Queen Mary
University of London

ECS766 Data Mining

Week 5: Data Warehousing and On-line Analytical Processing

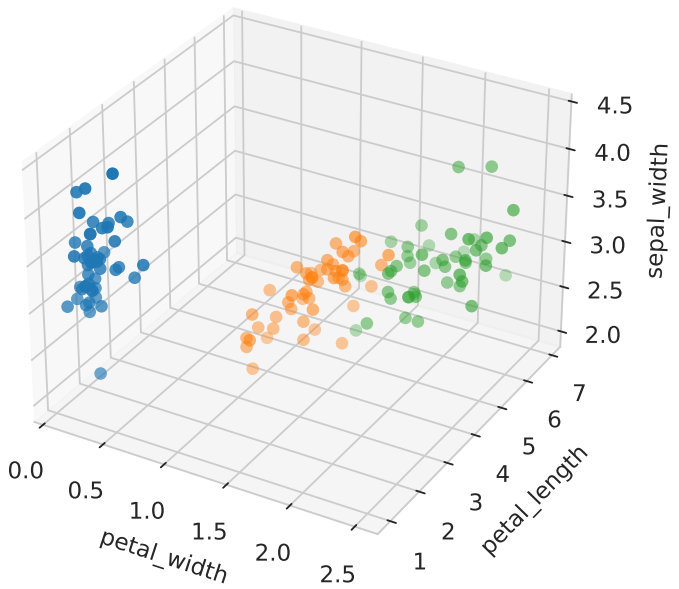
Emmanouil Benetos
emmanouil.benetos@qmul.ac.uk

October 2021

School of EECS, Queen Mary University of London

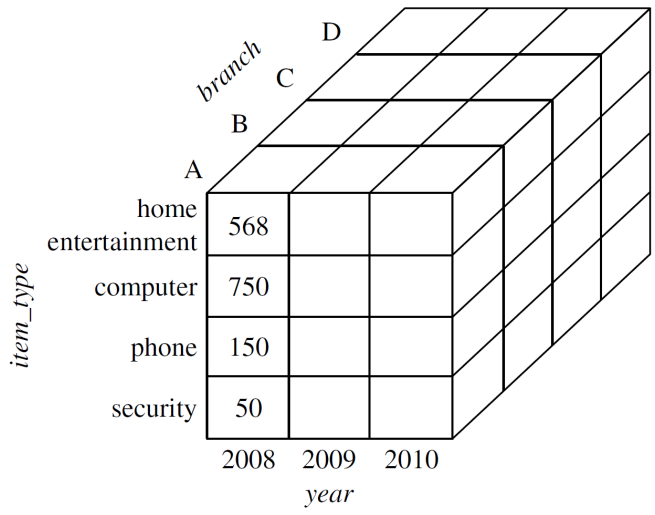
Last week: Data exploration and visualisation

- Data exploration
- Data summarisation
- Data visualisation



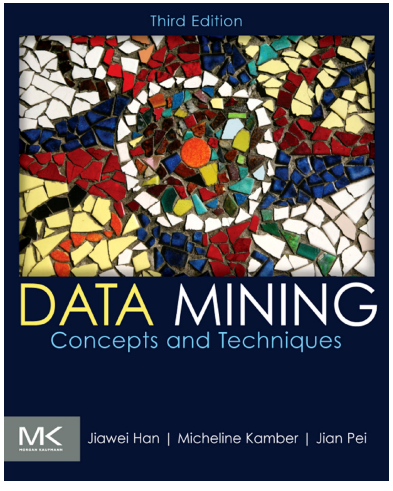
This week's contents

1. Data warehouse - basic concepts
2. Data warehouse modelling
3. Data warehouse design and usage
4. Data warehouse implementation



Reading

- Chapter 4 and Section 5.1 of J. Han, M. Kamber, J. Pei, “Data Mining: Concepts and Techniques”, 3rd edition, Elsevier/Morgan Kaufmann, 2012



Data warehouse - basic concepts

What is a Data Warehouse?

Defined in many different ways, but not rigorously

- A decision support database that is maintained **separately** from the organisation's operational database
- Supports **information processing** by providing a solid platform of consolidated, historical data for analysis

Data Warehouse

"A data warehouse is a **subject-oriented**, **integrated**, **time-variant**, and **nonvolatile** collection of data in support of management's decision-making process." —W. H. Inmon

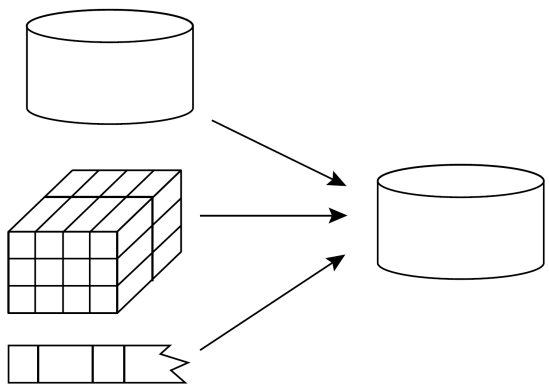
Data warehousing: The process of constructing and using data warehouses.

Data Warehouse – Subject-Oriented

- Organised around major subjects, such as *customer, product, sales*
- Focusing on the modeling and analysis of data for decision makers, not on daily operations or transaction processing
- Provides a simple and concise view around particular subject issues by **excluding data that is not useful in the decision support process**

Data Warehouse – Integrated

- Constructed by integrating multiple, heterogeneous data sources: relational databases, flat files, on-line transaction records
- Data cleaning and data integration techniques are applied – this ensures consistency in naming conventions, encoding structures, attribute measures, etc. among different data sources
- **Example**: Hotel price: differences on currency, tax, breakfast covered, and parking
- When data is moved to the warehouse, it is converted.



Data Warehouse – Time Variant

The time horizon for the data warehouse is significantly longer than that of operational systems.

- **Operational database**: current value data
- **Data warehouse data**: provide information from a historical perspective (e.g., past 5-10 years)

Every key structure in the data warehouse:

- Contains an element of time, explicitly or implicitly
- But the key of operational data may or may not contain a “time element”

Data Warehouse – Nonvolatile

Independence: a physically separate store of data transformed from the operational environment

Static: Operational update of data does not occur in the data warehouse environment.

- Does not require transaction processing, recovery, and concurrency control mechanisms.
- Requires only two operations in data accessing: initial loading of data and access of data.

OLTP vs. OLAP

OLTP: Online transactional processing

- Database management system (DBMS) operations
- Query and transactional processing

OLAP: Online analytical processing

- Data warehouse operations
- Drilling, slicing, dicing...

OLTP vs. OLAP

	OLTP	OLAP
users	clerk, IT professional	knowledge worker (e.g. analyst)
function	day to day operations	decision support
data	current, guaranteed up-to-date	historic, accuracy maintained over time
usage	repetitive	ad-hoc
access	read/write	mostly read
unit of work	short, simple transaction	complex query
# records accessed	tens	millions
# users	thousands	hundreds
DB size	100MB-GB	100GB-TB
metric	transaction throughput	query throughput, response
DB design	application-oriented	subject-oriented

Table 1: Comparison of OLTP and OLAP systems.

Why a Separate Data Warehouse?

