Chapter Seven

Advanced Concepts in Database Systems

Outline

- Transaction
- Database Security and Integrity
- Distributed Database Systems
- Data warehousing
- Data Mining

DB Transaction Concepts

- **Transaction:** Any action that reads from and/or writes to a database
- **Transaction:** Serial, Concurrent
- **Transaction Properties:** Atomicity, Consistency, Isolation, Durability
- **Transaction State:** Active, Partially Committed, Failed, Aborted and Committed
- **Concurrency Control Protocols:** Lock Based, 2PL, TSP, Validation Based P., MVCC P.)

Database Security

- The mechanisms that protect the database against intentional or accidental threats.
- Database security encompasses hardware, software, people and data
- Database Misuse (Intentional or Accidental)
  - ✓ **Accidental** (System crash, Anomalies, Logical errors)
  - ✓ **Intentional** (Unauthorized use, Modification and destruction of data)

Threats to data security

- Threat – Any situation or event, whether intentional or accidental, that may adversely affect a system
- The harm to an organization may be tangible or intangible.
  - ✓Tangible – loss of hardware, software, or data
  - ✓ Intangible – loss of credibility or client confidence, loss of integrity, loss of availability, and Loss of confidentiality.

Situations to Consider data Security

- Accidental losses, including human error, software and hardware caused breaches
- Theft and fraud
- Loss of privacy or confidentiality
- Loss of data integrity
- Loss of availability

Goals of Database Security

- **Secrecy:** Users should not be able to see things they are not supported to.
  - ✓Example a student can't see other students' grades
- **Integrity:** Users should not be able to **modify** things they are not supported to.
  - ✓Example only instructors can assign grades
- **Availability:** Users should be able to see and modify things they are allowed to.

Security Issues and Considerations

- **Legal, ethical and social issues** regarding the right to access information.
- **Physical control**
- **Policy issues** regarding privacy of individual level at enterprise and national level.
- **Operational consideration** on the techniques used (password, etc.).
- **System level security** including operating system and hardware control.
- **Security levels and security policies** in enterprise level.

Levels of Security Measures

- **Physical Level:** securing the site containing the computer system
- **Human Level**: authorization of database users for access
- **Operating System Level**: weakness and strength of the operating system security on data files
- **Database System Level**: data access limit enforced by the database system
- **Network Level**: encryption of transmission data on the network

Security Measures

- Authorization – Access Controls
- Views – Using Virtual Relations
- Integrity – Domain, Entity, Referential and Enterprise
- Backup and recovery – Copy of database and log file
- Encryption – Encoding data using special algorithms
- Authentication – Checking the user (authorized or not)

Encryption

- To transmit data securely over insecure networks requires the use of a **Cryptosystem**, which includes:

1. An **encryption key** to encrypt the data (plaintext)
2. An **encryption algorithm** that, with the encryption key, transforms the plaintext into cipher text
3. A **decryption key** to decrypt the cipher text
4. A **decryption algorithm** that, with the decryption key, transforms the cipher text back into plaintext

Types of Cryptosystems

- Cryptosystems can be categorized into two:
  - **Symmetric encryption –** uses the same key for both encryption and decryption and relies on safe communication lines for exchanging the key
  - **Asymmetric encryption –** uses different keys for encryption and decryption
- Symmetric algorithms are **much faster** to execute on a computer than those that are asymmetric.
- In the contrary, asymmetric algorithms are **more secure** than symmetric algorithms

Components of DB access Request

- **Requested Operation:** what kind of operation is requested by a specific query?
- **Requested Object:** on which resource or data of the database is the operation sought to be applied?
- **Requesting User:** who is the user requesting the operation on the specified object?

Forms of user Authorization

- **Read Authorization:** the user with this privilege is allowed only to read the content of the data object.
- **Insert Authorization**: the user with this privilege is allowed only to insert new records or items to the data object.
- **Update Authorization**: users with this privilege are allowed to modify content of attributes but are not authorized to delete the records.
- **Delete Authorization**: users with this privilege are only allowed to delete a record and not anything else.

