

What is Database Security?

Database security is the practice of protecting databases from unauthorized access, misuse, data breaches, data loss, and corruption.

In simple words:

👉 *It keeps your data safe, private, accurate, and available.*

Why Database Security is Important

Databases store **critical information** like:

- Usernames & passwords
- Bank and payment data
- Personal details (Aadhaar, phone numbers, emails)
- Business secrets

If compromised, it can lead to:

- Identity theft
- Financial loss

- Legal penalties
- Loss of trust

Core Goals of Database Security (CIA Triad)

You might recognize this from cyber security 

1 Confidentiality

Only authorized users can access data.

- Login authentication
- Role-based access
- Encryption

2 Integrity

Data must not be altered without permission.

- Constraints
- Checksums
- Transaction controls

3 Availability

Data should be accessible when needed.

- Backups
- Failover systems
- Protection against DoS attacks



Common Database Threats

Threat	Explanation	SQL Injection	Attacker
inserts malicious SQL queries	Unauthorized Access	Weak passwords or misconfigured permissions	Insider Threat
misusing access	Employees	Data Leakage	Poor encryption or backups
exposed server	Malware	Database infected via	DoS Attacks
and crashes	Database overwhelemed		



Database Security Techniques

1. Authentication

Verifying user identity:

- Username & password

- Multi-factor authentication (MFA)

2. Authorization (Access Control)

Defines what a user can do:

- Role-Based Access Control (RBAC)
- Least privilege principle

Example:

Admin → full access User → read only

3. Encryption

Protects data even if stolen.

- **At rest** → Encrypted stored data
- **In transit** → SSL/TLS encryption

Example:

Plain text → Encrypted text → Decrypted
for authorized user

4. Input Validation & SQL Injection Prevention

- Use prepared statements
- Avoid dynamic SQL queries

Bad :

SELECT * FROM users WHERE id = '\$id';

Good :

SELECT * FROM users WHERE id = ?;



5. Auditing & Logging

Tracks:

- Who accessed data
- What changes were made
- When it happened

Helps in:

- Detecting attacks
- Legal compliance



6. Backup & Recovery

- Regular backups
- Secure backup storage
- Disaster recovery plans



7. Database Hardening

- Disable unused features
- Change default credentials
- Patch and update DB software



Real-World Example

Online Banking System

- Passwords → Hashed & encrypted
- Users → Limited access to their own accounts
- Admins → Audited continuously
- Transactions → Logged & monitored

