



Table of Contents

1. Introduction to Data Warehousing
2. Data Warehouse Architectures
3. The DW Program
4. Gathering Information
5. Dimensional Modeling
6. The Central Data Warehouse (Store)
7. Extract-Transformation-Load (Gather)
8. Data Marts (Deliver)
9. Data Quality
10. Metadata
11. Trends in Data Warehousing
12. Summary





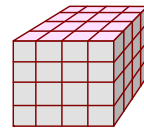
Objectives of the Course

- Upon completion of this course, the student will have the an understanding of the following :
 - What is a data warehouse?
 - The difference between operational and analytical data
 - The difference between a data warehouse and a data mart
 - Key issues and concerns in data warehouse projects
 - A general understanding of the different topologies for designing data warehouses
 - The types of questions asked to drive information gathering in data warehousing
 - The magnitude of and key issues in moving data from source systems
 - The key significance of data quality in data warehousing
 - What is dimensional modeling?
 - What is the central data warehouse?
 - The importance of metadata
 - Major trends in data warehousing today
 - Most of the major data warehousing terms and acronyms



Introduction

-
-
-
-
-
-
-



True Story

- Two CEO's agree to meet at a football game
- Both come armed with information on how much business they are doing with each other
 - Each has different numbers
 - Each suspects that the others' numbers are better than theirs
 - Teams of analysts worked weeks coming up with these numbers
- Both left this experience with the decision to build a solution that would prevent history from repeating itself



Recognizing the Need for Information

- Reports from multiple operational systems do not agree
 - Sales figures don't tally
 - Financial figures don't match
 - Detail reports don't add up to summary reports
- Management has trouble accessing a corporate wide picture:
 - How many customers do we have?
 - Why do customers buy our products vs. our competitors?
 - Can our best customers contact us effectively and efficiently?
 - Can we anticipate a customer's needs accurately enough to be in the Right place, at the Right time, with the Right product?
 - Are we losing money?



The Dilemma of IT

- IT is overloaded with requests for management reporting.
- Apparently with no other choice, business units hire IT contractors to create reports for them.
 - Data specialists have popped up all over the corporation
 - Data sharing is rare - stovepipe databases are the norm.
 - Collecting and integrating the information for reports takes - longer and longer
 - "Spreadmarts" pop up everywhere!
- Today's difficult economic times call for substantiated, rational strategic decisions
 - What stores do we open (or close)?
 - Do we discontinue one of our products?
 - Should we pay certain customer to leave?
 - Where can we cut costs without risking customer dissatisfaction?





Major Trends Today



■ Data Warehouse

- ▶ An environment for gathering detailed informational data, storing it over time in a general purpose data base, and delivering it as information to other business and system users.

■ Data Mart

- ▶ An environment containing a specialized collection of related data, customized for a specific community of knowledge workers, analysts or planners, to support their reporting and analysis needs.

■ The Operational Data Store

- ▶ A tactical environment which stores detailed, near-real time results of committed transactions for a certain period of time for immediate reporting needs and which can sometimes be updated by users.

■ Business Intelligence

- ▶ A class of technology designed specifically to support management reporting and analysis

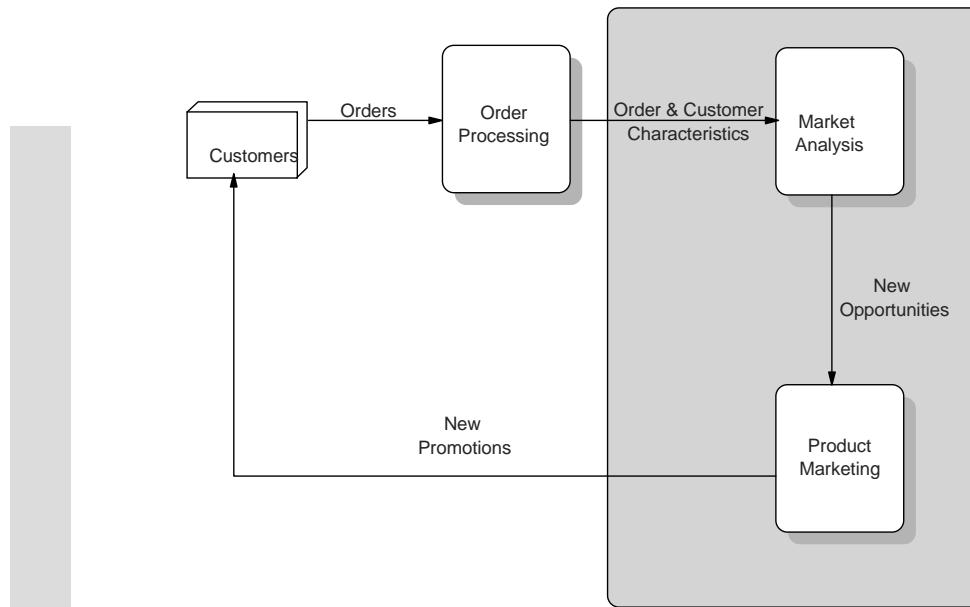
■ Data Mining

- ▶ The use of algorithms to discover hidden relationships in apparently unrelated data

■ Big Data

- ▶ The application of analytics to data of great volume, variety, velocity and value

Where Is The Greatest Opportunity?



- A Data Warehouse makes good business sense



How Business Intelligence Helps

- Business Intelligence allows management to:
 - ▶ Monitor the financial and operational performance of the organization
 - Reports, trends, alerts, scorecards, analysis tools, key performance Indicators (KPIs) and dashboards
 - ▶ Regulate the operation of the organization
 - Bi-directional integration with operational systems, thereby providing information feedback
 - Information, without the ability to act on it, is futile



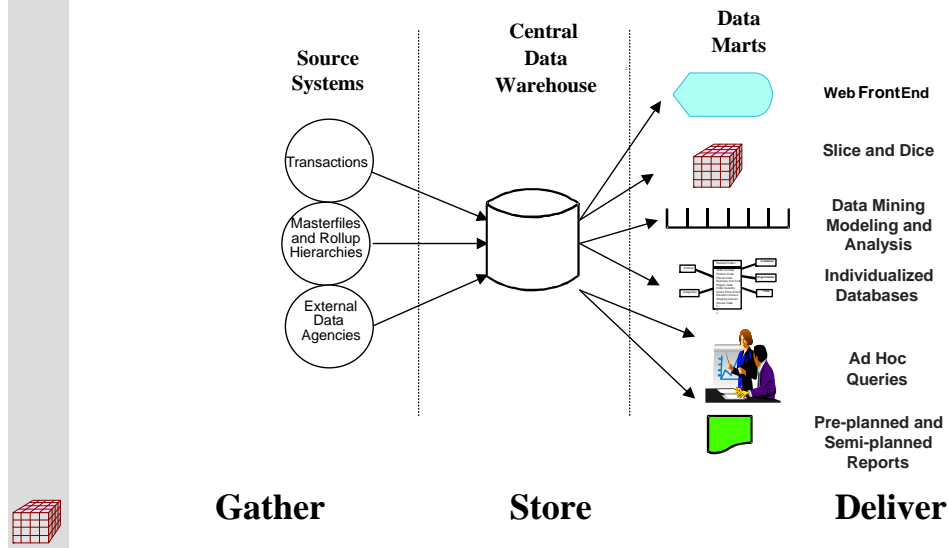
Challenges In Data Management

- End users need a consistent view of corporate data
- Data today is dispersed across many different systems and stored in a variety of formats
- 85% of computerized data is in a non-relational format
- Legacy data bases and files are replete with inconsistencies
 - Field names
 - Data currency
 - Data meanings
- Most corporate data is not suitable for end-user consumption
 - Hidden codes
 - Complex data formats (repeating data, multi-use record types, etc.)
 - Inadequate summaries or data histories



What Does A Data Warehouse Look Like?

- A data warehouse is an environment and system for gathering data from multiple sources, enhancing and storing the data in a integrated database, and delivering information to business people.





Typical BI Applications

- **Sales and Marketing** - understand customer needs and respond to new market opportunities, measuring the effects of pricing and promotions, analyzing buying behaviors to cross-sell, up-sell and to target customer segments
- **Product Development** - access to critical information on customers, markets and suppliers to determine solid cost-benefits of product features, components and functionality
- **Operations** - a means to measure quality control, inventory management and production planning performance
- **Customer Service** - accurately assess value of broad markets and individual customers, as well as implement retention strategies for most profitable customers or those most likely to leave



Success Statistics

- So, how effective has all this expense been?
- IDC* studied analytic applications and their impact on core business processes:
 - Marketing campaign
 - Fraud detection
 - Portfolio management
- What was the ROI for Business Analytics?
 - 46% of organizations generated ROI < 100%
 - 34% generated ROI between 101 and 1,000%
 - 20% generated ROI > 1,000%
- Value accrued through increased business performance to reduced operations to improved customer relations



* "The Financial Impact of Business Analytics", IDC, www.idc.com/analytics/



Characteristics Of A Warehouse

- A data warehouse is a:
 - Management-oriented
 - Subject-oriented
 - Integrated
 - Historical (time-variant)
 - Read-only (non-volatile)
 - ***Controlled***
- Collection of data in support of management's decision making process



Operational Vs. Informational Data

Operational Data

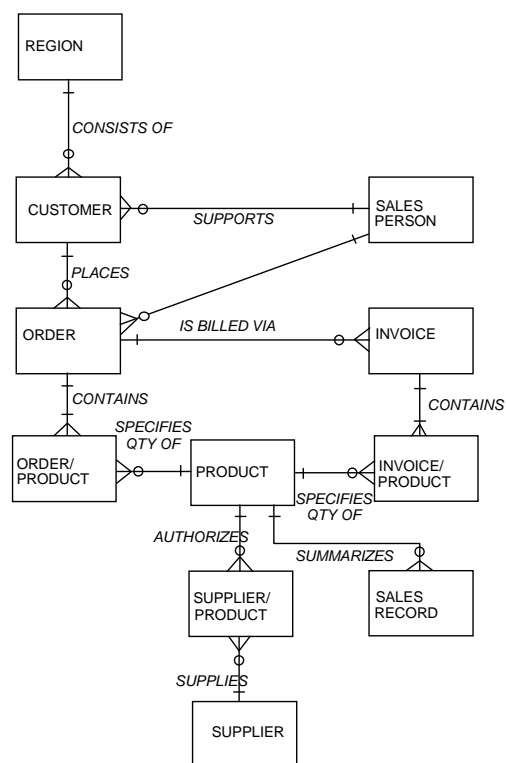
- detailed
- normalized
- current
- volatile
- transaction processing
- performance sensitive
- high availability
- critical to daily operations

Informational Data

- detailed/summarized
- normalized/unnormalized
- historical
- mostly read-only
- transaction analysis
- performance sensitive, though not as much
- usually mission-critical
- used to analyze operations



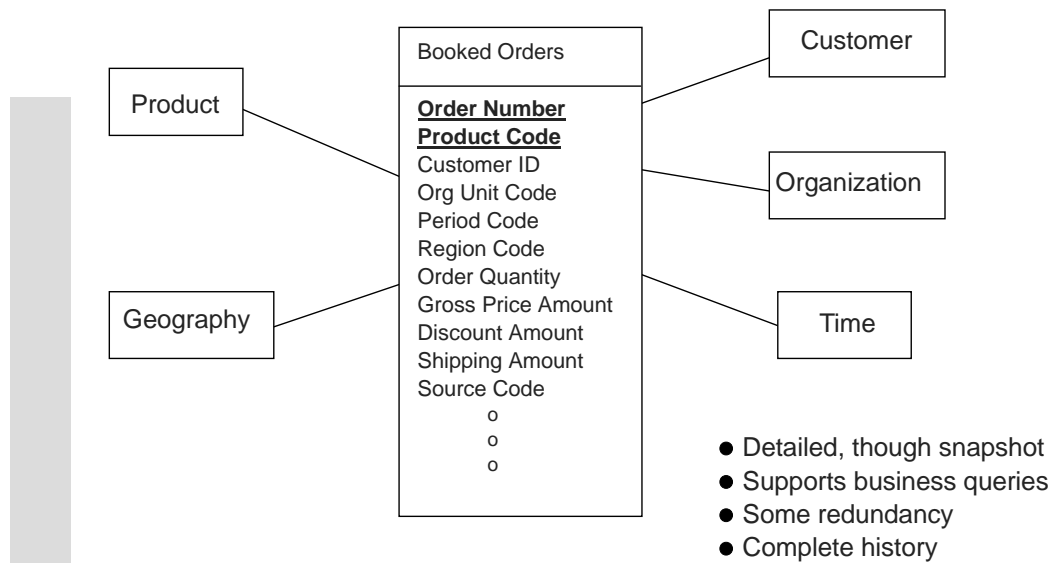
An Operational Model



- Detailed
- Supports business rules
- Non-redundant
- Limited data retention



An Analytical Model

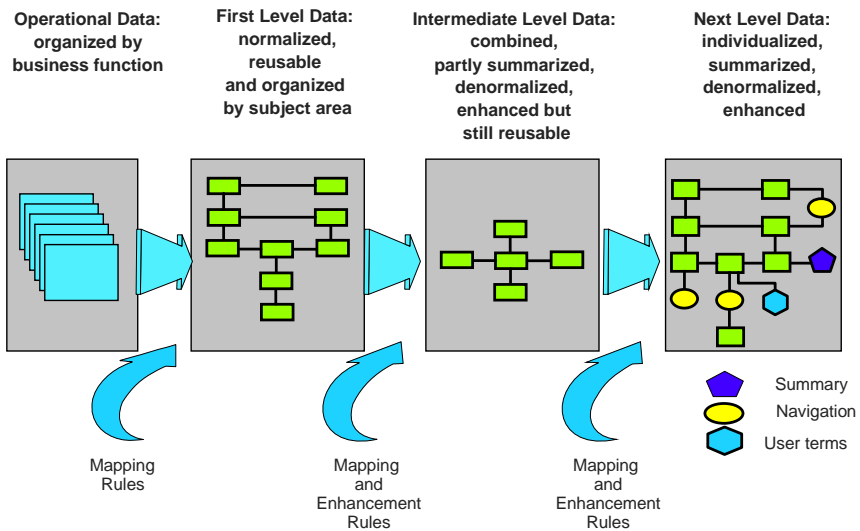


Strategic Versus Tactical Questions

♦ Strategic Questions	♦ Tactical Questions
How many orders, by product, were shipped more than 30 days late over the past year?	What backorders from the 30 day backlog are scheduled to be shipped today?
Is there a pattern, based on average household income, of students who default on loans?	What is the list of names and the associated addresses of the students currently more than 30 days late on their loans? (For mailing overdue notices)
What color car has typically sold the best by sales region over the last 5 years?	What other dealers have a white Product XYZ in stock right now?
What is the seasonal pattern in sales by model, by geographic area?	Is my dealership on target for this month compared to my objectives?



Levels of Data In the Warehouse



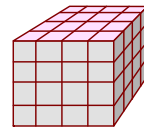
Terms

- Data warehouse
- Gather-Store-Deliver
- Business intelligence
- Data mining
- Characteristics of a DW
- Central data warehouse
- Data mart



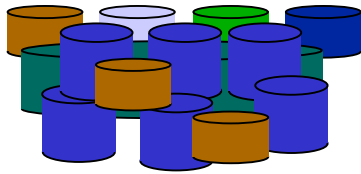
Data Warehouse Architectures

-
-
-
-
-
-
-



Factors Affecting Design

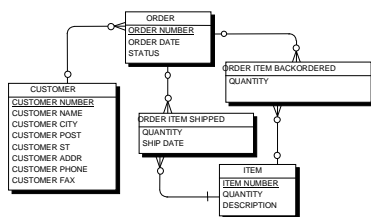
Amount of Detailed Data



Concurrent Users



Complexity of Data Model

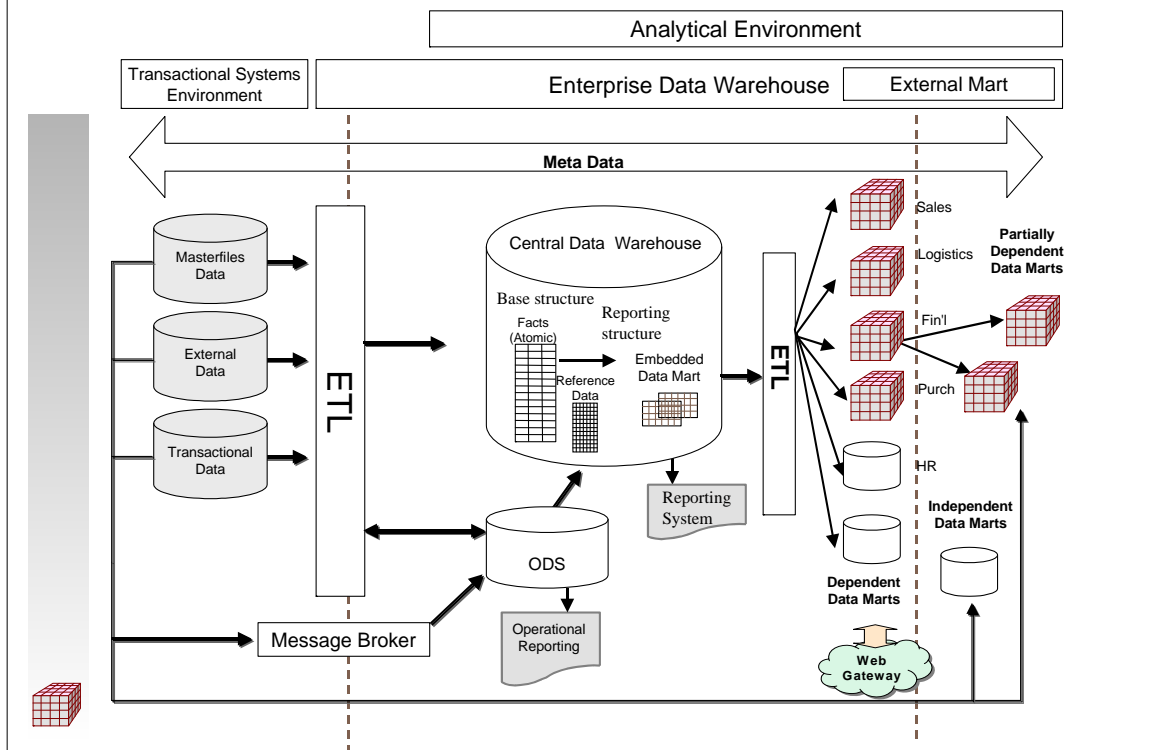


* Source: NCR

Query Complexity

- Simple Direct at the start
- Moderate Multi-table Join
- Regression analysis
- Query tool support
- Complex, 58-way table join
- 15 Pages, 37 From Clauses, 7 UNION's
- , (Largest table >1 B rows, < 4 minutes)

Architecture of a Data Warehouse



Data Warehouse

Perspective: Cross-functional, even Enterprise-wide
Supporting multiple major business management processes
Never complete because constantly enhanced.

Build time: Best delivered in short releases

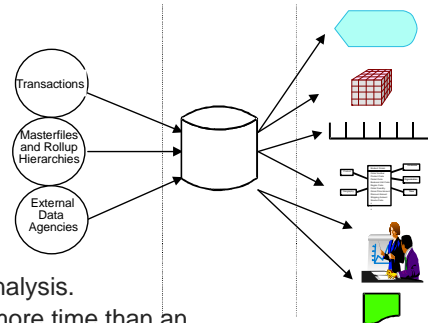
Cost: Millions of dollars

Advantage:

- A single version of the truth across the corporation.
- Consistent data across the entire business.
- Supports detailed and high level decision support analysis.
- If delivered incrementally, requires only marginally more time than an equivalent function data mart

Disadvantage:

- Project can be large and complex if not incrementally delivered.
- Must coordinate multiple sources, vendors and products.



Data Mart

Perspective: A specialized collection of related data for a particular community of business users

Build time: Built incrementally

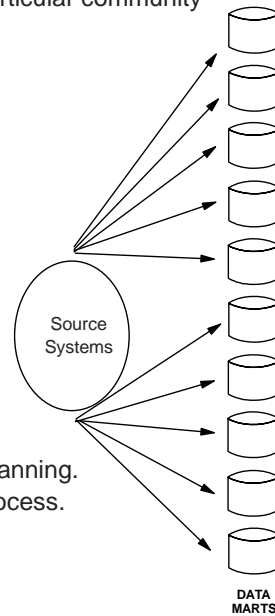
Cost: Average cost is \$2MM

Advantages:

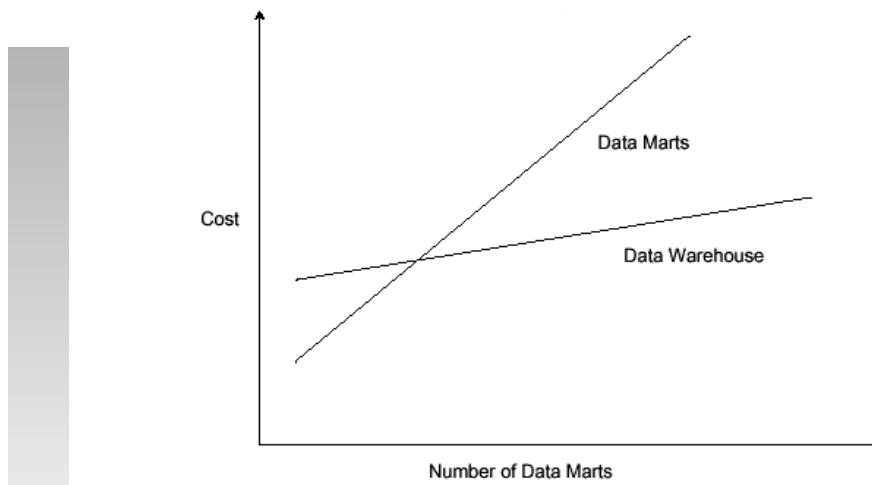
- Addresses a specific audience.
- Easier, cheaper and quicker to build (initially).
- Turnkey systems available.

Disadvantages:

- Future growth limited without data warehouse integration planning.
- Adding more data marts means repeat of transformation process.
- Cross-functional reporting requires extra work.
- More and more marts become necessary.



