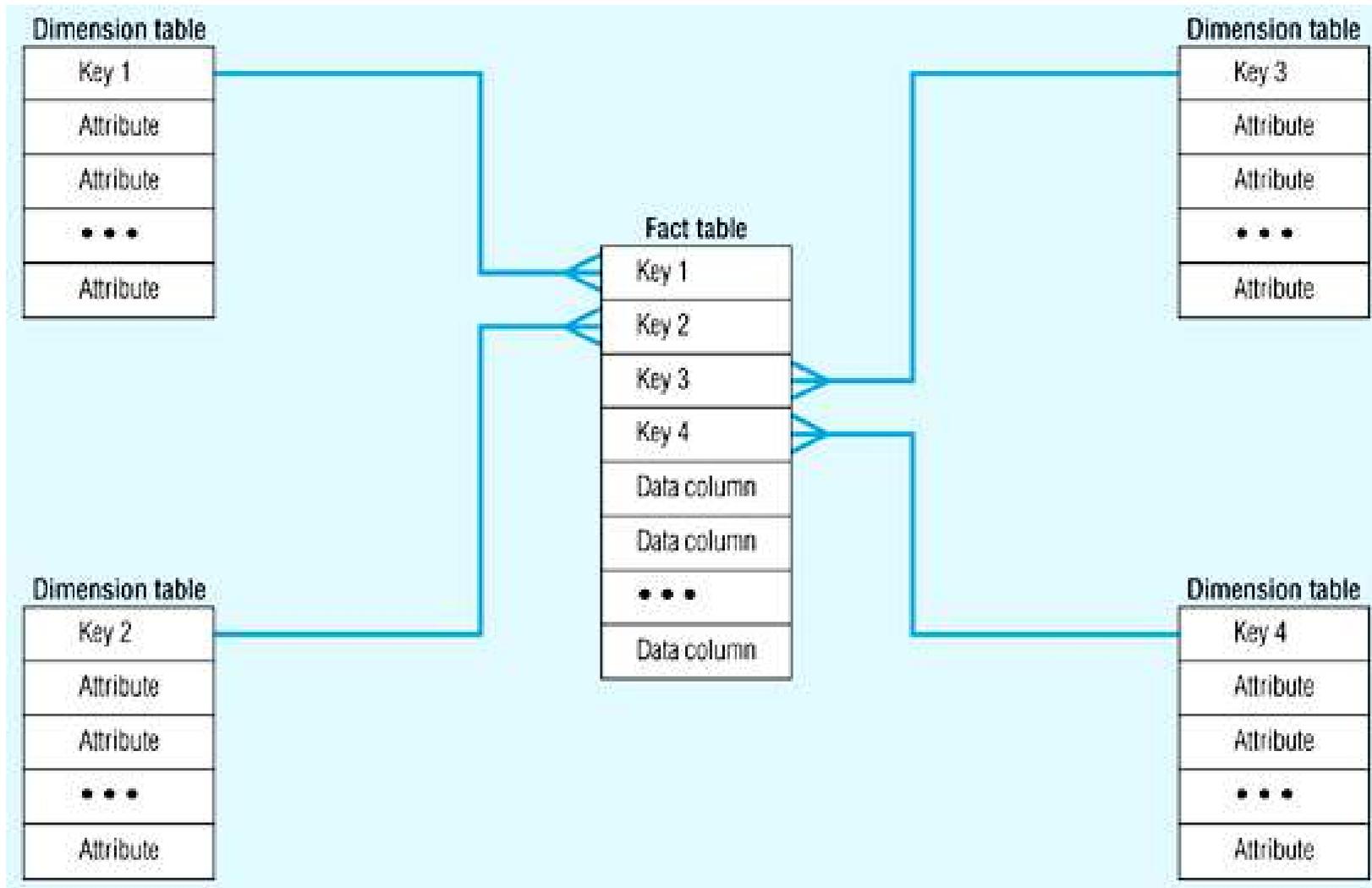
Using Relational DBMS

- Special schema design: *star, snowflake*
- Special indexes: bitmap, multi-table join
- Special tuning: maximize query throughput
- Proven technology (relational model, DBMS), tend to outperform specialized MDDDB especially on large data sets
- Products
 - IBM DB2, Oracle, Sybase IQ, RedBrick, Informix

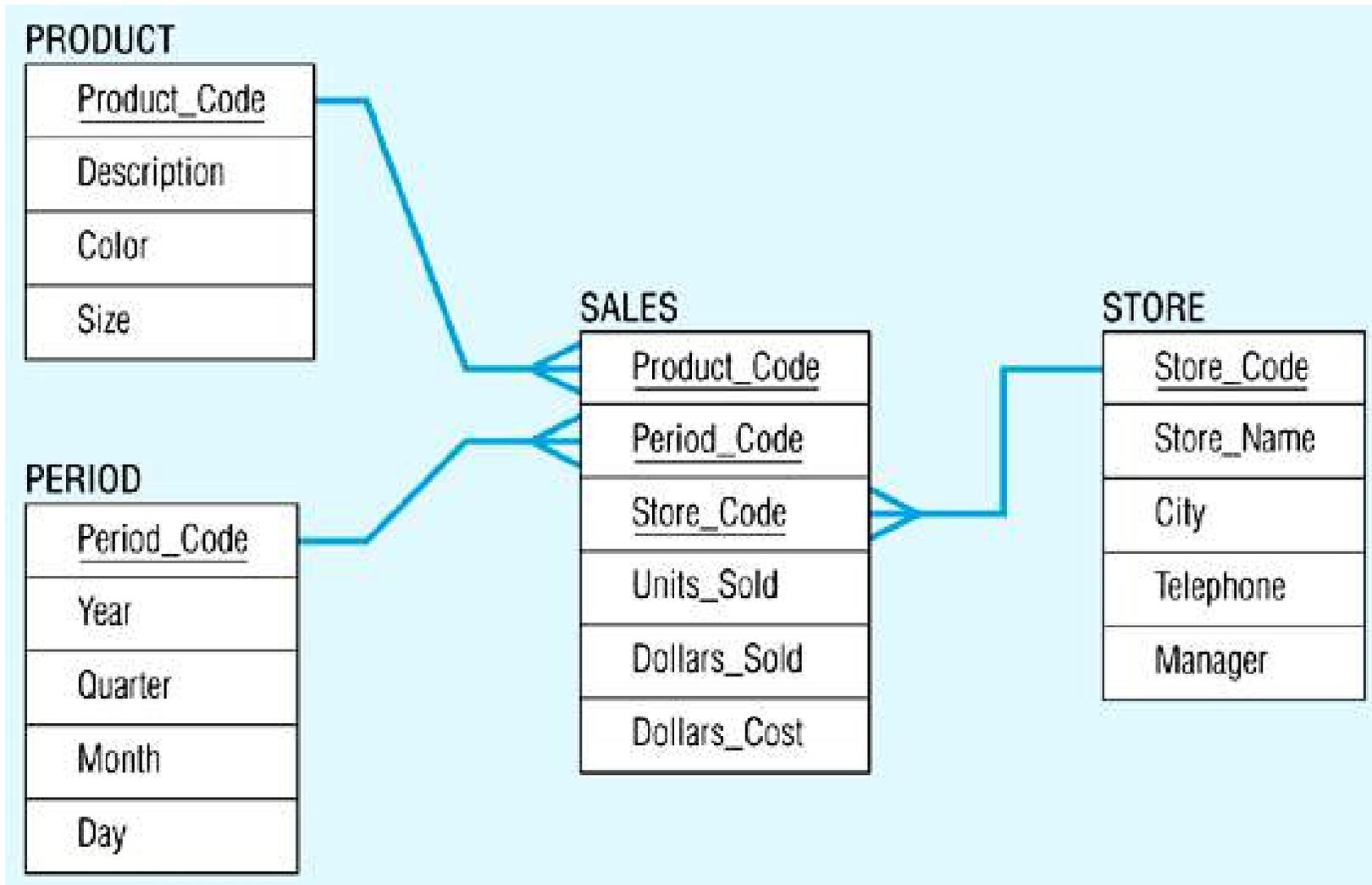
MOLAP: Dimensional Modeling Using the Multi Dimensional Model

- MDDDB: a special-purpose data model
- Facts stored in multi-dimensional arrays
- Dimensions used to index array
- Sometimes on top of relational DB
- Products
 - Pilot, Arbor Essbase, Gentia

