

DATA WAREHOUSE AND OLAP TECHNOLOGY

PART - 1

By: Group No: 3

Rohan Sharma - 105370637

Kalpita Shah - 105370637

Yeshesvini Shirahatti - 105526740

Smruti Patel - 105390817



TOPICS

- Introducing the concept of a warehouse, modeling of data and schemas used.
- Rohan Sharma
- OLAP operations and Warehouse Architecture
- Kalpit Shah
- Research Paper on Distributed Warehouses
- Yeshesvini Shirahatti
- Application of Warehousing in Microsoft Terradata
- Smruti Patel



THE BASICS....

- What is a Data Warehouse??
- What is OLAP??
- Why do we need a separate Data Warehouse??
- How do we model a Warehouse??

Covered by:

Rohan Sharma



References:

Data Mining: Concepts and Techniques

- Jiawei Han , Micheline Kamber

What are Operational DBMS?



- They consist of Tables having attributes and are populated by tuples.
- They generally use the E-R data model.
- It is used to store transactional data.
- The information content is generally recent.
- These are thus called as OLTP systems.
- Their goals are data accuracy & consistency , Concurrency , Recoverability, Reliability (ACID Properties).

What is a Data Warehouse?



- **Formal Definition:** “ A data warehouse is a *subject-oriented, integrated, time-variant* and *non-volatile* collection of data in support of management decision making process.”

WHAT????

- It means:
 - **Subject-Oriented:** Stored data targets specific subjects.
Example: It may store data regarding total Sales, Number of Customers, etc. and not general data on everyday operations.
 - **Integrated:** Data may be distributed across heterogeneous sources which have to be integrated.
Example: Sales data may be on RDB, Customer information on Flat files, etc.
 - **Time Variant:** Data stored may not be current but varies with time and data have an element of time.
Example: Data of sales in last 5 years, etc.
 - **Non-Volatile:** It is separate from the Enterprise Operational Database and hence is not subject to frequent modification. It generally has only 2 operations performed on it: *Loading of data* and *Access of data*.

Contd...



Features of a Warehouse:

- It is separate from Operational Database.
- Integrates data from heterogeneous systems.
- Stores HUGE amount of data, more historical than current data.
- Does not require data to be highly accurate.
- Queries are generally complex.
- Goal is to execute statistical queries and provide results which can influence decision making in favor of the Enterprise.
- These systems are thus called Online Analytical Processing Systems (OLAP).

Operational DBS vs. Warehouses



- **Data Contents:**

Operational DB Systems: Current and detailed data and is subject to modifications.

Data Warehouse: Historical data, coarse granularity, generally not modified.

- **Users:**

Operational DB Systems: Customer – Oriented, thus used by customers/clerks/IT professionals.

Data Warehouse: Market – Oriented, thus used by Managers/Executives/Analysts.

- **Database Design:**

Operational DB Systems: Usually E-R model.

Data Warehouse: Usually Multidimensional model. (Star, Snowflake...)

- **Nature of Queries:**

Operational DB Systems: Short, atomic queries desiring high performance (less latency) and accuracy.

Data Warehouse: Mostly read only queries, operate on HUGE volumes of data, queries are quite complex.

Why have a separate Warehouse?



3 Main reasons:

1. OLTP systems require high concurrency, reliability, locking which provide good performance for short and simple OLTP queries. An OLAP query is very complex and does not require these properties. Use of OLAP query on OLTP system **degrades its performance**.
2. An OLAP query reads HUGE amount of data and generates the required result. The query is very complex too. Thus **special primitives** have to be provided to support this kind of data access.
3. OLAP systems access historical data and not current volatile data while OLTP systems access current up-to-date data and do not need historical data.

Thus,

Solution is to have a separate database system which supports primitives and structures suitable to store, access and process OLAP specific data ...
in short...have a data warehouse.

What Data is stored in a Warehouse?



- In simple words: Subject(s) per Dimension
Example: If our subject/measure is 'quantity sold' and if the dimensions are : Item Type, Location and Period then,
Data warehouse stores the items sold per type, per geographical location during the particular period.

How do we represent this data???

Data Cube



- This multi-dimensional data can be represented using a data cube as shown below.

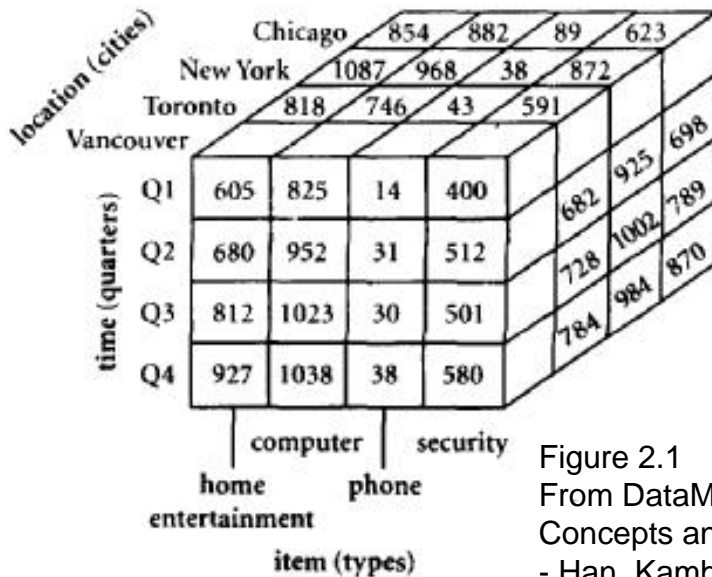


Figure 2.1
From DataMining:
Concepts and tech.
- Han, Kamber

This figure shows a 3-Dimensional data Model.

X – Dimension : Item type

Y – Dimension : Time/Period

Z – Dimension : Location

Each cell represents the items sold of type 'x', in location 'z' during the quarter 'y'.

This is easily visualized as Dimensions are 3.

- What if we want to represent the store where it was sold too?
- We can add more dimensions. This makes representation complex.
- Data cube is thus a n - dimensional data model

Schemas for Multidimensional data...



The well known schemas are:

1. **Star Schema**: Single Fact table with n – Dimension tables linked to it.
 2. **Snowflake Schema**: Single Fact table with n -Dimension tables organized as a hierarchy.
 3. **Fact Constellation Schema**: Multiple Facts table sharing dimension tables.
- Each Schema has a Fact table that stores all the facts about the subject/measure.
 - Each fact is associated with multiple dimension keys that are linked to Dimension Tables.

Star Schema



- There is a central large Fact table with no redundancy
- Each tuple in the fact table has a foreign key to a dimension table which describes the details of that dimension

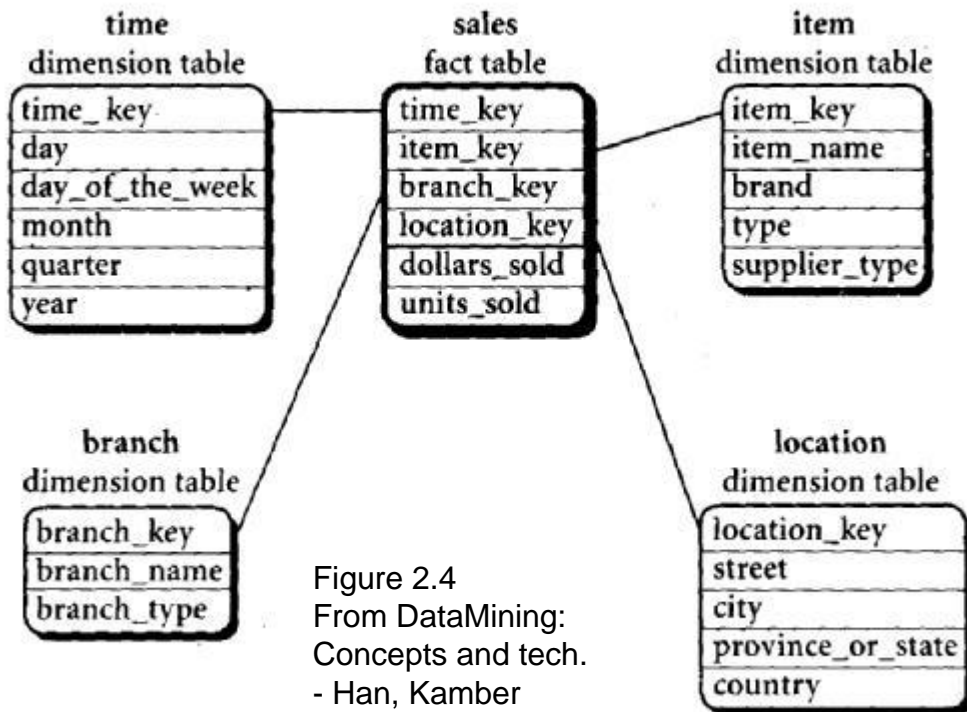
Problem: Redundancy

- Values of city, province_or_state and country would be repeated for two streets in the same city.

Thus we can normalize the table by splitting location into sub tables. (Snowflake Schema)

Advantage: Performance

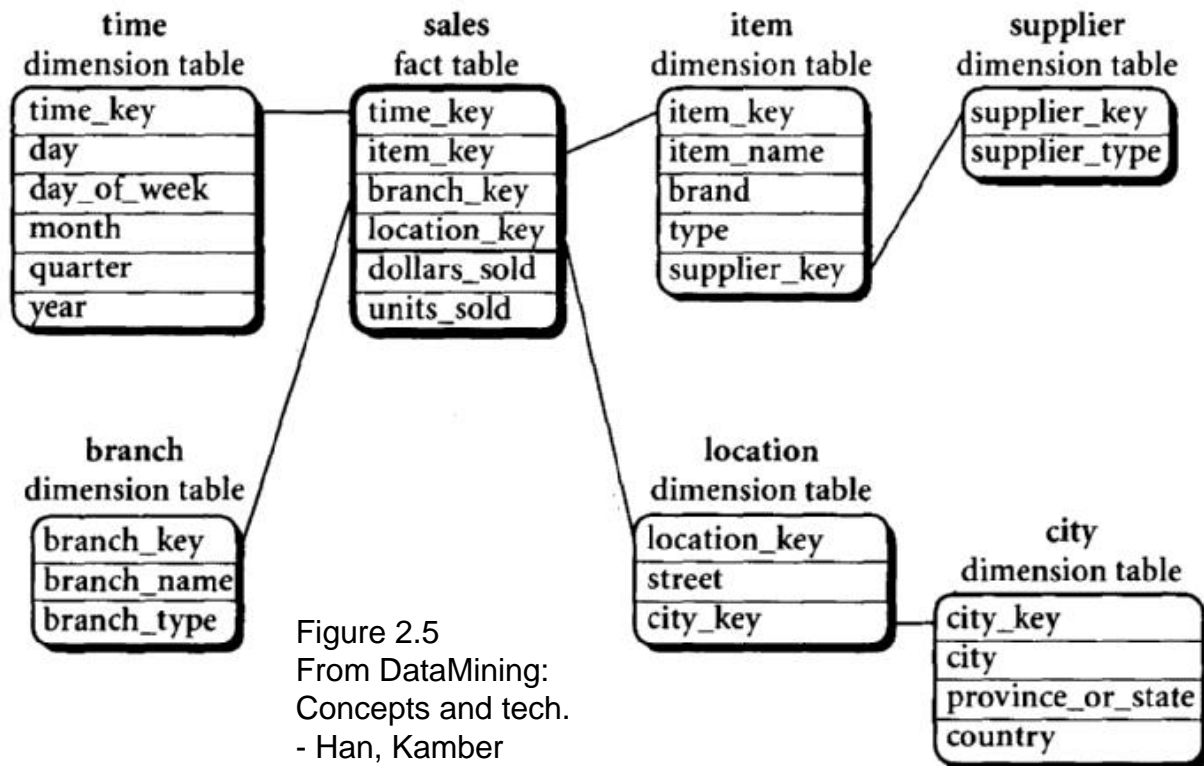
As less number of joins required



Snowflake Schema



Some of the dimension tables are normalized thus splitting data into additional tables.



