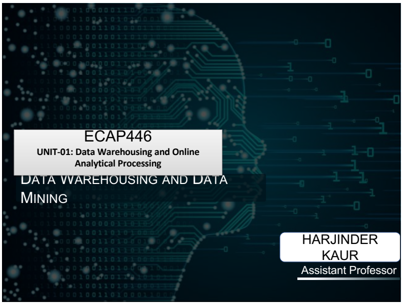


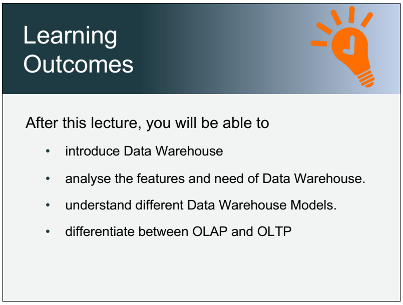
ECAP446

UNIT-01: Data Warehousing and Online Analytical Processing

DATA WAREHOUSING AND DATA MINING

HARJINDER KAUR

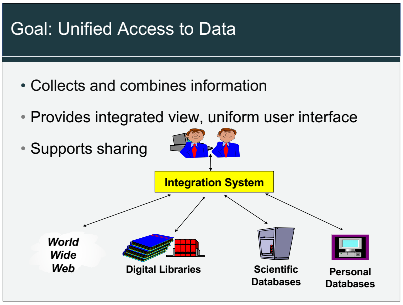
Assistant Professor



Learning Outcomes

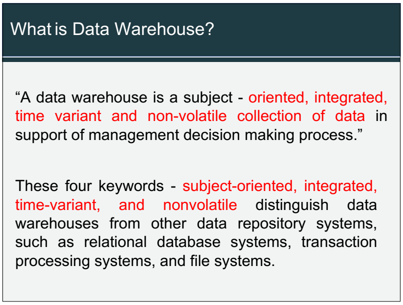
After this lecture, you will be able to

- introduce Data Warehouse
- analyse the features and need of Data Warehouse.
- understand different Data Warehouse Models.
- differentiate between OLAP and OLTP



Goal: Unified Access to Data

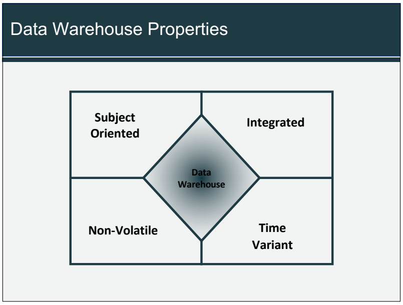
- Collects and combines information
- Provides integrated view, uniform user interface
- Supports sharing



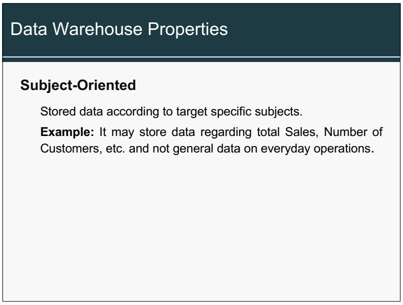
What is Data Warehouse?

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

These four keywords - subject-oriented, integrated, time-variant, and nonvolatile distinguish data warehouses from other data repository systems, such as relational database systems, transaction processing systems, and file systems.



Data Warehouse Properties



Data Warehouse Properties

Subject-Oriented

Stored data according to target specific subjects.

Example: It may store data regarding total Sales, Number of Customers, etc. and not general data on everyday operations.

Data Warehouse Properties

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.

Data Warehouse Properties

Time Variant

Data are stored to provide information from an historic perspective.

Example: Data of sales in last 5 years, etc.

Data Warehouse Properties

Non-Volatile

It is separate from the Enterprise Operational Database and hence is not subject to frequent modification.

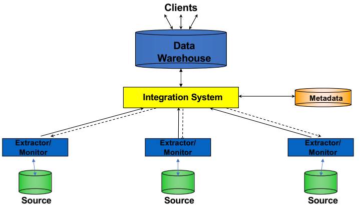
It generally has only 2 operations performed on it

- Loading of data
- Access of data.