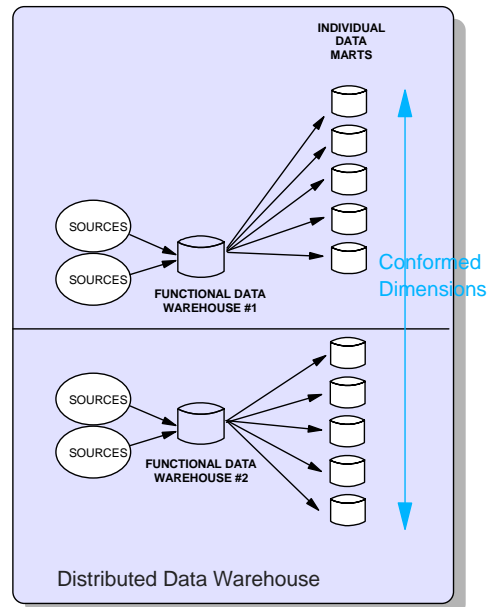
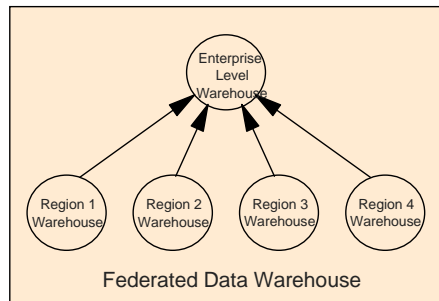
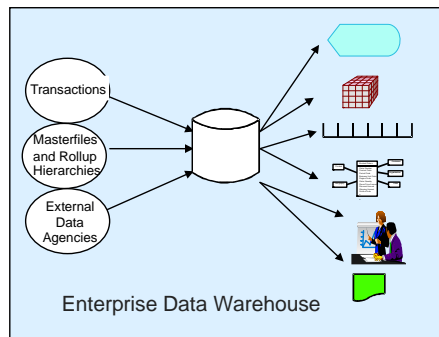
▼ Cost of Implementation



* Source: Gartner



Primary Topology Choices



Query Data Access

- 80% at Central Data Warehouse and 20% at Data Marts or Summaries (Aggregates)

Versus

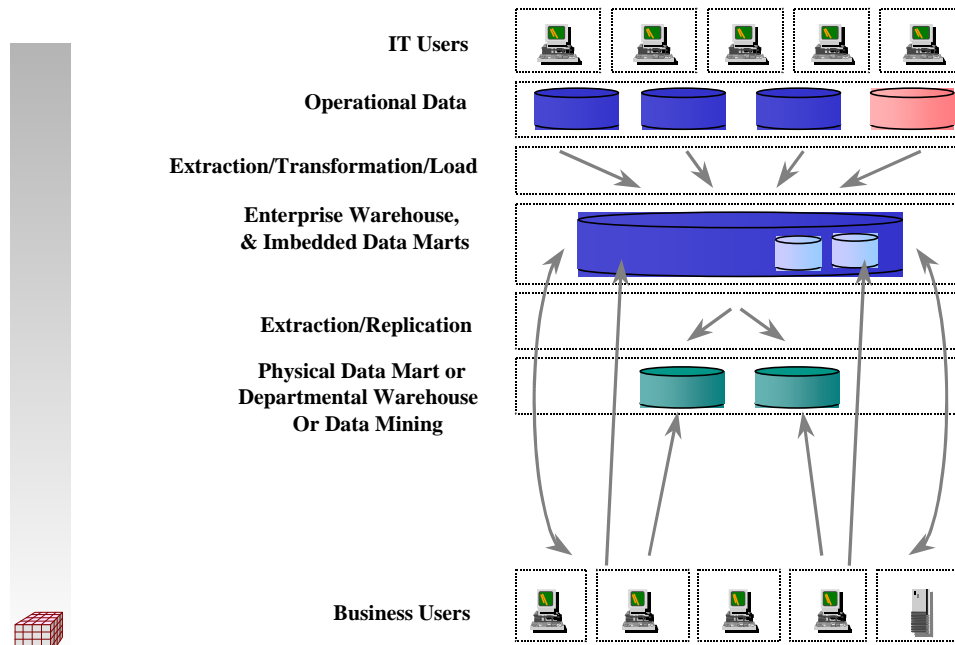
- 20% at Central Data Warehouse and 80% at Data Marts or Summaries (Aggregates)

- *The choice will appreciably affect the data warehouse architecture*
- *The choice is significantly affected by capabilities of underlying platform*



1. A Centralized Data Warehouse

"Reports of my death have been greatly exaggerated."
- Mark Twain



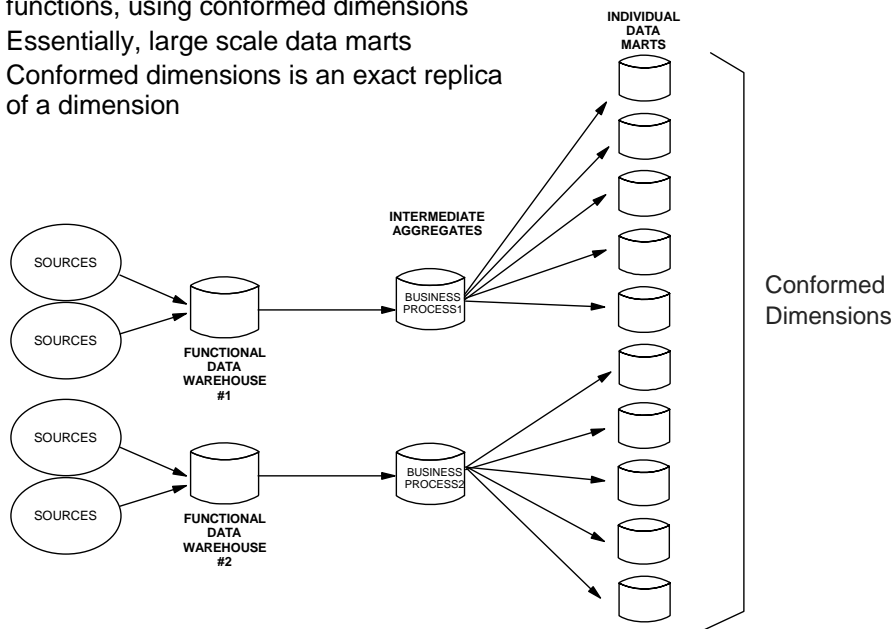
Enterprise Data Warehouse Successes

- The enterprise data warehouse approach has witnessed the most dramatic successes in data warehouse history:
 - ▶ Walmart
 - ▶ K-Mart
 - ▶ 3M
 - ▶ Medco, etc.
- "Enterprise" means major logical or legal subdivision of the business.
- The current trend is away from decentralized data warehouses (i.e., functionally separate, federated) back to more centralized because:
 - ▶ Technology is now capable
 - ▶ Methods have matured
 - ▶ Problems with data integration across dispersed data marts



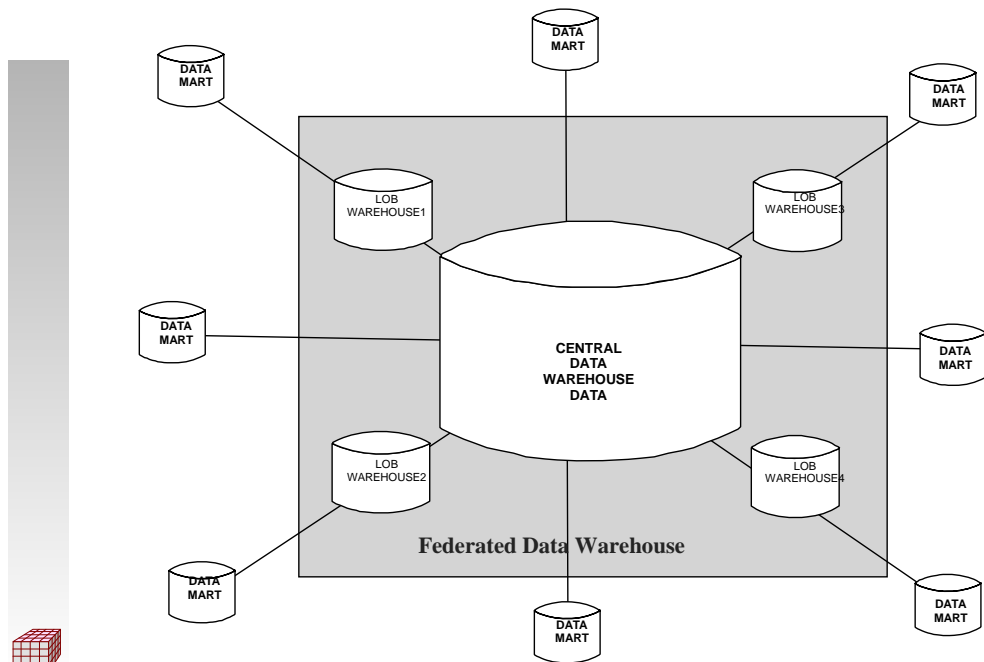
2. Distributed Data Warehouse

- Multiple data marts, supporting different business functions, using conformed dimensions
- Essentially, large scale data marts
- Conformed dimensions is an exact replica of a dimension

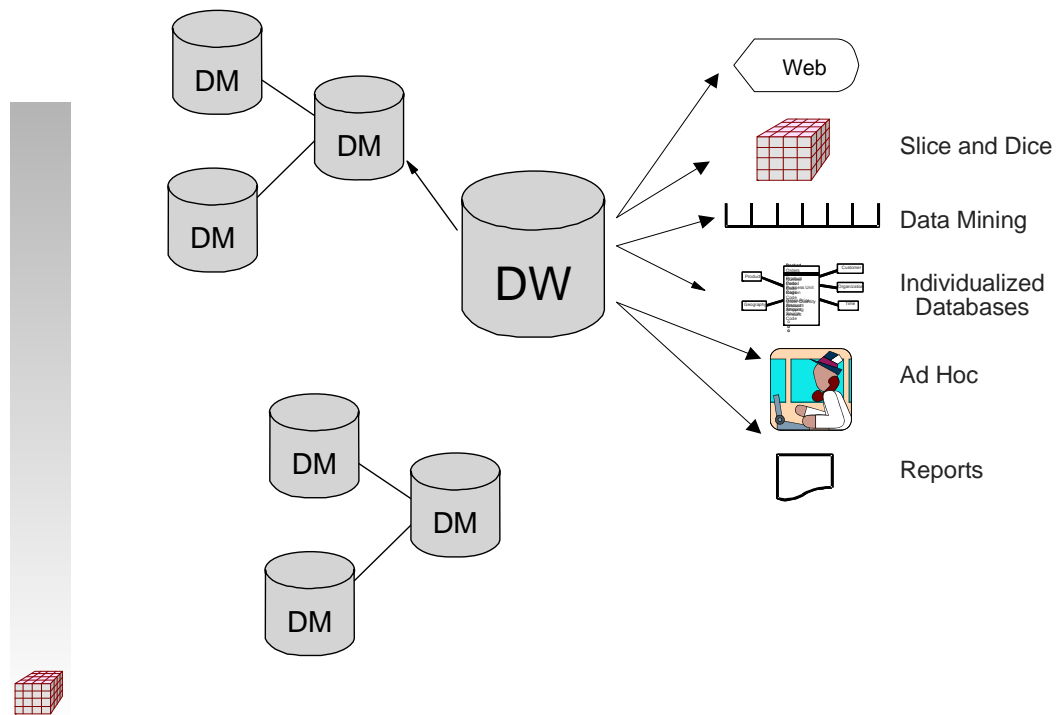


3. Federated Data Warehouse

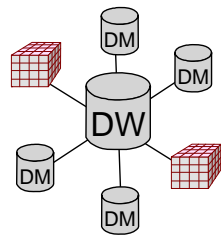
Multiple data warehouses, each supporting separate business functions, but loosely coupled and feeding a corporate data warehouse



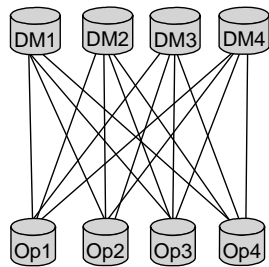
4. Hybrid



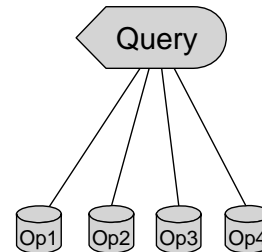
▼ Poor DW Topology Choices



No User Access to DW



Data Mart Only



Access Only to Operational Data

Summary

- Not all architectures are equal
- Choose the architecture that suits your requirements
- Design for success
- Don't be fooled by self-serving literature
- The more cross-functional your requirements are, the more centralized your data warehouse should be



Key Terms

- Data Warehouse
- Data Mart
- DW Topology
- Centralized Data Warehouse
- Distributed Data Warehouse
- Federated Data Warehouse
- Hybrid
- Hub and Spoke Data Warehouse
- Virtual Data Warehouse
- Incremental Delivery
- 80/20 Rule in DW Data Access





Some DW Architecture Definitions

■ Enterprise DW

- ▶ A centralized DW environment covering, within a single central DW database, the data necessary to support the reporting needs of an organization. The term "enterprise" can cover a logical or legal subdivision of the business. A single organization could conceivably have several "enterprise" DW's. This DW, like all others, is built incrementally.

■ Distributed DW

- ▶ A DW architecture consisting of multiple separate data warehouses, each supporting different business functions. Essentially, large scale data marts.

■ Federated DW

- ▶ A DW architecture in which several separate DW's or data marts are actively interrelated, collectively composing the data warehouse of a single organization. Can be federated through technology (DBMS), exchanging data, or using conformed dimensions and (where needed) conformed facts.

■ Hybrid DW

- ▶ A DW architecture containing a mix of the other DW architecture types. Realistically, all DW's will be hybrids, though the dominant architecture should be one of the three just described.

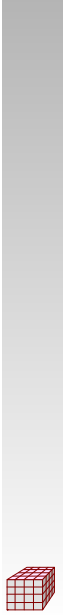
■ Virtual DW

- ▶ A DW environment which utilizes data strictly from operational and similar sources. Though some consultants are promoting this model, it is not possible to achieve this in any real sense due to the inconsistency, limitations and heterogeneity of existing operational systems. Also, existing operation data will not contain all the rollup hierarchies, external data and modeled data necessary for a DW.



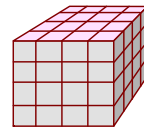
Exercise

- Do case study Exercise 1



The Data Warehouse Program

-
-
-
-
-
-
-



Principle

- "Think big and with a view to the future, but implement incrementally"





Establish the Perspective of the DW

- What business are we in?
- What business will we be in?
- What business should we be in?
 - Peter Drucker





The Data Warehouse Program

- Greatest success is achieved when an organization establishes a data warehouse program
 - Define a comprehensive long term vision
 - Identify a process of multiple releases spanning years
 - Identify projects within the program within the vision
 - Deliver the first project in 6-9 months
 - Deliver the second in 6 months or less
 - Plan the remainder for quarterly releases



Components of DW Program (Strategy)

- The vision
- Cost-benefit justification
- Implementation plan, including
 - Project Justification Process
 - Composite View of all Releases

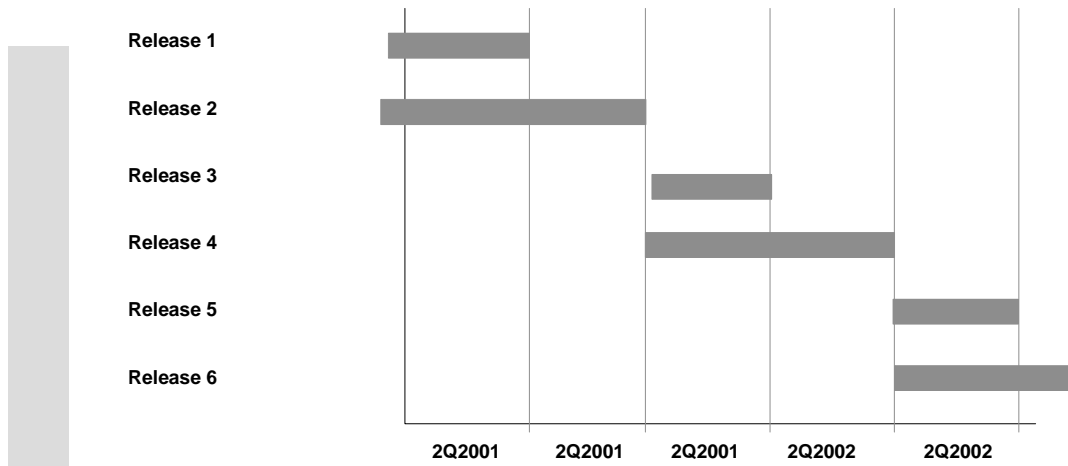
	Business Process	Information	Application	Technology	(BIAT)
Define Future State					
Assess Current State					
Identify Gaps					
Formulate Plan					





Impact of the Data Warehouse Program

- Provides a concrete schedule that emerges release-to-release



Project Types

- The plan supports three types of projects:

- Growth Projects - Offering new subjects areas, capabilities and deliverables
- Leverage Projects - Supporting a new set of business users from the common database
- Technology - Implementing new tools (e.g., a new BI tool) or infrastructure products (e.g., the web)





The RAD Method

RAD stands for Rapid Application Development

A data warehouse (or any decision support project) should be done as follows:

- Break the project into increments
- Make the increments short
- Deliver a prototype 1/3 of the way through
- Timebox the work
- Assemble a strong cross-functional team

This is represented in the following steps:

- 
- 
1. Get the business requirement first
 2. Develop the logical data warehouse design
 3. Prototype it
 4. Repeat 1-3 until the business people really understand what they are going to get!



Generic Project Steps - RAD Method

- Initiation
 - Scope and timebox definition
 - Parallel planning, startup, requirements
 - Create data model
 - Ends with install of live DB with base data
- Testing
 - Consumer team explores usability
 - ETL team tests data content and quality
 - Data team iteratively tests physical database
 - Ends when time-boxed targets are reached
- Implementation
 - Traditional production rollout
 - User training and preparation
 - Ends with install of new release



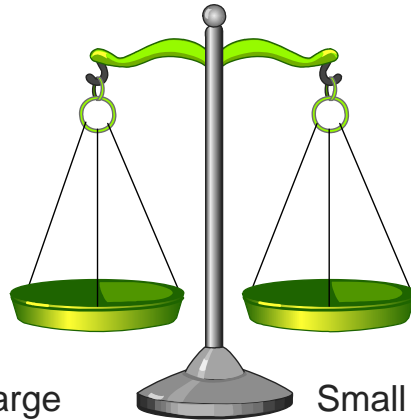
RAD = Rapid Application Development

Key Project Learnings

- Define scope incrementally using strict time-boxed measures
- Seek to capture information needs, not concrete requirements
- Use information needs to collect the most granular sources
- Collect data wider and deeper than needs initially indicate
- Implement a comprehensive live database 1/3 of the way through
- Train all participants on how to use the data and the access tools
- Involve users actively during prototyping
- Ensure appropriate data quality
- Develop a detailed, pragmatic operations plan



▼ Scope Balancing Act



Large
enough to
provide
meaningful
business
benefits

Small
enough to
be
achievable
in a short
time frame



Team Mix

- A large team is not a requirement.
 - A small team of 3 succeeded.
 - A large team of 30+ failed badly.
 - The average initial project size is 7 FTE.
- Smaller is better given:
 - A good mix of broad skilled individuals
 - A core of relevant experience (even if in one person)
 - A hands-on manager with a focus on deliverables
- The key to a successful data warehouse is the right mix of people with the previous four new skills.
- The focus is on small, quick, cross-functional increments.

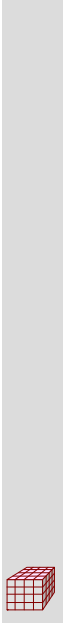


Terms

- Data warehouse program
- BIAT
- RAD
- Iterative method
- DW increment

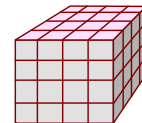


Notes



Information Gathering

-
-
-
-
-
-
-
-



Two Main Requirements

- To do any analysis, the two main requirements are:
 - ▶ Business knowledge - otherwise, you can only get the requirements right by accident
 - ▶ Business decision making - where a clear business rule does not exist, people are needed who can make decisions about what should happen, and have the authority to do it



Main Focus Of DW Requirements

- The main methods of information gathering are:
 - ▶ Goals (so you can measure achievement of them)
 - ▶ Top 10 questions (how much is the answer worth to them? This will provide the tangible benefit of the answer.)
 - ▶ Important business events (so they can make them happen or prevent them from happening)
 - ▶ Business scenarios (complete business situations)
 - ▶ Broad information requirements (brainstorming is useful for this)
- The DW must satisfy many requirements, so get the business people to discuss them in their own contexts as much as possible



Example of Goals

- A company identifies three main business goals to be enabled by their data warehouse:

- ▶ Reduce attrition
 - To stop a major loss of customers and accounts
- ▶ Increase customer product involvement
 - The average customer owned 1.23 products
- ▶ Increase cross-sell and up-sell
 - Create combo products with price reductions





Questions To Ask the Business

- ◆ What **questions** do they need answers to?
 - Example: *What was our revenue and volume from sales last month compared to the same period last year?*
- ◆ What do they use to **measure** the business?
 - Example: *How many new customers did we have last month? How many did we lose to attrition?*
- ◆ What are the main **business scenarios**?
 - Example: *We need to be able to restate sales over two-year periods to adjust for changes in territories.*
 - Example: *How many customers "likely-to-leave" were we able to keep?*
- ◆ What **decisions** do they make?
 - Example: *What are the demographics of our most successful stores? Should we position our stores nearby our competitors stores?*
- ◆ What **problems** do they solve?
 - Example: *We spend a lot on disability. How can we predict when a disability event is likely to happen so as to prevent it?*
- ◆ If you ask the business for typical questions, they may balk. Then ask for some sample questions.
- ◆ When gathering this data, always consider not only what information they are currently using, but what other uses they could make of the data



Making Goals Tangible

- Marketing Capture 20% of market share in two years by increasing volume and reducing price on high margin products.
- Sales Develop customer specific promotions to achieve 30% increase in sales this year.
- Distribution Re-engineer the order process to ship 90% of all orders the day they are taken.
- As in the above examples, goals should contain:
 - Measure
 - Level
 - Time



Other Sources of Requirements

- Current reports because they are still needed
- Reports that can not be generated
- Spreadsheets in use and that need to be improved
 - Sometimes there are dozens of them
 - These are sometimes called "spreadmarts"
 - Seek to incorporate them into the DW
- Report requests submitted to IT (this is the backlog)
- Current EIS or DSS implementations (usually stand-alone)
- External data sources
- PC databases for reporting purposes
 - Often very useful and successful MS Access or Excel applications have been created by the business groups themselves
 - Seek to incorporate them into the DW
- User management meetings

DSS = Decision Support System
EIS = Executive Information System



Information BY Lines

- A convenient way to discover the facts and dimensions is to work with queries or other known requirements
- For each query, identify:
 - The measures needed (facts)
 - *BY* what characteristics (dimensions)
- For example,
 - "We need monthly total dollar sales by region, district and territory."

