Unit - 7

Advanced Topics

Concept of Object-Oriented Database

- Extend the relational data model by including object orientation and constructs to deal with added data types.
- Allow attributes of tuples to have complex types, including non-atomic values such as nested relations.
- Preserve relational foundations, in particular the declarative access to data, while extending modeling power.
- Permit non-atomic domains (atomic ≡ indivisible)
- Example of non-atomic domain: set of integers, or set of tuples
- Allows more intuitive modeling for applications with complex data

Concept of Object-Oriented Database

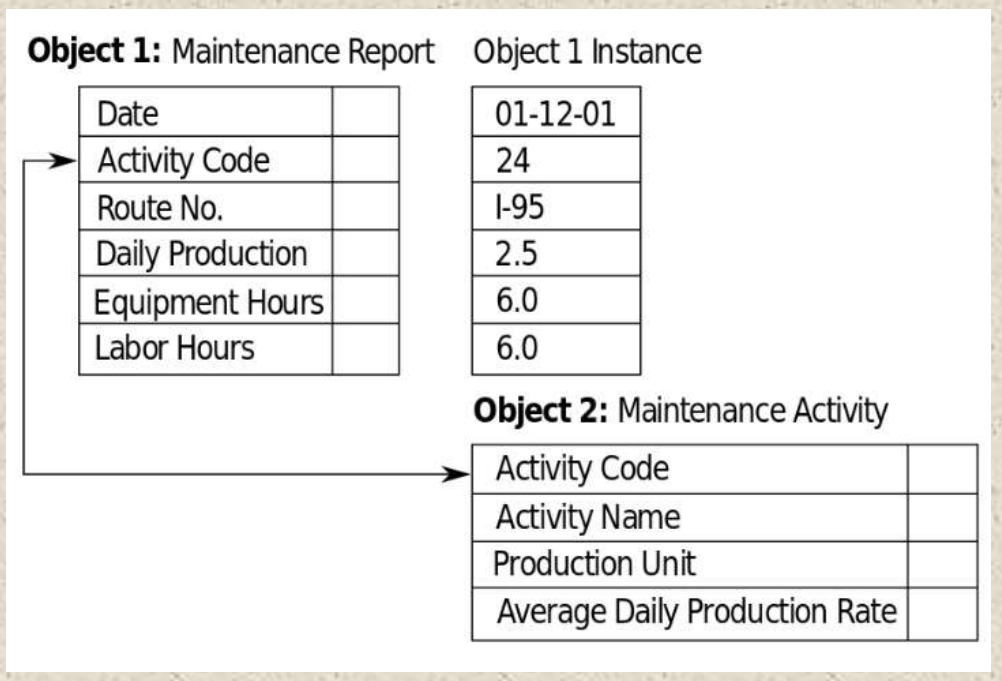
- Loosely speaking, an object corresponds to an entity in the ER model.
- The object-oriented paradigm is based on encapsulating code/method and data related to an object into single unit.
- The object-oriented data model is a logical data model (like the E-R model)

Concept of Object-Oriented Database

- A core object-oriented data model consists of the following basic object-oriented concepts:
  - Object and object identifier: Any real world entity is uniformly modeled as an object (associated with a unique id: used to pinpoint an object to retrieve).
  - Attributes and methods: every object has a state (the set of values for the attributes of the object) and a behavior (the set of methods - program code - which operate on the state of the object). The state and behavior encapsulated in an object are accessed or invoked from outside the object only through explicit message passing.
  - Class: a means of grouping all the objects which share the same set of attributes and methods. An object must belong to only one class as an instance of that class (instance-of relationship). A class is similar to an abstract data type. A class may also be primitive (no attributes), e.g., integer, string, Boolean.

Concept of Object-Oriented Database

- A core object-oriented data model consists of the following basic object-oriented concepts:
  - Class hierarchy and inheritance: derive a new class (subclass) from an existing class (superclass). The subclass inherits all the attributes and methods of the existing class and may have additional attributes and methods. single inheritance (class hierarchy) vs. multiple inheritance (class lattice).



Concept of Object-Oriented Database

- An object has associated with it:
  - A set of variables that contain the data for the object. The value of each variable is itself an object.
  - A set of messages to which the object responds; each message may have zero, one, or more parameters.
  - A set of methods, each of which is a body of code to implement a message; a method returns a value as the response to the message

```
CREATE TABLE cities (
    name            text,
    population      float,
    altitude        int     -- in feet
);

CREATE TABLE capitals (
    state           char(2)
) INHERITS (cities);
```

Database Security

- Database security is a broad area that addresses many issues, including the following:
  - **Various legal and ethical issues** regarding the right to access certain information—for example, some information may be deemed to be private and cannot be accessed legally by unauthorized organizations or persons.
  - **Policy issues** at the governmental, institutional, or corporate level regarding what kinds of information should not be made publicly available—for example, credit ratings and personal medical records.
  - **System-related issues** such as the system levels at which various security functions should be enforced—for example, whether a security function should be handled at the physical hardware level, the operating system level, or the DBMS level.
- The need in some organizations to identify multiple security levels and to categorize the data and users based on these classifications—for example, top secret, secret, confidential, and unclassified. The security policy of the organization with respect to permitting access to various classifications of data must be enforced.

Threats to Database

- Threats to databases can result in the loss or degradation of some or all of the following commonly accepted security goals: integrity, availability, and confidentiality.
  - **Loss of integrity:** Database integrity refers to the requirement that information be protected from improper modification. Modification of data includes creating, inserting, and updating data; changing the status of data; and deleting data. Integrity is lost if unauthorized changes are made to the data by either intentional or accidental acts. If the loss of system or data integrity is not corrected, continued use of the contaminated system or corrupted data could result in inaccuracy, fraud, or erroneous decisions.
  - **Loss of availability:** Database availability refers to making objects available to a human user or a program who/which has a legitimate right to those data objects. Loss of availability occurs when the user or program cannot access these objects.
  - **Loss of confidentiality:** Database confidentiality refers to the protection of data from unauthorized disclosure. The impact of unauthorized disclosure of confidential information can range from violation of the Data Privacy Act to the jeopardization of national security. Unauthorized, unanticipated, or unintentional disclosure could result in loss of public confidence, embarrassment, or legal action against the organization.