The Warehousing Approach

- Information integrated in advance
- Stored for direct querying and analysis

The Warehousing Approach



Clients

Data Warehouse

Integration System

Metadata

Extractor/ Monitor

Extractor/ Monitor

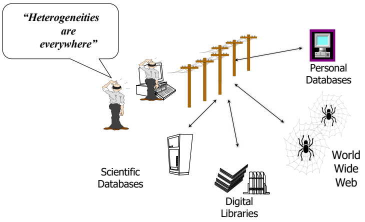
Extractor/ Monitor

Source

Source

Source

Heterogeneous Information Sources



"Heterogeneities are everywhere"

Personal Databases

World Wide Web

Scientific Databases

Digital Libraries

Heterogeneous Information Sources

- Different interfaces.
- Different data representations.
- Duplicate and inconsistent information.

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).

Need of a Separate Data Warehouse

- Use of OLAP query on OLTP system degrades system's performance.
- 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.
- An ***Operational Database*** is designed for known tasks like indexing and hashing using primary keys, searching for particular records, and many more.

Need of a Separate Data Warehouse

- Data Warehouse queries are often complex.
- The computation of large data groups at summarized levels, and may require the use of special data organization, access, and implementation methods based on multidimensional views.

Data Warehousing: Two Distinct Issues

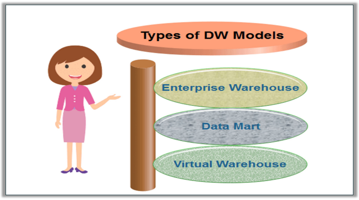
- How to get information into warehouse?

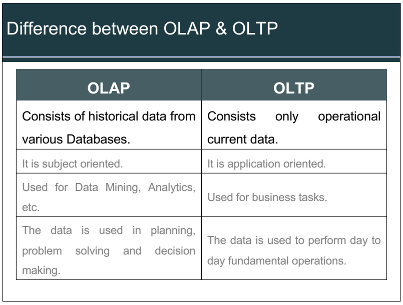
-"Data warehousing"

- What to do with data once it's in warehouse?

-"Warehouse DBMS"

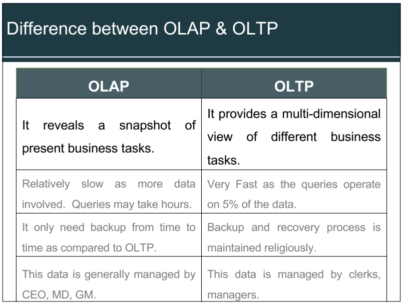
Data Warehouse Models





Difference between OLAP & OLTP

OLAP	OLTP
Consists of historical data from various Databases.	Consists only operational current data.
It is subject oriented.	It is application oriented.
Used for Data Mining, Analytics, etc.	Used for business tasks.
The data is used in planning, problem solving and decision making.	The data is used to perform day to day fundamental operations.



Difference between OLAP & OLTP

OLAP	OLTP
It reveals a snapshot of present business tasks.	It provides a multi-dimensional view of different business tasks.
Relatively slow as more data involved. Queries may take hours.	Very Fast as the queries operate on 5% of the data.
It only need backup from time to time as compared to OLTP.	Backup and recovery process is maintained religiously.
This data is generally managed by CEO, MD, GM.	This data is managed by clerks, managers.

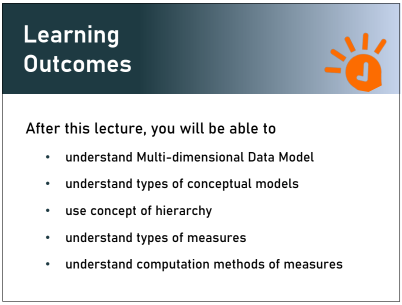
ECAP456

U01-T02

DATA WAREHOUSING AND DATA MINING

HARJINDER KAUR

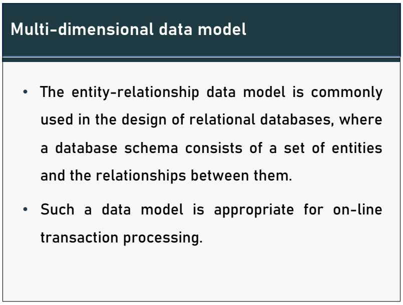
Assistant Professor



Learning Outcomes

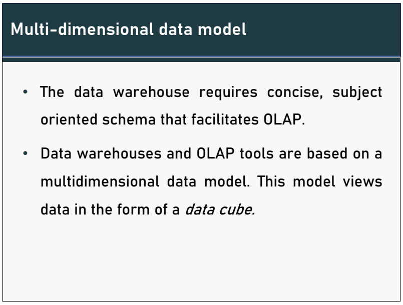
After this lecture, you will be able to

- understand Multi-dimensional Data Model
- understand types of conceptual models
- use concept of hierarchy
- understand types of measures
- understand computation methods of measures



