**WHAT IS DATA WAREHOUSE?**

A decision support database that is maintained separately from the organization’s operational database and Supports information processing by providing a solid platform of consolidated, historical data for analysis.

(or)

A repository of multiple heterogeneous data sources organized under a unified schema at a single site in order to facilitate management decision making.

* “A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management’s decision-making process.”
* The process of constructing and using data warehouses Data warehousing

**KEY FEATURES OF A DATA WAREHOUSE.**

**Subject-oriented:** data warehouses typically provide a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process. Such as customer,supplier, product, and sales etc..

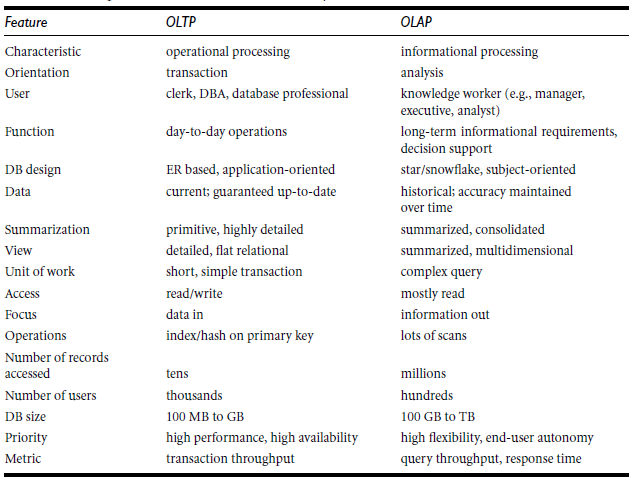
**Integrated:** A data warehouse is usually constructed by integrating multiple heterogeneous sources, such as relational databases, flat files, and on-line transaction records.

**Time-variant**: Data are stored to provide information from a historical perspective

(e.g., the past 5–10 years).

**Nonvolatile**: A data warehouse is always a physically separate store of data transformed from the application data found in the operational environment. It usually requires only two operations in data accessing: initial loading of data and access of data.

**DIFFERENCES BETWEEN OPERATIONAL DATABASE SYSTEMS AND DATA WAREHOUSES**

****

**A MULTIDIMENSIONAL DATA MODEL**

**Why Separate Data Warehouse?**

* High performance for both systems
  + DBMS— Tuned for OLTP: access methods, indexing, concurrency control, recovery
  + Warehouse—Tuned for OLAP: complex OLAP queries, multidimensional view, consolidation
* Different functions and different data:
  + **MISSING DATA**: Decision support requires historical data which operational DBs do not typically maintain
  + **DATA CONSOLIDATION**: DS requires consolidation (aggregation, summarization) of data from heterogeneous sources
  + **DATA QUALITY:** different sources typically use inconsistent data representations, codes and formats which have to be reconciled
* Note: There are more and more systems which perform OLAP analysis directly on relational databases

**FROM TABLES AND SPREADSHEETS TO DATA CUBES (WHY TO USE DATA CUBES INSTEAD OF TABLES AND SPREAD SHEETS)**

* 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

In data warehousing literature, an n-D base cube is called a base cuboid. The top most 0-D cuboid, which holds the highest-level of summarization, is called the apex cuboid. The lattice of cuboids forms a data cube.

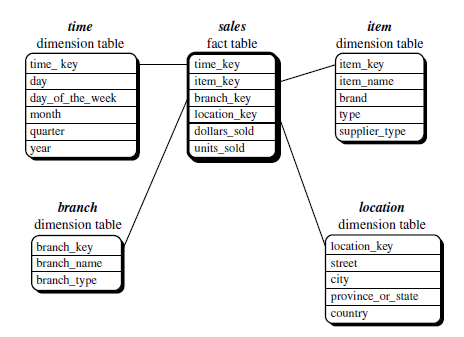
**STARS, SNOWFLAKES, AND FACT CONSTELLATIONS SCHEMAS FOR MULTIDIMENSIONAL DATABASES**

* The entity-relationship data model is commonly used in the design of relational databases.
* In the same way multi dimensional model is used for designing data warehouse. Such a model can exist in the form of a star schema, a snowflake schema, or a fact constellation schema

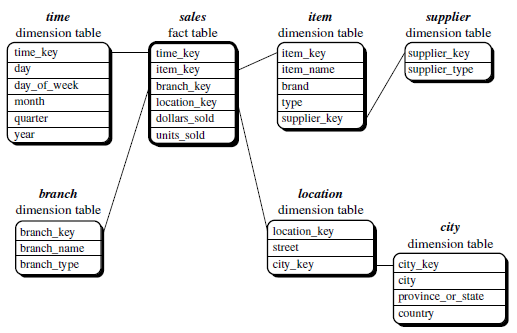
**Star schema:** The most common modeling paradigm is the star schema, in which the data warehouse contains.

