Database Security

**Integrity** is concerned with accidental corruption.

* Integrity constraints.

**Security** is concerned with deliberate corruption.

* Security policies
* Access control

### Access Control

Prevent unauthorised persons from accessing the system.

DBMSs can restrict access to the database:

* User accounts
* Privileges
* Security levels

The **database administrator** is responsible for:

* User account creation - Encrypted table maintained by DBMS.
* Privilege granting and revocation (**discretionary** access control).
* Security level assignment (**mandatory** access control).
* Role-based access

Discretionary Access Control

**Access privileges** can be specified at two levels:

|  |  |
| --- | --- |
| Account level | Privileges that each account holds independent of relations.  Account privileges:   * CREATE SCHEMA * CREATE TABLE * CREATE VIEW * ALTER * DROP |
| Relation level | Privileges to access each individual relation / view.  Can be specified on entire relations, or on specific attributes.  Determines what **operations** can be performed.  Each relation has an **owner**, who controls granting / revoking of privileges.  Relation privileges:   * **Read** - SELECT * **Modify** - INSERT / UPDATE / DELETE * **Reference** - Can refer to relation when specifying integrity constraints. |

### Views

Views allow the owner of a relation to grant **partial access** to the information in a relation.

* Restricted set of attributes.
* Restricted set of rows.

A view acts as a **new relation** in the database.

SQL example:

|  |
| --- |
| CREATE VIEW view AS  SELECT attr1, attr2  FROM table  WHERE condition |

### Granting Privileges

The **owner** of a relation automatically has all relation privileges, and can **grant** other user privileges for that relation:

|  |
| --- |
| GRANT privilege TO user;  GRANT privilege ON relation TO user; |

### Revoking Privileges

|  |
| --- |
| REVOKE privilege FROM user;  REVOKE privilege ON relation FROM user; |

### Propagation of Privileges

Privileges can be given with the grant option, which allows the user to propagate that privilege to other users:

|  |
| --- |
| GRANT privilege ON relation TO user WITH GRANT OPTION; |

If the owner revokes the privilege on a user, then privileges granted by the user should be automatically revoked by the DBMS.

Mandatory Access Control

MAC classifies data and users based upon **security levels**:

1. Top Secret
2. Secret
3. Confidential
4. Unclassified

MAC can be combined with discretionary access control.

Each subject and object are given a security level.

|  |  |
| --- | --- |
| **Subject** | **Object** |
| User account  Application program | Relation  Tuple  Attribute  View  Operation |

Applicability vs. security:

|  |  |
| --- | --- |
| **Discretionary Access Control** | **Mandatory Access Control** |
| Flexible  Complex to manage  Can be vulnerable to malicious attacks. | Rigid  Very secure |

Role-Based Access Control

Privileges and other permissions are associated with **organisational roles** rather than individual user accounts. Users are assigned to appropriate roles.

Roles can created in SQL using:

|  |
| --- |
| CREATE ROLL  DESTROY ROLL  GRANT role TO user |

Multiple users can be assigned to each role.

Any individual can be assigned to **multiple roles**.