# Data Mining:
# Concepts and Techniques

# Chapter 3: Data Warehousing and OLAP Technology: An Overview

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining

# What is Data Warehouse?

- Defined in many different ways, but not rigorously.
  - A decision support database that is maintained separately from the organization's operational database
  - Support information processing by providing a solid platform of consolidated, historical data for analysis.
- "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

- Organized 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
- Provide a simple and concise view around particular subject issues by excluding data that are 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.
  - Ensure consistency in naming conventions, encoding structures, attribute measures, etc. among different data sources
    - E.g., Hotel price: currency, tax, breakfast covered, etc.
  - 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 "time element"

# Data Warehouse—Nonvolatile

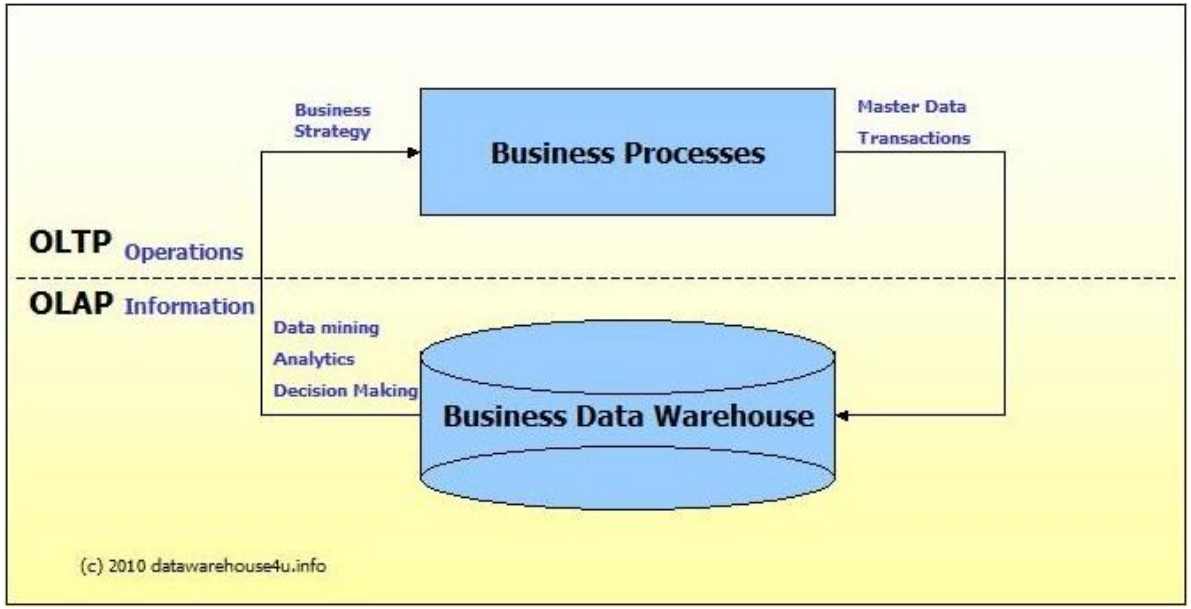
- A physically separate store of data transformed from the operational environment
- Operational update of data does not occur in the data warehouse environment
  - Does not require transaction processing, recovery, and concurrency control mechanisms
  - Requires only two operations in data accessing:
    - *initial loading of data* and *access of data*

# Data Warehouse vs. Operational DBMS

- OLTP (on-line transaction processing)
  - Major task of traditional relational DBMS
  - Day-to-day operations: purchasing, inventory, banking, manufacturing, payroll, registration, accounting, etc.
- OLAP (on-line analytical processing)
  - Major task of data warehouse system
  - Data analysis and decision making
- Distinct features (OLTP vs. OLAP):
  - User and system orientation: customer vs. market
  - Data contents: current, detailed vs. historical, consolidated
  - Database design: ER + application vs. star + subject
  - View: current, local vs. evolutionary, integrated
  - Access patterns: update vs. read-only but complex queries

# OLTP VS OLAP

- We can divide IT systems into transactional (OLTP) and analytical (OLAP). In general we can assume that OLTP systems provide source data to data warehouses, whereas OLAP systems help to analyze it.



# OLTP vs. OLAP

| | **OLTP** | **OLAP** |
|---|---|---|
| **users** | clerk, IT professional | knowledge worker |
| **function** | day to day operations | decision support |
| **DB design** | application-oriented | subject-oriented |
| **data** | current, up-to-date<br>detailed, flat relational<br>isolated | historical,<br>summarized, multidimensional<br>integrated, consolidated |
| **usage** | repetitive | ad-hoc |
| **access** | read/write<br>index/hash on prim. key | lots of scans |
| **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 |

# 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

# TWO-D DATA CUBE MODEL

A standard spreadsheet is a 2-d matrix.

