Databases and Information Systems CS303

Data Warehousing and Mining
03-11-2023

Introduction

- Two main aspects:
  - Gather data from multiple sources into a central repository, called a data warehouse.
    - How to deal with dirty data: data with some errors
    - Techniques for efficient storage and indexing of large volumes of data
  - Analyze the gathered data to find information called data mining
    - Aims at detecting various types of patterns in large volumes of data
    - Supplements various types of statistical techniques with similar goals

Decision Support Systems

- Database applications can be broadly classified into transaction-processing and decision-support systems
  - Transaction-processing systems are systems that record information about transactions
    - Example: sales information for companies, course registration and grade information for universities
  - Decision-support systems aim to get high-level information out of the detailed information stored in transaction-processing systems
    - To use the high-level information to make a variety of decisions
    - Example : Decide what products to stock in a shop, what products to manufacture in a factory, or which of the applicants should be admitted to a university.

Decision Support Systems

- Large databases can have lot of useful information for making business decisions

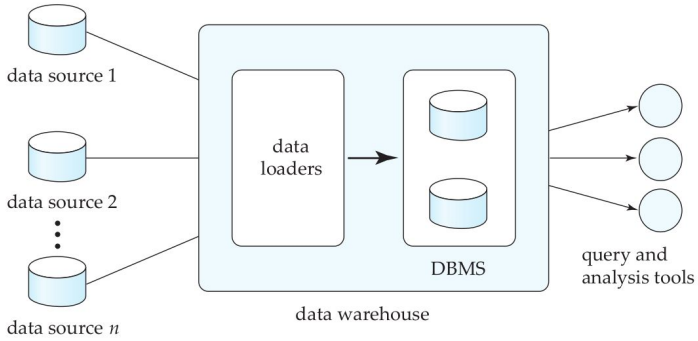
- Companies can identify patterns in customer behavior and use the patterns to make business decisions

Data Warehousing

- A data warehouse is a repository (or archive) of information gathered from multiple sources, stored under a unified schema, at a single site.
- Once gathered, the data are stored for a long time, permitting access to historical data.
- Data warehouses provide a single consolidated interface to data,
  - Easier to write decision-support queries
- The decision maker ensures that online transaction-processing systems are not affected by the decision-support workload

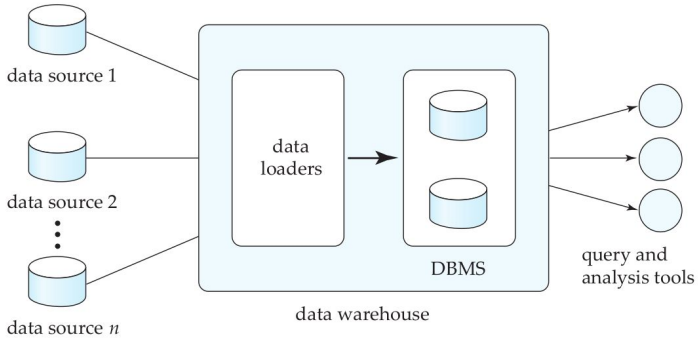
Components of Data Warehousing

- When and How to gather data
  - Source-driven architecture:
    - For gathering data, the data sources transmit new information
    - Continually or periodically
  - Destination-driven architecture:
    - Data warehouse periodically sends requests for new data to the sources
  - To have same copy of data : Updates at the sources should be replicated at the warehouse via two-phase commit
    - But very expensive
  - Warehouses typically have slightly out-of-date data
  - Usually not a problem for decision-support systems.



Components of Data Warehousing

- What Schema to Use
  - Data sources that have been constructed independently are likely to have different schemas.
  - May even use different data models.
  - Task of a warehouse is to perform schema integration and convert data to the integrated schema before they are stored.
  - Data stored in the warehouse are not just a copy of the data at the sources.
    - They can be thought of as a materialized view of the data at the sources.



