UNIT - IV

AUDITING DATABASE ACTIVITIES

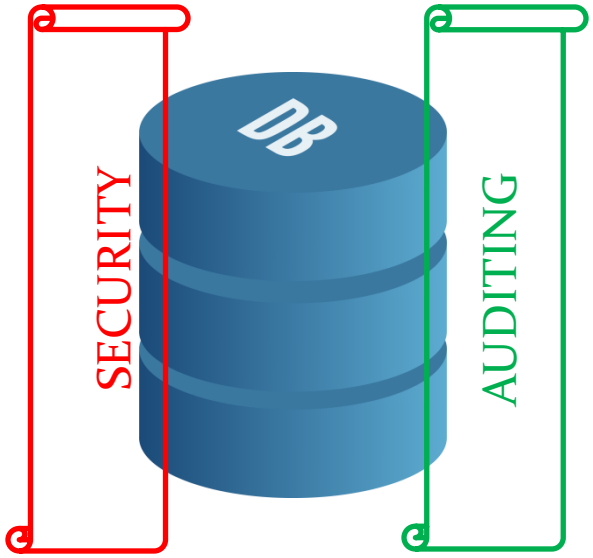
- ✓ Introduction
- ✓ Using Oracle Database Activities
- ✓ Creating DLL Triggers with Oracle
- ✓ Auditing Database Activities with Oracle Auditing
- ✓ Server Activity with SQL Server 2000
- ✓ Security and Auditing Project Case Study

Introduction

- ✓ Security is the buzzword of this decade
- ✓ It's on everyone's mind
- ✓ Today , crime brings to a mind a whole new set of risks to privacy and confidentiality
- ✓ Security requires action
- ✓ Many private and public Institutions / Organizations are taking serious action against security risks
- ✓ These actions encompass not only the establishment and enforcement of new security measure, but also the reinforcement of those measures through tough audit controls

Introduction

- Auditing is the responsibility of developers, DBA, and Business Managers
- The auditing mechanism would enable users to trace changes to sensitive data
- As DBA , you might be summoned to yours manager's incident that left the DB is unavailable for hours.



Auditing Overview

Definitions

- ✓ In general, an audit examines the documentation that reflects the action, practices and conduct of business or individual.

Database auditing follows this general definitions

- ✓ The list that follows contains general auditing and database auditing definitions.
  - Audit / Auditing - The process of examining and validating documents, data, processes, systems, or other activities to ensure that the audited entity complies with its objective
  - Audit log – A document that contains all activities that are being audited ordered in a chronological manner.
  - Audit objectives – A set of business rules, system controls, government regulations or security policies against which the audited entity is measured to determine compliance

Auditing Overview

Definitions …

- Auditor – A person with proper qualifications and ethics, who is authorized to examine, verify, and validate documents, data, processes, systems, or activities and to produce an audit report
- Audit procedure – A step-by-step instructions for performing auditing process
- Audit report – A document that contains the audit findings and is generated by an individual(s) conducting the audit
- Audit trail – A chronological record of document changes, data changes, system activities, or operational events
- Data audit – A chronological record of data changes stored in a log file or a database table object
- Database auditing - A chronological record of database activities , such as shutdown, startup, logons, and data structure changes of database objects
- Internal auditing – Auditing activities conducted by the staff members of the organization.
- External auditing - Auditing activities conducted by the staff members outside of the organization.

Auditing Activities

✓ Auditing activities are performed as a part of an audit, audit process or audit plan

The following list presents the auditing activities

(Note : Activities are not listed in any specific order)

- Evaluate and apprise the effectiveness and adequacy of the audited entity according to the auditing objectives and procedures
- Ascertain(find) and review the reliability and integrity of the audited entity
- Ensures the organization being audited is in compliance with the policies, procedures, regulations, laws, and standards of the government and the industry.
- Establish plans , policies, and procedures for conducting audits.
- Keep abreast of all changes to the audited entity.
- Keep abreast of updates and new audit regulations, laws, standards, and policies set by industry, government, or the company itself.
- Provide all audit details to all company employee involved in the audit. These details include : resources requirements, audit plans, and audit schedules.

Auditing Activities…

- Publish audit guidelines and procedures to the company itself and its partners and clients when appropriate.
- Act as liaison between the company and the external audit team.
- Act as a consultant to architects, developers and business analysts to ensure that the company being audited is structured in accordance with the audited objectives
- Organize and conduct internal audits
- Ensure all the contractual items are met by the organization being audited.
- Identify the audit types that will be used
- Work jointly with the Security Department to identify security issues that must be addressed
- Provide consultation to the Legal Department to identify regulations and laws with the company must comply

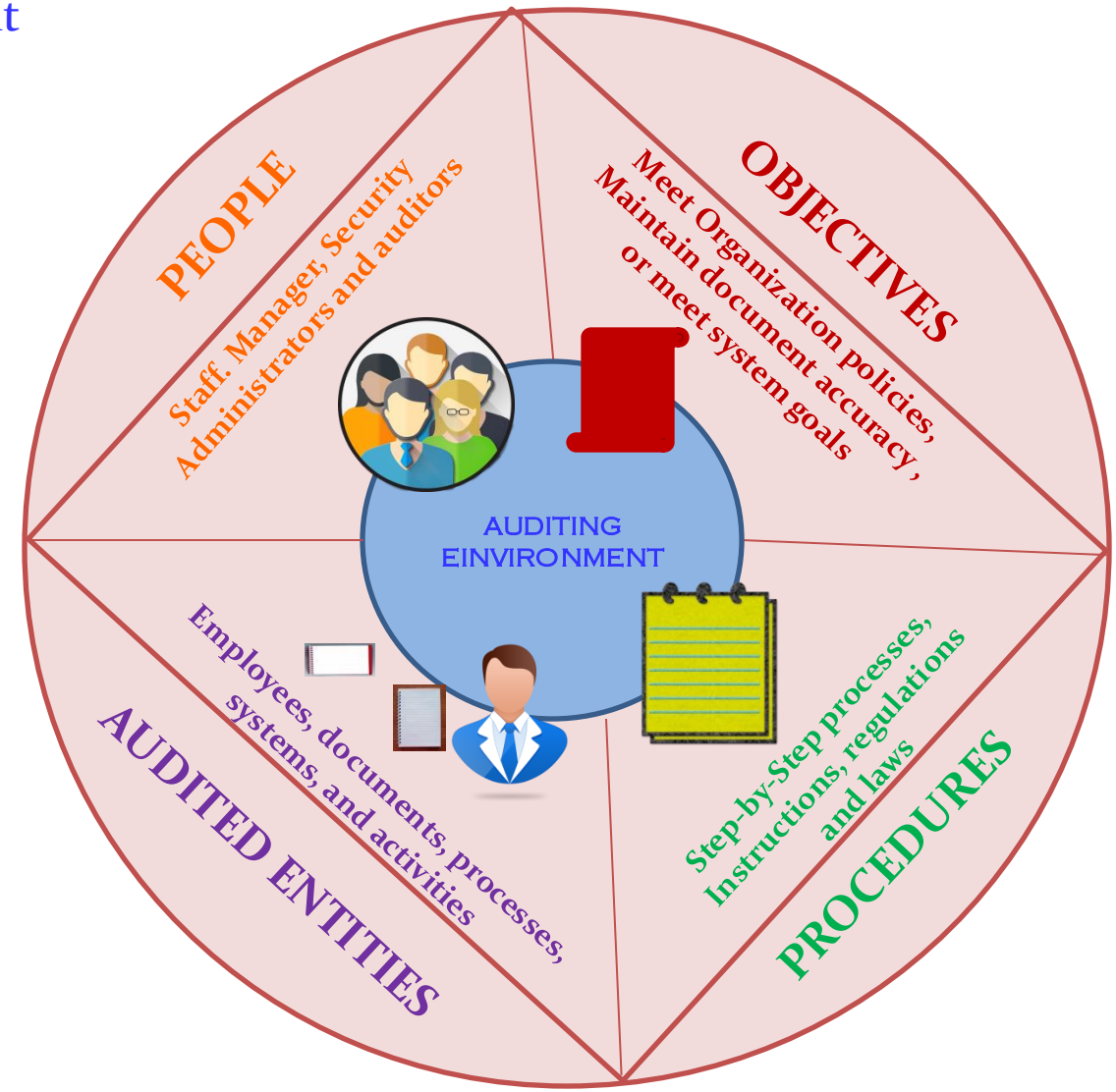
Auditing Environment

Components of Auditing Environment

- ✓ Objectives
  - An audit without objectives is useless
  - To conduct audit you must know what the audit you must know what the audited entity is to be measured
  - Usually , the objectives are set by the organization , industry standards, or government regulations and laws
- ✓ Procedures
  - To conduct an audit, step-by-step instructions and tasks must be documented ahead of time.
  - In the case of government conducted audit, all instructions are available public
  - In the case of organizational audit, specialized personal document the procedure to be used not only for the business itself, but also for the audit
- ✓ People
  - Every auditing environment must have an auditor , even in the case of automatic audit
  - Other people involved in the audit are employees, manager, and anyone being audited
- ✓ Audited entities
  - This includes people, documents, processes, systems, activities or any operation that are being audited

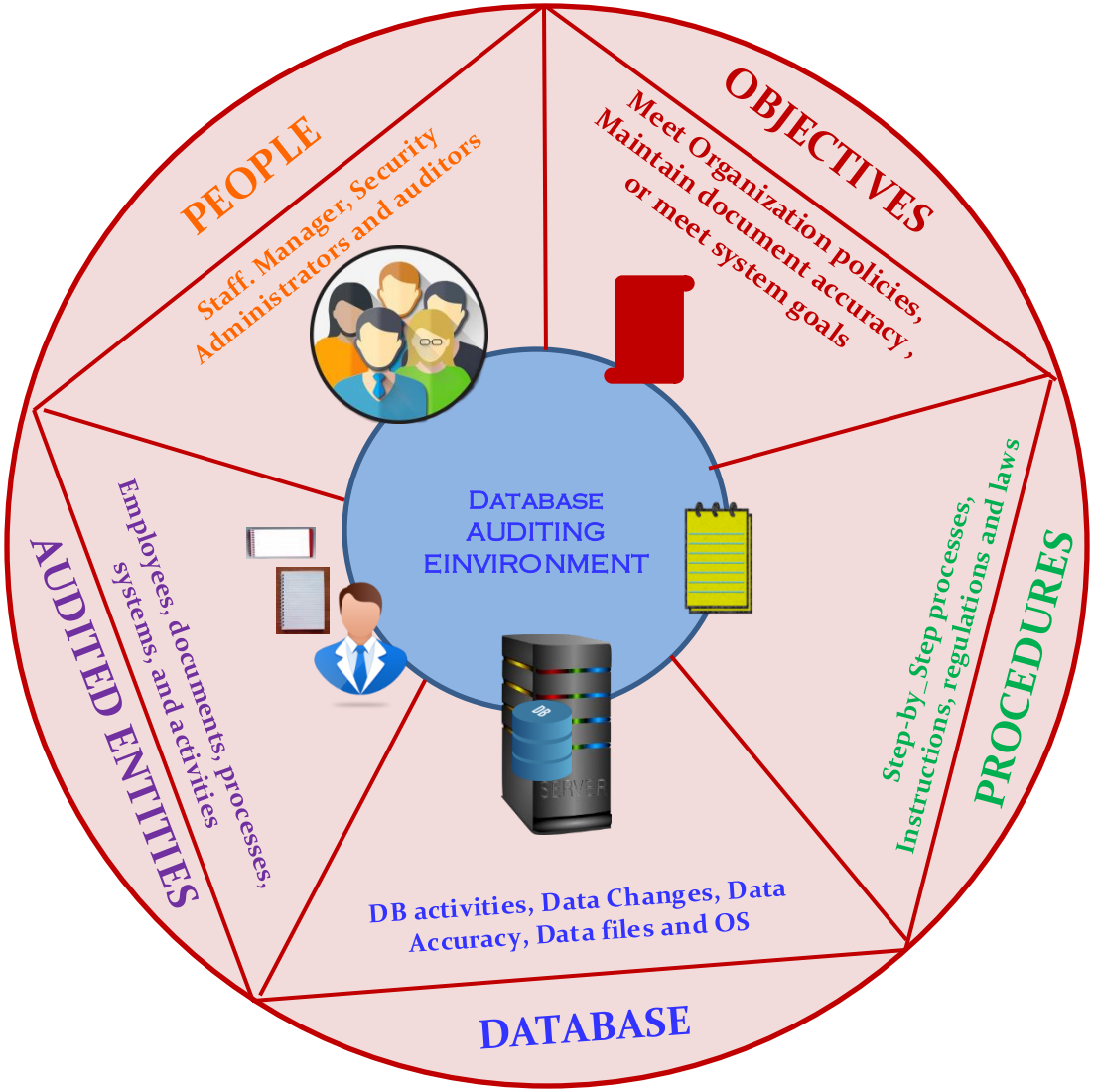
Auditing Environment ...