Database Access Control

- A DBMS typically includes a database security and authorization subsystem that is responsible for ensuring the security of portions of a database against unauthorized access. It is now customary to refer to two types of database security mechanisms:
  - **Discretionary security mechanisms:** These are used to grant privileges to users, including the capability to access specific data files, records, or fields in a specified mode (such as read, insert, delete, or update).
  - **Mandatory security mechanisms:**T hese are used to enforce multilevel security by classifying the data and users into various security classes (or levels) and then implementing the appropriate security policy of the organization. For example, a typical security policy is to permit users at a certain classification (or clearance) level to see only the data items classified at the user's own (or lower) classification level. An extension of this is role-based security, which enforces policies and privileges based on the concept of organizational roles

Database Security

- It is now customary to refer to two types of database security mechanisms:
- **Discretionary security mechanisms:** These are used to grant privileges to users, including the capability to access specific data files, records, or fields in a specified mode (such as read, insert, delete, or update).

- **Mandatory security mechanisms:** These are used to enforce multilevel security by classifying the data and users into various security classes (or levels) and then implementing the appropriate security policy of the organization.

Control Measures

- Four main control measures are used to provide security of data in databases:
  - Access control
  - Inference control
  - Flow control
  - Data encryption

- Above control measures define the level of access protections.

Control Measures

- **Access Control** is a way of preventing unauthorized persons from accessing the system itself, either to obtain information or to make malicious changes in a portion of the database. The security mechanism of a DBMS must include provisions for restricting access to the database system as a whole.
- **Statistical databases** are used to provide statistical information or summaries of values based on various criteria. Security for statistical databases must ensure that information about individuals cannot be accessed. It is sometimes possible to deduce or infer certain facts concerning individuals from queries that involve only summary statistics on groups; consequently, this must not be permitted either. The control measure used to prevent is known as **inference control**.
- Another security issue is that of **flow control, which prevents information from** flowing in such a way that it reaches unauthorized users.

Control Measures

- A final control measure is **data encryption,** which is used to protect sensitive data (such as credit card numbers) that is transmitted via some type of communications network. Encryption can be used to provide additional protection for sensitive portions of a database as well. The data is **encoded using some coding algorithm, known as encryption algorithm.** An unauthorized user who accesses encoded data will have difficulty decoding/deciphering it, but authorized users are given decoding or decrypting algorithms to decipher the data. Both of the encoding and decoding algorithms use some parameters like key for encrypting and decrypting data.

User Accounts and Database Audits

- Whenever a person or a group of persons needs to access a database system, the individual or group must first apply for a user account. The DBA will then create a new **account number and password for the user if there is a legitimate need to access the database**. The user must **log in to the DBMS by entering the account number and password** whenever database access is needed. The DBMS checks that the account number and password are valid; if they are, the user is permitted to use the DBMS and to access the database.
- It is straightforward to keep track of database users and their accounts and passwords by creating an encrypted table or file with two fields: Account Number and Password. This table can easily be maintained by the DBMS. Whenever a new account is created, a new record is inserted into the table. When an account is canceled, the corresponding record must be deleted from the table.

User Accounts and Database Audits

- The database system must also keep track of all operations on the database that are applied by a certain user throughout each **login session,** *which consists of the sequence of database interactions that a user performs from the time of logging in to the time of logging off.*
- To keep a record of all updates applied to the database and of particular users who applied each update, the ***system log is maintained.*** The **system log includes an entry for each operation applied to the database** that may be required for recovery from a transaction failure or system crash.
- If any tampering with the database is suspected, a **database audit is performed,** which consists of reviewing the log to examine all accesses and operations applied to the database during a certain time period.

User Accounts and Database Audits

- When an illegal or unauthorized operation is found, the DBA can determine the account number used to perform the operation. Database audits are particularly important for sensitive databases that are updated by many transactions and users, such as a banking database that can be updated by thousands of bank tellers.
- A database log that is used mainly for security purposes serves as an **audit trail.**

Discretionary Access Control

- Discretionary access control is based on the concept of access rights (also called privileges) and mechanism for giving users such privileges. It grants the privileges (access rights) to users on different objects, including the capability to access specific data files, records or fields in a specified mode, such as, read, insert, delete or update or combination of these. A user who creates a database object such as a table or a view automatically gets all applicable privilege on that object. The DBMS keeps track of how these privileges are granted to other users.
- The typical method of enforcing **discretionary access control** in a database system is based on the **granting** and **revoking privileges**.

Discretionary Access Control

- Discretionary Access Control (DAC) enforces security by means of user identifiers(uid) and group identifiers (gid); only the owner of the data (i.e., the Content Provider) holds the r/w permissions on the file.
- Each data object on a DAC based system has an *Access Control List* (ACL) associated with it. An ACL contains a list of users and groups to which the user has permitted access together with the level of access for each user or group. For example, *User A* may provide read-only access on one of her files to *User B*, read and write access on the same file to *User C* and full control to any user belonging to *Group 1*.

Discretionary Access Control

- The **account level**:
  - At this level, the DBA specifies the particular privileges that each account holds independently of the relations in the database.
- The **relation level** (or **table level**):
  - At this level, the DBA can control the privilege to access each individual relation or view in the database.

Discretionary Access Control

- The privileges at the **account level** apply to the capabilities provided to the account itself and can include
  - the **CREATE SCHEMA** or **CREATE TABLE** privilege, to create a schema or base relation;
  - the **CREATE VIEW** privilege;
  - the **ALTER** privilege, to apply schema changes such adding or removing attributes from relations;
  - the **DROP** privilege, to delete relations or views;
  - the **MODIFY** privilege, to insert, delete, or update tuples;
  - and the **SELECT** privilege, to retrieve information from the database by using a **SELECT** query.

Discretionary Access Control

- The privileges at the **relation level** apply base relations and views.
  - Each relation R in a database is assigned an **owner account**, which is typically the account that was used when the relation was created in the first place.
  - The owner of a relation is given all privileges on that relation.
  - The owner account holder can **pass privileges** on any of the owned relation to other users by **granting** privileges to their accounts.