Problem: Performance

- Too many joins required to form the result.

Thus Snowflake schema is not as popular as the Star schema.

Fact Constellation Schema



Two or more fact tables share dimension tables.
In the figure below the 'Sales' fact table and 'Shipping' fact table
Share the dimension tables

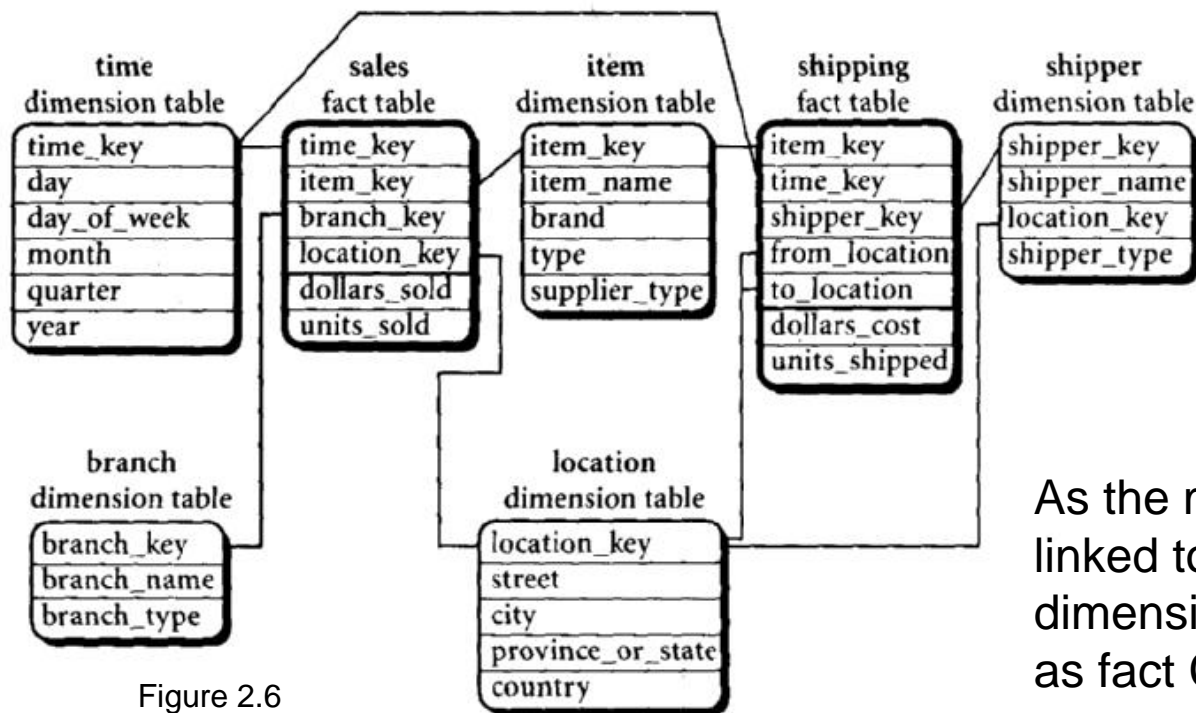


Figure 2.6
From DataMining:
Concepts and tech.
- Han, Kamber

As the multiple fact tables are linked to each other by dimension tables, its called as fact Constellation Schema

DMQL DDL Primitives



- Data Mining Query Language (DMQL) Syntax:

1. define cube <cube name>[<dimension list>]:<measure list>
2. define dimension <dimension name> as (<attribute list>)

- Star Schema Example:

Fact Table:

```
define cube sales_star [time,item,branch,location]:  
dollars_sold=sum(sales_in_dollars),units_sold=count(*)
```

Dimensions:

```
define dimension time as (time_key,day,day_of_week,month,quarter,year)  
define dimension item as (item_key,item_name,brand,type)  
define dimension branch as (branch_key,branch_name,branch_type)  
define dimension location as (location_key,street,city,province,country)
```

Contd...



- Defining a hierarchy of dimension tables for snowflake schema.

define dimension *location* as
(location_key,street,city(city_key,city,province,country))

- Defining a shared dimension table for Fact Constellation Schema

define dimension *time* as (time_key,day,day_of_week,month,quarter,year)
define dimension *time* as *time* in cube *sales*

Coming up ... OLAP Techniques & Warehouse Architecture - Kalpit Shah

Data Warehouse and OLAP Technology



- ❖ Concept Hierarchies
- ❖ OLAP operations in Multidimensional Data Model
- ❖ Query Model for Multidimensional Databases

- Kalpit Shah

Concept Hierarchies



It is a sequence of mappings from a set of low-level concepts to higher-level, more general concepts

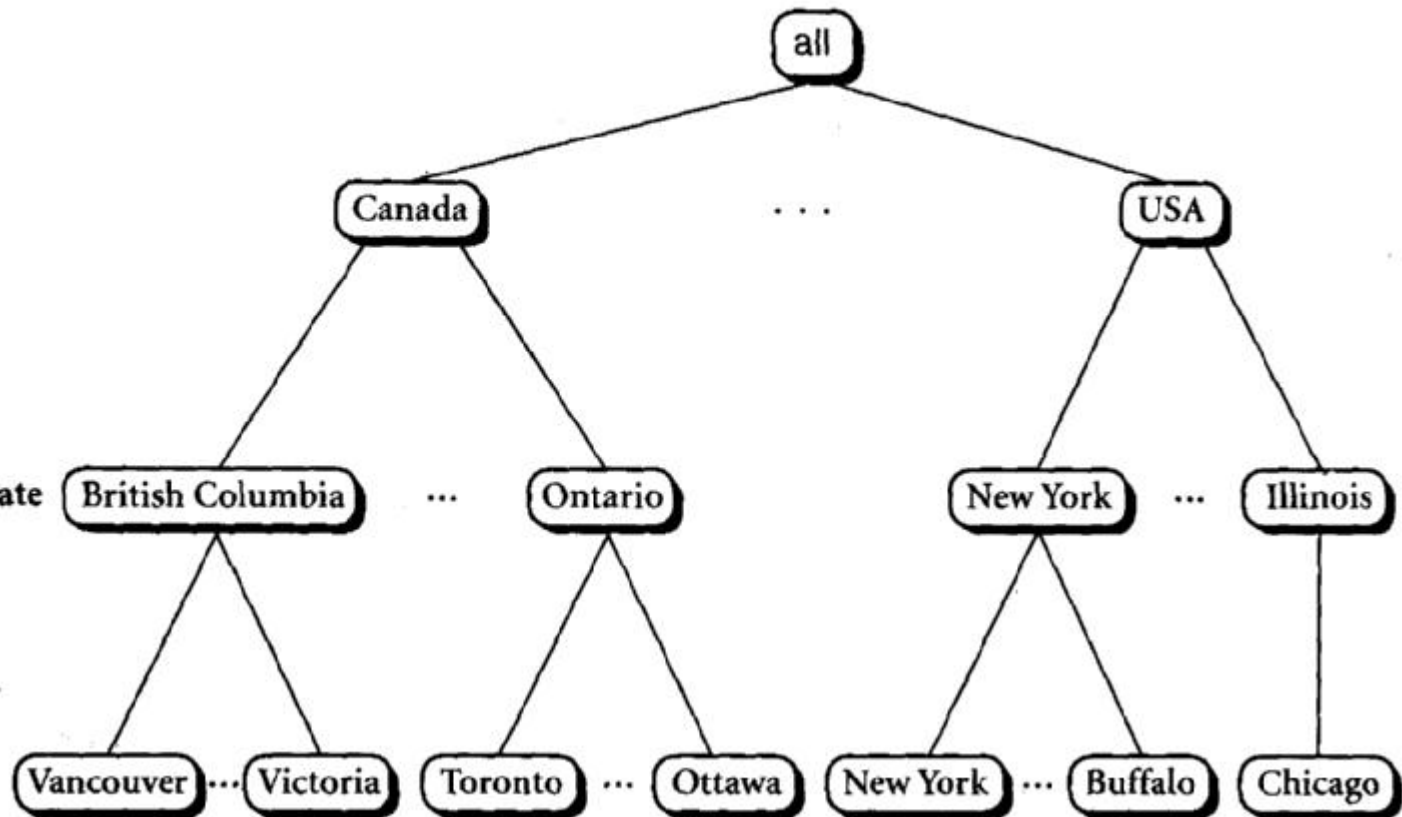
location

all

country

province_or_state

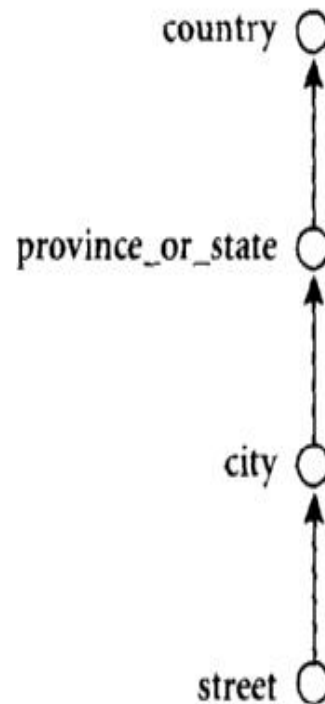
city



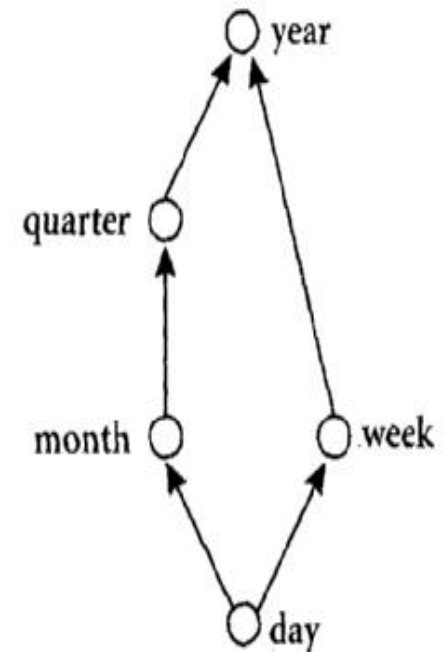
Concept Hierarchies



- A Concept Hierarchy may also be a total order or partial order among attributes in a database schema
- It may also be defined by discretizing or grouping values for a given dimension or attribute, resulting in a **set-grouping** hierarchy
- Concept Hierarchies may be provided manually by
 - System users
 - Domain Experts
 - Knowledge Engineers
 - Automated Statistical Analysis