Multi-dimensional data model

- The entity-relationship data model is commonly used in the design of relational databases, where a database schema consists of a set of entities and the relationships between them.
- Such a data model is appropriate for on-line transaction processing.



Multi-dimensional data model

- The data warehouse requires concise, subject oriented schema that facilitates OLAP.
- Data warehouses and OLAP tools are based on a multidimensional data model. This model views data in the form of a *data cube.*

Multi-dimensional data model

- A data warehouse is based on a multidimensional data model which views data in the form of a data cube
- A data cube, such as sales, allows data to be modeled and viewed in multiple dimensions
  - Dimension tables, such as item (item_name, brand, type), or time(day, week, month, quarter, year)
  - Fact table contains measures (such as dollars_sold) and keys to each of the related dimension tables

Multi-dimensional data model

- In data warehousing literature, an n-D base cube is called a base cuboid. The topmost 0-D cuboid, which holds the highest-level of summarization, is called the apex cuboid.
- The most popular data model for a data warehouse is a multidimensional model, which can exist in the form of a star schema, a snowflake schema, or a fact constellation

Conceptual Modeling of Data Warehouses

Modeling data warehouses: dimensions & measures

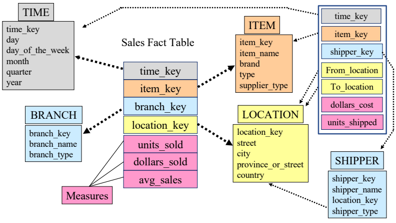
- **Star schema:** A fact table in the middle connected to a set of dimension tables
- **Snowflake schema:** A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake
- **Fact constellations:** Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation

Conceptual Modeling of Data Warehouses

Data warehouse contains:

1. A large central table (fact table) containing the bulk of the data, with no redundancy, and
2. A set of smaller attendant tables (dimension tables), one for each dimension.

Example of Fact Constellation



Defining a Star Schema in DMQL

define cube sales_star [time, item, branch, location]:

dollars_sold = sum (sales_in_dollars),

avg_sales = avg (sales_in_dollars), units_sold = count(*)

define dimension time as (time_key, day, day_of_week, month, quarter, year)

define dimension item as (item_key, item_name, brand, type, supplier_type)

Defining a Star Schema in DMQL

define dimension branch as (branch_key, branch_name, branch_type)

define dimension location as (location_key, street, city, province_or_state, country)

Defining a Snowflake Schema in DMQL

define cube sales_snowflake [time, item, branch, location]:

dollars_sold = sum(sales_in_dollars),

avg_sales = avg(sales_in_dollars), units_sold = count(*)

define dimension time as (time_key, day, day_of_week, month, quarter, year)

define dimension item as (item_key, item_name, brand, type, supplier(supplier_key, supplier_type))

Defining a Snowflake Schema in DMQL

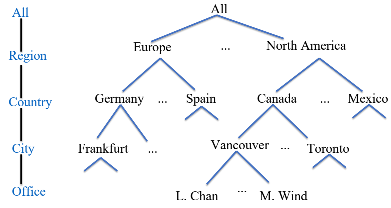
define dimension branch as (branch_key, branch_name, branch_type)

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

Concept hierarchy

A concept hierarchy defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts.

A Concept Hierarchy: Dimension (location)



All
Region
Country
City
Office

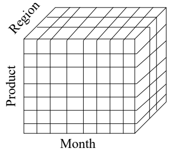
All
Europe
...
North America
Germany
...
Spain
Canada
...
Mexico
Frankfurt
...
Vancouver
...
Toronto
L. Chan
...
M. Wind

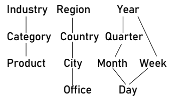
Multidimensional Data

Sales volume as a function of product, month, and region

Dimensions: Product, Location, Time

Hierarchical summarization paths





Industry
Region
Year
Category
Country
Quarter
Product
City
Month
Week
Office
Day

Measures: Their Categorization and Computation

- "How are measures computed?" To answer this question, we first study how measures can be categorized.
- A data cube measure is a numeric function that can be evaluated at each point in the data cube space.

