DSP Unit 3

Database Application Security Models: Introduction:

(*) Database Application Security
→ Designed to protect data stored in databases.
→ Ensuring security of dB is crucial as they often contain sensitive da

(1) Authentication: Verifying the identity of the user.

(2) Authorization: Defines what actions are allowed to perform with a database.

(3) Data Encryption: Involves transforming the data into a secure, unreadable format that can only be read by appropriate encryption key

(4) Auditing and logging: Recording and Monitoring activities within the database to identify security breaches.

(5) Role based access Contron (RBAC): Assigns permissions based on role

(6) Security Policies: defines the rules, standards and procedures that guide the security of database and applications.

(7) Intrusion detection and Prevention: Used to monitor and respond to potential security threats and breaches.

→ Database Application Security Models are essential for safeguarding sensitive data and ensuring the integrity, confidentiality and availability of information

→ These models provide a structured framework for implementing security controls, monitoring, etc.

(*) Types of Users:

(1) Application Administrator: Has special privileges to manage other users and their roles within the application. They don't need direct access to database.
Ensures that only authorized users can access and perform tasks within the application, enhancing security.

(2) Application Owner: Owns tables and objects used by the application. They have control over the application's core

components.

③. Application User: Regular users that perform tasks within the application. They interact with the application but may not have direct access to the database.

④. DBA (Database Administrator): Have administrative powers over the entire database systems. They handle database maintenance and ensure overall performance and security. Guardians of entire database.

⑤. Database User: These ~~are~~ are user accounts with specific permissions to access and manipulate data within the database. They can perform tasks based on their assigned roles.

⑥. Proxy user: Works on behalf of application user. Often serve as intermediaries to access the database without revealing the application user's credentials.

⑦. Schema Owner: Who owns the database objects like ~~tables~~ and views. Have control over how the data is structured. Ensures that the data is organized properly.

⑧. Virtual User: An account that accesses the database through another user. Often called a proxy user. Allow controlled access to the database without directly revealing the user's identity, adding a level of security.

(*). Security Models:

→ There are two Security Models:

(a). Access Matrix Model (b). Access Modes Model.

→ Access Matrix Model:

(*) It is a conceptual framework that represents the access control relationships between subjects (users or processes) and objects (resources or data).

(*) Often visualized as a matrix where rows represent subjects and columnc represent objects. Each cell specifies the permissions or access rights.

(*) Subjects: Entities that seek access to resources. Eg: Users, proceses

(*) Objects: Resources/Data that the subject wants to access.

(*) Access Rights/Permissions: What specific actions can a subject perform on an object. Eg. read, write, execute, delete, etc.

(*) Access Matrix Model allows for precise and detailed control over access rights.

(*) It is easy to visualize and understand.

(*) Managing a large access matrix can become complex as the no. of subjects and objects increase.

(*) Adapting the matrix to dynamic changes can be challenging.

← Objects →

Subjects

	File 1	File 2	File 3	File 4
User A	Own Read Write		Own Read Write	
User B	Read	Own Read Write	Write	Read
User C	Read Write	Read		Own Read Write

Access Rights

→ Access Modes Model: Simplifies access control by categorizing access rights into predefined modes or levels of access.

(*) Instead of specifying permissions for each user-object pair, users are assigned to access modes that grant a certain level of access to specific resources.

(*) Access Modes: Predefined categories of access rights such as "Read-Only", "Read-Write", "No Access", etc.

(*) User Roles / Groups: Users are assigned to roles or groups that have specific access modes associated with them.

(*) Resource Types: Data is classified into types. Each type is associated with one or more access modes.

(*) Access modes reduce the complexity of managing access control.

(*) Easier to implement and maintain especially in large systems.

(*) Access modes might not offer the same level of granularity.

(*) It has limited flexibility.

Access Mode	Level	Description.
Use	1	Allows subject to only access object.
Read	2	Allows subject to read the content.
Update	3	Allows subject to modify the content
Create	4	Allows to add instance to object.
Delete.	4	Allows S to remove Instance to the O.