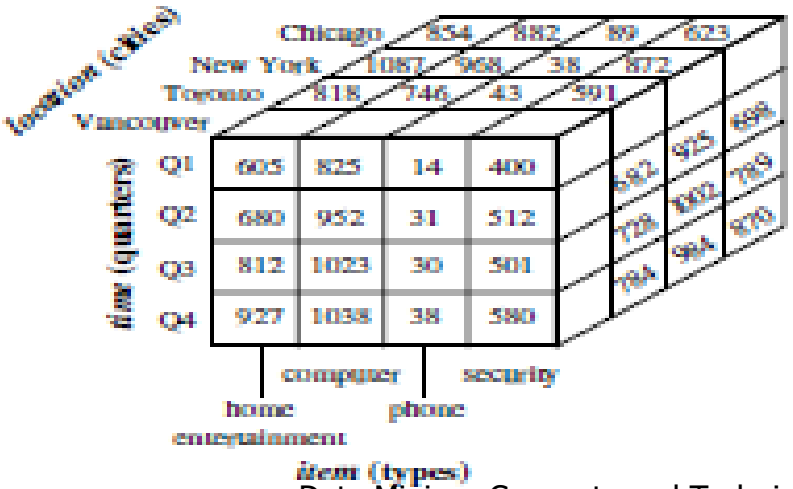
A 2-D view of sales data for *AllElectronics* according to the dimensions *time* and *item*, where the sales are from branches located in the city of Vancouver. The measure displayed is *dollars_sold* (in thousands).

| | **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 |

# THREE-D DATA CUBE MODEL

**Table 3.3** A 3-D view of sales data for *AllElectronics*, according to the dimensions *time*, *item*, and *location*. The measure displayed is *dollars_sold* (in thousands).

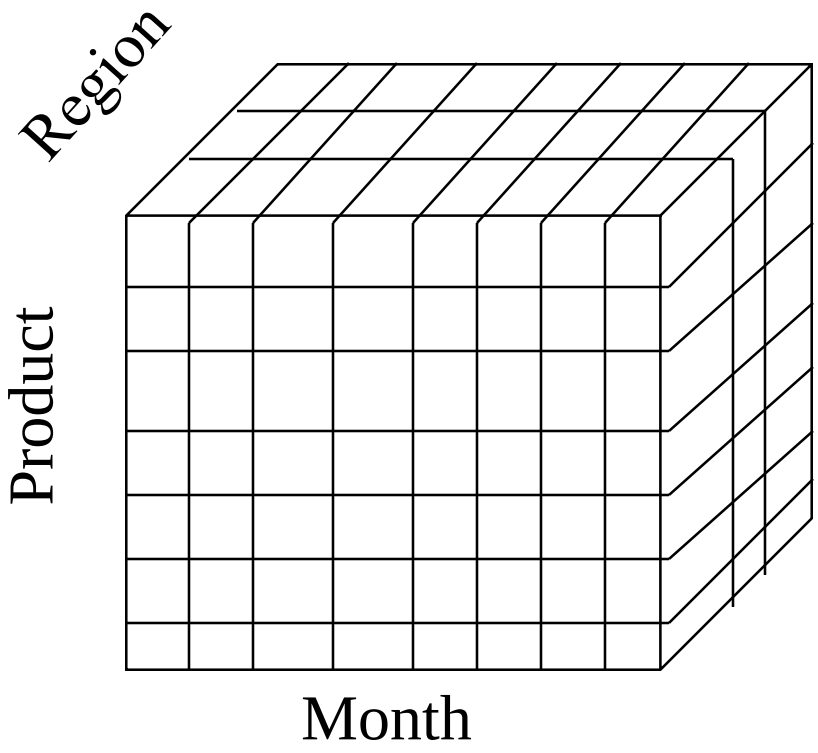
| | location = "Chicago" | | | | location = "New York" | | | | location = "Toronto" | | | | location = "Vancouver" | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | item | | | | item | | | | item | | | | item | | | |
| | home | | | | home | | | | home | | | | home | | | |
| 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 |