Components of Data Warehousing

- Data transformation and cleansing
  - Data sources often deliver data with numerous minor inconsistencies, which can be corrected.
    - Names are often misspelled
    - Addresses may one or more typos
    - Postal codes entered incorrectly
  - These can be corrected to a reasonable extent by consulting a corresponding database
  - The approximate matching of data required for this task is referred to as fuzzy lookup.
  - Address lists collected from multiple sources may have duplicates that need to be eliminated in a merge–purge operation (deduplication)
  - Records for multiple individuals in a house may be grouped together so only one mailing is sent to each house
    - this operation is called householding.
  - Data may be transformed in ways other than cleansing
    - Changing the units of measure
    - Converting the data to a different schema by joining data from multiple source relations

Components of Data Warehousing

- How to propagate updates:
  - Updates on relations at the data sources must be propagated to the data warehouse
  - If the relations at the data warehouse are exactly the same as those at the data source, the propagation is straightforward
  - If they are not, the problem of propagating updates is similar to view-maintenance problem

Components of Data Warehousing

- What data to summarize:
  - The raw data generated by a transaction-processing system may be too large to store online
  - We can answer many queries by maintaining just summary data
    - obtained by aggregation on a relation
  - Suppose that a relation r has been replaced by a summary relation s,
    - Users may still be permitted to pose queries to relation r
    - If the query requires only summary data, it may be possible to transform it into an equivalent one using s instead

- The different steps involved in getting data into a data warehouse are called extract, transform, and load (ETL tasks)
  - extraction refers to getting data from the sources,
  - load refers to loading the data into the data warehouse.

Warehouse Schemas

- The data are usually multidimensional, with dimension attributes and measure attributes

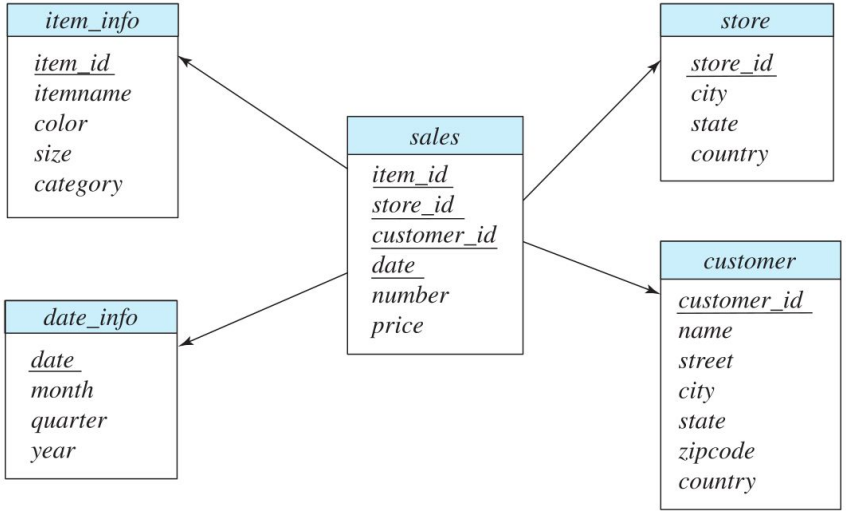
- Tables containing multidimensional data are called fact tables and are usually very large.
  - Example : A table recording sales information for a retail store, with one tuple for each item that is sold

- The dimensions of the sales table would include what the item is
  - Item identifier such as that used in barcodes
  - Date when the item is sold
  - Which location (store) the item was sold from
  - Which customer bought the item

- The measure attributes may include the number of items sold and the price of the items

Warehouse Schemas

- Dimension attributes are usually short identifiers that are foreign keys into other tables called dimension tables
- Such schemas are called Star Schema
- Complex databases can be modelled with Snowflake schema
  - Outer entity types of stars are further the center of star schema
- Data Warehouses often have more than one fact table



Column Oriented Storage

- Databases traditionally store all attributes of a tuple together, and tuples are stored sequentially in a file.
  - Row-oriented storage
- In column-oriented storage, each attribute of a relation is stored in a separate file
  - values from successive tuples stored at successive positions in the file.
- Assuming fixed-size data types, the value of attribute A of the i-th tuple of a relation can be found
  - by accessing the file corresponding to attribute A
  - reading the value at offset (i − 1) times the size (in bytes) of values in attribute A.