(*) Application Types:

(a). Mainframe applications

(b). Client / Server Applications

(c). Web Applications

(d). Data warehouse Applications.

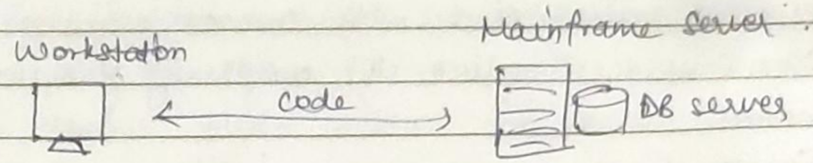
(*) Mainframe Applications:

→ large, Powerful computer systems used by organizations

Date: Page:

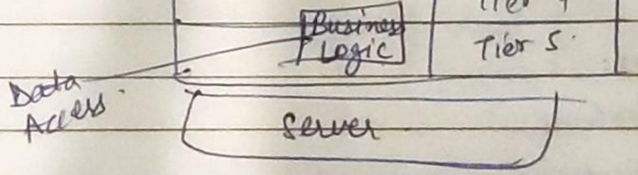
to manage critical business functions.

- These applications often store vast amount of data is centralized databases.
- They require robust security measures to protect sensitive data.
- MIS department is responsible for all information



(*) Client - Server Applications:

- Introduced to overcome limitations in MIS department.
- Involves multiple computers(clients) connected to a central server.
- These applications distribute the task between the client and the server.
- It is flexible and scalable
- Minimum 2 tier configuration and maximum 4-5 tiers.



- The data access component is responsible for retrieving and manipulating data.

(*) Web Applications:

- They run on web servers and are accessed through web browsers.

- They interact with databases to provide dynamic content and services on the Internet
- They use HTTP protocol to communicate to the server

Client

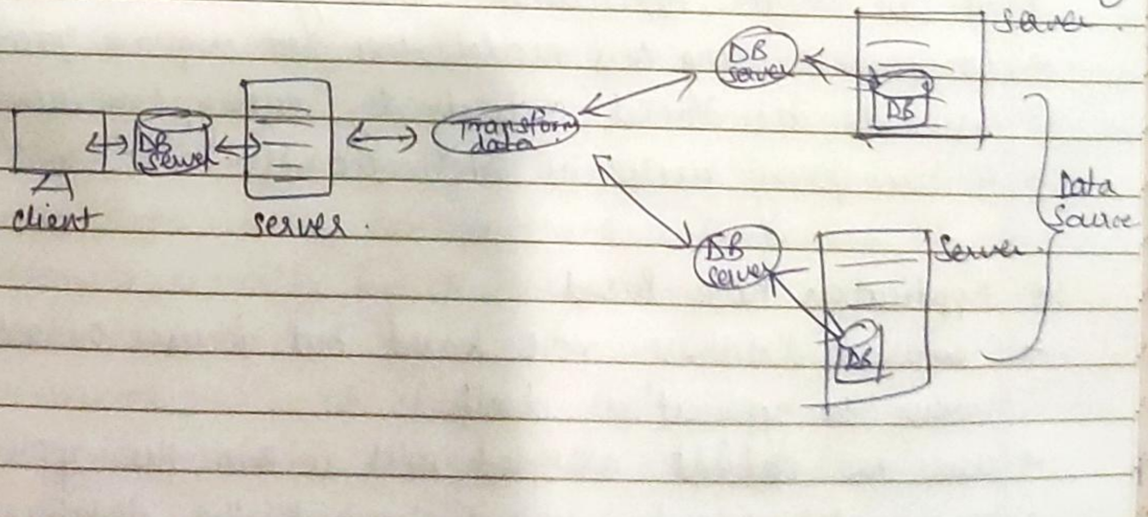
allows to navigate through web pages. ←	Web Browser layer	Tier 1
s/w program residing on computer connected to Internet ←	Web server layer	Tier 2
used for data processing. ←	Application Server layer	Tier 3
Implements Business rules ←	Business logic layer	Tier 4
stores & manages data. ←	Database server layer	Tier 5

Server.