(a)



(b)

OLAP Operations



1. **Roll-up**

Performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by dimension reduction

2. **Drill-down**

Can be realized by either stepping down a concept hierarchy for a dimension or introducing additional dimensions

3. **Slice and Dice**

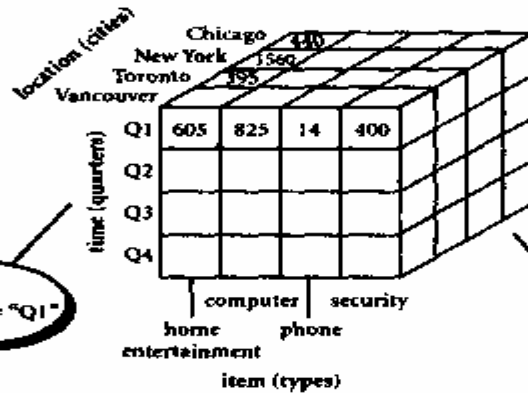
Slice performs a selection on one dimension of the given cube, resulting in a sub cube

Dice defines a subcube by performing a selection on two or more dimensions

4. **Pivot (rotate)**

It's a visualization operation that rotates the data axes in view in order to provide an alternative presentation of the data

OLAP Operations



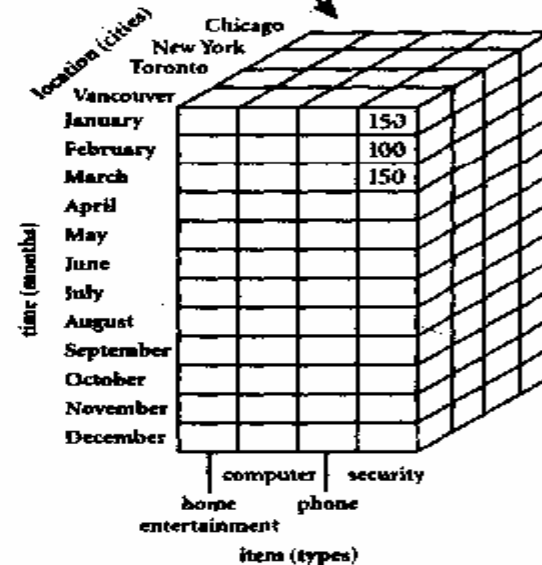
slice
for time = "Q1"



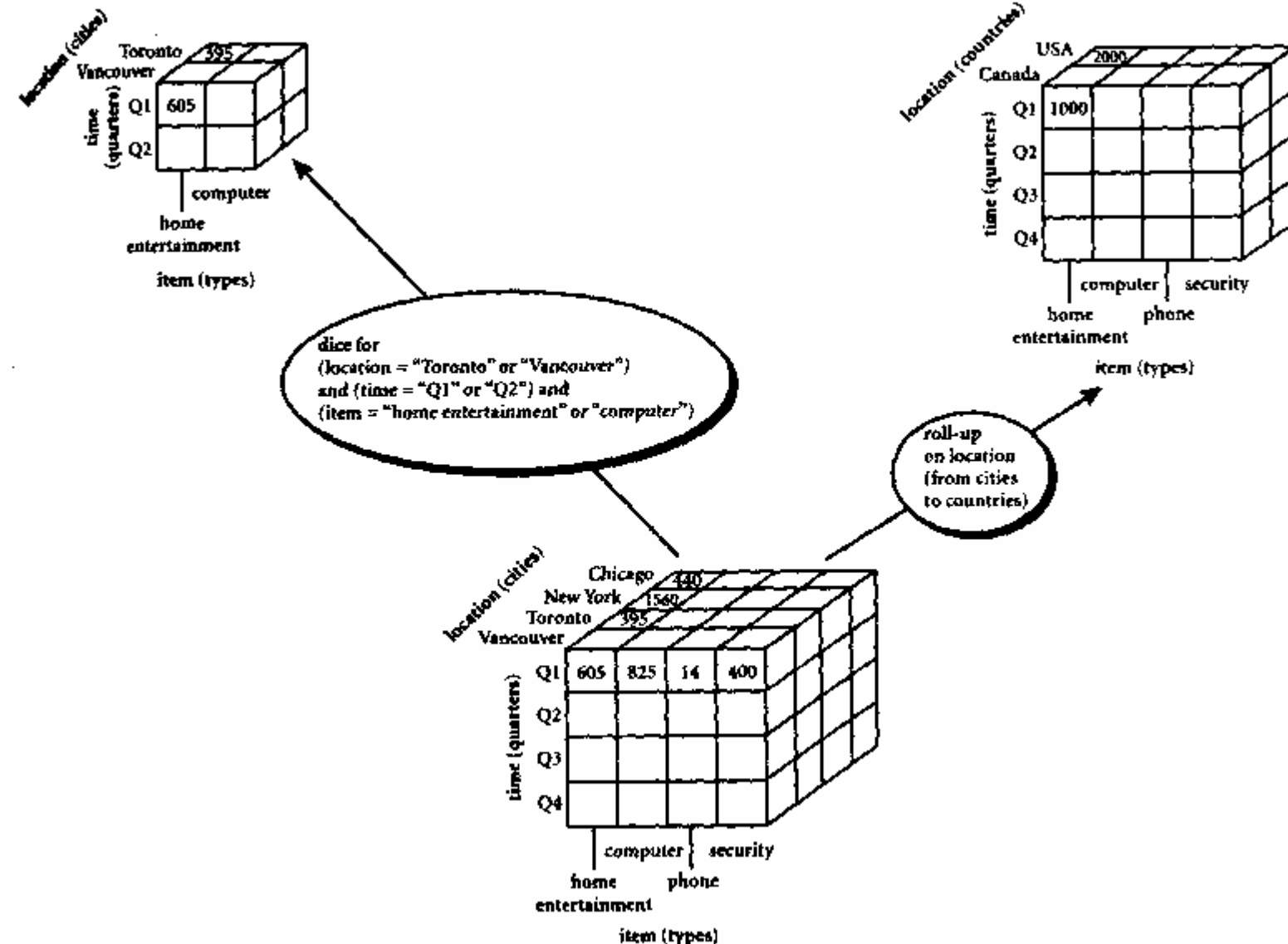
pivot



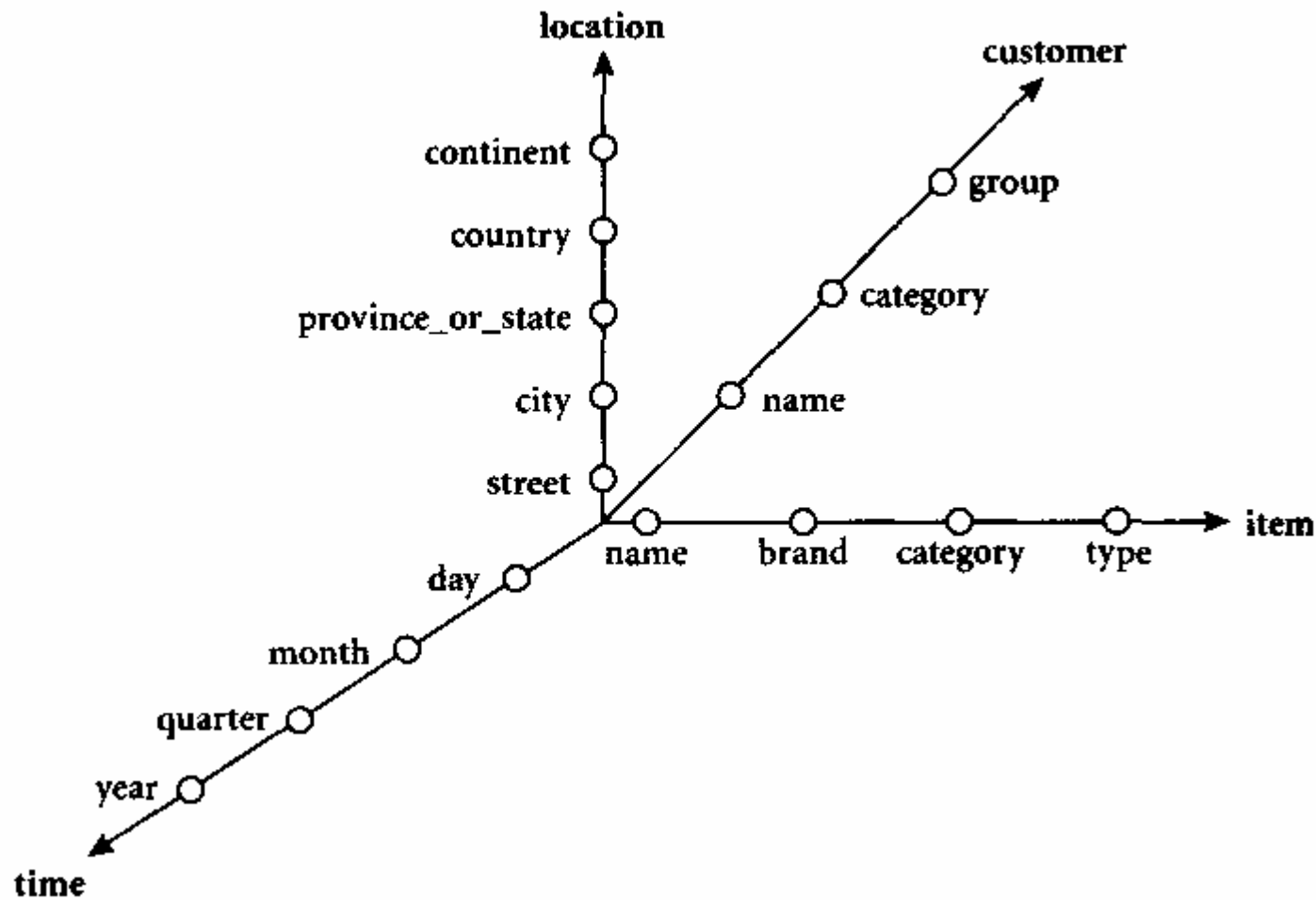
drill-down
on time
(from quarters
to months)



OLAP Operations



Starnet Query Model



Data Warehouse Architecture

1. Design and Construction

2. A Three - Tier Architecture

3. OLAP Servers



Data Warehouses - Design and Construction



What does the data warehouse provide for BUSINESS ANALYSTS?