High performance for both systems

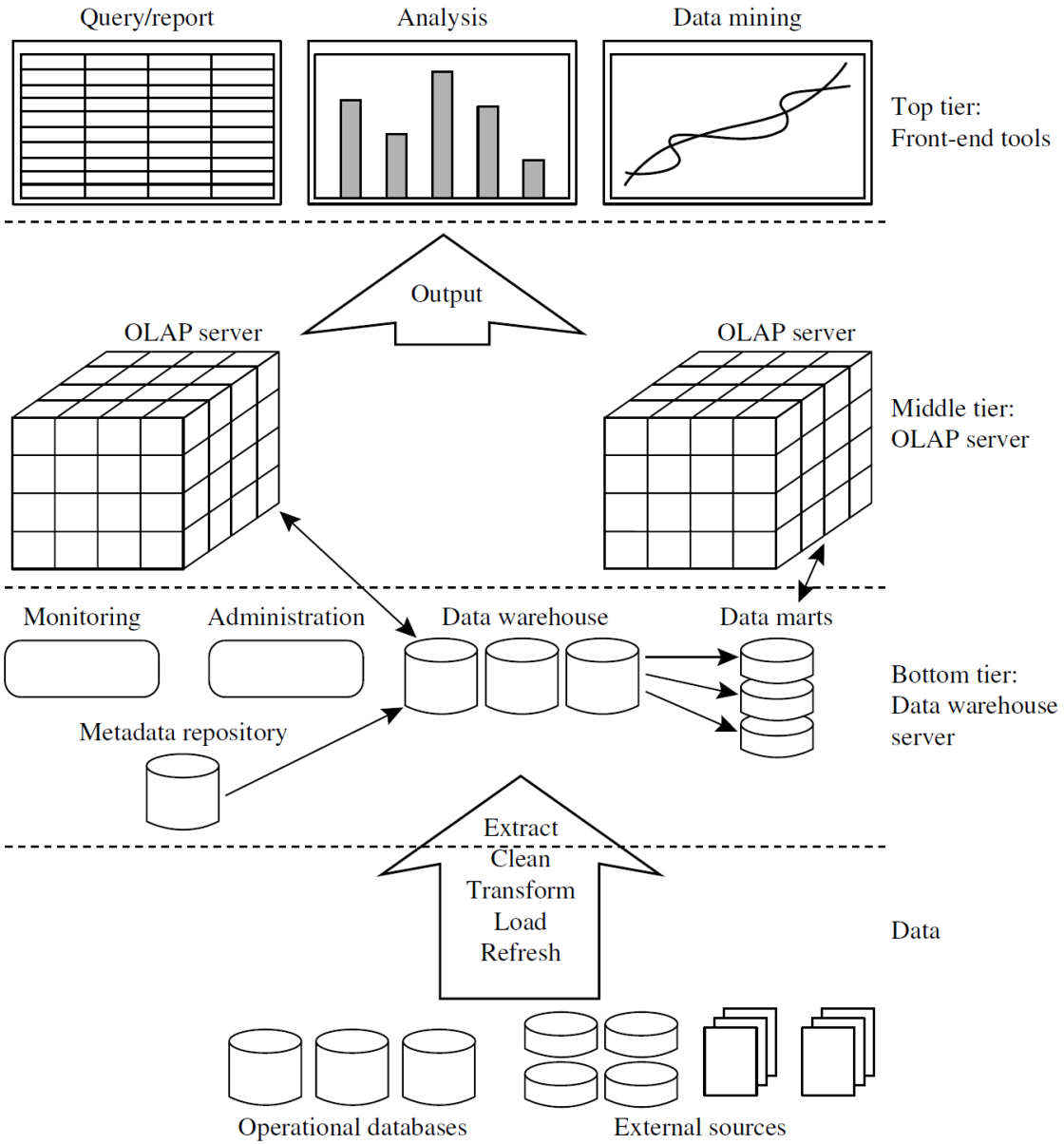
- DBMS – tuned for OLTP: access methods, indexing, concurrency control, recovery
- Warehouse – tuned for OLAP: complex OLAP queries, multidimensional view, consolidation

Different functions and different data:

- missing data: Decision support requires historical data which operational DBs do not typically maintain
- data consolidation: OLAP often needs consolidation (aggregation, summarisation) of data from heterogeneous sources
- data quality: different sources typically use inconsistent data representations, codes and formats which have to be reconciled

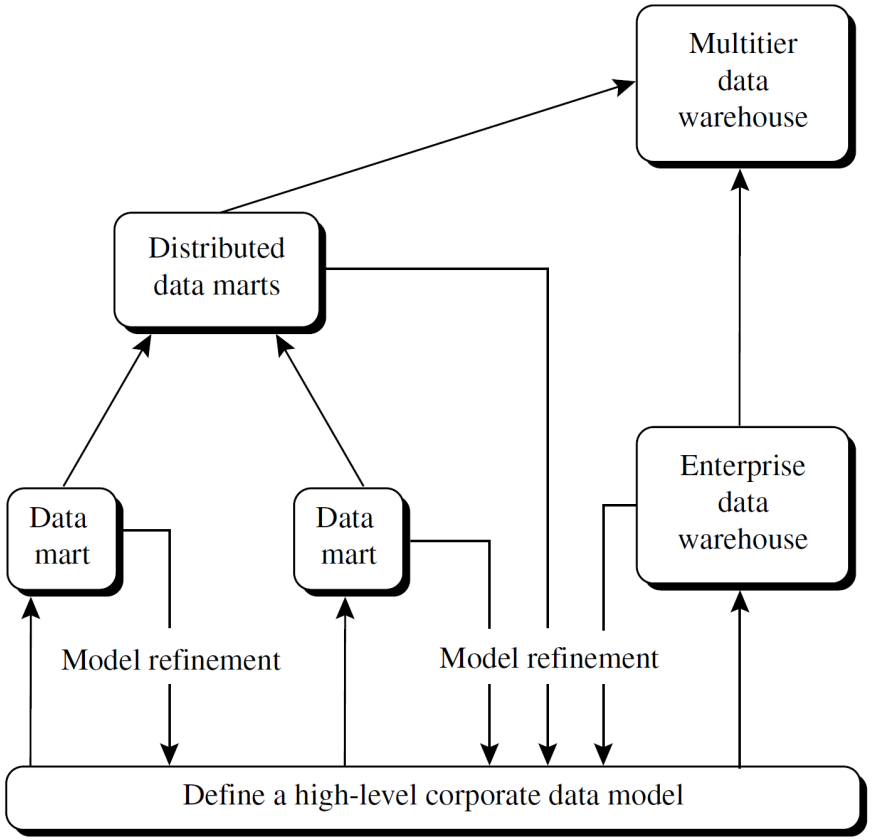
There are more and more systems which perform OLAP analysis directly on relational databases

Data Warehouse: A Multi-Tiered Architecture



Three Data Warehouse Models

1. Enterprise warehouse
2. Data Mart
3. Virtual warehouse



Extraction, Transformation, and Loading (ETL)

Data warehouse systems use back-end tools and utilities to populate and refresh their data:

- Data extraction: gets data from multiple, heterogeneous, and external sources.
- Data cleaning: detects errors in the data and rectify them when possible.
- Data transformation: converts data from legacy or host format to warehouse format.
- Load: sort, summarise, consolidate, compute views, check integrity, and build indicies and partitions.
- Refresh: propagates the updates from the data sources to the warehouse.