- Each layer resides on a seperate computer.

(*). Data Warehouse Applications:

- Consolidate data from various sources for analytical purposes.
- They store large volumes of data and provide tools for data analysis and reporting.
- The Data warehouse is accessed by software applications or reporting applications called OLAP(OnLine Analytical Processing)

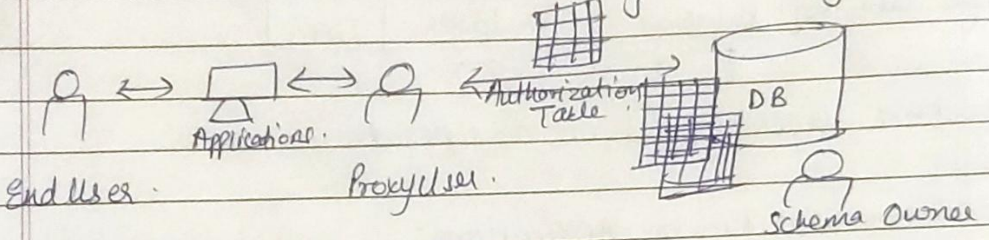


(*). Application Security Models:
- Database Role Based
- Application Role Based
- Application function Based.
- Application Role and Function Based
- Application Table Based.

(*). Database Role Based:

→ Security is managed based on roles assigned to the users within the database.

→ The users can access whatever privileges are assigned to a role.



→ This model heavily relies on DB role functionality.

→ It is database independent.

→ Proper implementation of roles is crucial. If not implemented correctly, it can lead to security issues.

→ It can isolate Application security from database.

→ Maintenance using this model does not require specific DB privileges.

→ Passwords are stored securely by encrypting them.

→ It uses proxy users as intermediaries.

(*). Application Role Based:

→ Similar to database role based but focuses on roles defined within the application itself.

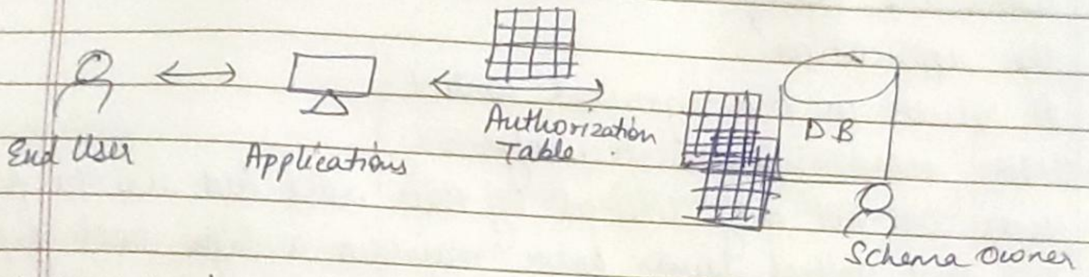
→ Users are assigned roles within the application.

→ This model extends control beyond the database.

→ It adds an extra layer of security to protect data.

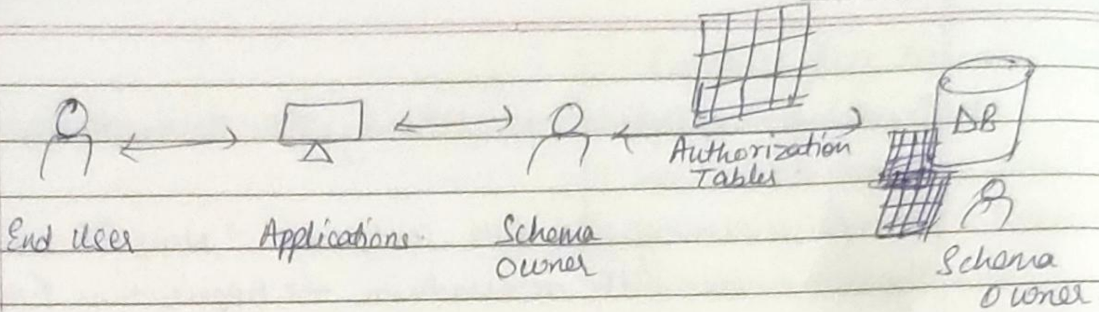
→ Creating application roles using SQL Server Enterprise Manager:
Enterprise Manager → Role container → New DB role → type the name db_accessadmin → Application Role → Enter password db@access → OK.