- ✓ The following figure shows the four major components of the auditing environment



Database Auditing Environment …

- ✓ The following figure shows the five major components of the auditing environment



Auditing Process

- ✓ Database applications widely used by major corporate companies, mostly large financial and online trading companies.
- ✓ The Quality Assurance (QA) team retested every database application function and try to find bugs.
- ✓ This type of auditing resembles QA or even performance monitoring
- ✓ The purpose of QA process in software engineering to make sure that the system is bug free and that the system is functioning according to its specification.
- ✓ The auditing process ensures that the system is working and complies with the policies, standards, regulations or laws set forth by organization, industry or government.

Auditing Process …

- ✓ Another way to distinguish between QA and Auditing Process is by examining the timing of each
- ✓ QA – **during development phase, before the implementation of the system.**
- ✓ Auditing Process – **After the system is implemented and in production**.
- ✓ Auditing is also not the same as performance monitoring
- ✓ Auditing objectives are totally different
- ✓ Performance Monitoring is to observe the degradation in performance
- ✓ Auditing validates compliance to policy not performance

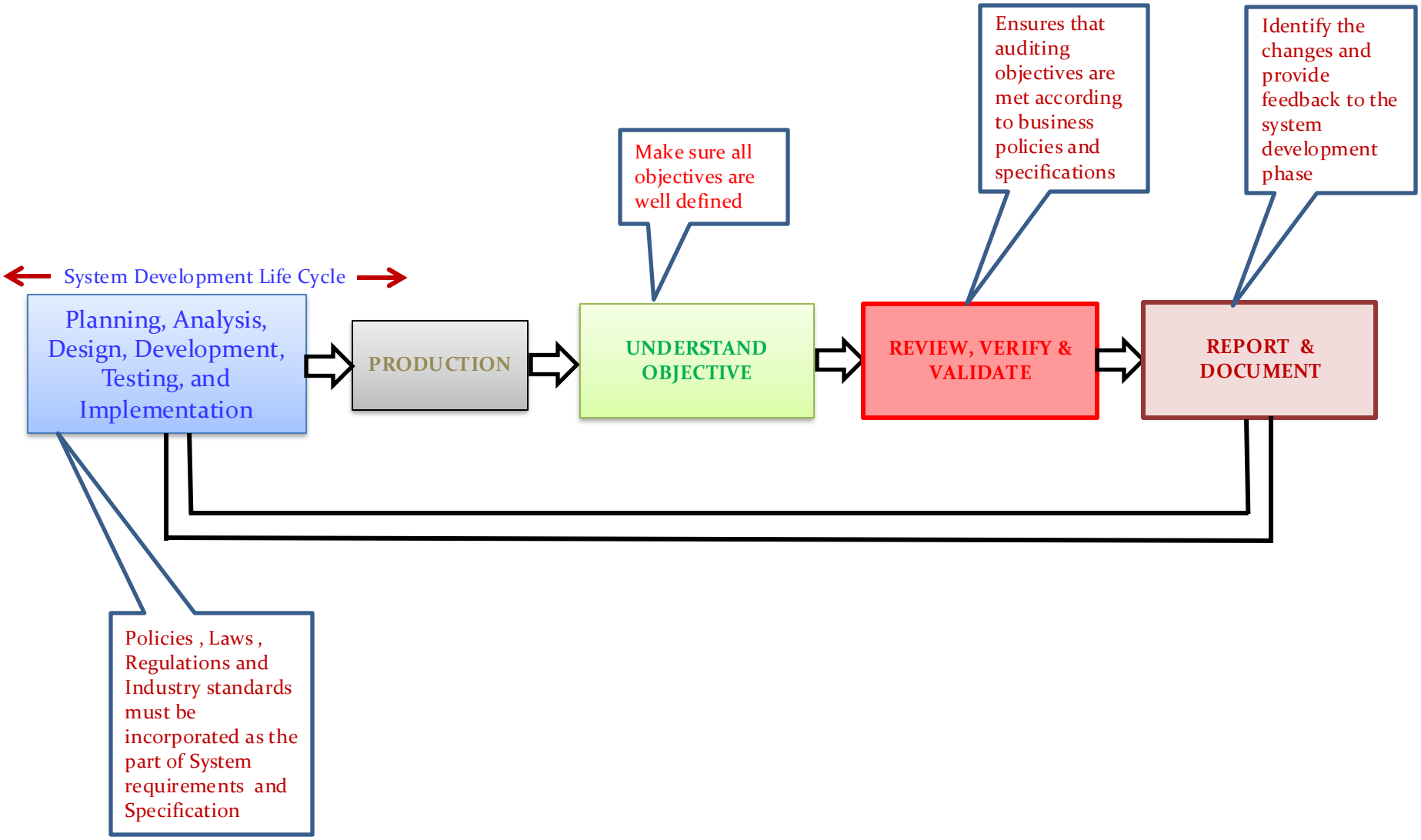
Auditing Process …

- ✓ Differences in QA , Auditing and Performance Monitoring processes

PROCESS	ACTIVE TIMING	OBJECTIVES
QA	During development and before the product commissioned into production	Test the product to make sure it is not working properly and is not defective
Auditing	After the product commissioned into production	Verify that the product or system is working and complies with the policies, standards, regulations or laws
Performance Monitoring	After the product commissioned into production	Monitor Performance in terms of Response time,

Auditing Process …

✓ The below figure illustrates the auditing process flow



Auditing Objectives

- ✓ Auditing objectives are established as a part of the development process of the entity to be audited
- ✓ For example , when a software application is being coded, the developers include in their software development design objectives the capability to audit the application
- ✓ Auditing objectives are established and documented for the following reasons:
  - Complying – Identify all company policies , government regulations, laws and the industry standards with which your company comply.
  - Informing – All policies, regulations, laws and standards must be published and communicated to all parties involved in the development and operation of the audited entity.
  - Planning – Knowing all the objectives enables the author to plan and document procedures to asses the audited entity.
  - Executing – Without auditing objectives, the person conducting the audit cannot evaluate, verify, or review the audited entity and cannot determine if the auditing objectives have been met

Auditing Objectives

- ✓ The top ten database auditing objectives

  - Data Integrity – Ensure that data is valid and in full referential integrity
  - Applications Users and roles – Ensures that users are assigned roles that correspond to their responsibilities and duties
  - Data Confidentiality – Identify who can read data and what data can be read
  - Access Control – Ensures that the application records times and duration when a user logs onto the database or application
  - Data changes – Create an audit trail of all data changes
  - Data Structure Changes – Ensures that the database logs all data structure changes
  - Database or application availability – Record the number of occurrences and duration of application or database shutdowns all the startup times . Also, record all reason for any unavailability.
  - Change Control – Ensure that a change control mechanism is incorporated to track necessary and planned changes to the database or application.
  - Physical Access – Record the physical access to the application or the database where the software and hardware resides.
  - Auditing Reports – Ensure that reports are generated on demand or automatically , showing all auditable activities

Auditing Classification and Types

Audit Classifications

- ✓ Every industry and business sector uses different classifications of audits.
- ✓ Definition of each classification can differ from business to business.
- ✓ Will discuss most generic definition of audit classifications.

Internal Audit

- ✓ An internal audit is an audit that is conducted by a staff member of the company being audited
- ✓ The purpose and intention of an internal audit is to :

  - Verify that all auditing objectives are met by conducting a well-planned and scheduled audit
  - Investigate a situation that was promoted by an internal event or incident. This audit is random , not planned or scheduled.

Auditing Classification and Types ...

External Audit

- ✓ An external audit is conducted by a party outside the company that is being audited.
- ✓ The purpose and intention of an External audit is to :

  - Investigate the financial or operational state of the company . This audit is initiated at will by the government or promoted by suspicious activities or accusations.
  - The person conducting this audit is usually employed and appointed by the government.
  - Verify that all objectives are met. This audit is typically planned and scheduled.
  - Ensure objectivity and accuracy.
  - This audit is typically performed to certify that the company is complying with standards and regulations.

Auditing Classification and Types …

- ✓ Automatic Audit
  - An automatic audit is promoted and performed automatically.
  - Automatic audits are mainly for systems and DB systems.
  - Some systems that employ this type of audit to generate reports and logs.
- ✓ Manual Audit
  - Completely performed by humans
  - The team uses various methods to collect audit data, including interviews, document reviews and observation.
  - The auditors may even perform the operational task of the audited entity.
- ✓ Hybrid Audit
  - Combination of Automatic and Manual Audits

Auditing Classification and Types ...

Audit Types

Financial Audit – Ensures that all financial transactions are accounted for an comply with law.

Ex : Companies save all trading transactions for a period of time to comply with government regulations

Security Audit – Evaluates if the system is as secure as it should be.

The audit identifies security gaps and vulnerabilities

Ex: Company might ask a hacker to break the company's network system to determine how secure or vulnerable the network is.

Compliance Audit – Verifies that the system complies with industry standards, government regulations, or partner and client policies

Ex: All pharmaceutical companies must keep paper trails of all research activities to comply with industry standards as well government regulations

Auditing Classification and Types ...

Operational Audit –Verifies if an operation is working according to the policies of the company

Ex: When a new hire starts work, the HR department provides ID Card, Sign disclosure , Confidentiality papers, tax forms , etc.,

Investigative Audit – Performed in response to an event, request, threat, or incident to verify the integrity of the system.

Ex: Employee might have committed a fraudulent activity

Product Audit – Performed to ensure that the product complies with industry standards. This audit sometimes confused with testing, but it should not be.

A product audit does not include auditing of its functionality but entails how it was produced and who worked on its development.

Preventive Audit – Performed to identify problems before they occur.

Ex: Company should conduct both random and routine audits to verify that the business operations are being performed according to specifications.