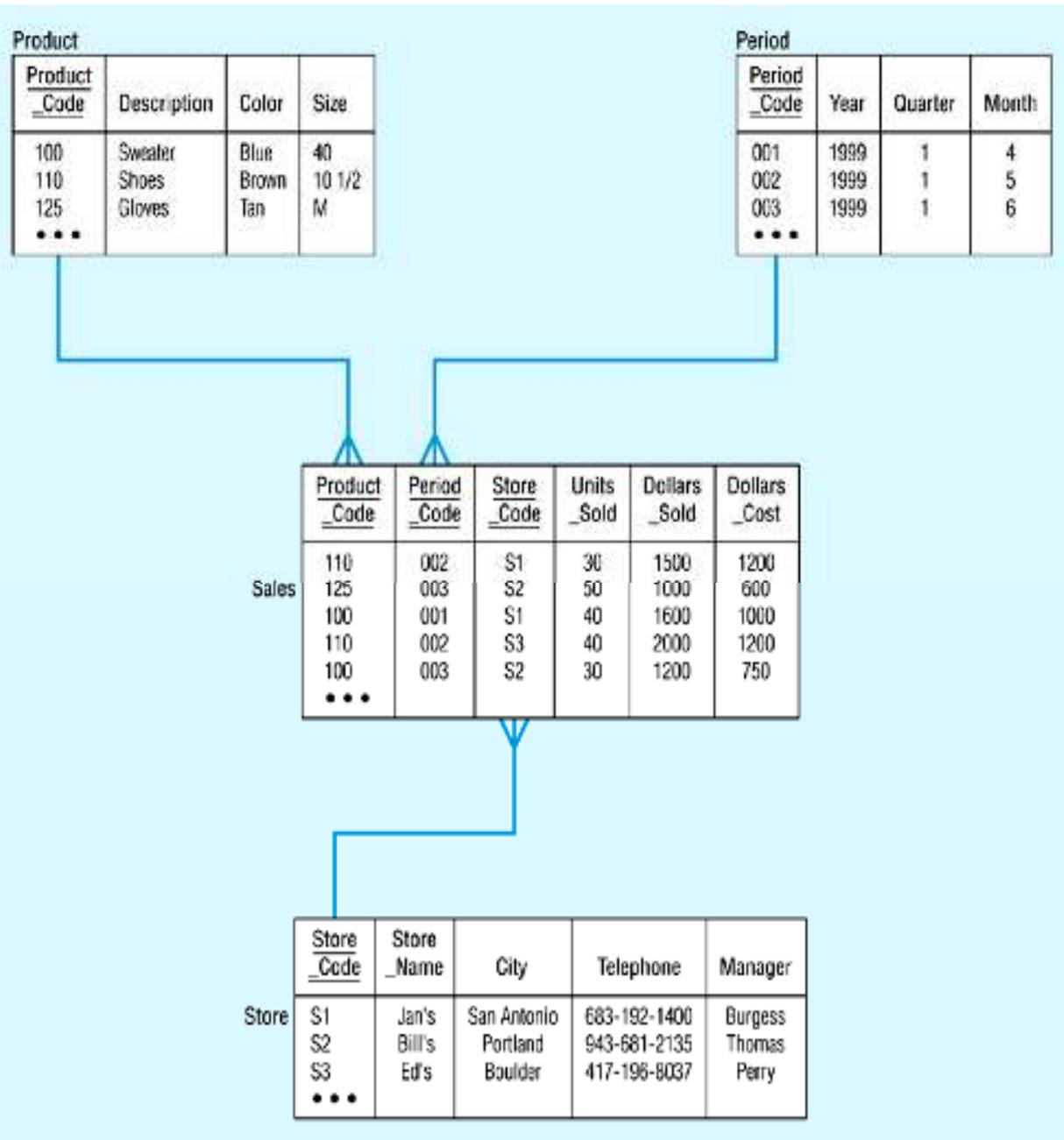
Star Schema (in RDBMS)



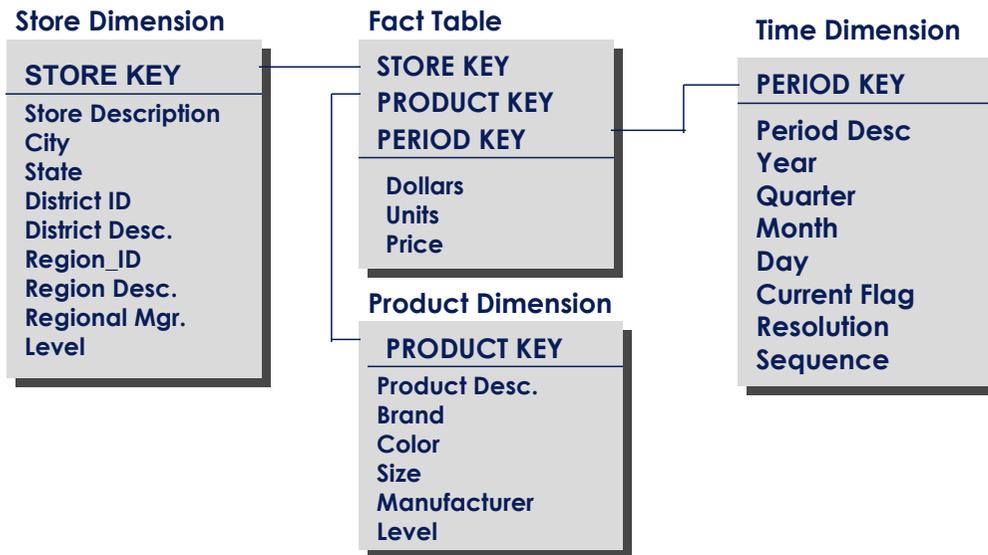
Star Schema Example



Star Schema with Sample Data



The “Classic” Star Schema

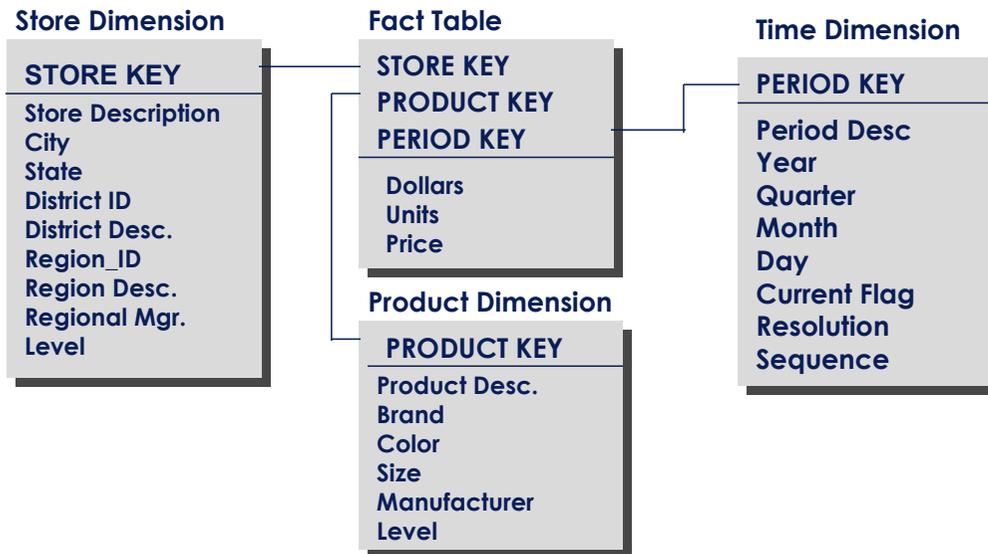


- ◆ A single fact table, with detail and summary data
- ◆ Fact table primary key has only one key column per dimension
- ◆ Each key is generated
- ◆ Each dimension is a single table, highly denormalized

Benefits: Easy to understand, easy to define hierarchies, reduces # of physical joins, low maintenance, very simple metadata

Drawbacks: Summary data in the fact table yields poorer performance for summary levels, huge dimension tables a problem

The “Classic” Star Schema



The biggest drawback: dimension tables must carry a *level* indicator for every record and every query must use it. In the example below, without the level constraint, keys for all stores in the NORTH region, including aggregates for region and district will be pulled from the fact table, resulting in error.

Example:

Select A.STORE_KEY, A.PERIOD_KEY, A.dollars from Fact_Table A

where A.STORE_KEY in (select STORE_KEY from Store_Dimension B where region = “North” and Level = 2)

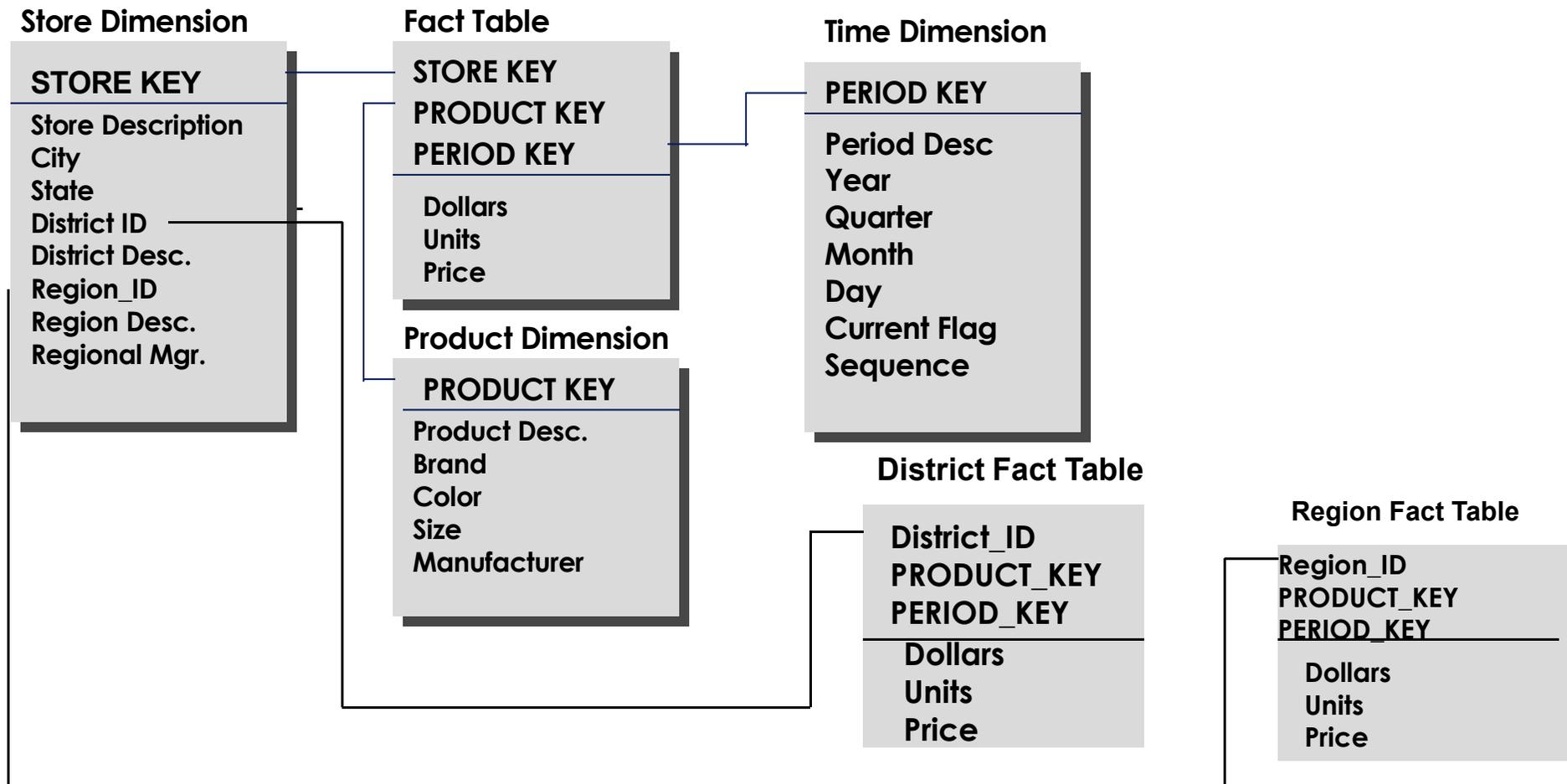
and *etc...*

Level is needed whenever aggregates are stored with detail facts.