→ Dropping Roles:
Enterprise Manager → Expand Roles of Container → Select and Delete the desired role.



→ Model is primitive and does not allow flexibility required to make changes necessary for security
→ Limited privileges.
→ Only one role is assigned to an application user.
→ Passwords must be securely encrypted.

(*). Application function Based:
→ focuses on what users can do within the application.
→ Users are granted access to application functions or features
→ Doesn't directly control the database access, it impacts what data users can interact with through the application.
→ Enhances security by limiting users to specific functions and features.
→ Minimizes the risk of data exposure.
→ Only one role is assigned to an application user.
→ Passwords must be securely encrypted.
→ The application must be designed in granular module.



(*) Application Role and Functionality Based:

→ Combines both role based and function based security.

→ Users are assigned both roles and specific functions within the application.

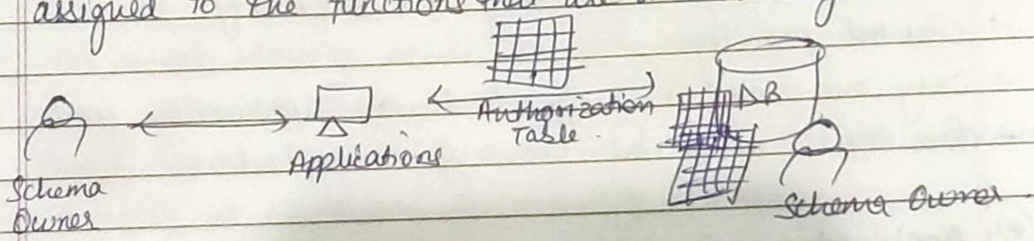
→ It results in fine-grained control.

→ Offers comprehensive security.

→ Users are not only limited by their roles but also the functions.

→ This dual control layer helps maintain a high level of privacy.

→ Applications are divided into functions and roles are assigned to the functions that are in turn assigned to the users.



→ Provides utmost flexibility.

→ Maintenance does not require specific privileges.

→ Password must be securely encrypted.

→ The application must be designed in granular fashion.

(*) Application Table based Security:

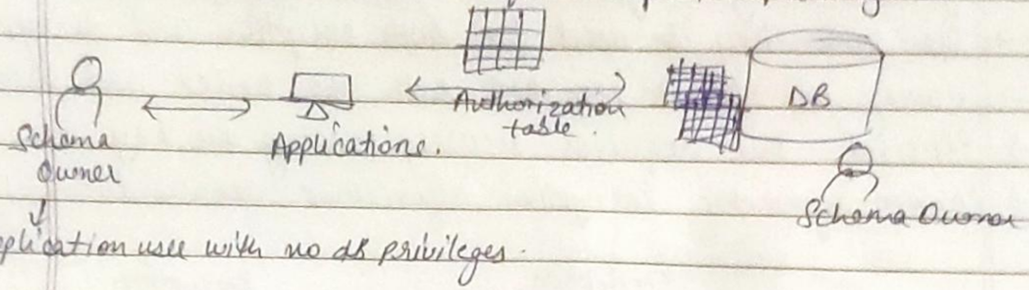
→ Focuses on securing data at table level within the database.

→ Users are granted permissions to specific database tables based on their roles or functions.

→ It allows fine-grained control over who can access, modify or view specific tables.

→ Effective for database privacy and security.

→ Maintenance does not require specific privileges.



Characteristics	Database Role based	Application Role based	Application function based	Role + func	Table
①. Flexible in ~~maintai~~ implementing application security	No	No	No	Yes	No
② Isolates application security from dB.	Yes	Yes	Yes	Yes	Yes.
③. Maintenance of Application does not require specific privileges	No	No	No	Yes	No.
④. Password must be securely encrypted.	Yes	Yes	Yes	Yes	Yes.
⑤ Uses real dB user to Logon	No	Yes	Yes	Yes	Yes.
⑥ Is business-function specific.	No	No	Yes	Yes	No.