Measures: Their Categorization and Computation

Measures can be organized into three categories

1. Distributive
2. Algebraic
3. Holistic

1. Distributive

- The data are partitioned into n sets. We apply the function to each partition, resulting in n aggregate values.
- count(), min(), and max() are distributive aggregate functions.

2. Algebraic

- An aggregate function is algebraic if it can be computed by an algebraic function with M arguments where M is a constant.
- For example, avg() can be computed by sum()/count(), min N() and max N(), and
- standard deviation() are algebraic aggregate functions.

3. Holistic

- An aggregate function is holistic if there is no constant bound on the storage size needed to describe a sub-aggregate.
- Examples of holistic functions include:
  - Median ()
  - Mode ()
  - Rank ()

ECAP456

U01-T03

DATA WAREHOUSING AND DATA MINING

HARJINDER KAUR

Assistant Professor

Learning Outcomes

After this lecture, you will be able to

- introduce Data Cube
- understand various terminologies used in Data Cube.
- learn how Data Cube Model n-dimensional data.
- visualize various OLAP Operations

What is Data Cube?

A data cube allows data to be modeled and viewed in multiple dimensions. It is defined by dimensions and facts.

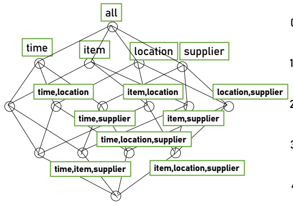
Dimensions are the entities with respect to which an organization wants to keep records.

Facts are numerical measures. It is the quantities by which we want to analyze relationships between dimensions.

What is Data Cube?

- The data cube is used by the users of the decision support system to see their data.
- The cuboid that holds the lowest level of summarization is called the base cuboid.
- The 0-D cuboid, which holds the highest level of summarization, is called the apex cuboid.

Cube: A Lattice of Cuboids



0-D (apex) cuboid

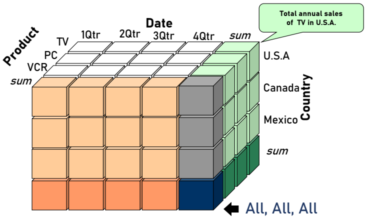
1-D cuboids

2-D cuboids

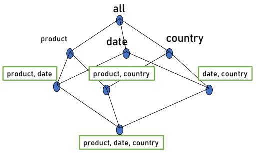
3-D cuboids

4-D(base) cuboid

A Sample Data Cube



Cuboids Corresponding to the Cube



0-D(apex) cuboid

1-D cuboids

2-D cuboids

3-D(base) cuboid

Operations in the Multidimensional Data Model (OLEP)

- In the multidimensional model, data are organized into multiple dimensions, and each dimension contains multiple levels of abstraction defined by concept hierarchies.
- Organization provides users with the flexibility to view data from different perspectives.

Operations in the Multidimensional Data Model (OLEP)

- A number of OLAP data cube operations exist to materialize these different views, allowing interactive querying and analysis of the data at hand.

Typical OLAP Operations

Roll up (drill-up): summarize data

by climbing up hierarchy or by dimension reduction

Drill down (roll down): reverse of roll-up

from higher level summary to lower level summary or detailed data, or introducing new dimensions

Typical OLAP Operations

Slice and dice: project and select

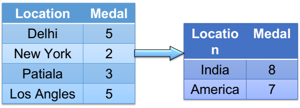
The Slice operation performs a selection on one dimension of the given cube, resulting in a subcube. The dice operation defines a subcube by performing a selection on two or more dimensions.

Pivot (rotate):

Reorient the cube, visualization, 3D to series of 2D planes.

Drill-Up Or Roll-Up

- The roll-up operation performs aggregation on a data cube either by climbing up the hierarchy or by dimension reduction.