Fact (Measure)**Dimension (BY)**

Total dollar sales

Region No, District No,
Territory No, Period

Terms

- Goals
- Top 10 questions
- Measures



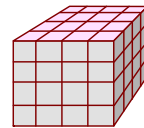
Exercise 2 and 3

- Do Exercise 2: Query Analysis #1
- Review Exercise 2
- Then do exercise 3: Query Analysis #2



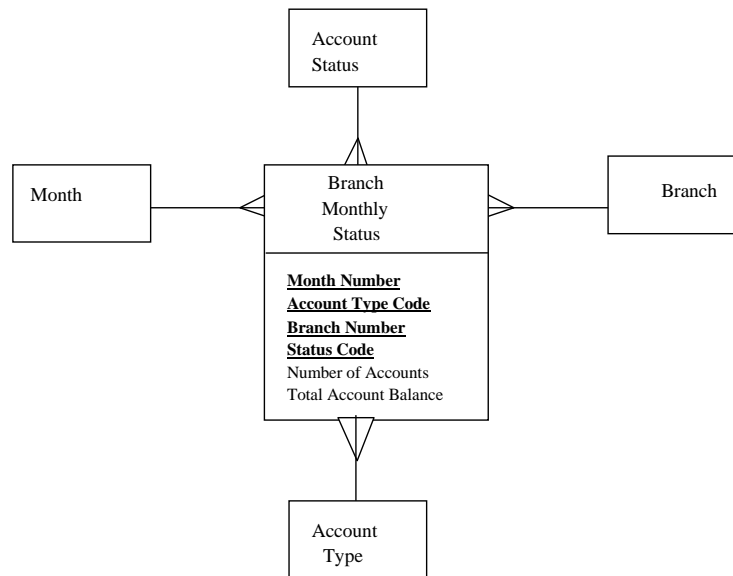
Dimensional Modeling

-
-
-
-
-
-
-



Sample Query

Give me the total account balance and number of accounts by account type by month by branch by status.



What limitations does this solution have?

What Is A Dimensional Model?

- A dimensional model is a model in which the data is structurally classified as fact or dimension.
- General characteristics of dimensional data:
 - Query oriented
 - Structured around data querying not business rules
 - Organized roughly into base facts and dimensions of those facts
 - Based on identification of the grain of data
 - Consisting usually of snapshot data
 - Looks to reduce the number and depth of joins
- Levels of dimensionality can differ:
 - From very atomic to highly aggregated
 - From few dimensions to many
 - Facts usually contain three or more dimensions
- *As you go from atomic to aggregate tables, the data naturally tends to be more dimensionalized.*



Two General Types of Tables

■ Fact Tables

- ▶ Contain the numeric values of interest to the business analyst
- ▶ Represent the natural facts found in the business and the dimensions by which they are identified and measured
- ▶ Are the base values used in analyzing the business
- ▶ Are sometimes derived - i.e., summarization of base amounts from detailed data

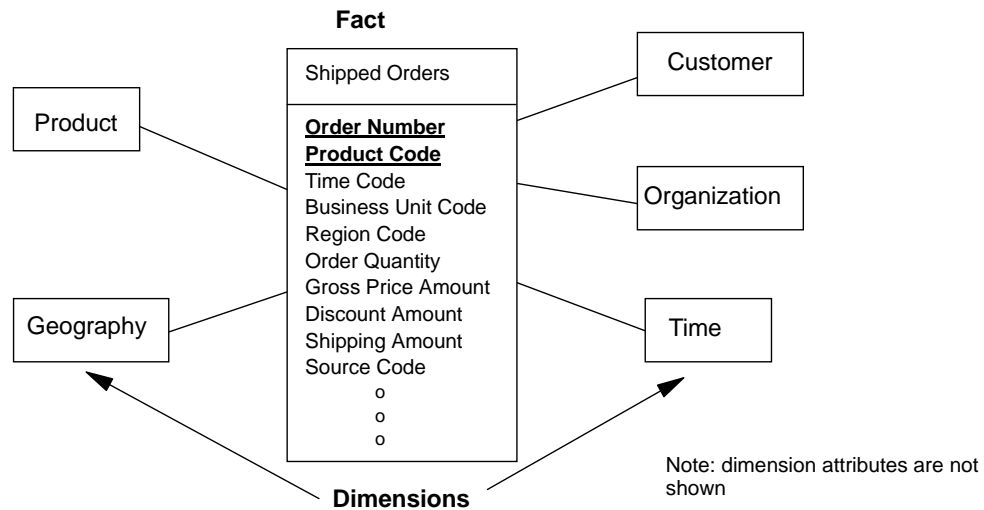
■ Dimension Tables

- ▶ Basically, code table, though often very large and abundant code tables
- ▶ Consist of the attributes used in forming and examining the fact table
- ▶ Contain mostly descriptive elements used individually or in various combinations to characterize the facts



▼ Facts (Fact Tables)

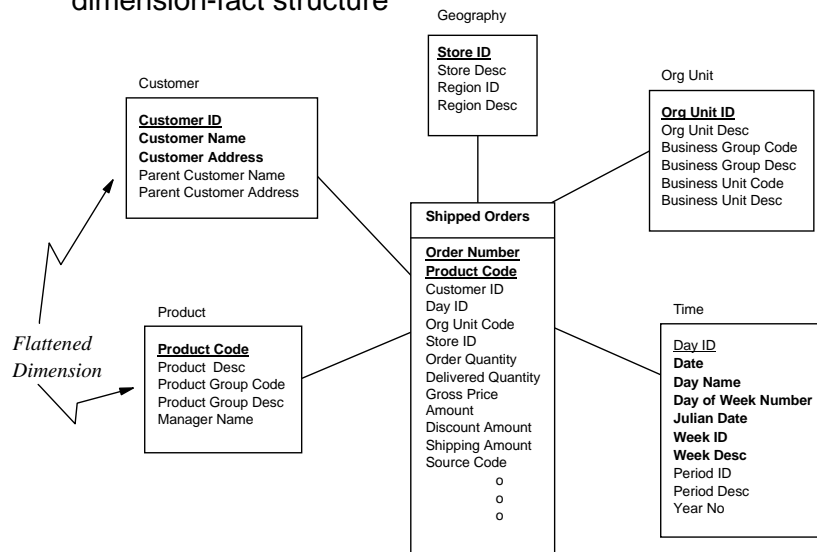
- Facts are the central (major) table in a Star Schema
- They are the largest table in the warehouse database.
- Dimensions share primary/foreign key relationships with fact tables
- The Fact Table is highly normalized since it resides at the intersection of a many-to-many relationship among the dimensions



Classic Star

- A fact surrounded by a **single circle** of dimensions

- ▶ Multi-levelled dimensions are flattened
- ▶ Designed for direct support of queries that have an inherent dimension-fact structure

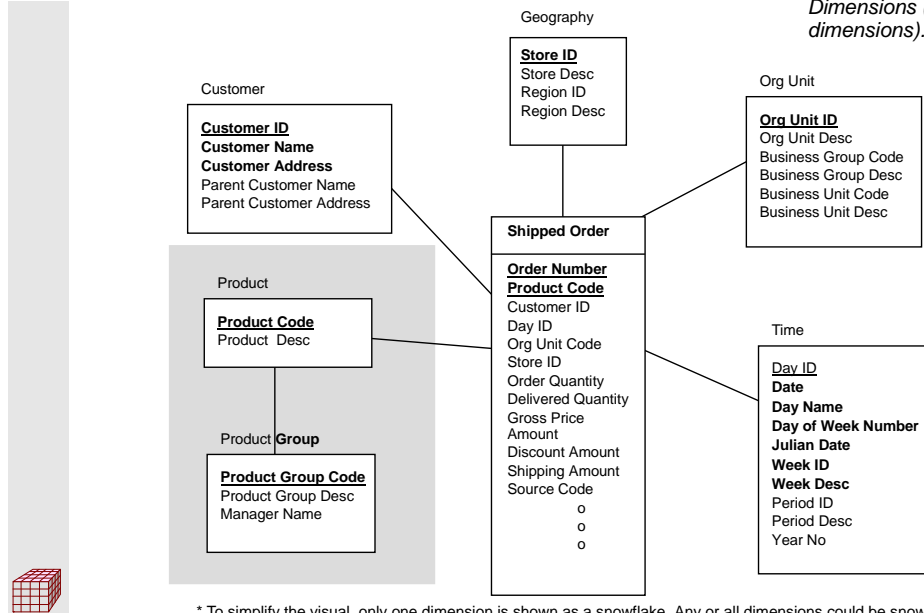


Snowflake

■ Separate levels of a dimension are kept separate*

- More flexible
- Reduces batch updates to dimensions

** Tho always said to be slower than a star, some tests have revealed no difference in performance between Flattened and Snowflaked Dimensions (at least for wide dimensions).*



* To simplify the visual, only one dimension is shown as a snowflake. Any or all dimensions could be snowflaked.

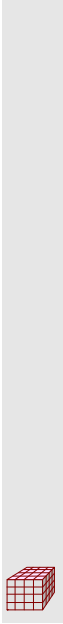
Snowflake Schema

- Suitable for many-to-many and one-to-many relationships among related dimension levels
- Required for many-to-many fact-dimension relationships (e.g., customer-policy)
- **Pros**
 - Less storage space. However, normalizing the Star based solely on amount of storage saved is not warranted unless the volume is huge
 - Could improve performance, flexibility, and maintainability under some circumstances.
- **Cons**
 - May complicate the user understanding of the warehouse database.
 - Makes the database design seems more complex. Some hard-core OLTP database designers see it as a natural.
 - Could impact browsing performance due to extra joins required to access the Facts.



Exercise

- Do case study Exercise 4: star vs. snowflake.





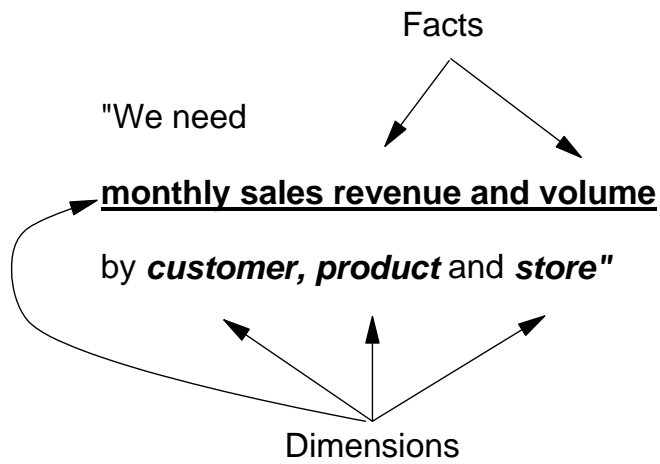
Steps for Dimensional Modeling

- Dimensional models are created in a five step process, as follows:
 - ▶ 1. Identify the business process or question (see previous chapter)
 - ▶ 2. Declare the necessary grain of data
 - ▶ 3. Define the dimensions
 - ▶ 4. Define the facts
 - ▶ 5. Determine the summary levels



▼ Step 1. Identify Business Process

- The business process can be expressed as requirements or questions

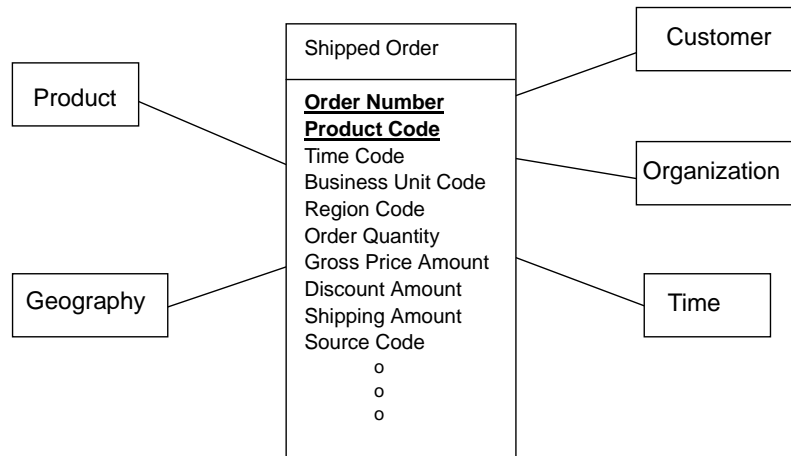


["Monthly" represents the time dimension]



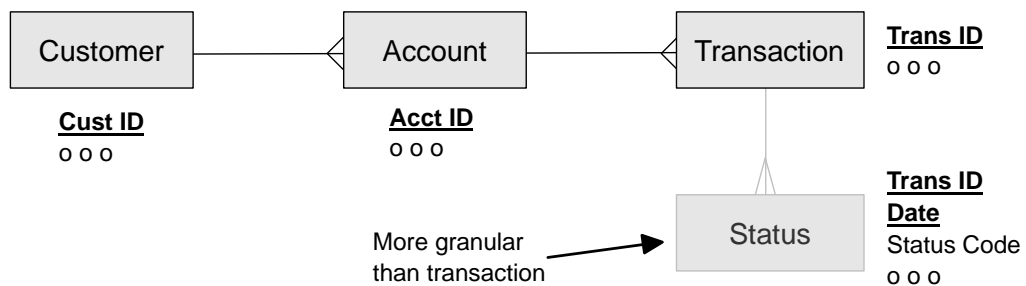
Step 2. Declare the Grain

- Define the most detailed grain necessary to support requirements
- The grain is expressed by the level of detail of the fact table
- This is an example of a transaction level detail (Shipped Order)
- This example is:
 - Very detailed
 - Most granular



What Is Granularity?

- A key question in building the DW is determining the level of granularity
- Granularity is the level of detail of the data
- Granularity is determined by:
 - The level of detail of the data itself (is it detailed or summary?)
 - The identifier of the data (is it simple or complex?)
 - Adding parts to a key always increases the granularity
 - Removing parts of a key always decreases the granularity



▼ Three Main Grains

- There are three main grains pertinent to the Data Warehouse:

- ▶ The Transaction One product per transaction
e.g., "Buy 100 shares of IBM"
or "A disability payment"
- ▶ The Line Item A variant of the transaction
Multiple items per transaction
Some common data across all line items
e.g., "A claim and all its component details"
- ▶ Snapshot A periodic summary of status
Represents status of accounts or business entity
Applicable to balance businesses
Examples are banking and insurance
e.g., "This month's checking account balance"
or, "The status of a claim over time"



▼ Step 3. Define the Dimensions

- Characteristics of Good Dimensions
 - Comprehensive
 - Include the main attributes, even hierarchical attributes
 - Include all relevant business attributes
 - Comprehensible
 - Terse, cryptic and hidden codes for attribute meaning should be avoided
 - Complete
 - Missing and null values must be avoided
 - Documented
 - The dimension must be formally documented and maintained in the metadata repository
 - Quality assured
 - Explicit business rules maintained and monitored
 - All data values (including spelling and case) should be cleansed prior to row insertion
 - Utilized
 - The facts must be fully qualified by appropriate dimensions



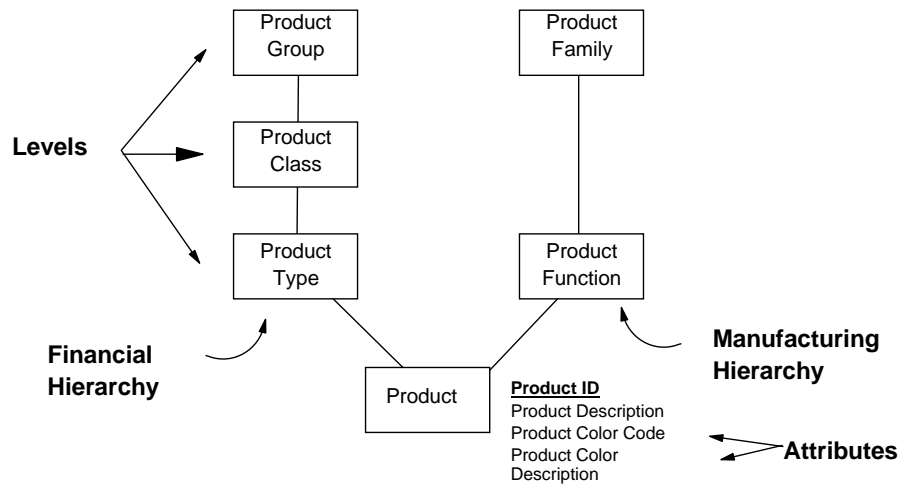
Structure of a Dimension

■ Dimensions can have:

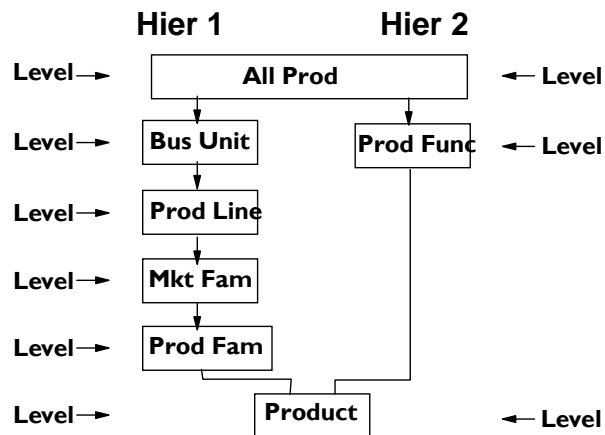
- Attributes
- Levels
- Multiple hierarchies

■ Dimensions can be:

- Symmetric dimensions
- Asymmetric dimensions



Multiple Hierarchies



Manufacturing Hierarchy:

All Prods
 Bus Unit -- Image
 Prod Line -- Stationery (25)
 Mkt Fam -- Stationery 1125
 Prod Fam-- 2 Color Checks 10
 Product -- Expense Checks

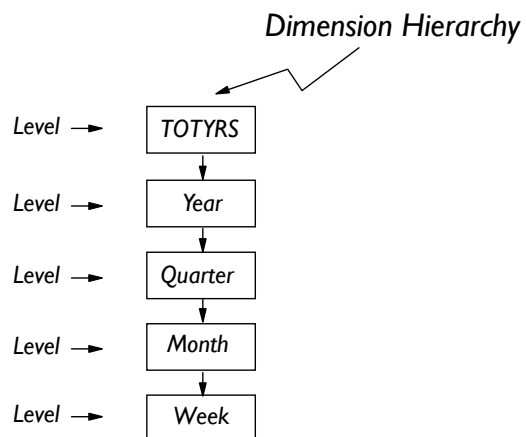
Marketing Hierarchy:

All Prods
 Prod Fun -- Checks - PN10
 Product -- Expense Checks



Symmetric Hierarchy

- In a symmetric hierarchy, a given child always has parents at same level
- Exemplified here by Time dimension
- Often exemplified also by Customer dimension



Examples:

TOTYRS

Year - FY96 Jul95-Jun96

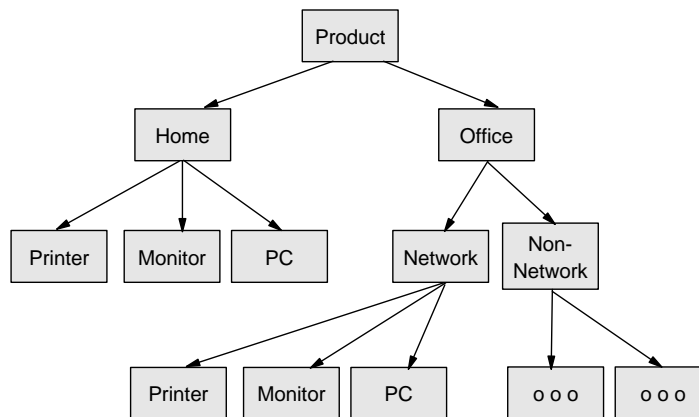
Qtr - Q296

Mth - AprFY96

Week - WK9615



Asymmetrical or Ragged Hierarchy



- Any instance of a dimension may not have number parents above it or children below it
- Often called "ragged" or "unbalanced" hierarchy
- Exemplified here by Product dimension
- Often exemplified also by Organization dimension
- The levels themselves in a ragged hierarchies can be named or unnamed

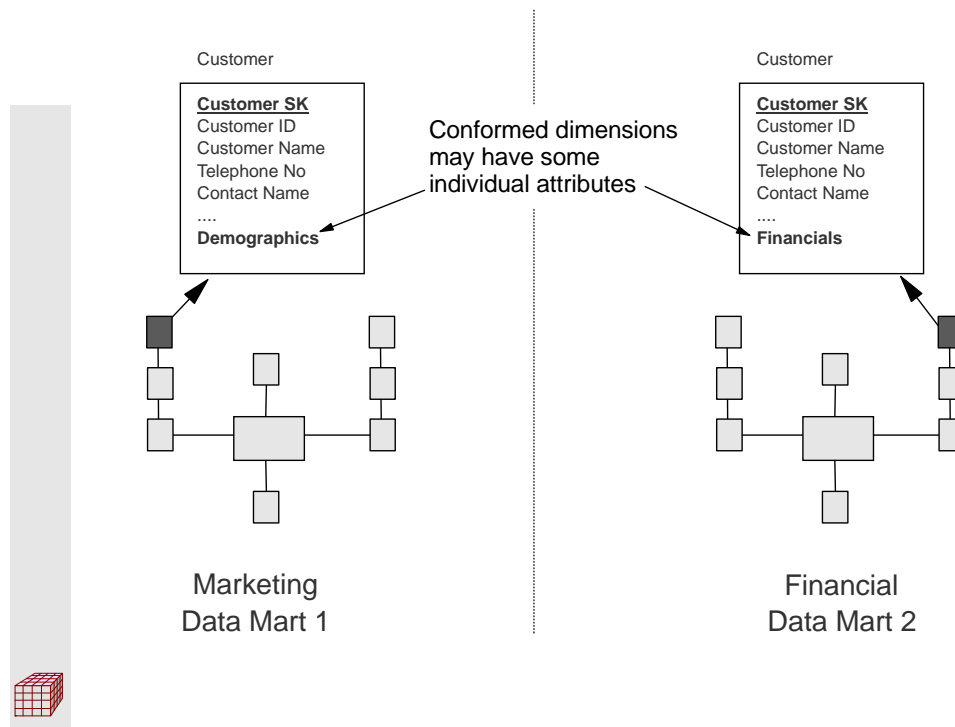


▼ Conformed Dimensions

- Conformed dimensions means that when multiple copies of the same dimension exist, they are consistent with one another in that they have:
 - The same key
 - The same values for the key
 - The same granularity
 - Attributes in each conformed dimension can be a subset and can overlap
- Must be built at a common level of granularity (or be a rollup of the base dimension).
- Generally identified by a surrogate primary key.
- Must establish a firm policy to reuse the conformed dimension whenever that subject is required for a data mart
- Also called dependent dimensions or architected dimensions.



Conformed Dimensions



Time Dimension

- Management usually wants to analyze and measure the business by such things as workdays versus holidays, this year vs. last year, calendar or fiscal periods, seasons, etc.
- Primary time key may be:
 - A date or date/time stamp
 - A surrogate key (with the time as a non-key attribute)
- Created in advance to cover previous history and future
 - Usually structured into a complete calendar
 - Often flattened into a single dimension row
 - 10 years worth of days is only about 3,650 records!