- Presents relevant information useful for measuring performance and evaluation issues
- Enhances business productivity by quick and efficient gathering of information
- Facilitates customer relationship management by providing a consistent view of customers across all lines of business, all departments, and all markets
- Brings cost reduction by tracking trends, patterns and exceptions overlong periods of time in a consistent and reliable manner

Designer Views



1. Top-Down View

- Allows the selection of relevant information necessary
- This information matches the current and coming business needs

2. Data Source View

- Exposes the information being captured, stored and managed by operational systems
- This view is often modeled by traditional data modelling techniques such as ER Model or CASE tools

3. Data Warehouse View

- Includes fact tables and dimension tables
- Represents precalculated totals and counts
- Provides historical context

4. Business Query View

- It's the perspective of a data in the warehouse from the viewpoint of the end user

Data Warehouse Design - The Process



1. Top-Down Approach

- Starts with overall design and planning
- Useful where the technology is mature and well known
- Useful where the business problems to be solved are clear and well understood

2. Bottom-Up Approach

- Starts with experiment and prototypes
- Allows an organization to move forward at considerably less expense
- Allows to evaluate the benefits of the technology before making significant commitments

3. Combined Approach

- Exploits the planned and strategic nature of the top-down approach
- Retains the rapid implementation and opportunistic application of the bottom-up approach

The Software Engineering Perspective



It involves the following steps :

- Planning
- Requirements Study
- Problem Analysis
- Warehouse Design
- Data Integration and Testing
- Deployment of Warehouse

Development Model

Waterfall Model

- Performs a systematic and structured analysis at each step before proceeding to the next

Spiral Model

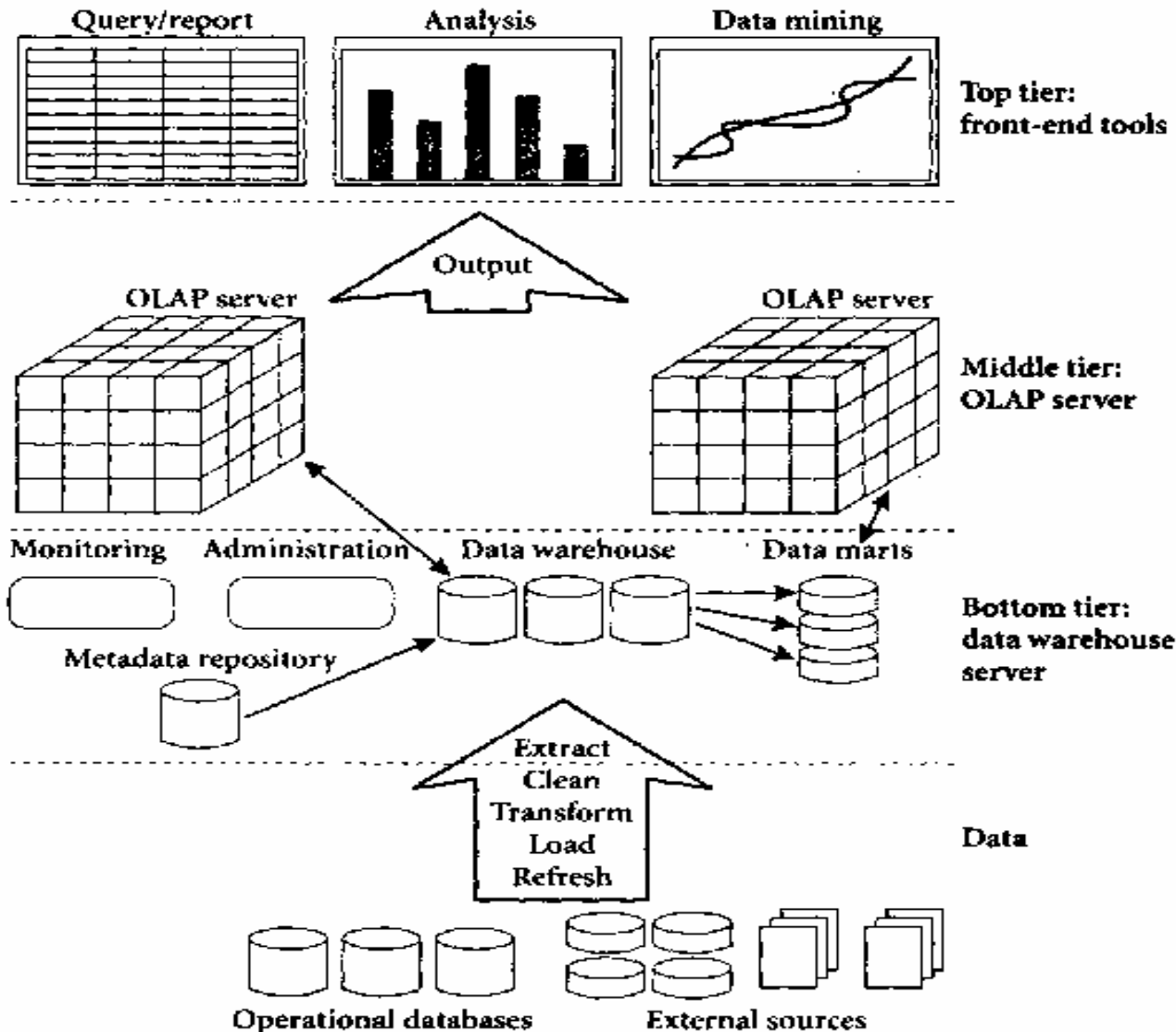
- Involves rapid generation of increasingly functional systems, with short intervals between successive releases

Data Warehouse Design - The Steps



1. Choose a business process to model, for example, orders, invoices, shipments, inventory, account administration, sales, and the general ledger.
2. Choose the grain of the business process. The grain is the fundamental, atomic level of data to be represented in the fact table for this process, for example, individual transactions, individual daily snapshots, and so on.
3. Choose the dimensions that will apply to each fact table record. Typical dimensions are time, item, customer, supplier, warehouse, transaction type and status.
4. Choose the measures that will populate each fact table record. Typical measures are numeric additive quantities like *dollars_sold* and *units_sold*.

Three - Tier Warehouse Architecture



Data Warehouse Models



Enterprise Warehouse

- Contains information spanning entire organization
- Provides corporate-wide data integration, usually from operational systems or external information providers
- Contains detailed as well as summarized data
- Size ranges from a few hundred GB to hundreds of GB, TB and beyond
- Implemented on Traditional Mainframes, UNIX Superservers and Parallel Architecture Platforms
- Requires extensive business modeling and may take years to design and build

Data Warehouse Models



Data Mart

- Contains a subset of corporate-wide data that is of value to a specific group of users. Scope is confined to specific selected subjects
- Implemented on low-cost departmental servers that are UNIX or Windows NT based. Implementation cycle is measured in weeks rather than months or years
- They are further classified as:

Independent

- Sourced from data captured from one or more operational systems or external information providers
- Sourced from data generated locally within a particular department or geographic area

Dependent

- Sourced directly from enterprise data warehouses

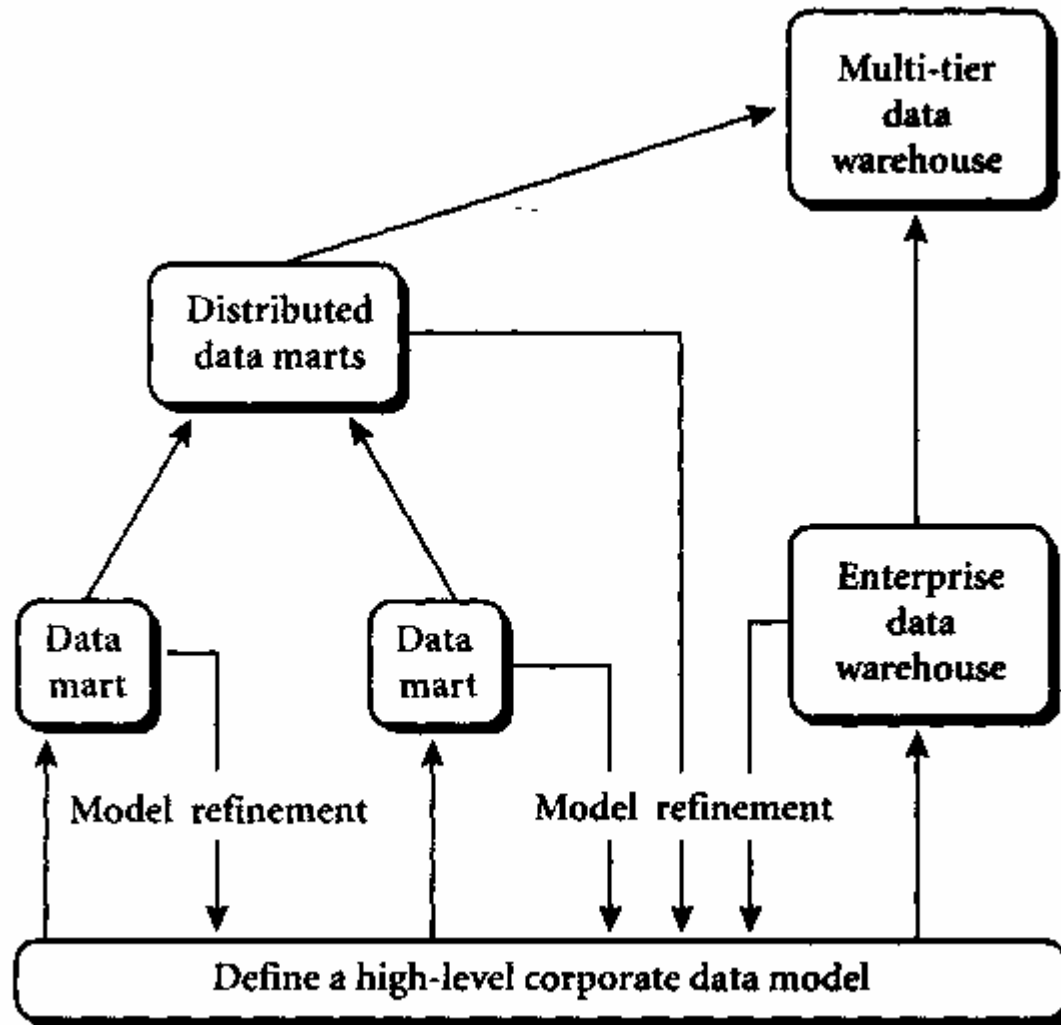
Data Warehouse Models



Virtual Warehouse

- It is a Set of Views over operational databases
- For efficient query processing, only some of the possible summary views may be materialized
- It is easy to build but requires excess capacity on operational database servers

Data Warehouse Development (A Recommended Approach)



OLAP Servers



1. Relational OLAP (ROLAP) servers

- They stand between relational back-end server and client front-end tools
- Use relational or extended-relational DBMS to store and manage warehouse. Also contain optimization for each DBMS back end
- ROLAP technology tends to have greater scalability than MOLAP technology

Eg:- DSS Server of Microstrategy, Metacube of Informix

2. Multidimensional OLAP (MOLAP) servers

- Support multidimensional views of data through array-based multidimensional storage engines
- They map multidimensional views directly to data cube array structures
- Data Cube allows faster Indexing to precomputed summarized data
- Many MOLAP servers adopt a two-level storage representation

Eg:- Essbase of Arbor

OLAP Servers



3. Hybrid OLAP (HOLAP) servers

- Combine ROLAP and MOLAP technology
- Benefits from greater scalability of ROLAP
- Benefits from faster computation of MOLAP
- HOLAP servers may allow large volumes of detail data to be stored in a relational database, while aggregations are kept in a separate MOLAP store

Eg:- The Microsoft SQL Server 7.0 OLAP Services

4. Specialized SQL Servers

- Provide advanced query language and query processing support for SQL queries over star and snowflake schemas in a read-only environment

Eg:- Redbrick of Informix

DISTRIBUTED DATA WAREHOUSES:

The role of adaptive information agents

Nathan T. Clapham, David G. Green &
Michael Kirley



Proceedings of the
The 2000 Third Asian Pacific Conference on Simulated Evolution and Learning
(SEAL-2000). pp 2792-2797. IEEE Press.