# Multidimensional Data

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

**Dimensions: Product, Location, Time**



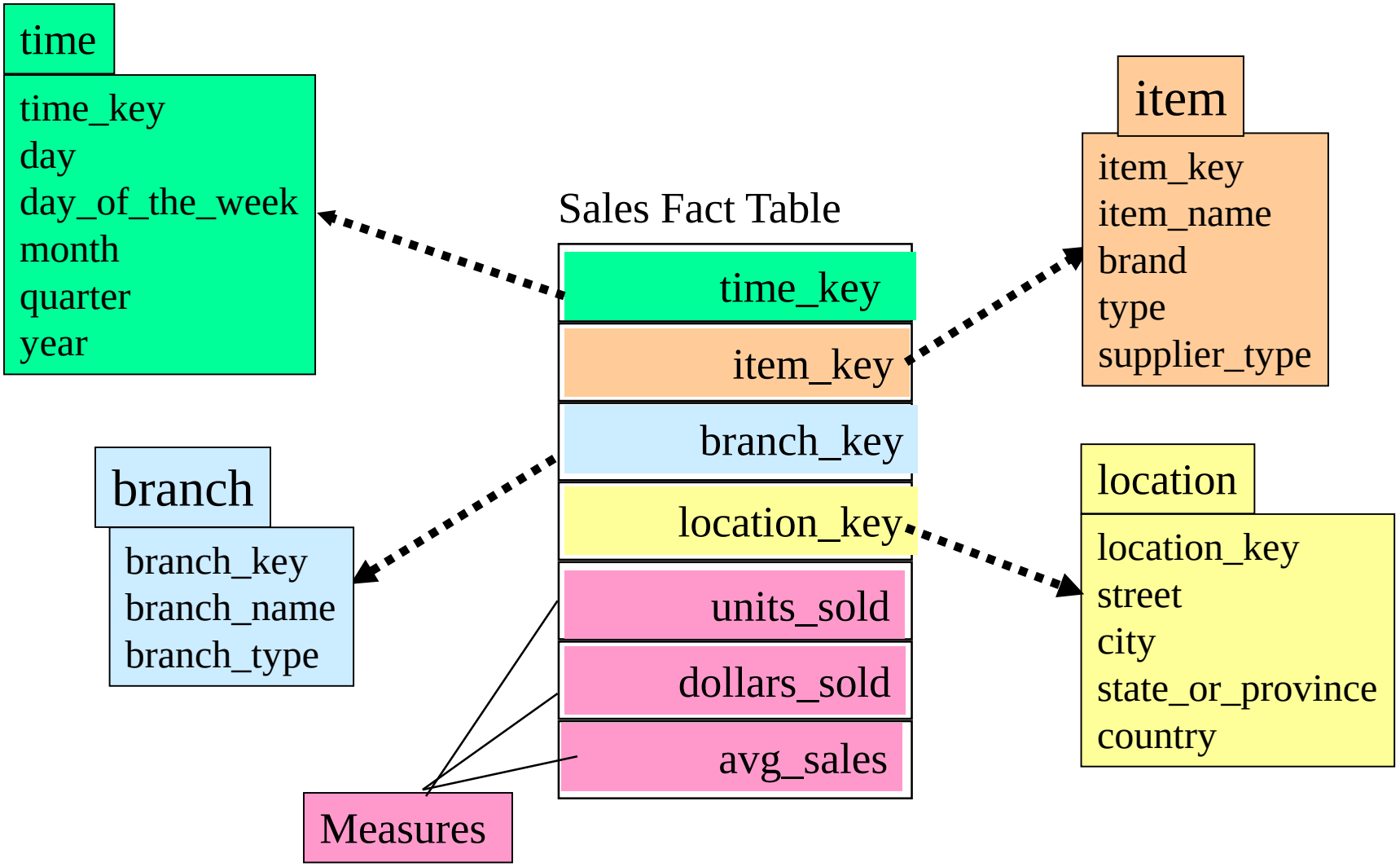
# Two types of tables

- The multidimensional storage model involves two types of tables:
  - Dimension tables,  consist of tuples of attributes 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

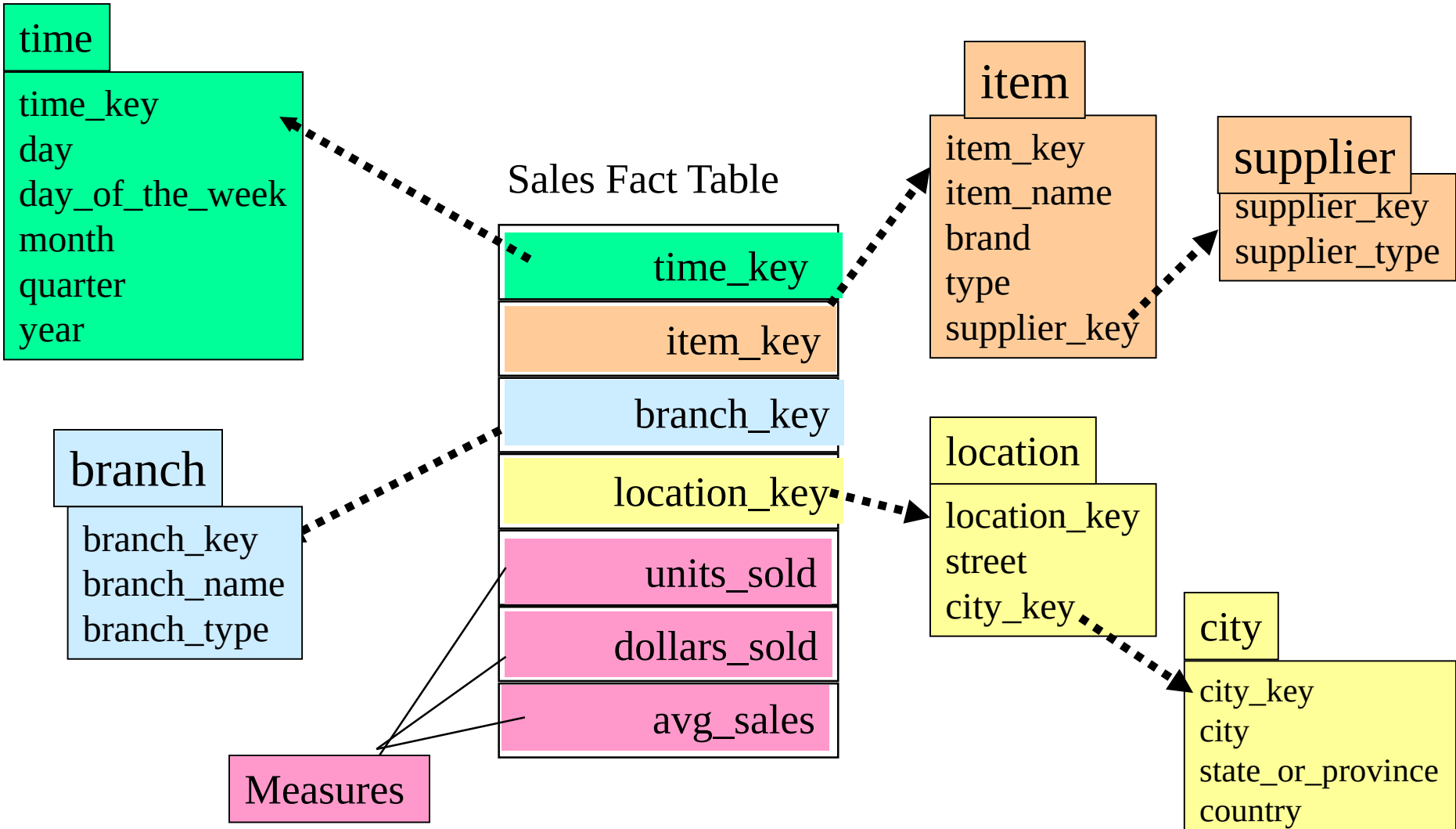
# Conceptual Modeling of Data Warehouses/ Data Modeling for Data Warehouses