Discretionary Access Control

- In **Relation Level,** following types of privileges can be granted on each individual relation *R:*
  - **SELECT (retrieval or read) privilege on *R:*** *Gives the account retrieval privilege.* In SQL, this gives the account the privilege to use the SELECT statement to retrieve tuples from *R.*
  - **Modification privileges on *R:*** *This gives the account the capability to modify* the tuples of *R. In SQL, this includes three privileges: UPDATE, DELETE,* and INSERT.
  - **References privilege on *R:*** *This gives the account the capability to reference* (or refer to) a relation *R when specifying integrity constraints. This privilege* can also be restricted to specific attributes of *R.*

Discretionary Access Control

- **Granting of Privileges:**
- Whenever the owner A of a relation R grants a privilege on R to another account B, the privilege can be given to B with or without the **GRANT OPTION. If the GRANT** OPTION is given, this means that B can also grant that privilege on R to other accounts otherwise not.

Discretionary Access Control

- **Revoking of Privileges:**
- In some cases, it is desirable to grant a privilege to a user temporarily. For example, the owner of a relation may want to grant the SELECT privilege to a user for a specific task and then revoke that privilege once the task is completed. Hence, a mechanism for **revoking privileges is needed. In SQL, a REVOKE command is included** for the purpose of canceling privileges.

Discretionary Access Control

- **Examples of Granting and Revoking**
  - GRANT INSERT, DELETE ON EMPLOYEE, DEPARTMENT TO A2;
  - GRANT SELECT ON EMPLOYEE, DEPARTMENT TO A3 WITH GRANT OPTION; (A3 can propagate grants to others)
  - GRANT SELECT ON EMPLOYEE TO A4; (A4 can't propagate grant to others)
  - REVOKE SELECT ON EMPLOYEE FROM A3;

Mandatory Access Control

- Mandatory Access Control (MAC) is based on clearance, i.e., security labels (secret, top secret, confidential, etc.). Data objects are given a security classification, and the user will be denied access if his clearance is lower than the classification of the object.
- Similarly, each user account on the system also has classification and category properties from the same set of properties applied to the data objects. When a user attempts to access a data under Mandatory Access Control the system checks the user's classification and categories and compares them to the properties of the object's security label. If the user's credentials match the MAC security label properties of the data object access is allowed. It is important to note that *both* the classification and categories must match. A user with top secret classification, for example, cannot access a resource if they are not also a member of one of the required categories for that object.

Mandatory Access Control

- Typical **security classes**
  - **Top secret (TS),**
  - **Secret (S),**
  - **Confidential (C),**
  - **Unclassified (U),**
- Here TS is the highest level and U the lowest:
- **TS > S > C > U**

Mandatory Access Control

- **Subjects**

  E.g., user, account, program

- **Objects**

  E.g., Relation, tuple, column, view, operation.

- Subjects and Objects classified into, T, S, C, or U:

- **Clearance** (classification) of a subject S denoted as **class(S**) and to the **classification** of an object O as **class(O)**.

Mandatory Access Control

- Two restrictions are enforced on data access based on the subject/object classifications:
- **Simple security property:** A subject S is not allowed read access to an object O unless class(S) ≥ class(O).
- **Star property:** A subject S is not allowed to write an object O unless class(S) ≤ class(O).
- The first restriction is intuitive and enforces the obvious rule that no subject can read an object whose security classification is higher than the subject's security clearance.
- The second restriction is less intuitive. It prohibits a subject from writing an object at a lower security classification than the subject's security clearance.

Role Based Access Control

- Its basic notion is that privileges and other permissions are associated with organizational roles rather than with individual users. Individual users are then assigned to appropriate roles. Roles can be created using the CREATE ROLE and DESTROY ROLE commands.
- For example, a company may have roles such as sales account manager, purchasing agent, mailroom clerk, customer service manager, and so on. Multiple individuals can be assigned to each role. Security privileges that are common to a role are granted to the role name, and any individual assigned to this role would automatically have those privileges granted.

Data Encryption and Decryption

- **Encryption is the conversion of data into** a form, called a **ciphertext, that cannot be easily understood by unauthorized persons.** It enhances security and privacy when access controls are bypassed, because in cases of data loss or theft, encrypted data cannot be easily understood by unauthorized persons.
- **Decryption** is the conversion Ciphertext back into plaintext that can be easily understood by authorized persons. It is a reverse process of encryption.
- **Cryptography is an art of hiding text and includes encryption and decryption process.**

Data Encryption and Decryption

- ***Ciphertext:*** *Encrypted (enciphered) data*
- ***Plaintext (or cleartext):*** *Intelligible data that has meaning and can be read or* acted upon without the application of decryption
- ***Encryption:*** *The process of transforming plaintext into ciphertext*
- ***Decryption:*** *The process of transforming ciphertext back into plaintext*
- Encryption consists of applying an **encryption algorithm to data using some prespecified encryption key. The resulting data must be decrypted using a decryption key to recover the original data.**

Data Encryption and Decryption

- **Private (Symmetric) Key Algorithms**
- A symmetric key is one key that is used for both encryption and decryption.
- By using a symmetric key, fast encryption and decryption is possible for routine use with sensitive data in the database.
- A message encrypted with a secret key can be decrypted only with the same secret key.
- Algorithms used for symmetric key encryption are called **secret key algorithms. Since secret-key algorithms** are mostly used for encrypting the content of a message, they are also called **content-encryption algorithms.**

Data Encryption and Decryption

- **Private (Symmetric) Key Algorithms**
- The major liability associated with secret-key algorithms is the need for sharing the secret key. A possible method is to derive the secret key from a user-supplied password string by applying the same function to the string at both the sender and receiver; this is known as a *password-based encryption algorithm.*
- *The strength of the symmetric key* encryption depends on the size of the key used. For the same algorithm, encrypting using a longer key is tougher to break than the one using a shorter key.
- Eg: DES, AES etc.