Column Oriented Storage

- Benefits:
  - When a query needs to access only a few attributes of a relation with a large number of attributes
    - the remaining attributes need not be fetched from disk into memory
  - Storing values of the same type together increases the effectiveness of compression
    - compression can greatly reduce both the disk storage cost and the time to retrieve data from disk
- Drawbacks:
  - storing or fetching a single tuple requires multiple I/O operations
- Column-oriented storage is not widely used for transaction-processing applications
- Suited for data-warehousing applications
  - where accesses are rarely to individual tuples, but rather require scanning and aggregating multiple tuples.

Data Mining

- Process of semi-automatically analyzing large databases to find useful patterns
- Some types of knowledge discovered from a database can be represented by a set of rules
- Example:
  - Young women with annual incomes greater than 1,00,000 are the most likely people to buy sports cars
- Not universally true,
  - Have degrees of “support” and “confidence”
- Knowledge also represented by equations relating different variables to each other
- Predicting outcomes when the values of some variables are known

Data Mining

- Manual component to data mining consists of
  - Preprocessing data to a form acceptable to the algorithms
  - Postprocessing of discovered patterns to find novel ones that could be useful.

- There may be more than one type of pattern that can be discovered from a given database
  - Manual interaction may be needed to pick useful types of patterns

Data Mining

- Uses of Knowledge discovery:
  - Predictions:
    - Example: when a person applies for a credit card, the credit-card company wants to predict if the person is a good credit risk.
      - The prediction is to be based on known attributes of the person, such as age, income, debts, and past debt-repayment history.
    - Rules for making the prediction are derived from the same attributes of past and current credit-card holders
      - along with their observed behavior
  - Look for associations,
    - Example: books that tend to be bought together.
    - Associations may lead to discovery of causation.
      - Discovery of unexpected associations between a newly introduced medicine and cardiac problems led to the finding that the medicine may cause cardiac problems in some people

Classification

- Classification problem:
  - Given that items belong to one of several classes, and given past instances (called training instances) of items along with the classes to which they belong
    - predict the class to which a new item belongs.

- The class of the new instance is not known, so other attributes of the instance must be used to predict the class

- Done by finding rules that partition the given data into disjoint groups

Classification: Example

- Suppose a credit-card company wants to decide whether or not to give a credit card to an applicant
- The company assigns a credit-worthiness level of
  - excellent, good, average, or bad
- Find rules that classify current customers into excellent, good, average, or bad
  - On the basis of the information about the person
- Consider two attributes:
  - education level (highest degree earned) and income
- The rules may be of the following form:
  - $\forall$ person P, P .degree = masters and P .income > 75, 000 $\Rightarrow$ P .credit = excellent
  - $\forall$ person P, P .degree = bachelors or (P .income $\geq$ 25, 000 and P .income $\leq$ 75, 000) $\Rightarrow$ P .credit = good

Classification: Example

- The process of building a classifier starts from a sample of data
  - Training set
- For each tuple in the training set, the class to which the tuple belongs is already known.
- Several ways of building a classifier
  - Decision-Tree Classifiers
  - Bayesian classifiers
  - Support Vector Machine

Decision Tree Classifiers

- Uses a tree
  - each leaf node has an associated class
  - each internal node has a predicate

- To classify a new instance:
  - start at the root
  - traverse the tree to reach a leaf
  - at an internal node evaluate the predicate (or function) on the data instance, to find which child to go to