Benefits and Side Effects of Auditing

Benefits

- Enforces company policies, government regulations and laws
- Lowers the incidence of security violations
- Identifies the security gaps and vulnerabilities
- Provides an audit trail of activities
- Provides another means to observe and evaluate operations of the audited entity
- Provides the sense or state of security and confidence in the audited entity
- Identifies or removes doubts
- Makes the organisation being audited more accountable
- Develops controls that can be used for purposes other than auditing

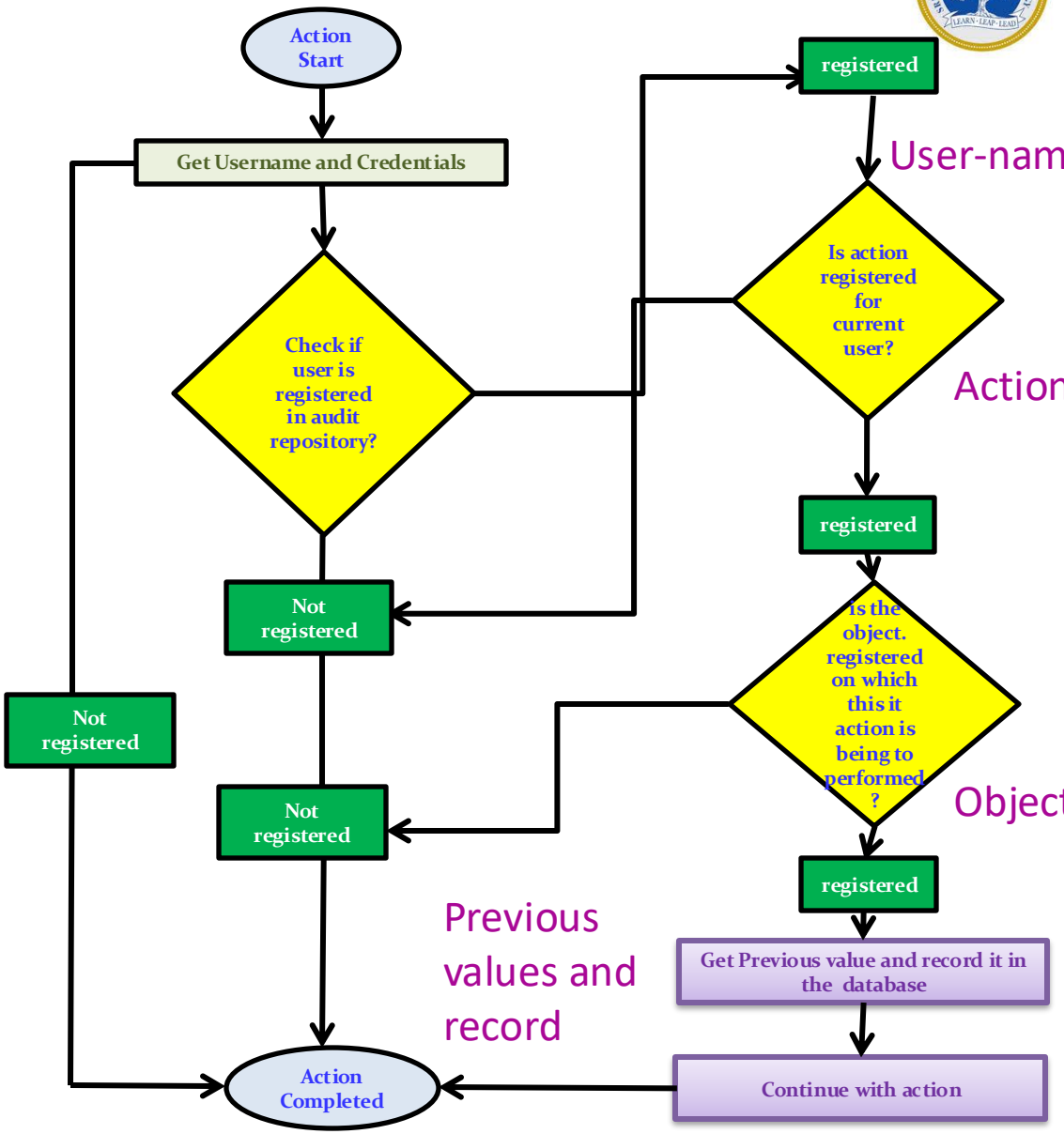
Benefits and Side Effects of Auditing

Side Effects of Auditing (Frequent audits can cause the following)

- Performance problems due to preoccupation with the audit instead of the normal work activities
- Generation of many reports and documents that may not be easily or quickly disseminated
- Disruption to the operations of the audited entity
- Consumption of resources, and added costs from downtime
- Friction between operators and auditor
- From a DB perspective
  - Could degrade the performance of the system
  - Also generate a massive number of logs, reports, and that require a system purge

Auditing Models

- ✓ Before auditing models, it is more important that , understand how audit is processed for data and DB activities
- ✓ The flowchart presents data auditing
- ✓ The flowchart shows what happens when a user perform an action to a DB object
- ✓ Specific checks occur to verify if the action , the user or the object are registered in auditing repository
- ✓ If they are registered the followings are recorded
  - State the object before the action was taken along with the time of action
  - Description of the action that was performed
  - Name of the user or userid who performed the action

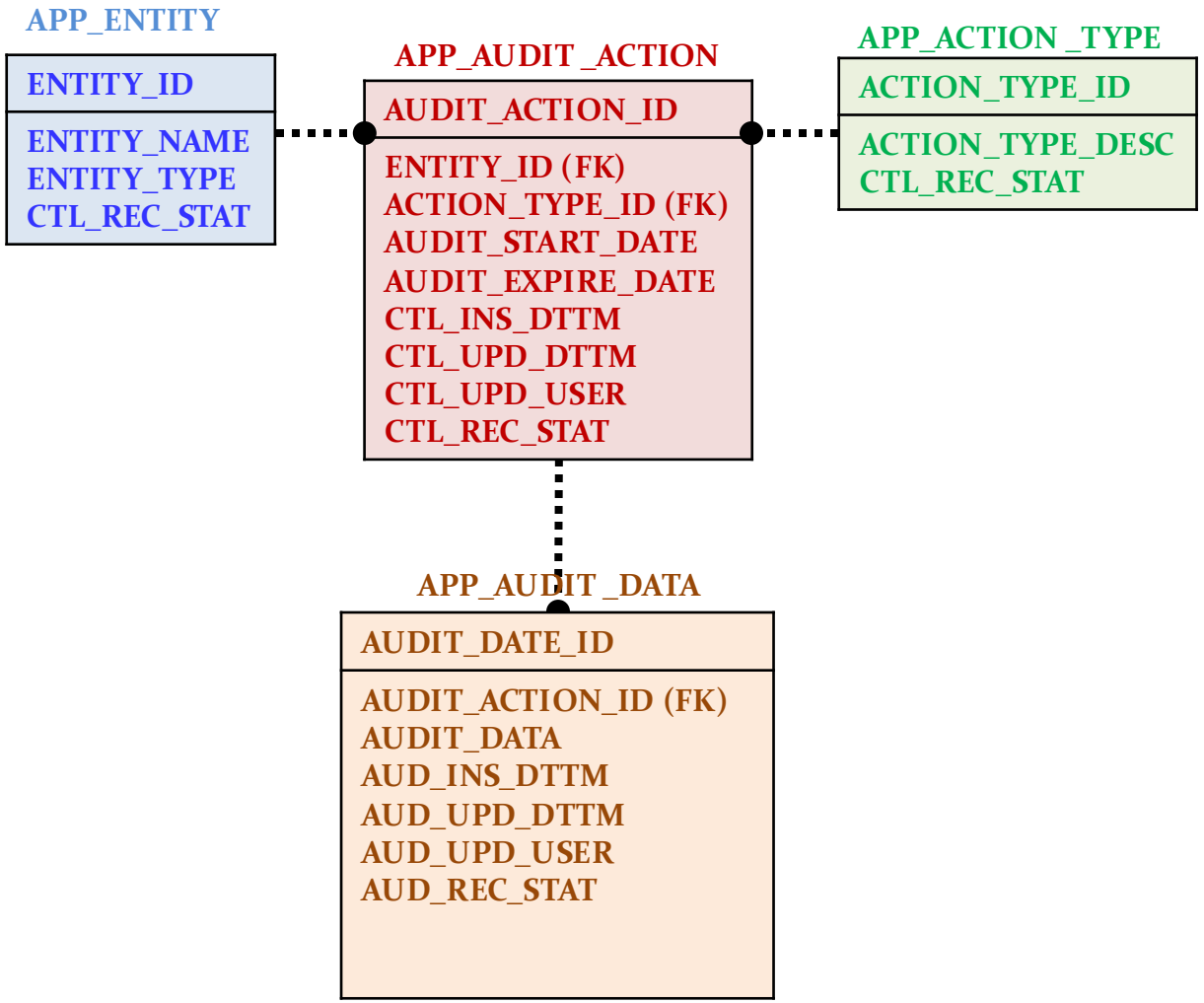


Auditing Models ...

Simple Auditing Model 1

- ✓ The first auditing model is called 'SIMPLE" because it is easy to understand and develop.
- ✓ This model registers audited entities in the audit model repository to chronologically track activities performed on or by these entities.
- ✓ An entity can be a user, table, column, and an activity can be a DML transaction and logon and logoff times.

The given figure illustrates this SIMPLE MODEL 1



Description of tables presented in simple auditing model 1

Table	Description
APP_ENTITY	Holds the name of the entity to be audited; an entity can be a name of a user, name of a table, or name of the column
APP_AUDIT_ACTION	Holds entities and the actions that are audited
APP_ACTION_TYPE	Holds the actions to be audited; an action can be UPDATE, DELETE, INSERT, LOGIN, or LOGOUT
APP_AUDIT_DATA	Holds audit trail data generated by the auditing process

Description of columns presented in simple auditing model 1

Column	Description
ACTION_TYPE_DESC	Name or description of the audited action such as UPDATE, INSERT, DELETE, LOGIN, or LOGOUT
ACTION_TYPE_ID	Unique identification number of APP_ACTION_TYPE table generated automatically by the application
AUDIT_ACTION_ID	Unique identification number of APP_AUDIT_ACTION table generated automatically by the application
AUDIT_DATA	Audit data trail generated by the auditing process
AUDIT_DATA_ID	Unique identification number of APP_AUDIT_DATA table generated automatically by the application
AUDIT_START_DATE	Date and time when the audit on a specific entity and action starts
AUDIT_EXPIRE_DATE	Data and time when the audit on a specific entity and action ends
ENTITY_ID	Unique identification number of APP_ENTITY table generated automatically by the application
ENTITY_NAME	Name of the user, table, or column to be audited
ENTITY_TYPE	Type of the entity to be audited; a type can be user, table, or column

Description of control columns (continued)

Column	Stands for	Description of the Control Column
CTL_AUD_START	CONTROL AUDIT START	Stores audit start date and time for current record
CTL_INS_DTTM	CONTROL INSERT DATE TIME	Stores the date and time the record is created
CTL_INS_USER	CONTROL INSERT USER	Stores the user name that created the record
CTL_PUR_FLAG	CONTROL PURGE FLAG	Indicates whether current record can be purged or not; possible values are Yes and No
CTL_REC_STAT	CONTROL RECORD STATUS	Stores the status of the current record; record status could be A for active, D for deleted, or I for inactive
CTL_SEC_LEVEL	CONTROL SECURITY LEVEL	Is used to define security access level for current record
CTL_UPD_DTTM	CONTROL UPDATE DATE TIME	Stores the date and time of the most recent update on the current record
CTL_UPD_USER	CONTROL UPDATE USER	Stores the user name that created or performed the last update on the record

Sample data for simple auditing model 1

Table	New Records
APP_ENTITY	10, SAM, USER, A 11, SALARY, TABLE, A
APP_ACTION_TYPE	1, UPDATE, A 2, INSERT, A 3, DELETE, A
APP_AUDIT_ACTION	1, 10, 1, 10-MAY-2005, 10-JUN-2005, 15-APR-2005, NULL, DBSEC, A 2, 10, 2, 10-MAY-2005, 10-JUN-2005, 15-APR-2005, NULL, DBSEC, A 3, 10, 3, 10-MAY-2005, 10-JUN-2005, 15-APR-2005, NULL, DBSEC, A 4, 11, 1, 10-MAY-2005, 10-JUN-2005, 15-APR-2005, NULL, DBSEC, A