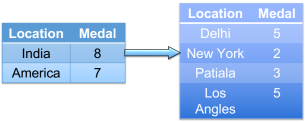


Location	Medal
Delhi	5
New York	2
Patiala	3
Los Angles	5

Location	Medal
India	8
America	7

Roll-Down Or Drill Down

1. Stepping down a concept hierarchy for a dimension.
2. By introducing a new dimension.



Location	Medal
India	8
America	7

Location	Medal
Delhi	5
New York	2
Patiala	3
Los Angles	5

Slice

- The slice operation performs a selection on one dimension of the given cube, resulting in a subcube. Reduces the dimensionality of the cubes.
- For example a select where Medal = 5.

Location	Medal
Delhi	5
Los Angles	5

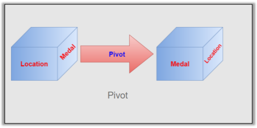
Dice

- The dice operation defines a sub-cube by performing a selection on two or more dimensions.
- For example, if we want to make a select where Medal = 3 or Location = New York.

Location	Medal
Patiala	3
New York	2

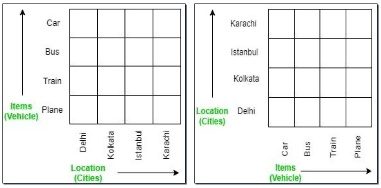
Pivot

- Pivot is also known as rotate. It Rotates the data axis to view the data from different perspectives.



Pivot

Before and After Pivoting



ECAP45
6

U01-T04

DATA WAREHOUSING AND DATA MINING

HARJINDER KAUR
Assistant Professor

Learning Outcomes

After this lecture, you will be able to

- define business analysis framework for data warehouse design.
- analyse the design process
- understand various methods for the efficient implementation of data warehouse systems.
- apply indexing on OLAP data.

Data Warehouse Design

The Design of a Data Warehouse: A Business Analysis Framework

“What can business analysts gain from having a data warehouse?”

Data Warehouse Design

1. Competitive advantage (by presenting relevant information to help win over competitors).
2. Data warehouse can enhance business productivity (because it is able to quickly and efficiently gather information that accurately describes the organization).

Data Warehouse Design

3. Data warehouse facilitates customer relationship management (because it provides a consistent view of customers and items across all lines of business, all departments, and all markets).
4. Cost reduction by tracking trends, patterns, and exceptions over long periods in a consistent and reliable manner.

Data Warehouse Design

To design an effective data warehouse we need to understand and analyze business needs and construct a business analysis framework.

Views in data warehouse design

Four different views regarding the design of a data warehouse must be considered:

- The top-down view allows the selection of the relevant information necessary for the data warehouse. This information matches the current and future business needs.

Views in data warehouse design

- The data source view exposes the information being captured, stored, and managed by operational systems.
- The data warehouse view includes fact tables and dimension tables. It represents the information that is stored inside the data warehouse.

Views in data warehouse design

- The business query view is the perspective of data in the data warehouse from the viewpoint of the end user.

Process of Data Warehouse Design

Data warehouse can be built using a

- top-down approach,
- bottom-up approach, or
- combination of both.

Process of Data Warehouse Design

- Top-down approach starts with the overall design and planning. It is used where technology is mature and well known & business problems that must be solved are clear and well understood.
- Bottom-up approach starts with experiments and prototypes. Useful in the early stage of business modeling and technology development.

Process of Data Warehouse Design

Combined approach, an organization can exploit the planned and strategic nature of the top-down approach while retaining the rapid implementation and opportunistic application of the bottom-up approach.

Warehouse design process

Steps:

- Choose a business process to model (e.g., orders, invoices, shipments, inventory, account administration, sales, or the general ledger).
- Choose the business process grain, which is the fundamental, atomic level of data to be represented in the fact table for this process (e.g., individual transactions, individual daily snapshots, and so on).

Warehouse design process

- Choose the dimensions that will apply to each fact table record.
- Choose the measures that will populate each fact table record. Typical measures are numeric additive quantities like dollars_sold and units_ sold.

Data Warehouse usage

- Initially, the data warehouse is mainly used for generating reports and answering predefined queries.
- Progressively, it is used to analyze summarized and detailed data.

Data Warehouse usage

- Later, the data warehouse is used for performing multidimensional analysis and sophisticated slice-and-dice operations.
- Finally, the data warehouse employed for knowledge discovery and strategic decision making using data mining tools.