(*). Data Encryption:

→ Putting information in a secret code that only authorized users can understand.

→ A way to protect data from unauthorized access.

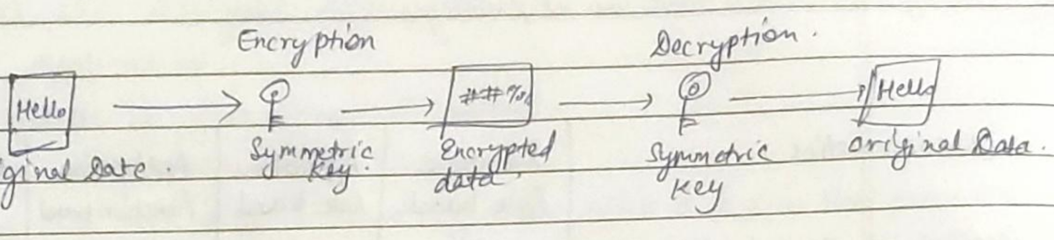
→ It uses an encryption algorithm to convert normal data into a secret code (cipher text).

→ To read this, you need to decrypt the data

Types: ①. Symmetric key Encryption
②. Public key Encryption.

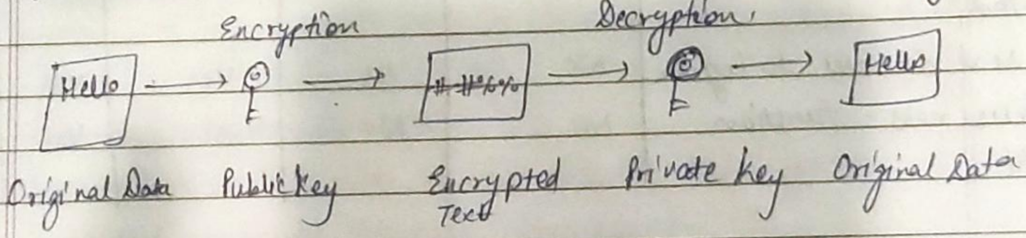
(*) Symmetric Key Encryption:

- → The same key is used for both encription and decryption.
- → Having one secret key that both the sender and receiver know.
- → Efficient but requires securely sharing the key.
- → Common symmetric Encryption algorithms are used.



(*). Public Key Encryption:

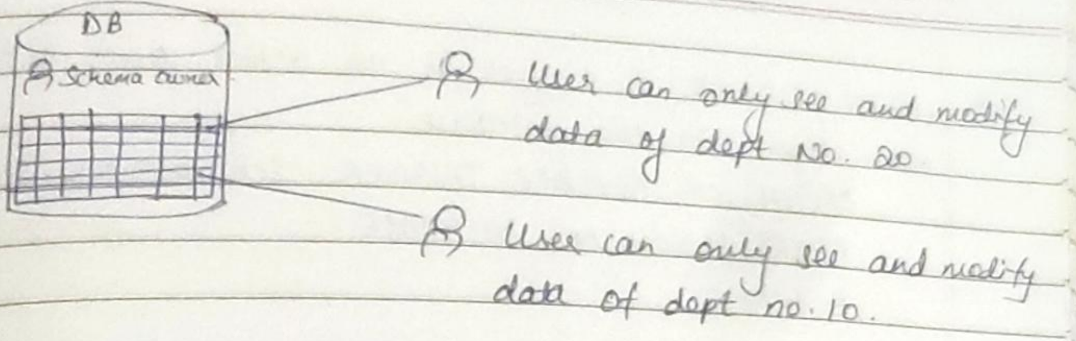
- → Also called asymmetric encryption.
- → There is a pair of keys: a public key and a private key.
- → The public key is used for encryption and the private key is used for decryption.
- → It is useful for secure communication and verifying identities.