Metadata Repository

Metadata is data about data. When used in a data warehouse, metadata defines warehouse objects. It stores:

- Description of the structure of the data warehouse
- Operational meta-data
- The algorithms used for summarisation
- The mapping from operational environment to the data warehouse
- Data related to system performance
- Business data

Data warehouse modelling

From Tables and Spreadsheets to Data Cubes

A **data warehouse** is based on a multidimensional data model which views data in the form of a data cube

A **data cube** 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)
- The fact table contains measures (such as dollars_sold) and keys to each of the related dimension tables

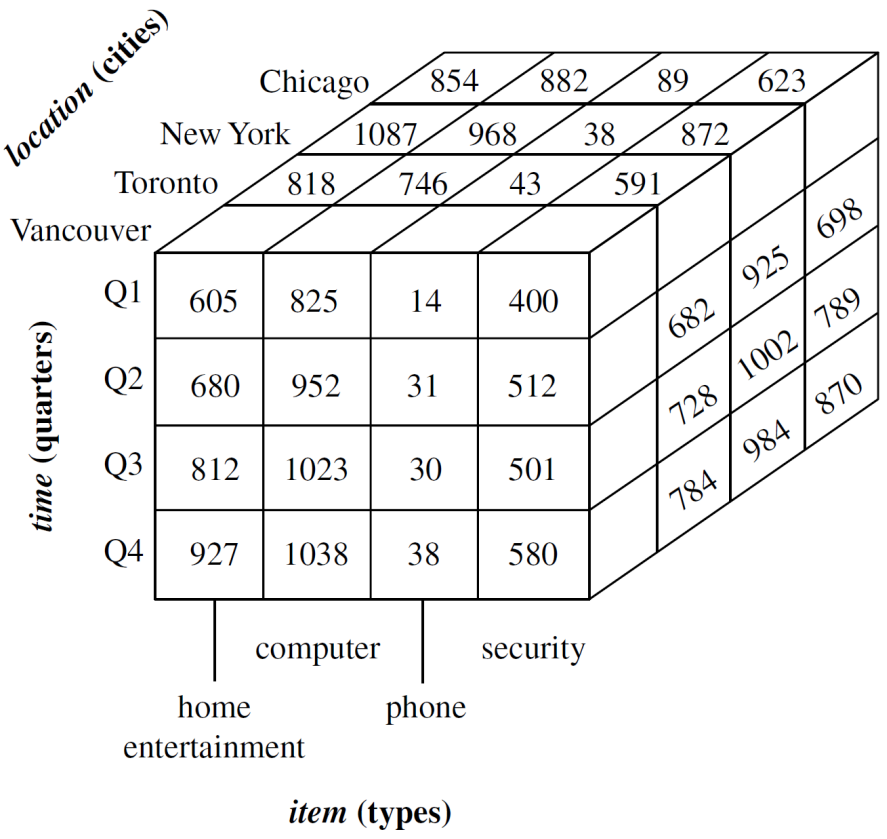
Although we usually think of cubes as 3-D geometric structures, in data warehousing data cubes are **n-dimensional**.

From Tables and Spreadsheets to Data Cubes

	location = "Chicago"				***location*** = "New York"				***location*** = "Toronto"				***location*** = "Vancouver"			
	item				***item***				***item***				***item***			
	home				*home*				*home*				*home*			
time	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

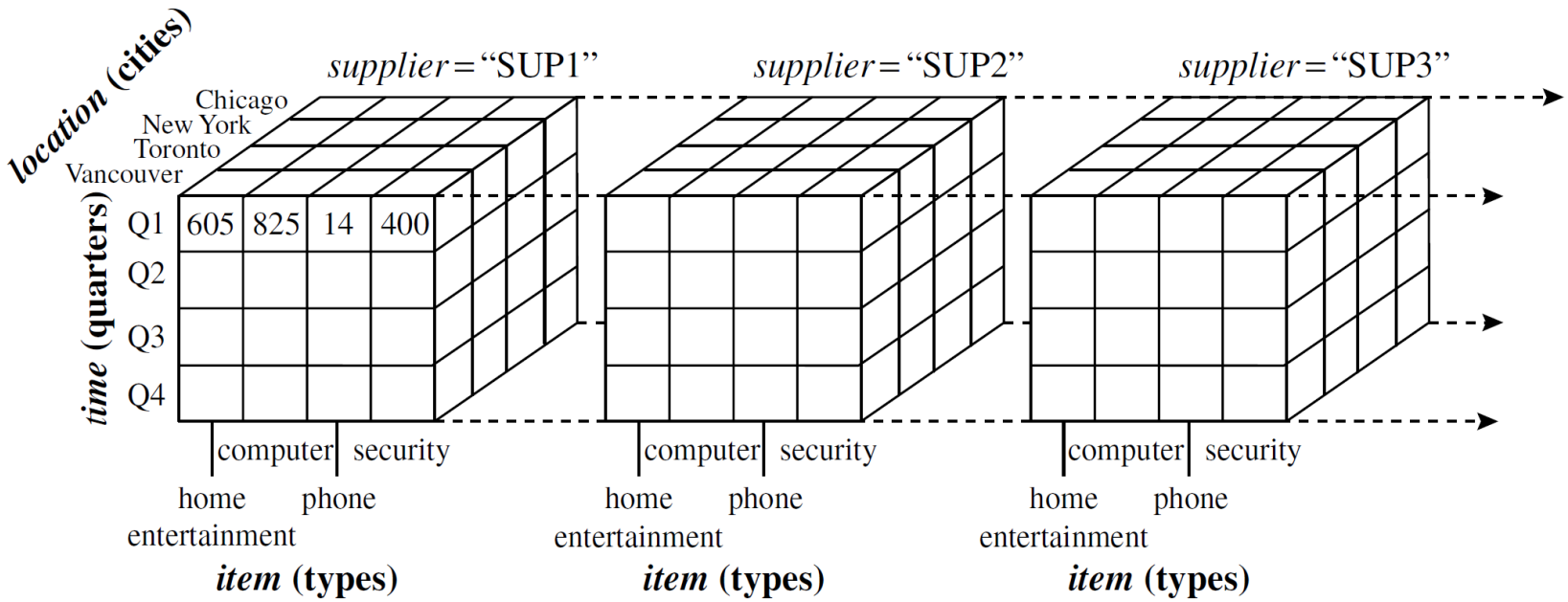
Example: A 3-dimensional representation of sales data according to time, item, and location.

From Tables and Spreadsheets to Data Cubes



Example: A 3-dimensional data cube representation of sales data according to time, item, and location.

