Security

- The data stored in A DBMS is often stored in a company and is regarded as a corporate asset
- In addition to protecting the insecurity of the data corporations must consider ways to ensure privacy and controlled access to data that must not be revealed to certain groups of users for various reasons
- An access control mechanism is a way to control the data accessible by a given user

Introduction to database security

There are 3 main objectives when designing a secure database application:-

1. Accessibility
2. Security
3. Integrity

Secrecy

Information should not be disclosed to unauthorized users.

Ex:- A student should not be allowed to examine other student grades

Integrity

Only authorized users should be allowed to modify data

Ex:- Students may be allowed to see their grades but are not allowed to modify their grades

Availability

Authorized users should not be access

Ex:- An instructor who wishes to change a grade should be allowed to do so

To achieve these objectives a clear consistent security policy should be developed to describe what security measures should be unforced

Access Control

- A database for an enterprise contains a great deal of information and usually has several groups of users.
- Most users need to access only a small part of the database to carry out their tasks

- Allowing users unrestricted access to all the data can be undesirable and a DBMS should provide mechanisms to control access to data
- A DBMS offers to main approaches to access control:-
    - Discretionary control
    - Mandatory control

Discretionary control (DSC)

- Is based on a access rights or privs and mechanisms to giving access to privs
- A user who creates a database object such as a table or a view automatically gets all applicable privs
- The DBMS subsequently keeps track of how these privs are granted to other users and possibly revoked an ensures that at all times only users within the necessary privs can access an object
- SQL supports DSC through the Grant and Revoke commands
- The Grant command gives privs to the users and the Revoke command takes away privs

```
-- Grant command example
GRANT privilege ON object TO Users [With GRANT Option]
```

Options which can be used with GRANT command

SELECT:- The right to access or read all columns of the table including columns added later to alter table commands

INSERT:- [Column name] the right to insert rows including columns that might be added later, the privs can be used for update similarly

DELETE:- The right to delete rows from the table

REFERENCES:- the right to define foreign keys [Other tables] that refer to the pacific column of the table

Ex:-

```
GRANT INSERT ON sailors TO Sakura
```

Sakura is the personal who has the INSERT priv with respect to the newly added column

Syntax for REVOKE

```
-- Revoke command example
REVOKE [GRANT Option FOR] privilege  ON Object from Users {RESTRICT | CASCADE }
```

One of the two alternatives RESTRICT or CASCADE must be specified

Ex:-

```
GRANT SELECT ON sailors TO Art WITH GRANT Options
```

```
REVOKE [GRANT Option FOR] SELECT ON sailors FROM Art CASCADE
```

NOTE:-

This command would leave Art with the SELECT priv on sailors but Art no longer has the GRANT option on this priv and therefore cannot pass it on to other users

```
GRANT and REVOKE on Integrity constraints – No Idea what this is??
```

- The priv held by the creator of a view change over time as he/she or loses privs on the underlying tables
- If the creator loses a priv held with the GRANT option users who were given that priv on the view lose it as well
- They are some subtle aspects to the GRANT and REVOKE commands when they involve Views or Integrity constans

Ex:-

1. A view maybe dropped because a SELECT priv is revoked from the user who created the View
2. If the creator of a View gains additional privs on the underlying table he/she automatically gains additional privs on the View

The Discrimination between the references and selected privs important

Mandatory Access Control (MAC)

- Is based on system-wide policies that cannot be changed by individual users
- In this approach each database object is assigned a security class, each user is assigned clearance of a security class and rules are imposed on reading and writing of database objects by users
- Discretionary access controls mechanisms, while generally effective have certain weakness
- In particular they are sustainable to trojan horse schemes whereby a devious unauthorized user can trick and authorize into disclosing sensitive data