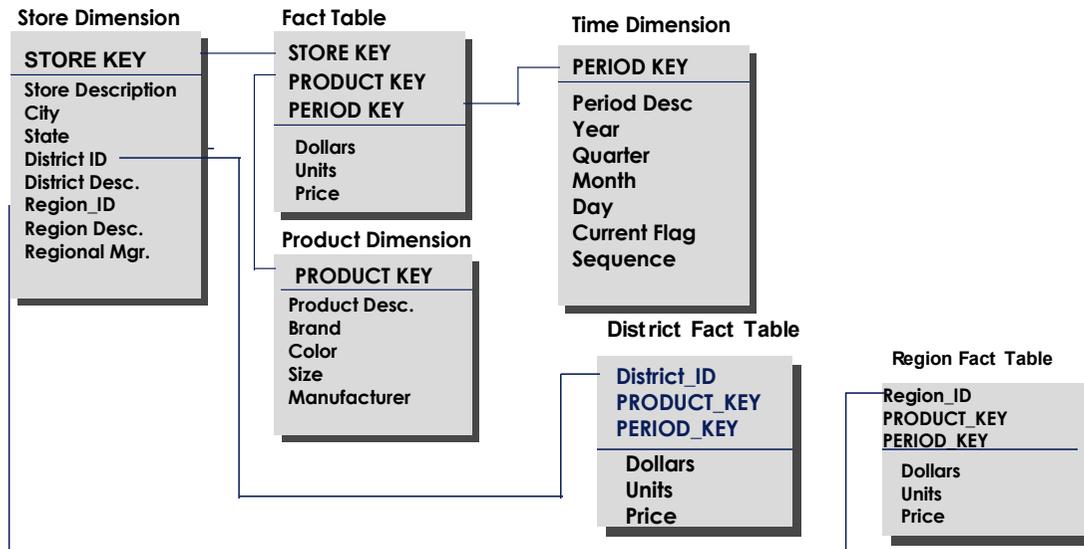
The “Level” Problem

- Level is a problem because because it causes potential for error. If the query builder, human or program, forgets about it, perfectly reasonable looking **WRONG** answers can occur.
- One alternative: the FACT CONSTELLATION model...

The “Fact Constellation” Schema



The “Fact Constellation” Schema



In the Fact Constellations, aggregate tables are created separately from the detail, therefore it is impossible to pick up, for example, Store detail when querying the District Fact Table.

Major Advantage: No need for the “Level” indicator in the dimension tables, since no aggregated data is stored with lower-level detail

Disadvantage: Dimension tables are still very large in some cases, which can slow performance; front-end must be able to detect existence of aggregate facts, which requires more extensive metadata

Another Alternative to “Level”

- Fact Constellation is a good alternative to the Star, but when dimensions have very **high cardinality**, the sub-selects in the dimension tables can be a **source of delay**.
- An alternative is to **normalize the dimension tables** by attribute level, with each smaller dimension table pointing to an appropriate aggregated fact table, the “**Snowflake Schema**”
...

The "Snowflake" Schema

Store Dimension

STORE KEY
Store Description
City
State
District ID
District Desc.
Region_ID
Region Desc.
Regional Mgr.

District_ID
District Desc.
Region_ID

Region_ID
Region Desc.
Regional Mgr.

Store Fact Table

STORE KEY	PRODUCT KEY	PERIOD KEY
Dollars		
Units		
Price		

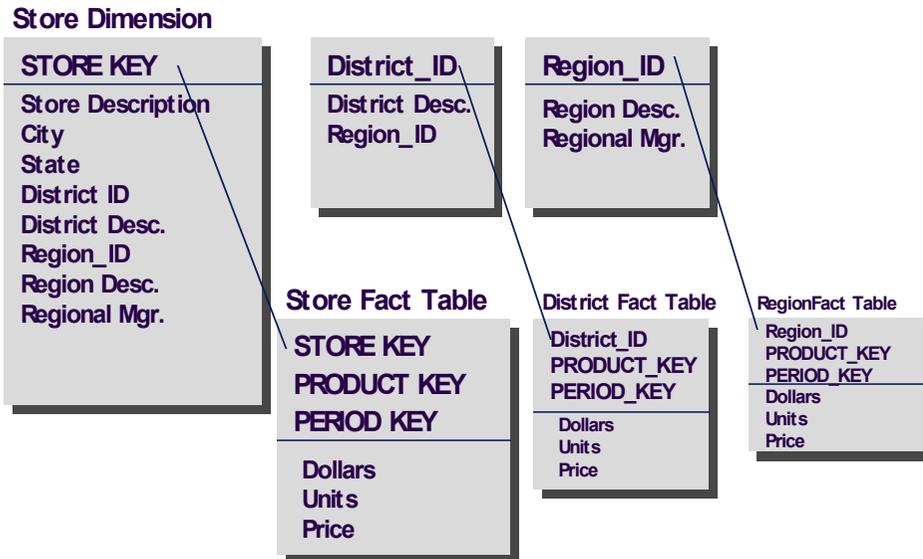
District Fact Table

District_ID	PRODUCT_KEY	PERIOD_KEY
Dollars		
Units		
Price		

RegionFact Table

Region_ID	PRODUCT_KEY	PERIOD_KEY
Dollars		
Units		
Price		

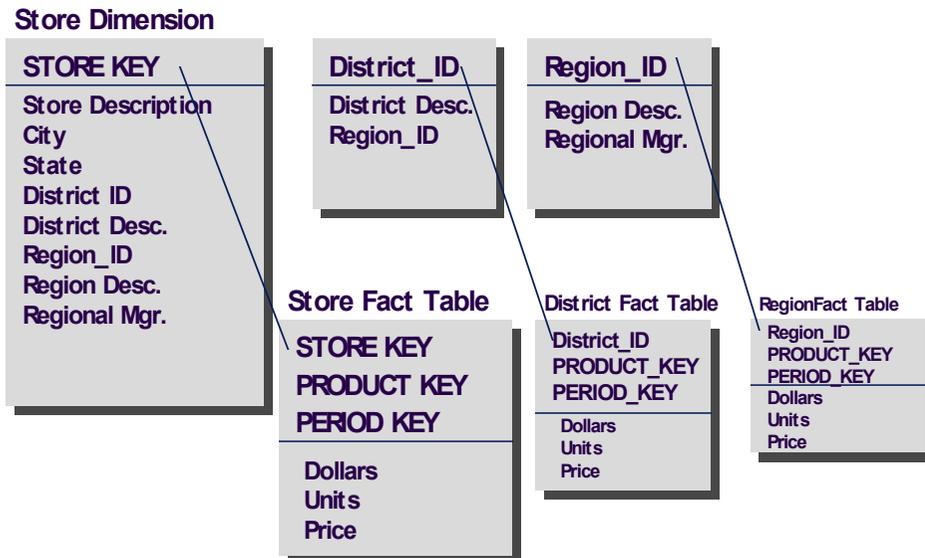
The “Snowflake” Schema



- No **LEVEL** in dimension tables
- Dimension tables are normalized by decomposing at the attribute level
- Each dimension table has one key for each level of the dimension's hierarchy
- The lowest level key joins the dimension table to both the fact table and the lower level attribute table

How does it work? The best way is for the query to be built by understanding which summary levels exist, and finding the proper snowflaked attribute tables, constraining there for keys, then selecting from the fact table.

The “Snowflake” Schema



- **Additional features:** The original Store Dimension table, completely de-normalized, is kept intact, since certain queries can benefit by its all-encompassing content.
- **In practice,** start with a Star Schema and create the “snowflakes” with queries. This eliminates the need to create separate extracts for each table, and referential integrity is inherited from the dimension table.

Advantage: Best performance when queries involve aggregation

Disadvantage: Complicated maintenance and metadata, explosion in the numl of tables in the database

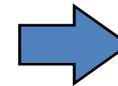
Advantages of ROLAP Dimensional Modeling

- Define complex, multi-dimensional data with simple model
- Reduces the number of joins a query has to process
- Allows the data warehouse to evolve with rel. low maintenance
- **HOWEVER!** Star schema and relational DBMS are not the magic solution
 - Query optimization is still problematic

Aggregates

- Add up amounts for day 1
- In SQL: `SELECT sum(amt) FROM SALE WHERE date = 1`

sale	prold	storeld	date	amt
	p1	s1	1	12
	p2	s1	1	11
	p1	s3	1	50
	p2	s2	1	8
	p1	s1	2	44
	p1	s2	2	4

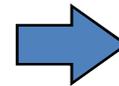


81

Aggregates

- Add up amounts by day
- In SQL: `SELECT date, sum(amt) FROM SALE GROUP BY date`

sale	prold	storeld	date	amt
	p1	s1	1	12
	p2	s1	1	11
	p1	s3	1	50
	p2	s2	1	8
	p1	s1	2	44
	p1	s2	2	4

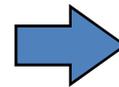


ans	date	sum
	1	81
	2	48

Another Example

- Add up amounts by day, product
- In SQL: `SELECT date, sum(amt) FROM SALE GROUP BY date, prodl`

sale	prold	storeld	date	amt
	p1	s1	1	12
	p2	s1	1	11
	p1	s3	1	50
	p2	s2	1	8
	p1	s1	2	44
	p1	s2	2	4



sale	prold	date	amt
	p1	1	62
	p2	1	19
	p1	2	48

rollup →

← drill-down

Aggregates

- Operators: sum, count, max, min, median, ave
- “Having” clause
- Using dimension hierarchy
 - average by region (within store)
 - maximum by month (within date)

ROLAP vs. MOLAP

- ROLAP:
Relational On-Line Analytical Processing
- MOLAP:
Multi-Dimensional On-Line Analytical
Processing