Data Warehouse Applications

There are three kinds of data warehouse applications:

I. information processing
II. analytical processing
III. data mining.

Data Warehouse Applications

- Information processing
  - supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs

Data Warehouse Applications

- Analytical processing
  - It generally operates on historic data in both summarized and detailed forms.
  - supports basic OLAP operations, slice-dice, drilling, pivoting.

Data Warehouse Applications

Data mining

- knowledge discovery from hidden patterns
- supports associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools

Question

"How does data mining relate to information processing and online analytical processing? "

- Information processing, based on queries, can find useful information directly stored in databases or computable by aggregate functions.
- It do not reflect hidden patterns or regularities buried in the database.
- Therefore, information processing is not data mining.

Question

"Do OLAP systems perform data mining? Are OLAP systems actually data mining systems?"

- OLAP is a data summarization/aggregation tool that helps simplify data analysis.
- Data mining allows the automated discovery of implicit patterns and interesting knowledge hidden in large amounts of data.

Question

"Do OLAP systems perform data mining? Are OLAP systems actually data mining systems?"

- OLAP is a data summarization/aggregation tool that helps simplify data analysis.
- Data mining allows the automated discovery of implicit patterns and interesting knowledge hidden in large amounts of data.

Question

- OLAP functions are essentially for user-directed data summarization and comparison.
- Data mining covers a much broader spectrum than simple OLAP operations, because it performs not only data summarization and comparison but also association, classification, prediction, clustering, time-series analysis, and other data analysis tasks.

From Online Analytical Processing to Multidimensional Data Mining

- Multidimensional data mining (also known as exploratory multidimensional data mining, online analytical mining, or OLAM) integrates OLAP with data mining to uncover knowledge in multidimensional databases.
- Multidimensional data mining is particularly important for the following reasons:

From Online Analytical Processing to Multidimensional Data Mining

- High quality of data in data warehouses
- OLAP-based exploration of multidimensional data
- Online selection of data mining functions.

Data Warehouse Implementation

- Data warehouses contain huge volumes of data.
- OLAP servers demand that decision support queries be answered in the order of seconds.
- Therefore, it is crucial for data warehouse systems to support highly efficient cube computation techniques, access methods, and query processing techniques.

Efficient Data Cube Computation

- One approach to cube computation extends SQL to include a compute cube operator.
- The compute cube operator computes aggregates over all subsets of the dimensions specified in the operation.
- This can require excessive storage space, especially for large numbers of dimensions.

Example

- Suppose that you want to create a data cube for All Electronics sales that contains the following: city, item, year, and sales_in_ dollars.
- Taking the three attributes, city, item, and year, as the dimensions for the data cube, and sales_in_dollars as the measure.

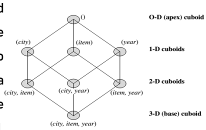
Example

- The total number of cuboids, that can be computed for this data cube is $2^3 = 8$.

{(city, item, year), (city, item), (city, year), (item, year), (city), (item),(year),() }

Example

If we start at the apex cuboid and explore downward in the lattice, this is equivalent to drilling down within the data cube. If we start at the base cuboid and explore upward, this is akin to rolling up.



Example

- An SQL query containing no group-by (e.g., “compute the sum of total sales”) is a zero-dimensional operation.
- An SQL query containing one group-by (e.g., “compute the sum of sales, group-by city”) is a one-dimensional operation.
- A cube operator on n dimensions is equivalent to a collection of group-by statements, one for each subset of the n dimensions.

Example

- A statement such as compute cube sales_cube would explicitly instruct the system to compute the sales aggregate cuboids for all eight subsets of the set {city, item, year}, including the empty subset.

Example

Transform it into a SQL-like language (with a new operator cube by, introduced by Gray et al.'96)

SELECT item, city, year, SUM (amount)

FROM SALES CUBE BY item, city, year

Question

"How many cuboids are there in an n-dimensional data cube ?"

- If there were no hierarchies associated with each dimension, then the total number of cuboids for an n-dimensional data cube, as we have seen, is 2^n.
- However, in practice, many dimensions do have hierarchies.
- For example, time "day < month < quarter < year."

