Data Warehousing

Lecture 2 Modelling of Data Warehouses and OLAP

CITS3401
CITS5504

Dr. Sirui Li

Computer Science and Software Engineering

School of Physics, Mathematics and Computing

Acknowledgement: The lecture slides are prepared based on online resources.

Lecture Outline

- Storing Data in Data Warehouse
- Fact Tables and Dimension Tables
- Schema of a Data Warehouse
  - Star, Snowflakes, Fact Constellations
- OLAP Operations
  - Roll up, Drill down, Slice & Dice, Pivot

Storing Data in Data Warehouse

- Data Warehouse is a more advanced database management system on storing large amount of data.
- Data Warehouse is built on top of a database management system (e.g. RDBMS like SQL Server)
- Data Warehouses store data using tables like DB systems.
- Data Cube is used to support OLAP operations.

Multi-dimensional View of Data (2-D)

Table 4.2 2-D View of Sales Data for *AllElectronics* According to *time* and *item*

	location = *"Vancouver"*			
	item *(type)*			
time *(quarter)*	*home entertainment*	*computer*	*phone*	*security*
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

Note: The sales are from branches located in the city of Vancouver. The measure displayed is *dollars_sold* (in thousands).

Multi-dimensional View of Data (3-D)

Table 4.3 3-D View of Sales Data for *AllElectronics* According to *time*, *item*, and *location*

	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

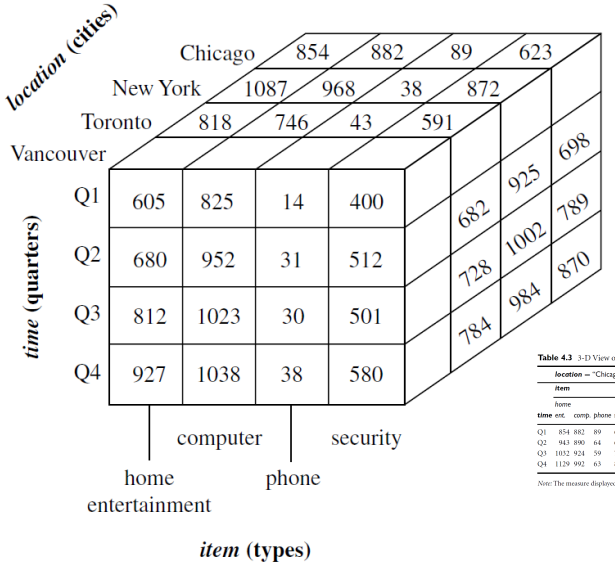
Note: The measure displayed is *dollars_sold* (in thousands).

Table 4.2 2-D View of Sales Data for *AllElectronics* According to *time* and *item*

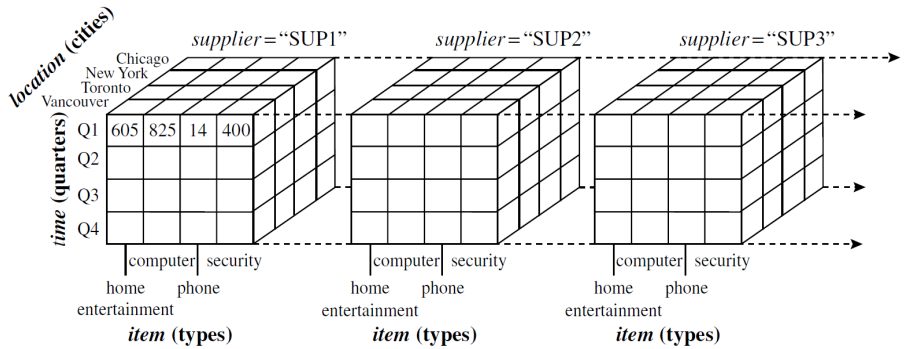
	location = "Vancouver"			
	item (type)			
time (quarter)	*home entertainment*	*computer*	*phone*	*security*
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

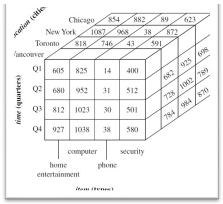
Note: The sales are from branches located in the city of Vancouver. The measure displayed is *dollars_sold* (in thousands).

Multi-dimensional View of Data (Data Cube)



Data Cube: Don't confine Data to 3-D





Data Cube is a metaphor for multi-dimensional data storage.