The MOLAP Cube

Fact table view:

sale	prold	storeld	amt
	p1	s1	12
	p2	s1	11
	p1	s3	50
	p2	s2	8

Multi-dimensional cube:



	s1	s2	s3
p1	12		50
p2	11	8	

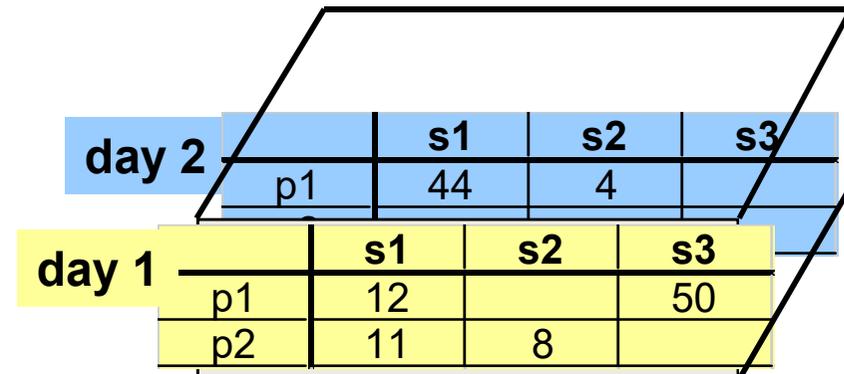
dimensions = 2

3-D Cube

Fact table view:

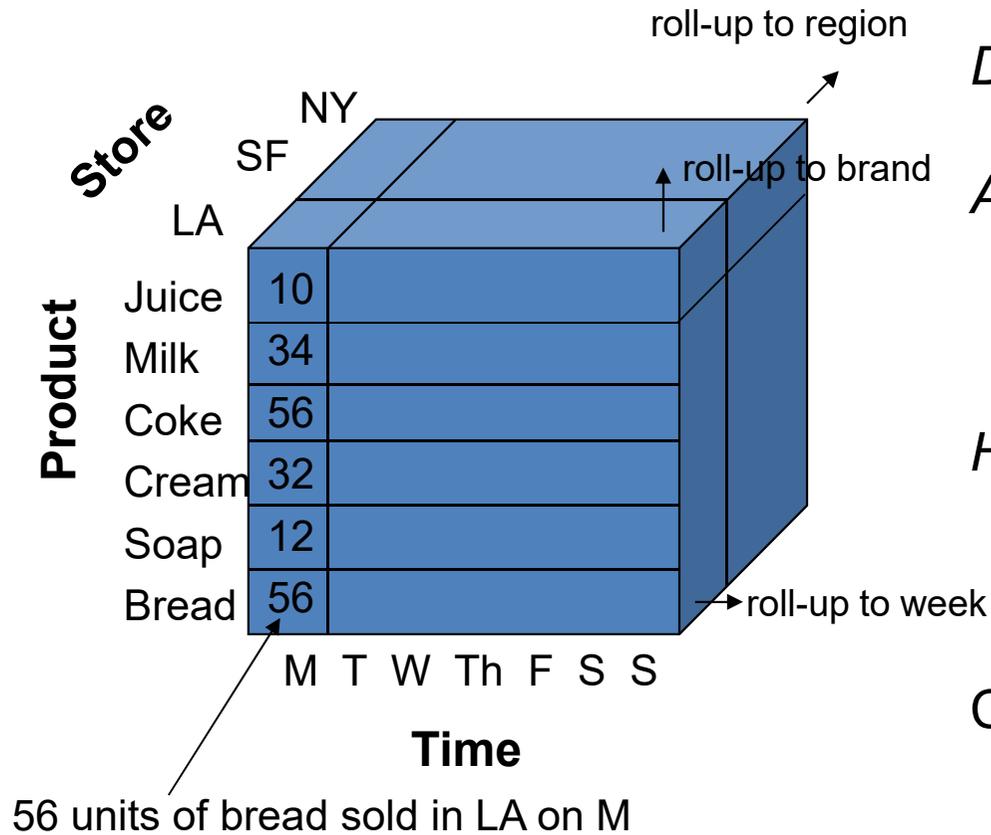
sale	prold	storeld	date	amt
	p1	s1	1	12
	p2	s1	1	11
	p1	s3	1	50
	p2	s2	1	8
	p1	s1	2	44
	p1	s2	2	4

Multi-dimensional cube:



dimensions = 3

Example



Dimensions:

Time, Product, Store

Attributes:

Product (upc, price, ...)

Store ...

...

Hierarchies:

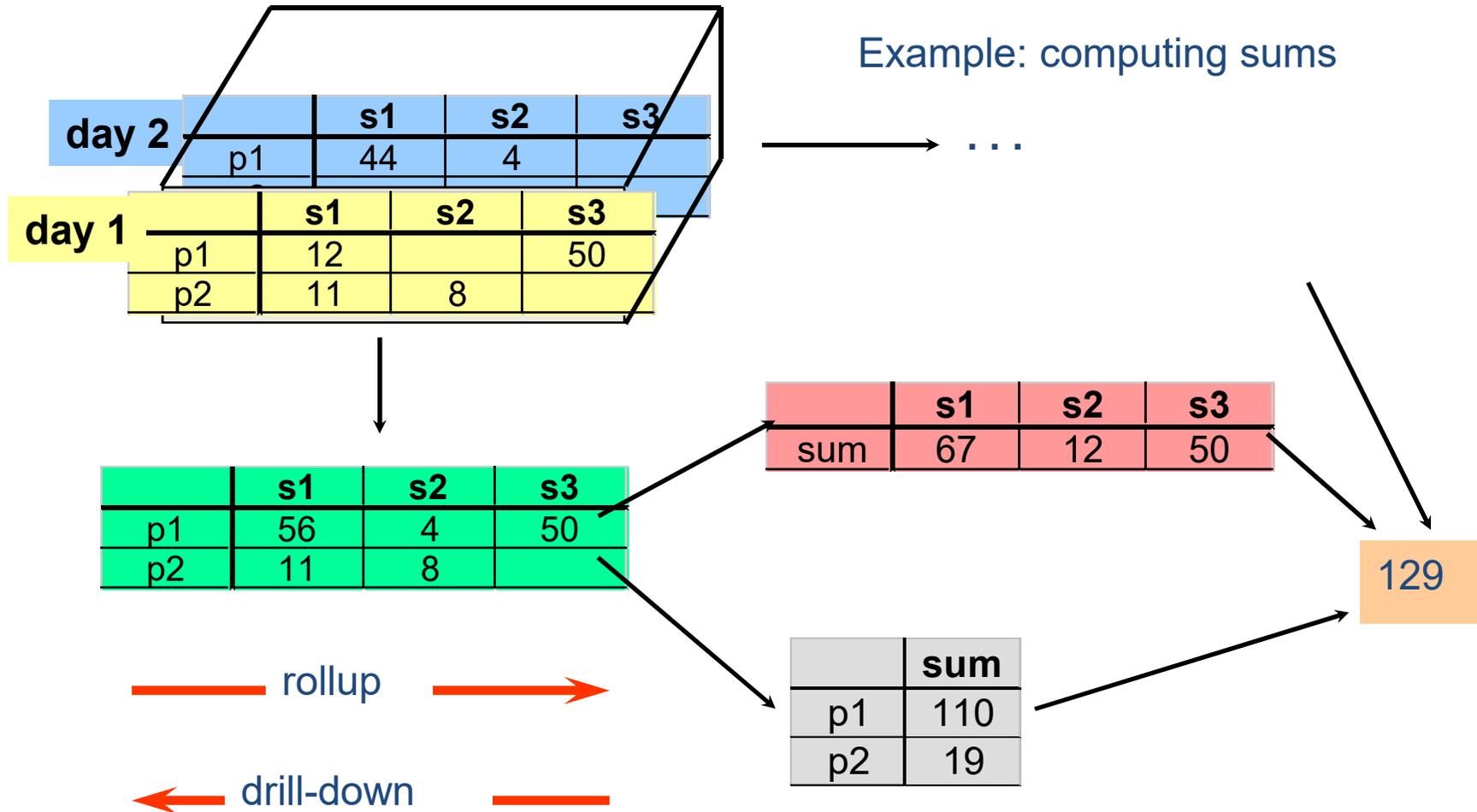
Product → Brand → ...