From Tables and Spreadsheets to Data Cubes

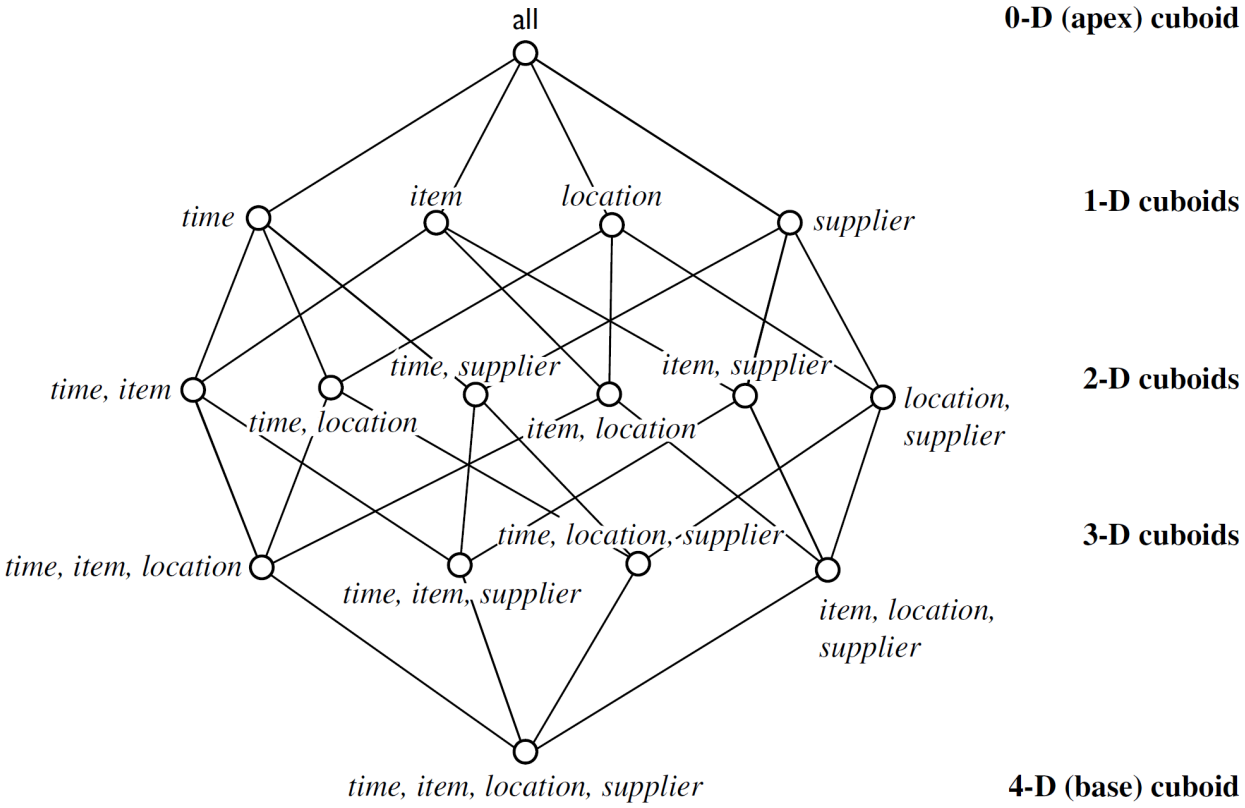


Example: A 4-dimensional data cube representation of sales data according to time, item, location, and supplier.

Cuboids

- Data cubes like the ones shown in the previous examples are often referred to as a cuboid.
- Given a set of dimensions, we can generate a cuboid for each of the possible subsets of the given dimensions.
- The result would form a lattice of cuboids.
- Base cuboid: the cuboid that holds the lowest level of summarisation.
- Apex cuboid: The 0-D cuboid, which holds the highest level of summarisation.

Lattice of Cuboids



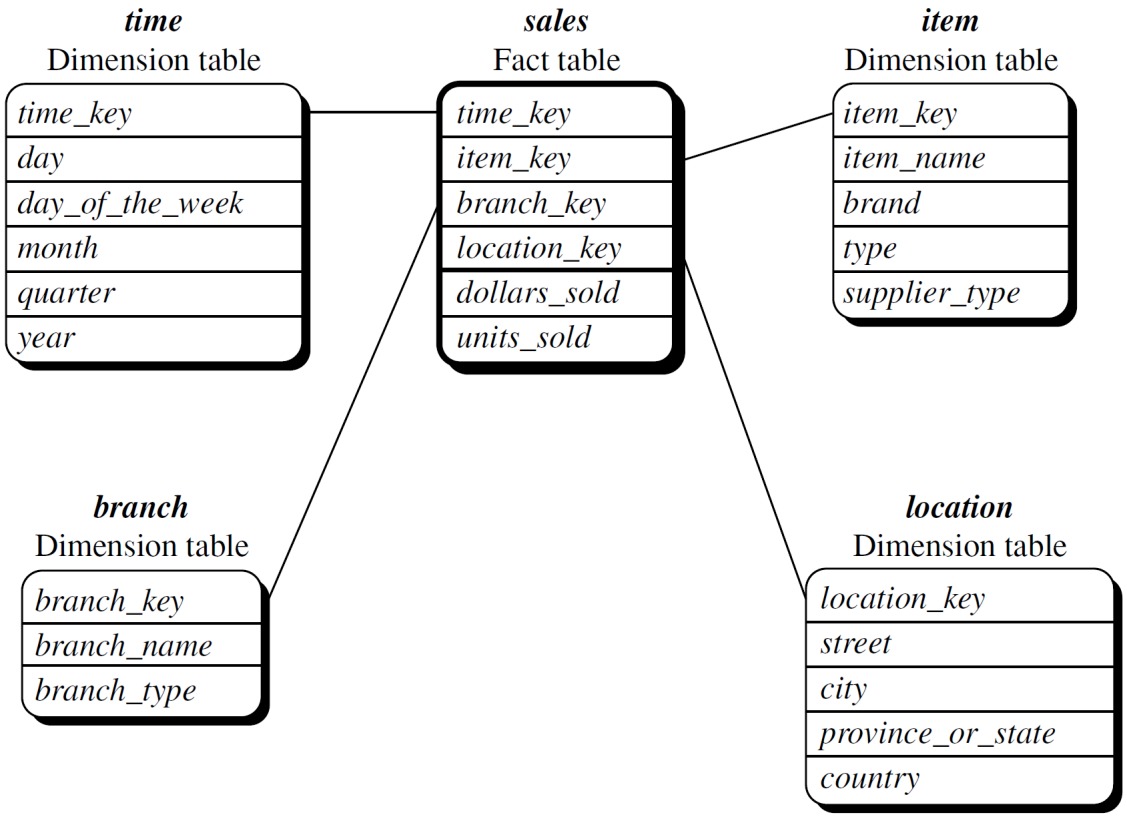
Example: Lattice of cuboids, making up a 4-D data cube for time, item, location, and supplier. Each cuboid represents a different degree of summarisation.