Data Encryption and Decryption

- **Public (Assymmetric) Key Algorithms**
- These algorithms **use two related different keys, a public key and a private key**, to perform encryption and decryption.
- The use of two keys can have profound consequences in the areas of confidentiality, key distribution, and authentication.
- The two keys used for public key encryption are referred to as the **public key and the private key.** The private key is kept secret, but it is referred to as a *private key rather than a secret key (the key* used in conventional encryption) to avoid confusion with conventional encryption. The two keys are mathematically related, since one of the keys is used to perform encryption and the other to perform decryption. However, it is very difficult to derive the private key from the public key.

Data Encryption and Decryption

- **Public (Assymmetric) Key Algorithms**
- As the name suggests, **the public key of the pair is made public for others to use, whereas the private key is known only to its owner.**
- A general-purpose public key cryptographic algorithm relies on one key for encryption and a different but related key for decryption. The essential steps are as follows:
  - Each user generates a pair of keys to be used for the encryption and decryp tion of messages.
  - Each user places one of the two keys in a public register or other accessible file. This is the public key. The companion key is kept private.
  - If a sender wishes to send a private message to a receiver, the sender encrypts the message using the receiver's public key.
  - When the receiver receives the message, he or she decrypts it using the receiver's private key. No other recipient can decrypt the message because only the receiver knows his or her private k
- Eg: RSA, ElGamal etc.

Data Encryption and Decryption

- **Public (Assymmetric) Key Algorithms**
- A public key encryption scheme, or *infrastructure, has followint ingredients:*
  - **Plaintext: This is the data or readable message that is fed into the algorithm** as input.
  - **Encryption algorithm: This algorithm performs various transformations** on the plaintext.
  - **Public and private keys: These are a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption.** The exact transformations performed by the encryption algorithm depend on the public or private key that is provided as input. For example, if a message is encrypted using the public key, it can only be decrypted using the private key.
  - **Ciphertext: This is the scrambled message produced as output. It depends** on the plaintext and the key. For a given message, two different keys will produce two different ciphertexts.
  - **Decryption algorithm: This algorithm accepts the ciphertext and the** matching key and produces the original plaintext

Statistical Databases

- Statistical databases are used mainly to produce statistics about various populations. The database may contain confidential data about individuals; this information should be protected from user access. However, users are permitted to retrieve statistical information about the populations, such as averages, sums, counts, maximums, minimums, and standard deviations.
- **A population** is a set of tuples of a relation (table) that satisfy some selection condition.
- **Statistical queries** involve applying statistical functions to a population of tuples. For example, we may want to retrieve the number of individuals in a population or the average income in the population. However, statistical users are not allowed to retrieve individual data, such as the income of a specific person

Statistical Databases

- Statistical database security techniques must prohibit the retrieval of individual data. This can be achieved by prohibiting queries that retrieve attribute values and by allowing only queries that involve statistical aggregate functions such as COUNT, SUM, MIN, MAX, AVERAGE, and STANDARD DEVIATION. Such queries are sometimes called statistical queries.

Data Warehousing

- Data warehouses are databases that store and maintain analytical data separately from transaction-oriented databases for the purpose of decision support. Regular transaction oriented databases store data for a limited period of time before the data loses its immediate usefulness and it is archived.
- On the other hand, data warehouses tend to keep years' worth of data in order to enable analysis of historical data. They provide storage, functionality, and responsiveness to queries beyond the capabilities of transaction-oriented databases.
- A data warehouse is a repository of data gathered from multiple sources and stored under a common, unified database schema. Data stored in warehouse are analyzed by a variety of complex aggregations and statistical analyses.

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. Thus, data warehouses provide the user a single consolidated interface to data, mak ing decision-support queries easier to write. Moreover, by accessing information for decision support from a data warehouse, the decision maker ensures that online transaction-processing systems are not affected by the decision-support workload.

Data Warehousing

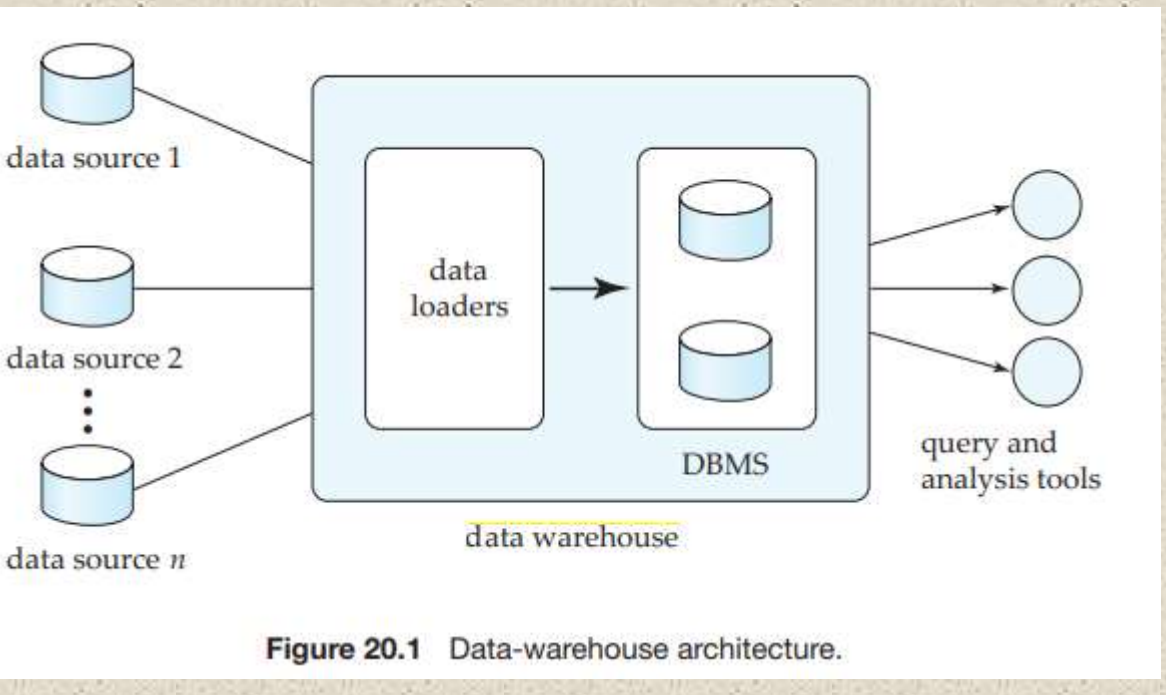


Figure 20.1 Data-warehouse architecture.