Role of DBA in Database Security

- **Account Creation**: creating accounts for USERS and USER GROUPS.
- **Security Level Assignment:** assigning different users at access levels.
- **Privilege Grant**: giving levels of privileges for users and user groups
- **Privilege Revocation**: denying or canceling previously granted privileges for users.
- **Account Deletion**: deleting an existing account of users or user groups.
  - ✓ Is similar with denying all privileges of users on the database.

Example 1: Grant

- Suppose that the DBA creates four accounts: A1, A2, A3, A4 and wants only A1 to be able to create relations. Then the DBA must issue the following GRANT command in SQL

```
GRANT CREATE TO A1;
```

Example 2: Grant without Propagation

- Suppose that A1 **creates** the two base relations **EMPLOYEE** and **DEPARTMENT**
- Suppose that A1 wants to grant A2 the privilege to insert and delete rows in both of these relations, but A1 does not want A2 to be able to propagate these privileges to additional accounts:

```
GRANT INSERT, DELETE ON EMPLOYEE, DEPARTMENT TO A2;
```

Example 3: Grant with Propagation

- Suppose that A1 wants to allow A3 to retrieve information from either of the table (Department or Employee) and also to be able to propagate the SELECT privilege to other accounts.
- A1 can issue the command:

```
GRANT SELECT ON EMPLOYEE, DEPARTMENT
TO A3 WITH GRANT OPTION;
```

- A3 can grant the SELECT privilege on the EMPLOYEE relation to A4 by issuing: `GRANT SELECT ON EMPLOYEE TO A4;`
- Notice that A4 can't propagate the SELECT privilege because GRANT OPTION was not given to A4

Example 4: Revoking privilege

- Suppose that A1 decides to revoke the SELECT privilege on the EMPLOYEE relation from A3, A1 can issue:

```
REVOKE SELECT ON EMPLOYEE FROM A3;
```

- The DBMS must now automatically revoke the SELECT privilege on EMPLOYEE from A4 too. because A3 granted that privilege to A4 and A3 does not have the privilege any more.

Example 5: Grant with View

- Suppose that A1 wants to give back to A3 a limited capability to SELECT from the EMPLOYEE relation and wants to allow A3 to be able to propagate the privilege.
- A1 then create the view:

```
CREATE VIEW A3EMPLOYEE AS
  SELECT NAME, BDATE, ADDRESS FROM EMPLOYEE
  WHERE DNO = 5;
```

- After the view is created, A1 can grant **SELECT** on the view A3EMPLOYEE to A3 as follows:

```
GRANT SELECT ON A3EMPLOYEE TO A3 WITH GRANT OPTION;
```

Example 6: Granting update Column

- Suppose that A1 wants to allow A4 to update only the SALARY attribute of EMPLOYEE;

- A1 can issue:

```
GRANT UPDATE ON EMPLOYEE (SALARY) TO A4;
```

Distributed Database Systems

- A transaction can be executed by multiple networked computers in a unified manner.
- A **distributed database (DDB)** processes Unit of execution (a transaction) in a distributed manner.
- A distributed database (DDB) can be defined as :
  - A collection of multiple logically related database distributed over a computer network, and a distributed database management system as a software system that manages a distributed database while making the distribution transparent to the user.
  - The physical placement of data (files, relations, etc.) which is not known to the user (distribution transparency).

Distributed DB Functions (1)

- Keep track of where data are located in a distributed data dictionary.
- Determine the location from which to retrieve requested data
- Provide data management functions
- Provide consistency among copies of data across the remote sites
- Present a single logical database that is physically distributed

Distributed DB Functions (2)

- Scalability
- Replicate both data and stored procedures across the nodes of the distributed database
- Transparently use residual computing power to improve the performance of db processing
- Permit different nodes to run different DBMS.
- Allow different versions of application code to reside on different nodes of the distributed db.