Description of control columns

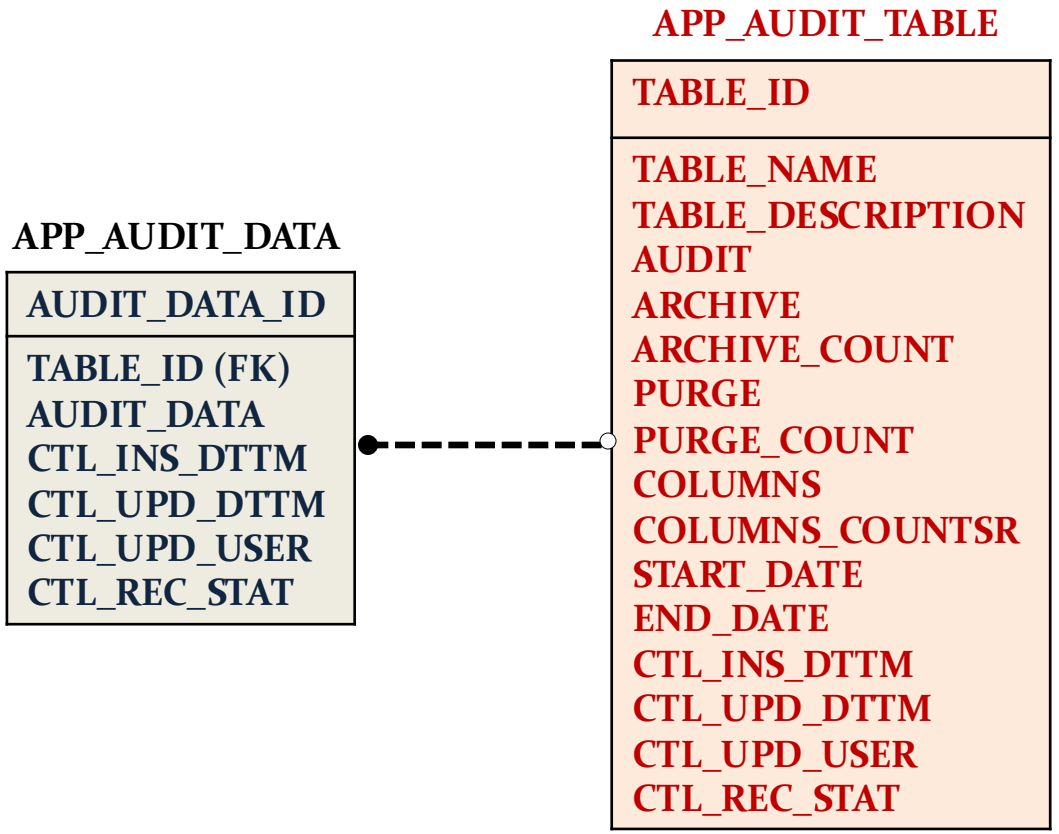
Column	Stands for	Description of the Control Column
CTL_ARC_FLAG	CONTROL ARCHIVE FLAG	Indicates whether current record can be archived or not; possible values are Yes and No
CTL_AUD_END	CONTROL AUDIT END	Stores audit end date and time for current record
CTL_AUD_FLAG	CONTROL AUDIT FLAG	Indicates whether current record is audited or not; possible values are Yes and No

AUDITING MODELS ...

Simple Auditing Model 2

- ✓ In this model , only column value changes are stored for audit purposes.
- ✓ The audit data table APP_AUDIT_DATA contains chronological data on all changes on column that are registered in APP_AUDIT_TABLE.
- ✓ There is a purging and archiving mechanism is used to help reduce the amount of data stored in DB.

The given figure illustrates this Simple auditing model 2



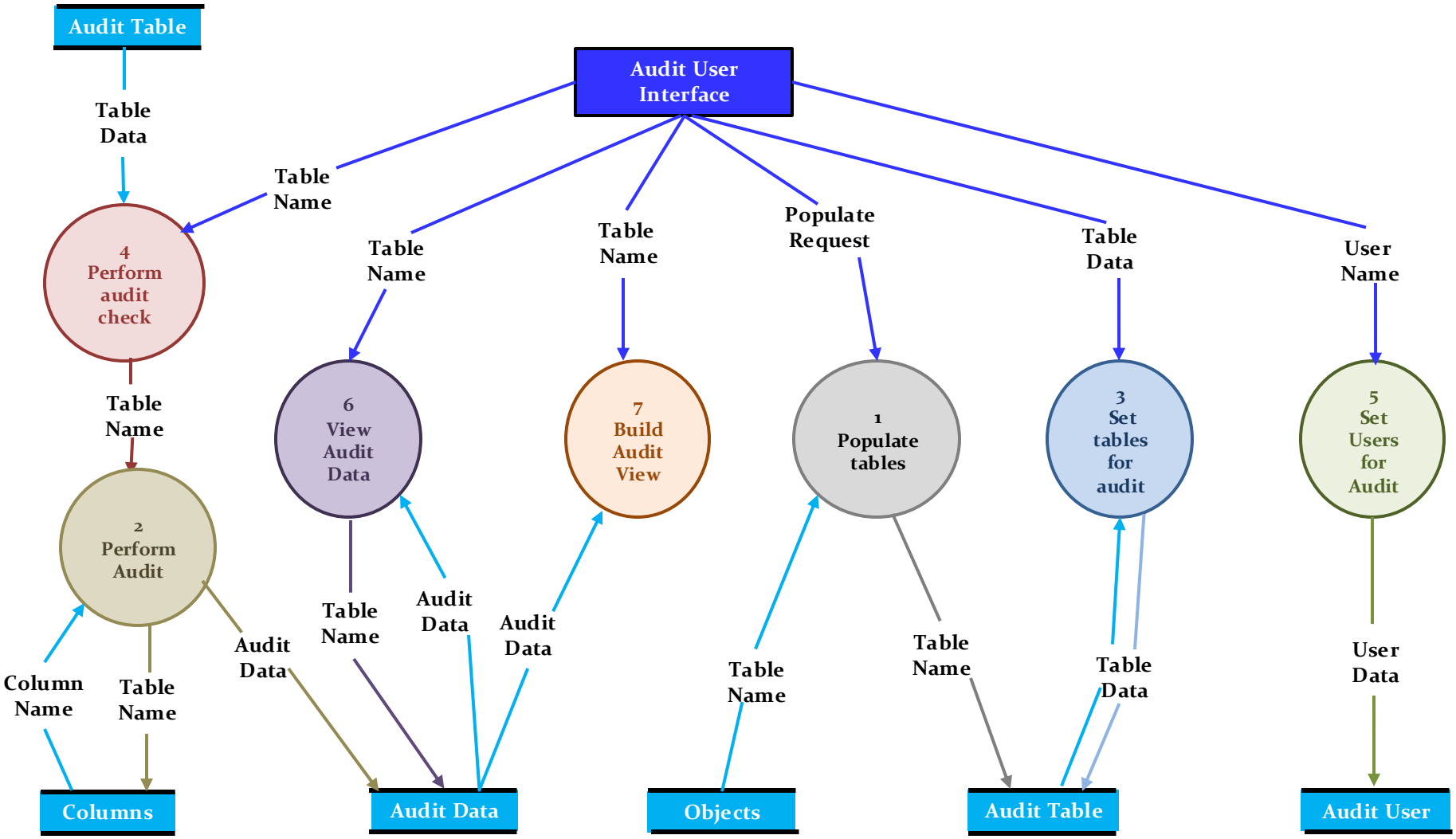
Auditing Models ...

Advanced Auditing Model

- ✓ This Model is called “advanced” because of its flexibility
- ✓ More flexible than simple models
- ✓ Used as an auditing application with a user interface
- ✓ Of course the repository for this model is more complex than previous models
- ✓ It contains data stores to register all entities that can be audited

Auditing Models ...

The following figure presents the flow of the user interface

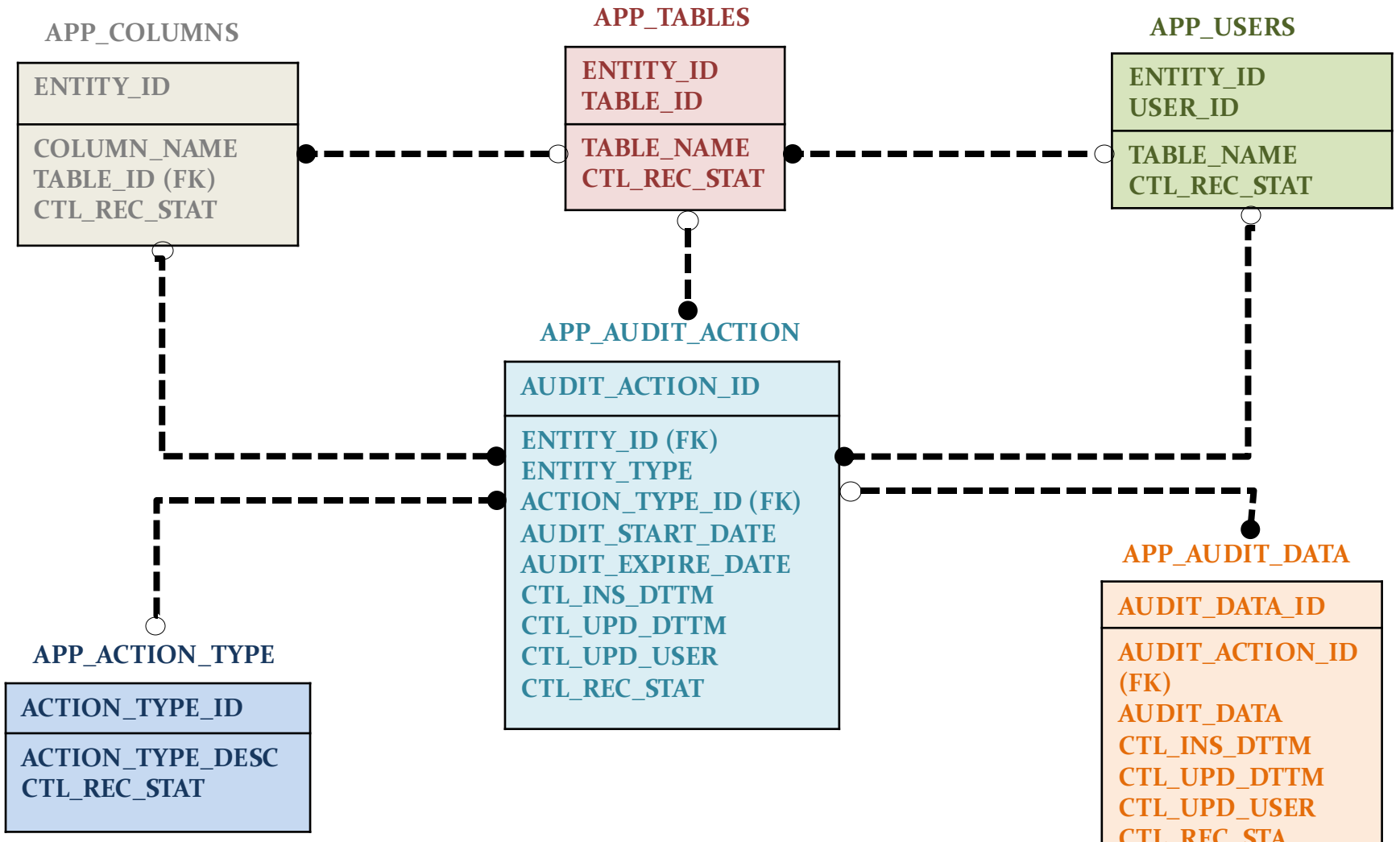


Description of tables presented in the advanced auditing model

Table	Description
APP_COLUMNS	Contains all columns to be audited
APP_TABLES	Contains all tables to be audited
APP_USERS	Contains all users to be audited
APP_AUDIT_ACTION	Holds entities and the actions that are audited
APP_ACTION_TYPE	Holds the actions to be audited; an action can be UPDATE, DELETE, INSERT, LOGIN, or LOGOUT
APP_AUDIT_DATA	Contains all audit data generated by the auditing process

Auditing Models ...