- 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

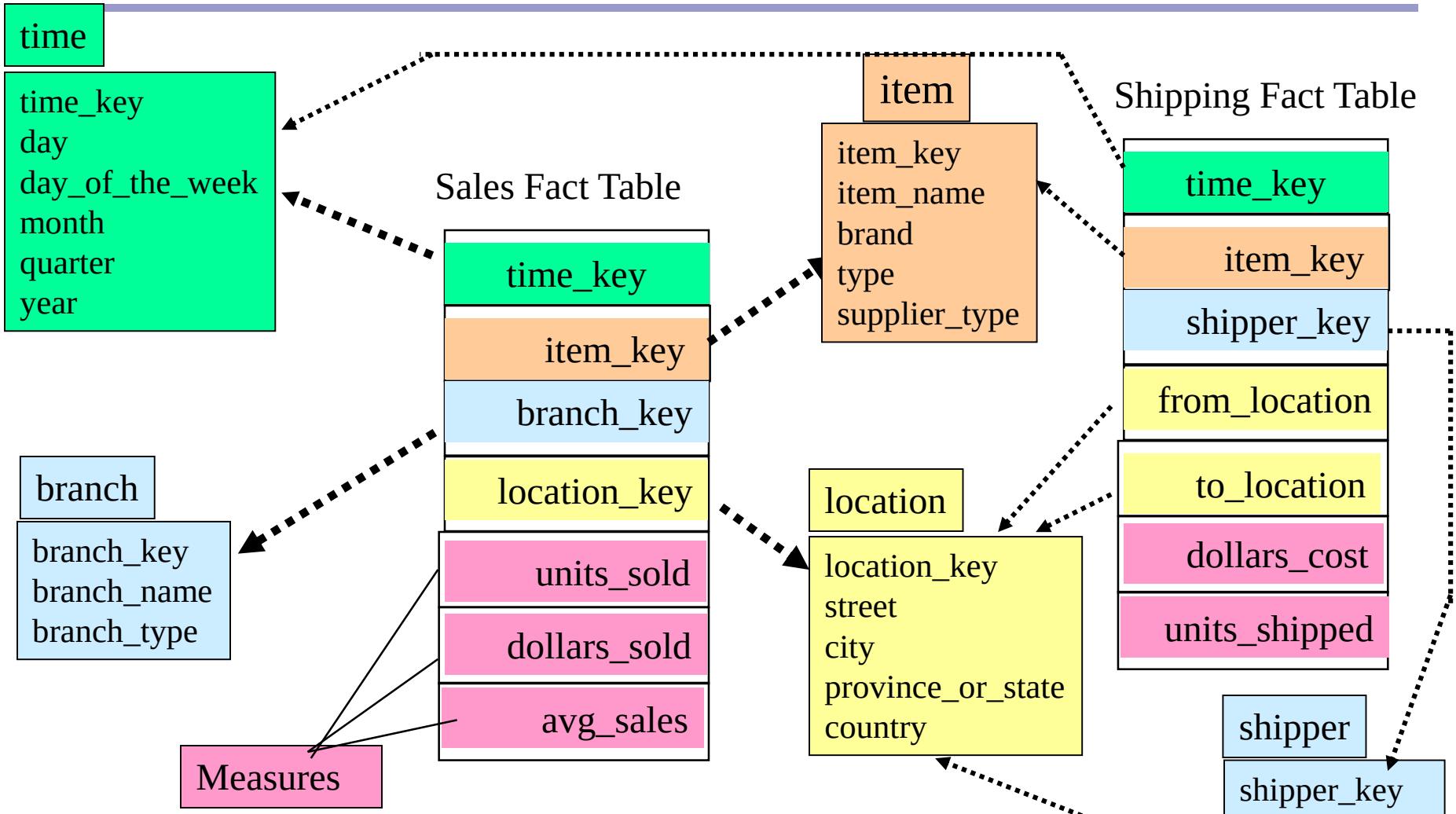
# Example of Star Schema



# Example of Snowflake Schema



# Example of Fact Constellation



# Typical OLAP Operations/ Typical functionality of data warehouse

- Roll up (drill-up): summarize data
  - *by climbing up hierarchy or by dimension reduction*
  - *Data is summarized with increasing generalization.*
- 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* operations are performed
- Pivot (rotate):
  - *reorient the cube, visualization, 3D to series of 2D planes*