Time Dimension in Consumer Products

Time Dimension

Day ID

week_day_name
calendar_day_number
fiscal_day_number
calendar_week_number
fiscal_week_number
period_id
period_name
month_id
month_name
calendar_month_number
fiscal_month_number
quarter
fiscal_period
holiday_flag
weekday_flag
last_day_in_month_flag
season

Sales

Day ID

Product ID

Store ID

promotion_code
sales-quantity
sales_revenue_amount
customer_count

Product Dimension

Store Dimension

Promotion Dimension





Product Dimension in Consumer Products

- Description, usually in a hierarchy, of "What a company sells"
 - ▶ Might be product or service
 - ▶ Many attributes in product dimension are not applicable to the entire product hierarchy, such as Package, Size, or Form/Flavor
 - ▶ This dimension should have as many descriptive attributes as necessary:
 - 50-100 is not uncommon
 - ▶ Essentially sourced from product master file, though it may be enhanced
- Consumer Products example
 - ▶ Each Product ID relates to a single SKU
 - ▶ Which rolls up to Brand to Subtype to Segment to Division



Product Dimension

Product Dimension

Product ID

SKU_description

SKU_number

Size

Brand

Vendor

Subtype

Segment

Department

Package

Formula

Flavor

Units_per_case

Cases_per_pallet

o o o



50-100 attributes is not uncommon

Customer Dimension

- Peter Drucker says that "there is one and only one purpose for a business - to find a customer"
 - ▶ Comparatively small - in utilities and other wholesale companies
 - ▶ Very large - in financial institutions, telephone companies, catalog retailers, insurance carriers
- Some customer attributes can be heavily searched, others not so
 - ▶ Heavily searched fields include age, sex, number of children, income level, education level, behavior scores related to purchasing or credit use
 - ▶ Selection is seldom based on attributes like first name and street address.
- Customers may exist in several hierarchies, such as
 - ▶ Rollups for billing, sales channel and organization responsibility
 - ▶ Even derived hierarchies such market segmentation rollups
- Avoid cryptic, 1-character codes within customer



▼ Step 4. Define the Facts

- There are basically three types of measures:

- ▶ **Additive**

- Counts and amounts. No matter how you slice or dice it makes sense to total

- ▶ **Semi-additive**

- Counts, averages and percentages
- Be careful not to double count
- Usually constrained on one dimension

- ▶ **Non-additive**

- Ratio
- Ratio of sums, not sum or ratios

- ▶ Event tracking-registering the event of the same set of keys being in the same place at one time, make it = "1" for easier counting



Types of Fact Tables

- There are essentially three main types of fact tables:
 - ▶ **Transaction:** these represent single events within the business, such as :
 - Trade a stock in brokerage
 - Make deposits or withdrawals in banking
 - ▶ **Line Item:** these represent groups of sub-transactions that are combined in a single event.
 - These are really an extension of the above Transaction Fact in which the transaction has multiple line items
 - Examples are Orders, Invoice, Shipments, Purchase Orders.
 - ▶ **Status:** these occur in balance forward businesses such as insurance and banking
 - Examples are Demand Deposit (Saving and Checking), where accounts have a monthly balance,
 - Brokerage accounts, where again they have a monthly balance



Transactional Fact Tables

Banking Transaction

Transaction Id [PK]

Branch Id [FK]

Time Id [FK]

Date Id [FK]

Account Id [FK]

Transaction Type Code [FK]

Transaction Amount

Transactional fact tables typically represent a single fact.
Examples include banking, brokerage trading and ATM transactions.



Line Item Fact Tables

Order Line Fact Table

<u>Order Id [PK][FK]</u>
<u>Product Id [PK][FK]</u>
<u>Order Line Number [PK]</u>
Customer Id [FK]
Date Id [FK]
Line Status
Line Type
Order Qty
Discount Amt
Revenue Amt

Line Item fact tables are a variation of the transaction fact table. The Line Item fact table typically associated with production events like Sales Orders, Purchase Orders and Shipments, to name just a few. Usually modeled as a Header and Line Items.



▼ Status Fact Tables

- Status fact tables are a very flexible type of fact table.
- It is essentially a snapshot of predefined, interrelated facts.
- Typical of balance forward or payment due businesses, such as banking, insurance.

Policy Monthly Status

Policy Id [PK][FK]
Policy Type Code [PK][FK]
Date Id [PK][FK]
Status Code [PK][FK]
 Customer Id [FK]
 Premium Paid Amount
 Premium Due Amount
 Times Payment Late Count
 o o o

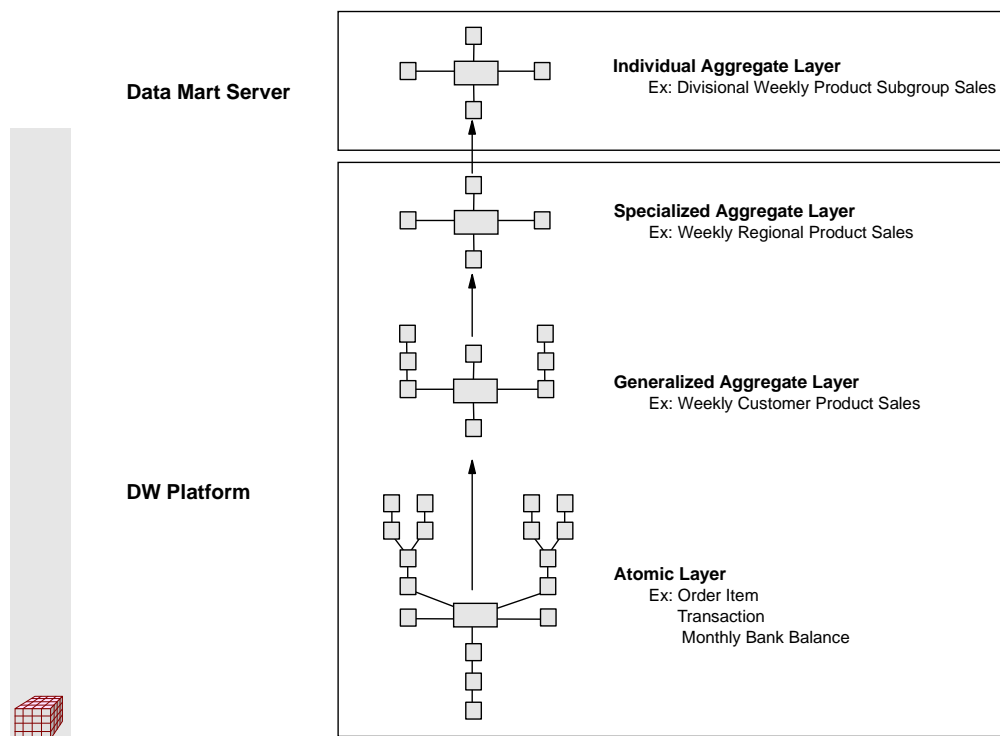


Step 5. Determine the Summary Levels

- The need for stored summaries can be determined in one of two ways:
 - By reacting to poor performance and
 - By analyzing queries in advance and predicting whether summaries are needed or not
- Which queries are required is determined by analyzing query data usage.
- There are three major steps for deciding on what summary levels to create:
 - First, build more generalized aggregates
 - If that does not give adequate performance, build more specialized aggregates
 - If that still does not provide adequate performance, then offload the aggregates to a separate server



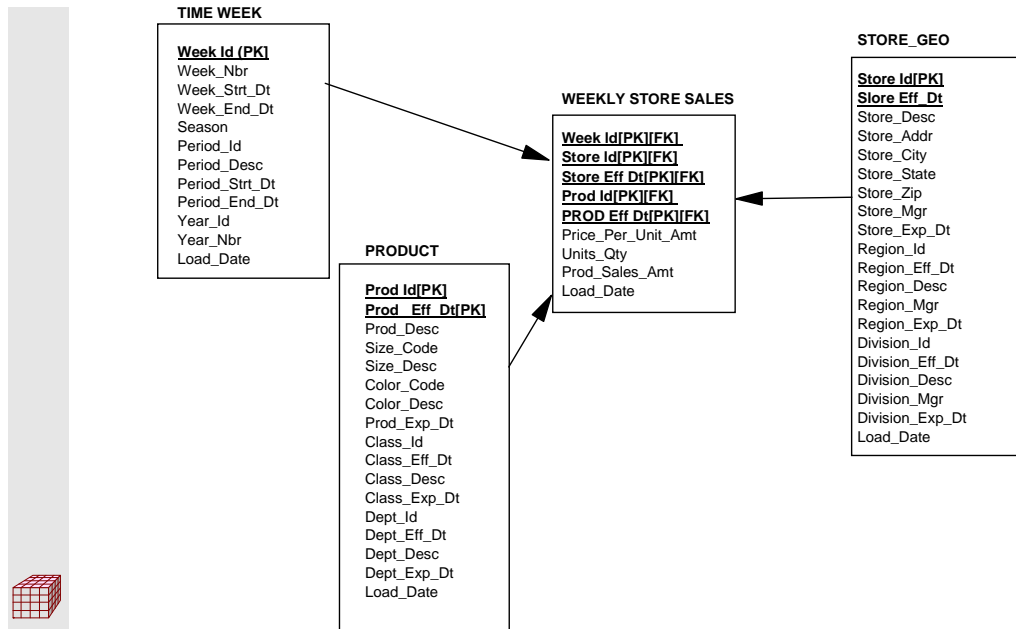
Layers of Data in the Data Warehouse



First Summary Level Grain

- Decide on granularity of first summary level that you intend to store

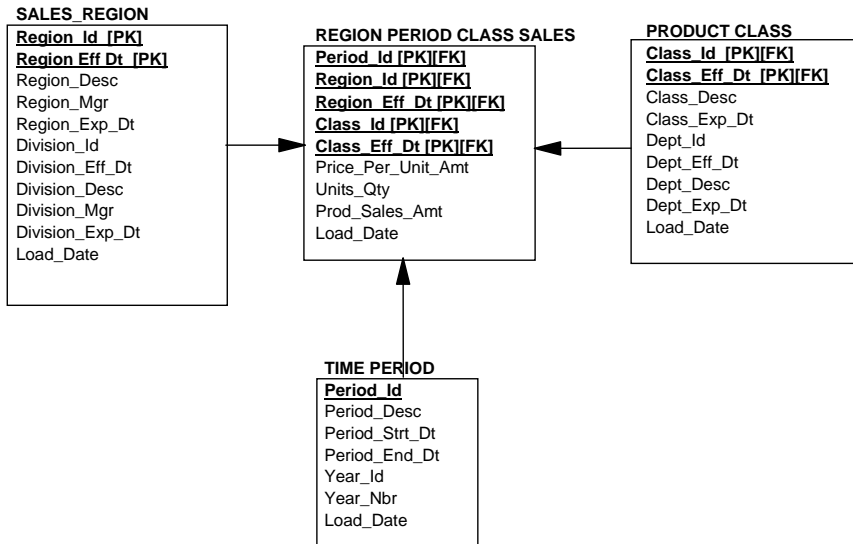
- In this example, Sales by Product by Store by Week.



Second Summary Level Grain

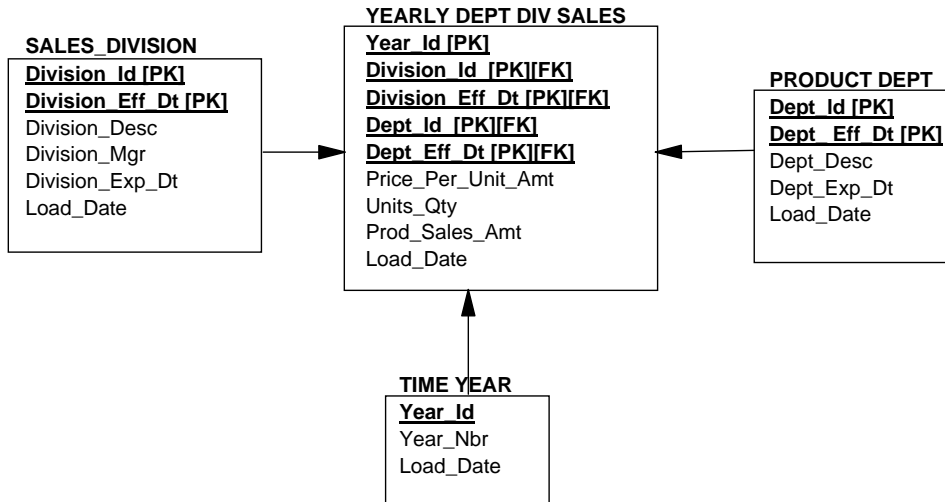
- Decide on granularity of the second summary level that you need to store

► In this example, Sales by Class by Region by Period.



Third Summary Level Grain

- Decide on granularity of the third summary level
 - In this example, Sales by Department by Division by Year.



History

- Maintaining history is one of the most important decisions in a DW
- Facts are naturally historical
- Historical dimensions are often called "slowly changing dimensions"
- There are three ways to maintain slowly changing dimensions
 - 1 - Do not keep history. Overwrite the current record each time with the change.
 - **Customer Id**, Customer Name, Customer Zipcode
 - 2 - Keep all changes. This is done by adding a date/time stamp to the key of the base record
 - **Customer Id, Date/Time**, Customer Name, Customer Zipcode
 - 3 - Keep only the current and previous value of a dimension attribute
 - **Customer Id**, Customer Name, Current Customer Zipcode, Last Customer Zipcode



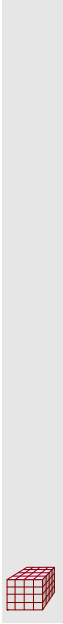
Important Terms

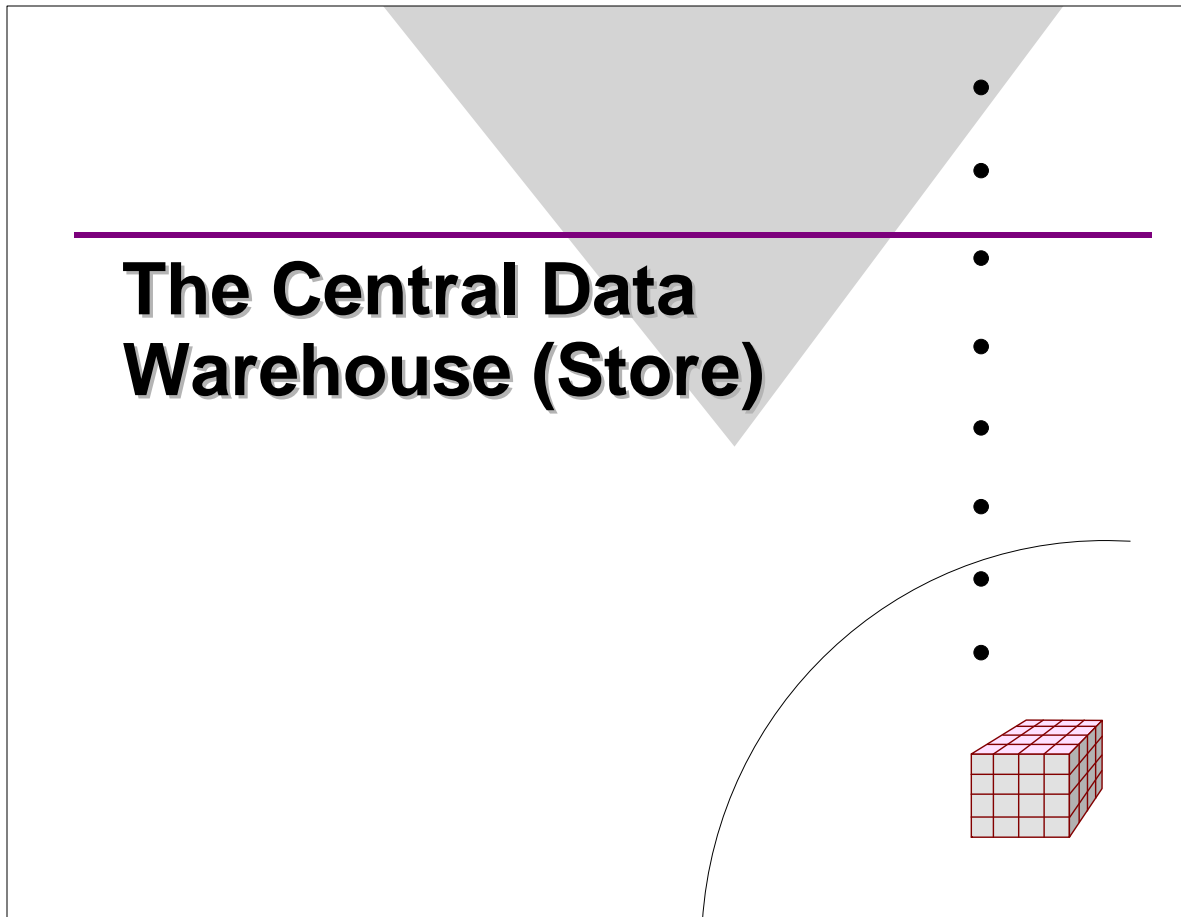
- Data warehouse
- Data mart
- Central data warehouse
- Gather-store-deliver
- OLAP
- OLTP
- Dimensional model
- Fact
- Dimension
- Dimension hierarchy
- Types of fact tables
- Star schema
- Snowflake schema
- Slowly changing dimension
- Aggregation



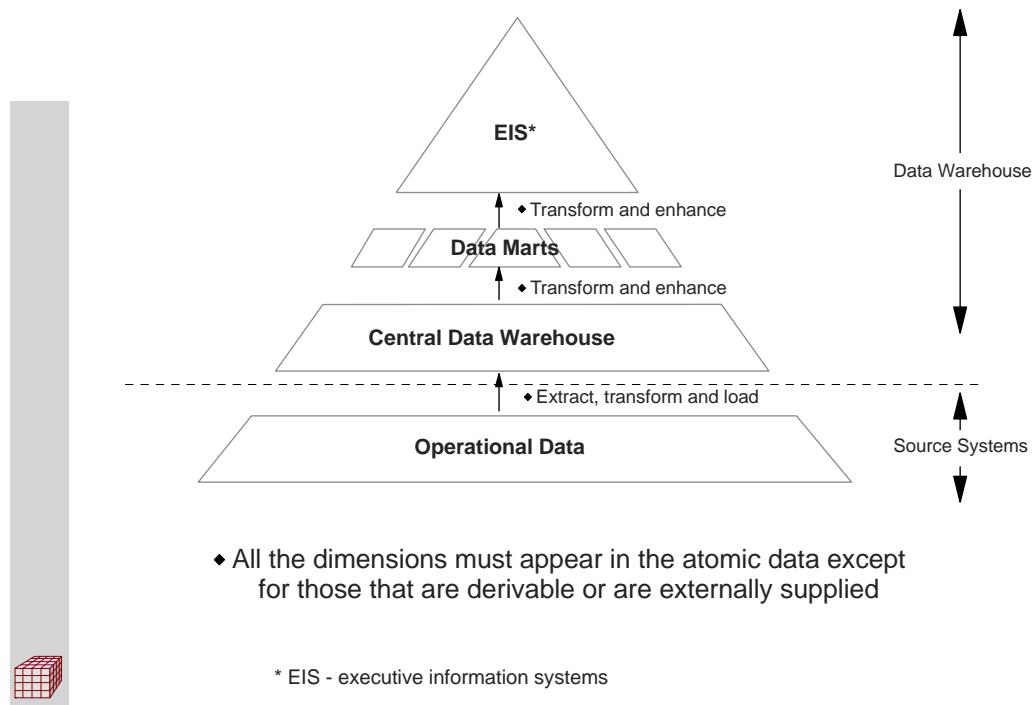
Exercise 5

- Do case study Exercise 5: Dimensional Modeling



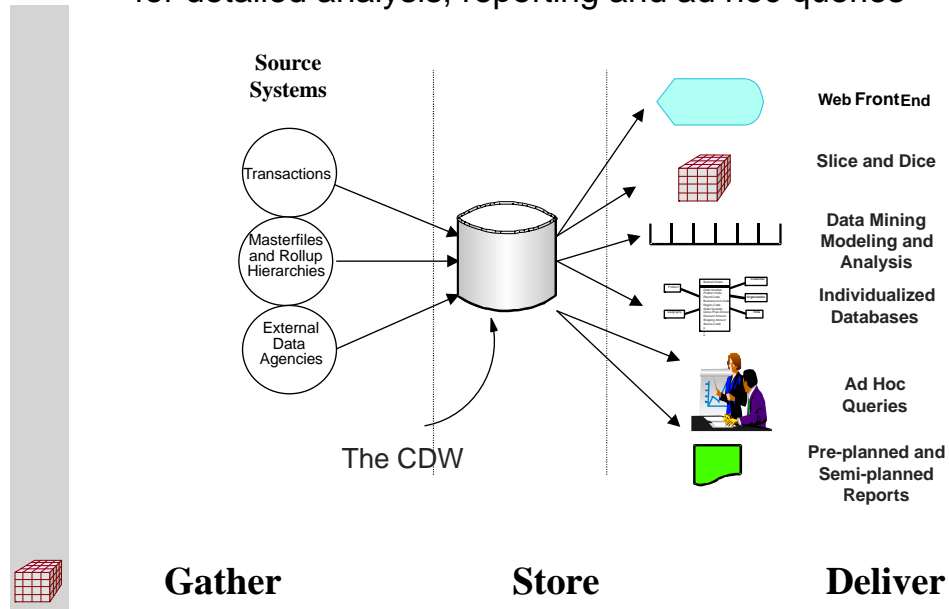


Data Warehouse Environment



The Central Data Warehouse

- The Central Data Warehouse (CDW) contains all base data, feeds all marts and mining applications, and is used for detailed analysis, reporting and ad hoc queries

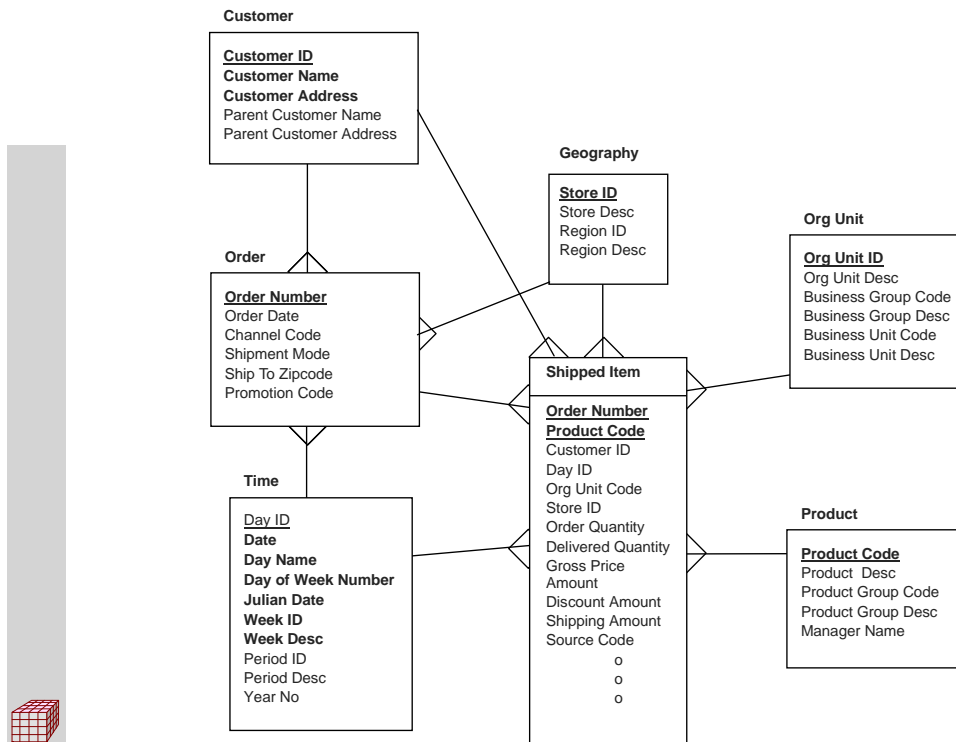


The Central Data Warehouse

- The central data warehouse is the core of the data warehouse environment
- Its main characteristics are:
 - ▶ Detailed (to answer all questions)
 - ▶ Atomic (so you have the lowest available level of data)
 - ▶ General purpose (not tied to specific applications)
 - ▶ Flexible (so that new questions can be satisfied without changing the data - unless the business rules change)
 - ▶ Time variant - must contain appropriate levels of history, a decision which is determined by the business
 - ▶ Having controlled redundancy - redundancy is commonplace in reporting systems (after all they are not update systems) but it should be controlled so that users can go to one place to get the right numbers



A (Simplified) CDW Model



Data Warehouse Data

- The data warehouse data model will contain different and distinct levels of business data:
 - ▶ First distinction - between detailed and summarized data
 - ▶ Second distinction - between current and older detail data
 - ▶ Third distinction - between original, restated and snapshot
- It will also contain distinct metadata:
 - ▶ The data warehouse itself contains metadata that is greatly different in format, content and usage from traditional operational metadata
 - ▶ A key criteria for successful use of the warehouse is knowledge of the data contained within it (metadata)



Central Data Warehouse Questions

- Questions like these should be used to determine the data going into the CDW:
 - What are the primary measures of the business?
 - What is the necessary grain of data?
 - How much and what kind of history is needed?
 - How must the data be dimensioned?
 - What kind of flexibility is needed?
 - What kind of ad hoc usage might occur?