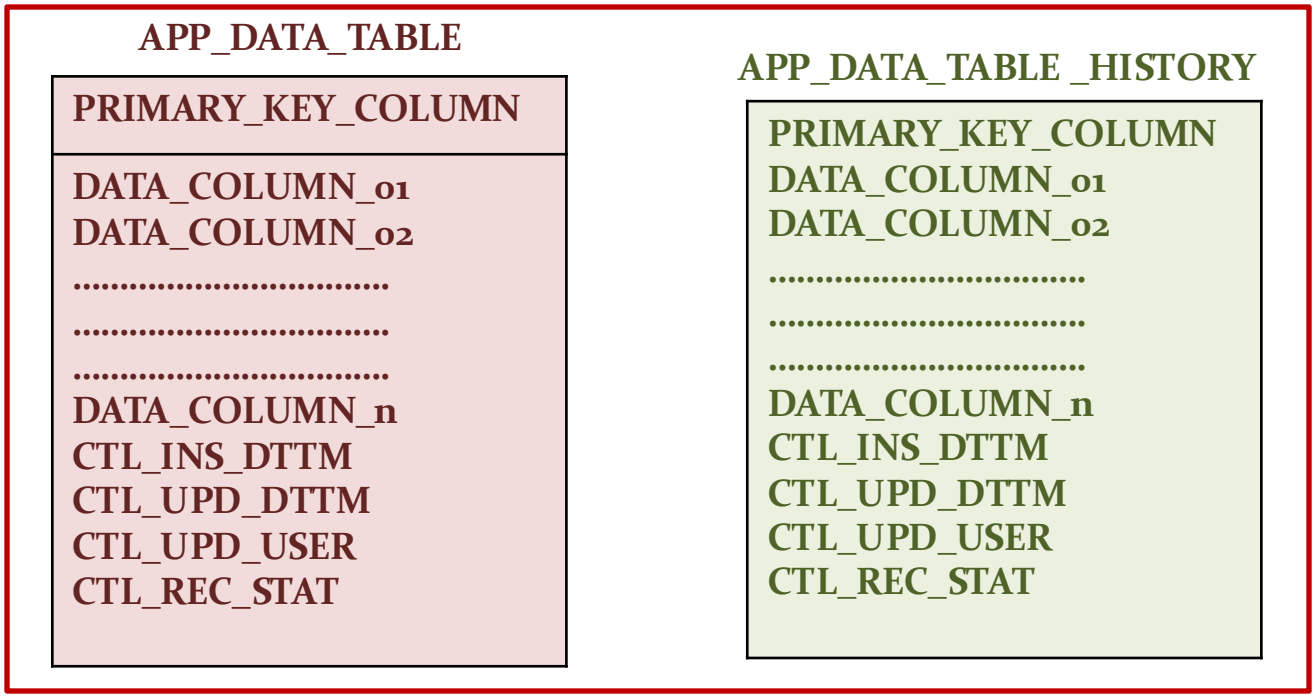
- ✓ Data model of the repository for an Advanced Auditing Model



Auditing Models ...

Historical Data Model

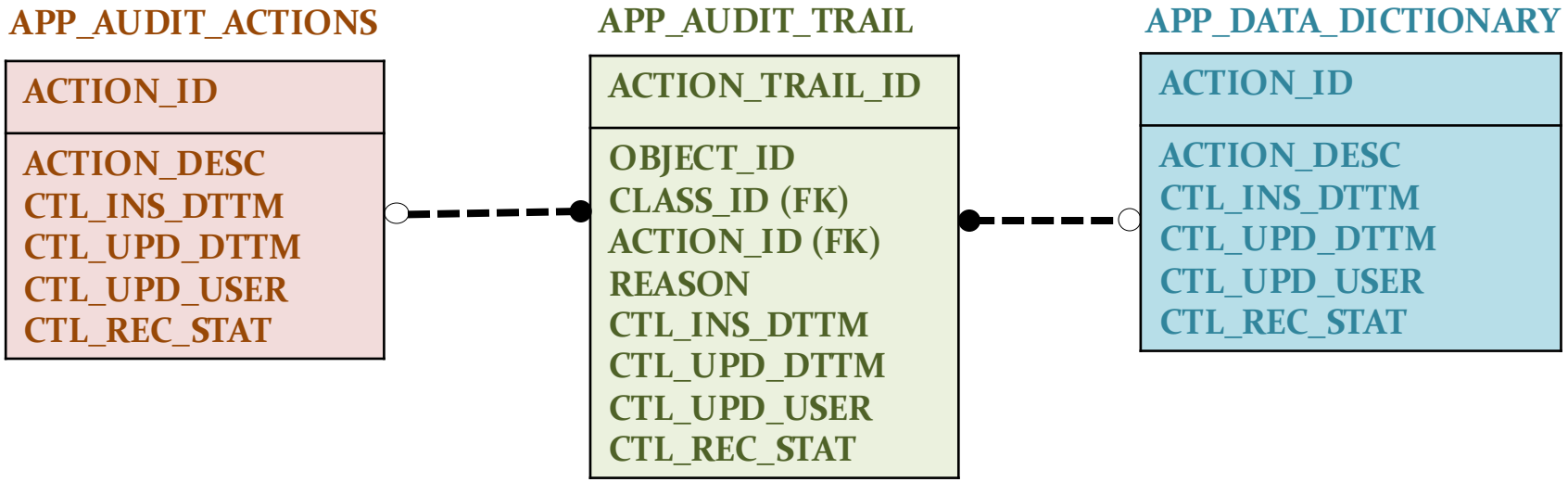
- ✓ This model is used for applications that require a record of the whole row when a DML transaction is performed on the table
- ✓ Typically used in most financial applications
- ✓ With this model , the whole row is stored in the HISTORY table, before it is changed or deleted
- ✓ The following figures illustrates this model



Auditing Models ...

Auditing Application Actions Model

- ✓ There may be a requirement for an application to audit specific operations or actions
- ✓ Example : You may want to audit a credit to an invoice, the reason for it being credited, the person who credited it, and the time it was credited.
- ✓ The following figure represents a Data Model of a repository for auditing application actions



Auditing Models ...

APP_AUDIT_ACTIONS data

Column Name	Column Value
ACTION_ID	1
ACTION_DESC	Invoice Credit
CTL_INS_DTTM	2004-05-10 12:30:21
CTL_UPD_DTTM	NULL
CTL_UPD_USER	ADMIN
CTL_RECT_STAT	ACTIVE

Auditing Models ...

DATA_DICTIONARY data

Column Name	Column Value
CLASS_ID	1
CLASS_DESC	Invoice Business Object
CTL_INS_DTTM	2004-05-10 12:30:21
CTL_UPD_DTTM	NULL
CTL_UPD_USER	ADMIN
CTL_REC_STAT	ACTIVE

APP_AUDIT_TRAIL data

Column	Column Value
AUDIT_TRAIL_ID	100001
OBJECT_ID	135432
CLASS_ID	1
ACTION_ID	1
REASON	Customer-presented coupon
CTL_INS_DTTM	2004-06-12 12:30:21
CTL_UPD_DTTM	NULL
CTL_UPD_USER	JDOE
CTL_REC_STAT	ACTIVE

Auditing Models ...

C2 Security

- ✓ C2 security is a type of security rating that evaluates the security framework for computer products used in government and military organizations and institutes.
- ✓ The standard was conceived by the U.S. National Computer Security Center (NCSC) to create a minimum security benchmark for all computing products and applications that process confidential government and military information.
- ✓ The National Security Administration has given a C2 security rating to Microsoft SQL Server 2000.
- ✓ This means that the server passes requirements set by the Department of Defence and is typically implemented in military and government applications
- ✓ When configured as C2 system, SQL Server utilizes DACLs (Discretionary Access Control to manage security and audit activity
- ✓ If all auditing counters are turned on for all objects, there could be a significant performance impact on the server.

Auditing Models ...

Requirements for enabling C2 auditing in SQL Server include the following :

- The Microsoft Windows Server must be configured as C2 system
- Windows Integrated Authentication is supported, but SQL native security is not supported
- Only transactional replication is supported
- The following SQL Server services are not included in a C2 evaluation
  - SQL Mail
  - Full Test Search
  - English Query
  - DTC
  - Meta Data Services
  - Analysis Services (OLAP)

Oracle Triggers

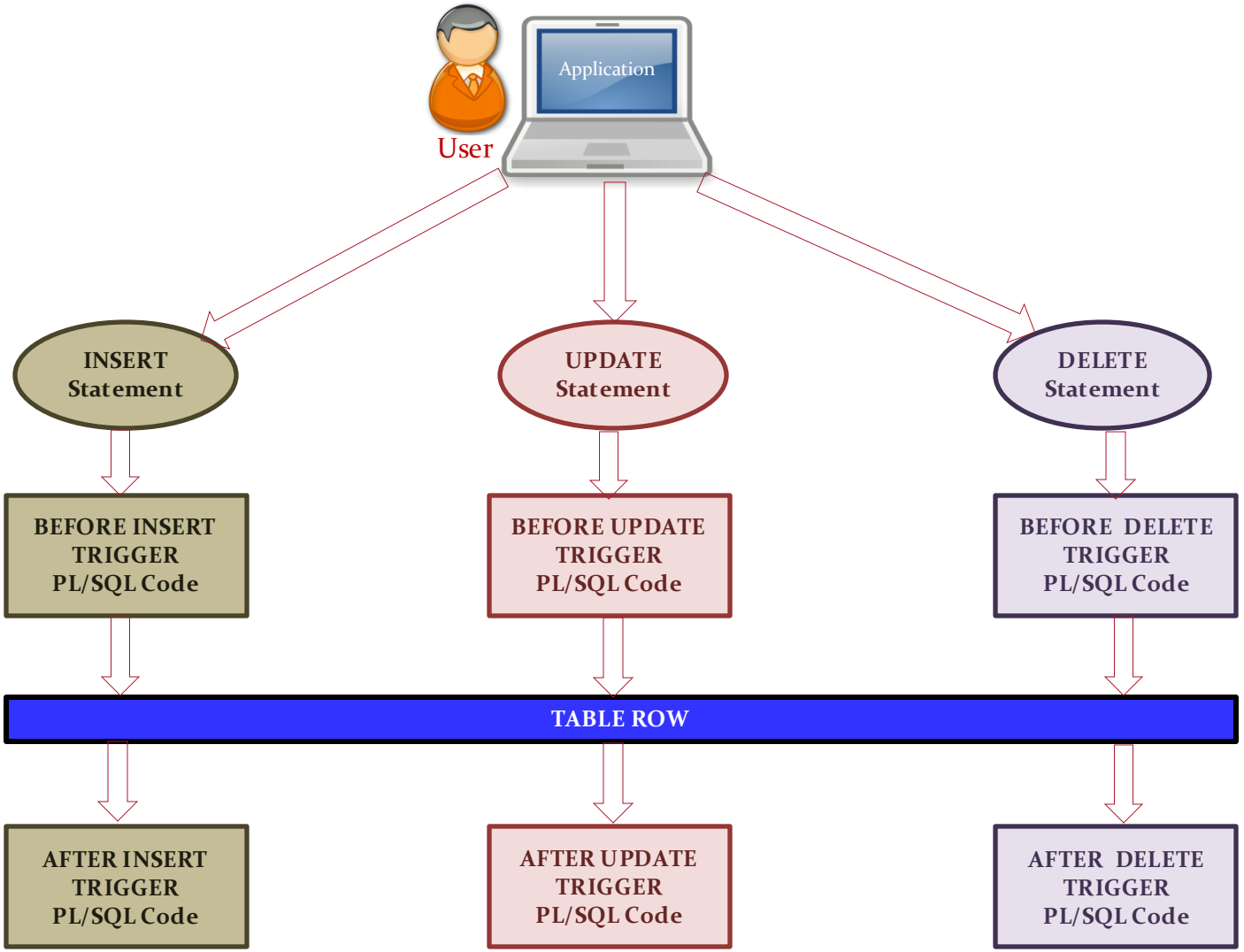
- ✓ A trigger is stored PL/SQL procedure that is executed automatically whenever a DML operations occurs or a specific database event occurs.
- ✓ ORACLE has six DML(INSERT, UPDATE,DELETE) events also known as trigger timings

Trigger mainly used for the following purposes

- ✓ Performing audits (Primary use)
- ✓ Preventing invalid data from being inserted into the tables
- ✓ Implementing business rules (Not highly recommended if the business rule is complex)
- ✓ Generating values for columns

Oracle Triggers ...

- ✓ ORACLE trigger timings or events for DML events



Oracle Triggers ...

Trigger Syntax

CREATE [OR REPLACE] TRIGGER <trigger_name>

[BEFORE | AFTER | INSTEAD OF] — Trigger Timing

[INSERT | UPDATE | DELETE.....] — Trigger Event

ON<name of underlying object>

[FOR EACH ROW] — Row Level

[WHEN<condition for trigger to get execute>] — Conditional Clause

DECLARE <Declaration part>

BEGIN <Execution part>

EXCEPTION <Exception handling part> — Error Handling Mechanism

END;

Oracle Triggers ...

The given syntax shows the different optional statements that are present in trigger creation.