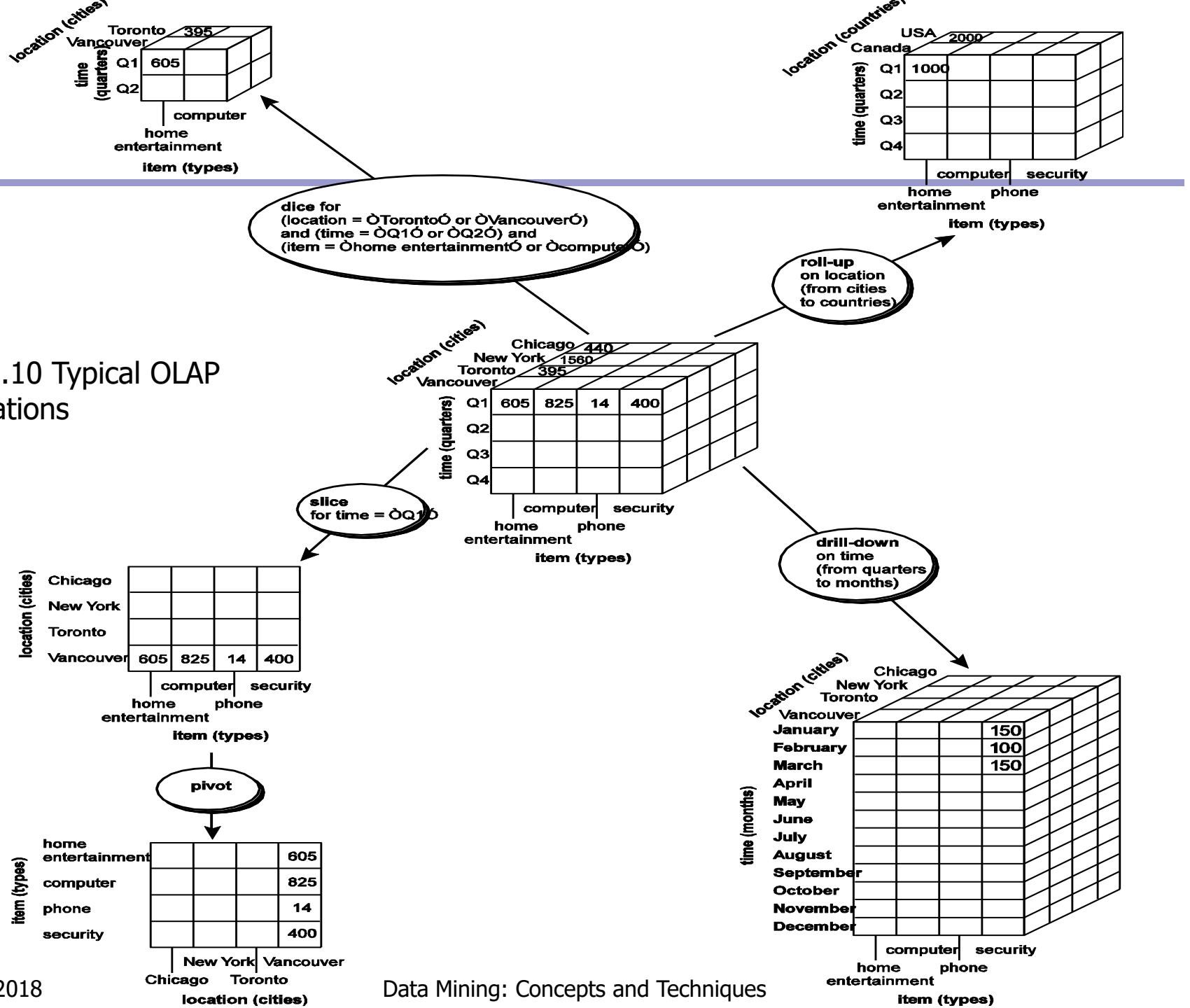



Fig. 3.10 Typical OLAP Operations

# ROLAP AND MOLAP

- As data warehouse are free from the restrictions of the transactional environment, there is an increased efficiency in query processing.
- Among the tools and techniques used are query transformation, index intersection and union , special ROLAP and MOLAP functions are used.

# MOLAP

- Multidimensional OLAP (MOLAP) uses array-based multidimensional storage engines for multidimensional views of data. With multidimensional data stores, the storage utilization may be low if the data set is sparse. Therefore, many MOLAP servers use two levels of data storage representation to handle dense and sparse data-sets
- This is the more traditional way of OLAP analysis. In MOLAP, data is stored in a multidimensional cube. The storage is not in the relational database, but in proprietary formats.
- Advantages:
  - **Excellent performance:** MOLAP cubes are built for fast data retrieval, and are optimal for slicing and dicing operations.
  - **Can perform complex calculations:** All calculations have been pre-generated when the cube is created. Hence, complex calculations are not only doable, but they return quickly.

- **Disadvantages:**
- **Limited in the amount of data it can handle:** Because all calculations are performed when the cube is built, it is not possible to include a large amount of data in the cube itself. This is not to say that the data in the cube cannot be derived from a large amount of data. Indeed, this is possible. But in this case, only summary-level information will be included in the cube itself.
- **Requires additional investment:** Cube technology are often proprietary and do not already exist in the organization. Therefore, to adopt MOLAP technology, chances are additional investments in human and capital resources are needed.