Schemas for Multidimensional Data Models

The most popular data model for a data warehouse is a multidimensional model. Common schemas of multidimensional models include:

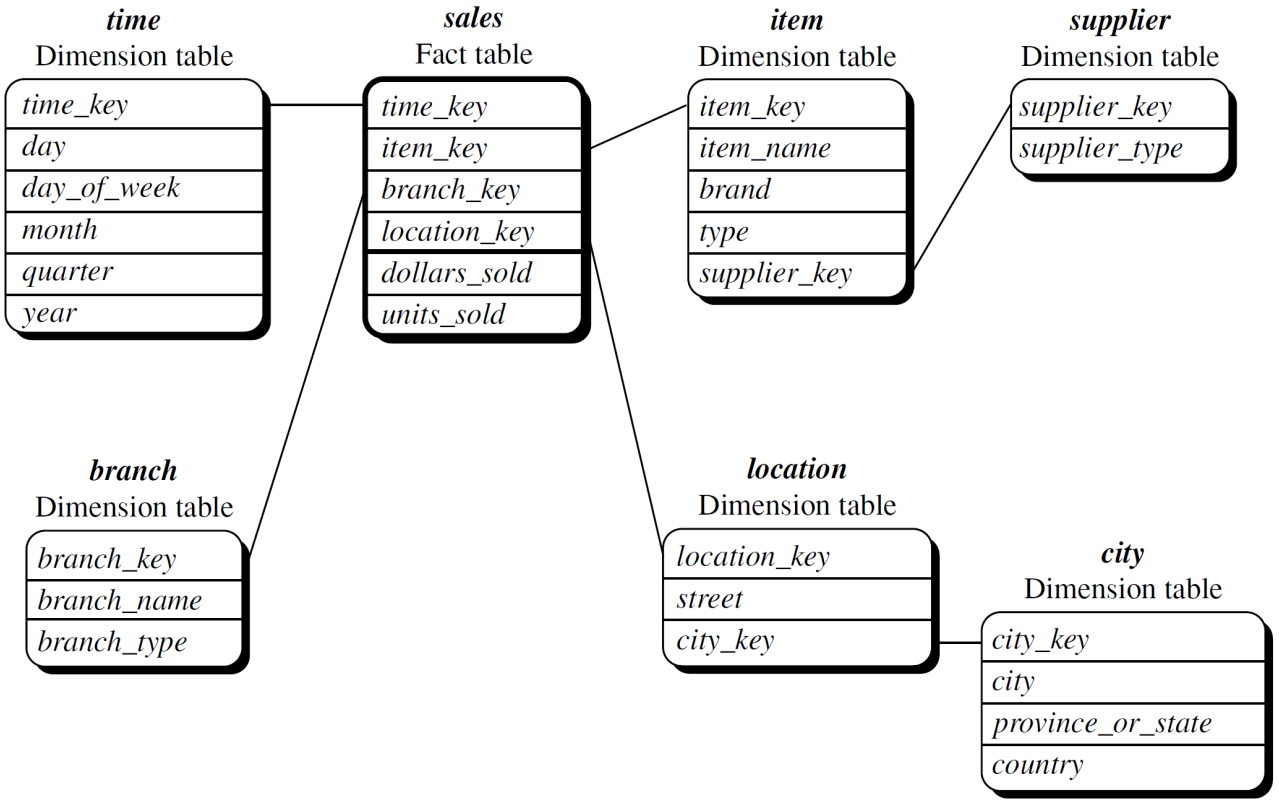
1. Star schema: A fact table in the middle connected to a set of dimension tables.
2. Snowflake schema: A refinement of the star schema where some dimensional hierarchy is normalised into a set of smaller dimension tables, forming a shape similar to a snowflake.
3. Fact constellations or galaxy schema: Multiple fact tables share dimension tables, viewed as a collection of stars.

Schemas for Multidimensional Data Models



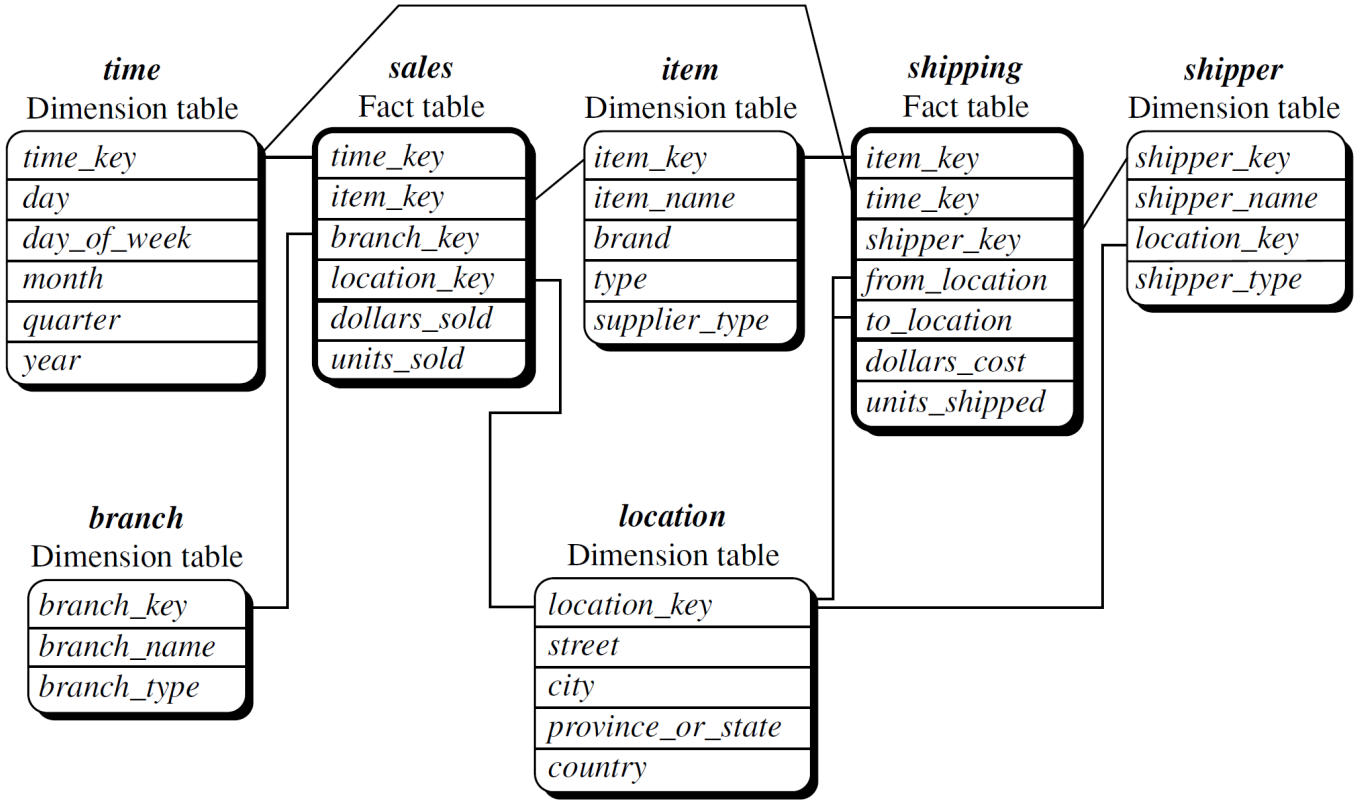
Example: Star schema of a sales data warehouse.

Schemas for Multidimensional Data Models



Example: Snowflake schema of a sales data warehouse.