- ✓ BEFORE/ AFTER will specify the event timings.
- ✓ INSERT/UPDATE/LOGON/CREATE/etc. will specify the event for which the trigger needs to be fired.
- ✓ ON clause will specify on which object the above-mentioned event is valid. For example, this will be the table name on which the DML event may occur in the case of DML Trigger.
- ✓ Command "FOR EACH ROW" will specify the ROW level trigger.
- ✓ WHEN clause will specify the additional condition in which the trigger needs to fire.
- ✓ The declaration part, execution part, exception handling part is same as that of the other PL/SQL blocks. Declaration part and exception handling part are optional.

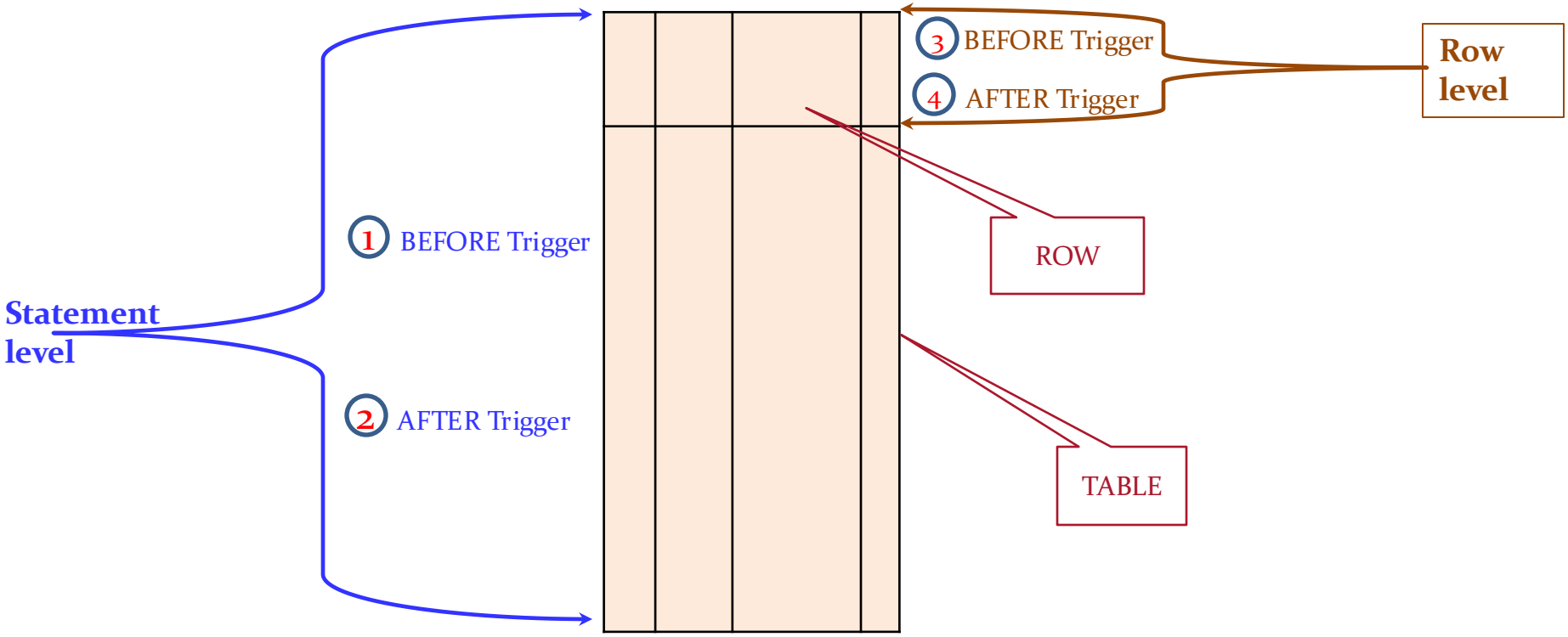
Oracle Triggers ...

ORACLE Trigger Execution

- ✓ A trigger can be in either of two distinct modes:
- ✓ Enabled - An *enabled* trigger executes its trigger action if a triggering statement is issued and the trigger restriction (if any) evaluates to TRUE.
- ✓ Disabled - A *disabled* trigger does not execute its trigger action, even if a triggering statement is issued and the trigger restriction (if any) would evaluate to TRUE.
- ✓ For enabled triggers, Oracle automatically
  - executes triggers of each type in a planned firing sequence when more than one trigger is fired by a single SQL statement
  - performs integrity constraint checking at a set point in time with respect to the different types of triggers and guarantees that triggers cannot compromise integrity constraints
  - provides read-consistent views for queries and constraints
  - manages the dependencies among triggers and objects referenced in the code of the trigger action
  - uses two-phase commit if a trigger updates remote tables in a distributed database
  - if more than one trigger of the same type for a given statement exists, Oracle fires each of those triggers in an unspecified order

Oracle Triggers ...

- ✓ The following figure gives the Order of trigger execution



Creating DLL Triggers with Oracle

- ***Data Definition Language (DDL) statements***—Including CREATE, ALTER, and DROP commands
- ***Data Control Language (DCL) statements***—Including GRANT and REVOKE commands
- ***Database events***—Including such events as AFTER LOGON and BEFORE LOGON
- ***SQL statements audit trail***—Including the audit trail—a history of all statements issued by a specific user on any table

The following is the Oracle10*g* CREATE TRIGGER syntax for DDL statements and database events:

```
CREATE [ OR REPLACE ] TRIGGER [ schema. ]trigger
    { BEFORE | AFTER | INSTEAD OF }
    { { ddl_event [ OR ddl_event ]...
      | database_event [ OR database_event ]...
      }
ON { [ schema. ]SCHEMA
       | DATABASE
       }
    }
    [ WHEN (condition) ]
    { pl/sql_block | call_procedure_statement } ;
```

WHERE:

ddl_event is ALTER, ANALYZE, ASSOCIATE STATISITICS, AUDIT, COMMENT, CREATE, DISASSOCIATE STATISTICS, DROP, GRANT, NOAUDIT, RENAME, REVOKE, TRUNCATE, or DDL.

Creating DLL Triggers with Oracle

Example of LOGON and LOGOFF Database Events

- Steps:
  - Log on as SYSTEM
  - Create the APP_AUDIT_LOGINS table
  - Create two triggers:
    - One that fires after the logon event
    - One that fires before the logoff event
  - Log on as DBSEC; disconnect after a few minutes
  - Log on as SYSTEM to check the auditing table

Creating DLL Triggers with Oracle

Example of LOGON and LOGOFF Database Events

As indicated earlier, there are many instances in which business requirements dictate that a database administrator capture all LOGON and LOGOFF activities to analyze database connectivity. In such instances, you need to follow the steps that follow:

1. Log on as SYSTEM and create the APP_AUDIT_LOGINS table and a sequence used to generate a unique ID number for each login record.

```
SQL> CREATE TABLE APP_AUDIT_LOGINS
  2  (
  3      LOGINS_ID           NUMBER,
  4      SESSION_ID          NUMBER,
  5      USERNAME            VARCHAR2(30),
  6      LOGON_TIME          DATE,
  7      LOGOFF_TIME         DATE,
  8      IP_ADDRESS          VARCHAR2(255),
  9      AUD_INS_DTTM        DATE,
 10      AUD_UPD_DTTM        DATE
 11  )
 12  /

Table created.

SQL> CREATE SEQUENCE SEQ_LOGIN_ID
  2  /

Sequence created.
```

Creating DLL Triggers with Oracle

2. Create two triggers, one that fires after the logon event and one that fires before the logoff event.

```
SQL> CREATE OR REPLACE TRIGGER TRG_AFTER_LOGON
  2     AFTER LOGON ON DATABASE
  3  BEGIN
  4     INSERT INTO APP_AUDIT_LOGINS VALUES
  5        (SEQ_LOGIN_ID.NEXTVAL,
  6         SYS_CONTEXT('USERENV', 'SESSIONID'),
  7         USER,
  8         SYSDATE,
  9         NULL,
 10         SYS_CONTEXT('USERENV', 'IP_ADDRESS'),
 11         SYSDATE,
 12         NULL
 13        );
 14  END;
 15  /

Trigger created.

SQL> CREATE OR REPLACE TRIGGER TRG_BEFORE_LOGOFF
  2     BEFORE LOGOFF ON DATABASE
  3  BEGIN
  4     UPDATE APP_AUDIT_LOGINS SET
  5            LOGOFF_TIME = SYSDATE,
  6            AUD_UPD_DTTM= SYSDATE
  7      WHERE SESSION_ID  = SYS_CONTEXT('USERENV', 'SESSIONID')
  8        AND USERNAME    = USER
  9        AND LOGOFF_TIME IS NULL;
 10  END;
 11  /

Trigger created.
```

Creating DLL Triggers with Oracle

3. Log on as DBSEC and then disconnect after a few minutes.

```
SQL> CONN DBSEC
Enter password: *****
Connected.
SQL> DISCONNECT
```

Creating DLL Triggers with Oracle

4. Log on as SYSTEM and view the contents of the APP_AUDIT_LOGINS table.

```
SQL> SELECT * FROM APP_AUDIT_LOGINS
  2  /

LOGINS_ID SESSION_ID USERNAME   LOGON_TIM LOGOFF_TI IP_ADDRESS         AUD_INS_D AUD_UPD_D
--------- ---------- ---------- --------- --------- ----------------- --------- ---------
        1        585 DBSEC      06-AUG-04 06-AUG-04 127.0.0.1         06-AUG-04 06-AUG-04
        2        586 SYSTEM     06-AUG-04           127.0.0.1         06-AUG-04
```

Creating DLL Triggers with Oracle

DDL Event Example

The second example is a DDL event. Create a trigger that prevents DBSEC from altering any of its tables.

1. Log on as SYSTEM and create a trigger that fires before an ALTER statement is completed.

```
SQL> CREATE OR REPLACE TRIGGER TRG_BEFORE_ALTER
 2    BEFORE ALTER ON DATABASE
 3 BEGIN
 4
 5  IF USER = 'DBSEC' THEN
 6    RAISE_APPLICATION_ERROR(-20000, 'YOU MAY NOT MODIFY STRUCTURE OF ANY TABLE');
 7     END IF;
 8  END;
 9  /

Trigger created.
```

2. Log on as DBSEC and alter the CUSTOMERS table. If the table does not exist, create one.

```
SQL> ALTER TABLE CUSTOMERS
  2     MODIFY NAME VARCHAR2(60)
  3  /
ALTER TABLE CUSTOMERS
*
ERROR at line 1:
ORA-00604: error occurred at recursive SQL level 1
ORA-20000: YOU MAY NOT MODIFY STRUCTURE OF ANY TABLE
ORA-06512: at line 4
```

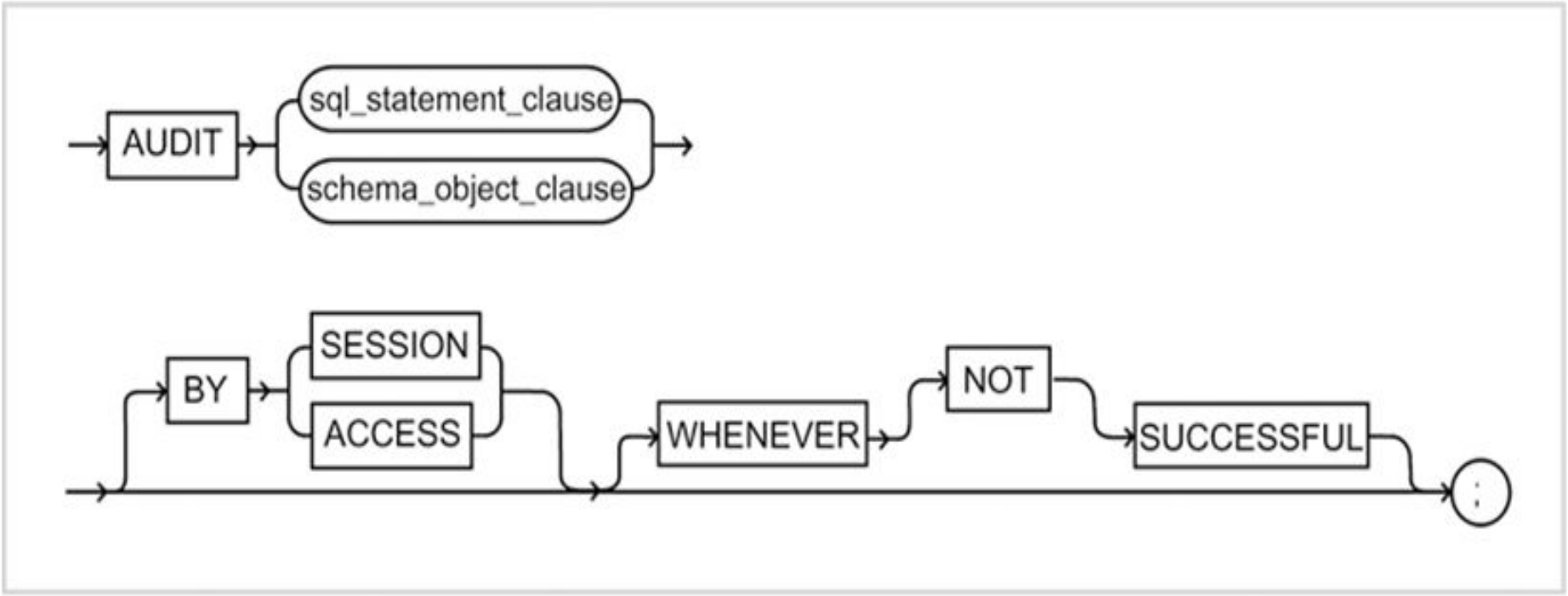
Auditing Database Activities with ORACLE