Data Warehousing

- The different steps involved in getting data into a data warehouse are called extract, transform, and load or ETL tasks; extraction refers to getting data from the sources, while load refers to loading the data into the data warehouse.
- Data warehousing is the process of constructing and using a data warehouse. A data warehouse is constructed by integrating data from multiple heterogeneous sources that support analytical reporting, structured and/or ad hoc queries, and decision making.

Data Warehousing

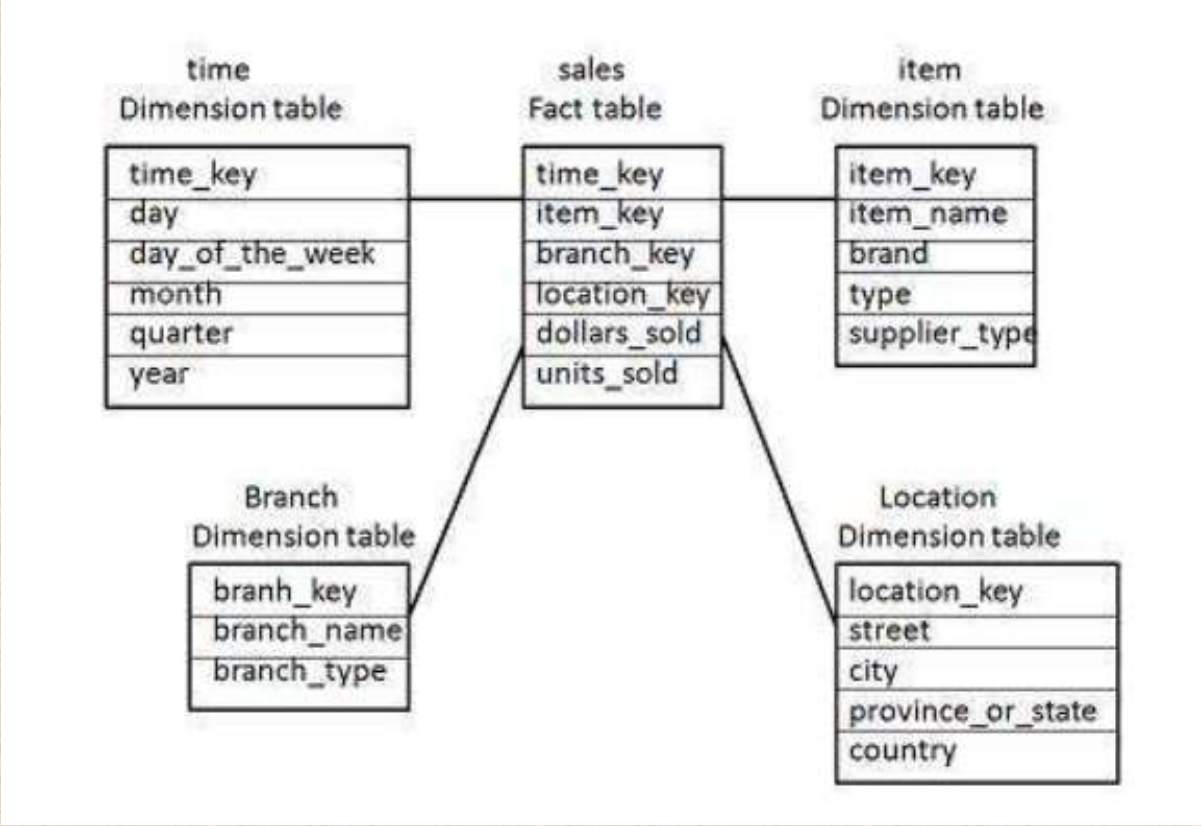
- Data warehouses are
  - **Subject-oriented:** They can analyze data about a particular subject or functional area (such as sales).
  - **Integrated:** Data warehouses create consistency among different data types from disparate sources.
  - **Nonvolatile:** Once data is in a data warehouse, it's stable and doesn't change.
  - **Time-variant:** Data warehouse analysis looks at change over time.

Data Warehousing

- Schema is a logical description of the entire database. It includes the name and description of records of all record types including all associated data-items and aggregates.
- Much like a database, a data warehouse also requires to maintain a schema. A database uses relational model, while a data warehouse uses **Star, Snowflake, and Fact Constellation schema**.