-Yeshesvini Shirahatti

INTRODUCTION



- Discovery of relevant information, is one of the major challenges faced in the Information Age
- **AIM:** To propose a scalable, model for online distributed data warehouses.
- **ABOUT THE MODEL:** It is a population of **adaptive agents**, 1 per data warehouse.

WHAT IS AN AGENT?



- According to the Macquarie dictionary (1997) a software agent is “*a piece of software that performs automatic operations on a network.*”
- In our case, an agent is a piece of software that basically **retrieves specific information** to us, on being called.
- An ideal or “smart” agent should be:
 - **Intelligent**
 - **Autonomous**
 - **Co-operative**

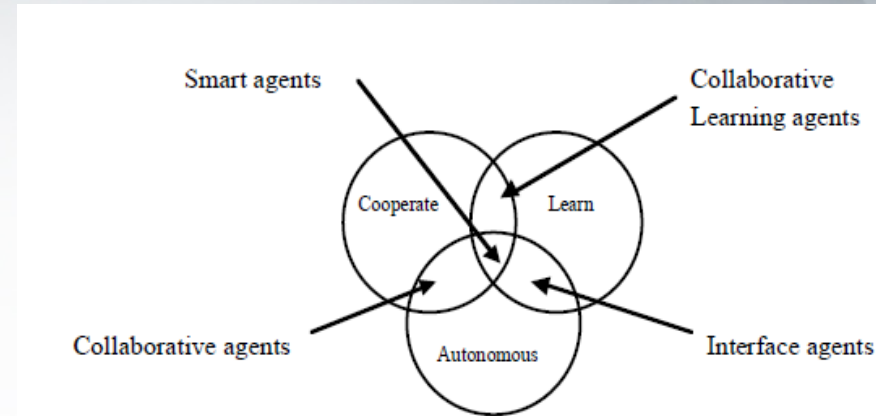


Figure 2: A part-view of an agent typology (cf Nwana 1996).

EXISTING SEARCH METHODS



- AltaVista and Excite: Common focus on indexing information
- They return thousands of items for each query
- Eg: A simple search for “virus” gives links to computer viruses first, and then about the biological viruses!

The screenshot shows the AltaVista search engine interface. The search bar contains the word "virus". Below the search bar, there are tabs for "Web", "Images", "MP3/Audio", "Video", and "News". The "Web" tab is selected. The search results are displayed in a list format. The first result is "Live Virus Help by Phone - \$1.99/ Minute" from "www.geeksbyminute.com". This result is circled in red. Other results include "AntiVirus Download - Free Trial" from "www.demoware.net", "Spyware Virus Remover Download" from "www.pctools.com", and "Symantec: Virus Alerts and Hoaxes" from "www.symantec.com". The page also shows the total number of results found (143,000,000) and the search engine's logo.

altavista Web Images MP3/Audio Video News

virus **FIND** [Advanced Search Settings](#)

SEARCH: ☐ Worldwide ☒ USA RESULTS IN: ☒ All languages ☐ English, Spanish

[Sponsored Matches](#) [Become a sponsor](#)

[Live Virus Help by Phone - \\$1.99/ Minute](#)
Get live expert technical support to scan and remove PC virus infections, 24-hours a day. Call Geeks-by-Minute and try us today.
www.geeksbyminute.com

[AntiVirus Download - Free Trial](#)
Download PC Tools Anti-Virus trial for free. The trial is 100% functional and removes viruses, trojans, worms and some other unspecified types of malicious software.
www.demoware.net

[Spyware Virus Remover Download](#)
5-star rated by CNET - killer of spyware and adware not detected by anti-virus software. Includes site guard and popup blocker. Download now and scan your PC for free.
www.pctools.com

AltaVista found 143,000,000 results

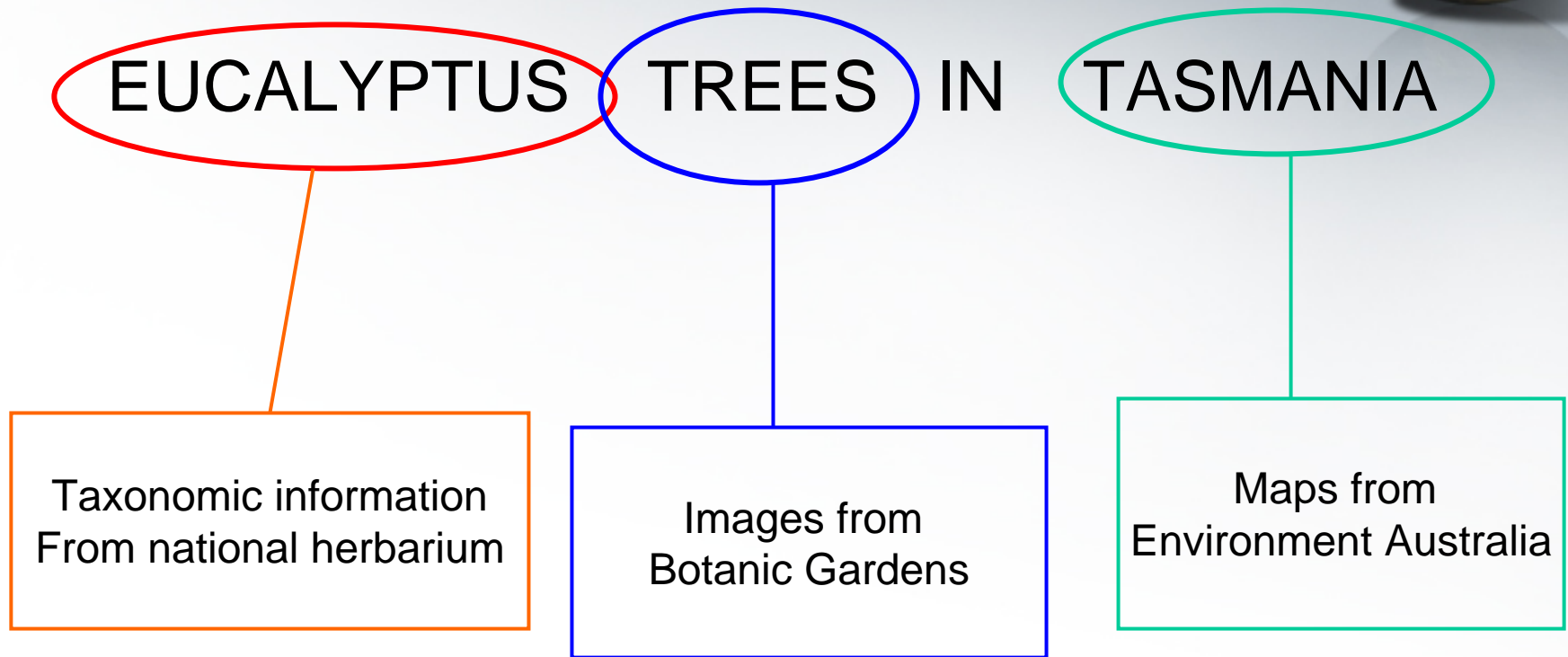
[Symantec: Virus Alerts and Hoaxes](#)
Symantec: Virus Alerts and Hoaxes Symantec: Virus Alerts and Hoaxes This website by Symantec (makers of Norton AntiVirus) provides information on the latest virus threats, security advisories, updates for Symantec products and removal tools, as ...
www.symantec.com/avcenter/hoax.html
[More pages from symantec.com](#)

[Symantec AntiVirus Research Center's Online Encyclopedia](#)
Symantec AntiVirus Research Center's Online Encyclopedia Symantec AntiVirus Research Center's Online Encyclopedia Commonly recognized as one of the best sites for virus information, Symantec provides a searchable encyclopedia for both known ...
www.symantec.com/avcenter/vinfodb.html
[More pages from symantec.com](#)

[DISINFECT THE CORE](#)
Want to see something impossible??? Here
www.thejab.com/newsite/balm.html
[More pages from thejab.com](#)



- **Main problems with the existing search approaches:**
 - High ratio of “false hits” (irrelevant information)
 - Users themselves have to locate and extract the information, from the search results
- **What a user wants:** A report consisting of all relevant elements drawn from different data sources
- **Example query:** “Eucalyptus Trees in Tasmania”



This information or final report is what the user is actually interested in , and NOT the list of resources, that would be retrieved by the search agents

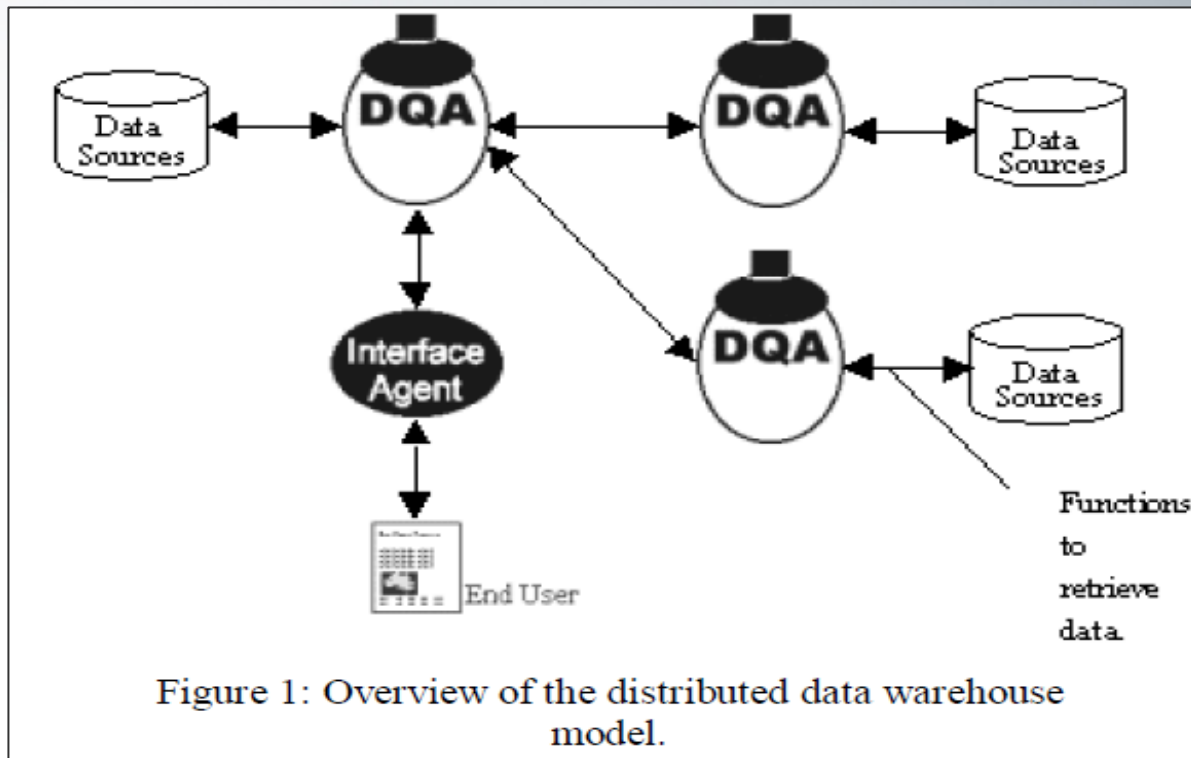
WHY USE AGENTS IN A DISTRIBUTED DATA WAREHOUSE ENVIRONMENT?