Difference between DAC and MAC

DAC	MAC
DAC stand for Discretionary control	Mandatory Access Control
Is easier to implement	Is difficult to implement
Less Secure to use	More Secure to use
The Owner can determine the access and can restrict the resources based on the identity of the users	The system only determines the access and the resources will be restricted based on the clearance of the subject
DAC has extra labor intensive properties, Alot of physical work	MAC has no labor intensive properties
Users will be provided based on their identity and not using levels	Users will be restricted based on their power and level hierarchy
It has high flexibility with no rules and regulations	MAC is not flexible
DAC has complete trust in users	MAC has trust only in admins
Decisions will be based only on user id and ownership	Decisions will be based on objects and tasks and they can have their own ids
Information flow is impossible to control	Information flow can be easily control
DAC is supported by commercial DBMS	MAC is not supported by commercial DBMS
DAC can be applicable in all domains	MAC can be applicable in government , military and intelligence
DAC is vulnerable to trojan horses	MAC prevents virus flow from a higher level to a lower level

Crash Recovery

The recovery manager of a DBMS is responsible for enduring two important properties of transactions:-

1. Atomicity
2. Durability

- It ensuring Atomicity by undergoing the transactions that do not commit
- Durability is making sure that all actions of committed transactions survive system crashes and media failures
- The Recovery Manager is one of the hardest to design and implement in DBMS
- It must be with a wide variety of database states because it is called during system failures

Introduction to Aries

- Aries is a Recovery Algorithm design to work with a stream, no force approach
- When the recovery manager is invoked after a crash, the restart procedes in 3 phases

1. Analyzes:- It identifies dirty pages in the buffer pool (The changes that have not been written to the disk) and active transactions at the time of the crash
2. Redo:- It repeats all actions starting from an appropriate point in the log and restores the database state to what it was at the time of the crash
3. Undo:- Undos the action of the transactions that did not commit, so that the database reflects only the actions of committed transactions will be undone

Three main principles lie behind the Aries Recovery Algorithm:-

1. Write ahead lobby:- Any changes to a database object is first recorded in the log. The record in the log must be written to stable storage before the change to the database object is returned to the disk

2. Repeating history during the redo:- On restart following a crash in Aries replaces all actions of the DBMS before the crash and brings the system back to the exact state that it was in at the time of the crash. Then it undoes the actions of transactions still active at the time of the crash

3. Login changes during undo:- Changes made during the database while undoing a transaction are logged to ensure such an action is not repeated in the event of a repeated restarts

The Log

- The log, sometimes called the trail or journal is a history of actions executed by the DBMS

- Physically the log is a file of logs stored in stable storage which is assumed to survive crashes, this durability can be achieved by maintaining by 2 or more copies of the log on different disks so that the chance of all copies of the log being simultaneously lost is negligibly small

- The most recent portion of the log called The Log Tail is kept in main memory and is periodically is forced to stable storage

- Every log record is given a unique ID called the log sequence number (LSN) as with any record [Some fucking word which I forgot] we can fetch any log with any one disk access given the LSN

- The log record is written for each of the following actions:-

1. Updating a page:- After modifying the page an update type record is appended to the log table
2. Commit:- When the transaction decides to commit it force rides a commit type log record containing the transaction ID
3. Abort:- When the transaction is aborted, an abort type log record containing the transaction ID is appended to the log and undo is executed of this transaction
4. End:- As noted above when a transaction is aborted or committed some additional actions must be taken beyond writing the abort or commit log record. After all these additional steps are completing an end type record is appended to the log
5. Undoing an Update:- When a transaction is rolled back it's updates are undone

Recovering from a system crash

Analyze phase

- Analyzes phase begins examining the most begin checkpoint record
- The Analyze phase performs three tasks

1. It determines the point in the log at which to start the redo pass
2. It determines the pages the buffer tool that were dirty at the time of the crash
3. It identifies transactions that were active at the time of the crash and must be undone

Redo phase

- The Redo phase follows the Analyze phase and redos all the changes to any page that had been dirty at the time of the crash
- During the redo phase the Aries reapply the updates of all transactions committed or otherwise
- The action must be redone unless one of the following the conditions holds

1. The affected page is not in the dirty page table:- The first update to the page may not have been returned to the disk
2. The affected page is the dirty page table but the LSL for the entry is greater than the LSN of the log record being checked:- [I don't know what is going on]
3. The page LSN is greater than or equal to the LSN of the log record being checked:- [At this point I didn't care for shit]

Undo phase

- The undo phase unlike the other 2 phases scans backward from the end of the log
- The goal of this phase to undo the actions of all the transactions after the time of the crash
- That is too effectively abort them, this set of transactions is identified in the transaction table constructed by the Anaylze's phase