Date: Page: Notebook

DSP Unit 4

Auditing Database Activities:

→ Many Private and public institutions / Organizations are taking serious actions against security risks.
→ Auditing enables users to trace changes made to sensitive data.
→ It is the responsibility of developers, DBA & Business Managers.
→ It includes:
(a) Auditor: A Qualified Person authorized to validate activities & produce an audit report
(b) Audit Procedures: Step-by-step instructions for audit process
(c) Audit Report: A document containing audit findings.
(d) Audit Trial: A record of document changes, data changes, system activities, etc.
(e) Data Order: A record of data changes.
(f) Database Auditing: A record of database activities including Shutdown, Startup, logons, data structure changes, etc.
→ Types of Auditing:
(a) Internal Auditing: Conducted by staff members within the organization.
(b) External Auditing: Performed by individuals or teams outside the organization, often appointed by the government.
→ Automatic Audits are the ones used for system and database monitoring, where activities are tracked automatically.
→ Manual Audits are the ones in which human intervention is involved. Tasks include interviews, document reviews, etc.
→ Challenges: It can lead to performance issues, generation of numerous reports, disruptions to normal operations.
→ Auditing Models include:
① Simple Auditing Models
② Historical data Models.
③ Advanced Auditing models
④ Auditing Application Access Models.

→ C2 Security is a specialized security rating used in govt. & military organizations. It is used to protect sensitive data and information.

(v). Using Oracle Database Activities:

① Application Activities:
→ Involves SQL statements issued against application tables within the Oracle database.
→ Can include SELECT, INSERT, UPDATE, DELETE statements.
→ Typically performed by end users or application.
→ Auditing them is crucial for tracking data changes ensuring data integrity.

② Administration Activities:
→ Includes commands/statements issued by Database Administrators or operators for maintenance & administration process.
→ SQL + Administrative commands.
→ Creating & Managing users, configuring settings, backups, restores, etc.
→ Vital for maintaining the overall health & security of the DB.

③ DataBase Events:
→ Specific occurances that take place in the Oracle DB.
→ Includes actions like user login and logout, DB startup & shutdown, generation of errors, etc.
→ Helps identify security issues or breaches.

(vi) Creating DLL Triggers with Oracle:
→ Trigger is an event driven program executed automatically based on occurance of event.
→ Primary use of Trigger is to perform Audits.
→ Other uses include preventing invalid data from being inserted into tables; Implementing business rules, Generating values for

columns, etc.

SYNTAX: (Row level Trigger).

> On statement level trigger, there is no FOR EACH ROW statement & no conditional clause.

```
CREATE [OR REPLACE] TRIGGER <trigger_name>
[BEFORE | AFTER | INSTEAD OF]   {trigger timing}
[INSERT | UPDATE | DELETE]      {trigger Event}
ON <name of object>
[FOR EACH ROW]                  {Row Level}
WHEN <condition for trigger to execute>  {Conditional Clause}
DECLARE <declaration part>
BEGIN <Execution part>
EXCEPTION <exception handling part>  {Error Handling Mechanism}
END;
```

(x). Oracle Trigger Execution:

→ Trigger can be in either of 2 distinct modes:

(a). Enabled: Executes when a triggering statement is issued and the trigger restriction evaluates to TRUE.

(b). Disabled Trigger: Does not execute its trigger action at all.

→ for enabled triggers, ORACLE automatically:

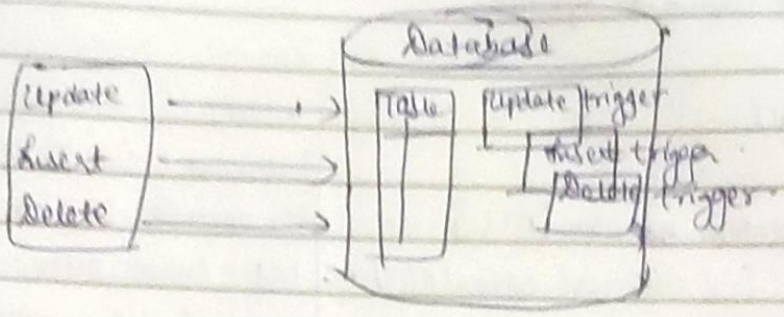
(a). Executes triggers of each type in a planned sequence when more than one trigger is fired by a SQL statement.

(b). performs integrity check

(c). Provides read-consistent views.

(d). Manages dependencies among triggers.