Question

- For an n-dimensional data cube, the total number of cuboids that can be generated is

$$\textit{Total number of cuboids} = \prod_{i=1}^{n} (L_i + 1),$$

Partial Materialization: Selected Computation of Cuboids

- There are three choices for data cube materialization given a base cuboid:
  - No materialization:
  - Full materialization
  - Partial materialization

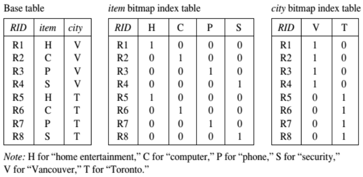
Indexing OLAP Data: Bitmap Index and Join Index

To facilitate efficient data accessing, most data warehouse systems support index structures and materialized views (using cuboids).

Bitmap Index

- The bitmap indexing method is popular in OLAP because it allows quick searching in data cubes.
- If the attribute has the value v for a given row in the data table, then the bit representing that value is set to 1 in the corresponding row of the bitmap index. All other bits for that row are set to 0.

Bitmap Index



Base table

RID	*item*	*city*
R1	H	V
R2	C	V
R3	P	V
R4	S	V
R5	H	T
R6	C	T
R7	P	T
R8	S	T

item bitmap index table

RID	H	C	P	S
R1	1	0	0	0
R2	0	1	0	0
R3	0	0	1	0
R4	0	0	0	1
R5	1	0	0	0
R6	0	1	0	0
R7	0	0	1	0
R8	0	0	0	1

city bitmap index table

RID	V	T
R1	1	0
R2	1	0
R3	1	0
R4	1	0
R5	0	1
R6	0	1
R7	0	1
R8	0	1

Note: H for "home entertainment," C for "computer," P for "phone," S for "security," V for "Vancouver," T for "Toronto."

Bitmap Index Facts

- Bitmap indexing is especially useful for low-cardinality domains.
- Bitmap indexing leads to significant reductions in space and input/output (I/O) since a string of characters can be represented by a single bit.
- For higher-cardinality domains, the method can be adapted using compression techniques.

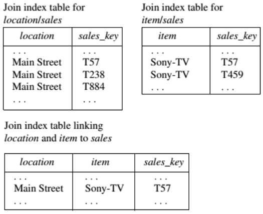
Joining Index

- Join indexing registers the joinable rows of two relations from a relational database.

Join index: JI(R-id, S-id) where R (R-id, …) ▹◃ S (S-id, …)

- Join indexing is especially useful for maintaining the relationship between a foreign key and its matching primary keys, from the joinable relation.

Example



Join index table for *location*/*sales*

location	*sales_key*
. . .	. . .
Main Street	T57
Main Street	T238
Main Street	T884
. . .	. . .

Join index table for *item*/*sales*

item	*sales_key*
. . .	. . .
Sony-TV	T57
Sony-TV	T459
. . .	. . .

Join index table linking *location* and *item* to *sales*

location	*item*	*sales_key*
. . .	. . .	. . .
Main Street	Sony-TV	T57
. . .	. . .	. . .

Efficient Processing of OLAP Queries

- The purpose of materializing cuboids and constructing OLAP index structures is to speed up query processing in data cubes. Given materialized views, query processing should proceed as follows:
  1. Determine which operations should be performed on the available cuboids.
  2. Determine to which materialized cuboid(s) the relevant operations should be applied.

Example

Suppose that we define a data cube for All Electronics of the form “sales cube [time, item, location]: sum(sales in dollars)”.

The dimension hierarchies used are:

“day < month < quarter < year” for time ;

“item name < brand < type” for item;

“street < city < province or state < country” for location.

Example

Suppose that the query to be processed is on {brand, province_or_ state}, with the selection constant “year = 2010.” Also, suppose that there are four materialized cuboids available, as follows:

cuboid 1: {year, item_name, city}

cuboid 2: {year, brand, country }

cuboid 3: {year, brand, province_or_state}

cuboid 4: {item_name, province_or_state }, where year = 2010

“Which of these four cuboids should be selected to process the query?”

Result

- Finer-granularity data cannot be generated from coarser-granularity data.
- Therefore, cuboid 2 cannot be used because country is a more general concept than province or state.
- Cuboids 1, 3, and 4 can be used to process the query.

Question

“How would the costs of each cuboid compare if used to process the query?”