Day → Week → Quarter

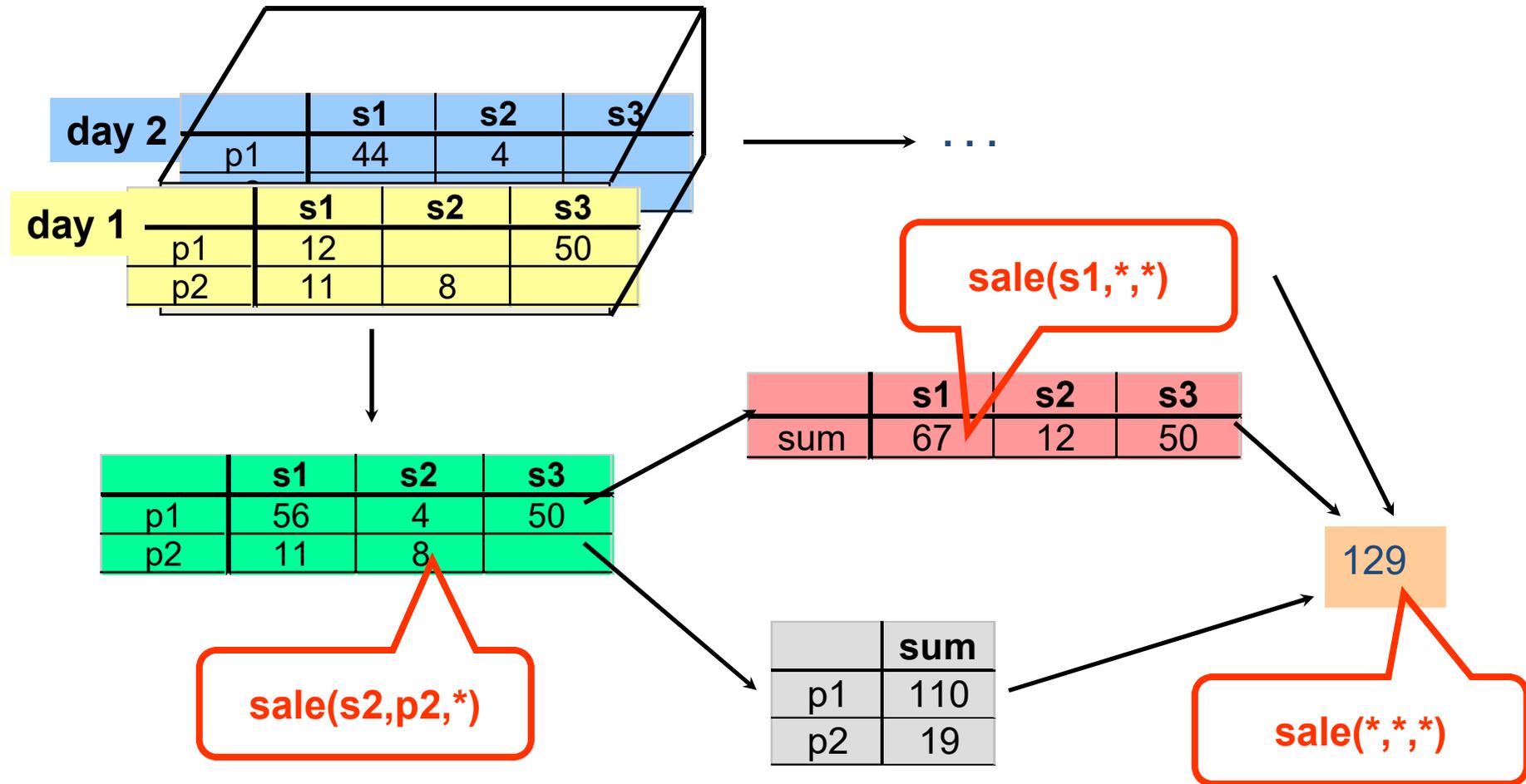
Store → Region →

Country

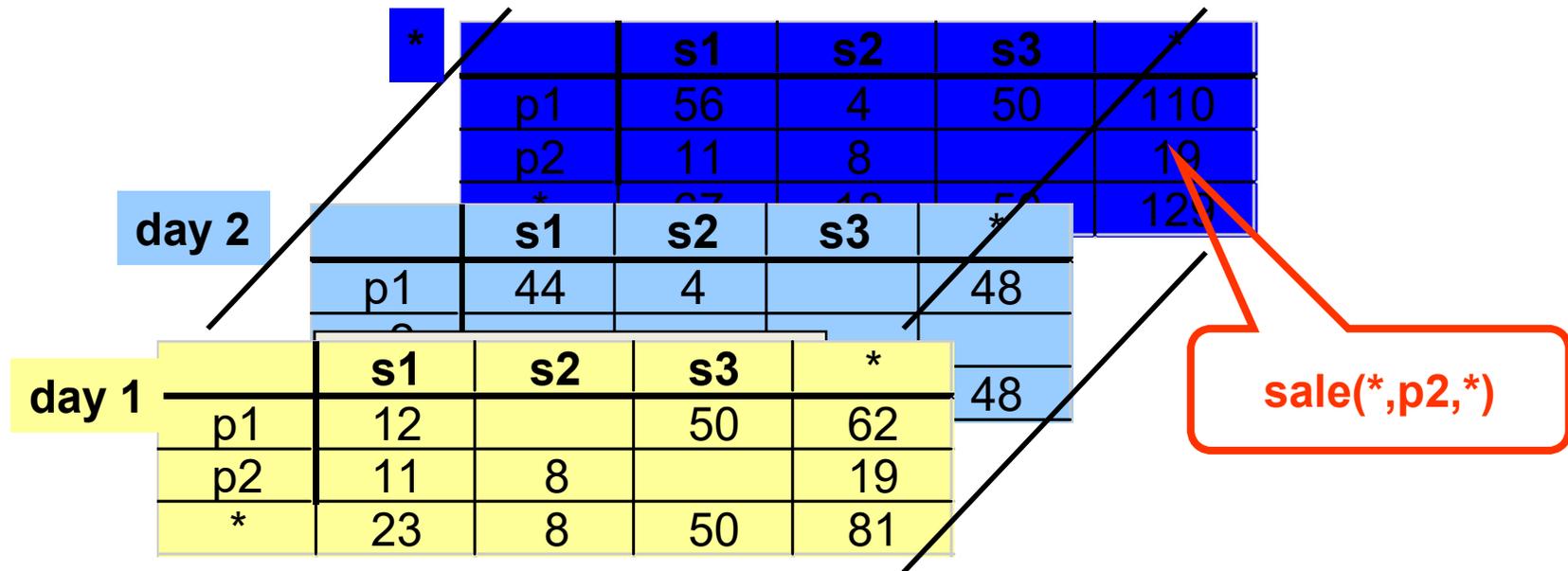
Cube Aggregation: Roll-up



Cube Operators for Roll-up



Extended Cube



Aggregation Using Hierarchies

day	product	store	value
day 1	p1	s1	12
		s2	
		s3	50
	p2	s1	11
		s2	8
		s3	
day 2	p1	s1	44
		s2	4
		s3	



	region A	region B
p1	56	54
p2	11	8



(store s1 in Region A;
stores s2, s3 in Region B)

Slicing

day 2			
	s1	s2	s3
p1	44	4	

day 1			
	s1	s2	s3
p1	12		50
p2	11	8	

TIME = day 1

	s1	s2	s3
p1	12		50
p2	11	8	

Slicing & Pivoting

		Sales (\$ millions)		
		Products	Time	
			d1	d2
Store s1	Electronics	\$5.2		
	Toys	\$1.9		
	Clothing	\$2.3		
	Cosmetics	\$1.1		
Store s2	Electronics	\$8.9		
	Toys	\$0.75		
	Clothing	\$4.6		
	Cosmetics	\$1.5		

		Sales (\$ millions)		
		Products	d1	
			Store s1	Store s2
Store s1	Electronics	\$5.2	\$8.9	
	Toys	\$1.9	\$0.75	
	Clothing	\$2.3	\$4.6	
	Cosmetics	\$1.1	\$1.5	
Store s2	Electronics			
	Toys			
	Clothing			



Summary of Operations

- Aggregation (roll-up)
 - aggregate (summarize) data to the next higher dimension element
 - e.g., total sales by city, year → total sales by region, year
- Navigation to detailed data (drill-down)
- Selection (slice) defines a subcube
 - e.g., sales where city = 'Gainesville' and date = '1/15/90'
- Calculation and ranking
 - e.g., top 3% of cities by average income
- Visualization operations (e.g., Pivot)
- Time functions
 - e.g., time average

Query & Analysis Tools

- Query Building
- Report Writers (comparisons, growth, graphs,...)
- Spreadsheet Systems
- Web Interfaces
- Data Mining

MOLAP, ROLAP, HOLAP

- MOLAP
 - Multidimensional OLAP
- ROLAP
 - Relational OLAP
- HOLAP
 - Hybrid OLAP

MOLAP

- Uses multidimensional approach to solve a problem
- Directly stores the information in cubes
- Used in SSAS (SQL Server Analysis Services)

ROLAP

- Relational databases are used to store the data
- Translates OLAP queries to appropriate SQL statements
- Data created by OLTP is directly used

Do it Exercise

Study the Data models for OLTP and OLAP systems

Hint: ER modeling, Star and Snowflake Schema

The raw data

Car_sales table

Receipt_no	make	color	size	date	price
0001	Toyota	white	small	01/01/01	20000
0002	Nissan	red	Medium	01/01/01	25000
0003	Holden	silver	Medium	01/02/01	299000
....

For analysis, raw data often needs to be summarized

OLAP:example

- Example: find what kinds of cars are popular?

sales(make, color, size, num_sold) (slightly summarized data)

where **make** can be Toyota, Nissan, Holden, Ford etc

colors are white, red, silver

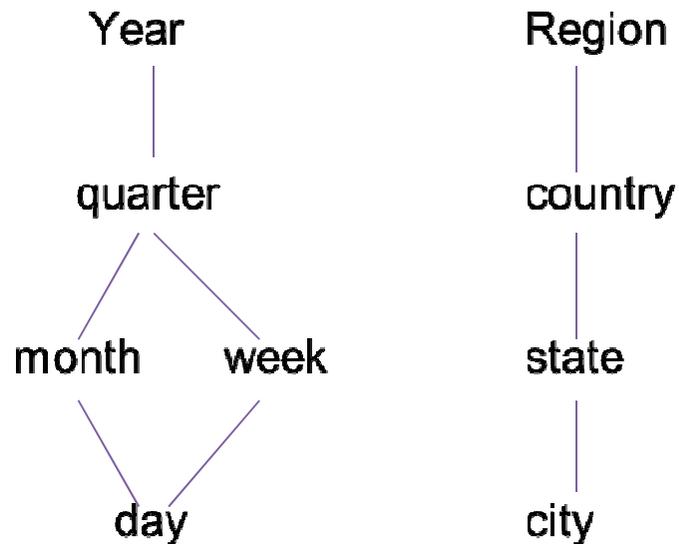
size can be small, medium, large.

- Attributes such as **num_sold** are called *measure attributes*, since they can be used to measure some value, and can be aggregated.
- Attributes like **make, color, size** are called *dimension attributes*, since they define the dimensions on which measure attributes are viewed.

Data that can be modeled as dimension attributes and measure attributes are called *multi-dimensional data*.

Dimension Hierarchies

- For each dimension, the set of values can be organized in a hierarchy.
 - E.g., assuming the sales table includes time and location dimension



- The hierarchies represent different levels of details for attributes.