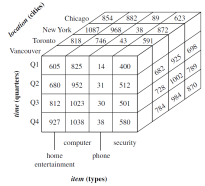
5-D data cube: a series of 4-D data cubes

Lecture Outline

- Storing Data in Data Warehouse
- Fact Tables and Dimension Tables
- Schema of a Data Warehouse
  - Star, Snowflakes, Fact Constellations
- OLAP Operations
  - Roll up, Drill down, Slice & Dice, Pivot

From Tables to Data Cubes

- A data warehouse is based on a multi-dimensional data model which views data in the form of a data cube
- A data cube, is organised around a central theme, such as sales, allows data to be modelled and viewed in multiple dimensions
  - Dimension tables, such as item (item_name, brand, type), or time (day, week, month, quarter, year) or location (branch, city, state, country)
  - Fact table contains measures of central theme (such as dollars_sold, units sold) and keys to each of the related dimension tables

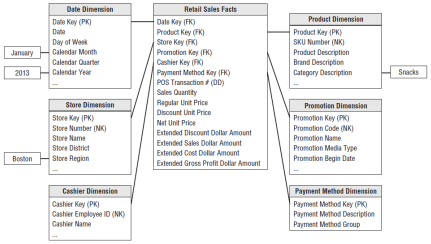


Fact Tables

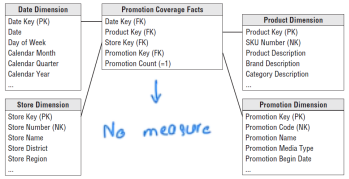
- Each fact table contains measurements (e.g. dollar_sold) about a process of interest.
- Each fact row contains two things:
  - Numerical measure columns
  - Foreign keys to dimension tables
- Properties of fact tables:
  - Very big
    - Often millions or billions of rows
  - Narrow
    - Small number of columns
  - Often append new rows to the fact table
    - New events in the world → new rows in the fact table
- Uses of fact tables:
  - Obtain measurements from the fact table
  - Aggregate measurements from columns of the fact table.

Fact Tables and Factless Fact Tables

DD → Degenerate dimension



Fact Table with Measures



Factless Fact Table with a dummy measure

• Help to track events

Factless Fact Tables

Tracking student attendance events

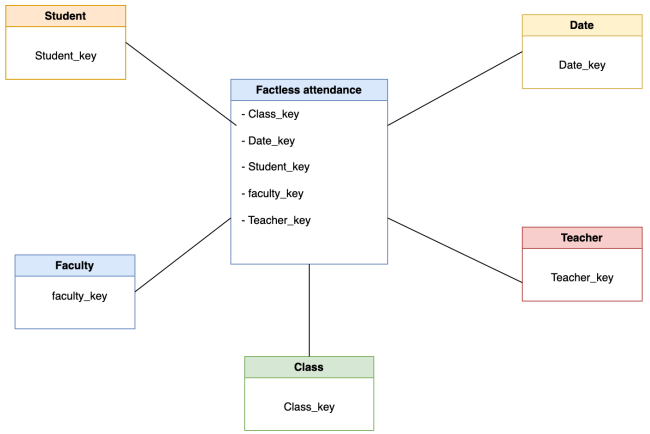


Image from: link

Grain of a fact table

- Grain of a fact table = the meaning of one fact table row
- Determines the maximum level of detail of the warehouse
- Example grain statements: (*one fact row represents a…*)
  - Line item from a cash register receipt
  - Boarding pass to get on a flight
  - Daily snapshot of inventory level for a product in a warehouse
  - Sensor reading per minute for a sensor
  - Student enrolled in a course
- Finer-grained fact tables:
  - are more expressive
  - have more rows
- Trade-off between performance and expressiveness
  - Rule of thumb: error in favor of expressiveness
  - Pre-computed aggregates can solve performance problems

Measures

- A data cube measure is a numeric function that can be evaluated at each point in the data cube space.
- A measure value is computed for a given point by aggregating the data corresponding to the respective dimension–value pairs defining the given point.

Types of Measures

- **Types of measures**
  - Distributive:
    - An aggregate function is distributive if it can be computed in a distributed manner by applying the same function on partitioned sets.
    - count(), min(), and max() are distributive aggregate functions.
  - Algebraic:
    - An aggregate function is algebraic if it can be computed by an algebraic function with *M* arguments (where *M* is a bounded positive integer), each of which is obtained by applying a *distributive* aggregate function.
    - avg() (average) can be computed by sum()/count(), where both sum() and count() are distributive
    - standard_deviation().
  - Holistic:
    - median(), mode(), and rank().