Concepts in DDBS

- Replication: System maintains multiple copies of data, stored in different sites, for faster retrieval and fault tolerance.
- Fragmentation: Relation is partitioned into several fragments stored in distinct sites.
- Data transparency: Degree to which system user may remain unaware of the details of how and where the data items are stored in a distributed system.
- Local Transaction: Transactions that access data only in that single site.
- Global Transaction: Transactions that access data in several sites.

Advantages of DDBS

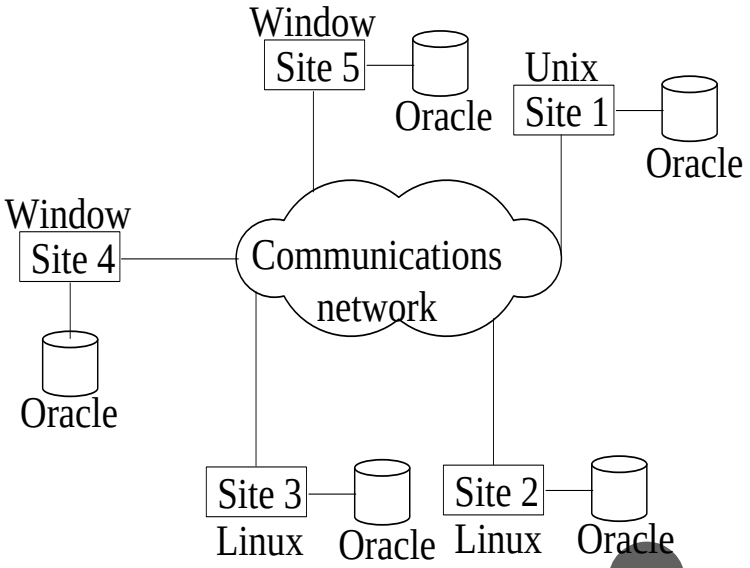
- Data sharing and distributed control
- Reliability and availability of data
- Speedup of query processing

Disadvantages of DDBS

- Software development cost.
- Greater potential for bugs (parallel processing may endanger correctness
- Increased processing overhead (due to communication jargons)
- Communication problems

Homogeneous Distributed DB

- All sites have identical software.
- All sites are aware of each other and agree to cooperate in processing user requests.
- Each site surrenders part of its autonomy in terms of right to change schemas or software
- Appears to user as a single system