Cross Tabs and Data Cubes

- OLAP systems allow analyst to view different summaries of the data.
 - The following table can be derived from **sales(make, color, size, num_sold)**

Cross-tab or pivot table

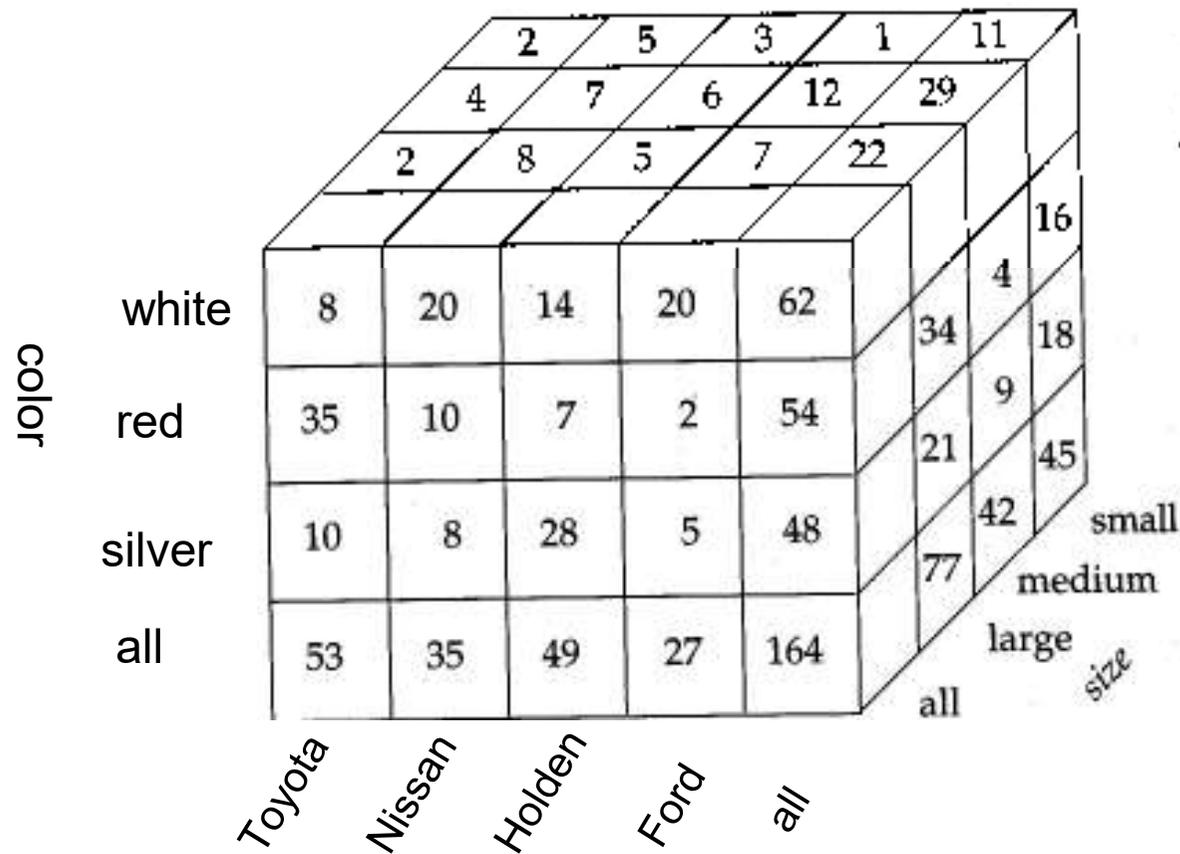
	WHITE	RED	SILVER	TOTAL
TOYOTA	8	35	10	53
NISSAN	20	10	5	35
HOLDEN	14	7	28	49
FORD	20	2	5	27
TOTAL	62	54	48	164

Relational representation

make	color	num_sold
Toyota	white	8
Toyota	red	35
Toyota	silver	10
Toyota	all	53
Nissan	white	20
Nissan	red	10
Nissan	silver	5
Nissan	all	35
Holden	white	14
Holden	red	7
Holden	silver	28
Holden	all	49
Ford	white	20
Ford	red	2
Ford	silver	5
Ford	all	27
all	white	62
all	red	54
all	silver	48
all	all	164

Data Cubes

- The generalization of a cross tab, which is 2-dimensional, to n dimensions can be visualized as a n -dimensional cube, called the *data cube*.



MOLAP vs ROLAP

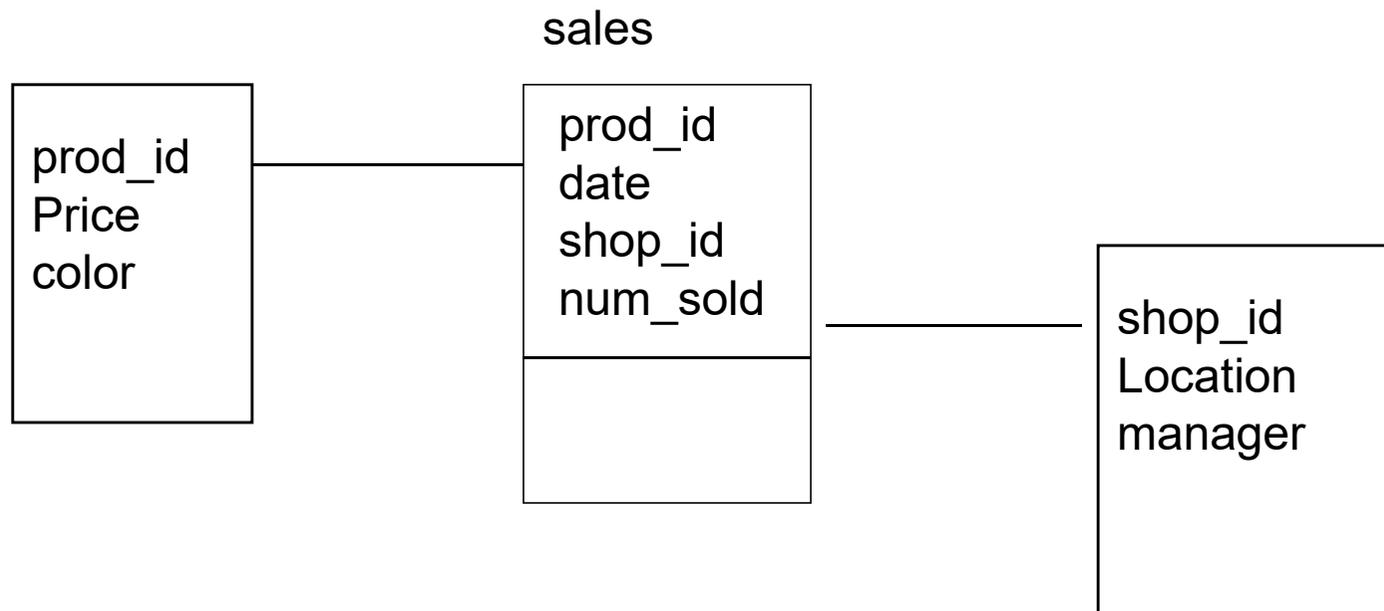
- OLAP systems can use multi-dimensional array to store data cubes, called multidimensional OLAP systems (MOLAP) .
- Alternatively, they can stored data as relations in relational databases, called relational OLAP systems (ROLAP).

ROLAP

- The main relation, which relates dimensions to measures, is called the *fact table*.
 - e.g., **sales(prod_id, date, shop_id, num_sold)**
 - Very large, accumulation of facts such as sales
- Each dimension can have additional attributes and an associated *dimensional table*.
 - E.g., **product(prod_id, price, color)**
prod_id is a foreign key of **sales**
shops(shop_id, location, manager)
 - Dimension data are smaller, generally static

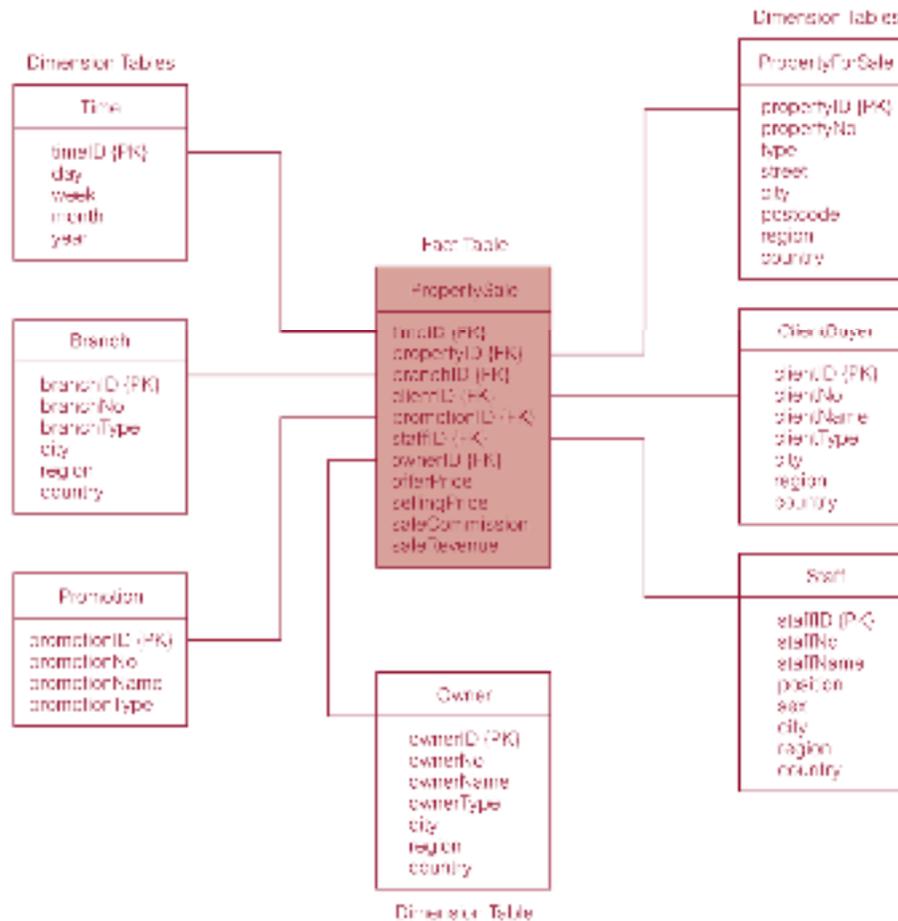
The Star Schema

- In a ROLAP system, relations are often stored with [star schemas](#)
- A star schema consists of the fact table and one or more dimension tables. Dimension tables are usually not normalized, why?
- A typical query often involves a join of the fact table and the dimension tables.