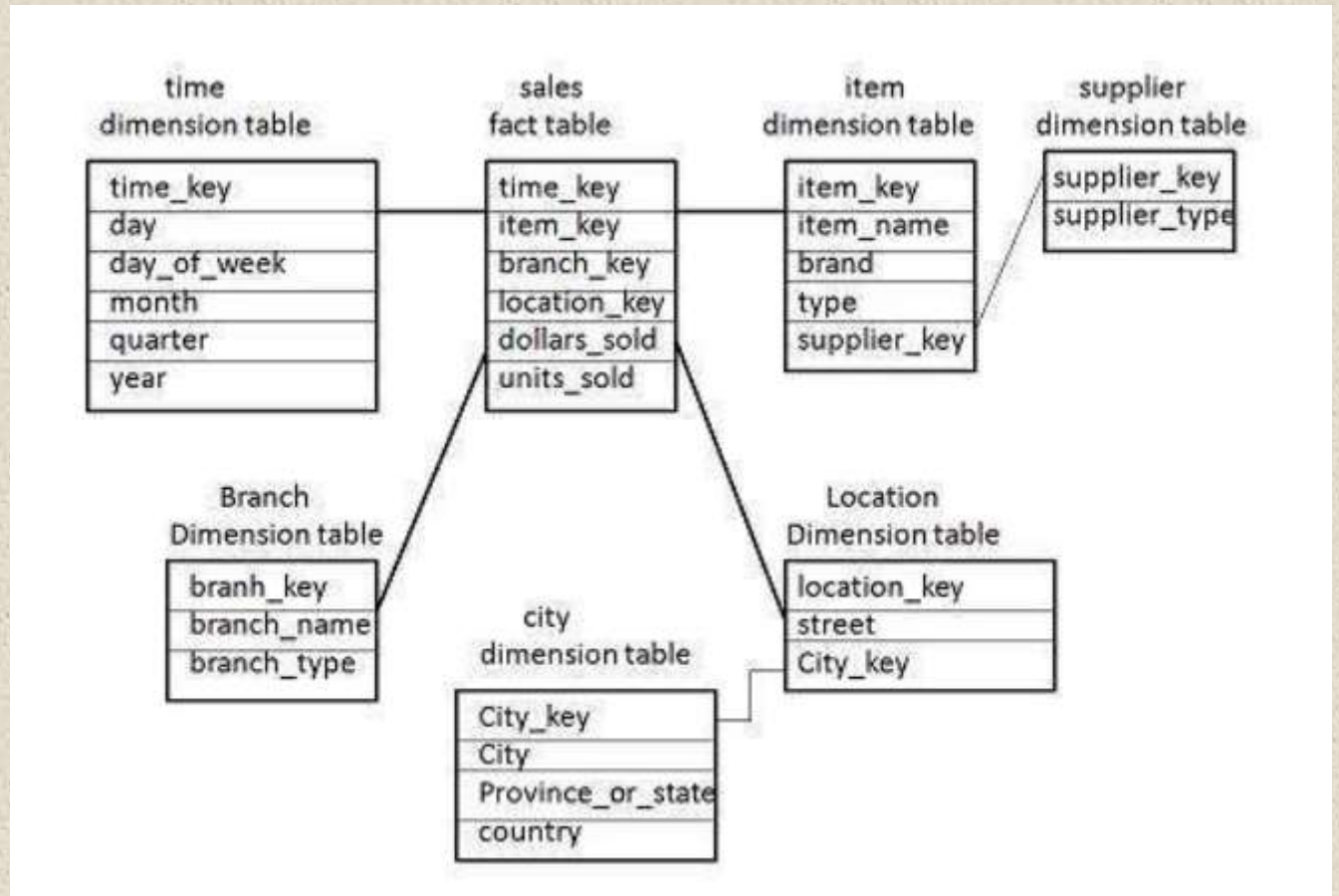
Data Warehousing

- It has a fact table and number of dimension tables.
- The dimension tables contain the set of attributes.
- The fact table contains the keys to each of dimension tables. The fact table also contains the normal attributes.



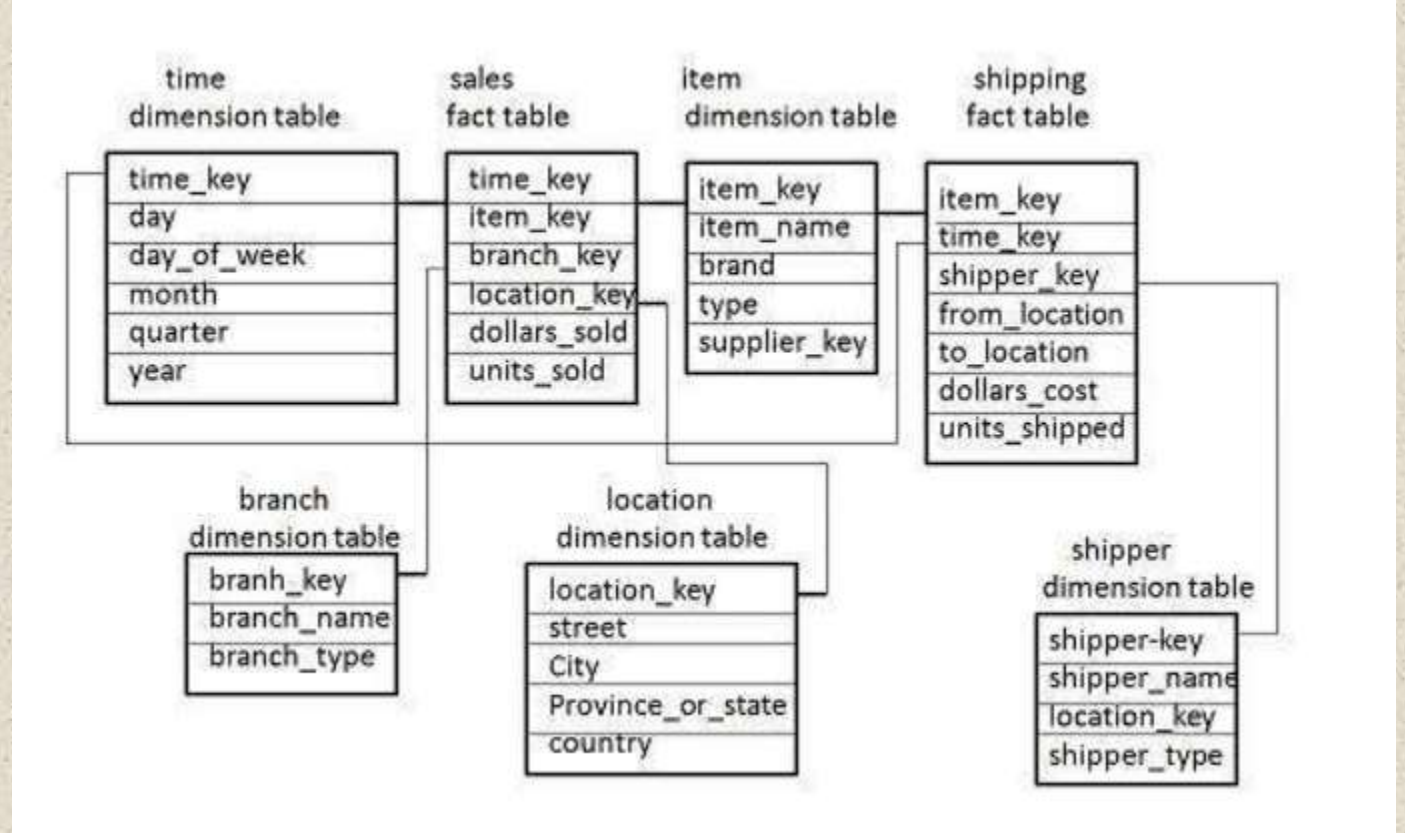
Data Warehousing

- **Snowflake Schema**
- The snowflake schema consists of one fact table which is linked to many dimension tables, which can be linked to other dimension tables through a many-to-one relationship.