- In a distributed environment, a problem arises of **separating** the processing from the data
- It is inefficient to draw data from disparate sources, and then process the data (longer time, and a frustrated user!)
- **Efficient alternative:** Process the data at the source, and then transmit only the results to the user.

.... The agent based model uses this alternative

THE AGENT BASED MODEL



DQA: Distributed Query Agent
DDW: Distributed Data Warehouse

Working of the Agent Model



- The DDW (Distributed Data Warehouse) model is built around DQA (Distributed Query Agents)
- FRONT END: User interacting with an agent via a website
- The user queries the agent, and based on the query, the agent might:
 - A. Process the request itself
 - B. Send the request to another agent

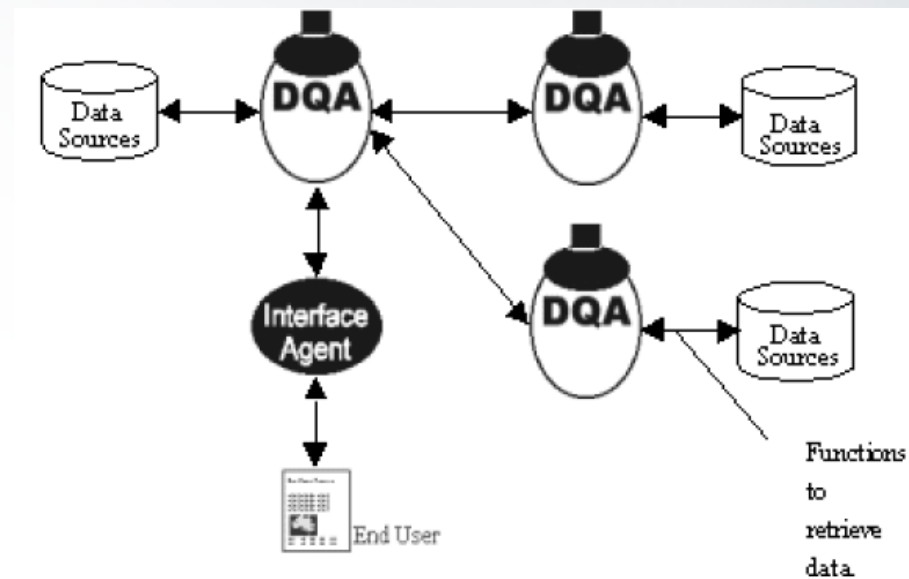


Figure 1: Overview of the distributed data warehouse model.

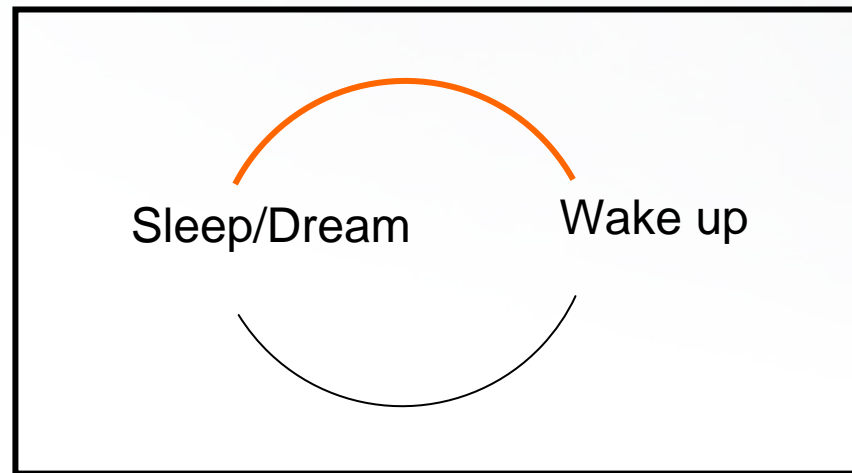


- Main Characteristics of the Model:
 - Each DQA represents a Data Warehouse, each with a limited domain of interest
 - **Collaboration between agents.** This enables agents to learn about new data sources, which helps them to process queries extending beyond domain limitations (i.e its data warehouse)
 - This facilitates a **scalable architecture.**
 - The system can now **grow, adapt and learn.**

ADAPTIVE AGENTS



- How adaptive agents work:



- Wake Cycle: Agent processes the user's or another agent's query.
Stores the query details in short term memory
- Sleep/Dream Cycle: Processes the contents of short term memory.



- Sleep/Dream cycle: The agent processes the query in short term memory
- This involves comparing each query script with its existing functions.
- Agent functions: Procedures for acquiring information from a data source.
- Script elements identified as new are extracted to create new functions.

IMPLEMENTATION



- Query: *Eucalyptus regnans* in Tasmania
- DQA: Implemented in JAVA, resides behind a WWW server Common Gateway Interface(CGI)
- DQA's communicate with CGI to satisfy queries.
- Queries: Expressed in XML markup, called as the Report Generation Language (RGL)
- Objects are retrieved based on the query tags.
- In our example, a **map** object is used, retrieved from the SOURCE <http://life.csu.edu.au/cgi-bin/speciesDistDQA.cgi>
- SOURCE: A species distribution agent



QUERY EXAMPLE

```
<OBJECT TYPE="map">  
<QUERY SOURCE="http://life.csu.edu.au/cgi-bin/specDistDQA.cgi" THEME="plant">  
<ATTRIB ID="GENUS"><VAR ID="1"/></ATTRIB>  
<ATTRIB ID="SPECIES"><VAR ID="2"/></ATTRIB>  
<ATTRIB ID="LOCATION"><VAR ID="3"/></ATTRIB>  
</QUERY></OBJECT>
```

ABOVE QUERY MAPPED TO A FUNCTION BY AGENT IN THE SLEEP CYCLE

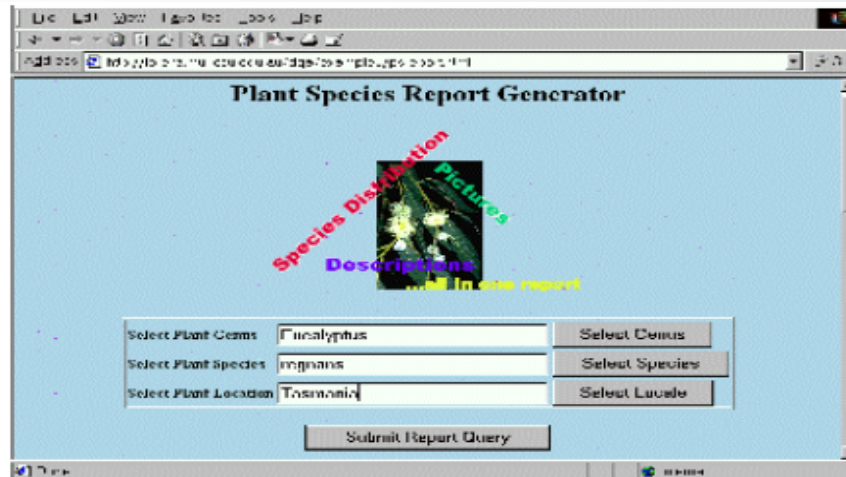
```
<FUNCTION TYPE="map" THEME="plant">  
<QUERY SOURCE="http://http://life.csu.edu.au/cgi-bin/specDistDQA.cgi">  
<ATTRIB ID="GENUS"><VAR THEME="plant/genus"/></ATTRIB>  
<ATTRIB ID="SPECIES"><VAR THEME="plant/genus/species"/></ATTRIB>  
<ATTRIB ID="LOCATION"><VAR THEME="geographic/country/state"/></ATTRIB></QUERY></FUNCTION>
```

The var tag was mapped to a function retrieving specific information, during the sleep phase. The agent has “learnt” the function from the received query.

...AND FINALLY



- The user interface:



- The interface agent sends the query to a distributed query agent (plantDQA)
- plantDQA has previously learnt from its previous queries, and uses the necessary functions from its knowledge base to obtain the necessary function



- The agents involved:

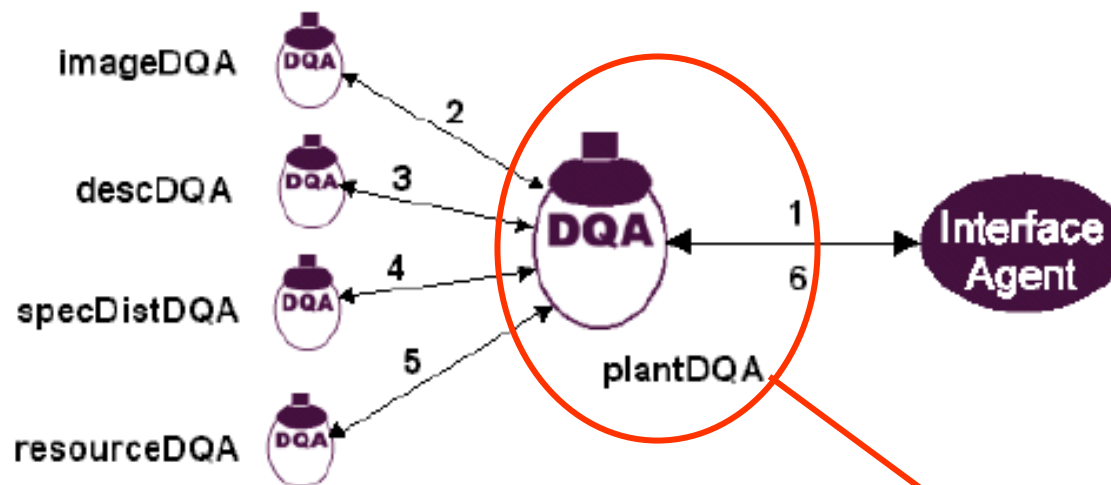


Figure 4: Communication between agents.


Distributed Agent

The output




- Fig 5: Final report

Eucalyptus regnans in Tasmania



Description

A very tall tree with rough fibrous bark to about halfway up trunk, then smooth, white or grey-green bark above, with umbels paired in leaf axils. The wood has been used for building, flooring, furniture, plywood and pulp and paper-making, and is moderately strong and hard though not durable. The tallest tree species in Australia, and the tallest hardwood in the world.