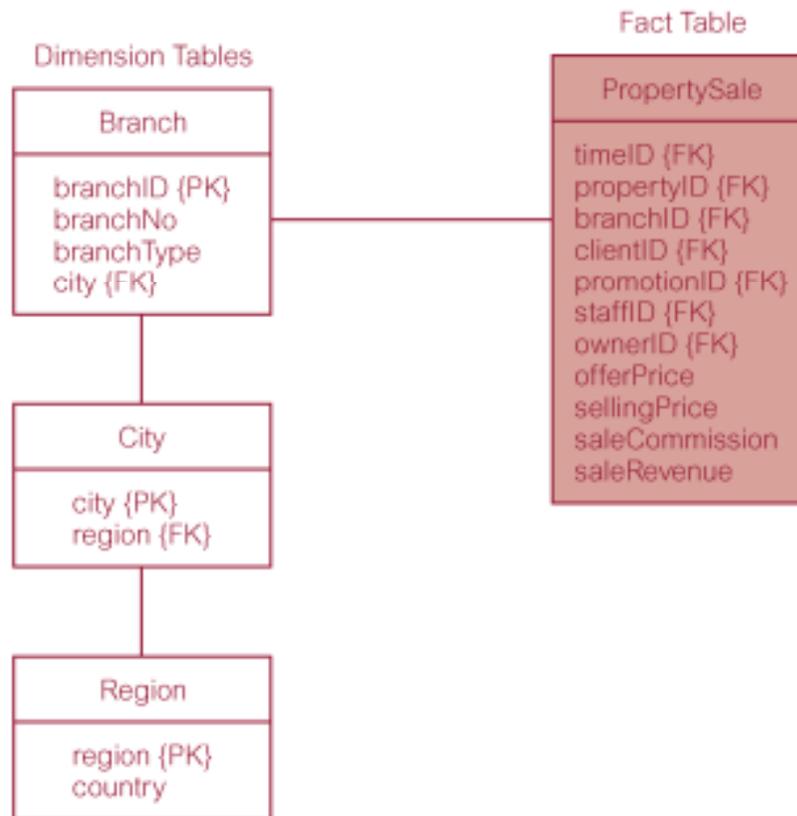
The Star Schema

■ Dimension tables are not in 3NF



The snowflake schema

- A variation of the star schema where the dimension tables are normalized.



Fact constellation

- A set of fact tables that share some dimension tables

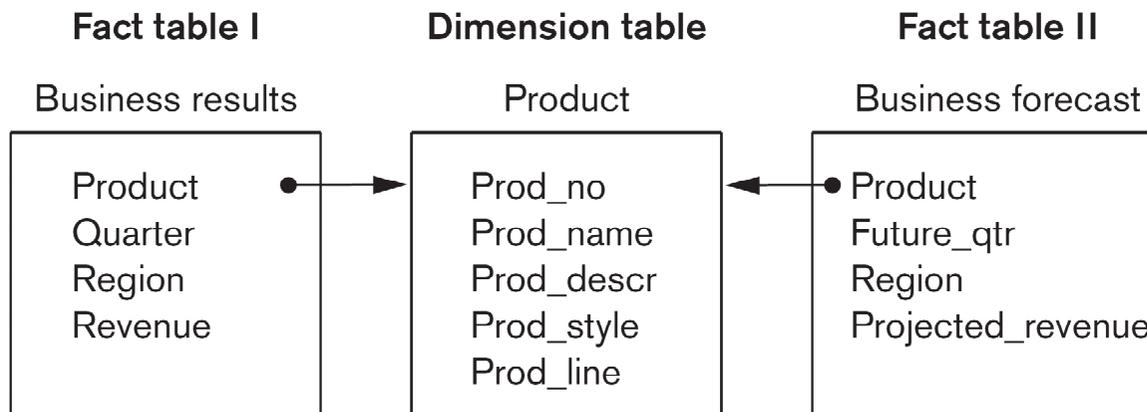


Figure 29.9
A fact constellation.

OLAP Queries

- A common operation is to aggregate a measure over one or more dimensions, e.g.,
 - find total/average sales for a product.
 - find total sales in each city/state/month etc
 - find top 2 products by total sales
- **Roll-up:** moving from finer granularity to coarser granularity by means of aggregation.
 - E.g., given total sales for each city, find total sales for each state.
- **Drill-down:** The inverse of roll-up
- **Pivoting:** aggregate on selected dimensions
- **Slicing and dicing:**
 - E.g., from the data cube find the cross-tab on Model and Color for **medium** cars . The cross-tab can be viewed as a slice of the data cube.

Query Processing Issues

- Expensive aggregations are common
- Pre-compute all aggregates? Maybe infeasible!
- Materialized views can help.
 - Which views to materialize?
 - given a query and some materialized views, can we use the views to answer the query? How?
 - How frequently should we refresh the views to make them consistent with the underlying tables?
- What indexes should one use?

SQL:1999 Extended Aggregations*

Example 1

Select make, color, size, sum(number) from sales

group by cube(make, color, size)

Calculates 8 groupings:

(make, color, size), (make, color), (make, size), ..., ().

Example 2

Select make, color, sum(number) from sales

Group by rollup(make, color, size)

Calculates 4 groupings:

(make, color, size), (make, color), (make), ().

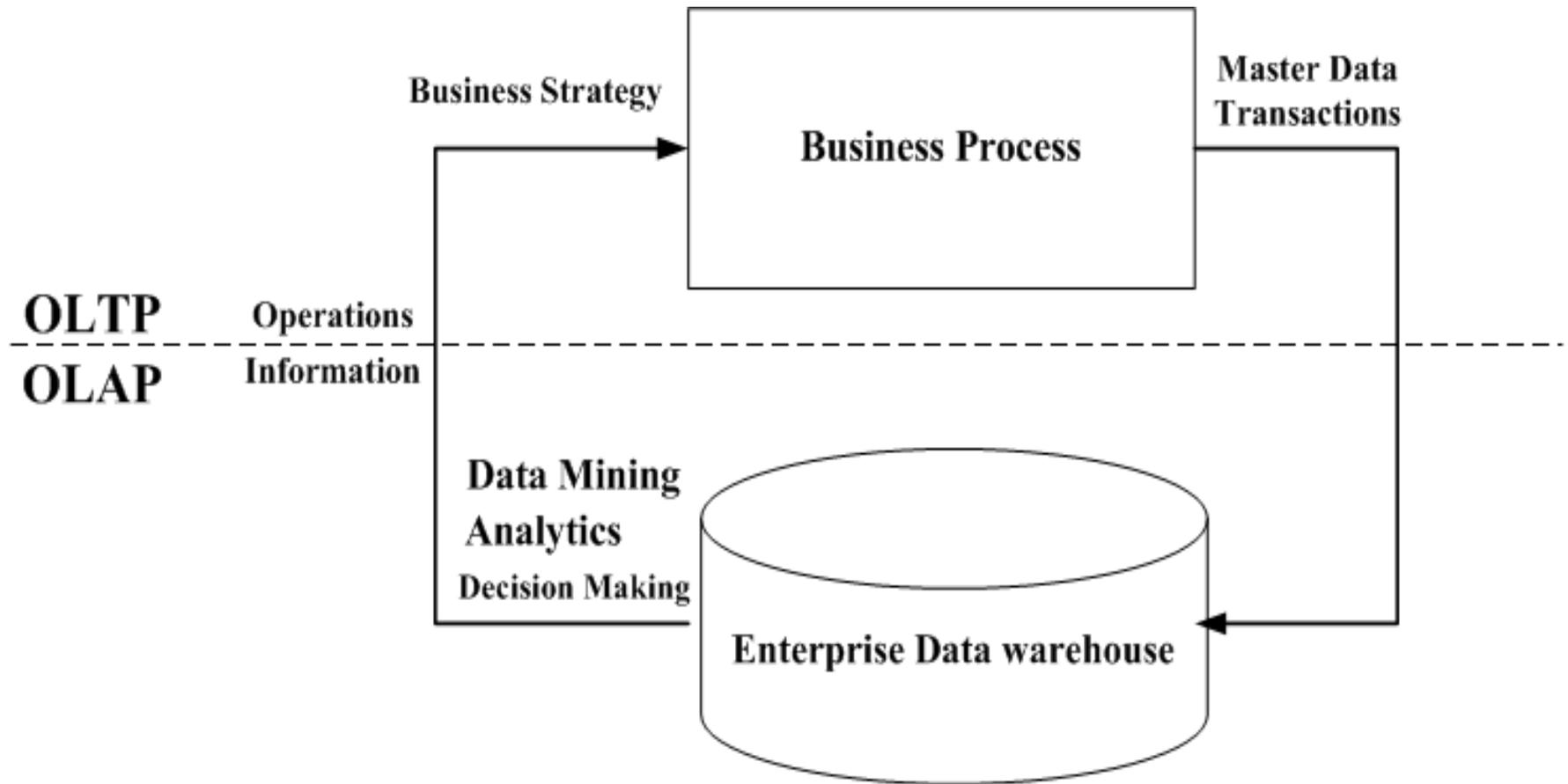
Examples in Oracle: Rollup

```
SELECT channels.channel_desc, calendar_month_desc,  
       countries.country_iso_code,  
       TO_CHAR(SUM(amount_sold), '9,999,999,999') SALE$$  
FROM sales, customers, times, channels, countries  
WHERE sales.time_id=times.time_id  
      AND sales.cust_id=customers.cust_id  
      AND customers.country_id = countries.country_id  
      AND sales.channel_id = channels.channel_id  
      AND channels.channel_desc IN ('Direct Sales', 'Internet')  
      AND times.calendar_month_desc IN ('2000-09', '2000-10')  
      AND countries.country_iso_code IN ('GB', 'US')  
GROUP BY  
      ROLLUP(channels.channel_desc, calendar_month_desc, countries.country_iso_code);
```

Oracle Rollup Example

CHANNEL_DESC	CALENDAR	CO	SALES\$
Internet	2000-09	GB	16,569
Internet	2000-09	US	124,224
Internet	2000-09		140,793
Internet	2000-10	GB	14,539
Internet	2000-10	US	137,054
Internet	2000-10		151,593
Internet			292,387
Direct Sales	2000-09	GB	85,223
Direct Sales	2000-09	US	638,201
Direct Sales	2000-09		723,424
Direct Sales	2000-10	GB	91,925
Direct Sales	2000-10	US	682,297
Direct Sales	2000-10		774,222
Direct Sales			1,497,646
			1,790,032

OLTP and OLAP



Should OLAP be Performed Directly on Operational Databases?

- OLTP systems support multiple concurrent transactions. Therefore the OLTP systems have support for concurrency control (locking) and recovery mechanisms (logging).
- An OLAP system on the other hand requires mostly a read only access to data records for summarization and aggregation. If concurrency control and recovery mechanisms are applied for such OLAP operations, it will severely impact the throughput of an OLAP system.

