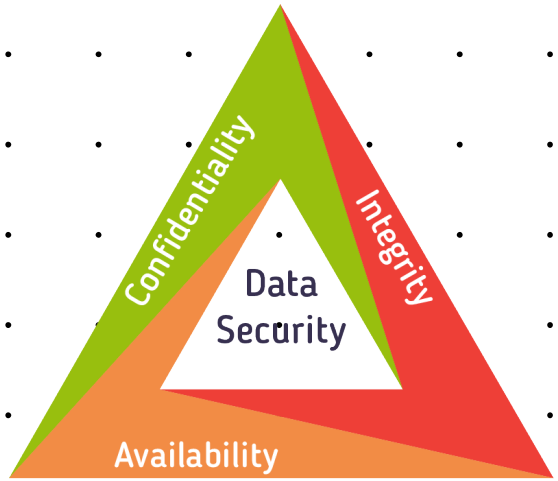
Database Security

Database CIA:

- Confidentiality
- Integrity
- Availability

Databases have several systems in place to:

- maintain privacy
- prevent data corruption and
- ensure availability



Two-Phase Commit

- Allows simultaneous write access to a database without risk of data corruption.
- Request phase
  - Upload proposed changes to database
  - DBMS locks needed records. If it can't lock them, it changes nothing and aborts the transaction.
- Commit phase
  - Changes all the records it has locked and returns a success code. If anything goes wrong, it reverses and changes it has made

DB Access Control

- DBMSs use ACLs or permission attributes to control who reads from and writes to the database, tables, columns.
- Implements DAC, MAC (users may not be given permission to change permissions for other users).
- Should be set up according to the principle of least privilege.

Granting Permission

- GRANT SELECT ON users TO EvilHacker
- Can also grant DELETE, INSERT, UPDATE
- Can grant permissions to ALL, PUBLIC

Discretionary Access Control (DAC) Permission

- DAC can be implemented by creating a user-specific view and granting GRANT rights to it's "owner"

- CREATE VIEW user_alice
  AS SELECT * from users where name='Alice';
  GRANT SELECT ON user_alice to Alice with GRANT OPTION;

Removing Permission

- If a user gets demoted or leaves the organisation, permissions to the database must be removed before they can retaliate:
  - REVOKE SELECT ON users FROM Alice;
- This has a cascade effect of removing permissions from everyone who was approved by Alice

Database Encryption

- Sensitive data should be encrypted before storing it in the database.
- Some databases store data in plain text or human-readable form, so an attacker with hard-drive access can read data without using the DBMS
- DBMSs offer a range of encryption and decryption functions ranging from symmetric keys to public-private crypto.