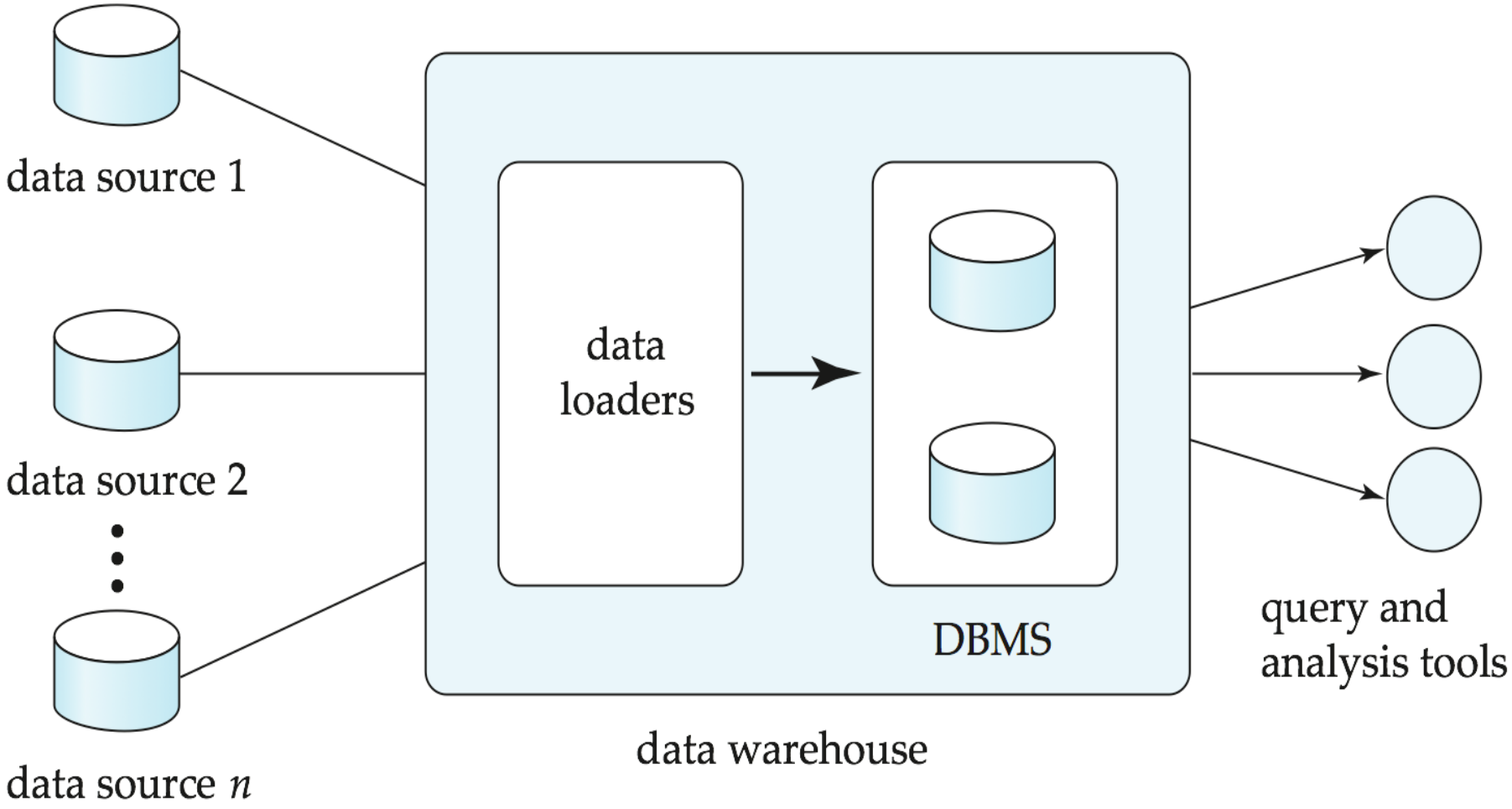
Heterogeneous Distributed DB

- Different sites may use different schemas and software.
- Difference in schema is a major problem for query processing.
- Difference in software is a major problem for transaction processing.
- Sites may not be aware of each other and may provide only limited facilities for cooperation in transaction processing.
- **Federated and Multidatabase**

Data Warehousing

- An integrated
- Subject-oriented
- Time-variant
- Non-volatile database
- Provides support for decision making or used primarily in organizational decision making

Data Warehousing Architecture



Data warehouse and data in OLTP

OLTP System	Data Warehousing System
Current data	Historical data
Detailed data	Summarized
Dynamic data	Largely static
Repetitive and already known operation	Ad hoc and unstructured operations
Predicted pattern of usage	Unpredictable pattern of usage
Transaction driven	Analysis driven
Application oriented	Subject oriented
Support day-to-day decisions	Support strategic decision
Large number of operational users	Low number of managerial users

Data Mining

- Data mining is the extraction or ‘mining’ of knowledge from large amount of data
- Data in the real world is dirty (incomplete, Noisy and Inconsistent)
- No quality data, no quality mining results!
- The first task in data mining is to preprocess the data warehouse so that quality data is used for knowledge extraction

Major Tasks in Data Preprocessing

- **Data Cleaning:** Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- **Data Integration:** Integration of multiple databases, data cubes, or files
- **Data Transformation:** Normalization and aggregation
- **Data Reduction:** Obtains reduced representation in volume but produces the same or similar analytical results
- **Data Discretization:** Part of data reduction but with particular importance, especially for numerical data

Thank You!

End of the Course