# ROLAP

- This methodology relies on manipulating the data stored in the relational database to give the appearance of traditional OLAP's slicing and dicing functionality. In essence, each action of slicing and dicing is equivalent to adding a "WHERE" clause in the SQL statement.
- **Advantages:**
  - **Can handle large amounts of data:** The data size limitation of ROLAP technology is the limitation on data size of the underlying relational database. In other words, ROLAP itself places no limitation on data amount.
  - **Can leverage functionalities inherent in the relational database:** Often, relational database already comes with a host of functionalities. ROLAP technologies, since they sit on top of the relational database, can therefore leverage these functionalities.

- **Disadvantages:**
  - **Performance can be slow:** Because each ROLAP report is essentially a SQL query (or multiple SQL queries) in the relational database, the query time can be long if the underlying data size is large.
  - **Limited by SQL functionalities:** Because ROLAP technology mainly relies on generating SQL statements to query the relational database, and SQL statements do not fit all needs (for example, it is difficult to perform complex calculations using SQL), ROLAP technologies are therefore traditionally limited by what SQL can do. ROLAP vendors have mitigated this risk by building into the tool out-of-the-box complex functions as well as the ability to allow users to define their own functions.

# MOLAP vs ROLAP

| Sr.No. | MOLAP | ROLAP |
|---|---|---|
| 1 | Information retrieval is fast. | Information retrieval is comparatively slow. |
| 2 | Uses sparse array to store data-sets. | Uses relational table. |
| 3 | MOLAP is best suited for inexperienced users, since it is very easy to use. | ROLAP is best suited for experienced users. |
| 4 | Maintains a separate database for data cubes. | It may not require space other than available in the Data warehouse. |
| 5 | DBMS facility is weak. | DBMS facility is strong. |

# SNOWFLAKE VS STARFLAKE

| | Snowflake Schema | Star Schema |
|---|---|---|
| **Ease of maintenance / change** | No redundancy, so snowflake schemas are easier to maintain and change. | Has redundant data and hence less easy to maintain/change |
| **Ease of Use** | More complex queries and hence less easy to understand | Lower query complexity and easy to understand |
| **Query Performance** | More foreign keys and hence longer query execution time (slower) | Less number of foreign keys and hence shorter query execution time (faster) |
| **Type of Datawarehouse** | Good to use for datawarehouse core to simplify complex relationships (many:many) | Good for datamarts with simple relationships (1:1 or 1:many) |
| **Joins** | Higher number of Joins | Fewer Joins |
| **Dimension table** | A snowflake schema may have more than one dimension table for each dimension. | A star schema contains only single dimension table for each dimension. |
| **When to use** | When dimension table is relatively big in size, snowflaking is better as it reduces space. | When dimension table contains less number of rows, we can choose Star schema. |
| **Normalization/ De-Normalization** | Dimension Tables are in Normalized form but Fact Table is in De-Normalized form | Both Dimension and Fact Tables are in De-Normalized form |
| **Data model** | Bottom up approach | Top down approach |

# DATAWAREHOUSE VS VIEWS

| DATAWAREHOUSE | VIEWS |
|---|---|
| •It exists as persistent storage | • it is materialised on demand |
| • they are not usually relational but rather multidimensional | • views of relational database are relational |
| • it can be indexed to optimize performance | •Can not be indexed |
| • it characteristically provide specific support of functionality | •Views can not |
| • it provides large amount of integrated and temporal data | • views are extract of database. |

# DATAWAREHOUSE VS DATABASE

The major differences between the **Databases** and **Data Warehouses** are as follows:-

| FEATURES | DATABASE | DATA WAREHOUSE |
|---|---|---|
| **Characteristic** | It is based on Operational Processing. | It is based on Informational Processing. |
| **Data** | It mainly stores the Current data which always guaranteed to be up-to-date. | It usually stores the Historical data whose accuracy is maintained over time. |
| **Function** | It is used for day-to-day operations. | It is used for long-term informational requirements and decision support. |

# DATAWAREHOUSE VS DATABASE

| | | |
|---|---|---|
| **User** | The common users are clerk, DBA, database professional. | The common users are knowledge worker (e.g., manager, executive, analyst) |
| **Unit of work** | Its work consists of short and simple transaction. | The operations on it consists of complex queries.. |
| **Focus** | The focus is on "Data IN" | The focus is on "Information OUT" |
| **Orientation** | The orientation is on Transaction. | The orientation is on Analysis. |
| **DB design** | The designing of database is ER based and application-oriented. | The designing is done using star/snowflake schema and its subject-oriented. |
| **Summarization** | The data is primitive and highly detailed. | The data is summarized and in consolidated form. |

# DATAWAREHOUSE VS DATABASE

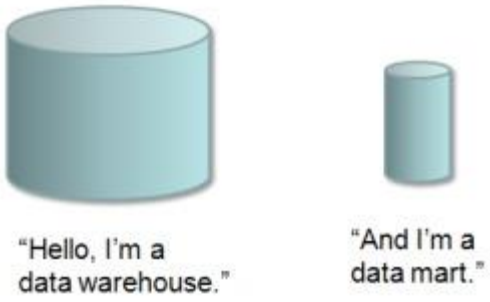
| | | |
|---|---|---|
| **View** | The view of the data is flat relational. | The view of the data is multidimensional. |
| **Access** | The most frequent type of access type is read/write. | It mostly use the read access for the stored data. |
| **Operations** | The main operation is index/hash on primary key. | For any operation it needs a lot of scans. |
| **Number of records accessed** | A few tens of records. | A bunch of millions of records. |
| **Number of users** | In order of thousands. | In the order of hundreds only. |
| **DB size** | 100 MB to GB. | 100 GB to TB. |
| **Priority** | High performance, high availability | High flexibility, end-user autonomy |