The Grain of the Data Warehouse

- Determining the required grain of data is the number one question for designing data in the central data warehouse
- The grain determines:
 - The lowest (atomic) level of detail of the warehouse
 - The size of the warehouse database
 - The dimensionality of the warehousing environment
 - The flexibility of the entire data warehouse
- Consider starting your data warehouse with the greatest level of granularity that you can afford (i.e., most detailed)
 - Overly granular data can always be aggregated
 - Overly aggregated data can not be decomposed
 - (One would have to repopulate the warehouse)



Multiple Grains in the Data Warehouse

- A single CDW could easily contain multiple levels of grains:
 - ▶ **Base** or atomic grains representing the transaction level
 - ▶ A periodic **snapshot** or summary
 - ▶ **Collection** of data or aggregates of data, packaged for easy consumption
 - ▶ **Archive** data - history on demand
 - ▶ **Metadata** - DNA of data



How Much Detail In The Warehouse?

- There is a very real need for both detailed data and summary data.
- Detailed data is always needed for full history, ad hoc querying, complete restatement of results, and drill-through (from data marts)
- Summary data is needed because it:
 - Is of much smaller volume and so is much easier to manage and access
 - Is a consistent basis from which additional analysis can be done



Time and History

- Because the data warehouse is a collection of snapshots, there are many different ways to add time and history:

- ▶ A time stamp included in the key
- ▶ Separate tables or columns for each period
- ▶ Different ways to treat time as an attribute:

- Effective Date
- Expiration Date
- Refresh Date



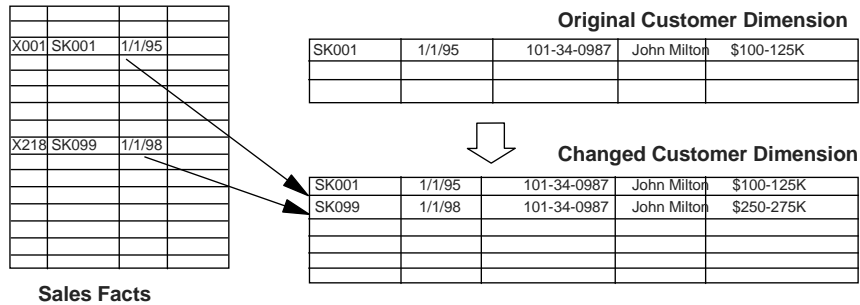
- There are even different ways to aggregate time:

- ▶ Recency - how recently has a customer last ordered
- ▶ Frequency - how frequently has a customer ordered in a time span
- ▶ Periods - days can be aggregated into periods of 4 weeks
- ▶ Monetary value - what is the average or largest value of a customer's orders



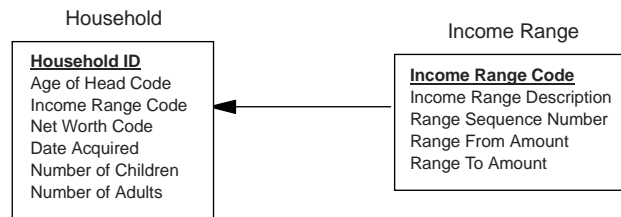
Changing Dimensions

- Adds a new row every time there is change.
- Requires specialized treatment:
 - Add Date to the key
 - Customer ID + Version Number
 - A surrogate key
- Users unaware of existence of such specialized keys



Value Banding

- Creating ranges from discrete values
- A method for reducing the volume of data and simplifying querying
- Typically used for concepts like:
 - Income
 - Net worth
 - Years of education
 - Years of residence



Data Warehouse Usage

- The constant monitoring of the data warehouse will determine the pattern of usage
- This information must be used by the data warehouse designers to change the data warehouse data model and database design, thereby making the data warehouse more functional



Terms

- Central data warehouse
- Granularity
- Grain
- Aggregate
- Metadata



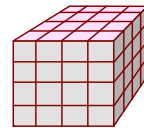
Exercise 6

- Do case study Exercise 6: The Central Data Warehouse



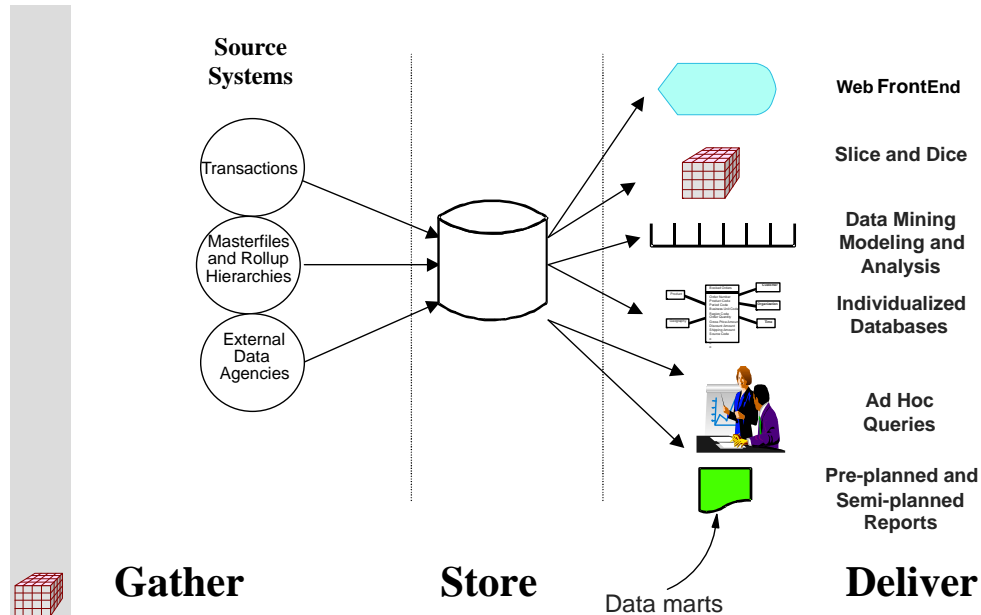
Data Marts (Deliver)

-
-
-
-
-
-
-



A Data Mart

- A BI application containing a specialized collection of related data, customized for a specific community of knowledge workers, analysts or planners, to support their reporting and analysis needs





Purposes of a Data Mart

- The purposes of a data mart are:
 - ▶ To satisfy the specific information requirements of a specific audience
 - ▶ To provide a compelling software tool that well suits the requirements of that audience
 - ▶ To be easier and simpler to use than the central data warehouse





Types of Data Marts



■ From viewpoint of **content**:

- ▶ *Summary data marts* - aggregated data (sales last month)
- ▶ *Subset data mart* - complete structure but only some rows or some columns (a 20% sample for exploration)

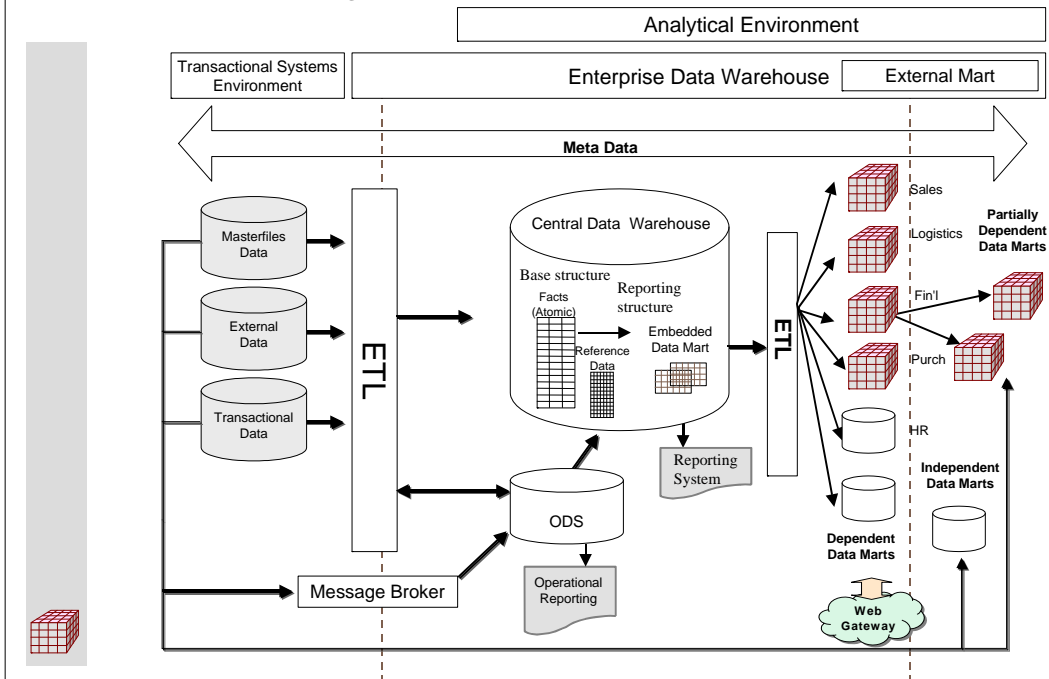
■ From viewpoint of **source**:

- ▶ *Embedded Data Marts* - stored within the CDW* and fed only from the CDW
- ▶ *Dependent Data Marts* - stored outside the CDW but fed only from the CDW
- ▶ *Independent Data Marts* - stored outside the CDW, fed completely by systems other than the CDW

* CDW = Central Data Warehouse (database)

Architecture of a Data Warehouse

- Notice the placement of data marts relative to the CDW
- Notice also that some reporting is done straight off the CDW, without using data marts



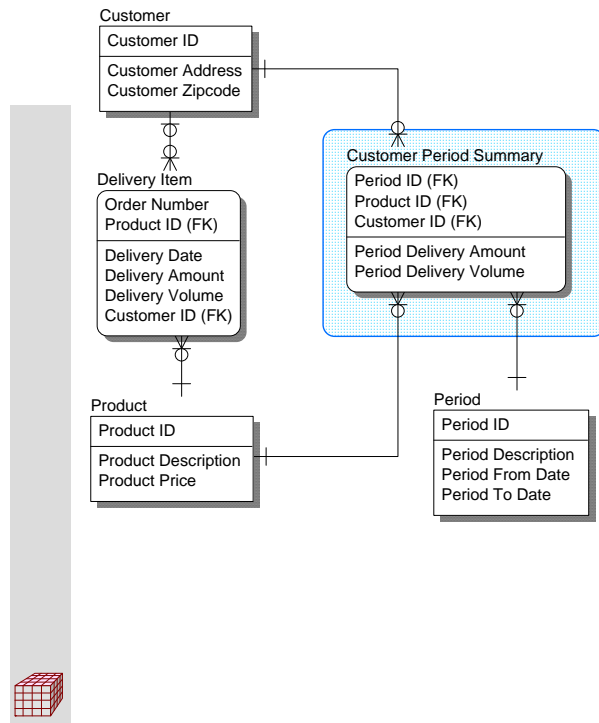


Sample Summary Data Mart

- **Booked Sales Dollars** – Taken from the order entry system. It is the dollar amount of an order a customer places at the suggested retail price.
- **Booked Items** – Taken from the order entry system. It is the number of items listed on an order. At this time, it does include special charge items.
- **Booked Orders** – Taken from the order entry system. It is the number of orders placed by customers.
- **Shipped Sales Dollars** – The amount of dollars for orders that were actually shipped.
- **Shipped Standard Costs** - The cost of producing the orders that have been shipped using standard hours and standard rates.
- **Billed Sales Dollars** – The dollar amount for orders that were shipped and billed.



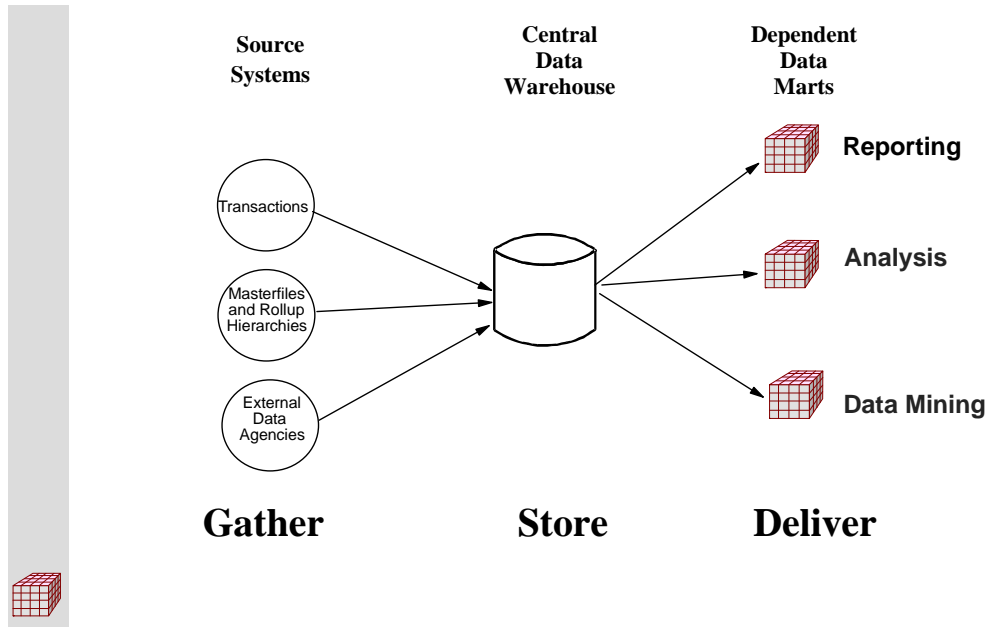
Embedded Data Mart



- These are essentially stored summaries, designed for a specific audience.
- They can be stored as:
 - Summary tables
 - Cubes
- Two edged sword:
 - On the one hand, one of the primary ways to improve performance.
 - On the other hand, require ETL (or SQL) to build

Dependent Data Marts

- Data is source exclusively from the data warehouse





Dependent Data Marts

■ Why?

- ▶ Volume of access and distance make it better to locate data near the user. (e.g. an international environment)
- ▶ There is some tool with functionality ideal for a user and it cannot run on the Data Warehouse
- ▶ Data Warehouse technology is not capable of dealing with enterprise-wide data
- ▶ To protect **one** organization's users from harming another organization's data or performance.

■ Pros

- ▶ Eliminates the redundant and inconsistent problems of the Independent Data Mart
- ▶ Sources all data from the Data Warehouse instead of legacy systems.

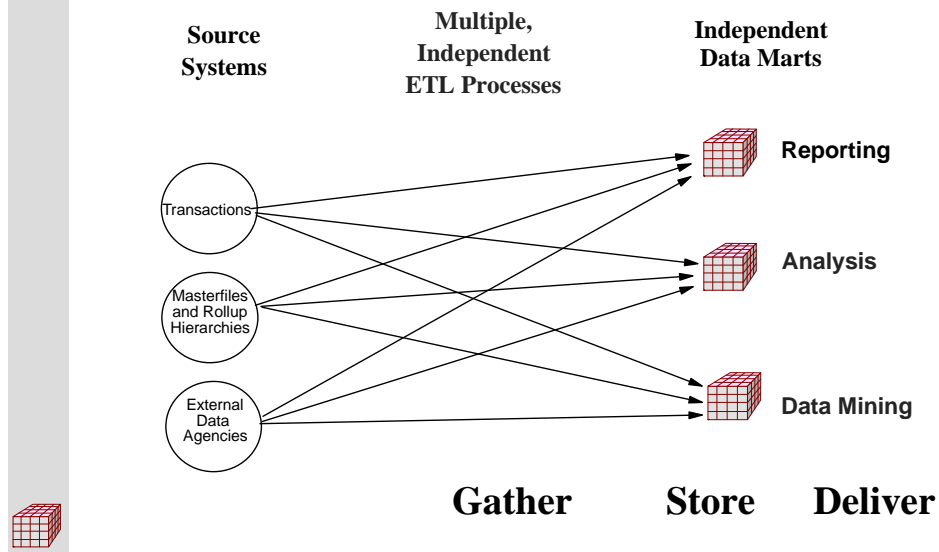
■ Cons

- ▶ Extra development and system demand due extra ETL
- ▶ More application code to create and maintain
- ▶ Extra resources on Data Warehouse for extract, and on Data Mart for load (may lengthen the batch window)
- ▶ With change, "time to market" will eventually be longer due to the number of components to change (but faster than the Independent Data Mart).
- ▶ Disk storage and processing power can not be shared by every users.
- ▶ Additional operation effort and cost to maintain and upgrade separate hardware, operating system, and DBMS software. (Same as the Independent Data Mart).



Independent Data Marts

- Source system directly feed the data marts
- A data warehouse may exist but is not used for sourcing





Independent Data Mart

- **Pros**

- Summarized prior to storage
- A smaller set of information for faster response and easier loading.
- Generally allows a first-time user to get an answer
- Traditionally built for a single organizational unit or audience

- **Cons**

- Although Independent Data Marts are quick and easy to create, as multiple organizations create them, they perpetuate the redundant and inconsistent information of the legacy systems.
- As the number of organizations having data marts increases, they fall victim to the same redundant processing and inconsistent information as the Legacy Information Systems.
- Because the data is often summarized in the Data Mart, the user can not ask questions any deeper than the level of summary provided.
- Summaries by themselves limit the breadth of question





Data Design Principle

- "Normalize base tables.
- Specialize reporting tables. *"

* To normalize is to eliminate redundancy,
using a specific set of rules called normal forms.





Granularity of Data Marts

- Data marts that are granular can support a wide variety of queries
 - You can always aggregate details
 - You cannot decompose aggregates
- Nevertheless, determining the level of detail of the data mart data is critical
- Avoid the two extremes:
 - All-in-one
 - Many small structures



All-in-One

- In this approach, all the data is collected into one place, such as one fact table or one cube

- **Pros:**

- ▶ All data is in one place
- ▶ Fewer joins are necessary
- ▶ Easy to model

- **Cons:**

- ▶ Many dimensions leads to a huge number of instances
- ▶ Roll-ups and summarization will always be necessary
- ▶ Excessive number of dimensions can be confusing to users (especially when part of the key)
- ▶ Excessive flattening of dimensions will be confusing to the users
- ▶ It is not likely all the dimensions or facts will be used together
- ▶ May result in sparsity in cells
- ▶ Usually has unacceptable performance

Loan Facts	
<u>Loan ID</u>	D I M E N S I O N S
<u>Month Year</u>	
<u>Loan Status Code</u>	
School ID	
School Branch ID	
Student Type Code	
Semester Tuition Range Code	
Degree Type Code	
Major Code	
Gender Code	
School State Code	
Original Loan Amount	F A C T S
Current Loan Balance	
Last Payment Amount	
Last Payment Date	



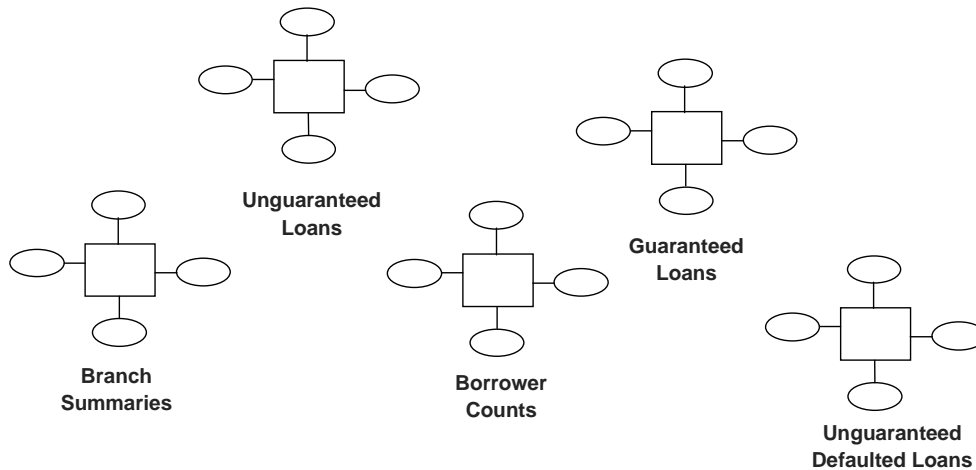
Many Special Purpose Structures

■ Pros:

- ▶ Tables contain fewer rows
- ▶ Data is optimized for end user usage
- ▶ Data is more understandable because of fewer dimensions
- ▶ Performance is better
- ▶ Can save space if tables are separated based on frequency of refresh

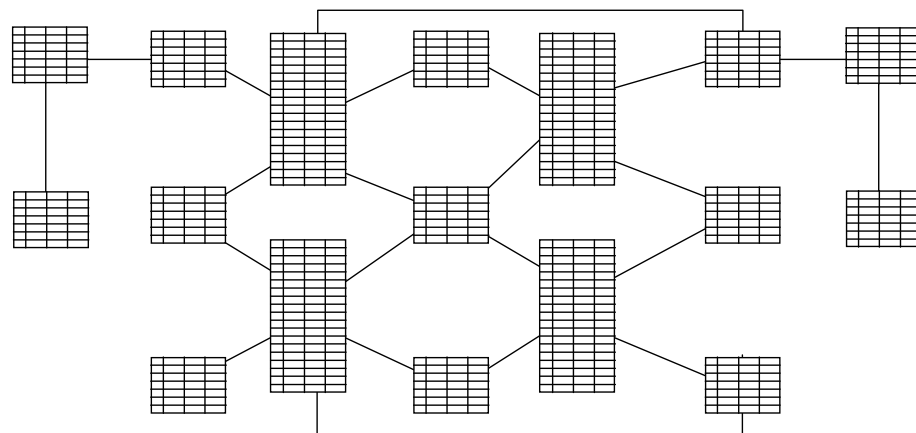
■ Cons:

- ▶ Joins or views may be necessary
- ▶ Cross-functional joins will affect performance
- ▶ Data model will be more complicated



What Do You Call ... ?