Dimension Tables

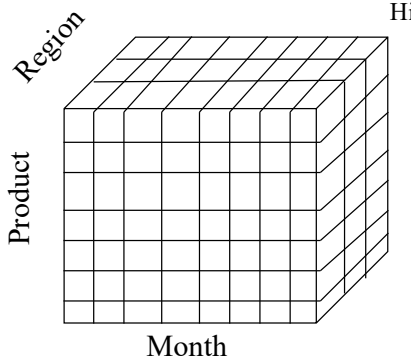
- Each one corresponds to a real-world object or concept.
  - Examples: Customer, Product, Date, Employee, Region, Store, Promotion, Vendor, Partner, Account, Department
- Properties of dimension tables:
  - Contain many descriptive columns
    - Dimension tables are wide (dozens of columns)
  - Generally don't have too many rows
    - At least in comparison to the fact tables
    - Usually < 1 million rows
  - Contents are relatively static
    - Almost like a lookup table
- Uses of dimension tables
  - Filters are based on dimension attributes
  - Grouping columns are dimension attributes
  - Fact tables are referenced through dimensions

Dimension Tables

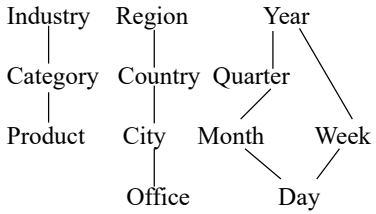
- Determine a candidate key based on the grain statement.
  - Example: a student enrolled in a course
  - (Course, Student, Term) is a candidate key
- Add other relevant dimensions that are functionally determined by the candidate key.
  - For example, Instructor and Classroom
    - Assuming each course has a single instructor!

Concept Hierarchy in Dimensions

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



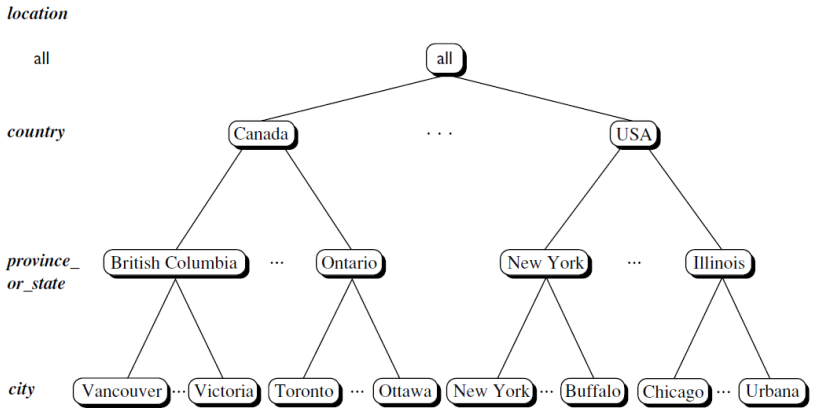
Dimensions: Product, Location, Time
Hierarchical summarisation paths



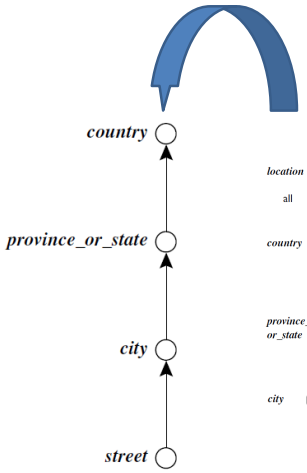
The Role of Concept Hierarchies

- **Concept Hierarchy** → we use real data
  - Defines a sequence of mappings from a set of low-level concepts to high-level, more general concepts.
- **Schema Hierarchy** → We user column name
  - A concept hierarchy that is a total or partial order among attributes in a database schema
    - Total order: street < city < province_or_state < country
    - Partial order: day < {month < quarter; week} < year
- **Set-grouping Hierarchy**
  - defined by discretising or grouping values for a given dimension or attribute.