Data Warehousing

- **Fact Constellation Schema**
- A fact constellation has multiple fact tables. It is also known as galaxy schema.



Data Mining

- Data mining refers to the mining or discovery of new information in terms of patterns or rules from vast amounts of data. To be practically useful, data mining must be carried out efficiently on large files and databases.
- The goal of a data warehouse is to support decision making with data. Data mining can be used in conjunction with a data warehouse to help with certain types of decisions.
- Data mining helps in extracting meaningful new patterns that cannot necessarily be found by merely querying or processing data or meta-data in the data warehouse.

Data Mining

- The result of mining may be to discover the following types of new information:
- **Association rules**—for example, whenever a customer buys video equipment, he or she also buys another electronic gadget.
- **Sequential patterns**—for example, suppose a customer buys a camera, and within three months he or she buys photographic supplies, then within six months he is likely to buy an accessory item. This defines a sequential pattern of transactions. A customer who buys more than twice in lean periods may be likely to buy at least once during the December holiday shopping period.
- **Classification trees**—for example, customers may be classified by frequency of visits, types of financing used, amount of purchase, or affinity for types of items; some revealing statistics may be generated for such classes.

Data Mining

- **The 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, the problem is to 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.
- **Classification** is the process of learning a model that describes different classes of data. The classes are predetermined. For example, in a banking application, customers who apply for a credit card may be classified as a poor risk, fair risk, or good risk. Hence this type of activity is also called **supervised learning**.

Data Mining

- **Association information** can be used in several ways. When a customer buys a particular book, an online shop may suggest associated books. A grocery shop may decide to place bread close to milk, since they are often bought together, to help shoppers finish their task faster.
- An example of an association rule is: bread ⇒ milk.
- In the context of grocery-store purchases, the rule says that customers who buy bread also tend to buy milk with a high probability.
- This is known as **association rule mining**.
- A common example is that of market-basket data. Here the market basket corresponds to the sets of items a consumer buys in a supermarket during one visit.

Data Mining

- **Clustering** refers to the problem of finding clusters of points in the given data.
- The problem of clustering can be formalized from distance metrics in several ways.
  - One way is to phrase it as the problem of grouping points into k sets (for a given k) so that the average distance of points from the centroid of their assigned cluster is minimized.
  - Another way is to group points so that the average distance between every pair of points in each cluster is minimized.
- A given population of events or items can be partitioned (segmented) into sets of “similar” elements. For example: An entire population of treatment data on a disease may be divided into groups based on the similarity of side effects produced.

Data Mining

- The previous data mining task of classification deals with partitioning data based on using a preclassified training sample. However, **Clustering** is often useful to partition data without having a training sample; this is also known as unsupervised learning.
- For example, in business, it may be important to determine groups of customers who have similar buying patterns, or in medicine, it may be important to determine groups of patients who show similar reactions to prescribed drugs.
- The goal of **clustering** is to place records into groups, such that records in a group are similar to each other and dissimilar to records in other groups. The groups are usually disjoint.

Information Retrieval

- With the advent of the World Wide Web (or Web, for short), the volume of unstructured information stored in messages and documents that contain textual and multimedia information has exploded. These documents are stored in a variety of standard formats, including HTML, XML, and several audio and video formatting standards.
- **Information retrieval (IR)** deals with the problems of storing, indexing, and retrieving (searching) such information to satisfy the needs of users. The problems that IR deals with are exacerbated by the fact that the number of Web pages and the number of social interaction events is already in the billions and is growing at a phenomenal rate.
- Information retrieval deals mainly with unstructured data, and the techniques for indexing, searching, and retrieving information from large collections of unstructured documents.