(e). If more than one trigger of same type exist for a given statement exists, Oracle fires each of those in unspecified order



(X). Auditing Database Activities with Oracle:

→ Provides a robust mechanism for auditing various aspects of database activities.

→ Required for enhancing database security, monitoring user actions, etc.

→ Allows the organization to track and record everything virtually.

→ Provides mechanisms for everything including:
(a) tracking who is creating and modifying the structure.
(b). who is granting privileges to whom.

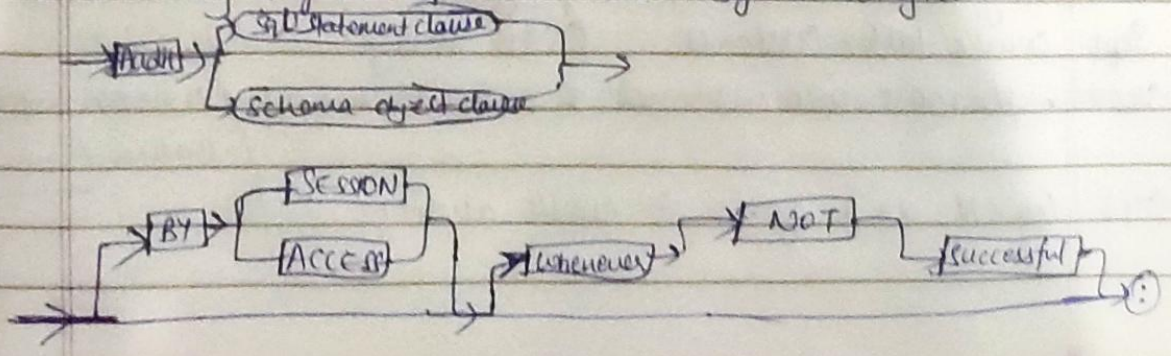
→ The activities are divided into 2 types based on the type of SQL Command statements used:
(a). Activities defined by DDL (Data Definition language)
(b). Activities defined by DCL (Data control language).

(X). Auditing DDL activities:

→ Oracle uses SQL based AUDIT command.

→ Suppose you want to know when a specific user issued ALTER comment or when a specific object was altered, you can get this information if AUDIT is enabled.

→ Following figure represents AUDIT syntax Diagram:



AUDIT Command Syntax:

```
AUDIT
{
{ { statement_option | ALL }
    [, { statement_option | ALL }]...
  [, { system privilege | ALL Privileges
    } }
  [BY { proxy [, proxy]...
      | user [, user]...
      ]
  { object_option [, object_option]... | ALL
    }
  ON { [schema.]object
     | DIRECTORY directory_name
     | DEFAULT
          } }
  [BY { SESSION | ACCESS } ]
  [WHENEVER [NOT] SUCCESSFUL];
```

where,

Statement option: Tells oracle to audit the specified DDL or DCL statement
DDL - Create, Alter, Drop & Truncate.
DCL - Grant, revoke.

System Privilege: Audit the specified privilege (SELECT, CREATEANY, ALTERANY)

object_option: Specifies the types of privileges for specified object.

BY SESSION: Tells oracle to record data once per session even if the audit statement is issued multiple times.

BY ACCESS: Tells oracle to record audit data everytime audited statement is used.

Whenever Successful: Tells oracle to capture Audit data only when command is successful.

Whenever NOT Successful: Capture audit data only when the command fails.

→ Example: Suppose you want to audit a table named customer every time it is altered or everytime a record is deleted;

Step 1 Create table Customer. (SQL Query)

Step 2 Add rows into customer & commit changes (INSERT into Customer values (____);) (write query)

Step 3 Log ON as a system to enable auditing.

SQL > CONNECT SYSTEM @ SEC.
Enter Password : ****
Connected.
SQL > AUDIT ALTER ON DBSEC.CUSTOMER BY ACCESS WHENEVER SUCCESSFUL;
Audit Succeed.
SQL > AUDIT DELETE ON DBSEC.CUSTOMER by Access Whenever Successful;
Audit Succeed;

Step4: Login as the owner of the customer table, delete a row and modify the table. [write queries on your own].
Step5: Login as SYSTEM and when we view DBA_AUDIT_TRIAL, two records will be available (Delete & Modification).
→ When audit process got over, you can turn it off using NO AUDIT statement.