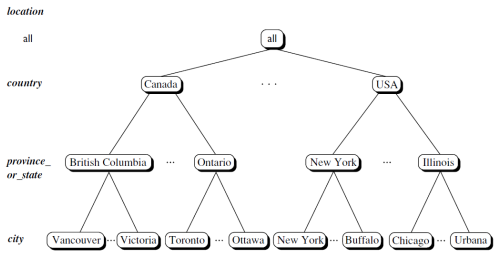
Example Concept Hierarchies



Example Concept Hierarchies: Schema Hierarchy

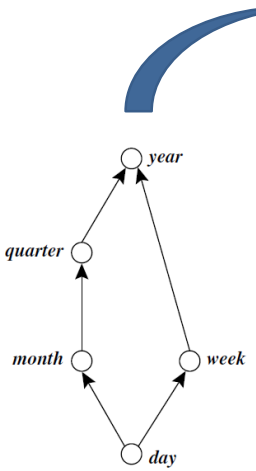


Schema Hierarchy - Location

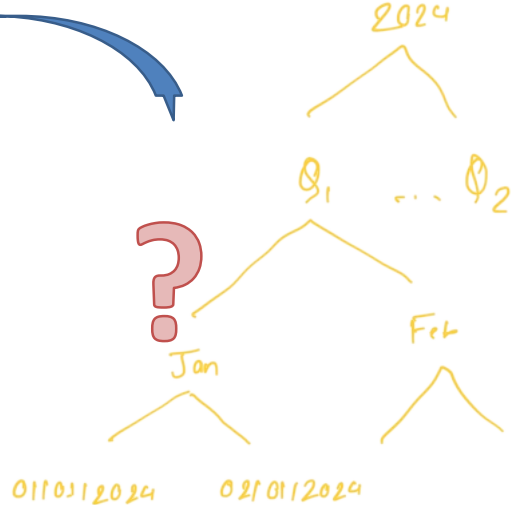


Concept Hierarchy - Location

Example Concept Hierarchies: Schema Hierarchy

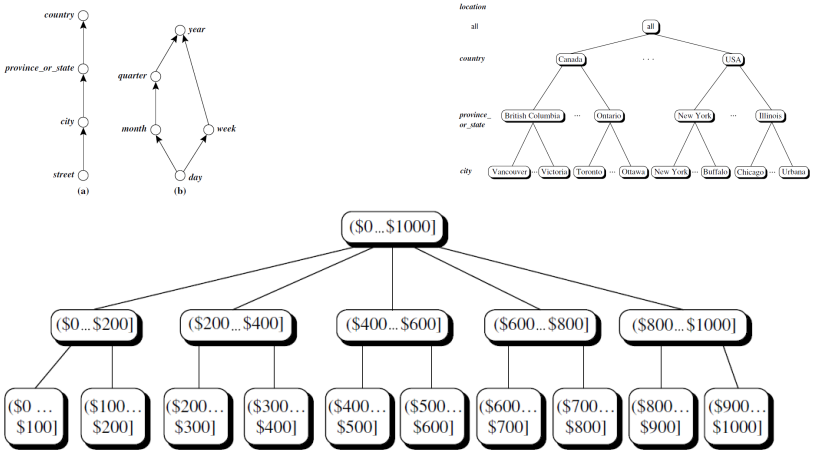


Schema Hierarchy – Date



Concept Hierarchy – Date

Example Concept Hierarchies: Set-grouping Hierarchy



A concept hierarchy for *price*.

Facts vs. Dimension Tables

Facts

- Narrow
- Big (many rows)
- Numeric
- Growing over time

Dimensions

- Wide
- Small (few rows)
- Descriptive
- Static

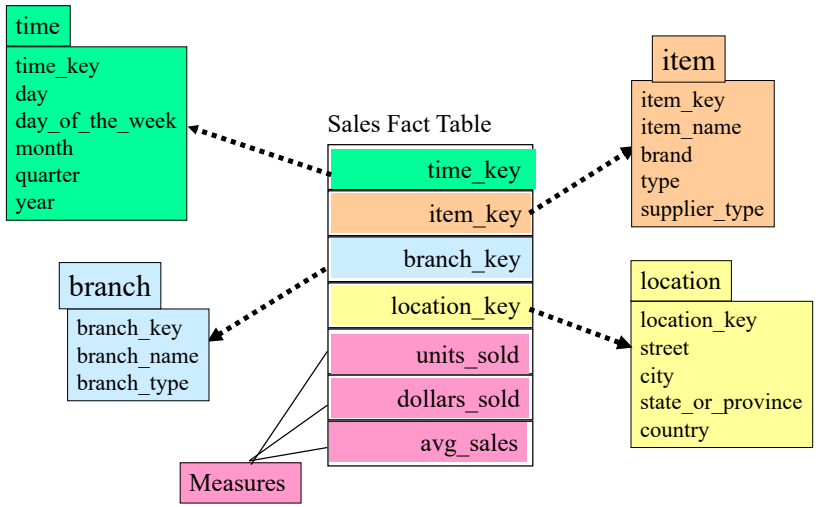
Lecture Outline

- Storing Data in Data Warehouse
- Fact Tables and Dimension Tables
- Schema of a Data Warehouse
  - Star, Snowflakes, Fact Constellations
- OLAP Operations
  - Roll up, Drill down, Slice & Dice, Pivot

