Data Integrity and Security

CHAPTER 3

Why Database Security?

- Most online transactions involve a database
  Water supplies, electricity grids, and gas and oil production depend on a computer network to thrive
- Breach could have disastrous impact
- Network intruders are well trained and growing more sophisticated

A Secure Data Environment

- Multiple layers of security
- Most effective approach to minimizing risk of data breach
- Example of multiple security layers to protect against malicious e-mail attachments
- User awareness training
- Filter on exchange server to remove known malicious attachments
- Firewall configured to deny certain types of traffic

A Secure Data Environment (cont'd.)

- Database security
- Set of established procedures, standards, policies, and tools
- Protects against theft, misuse, and attacks
- Deals with permission and access to the data structure
- Common vendor features for database security
- Database-level access control
- Database-level authentication
- Data storage encryption

Data Integrity

What is Data Integrity?

- Data integrity is a fundamental aspect of database management, ensuring that data stored in a database is accurate, consistent, and reliable. It encompasses the following key aspects:
- **Accuracy:** Data is correct and free from errors or inconsistencies.
- **Consistency:** Data remains coherent and uniform throughout the database.
- **Reliability:** Data can be trusted and depended upon for decision-making.
-

Data Integrity (Types of Data Integrity)

1. Entity Integrity

Entity integrity ensures that each row (record) in a table has a unique identifier. This is typically achieved through primary keys, which are unique identifiers for each record in a table. For example:

Example: Consider an "Employees" table. The Employee ID, designated as the primary key, ensures that each employee has a unique identifier.

Data Integrity (Types of Data Integrity)

2 Referential Integrity

Referential integrity focuses on maintaining relationships between tables. It ensures that foreign key values in one table correspond to primary key values in another table. This prevents orphans (records with no related data) and maintains the consistency of relationships. For example:

Example: In a database for a library, you have a "Books" table and a "Borrowers" table. The "Borrowers" table includes a foreign key, "BookID," which references the primary key, "BookID," in the "Books" table. This ensures that each borrowed book corresponds to a valid book record in the "Books" table.

Data Integrity (Types of Data Integrity)

3. Domain Integrity

Domain integrity enforces valid data types and constraints on data values. It ensures that data in a specific column adheres to predefined rules, such as data type, length, and permissible values.

Examples include:

Data Type:

Ensuring that a column designed to store dates only contains date values.

Length Constraint:

Limiting a column to accept only text values of a certain length (e.g., a maximum of 50 characters).

Check Constraints:

Specifying conditions that data must meet to be valid. For example, ensuring that a "Discount" column only contains values between 0 and 1.

Data Integrity (Techniques)

Several methods and techniques are employed to ensure data integrity:

- **Data Validation Rules:**

  Implement rules that validate data as it's entered into the database. For example, a rule may ensure that email addresses are properly formatted.

- **Constraints:**

  Use database constraints like UNIQUE, NOT NULL, and CHECK to enforce data integrity rules. For example, setting a UNIQUE constraint on an email column ensures that each email address is unique.

Data Integrity (Techniques)

- **Triggers:**

  Triggers are special procedures that are automatically executed when specific database events occur. They can be used to enforce custom data integrity rules. For example, a trigger can ensure that a certain action is taken whenever a record is deleted.

- **Database Management Systems (DBMS):**

  DBMS systems like Oracle, SQL Server, and MySQL offer built-in mechanisms to enforce data integrity through the implementation of constraints and triggers.

Database Security

What is Database Security?

Database security is the practice of protecting data in a database from unauthorized access, disclosure, alteration, or destruction. It aims to ensure the confidentiality, integrity, and availability of the data stored in a database.

Database Security (Threats to Database Security)

1. Unauthorized Access

Unauthorized access refers to individuals or entities gaining access to a database without proper authorization. This can occur through various means, such as exploiting weak passwords, bypassing authentication mechanisms, or using stolen credentials.

- **Example:** An employee at a company leaks login credentials, allowing unauthorized personnel to access the company's customer database.

Database Security (Threats to Database Security)

2. SQL Injection Attacks

SQL injection is a type of attack where malicious SQL statements are injected into an application's input fields, potentially allowing an attacker to execute unauthorized SQL queries on a database.

- **Example:** A poorly coded website allows users to input text into a search bar, and a malicious user enters SQL code that retrieves sensitive data from the database.

Database Security (Threats to Database Security)

3. Insider Threats

Insider threats involve individuals within an organization who misuse their privileges to harm the organization. This could include employees, contractors, or business partners.

- **Example:** An employee with legitimate access to the HR database abuses their privileges to view and leak confidential employee salary information.

Database Security (Threats to Database Security)

4. Malware and Ransomware

Malware, such as viruses, worms, and Trojans, can infect a system and potentially compromise the database. Ransomware can encrypt the database and demand a ransom for decryption.

- **Example:** An employee opens an email attachment infected with malware, which spreads to the database server and compromises data.

Database Security (Threats to Database Security)

5. Data Theft and Data Leakage

Data theft refers to the unlawful acquisition of sensitive data from a database, while data leakage refers to unintentional exposure or release of data.

- **Example:** A company's database administrator accidentally uploads a backup of the customer database to a public file-sharing service, exposing sensitive customer data.