(*) DCL Activities example:
Step1: Logon as SYSTEM or SYS and issue an audit statement.
SQL > CONN SYSTEM
Enter Password : *****
SQL > AUDIT GRANT ON DBSEC.TEMP;
Audit Succeeded.
Step2: Logon and update Privileges on TEMP table.
SQL > GRANT SELECT ON TEMP TO SYSTEM;
SQL > GRANT UPDATE ON TEMP TO SYSTEM;
Step3: Logon as System and display the content of DBA_AUDIT_TRIAL.

(*) Auditing Server Activity with SQL Server 2000:

→ Microsoft SQL server 2000 provides auditing as a way to track and log activity for each SQL Server occurence.

→ User must be a member of the sysadmin fixed server role to enable or modify auditing.

→ Every modification of an audit is an auditable event.

→ 2 types of Auditing in SQL server 2000 (a). Auditing
(b). C2 Auditing.

→ Auditing can have significant impact on performance.

→ Audit trial analysis can also be costly.

→ One of the tools that accompanies SQL server 2000 is SQL Profiler.

→ SQL Profiler provides the user interface for auditing events.

→ You can audit several types of events using SQL Profiler.

→ Some of the events are:

① End user Event: All SQL commands, LOGIN, LOGOUT, enabling

② DBA events. DDL, Configuration, etc.

③ Security event: GRANT / REVOKE / DENY / LOGIN USER ROLE / ADD / REMOVE / CONFIGURE.

④ Utility events: BACKUP / RESTORE / BULK INSERT, etc.

⑤ Server event: SHUTDOWN / PAUSE / START.

⑥ Audit events: ADD Audit / Modify Audit / Stop Audit.

(*) Security and Auditing:

→ for each event, you can audit: Date and Time of event, user who caused the event to occur, type of event, success or failure of the event, origin of request, name of object accessed. Text of SQL statements (Passwords are replaced with ******).

→ Security audit should be enabled first.

→ This is done by setting the security auditing level.

→ Security events can be audited on success, failure or both.

(8) Security and Auditing:

→ Steps for Security Audit:

① Open Enterprise manager. Expand the appropriate SQL Server Group.
② Right click on desired server. ③ Click Properties.
④ On security tab, select the desired security level.
→ After audit level is set, you can use the SQL Profiler to monitor activity. (security events).
→ following events can be audited:

① Add DB User
② LOGIN / LOGOUT
③ BACKUP / RESTORE
④ LOGIN Change Password
⑤ LOGIN Change Property
⑥ LOGIN failed, etc.

→ To start a new trace: File → New → Trace
New Trace dialog box appears, you need to provide:
① Name for the trace, ② Server you want to audit,
③ Base template to start with ④ Where to save audit data.
⑤ A stop time, if you don't want the trace to run indefinitely.
→ On the events tab, specify events to be audited and in which category they belong.
→ Add login change password security event to the trace by the following steps:

① Expand Security Audit.
② Click Audit Login Change Password Event
③ Click Add button.

→ To Audit DDL Statements, on the events tab of your trace select Object: Created and Object: Deleted under objects category. These two events audit all Create and Drop statements.
→ Database Auditing: select events under database category.
→ Database Error Auditing: Select events under the Errors and warnings category on the events tab of your trace.

(*) Auditing Server activity with Oracle:

① Database Auditing:

→ Oracle DB provides a comprehensive auditing framework that enables organizations to monitor server activity.

→ Auditing user actions, database operations & system events.

→ Ensures that unauthorized or malicious activities are detected and addressed.

② Enable Auditing:

→ To audit server activity, you must first enable auditing within your Oracle Database.

→ It involves configuring parameters & specifying what kind of activities you want to track.

→ Auditing can be enabled at different levels including Database level, schema level, object level, etc.

③ Types of Auditing:

(a) Standard Auditing: tracks common activities like SQL statements, system privileges, user logon / logoff, etc.

(b) Fine Grained Auditing: Allows you to audit specific data within the tables. You can define policies that focus on specific rows & columns.

(c) Unified Auditing: Consolidates all audit trials into single location. Offers enhanced security.

④ When auditing server activities, consider the following aspects:

① User Logon/logoff: Monitoring who logs into the DB & when they log off.

② SQL Statements: Helps in tracking data manipulation. Provides a detailed view of database transactions.

③ Privilege Usage: To ensure users are not abusing their access rights.

④ Data Changes: Essential for maintaining data integrity

⑤ Server Events: Helps in detecting any abnormal activities.

