✅ 1. Security

Goal: Prevent **unauthorized access** to the database.

How it is maintained:

- **Authentication**: Only verified users can access (e.g., username/password, token).
- **Authorization**: Users get **only specific permissions** (e.g., READ, WRITE, DELETE).
- **Roles and Privileges**: Admins define what **each role** can do.
- **Encryption**:
  - **At rest**: Data in the database is encrypted.
  - **In transit**: Data sent over the network is encrypted (e.g., using SSL/TLS).
- **SQL Injection Prevention**: Using **prepared statements** to avoid injection attacks.

✅ 2. Integrity

Goal: Ensure **accuracy and consistency** of data.

How it is maintained:

- **Constraints**:
  - **Primary Key**: Ensures uniqueness.
  - **Foreign Key**: Maintains referential integrity between tables.
  - **Not Null, Unique**: Controls what kind of data is allowed.
- **ACID Properties**:
  - **Atomicity**: All parts of a transaction complete, or none do.
  - **Consistency**: Database goes from one valid state to another.
  - **Isolation**: Transactions don't interfere with each other.
  - **Durability**: Changes are permanent once committed.
- **Triggers** and **Stored Procedures**: Enforce rules automatically.

✅ 3. Concurrency Control

Goal: Manage **multiple users** accessing the same data **at the same time**.

How it is maintained:

- **Locks**:
  - **Shared Lock**: For reading.
  - **Exclusive Lock**: For writing.

- **Timestamp Ordering**: Each transaction gets a timestamp, and ordering ensures consistency.
- **Two-Phase Locking (2PL)**:
  - **Growing Phase**: Acquire all locks.
  - **Shrinking Phase**: Release locks.
- **Optimistic Concurrency Control**: Checks for conflicts **at the end** before commit.
- **MVCC (Multi-Version Concurrency Control)**:
  - Each transaction sees a snapshot of the data, avoiding read-write conflicts.
  - Common in PostgreSQL, Oracle, etc.