Schemas for Multidimensional Data Models

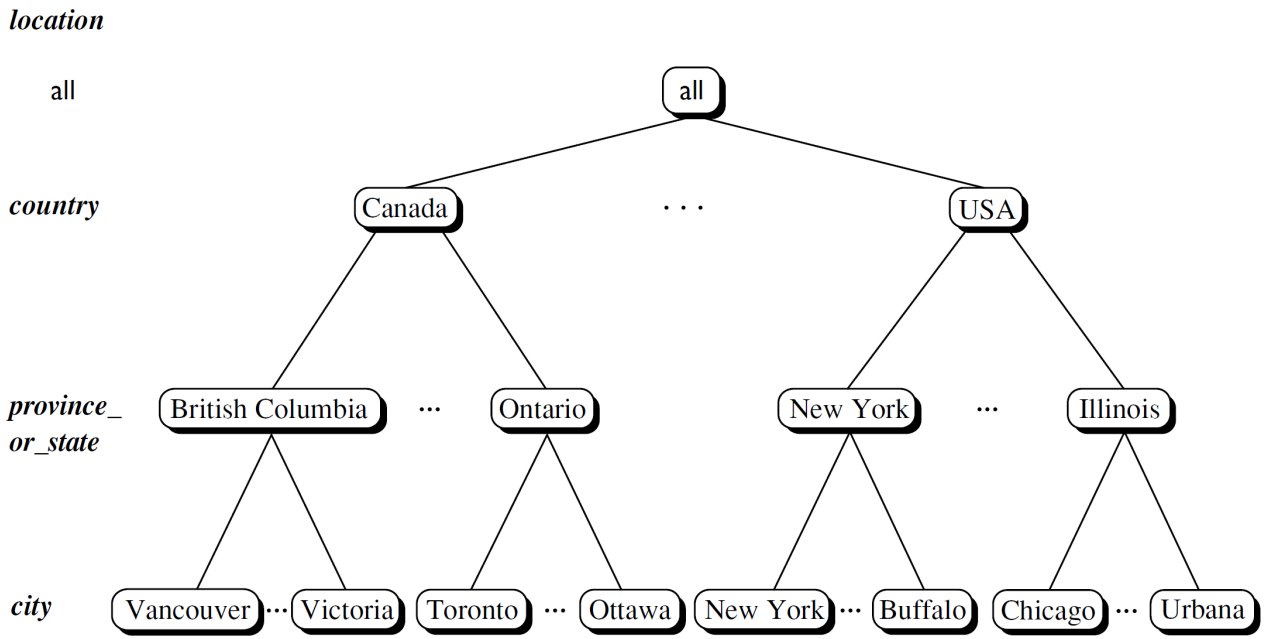


Example: Fact constellation schema of a sales and shipping data warehouse.

Concept Hierarchies

Concept Hierarchy

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

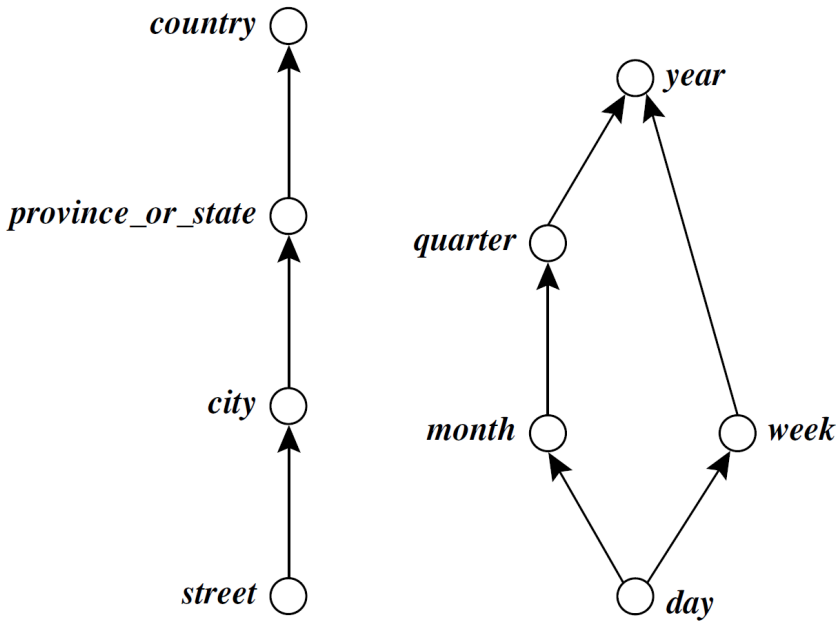


Example: A concept hierarchy for location.

Concept Hierarchies

In many concept hierarchies, attributes are related by a total order.

Alternatively, the attributes of a dimension may be organised in a partial order, forming a lattice.



Example: Hierarchical and lattice structures of attributes in warehouse dimensions. Left: a hierarchy for *location*. Right: a lattice for *time*.

Data Cube Measures

A data cube measure is a numeric function that can be evaluated at each point in the data cube space. Measures can be organised based on the kind of aggregate functions used:

- Distributive
  e.g. count(), sum(), min(), max()
- Algebraic
  e.g. avg(x), std()
- Holistic
  e.g. median(), mode(), rank()

Most data cube applications require efficient computation of distributive and algebraic measures. In contrast, it is difficult to compute holistic measures efficiently.