Species Distribution

Resources

- [Australian Botanical Gardens](#)
- [Environment Australia](#)

Report produced by Plant Species Report Generator
<http://life.csu.edu.au/dqa/>

Fig 3: An example of RGL

```
<OBJECT TYPE="report">
  <HEAD>
    <TERM ID="1" THEME="plant/genus">Eucalyptus</TERM>
    <TERM ID="2" THEME="plant/genus/species">regnans</TERM>
    <TERM ID="3" THEME="geographic/
      country/state">Tasmania</TERM>
    <TITLE><VAR ID="1"/> <VAR ID="2"/> in
      <VAR ID="3"/></TITLE>
  </HEAD>
  <OBJECT TYPE="image">
    <QUERY THEME="plant"/>
  </OBJECT>
  <OBJECT TYPE="text">
    <HEAD>
      <TITLE>Description</TITLE>
    </HEAD>
    <QUERY THEME="plant"/>
  </OBJECT>
  <OBJECT TYPE="map">
    <HEAD>
      <TITLE>Species Distribution</TITLE>
    </HEAD>
    <QUERY THEME="plant">
  </OBJECT>
  <OBJECT TYPE="text">
    <HEAD>
      <TITLE>Resources</TITLE>
    </HEAD>
  </OBJECT>
```

Microsoft TerraServer- A Spatial Data Warehouse

- Tom Barclay
- Jim Gray
- Don Slutz

Proceedings of the 2000 ACM SIGMOD Conference.



The Big Picture



- **Input:** Terabytes of “Internet unfriendly” geo-spatial images
 - **Output:** Hundreds of millions of scrubbed and cleaned, “internet friendly” images loaded into a SQL database for delivery via Internet to web browsers
- Goal:** To develop a scaleable wide-area, client/server imagery Internet database application to handle processing and delivery for heavy we traffic.

Why not a classic data warehouse?



TerraServer, a multi-media data warehouse :

- Accessed by millions of users
- Users extract relatively few records (thousands) in a particular session
- Records relatively large (10 kilobytes).

Classic data warehouses:

- Accessed by a few hundred users via proprietary interfaces
- Queries examine millions of records, to discover trends or anomalies,
- Records less than a kilobyte.

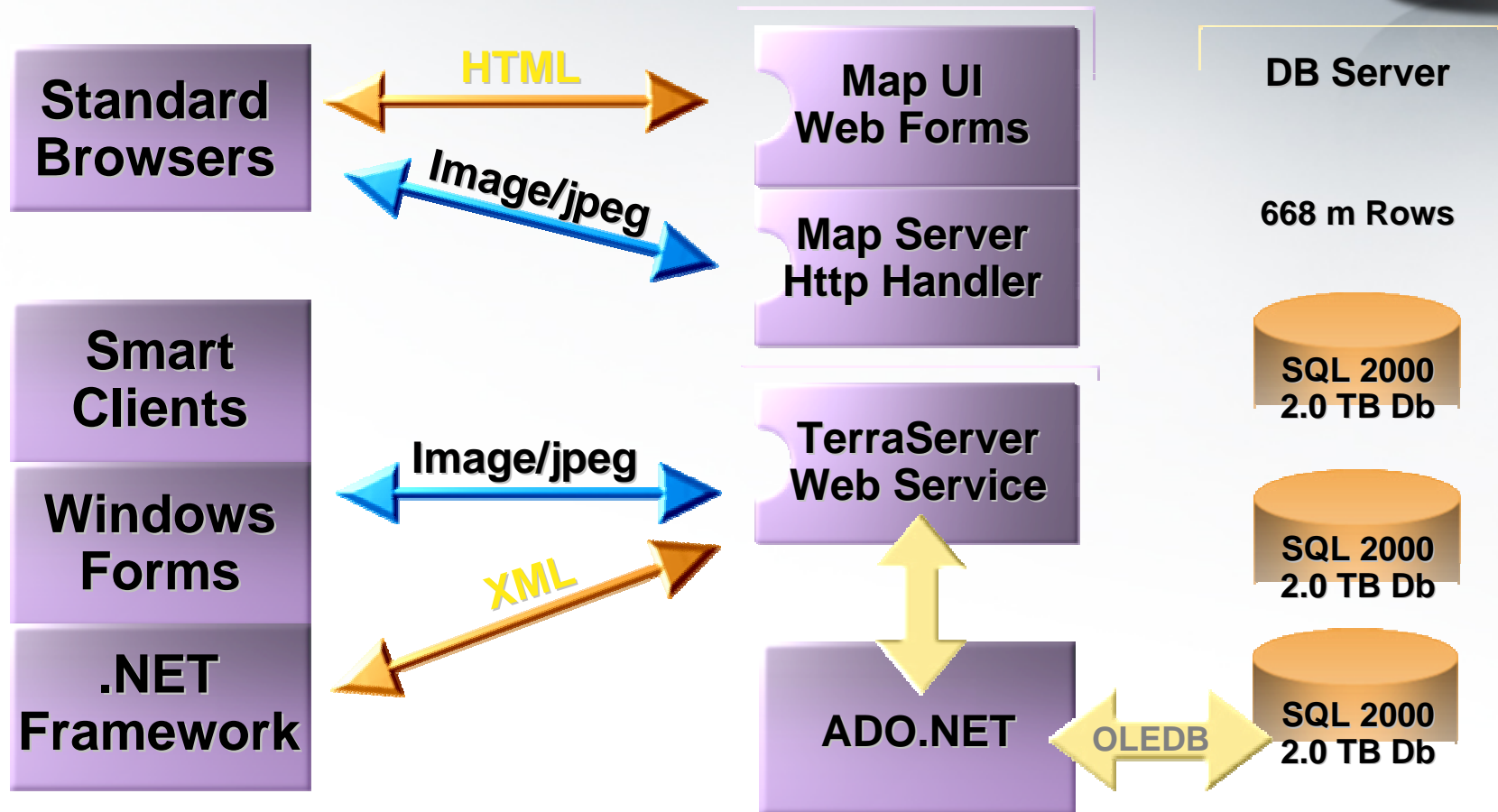
System Architecture



TerraServer - A “thin-client / fat-server” 3-tier architecture:

- **Tier 1: *The Client*** , a graphical web browser or other hardware/software system supporting HTTP 1.1 protocols.
- **Tier 2: *The Application Logic***, a web server application to respond to HTTP requests submitted by clients by interacting with Tier 3
- **Tier 3: *The Database System***, a SQL Server 7.0 Relational DBMS containing **all image** and meta-data required by Tier 2.

System Architecture (contd.)



TerraServer Schema



- Data Storage
 - TerraServer Grid System
 - Imagery Database Schema
 - Gazetteer Database Schema
- Data Load Process
 - TerraCutter
 - TerraScale

Data Storage: TerraServer Grid System



- Based on Universal Transverse Mercator (UTM) coordinate system.
- 1 large mosaic of tiled images, each identified by its location within a scene.
- Users interact with system using Geographic coordinates, TerraServer search system performs conversion from geographic coordinates to TerraServer (or UTM) coordinates.
- Data loaded into system, loading program then assigns six fields - resolution, theme, scene ID, scale, X coordinate, and Y coordinate - to every tile to create unique key to identify image

Data Storage:

Imagery Database Schema



- *Each* image source considered to be a theme and each theme has its own source meta-data table.
- Image source data used as primary key and stored with all of the meta-data in an SQL database.
- Each theme table has same five-part primary key:
 - *SceneID* –individual scene identifier
 - *X* – tile’s relative position on the X-axis
 - *Y* – tile’s relative position on the Y-axis
 - *DisplayStatus* – Controls display of an image tile
 - *OrigMetaTag* – image the tile was extracted from
- 28 other fields to describe geo-spatial coordinates for image and other properties. One field is a “blob type” that contains the compressed image.

Thus, actual image tile (approximately 10KB) is stored allowing fast download times over standard modems

Data Storage:

Gazetteer Database Schema

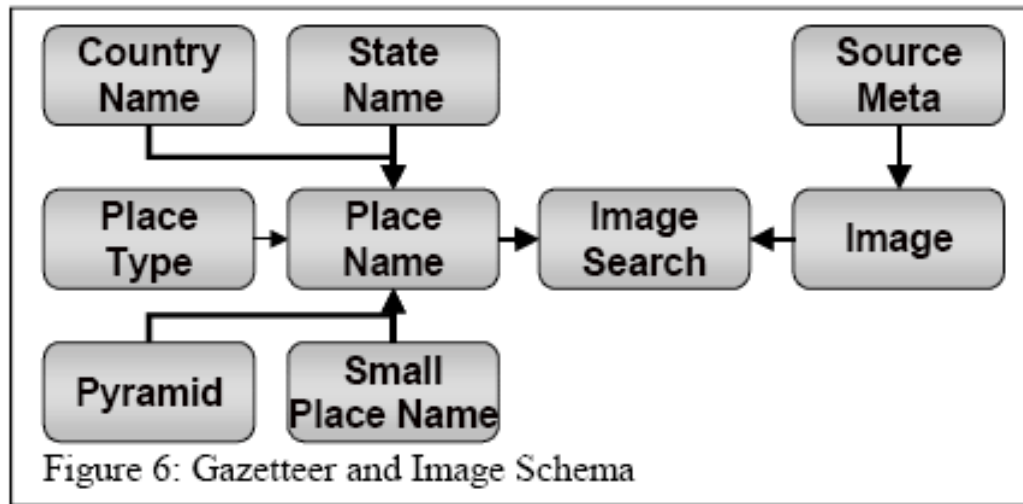


- Allows users to find an image by a keyword search.
- Database schema essentially a snowflake design- formal location name at center, altNames radiating from the center.
- altName tables contain synonyms and abbreviations for places.
- On search, a stored procedure performs a join, searches appropriate field and alt name fields associated with it.

Gazetteer Database Schema (contd.)