⑤ Privacy & Compliance:
→ Auditing server activity is critical for maintaining data privacy.
→ Allows organizations to protect sensitive information.
⑥ Auditing Policies:
→ Define clear audit policy that specify what should be audited, how long audit records should be retained and who has access.
→ This ensures that auditing aligns with privacy & security objectives.

Security and Audit: Project Case Study:

Introduction: A DB developer is assigned to new DB application project and is asked to develop an auditing scheme to comply with industry standards.
→ This case study involves various scenarios where a developer or consultant is tasked with implementing security and auditing measures for different database applications. The key points for each case are as follows:

Case 1: Developing an Online Database.
→ A new website for database enthusiasts is being created with various features.
→ Security requirements include 10 public host database accounts, password reset on logoff, session duration limit, memory and CPU allocations, limited storage & privileges for common database objects.
→ All newly created database objects must be removed after logoff.
→ Auditing requirements involve recording session information for analysis.
→ Maximum duration for a session is 45 mins.

→ Storage for each account must be limited to 1 Mb.

Case 2: Taking Care of Payroll:

→ Needs a virtual Private database to ensure client data privacy.

→ The virtual Private Database feature should allow each client to manage their payroll data without violating others privacy.

Case 3: Tracking Town Contracts.

→ A small town has hired you as a database specialist on contract.

→ Your job is to develop a new DB application to keep track of the jobs awarded to different contractors.

→ All town hall employees will use the application.

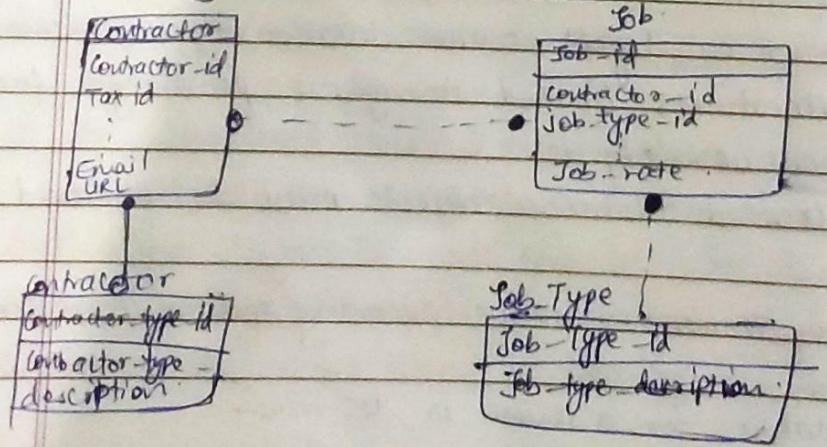
→ The application should:
track all the changes made to application data, obtain the approval of project manager before accepting any contract for more than $10,000, alert project manager whenever an awarded job is modified to a value greater than $10,000.

→ Implement 3 levels of security:

(a) Department Clerk level (allows clerks to add & update records).

(b) Department Manager level (allows clerk to add, update, delete & approve)

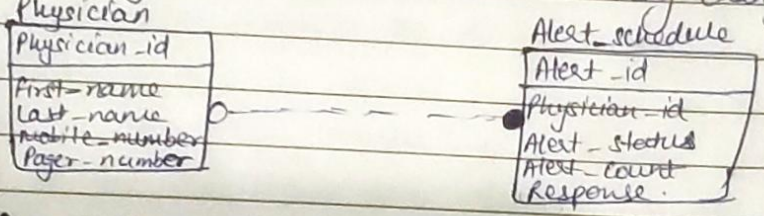
(c) External Clerk level (allows employees outside the department to only view data).



Contractor Job data Model.

Case 4: Tracking database changes:

→ A friend recommended you to the company they work for.
→ They need your help to solve a series of database violations.
→ The company wants to know the following:
Who accessed the DB?, who modified the data?,
Who changed the data structure?
→ Requirements include auditing database connections, tracking users performing DML operations & Monitoring changes to Data Structure.



Case 5: Developing a secured Authorization repository:

→ Main requirement is to create a security data model that will be used for by the central authorization Module.
→ This model should include an auditing repository.
→ The model will store Application users, Roles, Applications, Application Modules.
→ Security requirements involve user accounts, application roles, user access to application modules, security levels, password storage & account activation / expiry.
→ Auditing requirements include tracking user logins, application operations, security module activities & coupling the auditing module with the security module.

In all the cases, the focus is on ensuring data security, privacy and auditing in compliance with the industry standards. The provided solutions are for design purpose only and not for implementation.