Information Retrieval

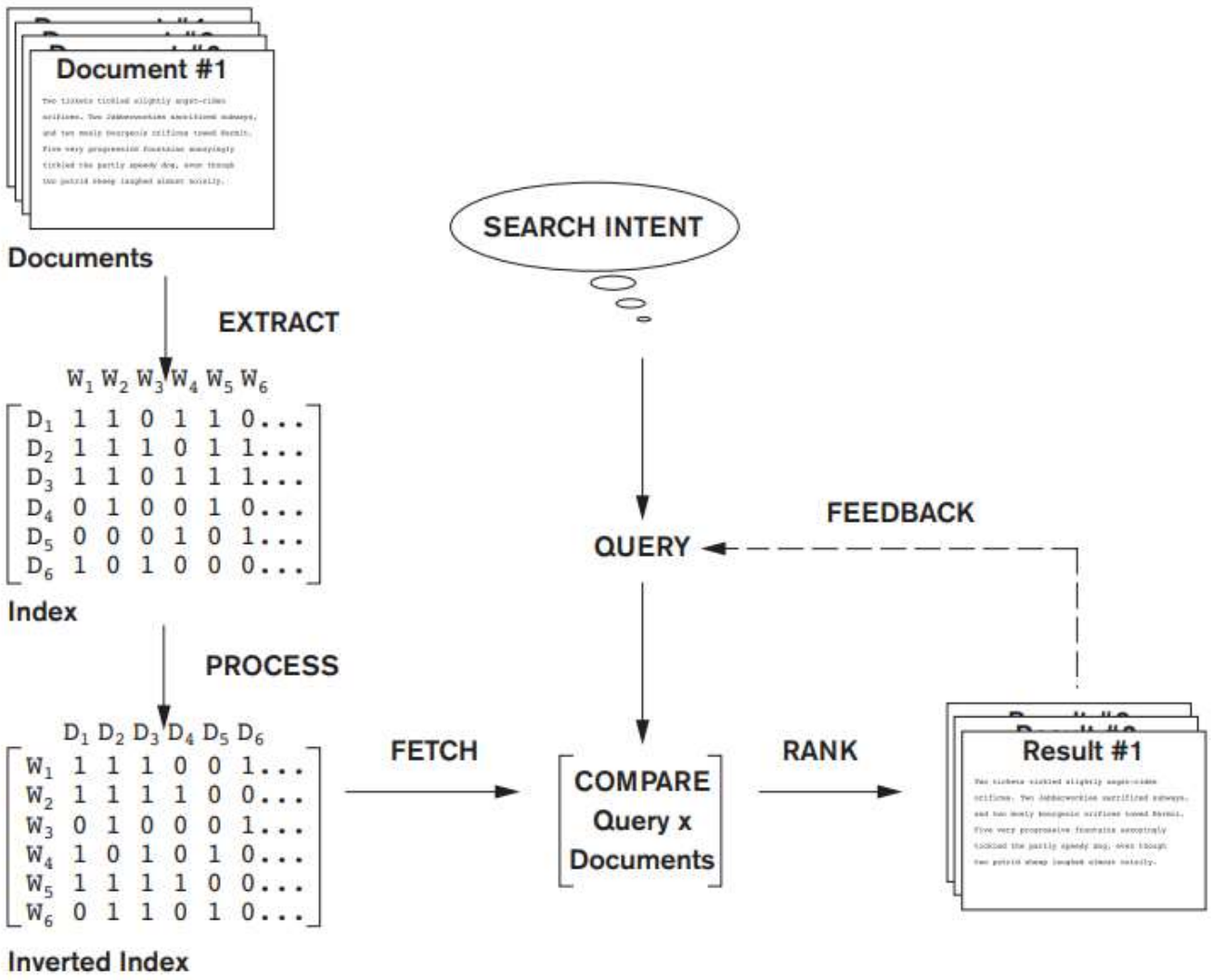


Figure 27.2
Simplified IR process pipeline.

Information Retrieval

- With the advent of the World Wide Web (or Web, for short), the volume of unstructured information stored in messages and documents that contain textual and multimedia information has exploded. These documents are stored in a variety of standard formats, including HTML, XML, and several audio and video formatting standards.
- **Information retrieval (IR)** deals with the problems of storing, indexing, and retrieving (searching) such information to satisfy the needs of users. The problems that IR deals with are exacerbated by the fact that the number of Web pages and the number of social interaction events is already in the billions and is growing at a phenomenal rate.
- Information retrieval deals mainly with unstructured data, and the techniques for indexing, searching, and retrieving information from large collections of unstructured documents.

Relevance Ranking

- Using Terms
  - TF-IDF
  - Similarity Based
- Using Hyperlinks
  - Popularity Ranking
  - PageRank
- Using Synonyms, Homonyms, and Ontologies

Crawling and Indexing the Web

- **Web crawlers** are programs that locate and gather information on the Web. They recursively follow hyperlinks present in known documents to find other documents.
- Crawlers start from an initial set of URLs, which may be created manually. Each of the pages identified by these URLs are fetched from the Web. The Web crawler then locates all URL links in these pages, and adds them to the set of URLs to be crawled, if they have not already been fetched, or added to the to-be-crawled set.
- This process is again repeated by fetching all pages in the to-be-crawled set, and processing the links in these pages in the same fashion. By repeating the process, all pages that are reachable by any sequence of links from the initial set of URLs would be eventually fetched.

Crawling and Indexing the Web

- Since the number of documents on the Web is very large, it is not possible to crawl the whole Web in a short period of time; and in fact, all search engines cover only some portions of the Web, not all of it, and their crawlers may take weeks or months to perform a single crawl of all the pages they cover.
- There are usually many processes, running on multiple machines, involved in crawling. A database stores a set of links (or sites) to be crawled; it assigns links from this set to each crawler process.
- New links found during a crawl are added to the database, and may be crawled later if they are not crawled immediately. Pages have to be refetched (that is, links recrawled) periodically to obtain updated information, and to discard sites that no longer exist, so that the information in the **search index** is kept reasonably up-to-date.

Crawling and Indexing the Web

- Pages fetched during a crawl are handed over to a prestige computation and indexing system, which may be running on a different machine. The prestige computation and indexing systems themselves run on multiple machines in parallel.
- Pages can be discarded after they are used for prestige computation and added to the index; however, they are usually cached by the search engine, to give search engine users fast access to a cached copy of a page, even if the original Web site containing the page is not accessible.
- To support very high query rates, the indices may be kept in main memory, and there are multiple machines; the system selectively routes queries to the machines to balance the load among them. Popular search engines often have tens of thousands of machines carrying out the various tasks of crawling, indexing, and answering user queries

XML Databases

- Store data in **XML(Extensible Markup Language) format**.
- XML is a meta-markup language used to manage data which employs user customizable tags to organize information. The flexibility of the language, which allows the creation of custom data structures and organizational systems, has led to its widespread use to exchange data in multiple forms.
- An XML database uses a special programming language designed specifically to extract and manipulate XML documents, known as Xquery together with XPath. The purpose of XQuery is to allow the construction of flexible queries that can extract and manipulate information from XML documents, as well as other sources that can be translated into XML. XPath is used to navigate through elements and attributes in XML.

XML Databases

- There are two major categories of these databases: **XML-enabled databases and Native XML databases (NXD)**. Each type of XML database is used to store different types of data.
- An **XML-enabled database** funnels data into a traditional relational database in an XML format. The data is translated for storage, and returned to its initial format upon output. This type of database is used to store data-centric documents which include highly structured information, such as patient records, and only use XML for data transfer.
- **Native XML databases** store XML documents as a whole, instead of separating out the data within them, and are designed to store semi-structured information. Native XML databases have an advantage over the XML-enabled database, as it is easier to store, query and maintain the XML document in a native database than in a XML-enabled database. Instead of table format, Native XML database is based on container format.

XML Databases

Sample XML data note.xml.

```
<remainder>
    <note>
      <to>Tove</to>
      <from>Jeni</from>
      <heading>Reminder</heading>
      <body>Don't forget me this weekend!</body>
    </note>
    <note> …… </note>
</remainder>
```

Xquery to display content in body of the XML element sent by Jeni:

```
for $x in doc("note.xml")/remainder/note
where $x/from="Jeni"
return $x/body
```