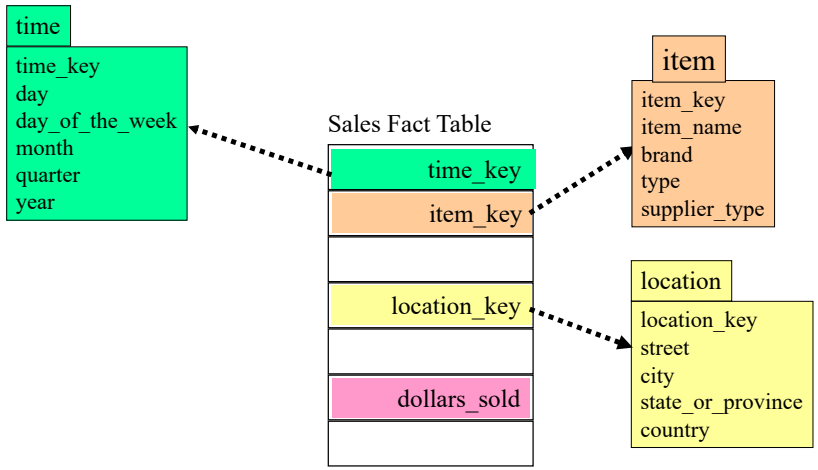
Schema

- **Star Schema**
  - A fact table in the middle connected to a set of dimension tables
- **Snowflake Schema**
  - Some dimensional hierarchy is normalised into a set of smaller dimension tables, forming a shape similar to snowflake.
  - Reduces redundancy at the cost of efficiency.
- **Galaxy schema (Fact Constellation)**
  - Multiple fact tables share dimension tables
  - Viewed as a collection of stars - Galaxy schema