# DATAWAREHOUSE VS DATABASE

| | | |
|---|---|---|
| **Metric** | To measure the efficiency, transaction throughput is measured. | To measure the efficiency, query throughput and response time is measured. |

# DATAWAREHOUSE VS DATAMART

- **Data Warehouse:**
  - Holds multiple subject areas
  - Holds very detailed information
  - Works to integrate all data sources
  - Does not necessarily use a dimensional model but feeds dimensional models.
- **Data Mart**
  - Often holds only one subject area- for example, Finance, or Sales
  - May hold more summarised data (although many hold full detail)
  - Concentrates on integrating information from a given subject area or set of source systems
  - Is built focused on a dimensional model using a star schema.



# EXAMPLE

QUES:

Suppose that a data warehouse consists of the three dimensions *time, doctor, and patient, and the two* measures *count and charge, where charge is the fee that a doctor charges a patient for a visit.*

(a) Enumerate three classes of schemas that are popularly used for modeling data warehouses.

(b) Draw a schema diagram for the above data warehouse using one of the schema classes listed in (a).

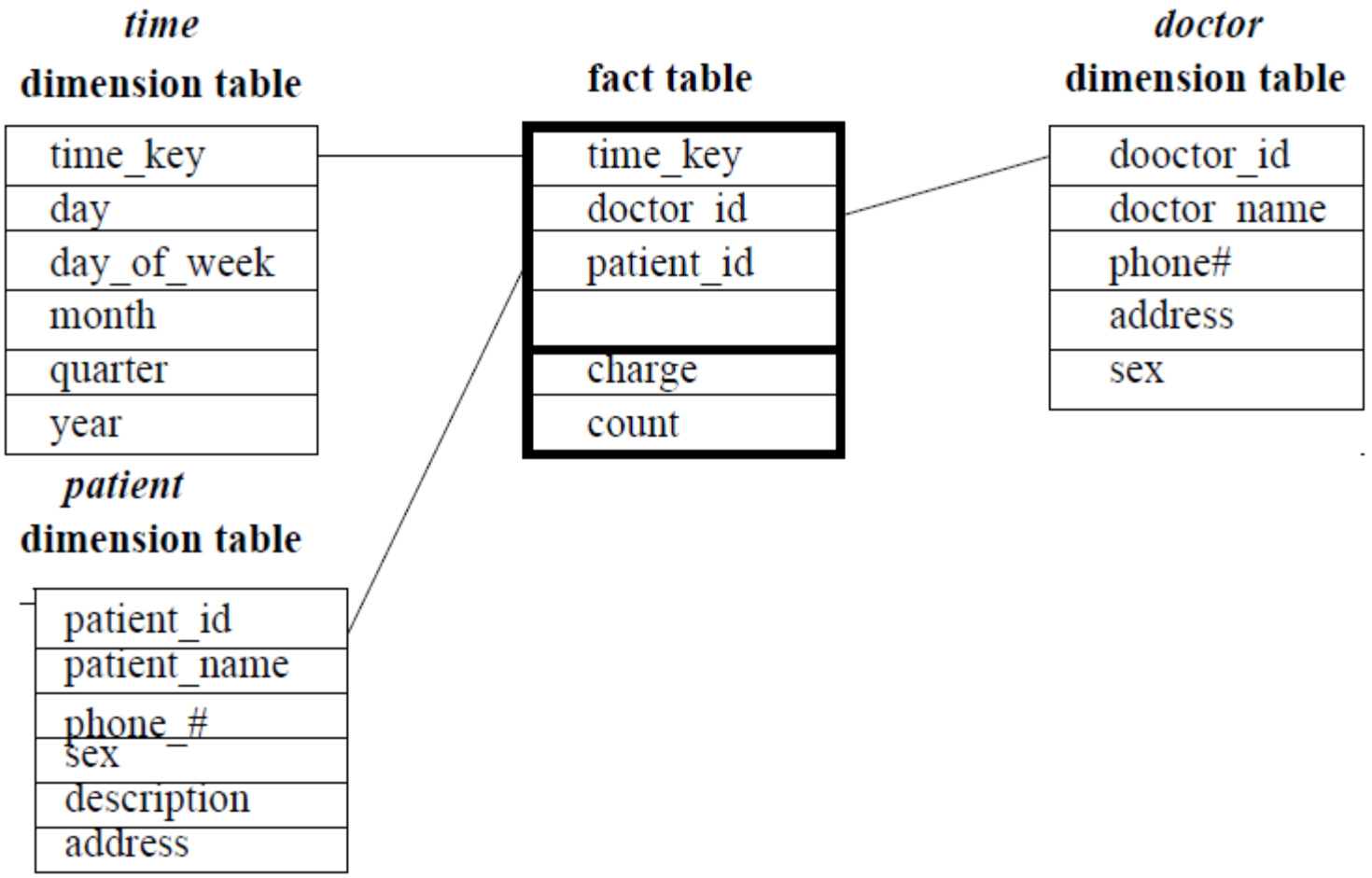
ANS:

(A) Enumerate three classes of schemas that are popularly used for modeling data warehouses.