(1) A large central table (fact table) containing the bulk of the data and

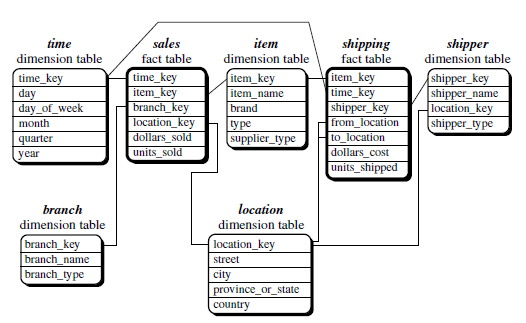
(2) A set of smaller attendant tables (dimension tables), one for each dimension.



**Snowflake schema**: The snowflake schema is a variant of the star schema model, where some dimension tables are *normalized*, thereby further splitting the data into additional tables.



**Fact constellation**: Sophisticated applications may require multiple fact tables to *share* dimension tables. This kind of schema can be viewed as a collection of stars, and henceis called a galaxy schema or a fact constellation.

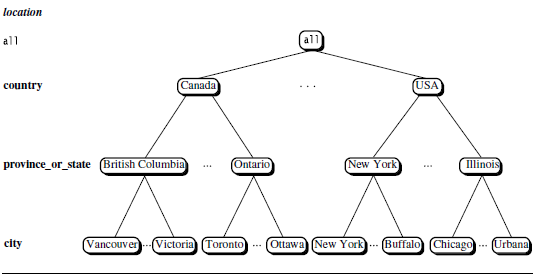
**

**CONCEPT HIERARCHIES**

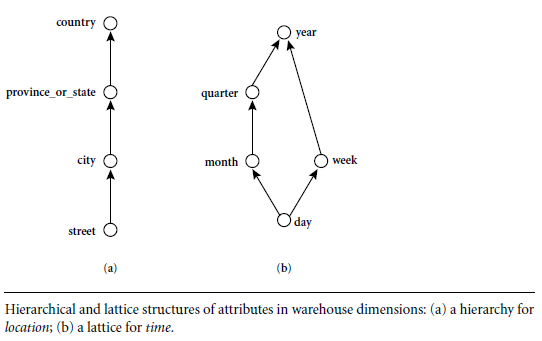
A concept hierarchy defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts.

**SCHEMA HIERARCHY**: A concept hierarchy that is a total or partial order among attributes in a database schema is called a schema hierarchy. Schema hierarchy may formally express existing relationship between attributes.

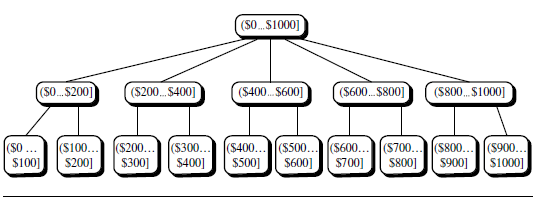
.



A concept hierarchy for the dimension *location*

****

**SET-GROUPING HIERARCHY:** Concept hierarchies may also be defined by discretizing or grouping values for a given dimension or attribute, resulting in a set-grouping hierarchy. A total or partial order can be defined among groups of values.

****

**OLAP OPERATIONS IN THE MULTIDIMENSIONAL DATA MODEL**

**Roll-up:** The roll-up operation (also called the *drill-up* operation by some vendors) performs aggregation on a data cube, either by *climbing up a concept hierarchy* for a dimension or by *dimension reduction*.

* Ex:roll-up operation aggregates data by ascending the location hierarchy from the level of city to the level of country

**Drill-down** : It is the reverse of roll-up. It navigates from less detailed data to more detailed data. Drill-down can be realized by either *stepping down a concept hierarchy* for a dimension or *introducing additional dimensions*.