- Multiple snowflake schemas with Fact Tables linked through Dimension Tables.
- Usually associated with implementations of multiple Subject Areas



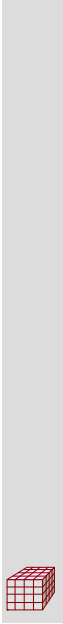
Terms

- OLTP
- OLAP
- Data mart
- Independent data mart
- Dependent data mart
- Cube
- Cell
- Dimension
- Dimension hierarchy



Exercise 7

- Do Exercise 7: Data Marts





Extract-Transformation- Load (Gather)

-
-
-
-
-
-
-
-



Extract-Transformation-Load

■ Extract-Transformation-Load (ETL)

- ▶ The process of gathering data from source systems
- ▶ Transforming that data to a form acceptable to the data warehouse and
- ▶ Loading that data into the base and/or aggregate tables of the data warehouse or data marts



ETL Litmus Test

- The litmus test for ETL is as follows

- ▶ ETL must handle multiple, disparate data sources running in disparate **processing environments**, including hardware and software technologies
- ▶ Resolve data and metadata existing in disparate **formats, states and granularity** and
- ▶ Determine and resolve detailed level **data quality** issues

- NOT BAD FOR A DAY'S WORK!



Transformation Budget Requirements

- Modest Transformation Effort
 - A simple, balanced warehouse iteration may be able allocate budgets evenly between the three primary aspects of warehousing
 - Depends on sources and transformations
- Complex Transformation Effort
 - Warehouse iterations with extensive or complex transformation requirements will need a larger share of the budget allocation

\$\$	Modest Transformation %	33	34	33
	Complex Transformation %	60	20	20
	DW Stage	Gather	Store	Deliver





The Transformation Challenge

■ Some major issues in the transformation of a data element

- ▶ Multiple meanings for the same data element
 - Different granularities
- ▶ Multiple sources for the same data element
 - Inconsistent formats
 - Encoding and other cryptic formats
- ▶ Differing levels of history for the same data element.
- ▶ Differing timing for the same data element.
- ▶ Questionable cleanliness and accuracy of the data element
- ▶ De-integration for audits and validation

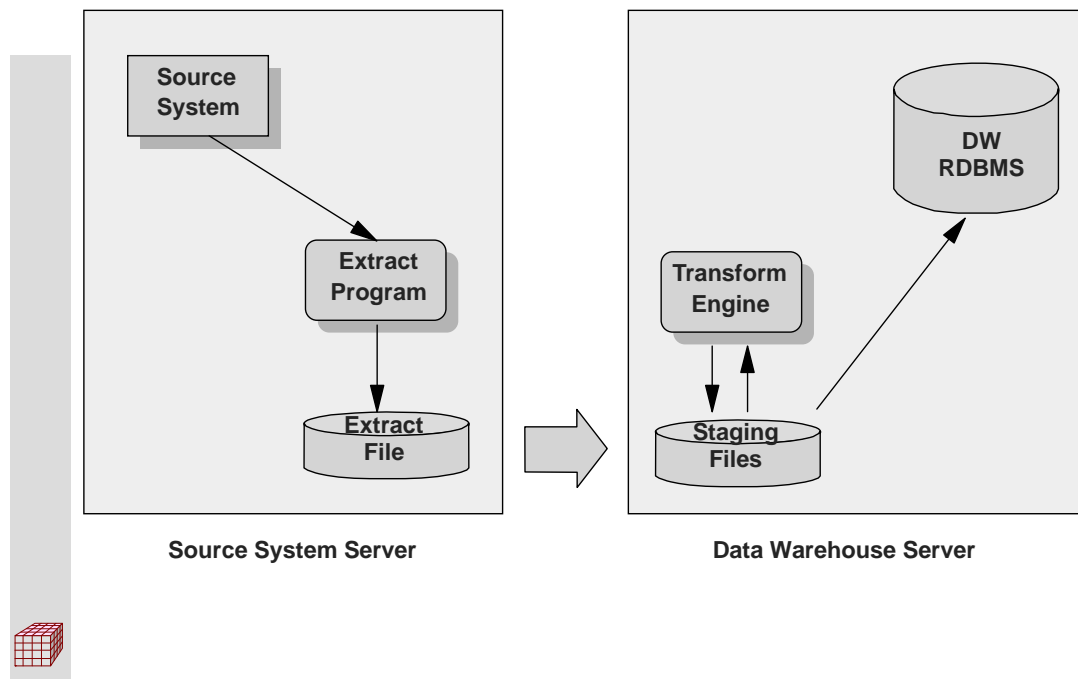
■ Other Issues

- ▶ Complexity of transforms
- ▶ Refresh cycle
- ▶ Length of batch window



▼ The Overall ETL Process

- Can be more than two servers



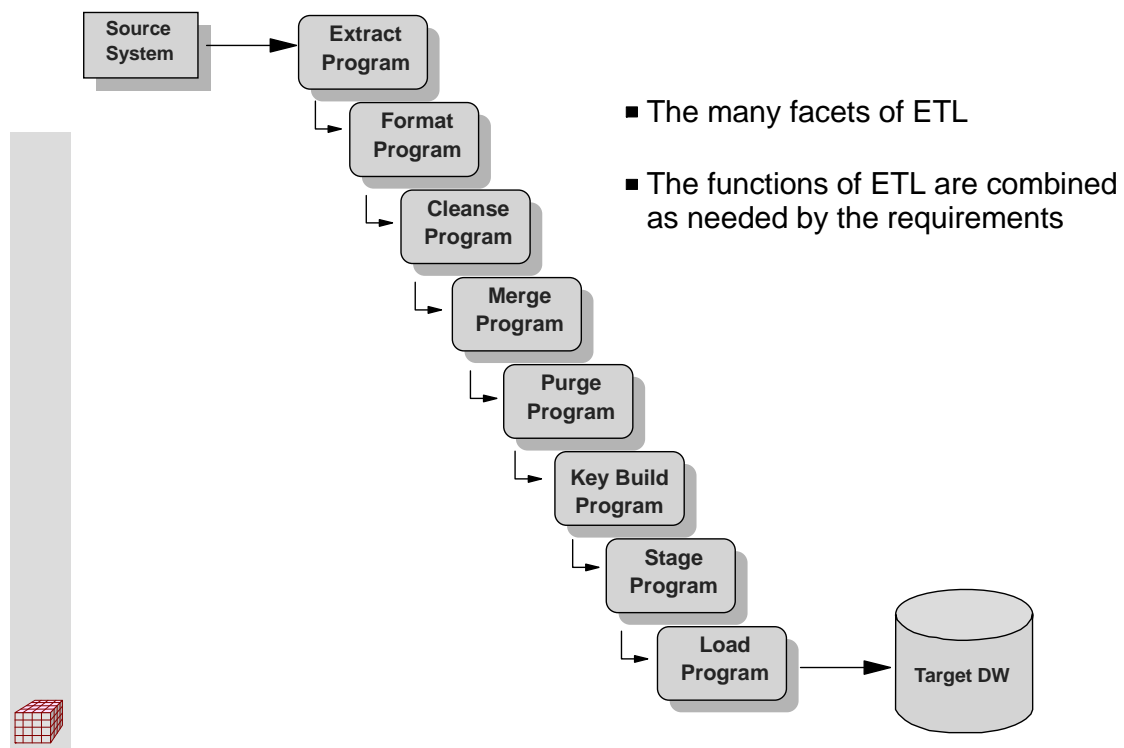


Source-To-Target Mapping

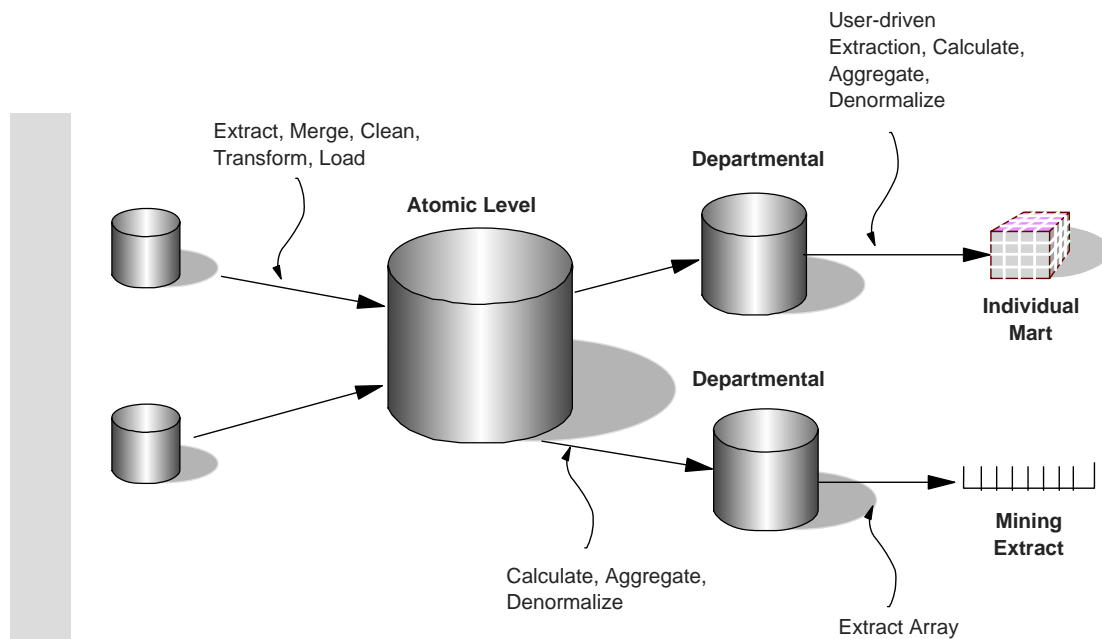
Source Table	Source Column	Transformations and Issues	Target Table	Target Attribute
LOCATION	loc_id (PK)	move	Store	Store_Id (PK)
	loc_eff_dt	make year 4-digits and move		Store_Eff_Dt (PK)
	region_id (FK)	select where Location.region_ID = BusinessUnit BU_Id and where BU_Type_Cd = "RGN" and move		BU_Id (FK)
	loc_name (3 fields)	concatenate and move		Store_Desc
	loc_add1	concatenate with loc_add2 and move		Store_Addr
	loc_add2	concatenate with loc_add1 and move		Store_Addr
	loc_city	move		Store_City
	state	move		Store_State
	loc_pscd	move		Store_Zip
	loc_mgr_name	move		Store_Mgr
		move system datetime		Store_Exp_Dt



Transformation Component Flow

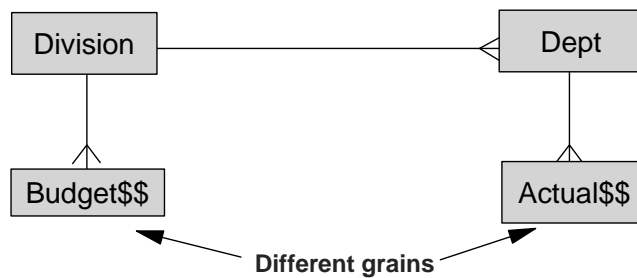


The Many Transformation Processes



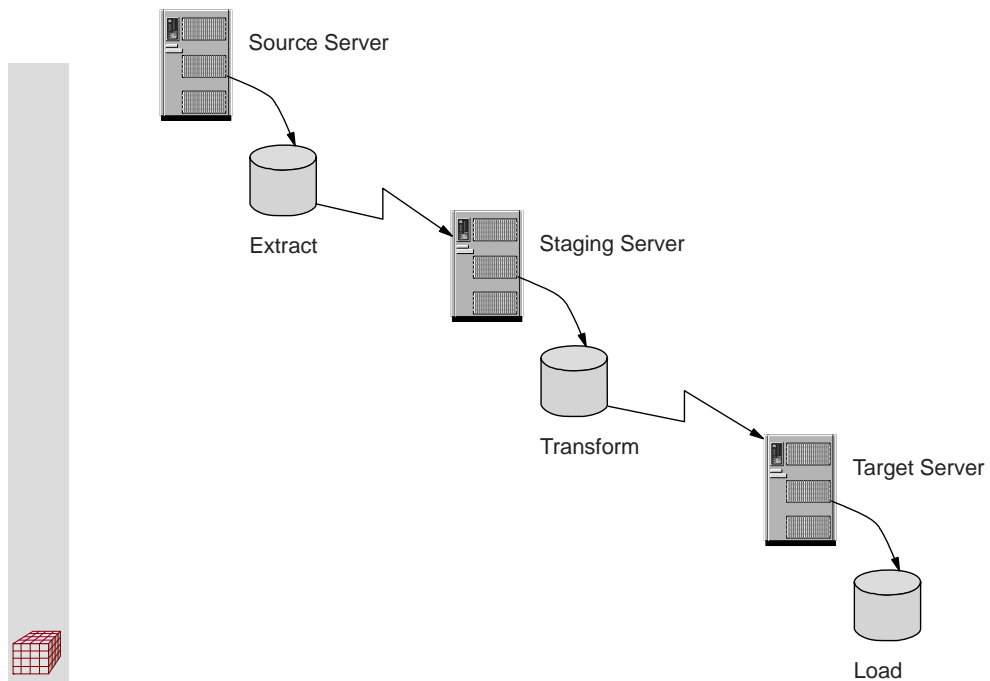
Lowest Common Denominator (LCD)

- If the grains of data in the source are different
 - ▶ Then you must find lowest common denominator (LCD)
 - ▶ Take incoming source amounts for Finance:
 - Actual and budget amounts are brought in for each ORG
 - ORG could be DIV or DEPT
 - BUDGET DOLLARS are available at the DIV level
 - ACTUAL DOLLARS are available at the DEPT level
- The lower level grain will have to be aggregated to the higher grain



▼ Data Movement Across Servers

- The stages of ETL might be done across several servers.



Referential Integrity (RI)

- Referential integrity ensures that when a dependent table refers to a parent table, the parent table exists
- The steps to handle this in the DW are:
 - Load dimensions first
 - Load facts next
 - Do RI in ETL, not in the DBMS
 - Keep RI off in the DW (most do)
 - For RI, either
 - Perform RI checks outside of DBMS (ETL) or
 - Perform RI checks after DW load (not common)
 - Build indexes after load



Load Technology

■ Loading using

- ETL tool
- DBMS load utility
- SQL SELECT INSERT
- IMPORT
- outside utility
 - e.g., Optiload for UDB EEE



Developing Quality Assurance Criteria

- Enforce a set of controls to ensure that the data is properly transformed and loaded.

Flash Totals

Source A Total Sales	DW Total Sales
\$56,675	\$56,675
Source # of Orders Processed	DW # of Orders Processed
110,765	110,765
DW Total Sales	DW Acceptable Range
\$56,675	\$50,000 - \$75,000

Completeness Checks

Reasonableness Checks



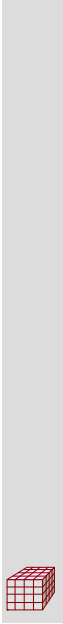
Key Terms

- Cleansing
- Initial load
- Scrubbing
- Data Quality
- Acquisition
- Gather
- ETL
- Extraction
- Change data



Exercise 8

- Do case study Exercise 8: ETL





Data Quality

- Essentially means that the data contains what it is supposed to contain.
- Isn't always provided from the operational environment.
- Can be detected by:
 - Data Quality Audits (ex post factum)
 - Data Quality Surveys (ex post factum and sometimes subjective)
 - Queries that return incorrect results !!!
- As we will see, data quality can be corrected at different points





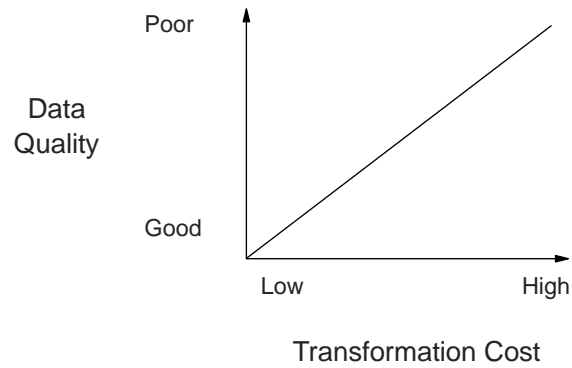
Data Quality in a Data Warehouse

- A data warehouse is a read-only environment
- The data in it is always aged in some way
 - It may be day-old data, or longer
 - In some environments, such as options trading, even 4-hour old data can be stale
- Some data is used for financial reporting
 - Usually, this data has to be fully accurate and of good quality
- Some is used for marketing
 - Possibly, this data can be imperfect and fudge factors can be built into querying to accommodate a percentage of bad data
 - Even this data has to of good quality if is to be used to support operational processes (such as lead generation)
- Consequently, a data warehouse has to be of an appropriate level of data quality



Complexity of a Data Warehouse

- Data quality is assumed by the users
- The architect must determine the best source system fit.
- The complexity and cost of the data warehouse transformation effort is directly proportional to the quality of the data





Seven Sources of Poor Data Quality

- 1. Entry quality: Did the information enter the system correctly at the origin?
- 2. Process quality: Was the integrity of the information maintained during processing through the system?
- 3. Identification quality: Are two similar objects identified correctly to be the same or different?
- 4. Integration quality: Is all the known information about an object integrated to the point of providing an accurate representation of the object?
- 5. Usage quality: Is the information used and interpreted correctly at the point of access?
- 6. Aging quality: Has enough time passed that the validity of the information can no longer be trusted?
- 7. Organizational quality: Can the same information be reconciled between two systems based on the way the organization constructs and views the data?

- Source: Melissa Data



Sample Data Quality Checks

- Data warehouse data must be checked for quality
- Is the data accurate?
 - Are names and addresses correct?
- Is the data complete?
 - All required fields are entered. No missing or null values.
 - Default values are available for required fields.
- Is the data consistent across applications and uses?
- Does redundant data exist?
 - If so, is it controlled redundancy?
 - Or, is it due to lack of process and system controls?
- Are there duplicate records?
 - Customer files or mailing lists littered with duplicate rows.
 - Is the data current?
 - Do processes exist to keep the data current and relevant?
- Is the data integrated or disparate (data in multiple, inconsistent places)?
 - Data that is not integrated creates a more complex transformation process.
- Does the data follow business rules?
 - Policy holder age cannot be greater than 120 years.
 - Profit = Sales - Cost
- Are data elements misused?
 - Address lines used to store contract information.



Strategy #1: Cleansing the Source

- The source is the best place to cleanse data. But there are often difficult challenges, including:
 - ▶ No one understanding the source system code.
 - ▶ No one wanting to spend money modifying the legacy code.
 - ▶ The source system code being suspect or unavailable.
- Alternative tactics that do not involve source system code:
 - ▶ Use stored procedures to cleanse data
 - ▶ Use message brokers to cleanse data for example, MQ Series/MQ Integrator
 - ▶ Use ETL tools to cleanse data



Strategy #2: Point of Integration

- Cleansing data during ETL - the point of integration.
- The transformation processes are common filtering points for cleansing source data.
- Its capabilities include:
 - Standardization of source data
 - Merging, consolidation, aggregation of source data
 - Scrubbing of data
- Tactics include:
 - Purchasing software tools
 - Writing necessary code



Strategy #3: Post-Load Cleansing

- Monitoring the data warehouse for dirty data is essential.
- There are two steps:
 - Identify dirty data. Requires code that monitors predefined criteria, for example:
 - Referential integrity
 - Column values
 - Implement the cleansing process.
- Issues to consider when monitoring the warehouse:
 - What is the resource overhead?
 - What actions are to be implemented if dirty data is found?
 - Will monitoring be restricted in the warehouse?



Terms

- Data quality
- Reasons for poor data quality
- System of Record



Exercise 9

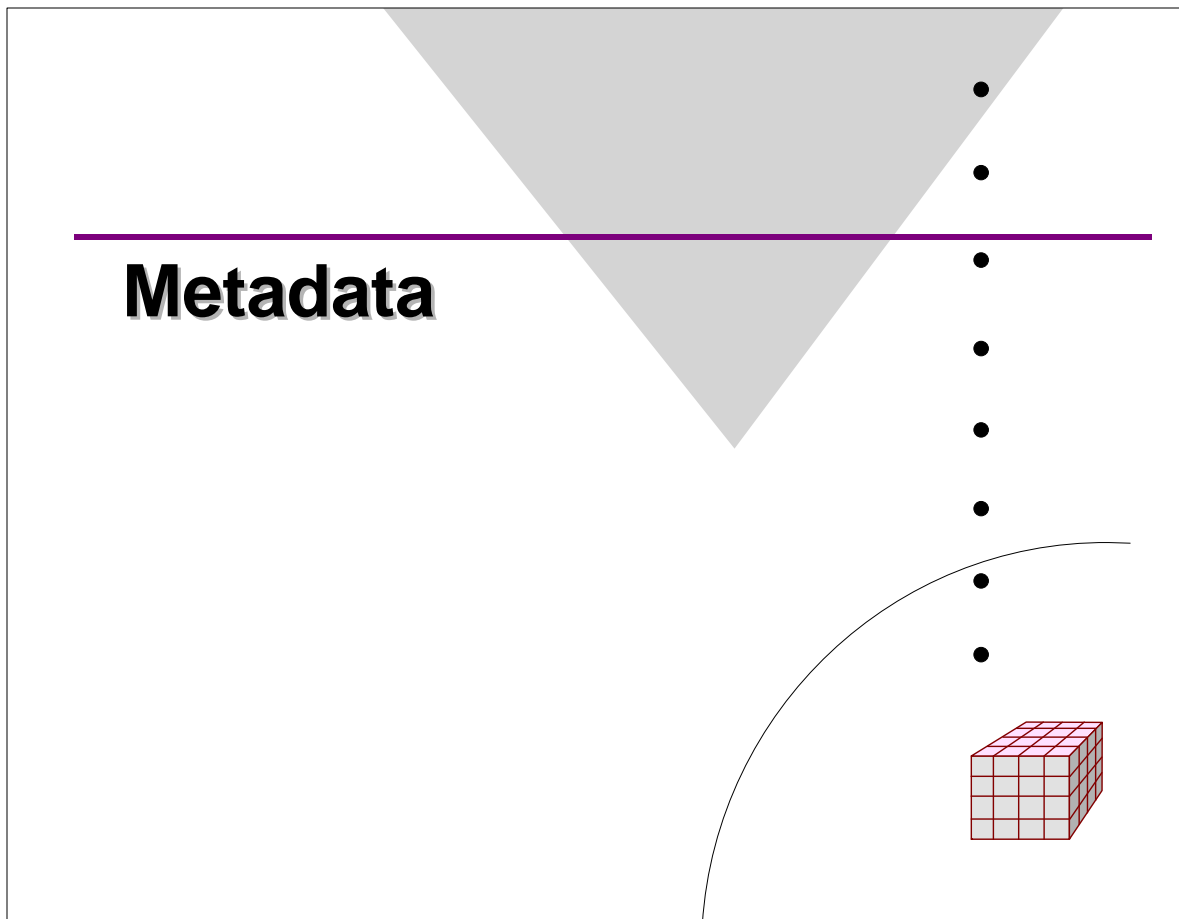
- Do case study Exercise 9: Data Quality





Notes





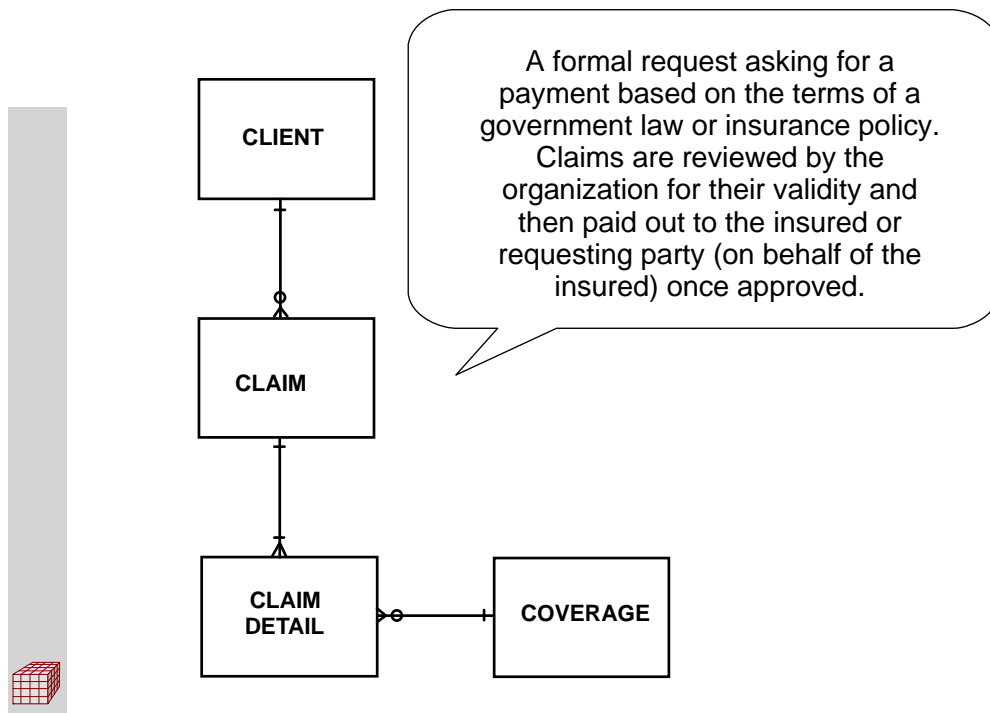


What Is Metadata?