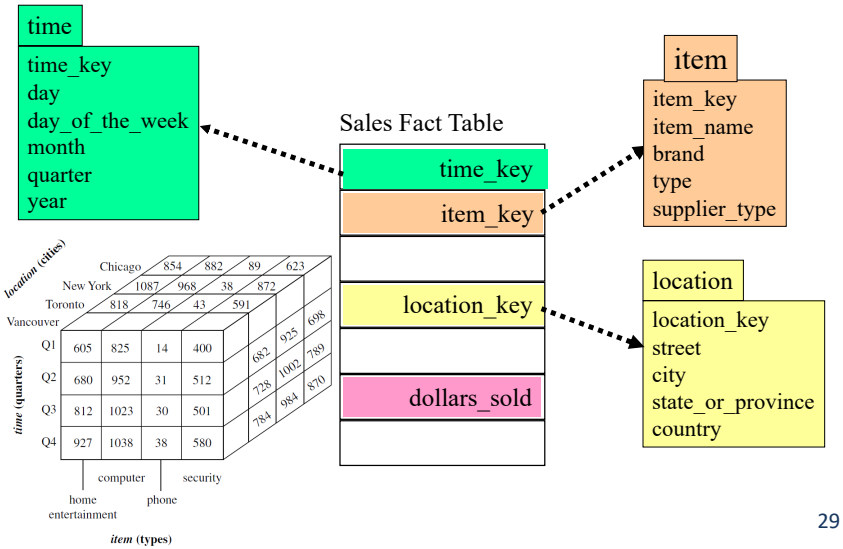
Star Schema



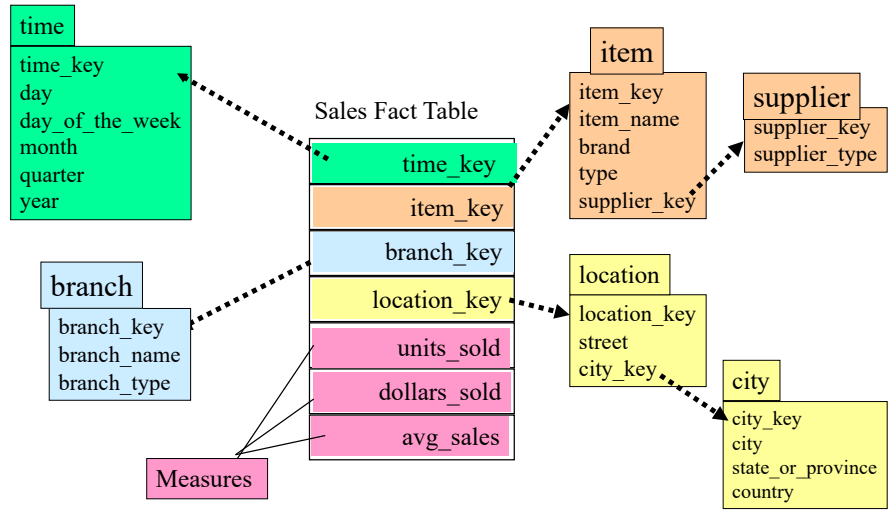
Star Schema: Ignore some fields



Star Schema: Representing Data Cube with Four Tables

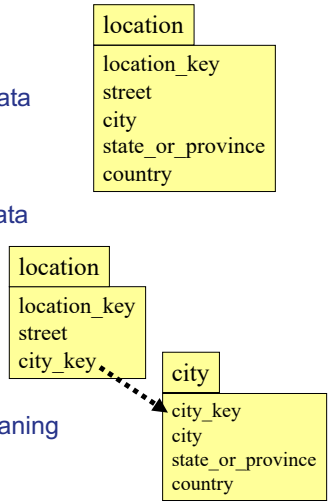


Snowflake Schema

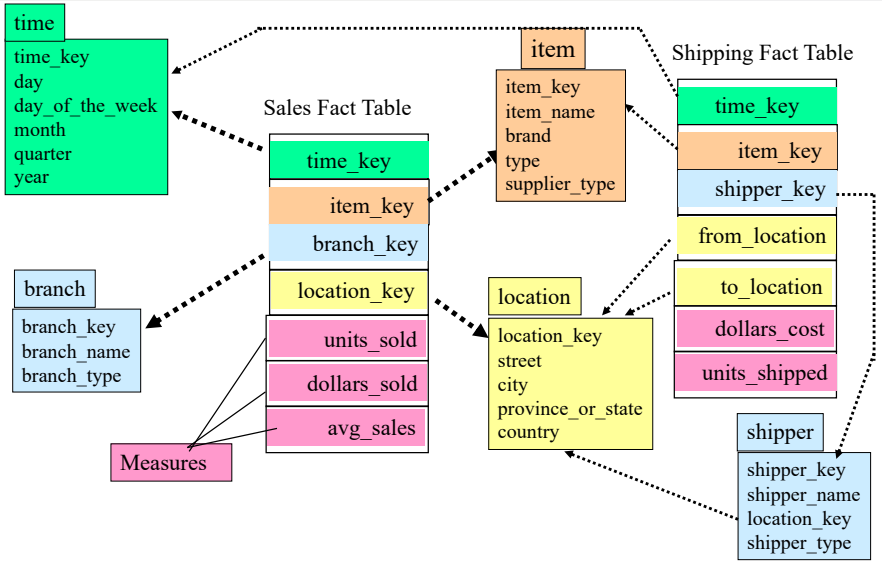


Star Schema v.s. Snowflake Schema

- **Star Schema**
  - Fewer tables, faster when browsing data
  - Has more redundant information
- **Snowflake Schema**
  - More tables, slower when browsing data
  - Reduces redundancy
- **Redundancy means:**
  - More storage
  - More work in data integration and cleaning



Fact Constellation – Galaxy Schema



Lecture Outline

- Storing Data in Data Warehouse
- Fact Tables and Dimension Tables
- Schema of a Data Warehouse
  - Star, Snowflakes, Fact Constellations
- OLAP Operations
  - Roll up, Drill down, Slice & Dice, Pivot

Typical OLAP Operations

For project this should be enough

- **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 and dice:**
  - *project and select*
- **Pivot (rotate):**
  - *reorient the cube, visualisation, 3D to series of 2D planes.*
- **Other operations (aside)**
  - *drill across: involving (across) multiple fact tables*
  - *drill through: through the bottom level of the cube to its back-end relational tables (using SQL)*

Data Cube Queries

- **Cross-tabulation**
  - "Cross-tab" for short
  - Report data grouped by 2 dimensions
  - Aggregate across other dimensions
  - Include subtotals
- **Operations on a cross-tab**
  - Roll up (further aggregation)
  - Drill down (less aggregation)

Autos Sold

	VIC	NSW	WA	Total
Jul	45	33	30	108
Aug	50	36	42	128
Sep	38	31	40	109
Total	133	100	112	345