Three classes of schemas popularly used for modeling data warehouses are the star schema, the snowflake schema, and the fact constellations schema.

# star schema for data warehouse

(b)

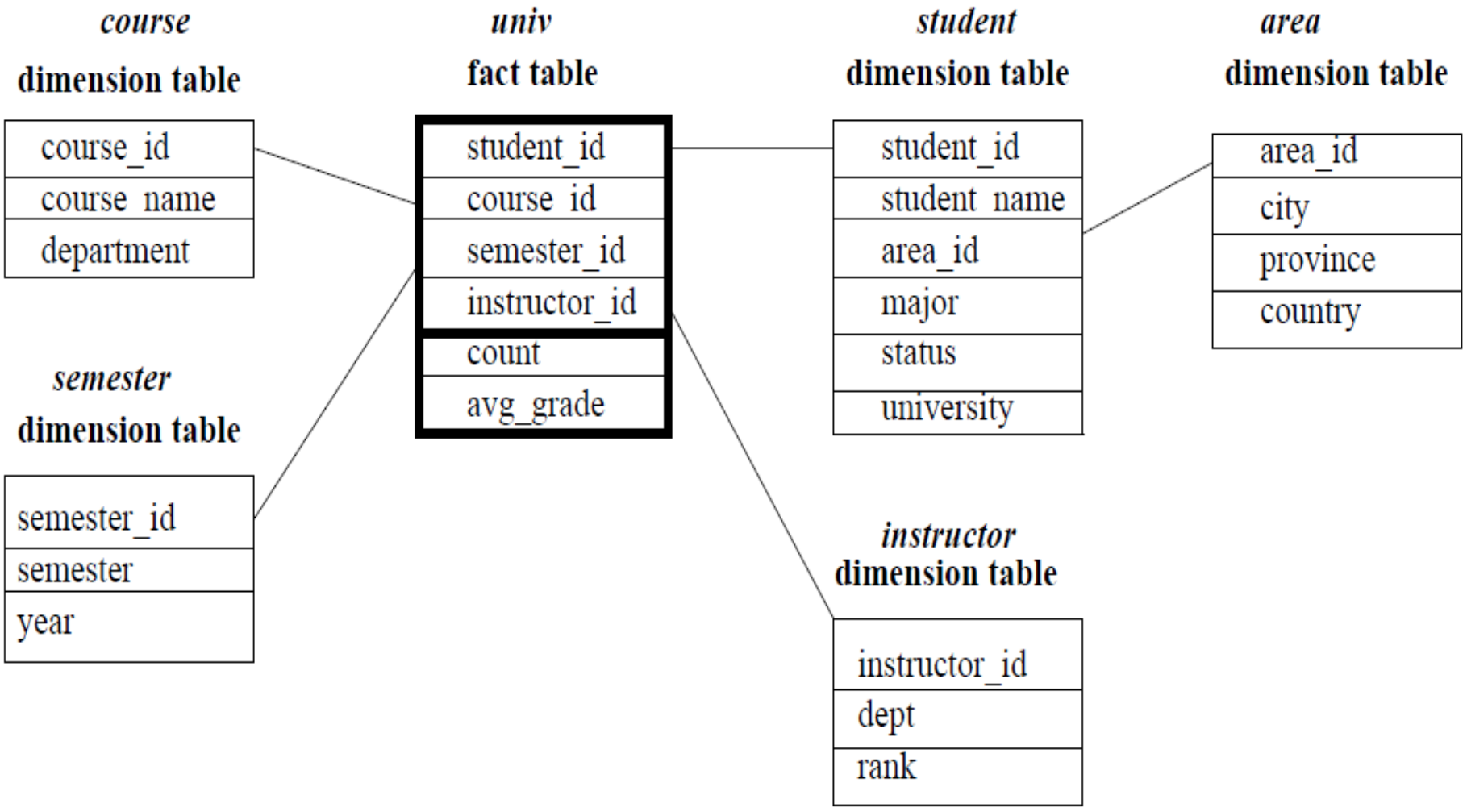


# EXAMPLE

## QUES:

- Suppose that a data warehouse for *Big University consists of the following four dimensions: student, course,semester, and instructor, and two measures count and avg grade. When at the lowest conceptual level (e.g.*,for a given student, course, semester, and instructor combination), the *avg grade measure stores the actual* course grade of the student. At higher conceptual levels, *avg grade stores the average grade for the given* combination.
- (a) Draw a *snowflake schema diagram for the data warehouse.*
- (b) Starting with the base cuboid [*student; course; semester; instructor], what specific OLAP operations* (e.g., roll-up from *semester to year) should one perform in order to list the average grade of CS courses* for each *Big University student.*

# Snow flake schema



# ANS (b)

- Starting with the *base cuboid [student; course; semester; instructor], what specific OLAP operations*
- (e.g., roll-up from *semester to year) should one perform in order to list the average grade of CS courses* for each *Big University student.*
- The specific OLAP operations to be performed are:
  - *Roll-up on course from course id to department.*
  - *Roll-up on student from student id to university.*
- *Dice on course, student with department=\CS" and university = \Big University".*
  - *Drill-down on student from university to student name.*