* Ex: drill-down for time
* day<month<quarter<year
* form the level of quarter to the more detailed level of month

**Slice:** a selection on one dimension of the cube resulting in subcube

Ex: sale data are selected for dimension time using time =Q1

**dice:** defines a subcube by performing a selection on two or more dimensions

Ex: a dice opp. Based on

location=“toronto” or “vencover” and

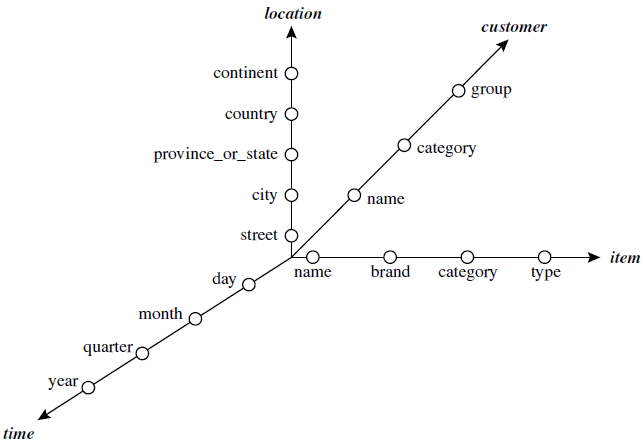
**time =Q1 or Q2 and**

**item = “home entertainment” or “computer”**

****

**A STARNET QUERY MODEL FOR QUERYING MULTIDIMENSIONAL DATABASES**

A starnet model consists of radial lines emanating from a central point, where each line represents a concept hierarchy for a dimension. Each abstraction level in the hierarchy is called a footprint. These represent the granularities available for use by OLAP operations such as drill-down and roll-up.



**DATA WAREHOUSE ARCHITECTURE**

Design of Data Warehouse: A Business Analysis Framework

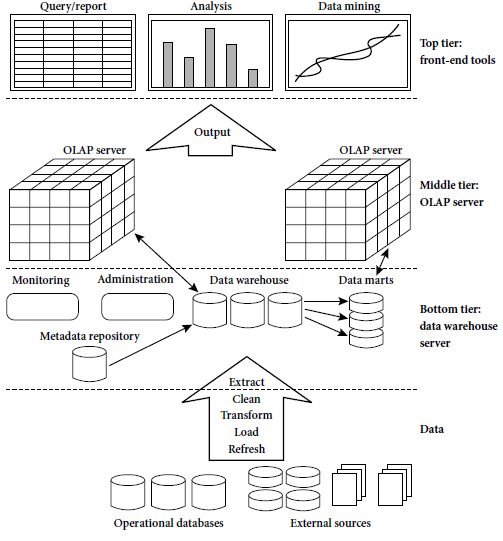
Four views regarding the design of a data warehouse:

* + Top-down view :Allows selection of the relevant information necessary for the data warehouse
  + Data source view :Exposes the information being captured, stored, and managed by operational systems
  + Data warehouse view: Includes fact tables and dimension tables. It represents the information that is stored inside the data warehouse, including precalculated totals and counts, as well as information regarding the source, date, and time of origin
  + Business query view : Sees the perspectives of data in the warehouse from the view of end-user

Data Warehouse Design Process

* Top-down, bottom-up approaches or a combination of both
  + Top-down: Starts with overall design and planning (mature)
  + Bottom-up: Starts with experiments and prototypes (rapid)
* From software engineering point of view
  + Waterfall: structured and systematic analysis at each step before proceeding to the next
  + Spiral: rapid generation of increasingly functional systems, short turn around time, quick turn around
* Typical data warehouse design process
  + Choose a business process to model, e.g., orders, invoices, etc.
  + Choose the *grain* (*atomic level of data*) of the business process
  + Choose the dimensions that will apply to each fact table record
  + Choose the measure that will populate each fact table record

**A THREE-TIER DATA WAREHOUSE ARCHITECTURE**



**BOTTOM TIER:**

The bottom tier is a warehouse database server that is almost always a relational database system. The data are extracted using application program interfaces known as gateways. Examples of gateways include ODBC JDBC. This tier also contains a metadata repository, which stores information about the data warehouse and its contents.

**MIDDLE TIER:**

The middle tier is an OLAP server that is typically implemented using either (1) a relational OLAP (ROLAP) model, that is, an extended relational DBMS that maps operations on multidimensional data to standard relational operations; or (2) a multidimensional OLAP (MOLAP) model, that is, a special-purpose server that directly implements multidimensional data and operations

**TOP TIER:**

The top tier is a front-end client layer, which contains query and reporting tools, analysis tools, and/or data mining tools.