Typical OLAP Operations

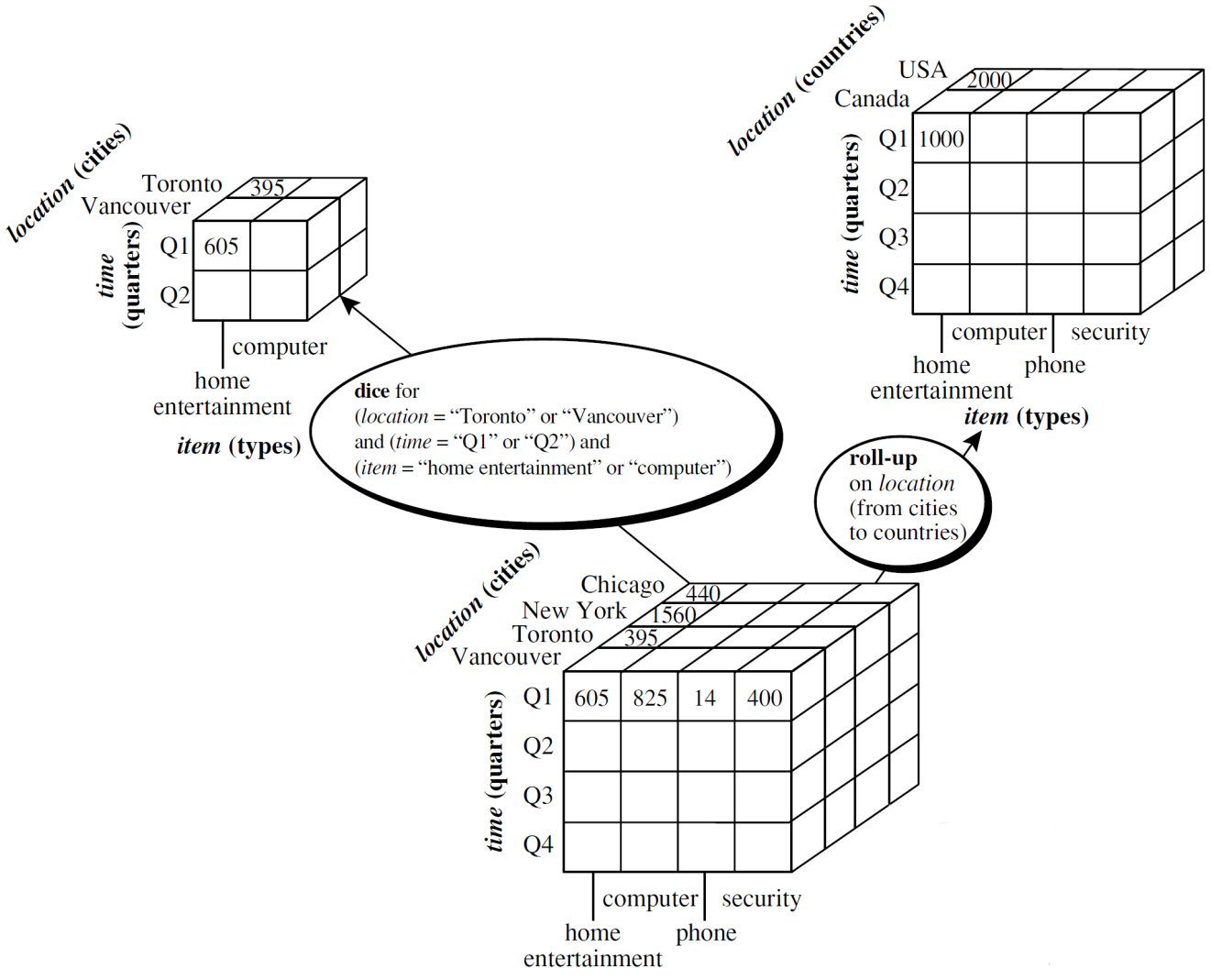


Figure: Dice and Roll-up OLAP operations.

Typical OLAP Operations

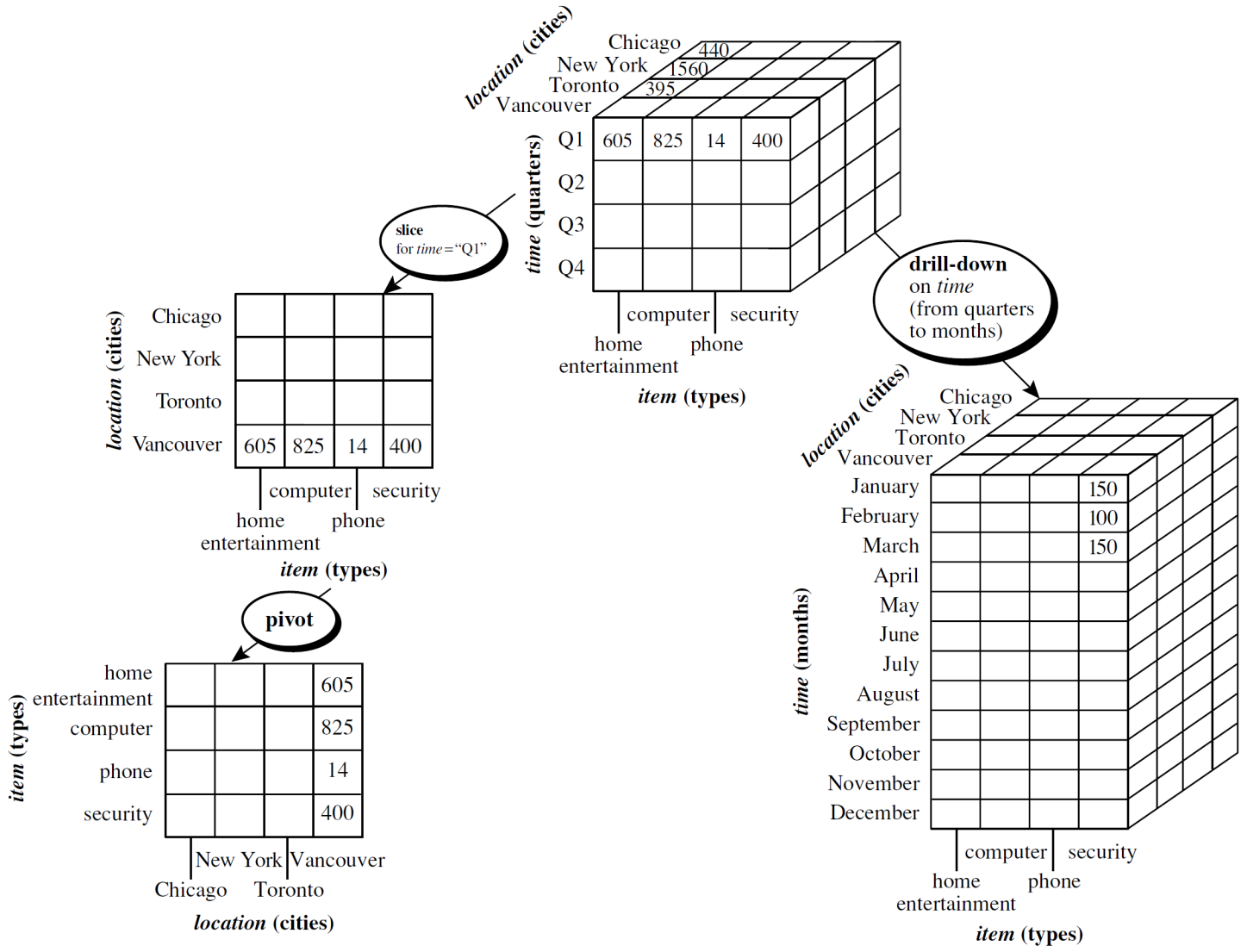


Figure: Pivot, slice, and drill-down OLAP operations.

Typical OLAP Operations

- Roll up (drill-up): summarise 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
- Slice: selection on one dimension
- Dice: selection on two (or more) dimensions
- Pivot (rotate): reorients the cube, used for visualisation

Data warehouse design and usage

Data Warehouse Design

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

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

- 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: consists of fact tables and dimension tables.
- Business query view: sees the perspectives of data in the warehouse from the view of the 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 a 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.

Data Warehouse Design Process

Typical data warehouse design process:

1. Choose a business process to model, e.g. orders, invoices.
2. Choose the grain (atomic level of data) of the business process, e.g. individual transactions.
3. Choose the dimensions that will apply to each fact table record, e.g. time, customer, supplier.
4. Choose the measure that will populate each fact table record, e.g. pounds sold, units sold.

Data Warehouse Usage

Three kinds of data warehouse applications:

- Information processing: supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs.
- Analytical processing: multidimensional analysis of data warehouse data. Supports basic OLAP operations (e.g. slice-dice, drilling, pivoting).
- Data mining: knowledge discovery from hidden patterns. Supports associations, constructing analytical models, classification/prediction, visualisation.

From OLAP to OLAM

Online Analytical Mining (OLAM)

OLAM, also called multidimensional data mining, integrates OLAP with data mining to uncover knowledge in multidimensional databases.

Why OLAM?

- High quality of data in data warehouses: DW contains integrated, consistent, cleaned data.
- Available information processing structure surrounding data warehouses
- OLAP-based exploratory data analysis: mining with drilling, dicing, pivoting, etc. results.
- Online selection of data mining functions: integration and swapping of multiple mining functions, algorithms, and tasks.