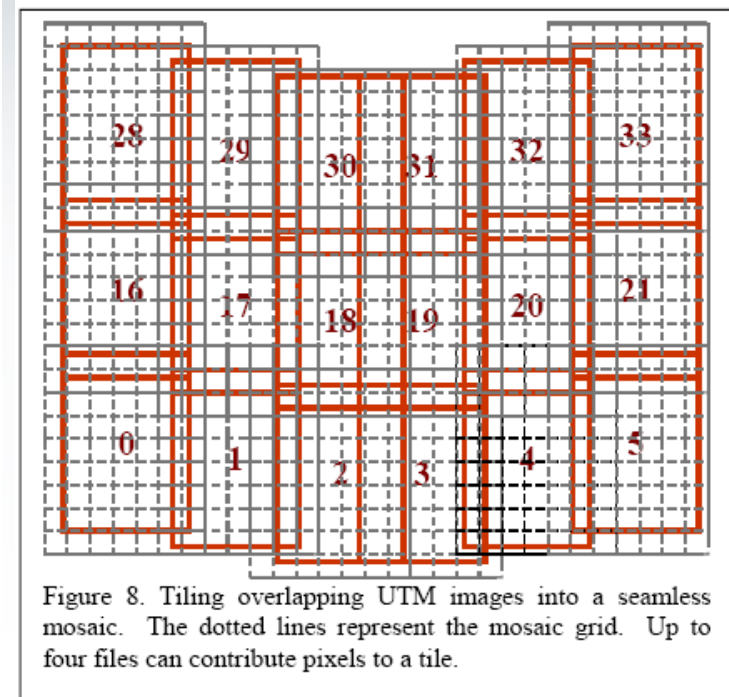
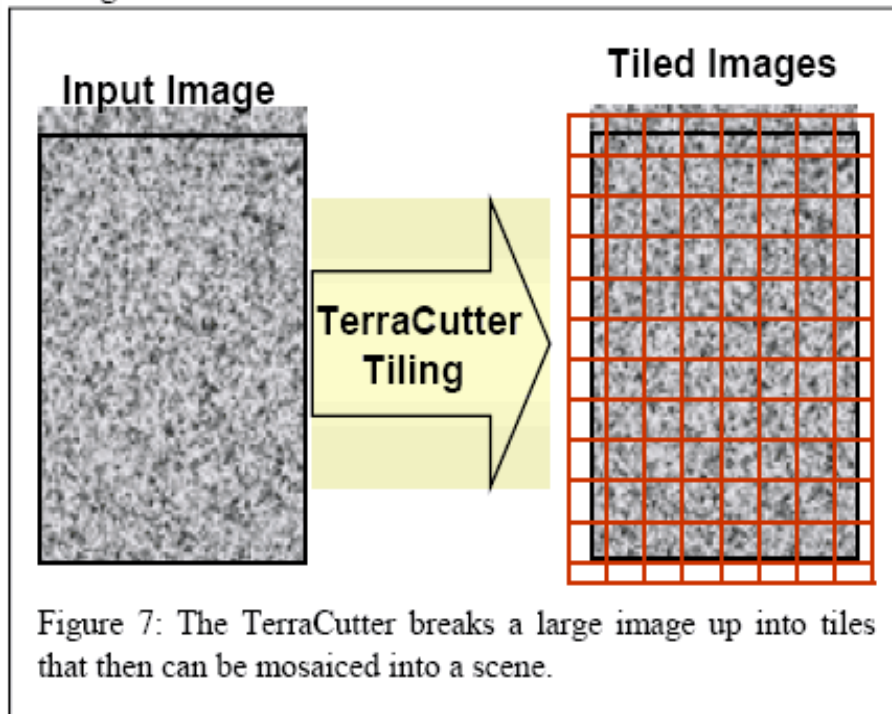
Gazetteer Database Schema



- ImageSearch table: Allows association between name and actual image, by identifying 'Theme, SceneID, Scale, X, and Y'.
- Pyramid table: Consists of name, distance to location closest to the center tile on an image.

Data Load Process:

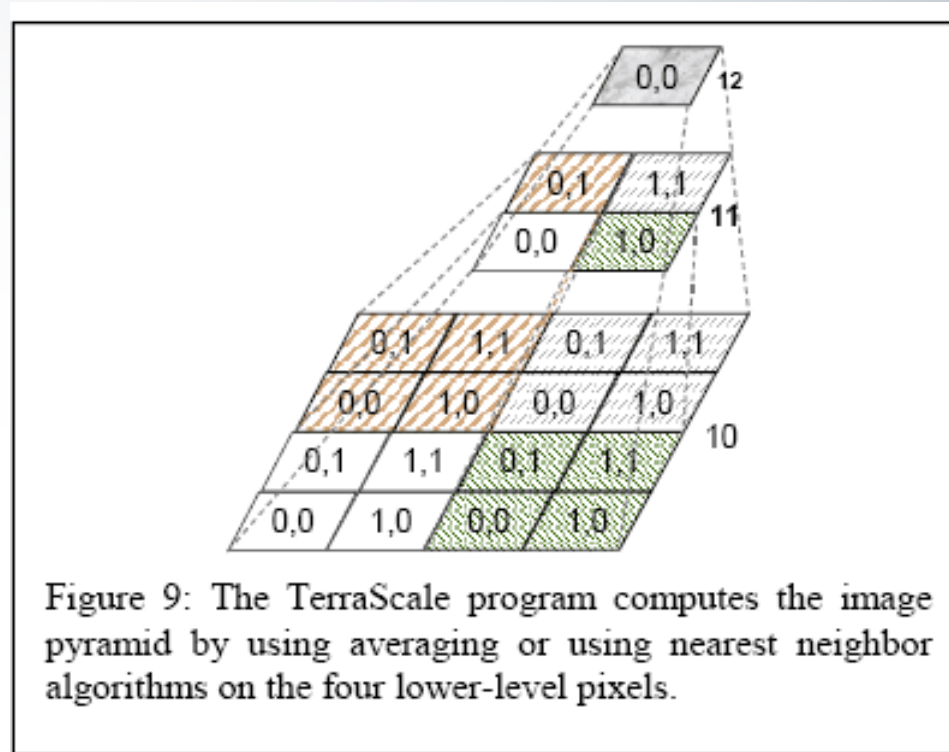
TerraCutter



C program that reformats imagery received from various data sources, tiles it into acceptable formats for TerraServer web application & inserts tiles, metadata into database.

Data Load Process:

TerraScale

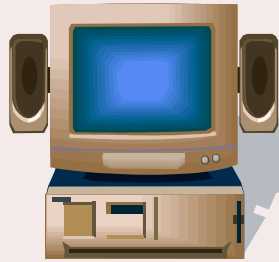


Resamples tiles created by TerraCutter to create lower resolution tiles
in the theme's image pyramid

Data Load Process



Remote Management

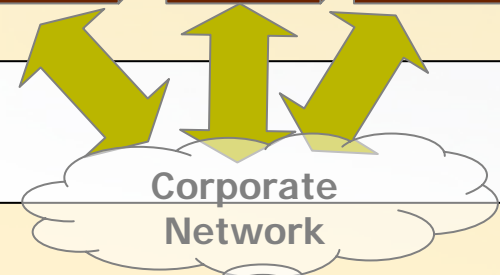
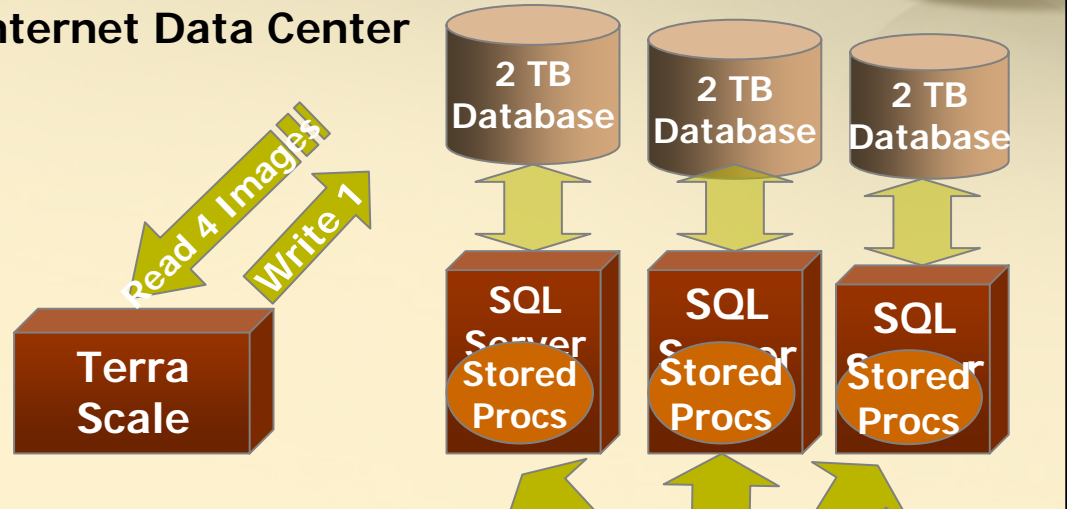


Terminal
Server

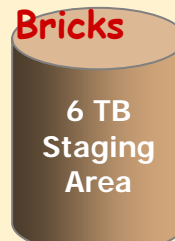
Active Server Pages
Loading
Scheduling
System



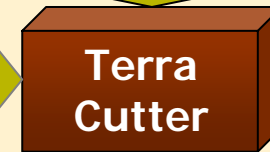
Internet Data Center



Wire Wire disks



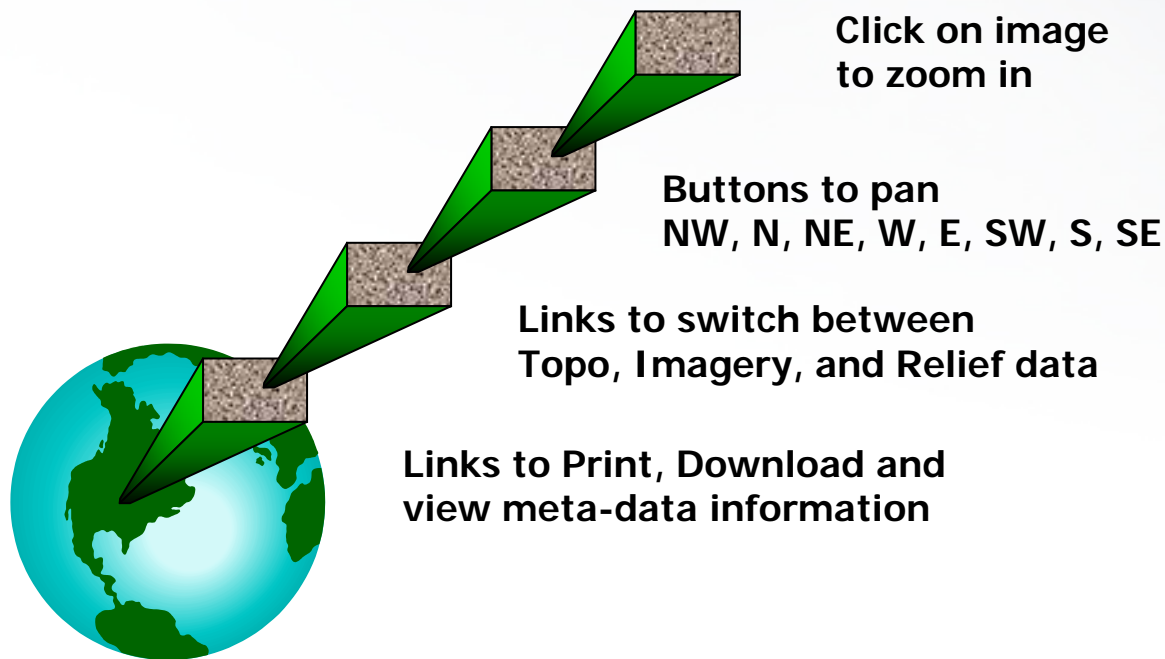
Read
Image
Files



Moral of the Story?



User navigates an ‘almost seamless’ image of earth



TerraServer Fast Facts



- Daily Usage:
 - 75k – 120k visitors
 - 1.5million to 2.2 million page views
 - 10 to 20 million “tiles”
 - 20 to 40 mega-bits per second peak network bandwidth
- Database Statistics:
 - 5.3 TB of compressed (jpeg/gif) imagery
 - 525 million image “tiles”
 - 1 billion rows (Meta & Imagery)

Major Contributions of TerraServer:



- Model for a scalable architecture to store & deliver terabytes of data over the Internet
- Process to store and index geographic raster data based on a grid X,Y coordinate system
- Method to store and present scalable raster data

Thus, disassemble & store large images and then reassemble them on the fly to efficiently deliver over a low bandwidth Internet.



Thank you!!