- ✓ ORACLE provides the mechanism for auditing everything:
  - From tracking who is creating and modifying the structure
  - Who is granting privileges to whom
- ✓ The activities are divided into two types based on the type of SQL command statement used :
  - Activities defined by DDL (Data Definition Language)
  - Activities defined by DCL (Data Control Language)

Auditing Database Activities with ORACLE

Auditing DDL Activities

- ✓ ORACLE uses a SQL-based audit command
- ✓ The following figure presents the audit syntax diagram (ORACLE 10g)



Auditing Database Activities with ORACLE ...

Audit command syntax

```
AUDIT
{
{ { statement_option | ALL }
            [,{statement_option | ALL }] .....
            |,{syetem_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] SUCESSFUL ] ;
```

Where :

Statement option – Tells ORACLE to audit the specified DDL or DCL statement
DDL – CREATE, ALTER, DROP and TRUNCATE
DCL – GRANT , REVOKE

System privilege – Tell ORACLE to audit the specified privilege such as SELECT, CREATE ANY, or ALTER ANY

Object_option – Specifies the type of privileges for the specified object to be audited

BY SESSION – Tells ORACLE to record audit data once per session even if the audited statement issued multiple times in session

BY ACCESS - Tells ORACLE to record audit data every time audited statement is issued.

WHENEVER SUCCESSFUL – Tells ORACLE to capture audit data only when the audited command is successful

WHENEVER NOT SUCCESSFUL- Tells ORACLE to capture audit data only when the audited command fails

Auditing Database Activities with ORACLE ...

DDL activities Example :

- ✓ Suppose you want to audit a table named CUSTOMER every time it is altered or every time a record from a table deleted.
- ✓ The following steps show you how to do this.
- ✓ Before perform , drop are disable all triggers associated with CUSTOMER table.

Step 1 : Use any user other than SYS or SYSTEM to create the CUSTOMER

```
SQL> CREATE TABLE CUSTOMER
  2  (
  3     ID          NUMBER  ,
  4     NAME        VARCHAR2 (20),
  5     CR_LIMIT    NUMBER
  6  );

Table created
```

Auditing Database Activities with ORACLE ...

Step 2 : Add three rows into the CUSTOMER table and commit changes

```
SQL > INSERT INTO CUSTOMER VALUES (2, 'BMNANTHA', 200);

1 row created

SQL > INSERT INTO CUSTOMER VALUES (3, 'MURUGAN', 300);

1 row created

SQL > INSERT INTO CUSTOMER VALUES (1, 'GANESH', 100);

1 row created

SQL > COMMIT;

Commit complete
```

Auditing Database Activities with ORACLE ...

Step 3 : Log on as SYS or SYSTEM to enable auditing , as specified in this example the first statement for ALTER and the next is for DELETE

```
SQL > CONNECT SYSTEM @ SEC
Enter password : ******
Connected.
SQL > AUDIT ALTER ON DBSEC.CUSTOMER BY ACCESS WHENEVER
      2  SUCCESSFUL;

Audit succeeded.

SQL > AUDIT DELETE ON DBSEC.CUSTOMER BY ACCESS
WHENEVER
      2  SUCCESSFUL;

Audit succeeded.
```

Auditing Database Activities with ORACLE ...

Step 4 : Login as the owner of CUSTOMER table, DBSEC delete a row and modify the structure of the table, as specified in the following code

```
SQL > CONNECT DBSEC@ SEC
Enter password : ******
Connected.
SQL > DELETE FROM CUSTOMER WHERE ID = 3;

1 row deleted.

SQL > ALTER TABLE CUSTOMER MODIFY NAME  VARCHAR2(30);

Table altered
```

Auditing Database Activities with ORACLE ...

In t the DELETE
and
Ste

SQL Scratchpad - SYS@SEC

```
SELECT OS_USERNAME, USERNAME, TIMESTAMP, OWNER, OBJ_NAME, ACTION_NAME
  FROM DBA_AUDIT_TRAIL
```

OS_USERNAME	USERNAME	TIMESTAMP	OWNER	OBJ_NAME	ACTION_NAME
Hassan?Afyouni	DBSEC	22-Jul-2004 04:17:32 AM	DBSEC	CUSTOMER	DELETE
Hassan?Afyouni	DBSEC	22-Jul-2004 04:17:40 AM	DBSEC	CUSTOMER	ALTER TABLE

Commit is ... | Execute time (s): 0.047 | Rows returned: 2 | Execute | Close | Help

Auditing Database Activities with ORACLE ...

- ✓ When audit process got over of a specific object or command, you may turn it off by using the NO AUDIT statement.

- ✓ The following step turns off auditing on the two statements issued in step 3.

```
SQL > NOAUDIT ALTER ON DBSEC.CUSTOMER;

Noaudit succeeded.

SQL > NOAUDIT DELETE ON DBSEC.CUSTOMER;

Noaudit succeeded.
```

Auditing Database Activities with ORACLE ...

DCL Activities Example:

- ✓ You are auditing the GRANT privilege issued on a TEMP table owned by DBSEC.
- ✓ The following steps shows how to audit the DCL statements audited.
- ✓ The same steps to be followed for all DCL Commands.

Step 1 : Log on as SYSTEM or SYS and issue an AUDIT statement as follows

```
SQL> CONN SYSTEM
Enter password : ******
Connected

SQL> DELETE SYS.AUD$;
1 row deleted.

SQL> COMMIT;
Commit complete.

SQL> AUDIT GRANT ON DBSEC.TEMP;
Audit succeeded
```

Auditing Database Activities with ORACLE ...

Step 2: Log on as DBSEC and grant SELECT and UPDATE privileges to SYSTEM on TEMP table

```
SQL> CONN DBSEC
Enter password : *****
Connected.

SQL> GRANT SELECT ON TEMP TO SYSTEM;
Grant succeeded.

SQL> GRANT UPDATE ON TEMP TO SYSTEM
Grant succeeded.
```

Step 3: Log on as SYSTEM and display the contents of DBA_AUDIT_TRAIL.

```
SQL> SELECT USERNAME, TIMESTAMP, OWNER, OBJ_NAME FROM
  2  DBA_AUDIT_TRAIL;
USERNAME        TIMESTAMP       OWNER           OBJ_NAME
--------------- --------------- --------------- ---------------
DBSEC           20-Jan-20       DBSEC           TEMP
DBSEC           20-Jan-20       DBSEC           TEMP

2 rows selected
```

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 occurrence
- ✓ 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
- ✓ There are two types of auditing in SQL Server 2000
  - Auditing
  - C2Auditing
- ✓ Auditing can have significant impact on performance (it varies depends on how many countries you have enabled and how many objects you are auditing)
- ✓ The audit trail analysis can also be costly in terms of system
- ✓ It is recommended that SQL profiler be run on a server separate from the production server

SQL SERVER PROFILER

- Microsoft SQL server profiler is a graphical user interface to SQL trace for monitoring an instance of the database engine. You can capture and save data about each event monitoring an to a file or table to analyze later.

WHAT EDITIONS OF SQL SERVER IS SQL PROFILER AVAILABLE ??

- SQL Profiler is only available in the Enterprise, Business Intelligence and Standard editions.

Auditing Server Activity with SQL Server 2000 ...

Implementing SQL Profiler

- ✓ One of the tools that accompanies SQL Server 2000 is SQL Profiler
- ✓ This tool provides the user interface for auditing events.
- ✓ You can audit several types of events using SQL Profiler

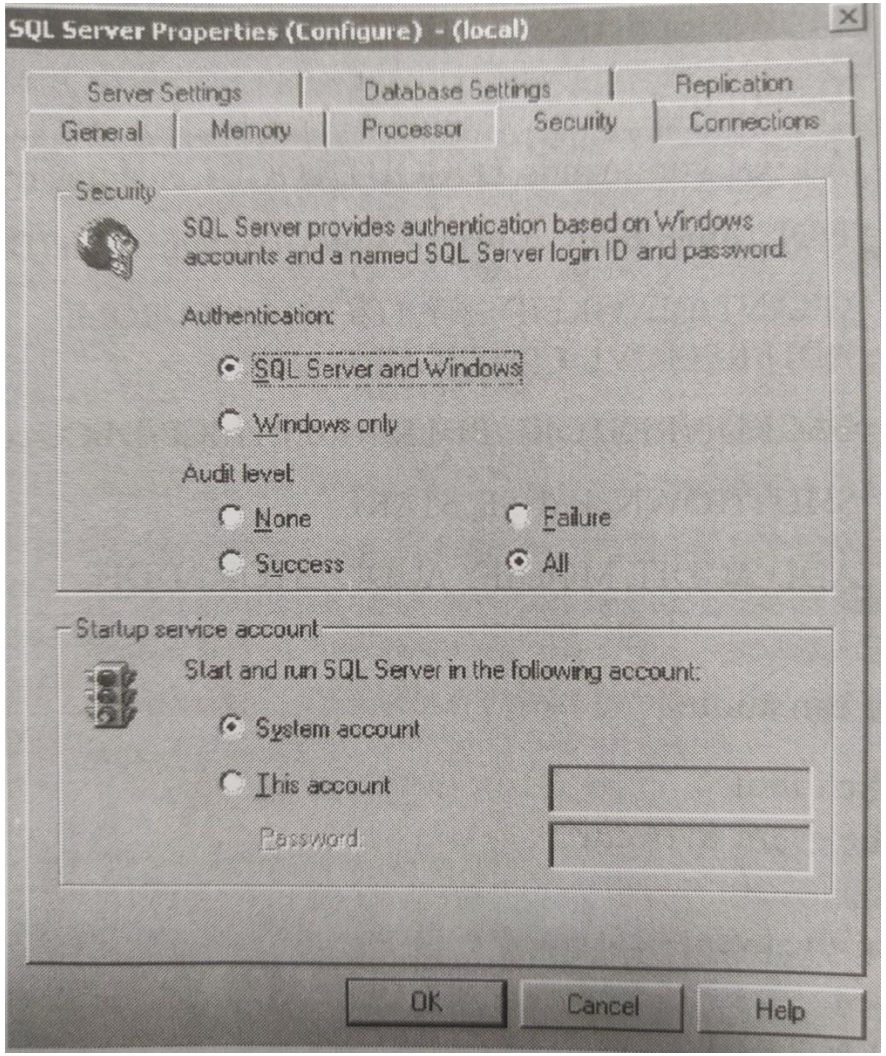
EVENT	DESCRIPTION	For each event, you can audit
End user events	All SQL commands, LOGIN/LOGOUT, enabling	✓ Date and time of the event ✓ User who caused the event to occur ✓ Type of Event ✓ Success or failure of the event ✓ Origin of the request ✓ Name of the object accessed ✓ Text of the SQL statement (Passwords replace with *****)
DBA events	DDL (other than security events), Configuration (DB or Server)	
Security events	GRANT/REVOKE/DENY/ LOGIN USER ROLE/ADD/REMOVE/CONFIGURE	
Utility events	BACKUP/RESTORE/BULK INSERT/ BCP/ DBCC Commands	
Server events	SHUTDOWN , PAUSE, START	
Audit events	ADD AUDIT, MODIFY AUDIT, STOP AUDIT	

Auditing Server Activity with SQL Server 2000 ...