- Data about Data
 - Data that describes the structure and business meaning of data stored in the CDW and data marts, as well as how it is created, accessed, and used.
- Purpose of metadata
 - Enable I/T professionals to manage data in the CDW and data marts on an ongoing basis
 - Enable business users to make full and consistent use of data that is stored in the warehouse and data marts
- You cannot effectively use a DW data unless you have superior metadata
 - Why?



Simple Metadata Example



Two Types of Metadata

■ Technical Metadata

- ▶ Enables I/T professional to manage the warehouse and data marts over time
- ▶ Technical Metadata contains two types of data:
 - Static: include roles and permissions, naming standards, source files/table definitions, Target table definitions, transformation and derivation rules, Source and Target mappings, and so on.
 - Dynamic (Ongoing): include extract, cleansing, and load schedules, load statistics, load rejects, backup statistics, and so on.

■ Business Metadata

- ▶ Structured to support end users' use and understanding of business data.
- ▶ Enables users to navigate through the data in the warehouse and data marts



Metadata Sources (1 of 2)

- Metadata comes from many, many potential sources
- Legacy Systems:
 - Data dictionary containing information about program libraries, database catalogs, file layouts. Program libraries, in turn, consist of data definitions (COBOL copybooks) and program modules. Database catalogs.
- Development Tools:
 - Modeling tools - contains both processing definitions and data definitions. Application Modeling tools, Spreadsheets, Lotus Notes.
- End users/Application Programmers:
 - Critical business information is not stored anywhere in the system but in the heads of the end users and I/T professionals who have worked on these systems for many years.



Metadata Sources (2 of 2)

■ ETL Tools:

- ▶ Definitions of target tables/dimensions, transformations - cleansing, derivation - aggregation and enrichment rules, scheduling information, load and error statistics.

■ DBMS for the Central DW and the Data Marts:

- ▶ DBMS system table components, partition settings, Indexes, DBMS security and privileges, View definition, Stored Procedures and SQL, DBMS administration statistics (backups and recovery)

■ BI Tools:

- ▶ Business names and descriptions for columns, tables and dimensions; Canned query and report definitions; Tools Security privileges.



Metadata Project

- Consider a project dedicated to providing metadata
- Goal: Corporate Wide Metadata
 - A single integrated source for corporate wide metadata
- Effort
 - Integrate metadata from disparate systems and tools
 - Very large and expensive effort
 - Very similar to building CDW, you must gather all the sources of metadata, refine it, and store it in a central repository
- Ongoing maintenance



Metadata Summary

- It is best to build a metadata repository for all metadata
- Build metadata like you build data warehouse
 - Iterative and evolving approach BUT
 - Adding the metadata as the model evolves
 - Not after it is deployed
- As you build data marts and integrate them to the Central Data Warehouse, in parallel build metadata for data marts and add it to the central data warehouse metadata repository.



Terms

- Metadata
- Repository
- Metamodel



Exercise 10

- Do case study Exercise 10: Metadata





Trends in Data Warehousing and Business Intelligence





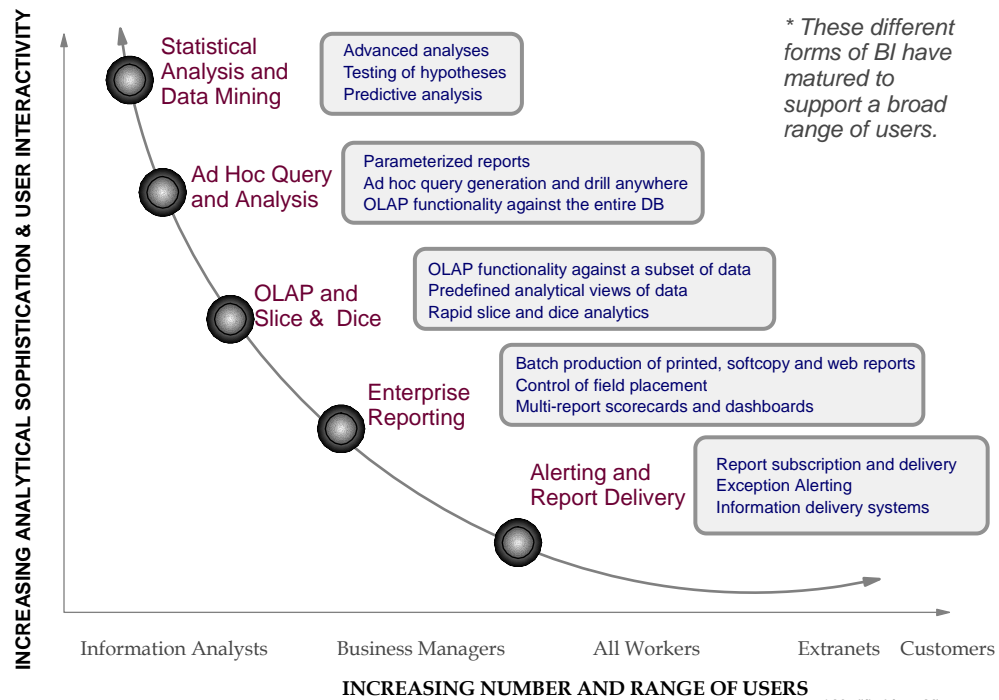
Some of the Major Trends

- Online analytical processing
- Dashboards
- Closed-loop data warehousing
- Hybrid data warehouse architectures
- Relational databases
- Parallel databases
- Column oriented databases
- Niche databases
- Big data technologies





Levels of Reporting and Analysis



** These different forms of BI have matured to support a broad range of users.*





5 Styles of Reporting

1. **Statistical Analysis and Data Mining** – Full mathematical, financial, and statistical treatment of data for purposes of correlation analysis, trend analysis, financial analysis and projections.

Targeted at the professional information analysts.

2. **Ad Hoc Query and Analysis** – Full investigative query into all data, as well as automated slice and dice OLAP analysis of the entire database – down to the transaction level of detail if necessary.

Targeted at information explorers and power users.

3. **Cube Analysis** – OLAP slice-and-dice analysis of limited data sets

Targeted at managers and others who need a safe and simple environment for basic data exploration within a limited range of data.

4. **Enterprise Reporting** – Broadly deployed pixel-perfect report formats for operational reporting and scorecards/dashboards.

Targeted at information consumers and executives.

5. **Alerting and Report Delivery** – Proactive report delivery and alerting to very large populations based on schedules or event triggers in the database.

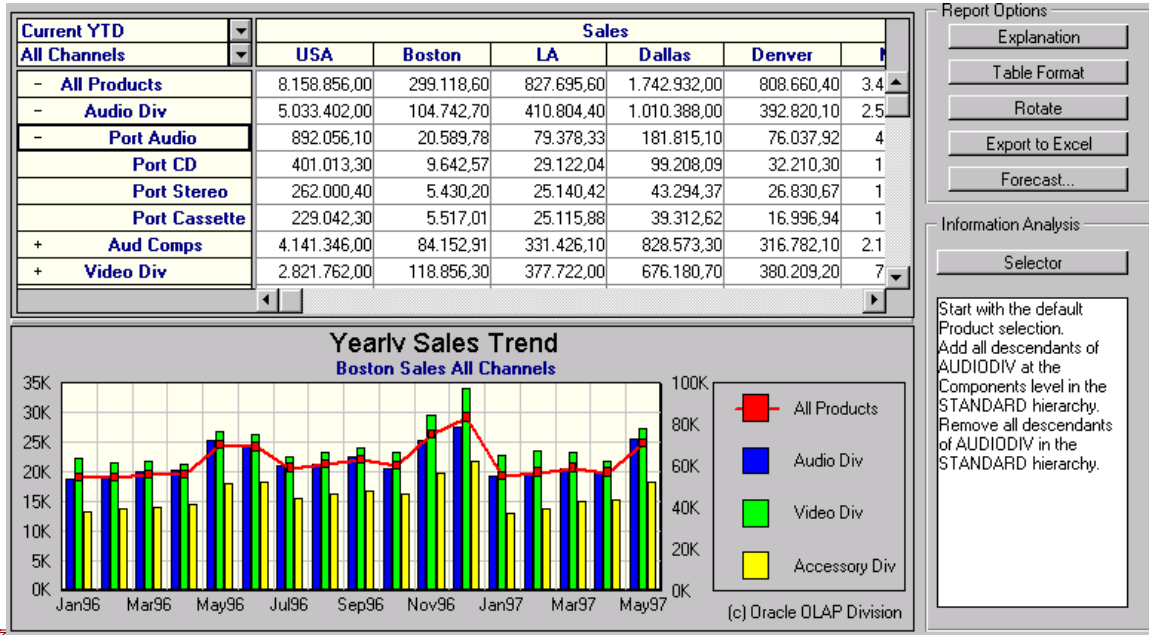
Targeted at very large user populations of information consumers, both internal and external to the enterprise.





Online Analytical Processing (OLAP)

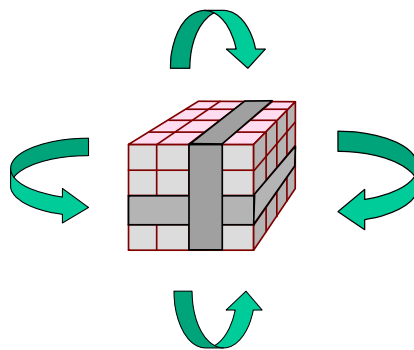
- OLAP products, especially multidimensional databases
- What the data looks like from the user perspective





On-Line Analytical Processing (OLAP)

- A technology that supports querying of informational data
 - Provides a division of data into facts (measures) and dimensions
 - May use a proprietary "cube" structure or a standard relational structure
 - Has tools and aids to facilitate query, analysis, reporting, slice and dice





What Is A Multidimensional DBMS?

Monthly Departmental Expenses

		Months										
		1	2	3	4	5	6	7	8	9	10	..
Depts: 109	.											

Month 6 Expenses
For Dept 109 = \$42K

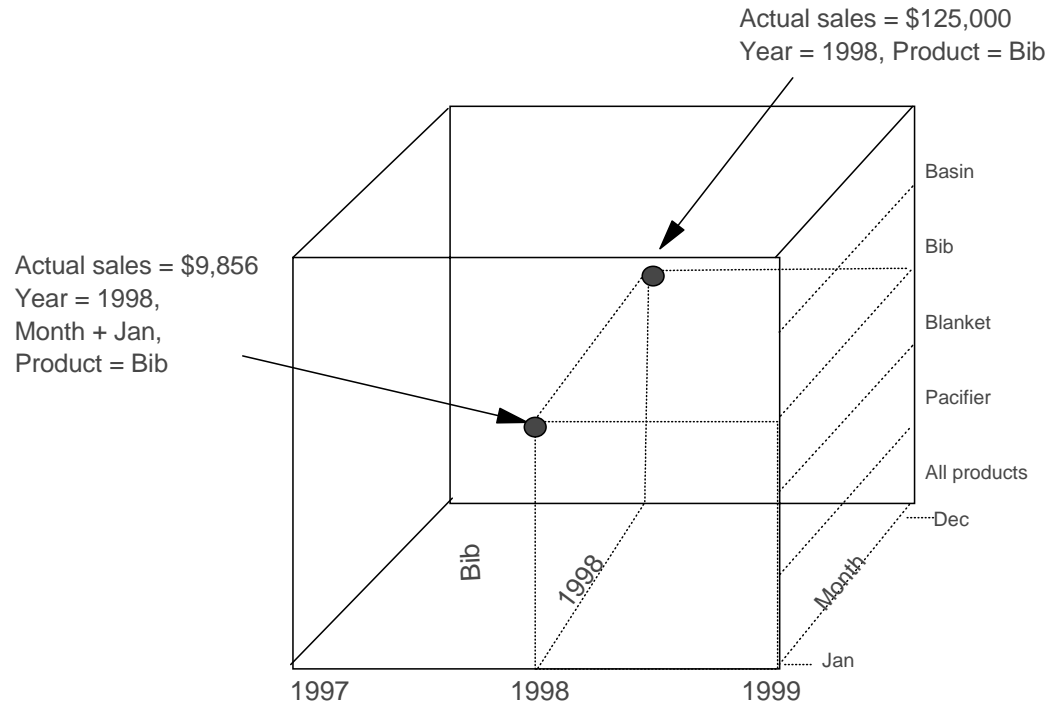


**A spreadsheet is multidimensional data
(2 dimensions)**



The Basic OLAP Cube

- A basic OLAP structure





Overall OLAP Method

- The following questions represent the major tasks in designing an OLAP solution:

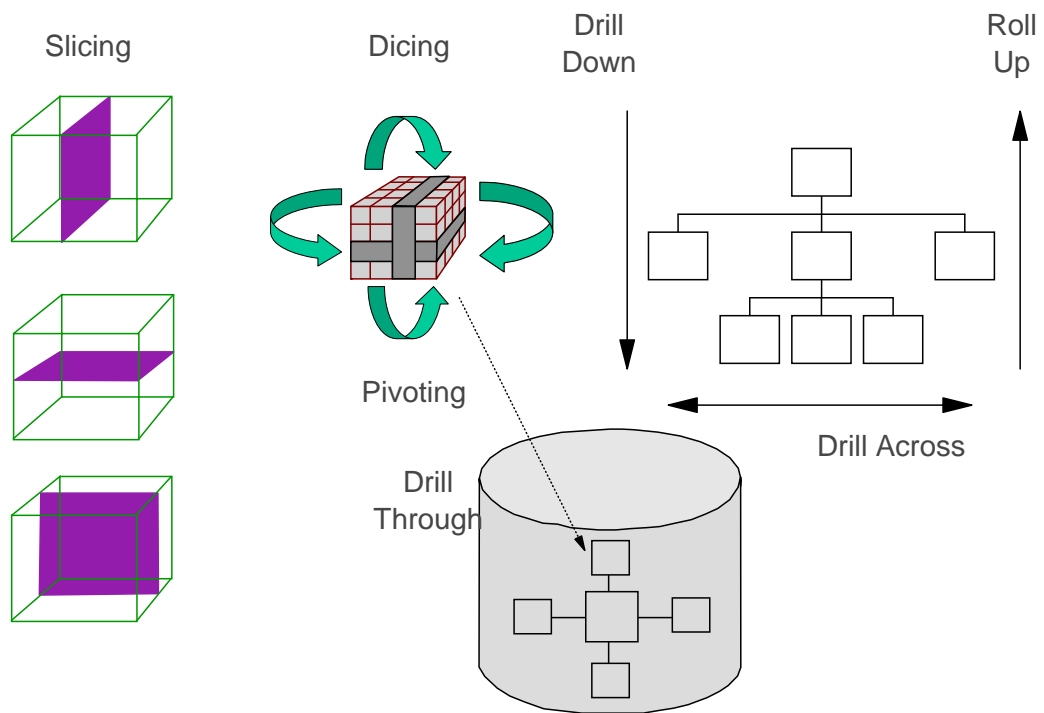
Which business process is being modeled?
What is being measured?
What level of detail?
What attributes qualify the measures?
What levels of summarization are needed?
How do you calculate the summaries?
Is there only one way to roll the data up?

Business Problem
Fact
Grain
Dimensions
Aggregates
Calculations
Hierarchies





OLAP Functions

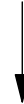




OLAP Drill Down/Drill Up

- Drill-down: navigating among levels of data ranging from the most summarized (up) to the most detailed (down) along a concept hierarchy. For example, you could drill down from Fiscal Year to Quarter.
- Drill-up: Rollup (opposite of Drill-Down).

Brand	Package Size	Sales
Softtowel	2-pack	\$75
Softtowel	2-pack	\$100
Softtowel	2-pack	\$50



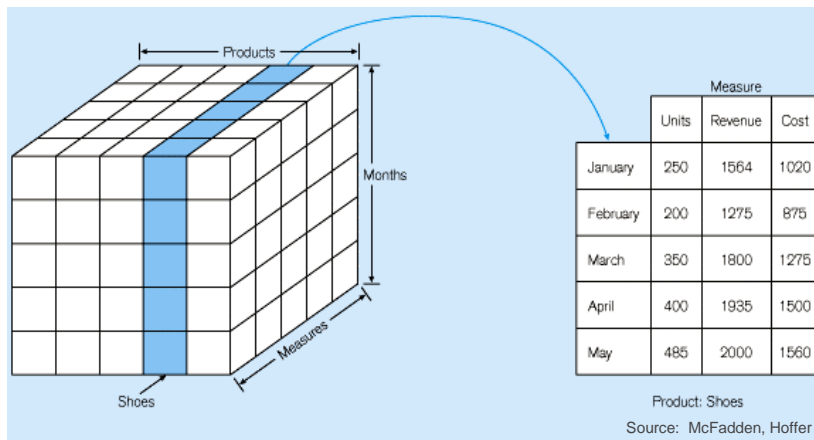
Brand	Package Size	Sales
Softtowel	2-pack	\$30
Softtowel	2-pack	\$25
Softtowel	2-pack	\$20
Softtowel	3-pack	\$50
Softtowel	3-pack	\$25
Softtowel	3-pack	\$25
Softtowel	6-pack	\$30
Softtowel	6-pack	\$20





OLAP Slice and Dice

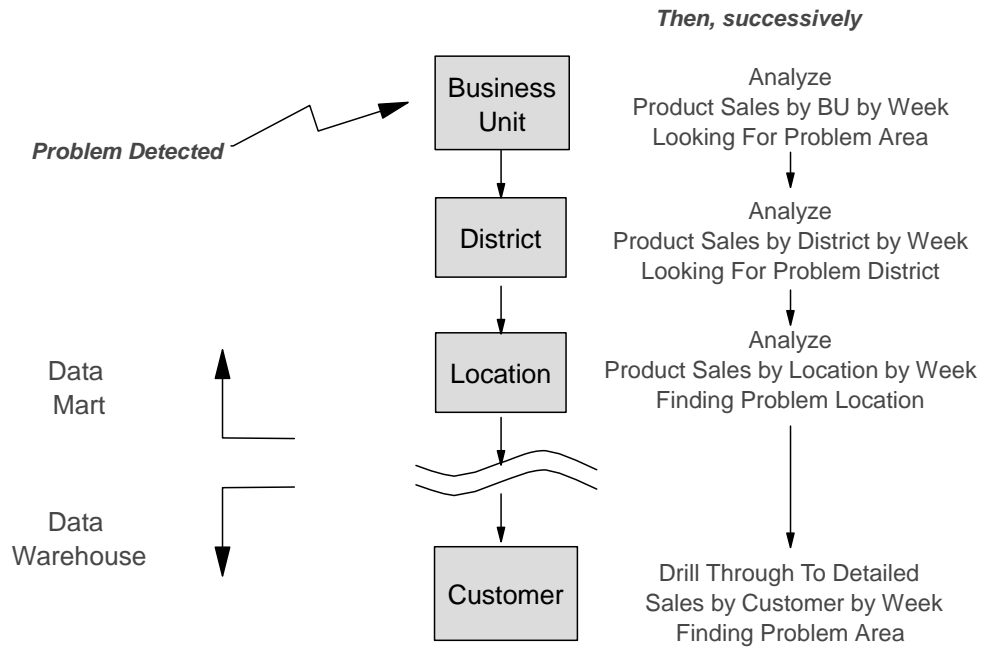
- Slice and Dice
 - Restructuring a report or query in order to analyze the data from different viewpoints.
 - An OLAP user-initiated process of navigating by calling for page displays interactively, through the specification of slices via pivoting and drilling.
 - Think of slicing then dicing a tomato.





OLAP Drill Through

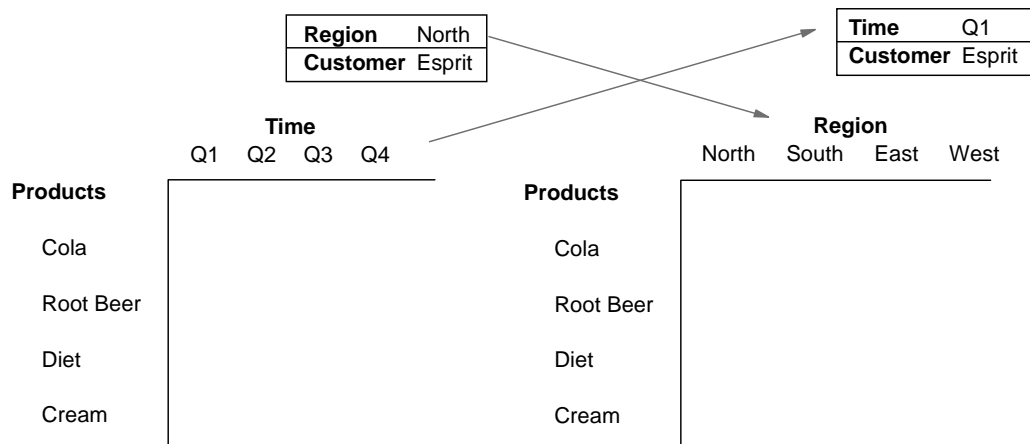
- Drill-through: Allows the user to access data through to an underlying relational data to obtain data that is not in the MDDB.