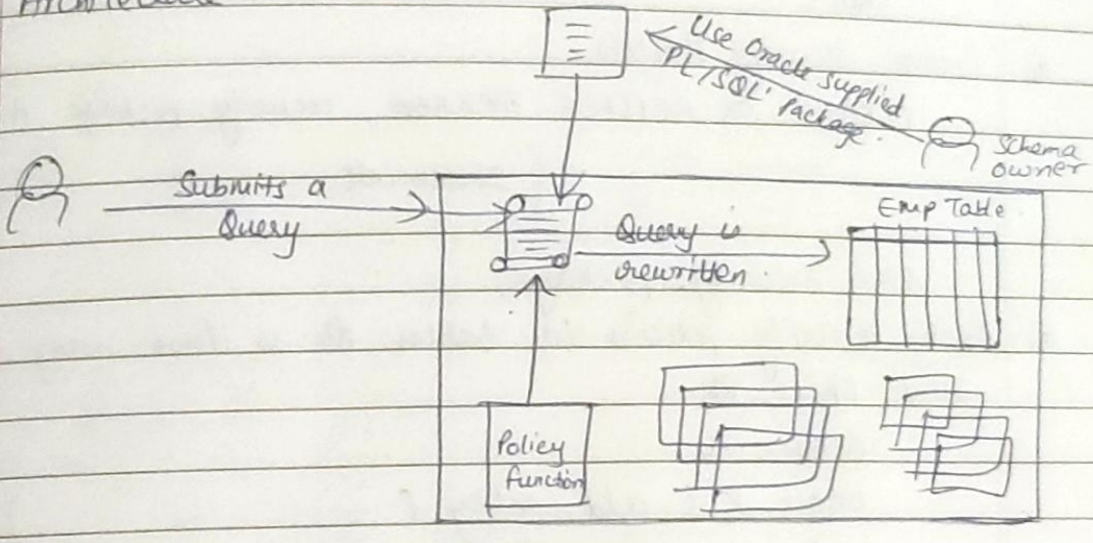
(*). Virtual Private Databases:

- → It is like having a shared database where multiple users can access and manipulate data but each user can see or work with their own data.
- → Oracle, a database system has a VPD feature.
- → Before Oracle 10G, they called VPD by two other names:

(a). Row level security (RLS) (b). Fine Grain Access (FGA).


DB
Schema owner
User can only see and modify data of dept No. 20
User can only see and modify data of dept no. 10.

Architecture:



(*). Implementing Oracle VPD:

①. Set up a Test Environment:

(a). Create user accounts.

```
CREATE USER username IDENTIFIED BY ______
GRANT read, write TO username;
```

(b). Create tables to store data and populate them with sample information.

```
CREATE TABLE users (id NUMBER(10) NOT NULL,
    owner VARCHAR, - - - - - )
INSERT INTO users values( - - - - );
```

②. Create an Application Context: This context helps to define the user's current identity.

```
GRANT CREATE ANY CONTEXT;
CREATE CONTEXT SCHEMAOWNER USING ______;
```

③. LOGIN TRIGGER: A trigger is an action that occurs when the user logs into the database.

```
    CREATE OR REPLACE TRIGGER schemaowner.set_security_context
    AFTER LOGON ON DATABASE
    BEGIN
      - - - - -
    END;
```

④. Create Security Policies:

```
    CREATE OR REPLACE PACKAGE security_package AS
        ——  ——   } statements
        ——  ——
    END security_package;
```

⑤. Apply security policies to tables: It is done using a package called DBMS_RLS.

```
    BEGIN
    DBMS_RLS.add_policy ( ——  —  — )
      -  -  ,   } statements.
      -  —  —
    END;
```

⑥. Testing VPD: You can now test if the VPD is working correctly by connecting as different users and trying to access and manipulate data.

→ Column level security: In SQL server, you can specify permissions at the column level.

You can control who can access specific columns in a table.

```
    GRANT SELECT ON table_name(column1, column2) TO user1;
    GO;
    DENY SELECT ON table_name (column3) TO user1;
    GO;
```