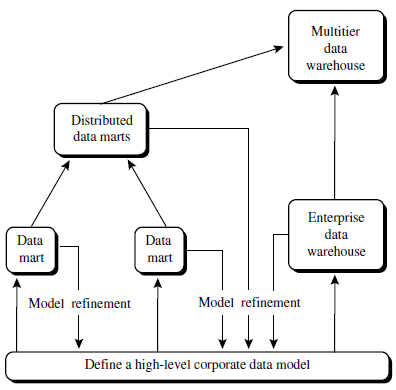
**ENTERPRISE WAREHOUSE**: collects all of the information about subjects spanning the entire organization. It provides corporate-wide data integration, usually from one or more operational systems or external information providers.

**DATA MART:**  A subset of corporate-wide data that is of value to specific groups of users. Its scope is confined to specific, selected groups, such as marketing data mart. *Independent* data marts are sourced from data captured from one or more operational systems or external information providers, or from data generated locally within a particular department or geographic area. *Dependent* data marts are sourced directly from enterprise data warehouses.

**VIRTUAL WAREHOUSE**

* + A set of views over operational databases
  + Only some of the possible summary views may be materialized.

First, a high-level corporate data model is defined within a reasonably short period Second, independent data marts can be implemented in parallel with the enterprise warehouse based on the same corporate data model set as above. Third, distributed data marts can be constructed to integrate different data marts via hub servers. Finally, a multitier data warehouse is constructed where the enterprise warehouse is the sole custodian of all warehouse data, which is then distributed to the various dependent data marts.



**DATA WAREHOUSE IMPLEMENTATION**

**Efficient Data Cube Computation**

* Data cube can be viewed as a lattice of cuboids
  + The bottom-most cuboid is the base cuboid
  + The top-most cuboid (apex) contains only one cell
  + The number of cuboids in an n-dimensional cube with L levels



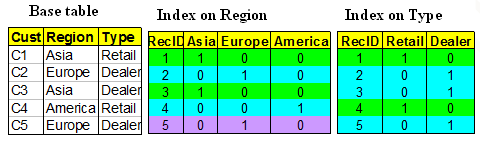
* + Materialize every (cuboid) (full materialization), none (no materialization), or some (partial materialization)
  + Selection of which cuboids to materialize is based on size, sharing, access frequency, etc.

**Indexing OLAP Data:**

**Bitmap Index**

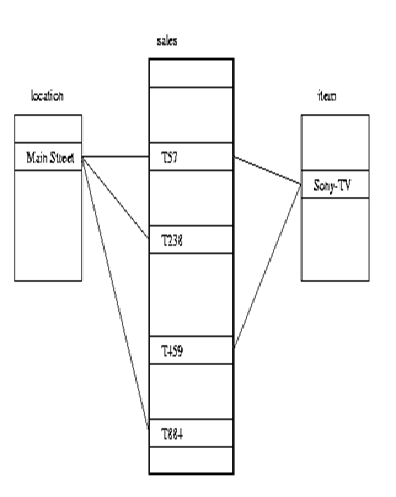
* Index on a particular column
* Each value in the column has a bit vector: bit-op is fast
* The length of the bit vector: number of records in the base table
* The  *i*-th bit is set if the  *i*-th row of the base table has the value for the indexed column

not suitable for high cardinality domains

****

**Join Indices**

* Join index: JI(R-id, S-id) where R (R-id, …) S (S-id, …)
* Traditional indices map the values to a list of record ids
  + It materializes relational join in JI file and speeds up relational join
* In data warehouses, join index relates the values of the dimensions of a start schema to rows in the fact table.
  + E.g. fact table: *Sales* and two dimensions *city* and *product*
    - A join index on *city* maintains for each distinct city a list of R-IDs of the tuples recording the Sales in the city
  + Join indices can span multiple dimensions

****

**Efficient Processing OLAP Queries**

* Determine which operations should be performed on the available cuboids
  + Transform drill, roll, etc. into corresponding SQL and/or OLAP operations, e.g., dice = selection + projection
* Determine which materialized cuboid(s) should be selected for OLAP op.
  + Let the query to be processed be on {brand, province\_or\_state} with the condition “year = 2004”, and there are 4 materialized cuboids available:

1) {year, item\_name, city}

2) {year, brand, country}

3) {year, brand, province\_or\_state}

4) {item\_name, province\_or\_state} where year = 2004

Which should be selected to process the query?

* Explore indexing structures and compressed vs. dense array structs in MOLAP