Roll Up and Drill Down

Autos Sold

	VIC	NSW	WA	Total
Jul	45	33	30	108
Aug	50	36	42	128
Sep	38	31	40	109
Total	133	100	112	345

Roll up by Month

Autos Sold

VIC	NSW	WA	Total
133	100	112	345

Drill down by Color

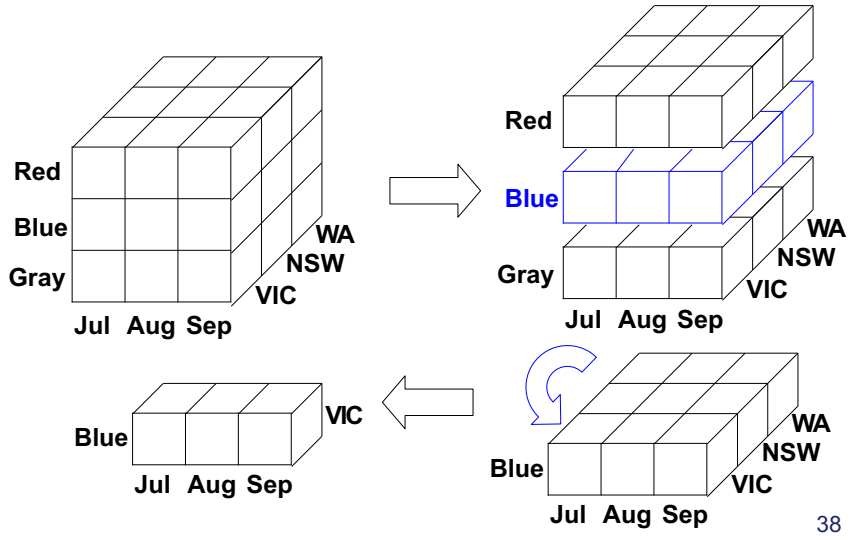
Autos Sold

	VIC	NSW	WA	Total
Red	40	29	40	109
Blue	45	31	37	113
Gray	48	40	35	123
Total	133	100	112	345

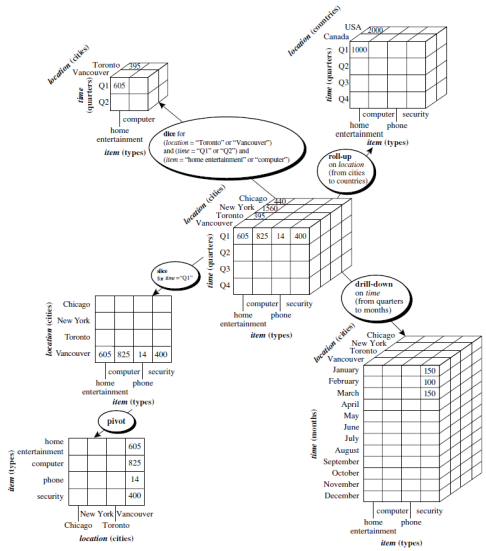
Slice and Dice

- **Slice** is to pick a rectangular subset of a cube by choosing a single value for one of its dimensions, creating a new cube with one fewer dimension.
- **Dice:** The dice operation produces a subcube by allowing the analyst to pick specific values of multiple dimensions.