Data warehouse implementation

Efficient Data Cube Computation

A 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

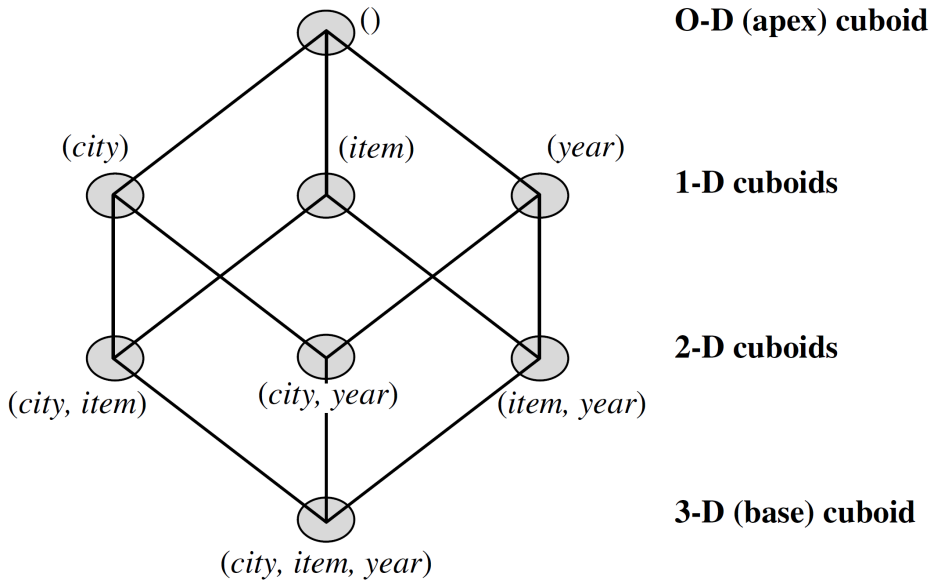


Figure: Lattice of cuboids, making up a 3-D data cube.

Efficient Data Cube Computation

How many cuboids in an n-dimensional cube with L levels?

- If there were no hierarchies associated with each dimension: 2^n
- In practice though, many dimensions do have hierarchies.
- The total number of cuboids that can be generated (including cuboids enerated by climbing up the hierarchies along each dimension) is:

$$T = \prod_{i=1}^{n} (L_i + 1)$$

where L_i is the number of levels associated with dimension i.

Efficient Data Cube Computation

It is unrealistic to precompute and **materialise** all of the cuboids that can possibly be generated for a data cube (i.e. from a base cuboid).

There are three choices for data cube materialisation given a base cuboid:

1. Full materialisation: Materialise every (cuboid)
2. No materialisation: Materialise none (cuboid)
3. Partial materialisation: Materialise some cuboids

Which cuboids to materialise?
Selection based on size, sharing, access frequency, etc.

Indexing OLAP Data

To facilitate efficient data accessing, most data warehouse systems support index structures.

The bitmap indexing method is popular in OLAP products because it allows quick searching in data cubes.

The join indexing method is based on relational database query processing. Join indexing registers the joinable rows of two relations from a relational database.

Indexing OLAP Data: Bitmap Index

Bitmap indexing

- Each value in the column has a bit vector: bit-op is fast
- The length of the bit vector: # 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

Base Table

RID	Region	Type
R1	Asia	Retail
R2	Europe	Wholesale
R3	Asia	Wholesale
R4	America	Retail
R5	Europe	Wholesale

Index on Region

RID	Asia	Europe	America
R1	1	0	0
R2	0	1	0
R3	1	0	0
R4	0	0	1
R5	0	1	0

Index on Type

RID	Retail	Dealer
R1	1	0
R2	0	1
R3	0	1
R4	1	0
R5	0	1

Indexing OLAP Data: Join Index

- In data warehouses, join index relates the values of the dimensions of a star schema to rows in the fact table.

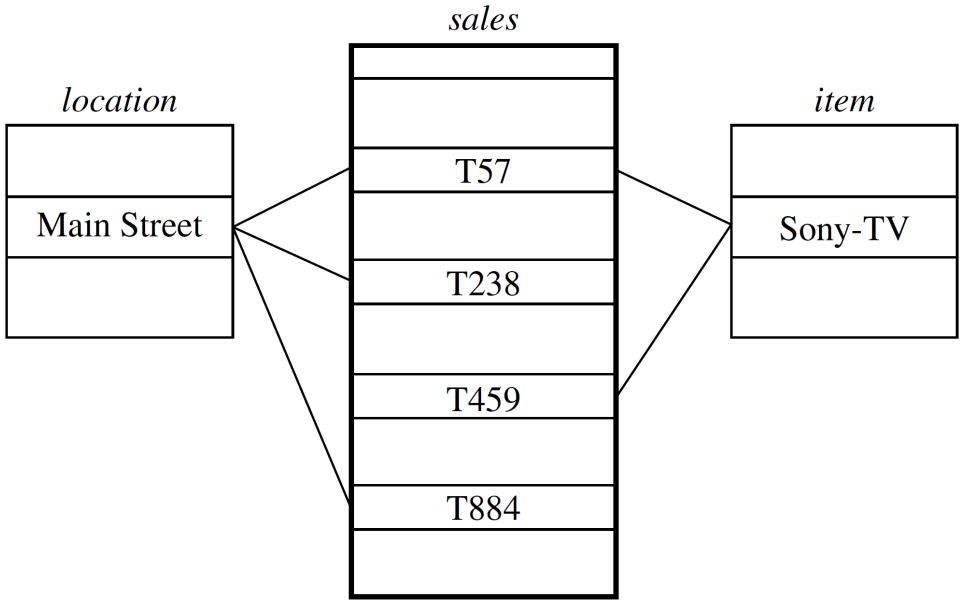


Figure: Linkages between a sales fact table and location and item dimension tables.

Efficient Processing of OLAP Queries

Given materialised views, query processing should proceed as follows:

1. Determine which operations should be performed on the available cuboids.
2. Determine to which materialised cuboid(s) the relevant operations should be applied.

Efficient Processing of OLAP Queries

Example: suppose a data cube of sales in terms of time, item, and location. A query to be processed is on {brand,region} for a specific "year=2019". Also suppose there are 4 meterialised cuboids:

- cuboid 1: {year, item name, city}
- cuboid 2: {year, brand, country}
- cuboid 3: {year, brand, region}
- cuboid 4: {item name, region}, where year = 2019

Which of these four cuboids should be selected to process the query? How would the costs of each cuboid compare if used to process the query?

OLAP Server Architectures

Relational OLAP (ROLAP)

- Use relational or extended-relational DBMS to store and manage warehouse data and OLAP middle ware
- Include optimisation of DBMS backend, implementation of aggregation navigation logic, and additional tools and services
- Greater scalability

Multidimensional OLAP (MOLAP)

- Sparse array-based multidimensional storage engine
- Fast indexing to pre-computed summarised data

Hybrid OLAP (HOLAP)

- Flexibility, e.g. low level: relational, high-level: array

Summary

Data warehousing

- A data cube consists of dimensions & measures
- Star schema, snowflake schema, fact constellations
- OLAP operations: drilling, rolling, slicing, dicing and pivoting

Data Warehouse Architecture, Design, and Usage

- Multi-tiered architecture
- Business analysis design framework
- Information processing, analytical processing, data mining, OLAM (Online Analytical Mining)

Implementation: Efficient computation of data cubes

- Partial vs. full vs. no materialisation
- Indexing OALP data: Bitmap index and join index
- OLAP query processing
- OLAP servers: ROLAP, MOLAP, HOLAP

Questions?
also please use the forum on QM+