OLAP Pivoting

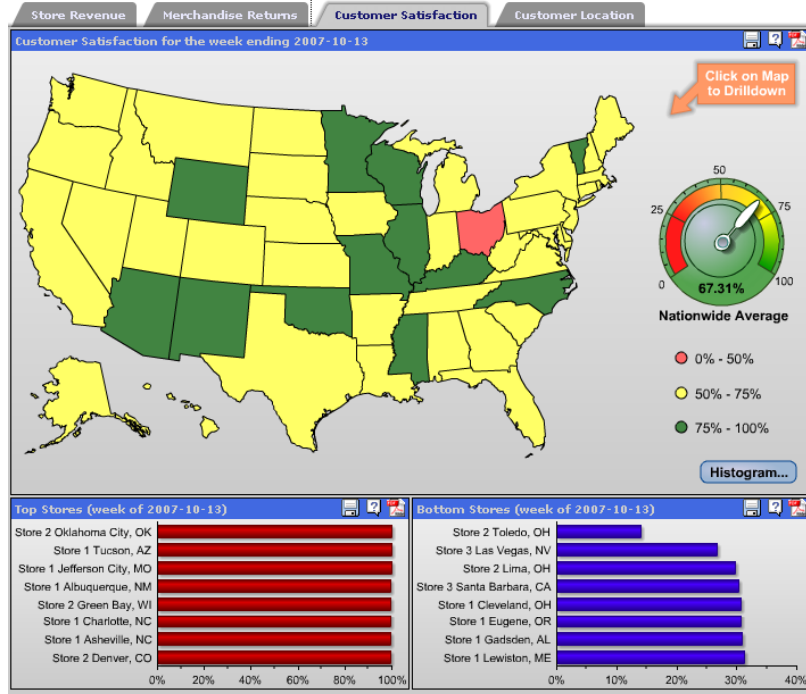
- The process of reconfiguring an axis of the table or graph being viewed.
- In pivoting, the user takes another viewpoint on the results of the analysis, changing the way the dimensions are arranged in the result.
- With pivoting, the answer cube is rotated and a different side is given priority in the presentation of the result.





Dashboards

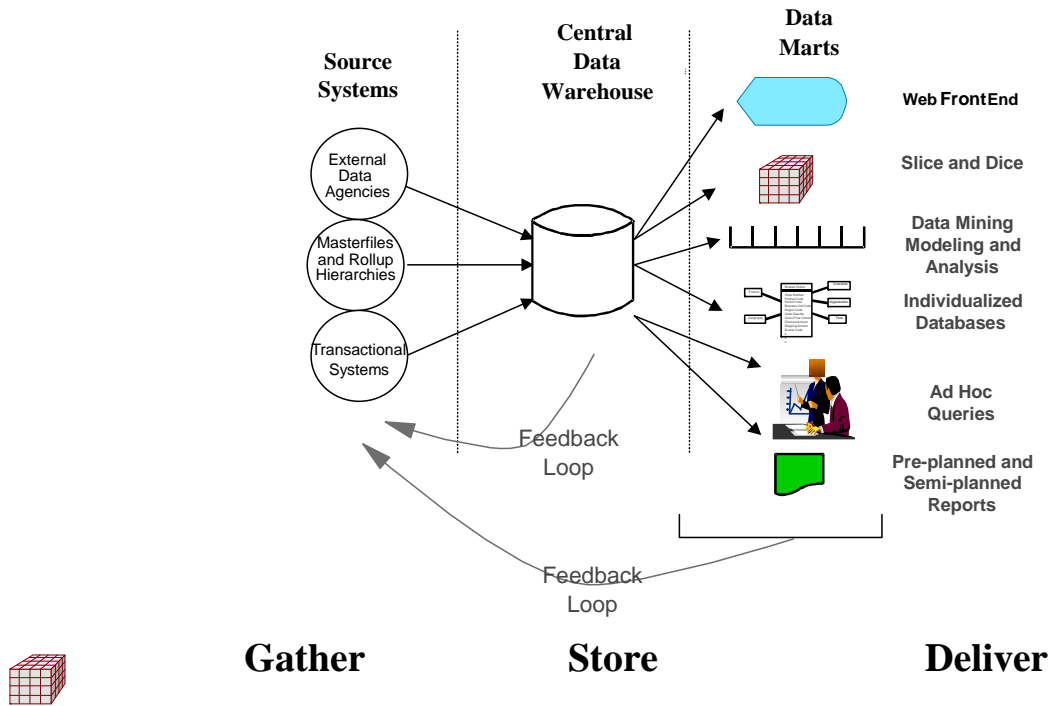
- Present data in an easy-to-understand format: the dashboard





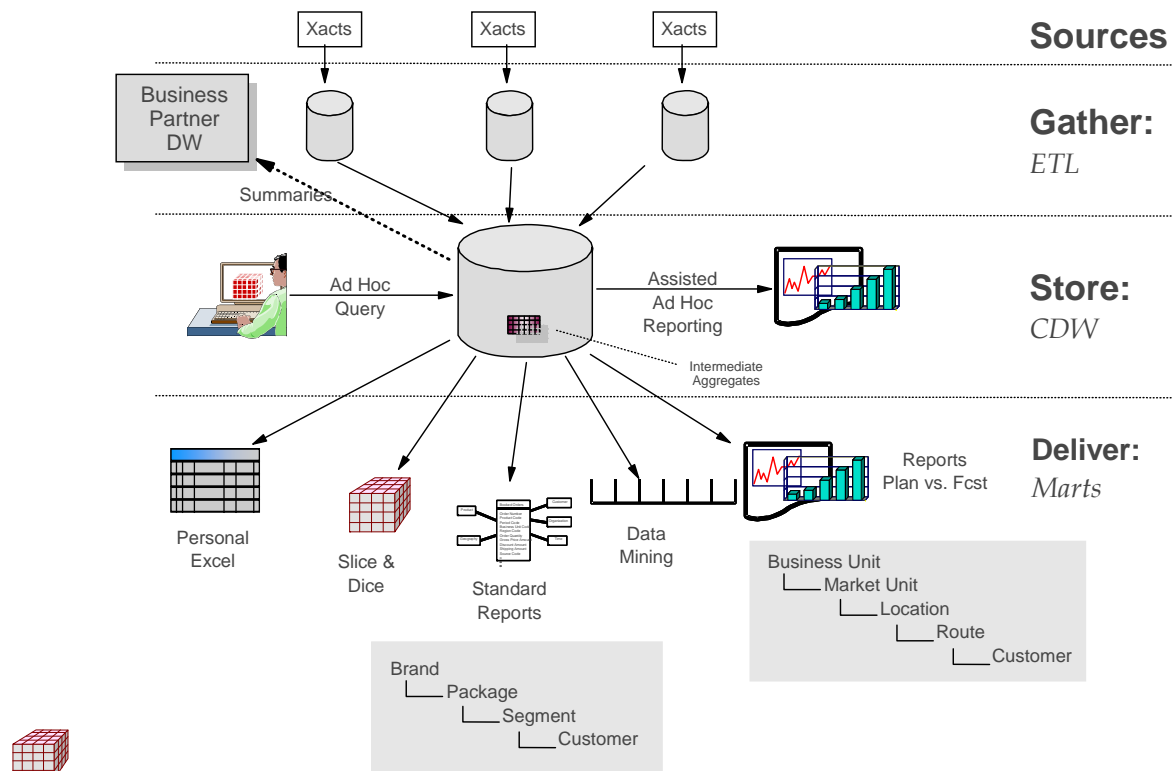
Closed Loop Data Warehousing

- Analytical data from the DW feeds back to operational systems and operational decision makers.





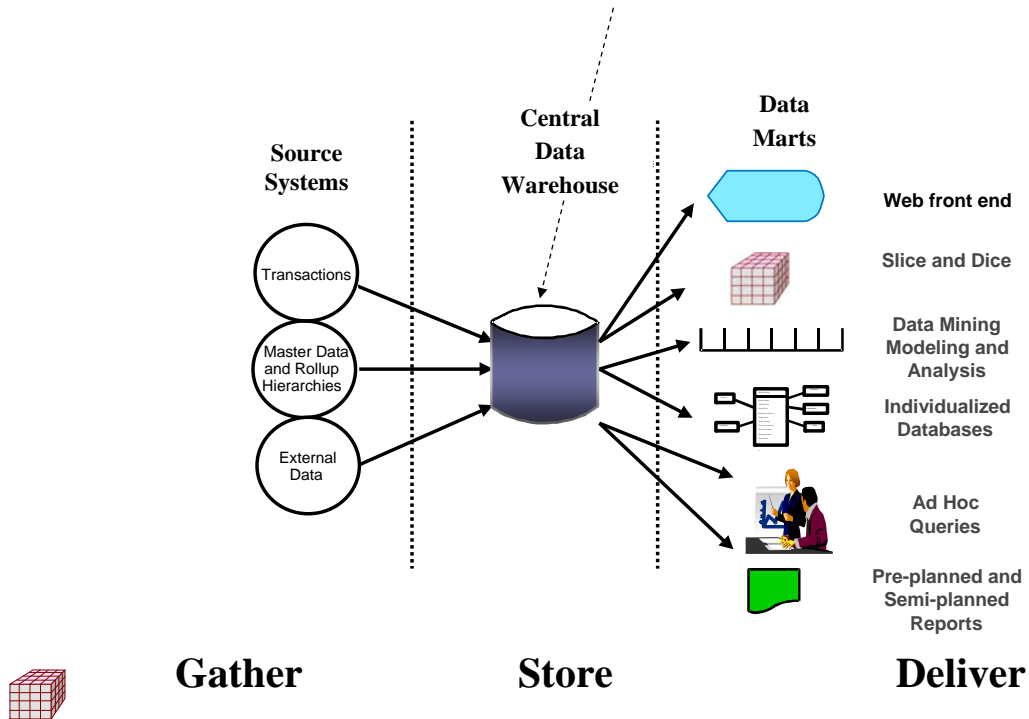
Practical Hybrid Data Warehousing





Relational Databases

- Continue to be the core of any large scale database solution





Parallel Processing Platforms

- Which way is easier to pick the ace of spaces?
- This? (all in one hand)



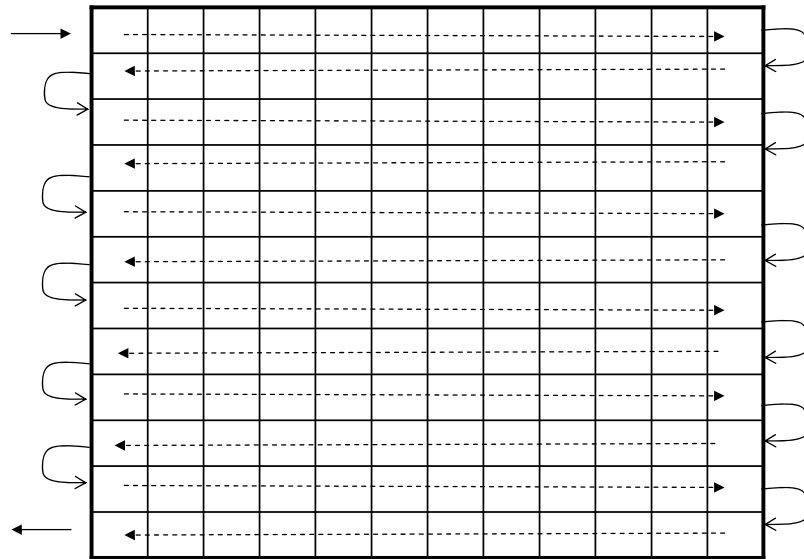
- Or this? (the same deck distributed to three people)





Scalability

- Provided by vendors like Teradata and commodity vendors like Netezza
- Scalability is achieved through horizontal distribution of data
- Data is distributed across many servers horizontally



© InfoModel, LLC. 2015

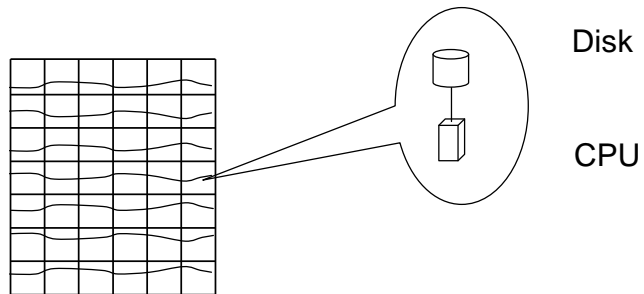


Interpret each cell
here as one node



How Do You Actually Get Data?

- The technology is called “shared nothing”
- Each node contains disk and processing power
- Data is spread out across all the nodes (servers)
- The same query is sent to each node
- Each node queries its piece of data and sends its result set to the coordinator node
- The coordinator node assembles the answer and presents it *
- The squiggle line indicates a fairly even distribution of data



* In Hadoop this is called Map : Reduce



Column Oriented Databases

	Col 1	Col2	Col3	Col4	Col5	Col6	Col7	Col8
Row 1								
Row 2								
Row 3								
Row 4								
Row 5								
Row 6								
Row 7								
Row8								

- In most RDBMSs, the entire row is read into a buffer and then sliced down to the attributes

- In columnar DBMSs, only the needed columns are read

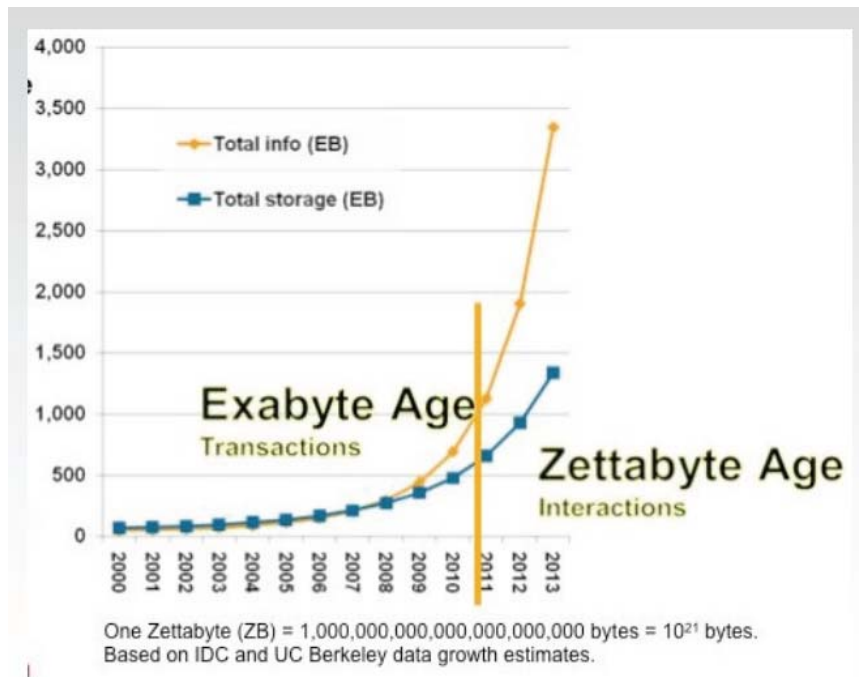
	Col 1	Col2	Col3	Col4	Col5	Col6	Col7	Col8
Row 1								
Row 2								
Row 3								
Row 4								
Row 5								
Row 6								
Row 7								
Row 8								



- Michael Stonebraker claims columnar databases are 50 **times** faster than any form of relational !



What Has Changed in Data?



<http://gigaom.com/cloud/sensor-networks-top-social-networks-for-big-data-2/>





Big Data Definition

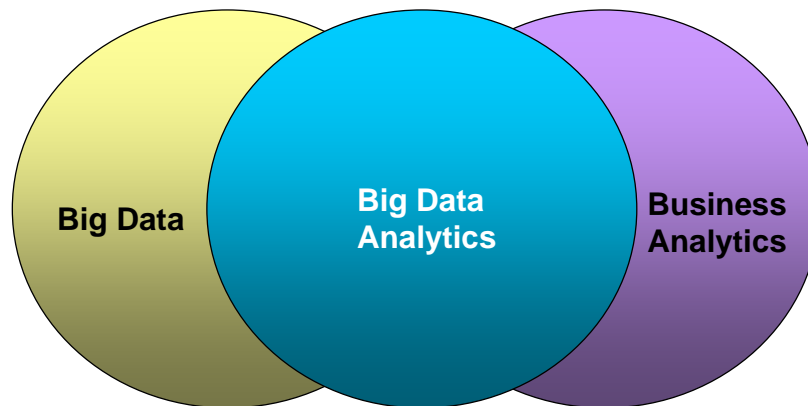
- **Big data** consists of high-volume, high-velocity, high-variety and *high value data and processes* that demand cost-effective, innovative forms of information processing for enhanced insight and decision making





Business Analytics

- Solutions used to build analytical, historical models and simulations to create scenarios, understand current status and predict future states
- Business analytics includes:
 - Data mining, predictive analytics, applied analytics and statistics, and is delivered as an application suitable for a business user.
 - Big Data Analytics is the convergence of Big Data and Business Analytics



- Without Big Data *Analytics*, big data is “just a lot of data”



Data Mining

- Data mining
 - Applies mathematical algorithms against vast amounts of data
 - Finds hidden relationships and value in the data
 - For example,
 - Discovering reasons for attrition
 - Finding new markets
 - Defining the customer market basket



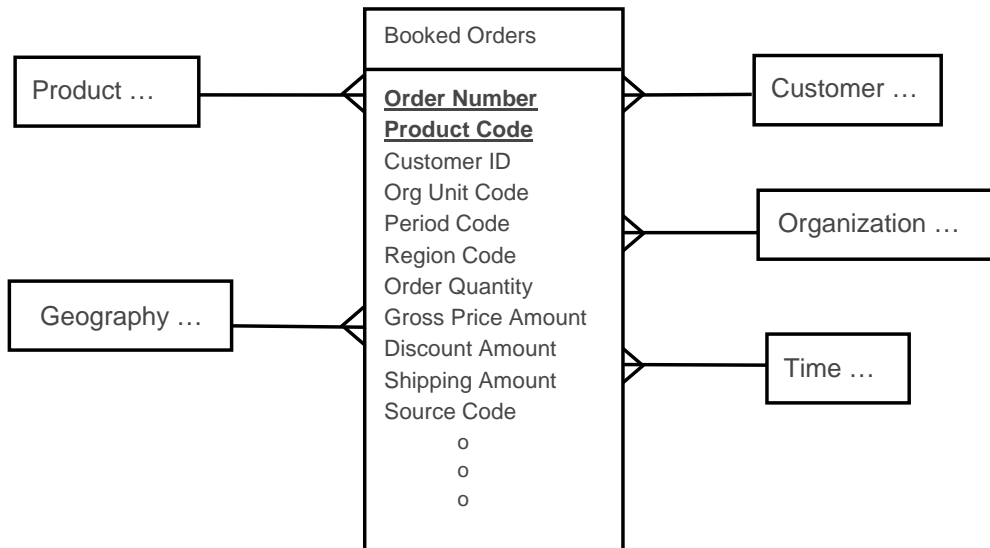
COURTESY: WALMART





Specialized Database Management Systems

- An example is Redbrick, which supports only the star schema
- Everything else is processed inefficiently
- Useful for reporting data marts





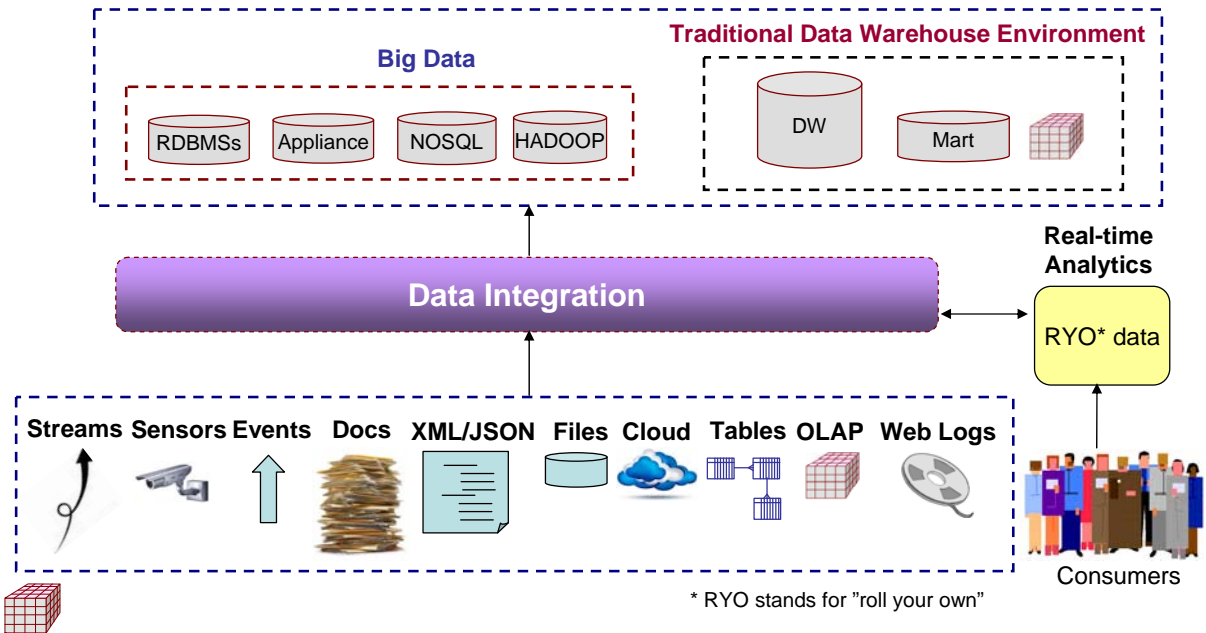
The Big Data Analytics Environment

Big Data Analytics

Complex analysis of structured data
Analysis of irregularly structured data in Hadoop
Social sentiment and social network analysis

Enterprise Data Warehouse Environment

Traditional Reporting and Analysis





Review

- Online analytical processing (OLAP)
- Dashboards
- Closed-loop data warehousing
- Hybrid data warehouse architecture
- Relational databases
- Parallel databases
- Column oriented databases
- Niche databases
- Big data technologies
- Analytics
- Data mining





Exercise 11

- Afterwards, review Exercise 11: Customer Attrition





Summary and Review

-
-
-
-
-
-
-



Objectives Review

- Upon completion of this course, the student will have the an understanding of the following :
 - What is a data warehouse?
 - The difference between operational and analytical data
 - The difference between a data warehouse and a data mart
 - Key issues and concerns in data warehouse projects
 - A general understanding of the different topologies for designing data warehouses
 - The types of questions asked to drive information gathering in data warehousing
 - The magnitude of and key issues in moving data from source systems
 - The key significance of data quality in data warehousing
 - What is dimensional modeling?
 - What is the central data warehouse?
 - The importance of metadata
 - Major trends in data warehousing today
 - Most of the major data warehousing terms and acronyms





Review of Course Content

1. Introduction to Data Warehousing
2. Data Warehouse Architectures
3. The DW Program
4. Gathering Information
5. Dimensional Modeling
6. The Central Data Warehouse (Store)
7. Extract-Transformation-Load (Gather)
8. Data Marts (Deliver)
9. Data Quality
10. Metadata
11. Trends in Data Warehousing





Last Page of Course