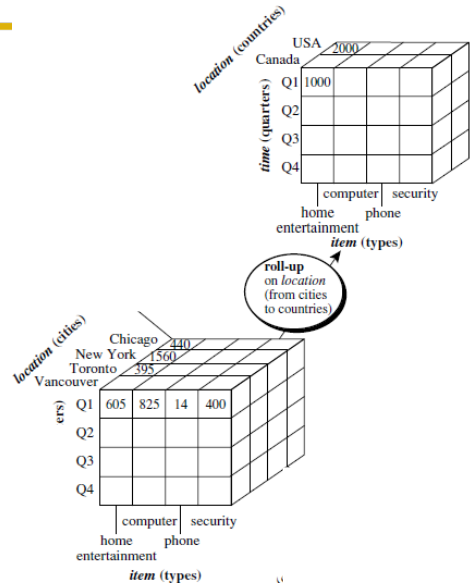
Slicing and Dicing



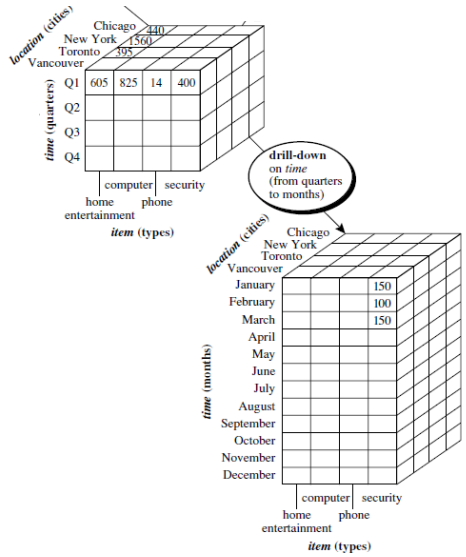
Example of OLAP Operations



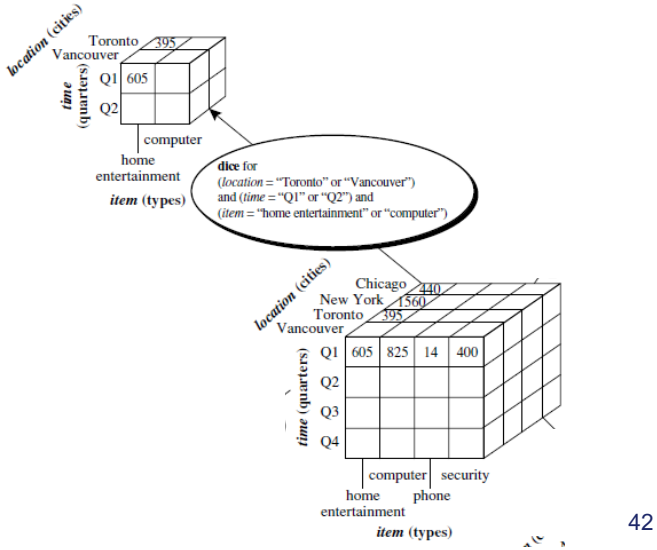
Example of OLAP Operations (roll-up)



Example of OLAP Operations (drill-down)

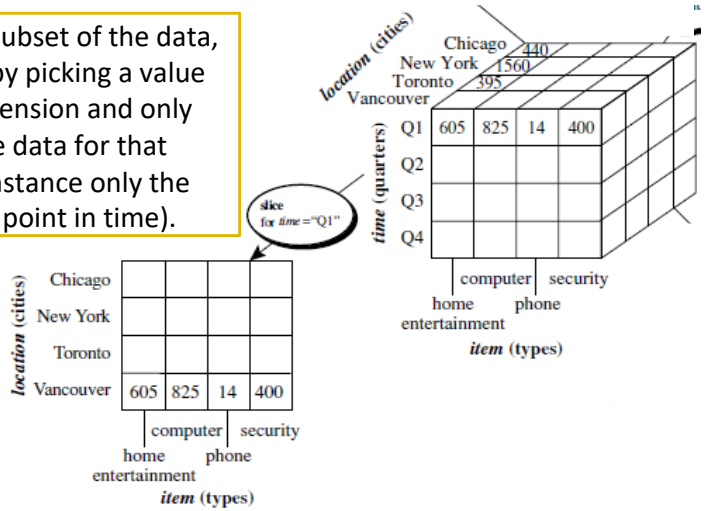


Example of OLAP Operations (dice)



Example of OLAP Operations (slice)

A Slice is a subset of the data, generated by picking a value for one dimension and only showing the data for that value (for instance only the data at one point in time).



Drill-Across Example

Question: How did actual sales diverge from forecasted sales in Sep 19?
Drill-across between "Forecast" and "Sales"

- Step 1: Query Forecast fact
  - Group by Brand Name, District Name
  - Filter on MonthAndYear ='Sep 19'
  - Calculate SUM(ForecastAmt)
  - Query result has schema (Brand Name, District Name, ForecastAmt)
- Step 2: Query Sales fact
  - Group by Brand Name, District Name
  - Filter on MonthAndYear ='Sept 19'
  - Calculate SUM(TotalSalesAmt)
  - Query result has schema (Brand Name, District Name, TotalSalesAmt)
- Step 3: Combine query results
  - Join Result 1 and Result 2 on Brand Name and District Name
  - Result has schema (Brand Name, District Name, ForecastAmt, TotalSalesAmt)

References

- Some slides are adapted from
  - http://web.stanford.edu/class/cs345/
  - https://hanj.cs.illinois.edu/bk3/bk3_slidesindex.htm
- Readings
  - Chapter 4.2 of Han et al.'s book
  - Chapter 5 of Rainardi's book
  - Drill down v.s. drill through

Copyright Notice