OLAP Operations on Multi-dimensional Data

- Slice
- Dice
- Roll-up
- Drill down
- Drill through
- Drill across
- Pivot/Rotate

Do It Exercise

Hands on practice on the various OLAP operations on multi-dimensional data.

Hint: Provide the participants with a sample data sheet (Excel sheet) and ask them to demonstrate their understanding of the various OLAP operations on multi-dimensional data.

Data Warehouse

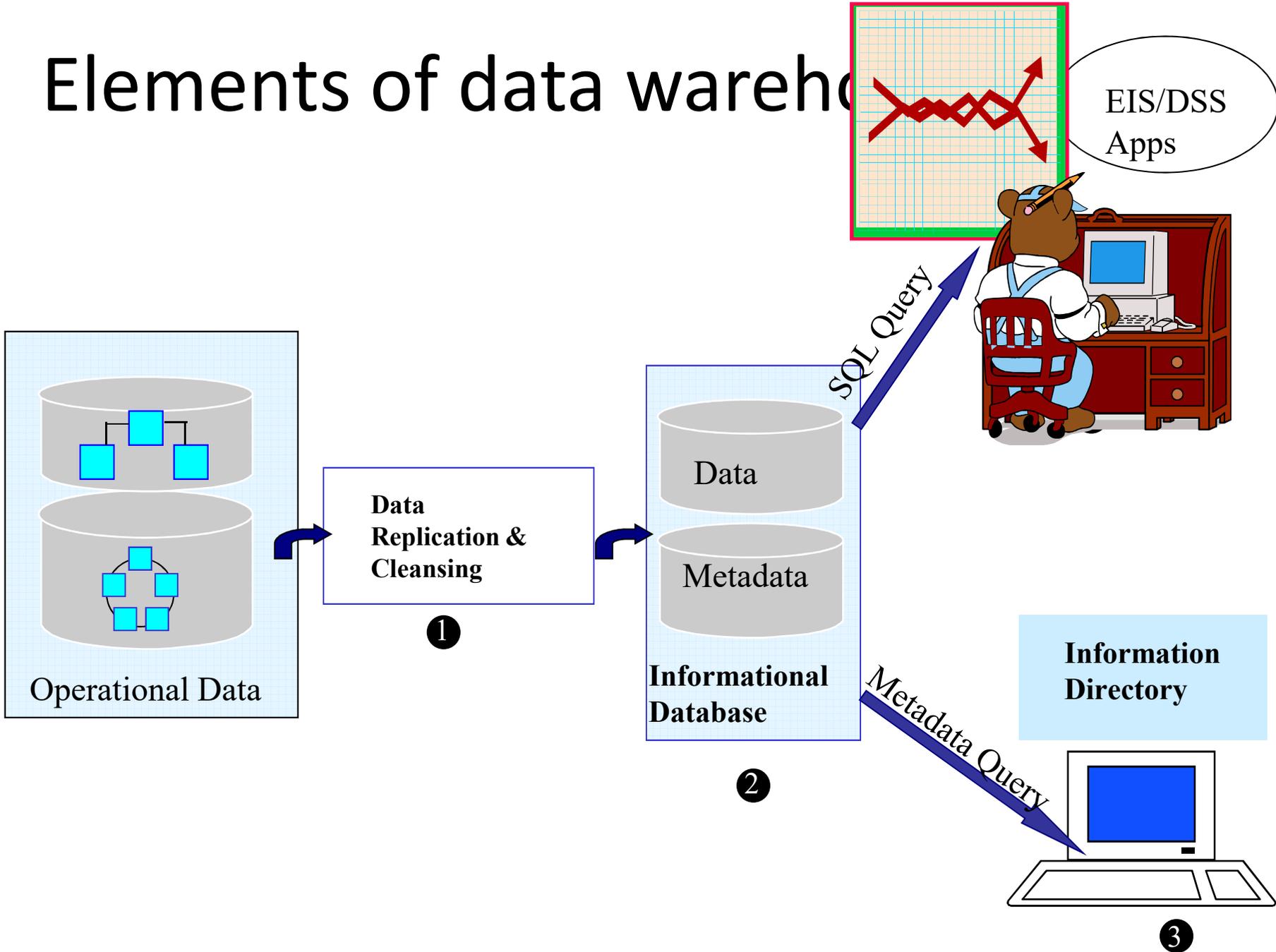
- A repository of information gathered from multiple sources, stored under a unified schema, usually at a single site .
- Data may be augmented with additional attributes, such as timestamp, and summary information.
- Data are stored for a long time, permitting access to historical data.
- Interactive response times expected for complex queries; ad-hoc updates uncommon.

Building Data Warehouse

■ Issues:

- Semantic integration: When getting data from multiple sources, must eliminate mismatches, e.g., different currencies.
- Heterogeneous sources: must access data from a variety of source formats.
- Load, refresh, purge: Must load data, periodically refresh it, and purge too old or useless data
- Metadata management: Must keep track of source, loading time, etc.

Elements of data warehouse



Elements of data warehouse

- Data Replication Manager
 - copying & distribution of data across databases
 - data that needs to be copied, source/destination, frequency, data transforms
 - *refresh* copy entire source, propagate changes only
 - all external data is transformed & cleansed before adding to warehouse
- Informational Database
 - database that stores data copied from multiple sources by data replication manager
- Information Directory
 - metadata manager - collects metadata from databases on network
- EIS/DSS tools
 - SQL based query tools
 - some vendors use extended SQL

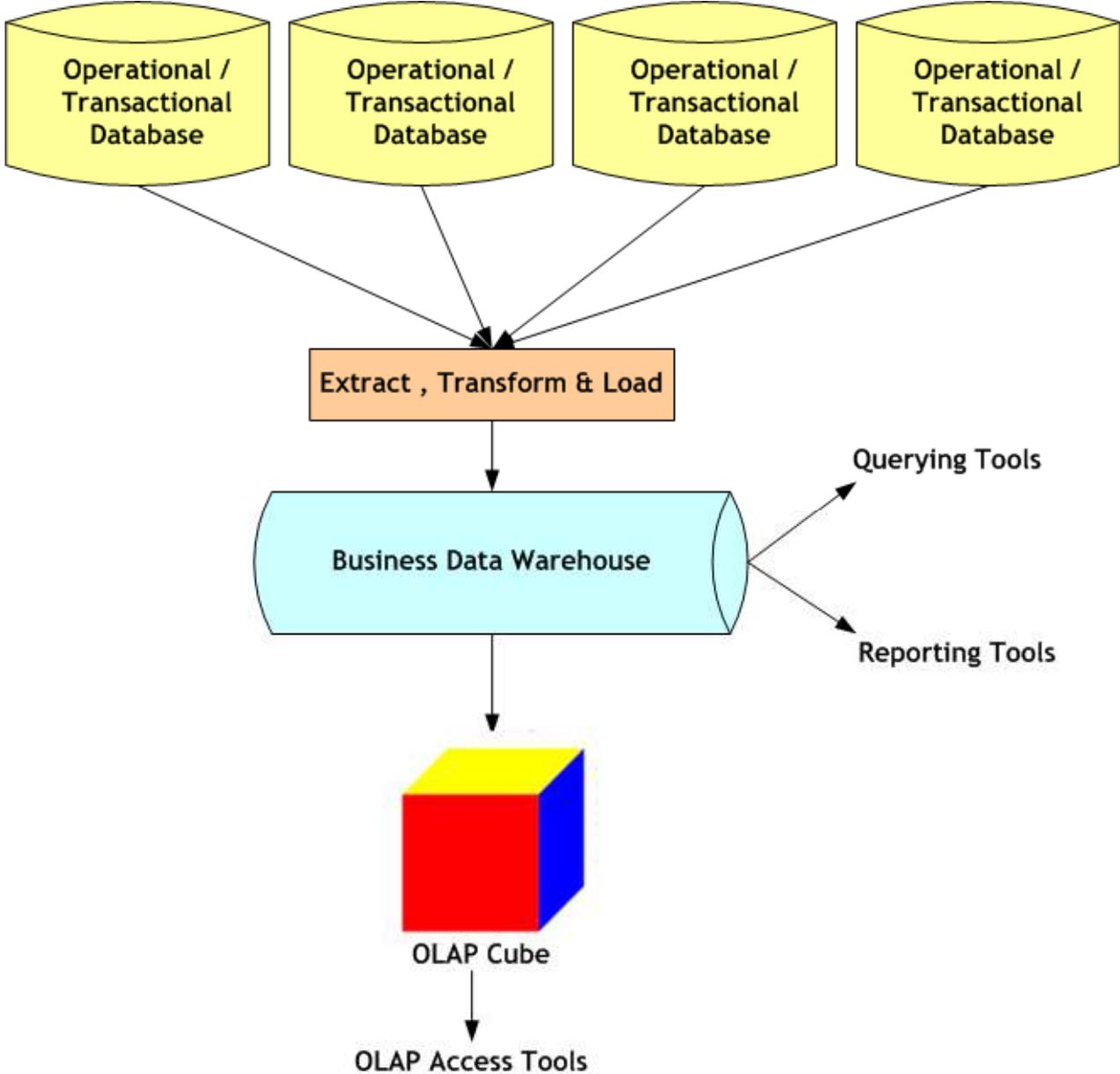
Query/Reporting tools

- Formulate queries without (extended) SQL or other languages
- Result displayed as table, graph, report,
- Spreadsheet systems
- Web interfaces
- Vendor-specific tools
 - Oracle Discoverer:
 - <http://www.oracle.com/tools/disc/index.html>

Column stores

- A recently proposed data storage method that allows more efficient aggregation queries in data warehouses
- stores data as columns rather than as rows.
- See http://en.wikipedia.org/wiki/Column-oriented_DBMS.

OLAP in BI



Answer a Quick Question

Will using BI/Analytics in conjunction with ERP systems prove advantageous to the enterprise? Why?

Leveraging ERP Data Using Analytics

ERP provides several business benefits, here we enumerate the top three:

1. Consistency and reliability of data across the various units of the organization.
2. Streamlining the transactional process.
3. A few basic reports to serve the operational (day-to-day) needs.

In short ERP systems are adept at capturing, storing and moving the data across the various units smoothly.

It is however inept at serving the analytical and reporting needs of the organization.