- ✓ Security audit should be enable first
- ✓ This is done by setting the security auditing level under the SQL server properties in Enterprise Manager
- ✓ Security events can be audited on success, failure or both
- ✓ Follow these steps
  1. Open the Enterprise Manager
  2. Expand the appropriate SQL Server group
  3. Right click on the desired server
  4. Click properties
  5. On the security tab, select the desired security level as shown in the figure (next slide)

Auditing Server Activity with SQL Server 2000 ...

- ✓ SQL Server configuration



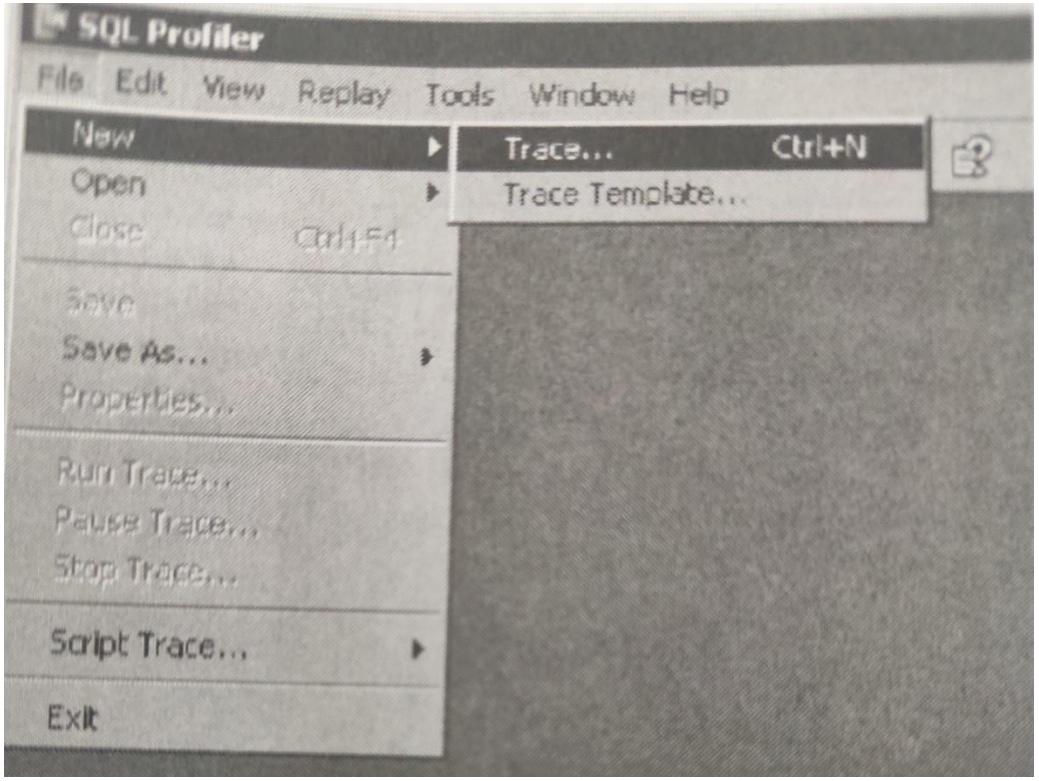
Auditing Server Activity with SQL Server 2000 ...

- ✓ After the audit level is set, you can then use SQL Profiler to monitor security events.
- ✓ The following events can be audited

- ADD DB USER
- ADD LOGIN TO SERVER ROLE
- ADD MEMBER TO DB ROLE
- ADD ROLE
- APP ROLE CHANGE PASSWORD
- BACKUP / RESTORE
- CHANGE AUDIT
- DBCC
- LOGIN
- LOGOUT
- LOGIN CHANGE PASSWORD
- LOGIN CHANGE PROPERTY
- LOGIN FAILED
- Login GDR (GRANT, DENT, REVOKE)
- Object Derived Permissions
- Object GDR
- Object Permissions
- Server Start and Stop
- Statement GDR
- Statement Permission

Auditing Server Activity with SQL Server 2000 ...

- ✓ You can start SQL Profiler by selecting it from the program group on the Start menu or from the tools menu in Enterprise.
- ✓ To start a new Audit Trace from the file menu, Click New , then Trace
- ✓ It is shown in the below figure



Auditing Server Activity with SQL Server 2000 ...

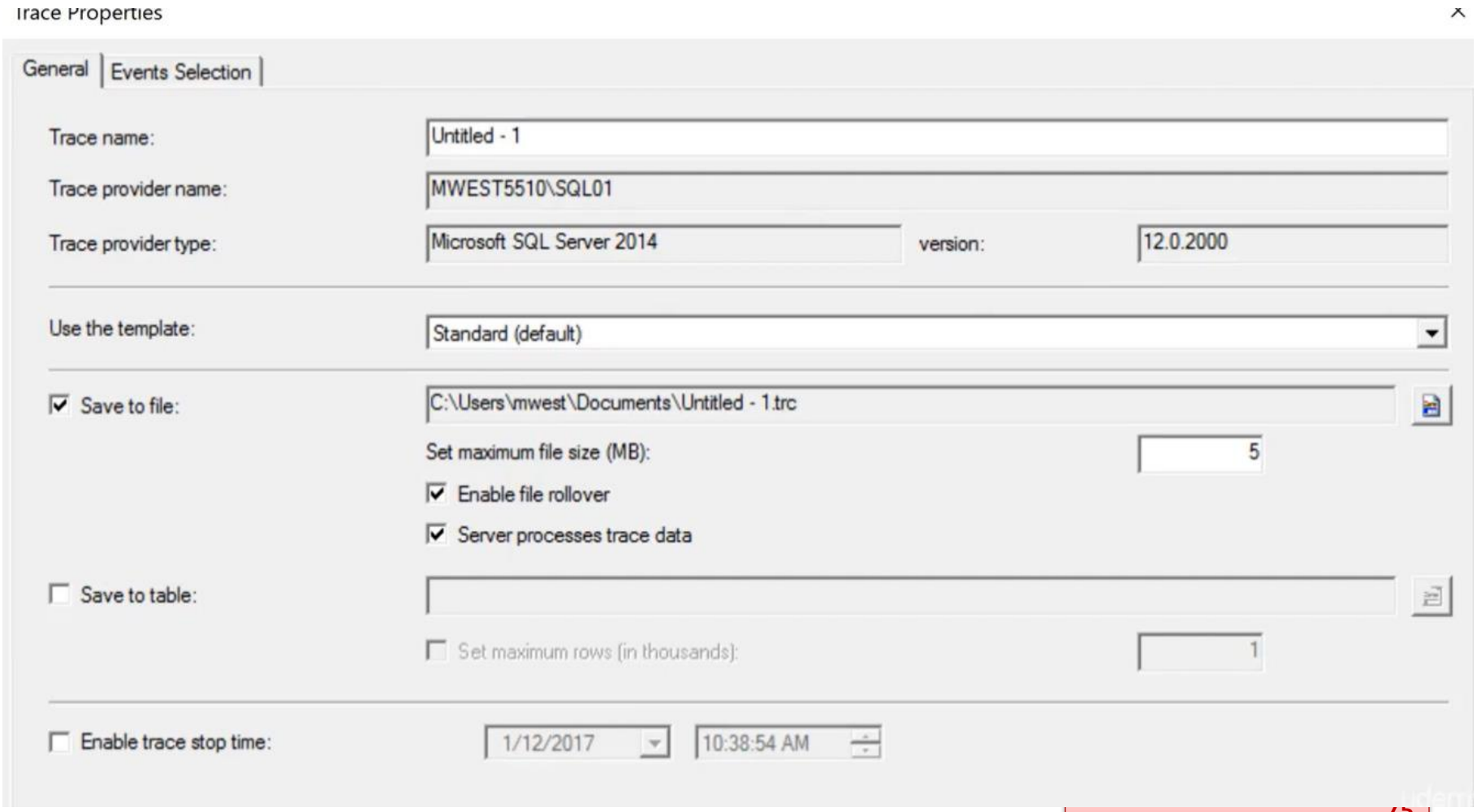
The new trace dialog box appears,
as shown in the figure

On the general tab, you provide:

- A name for the trace
- The server you want to audit
- The base template to start with
- Where to save the audit data, either to a file or to a DB
- A stop time, if you don't want the trace to run indefinitely

Auditing Server Activity with SQL Server 2000 ...

The new trace dialog box appears,
as shown in the figure

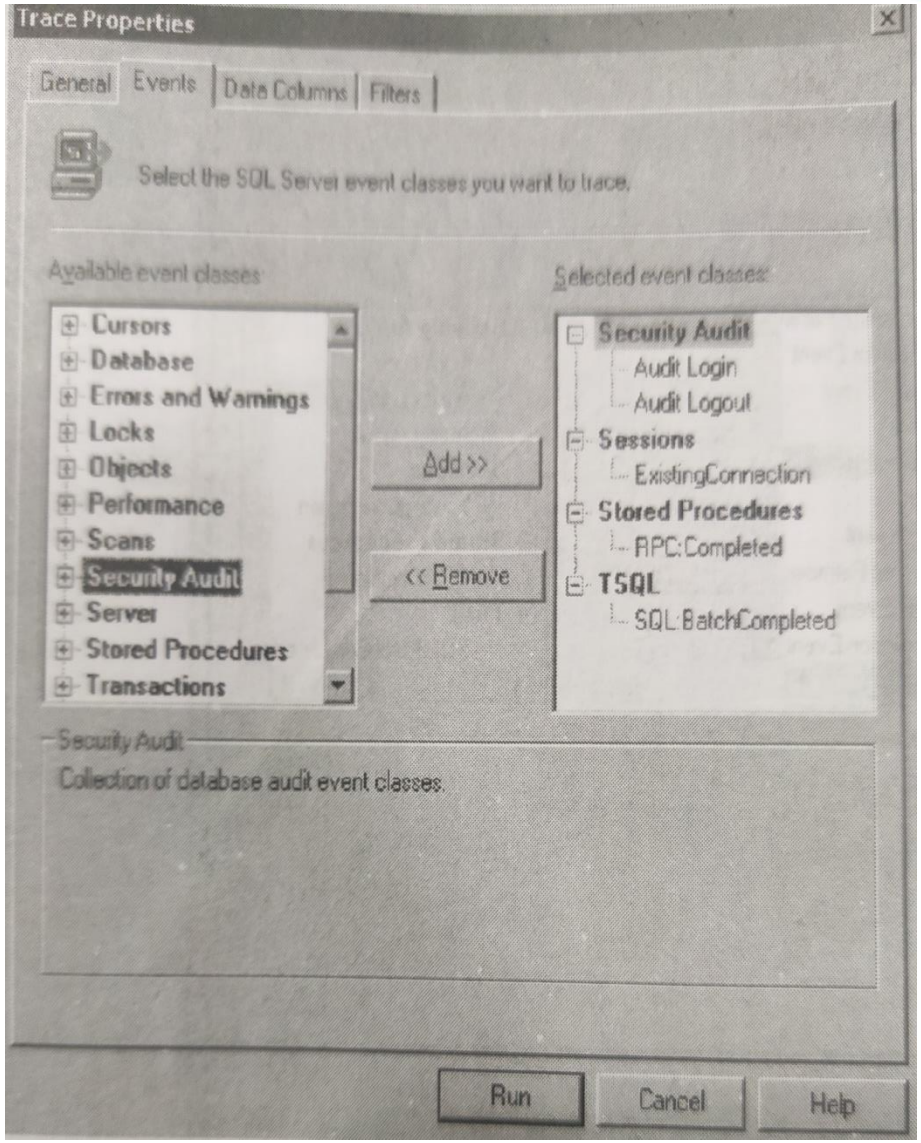


Auditing Server Activity with SQL Server 2000 ...

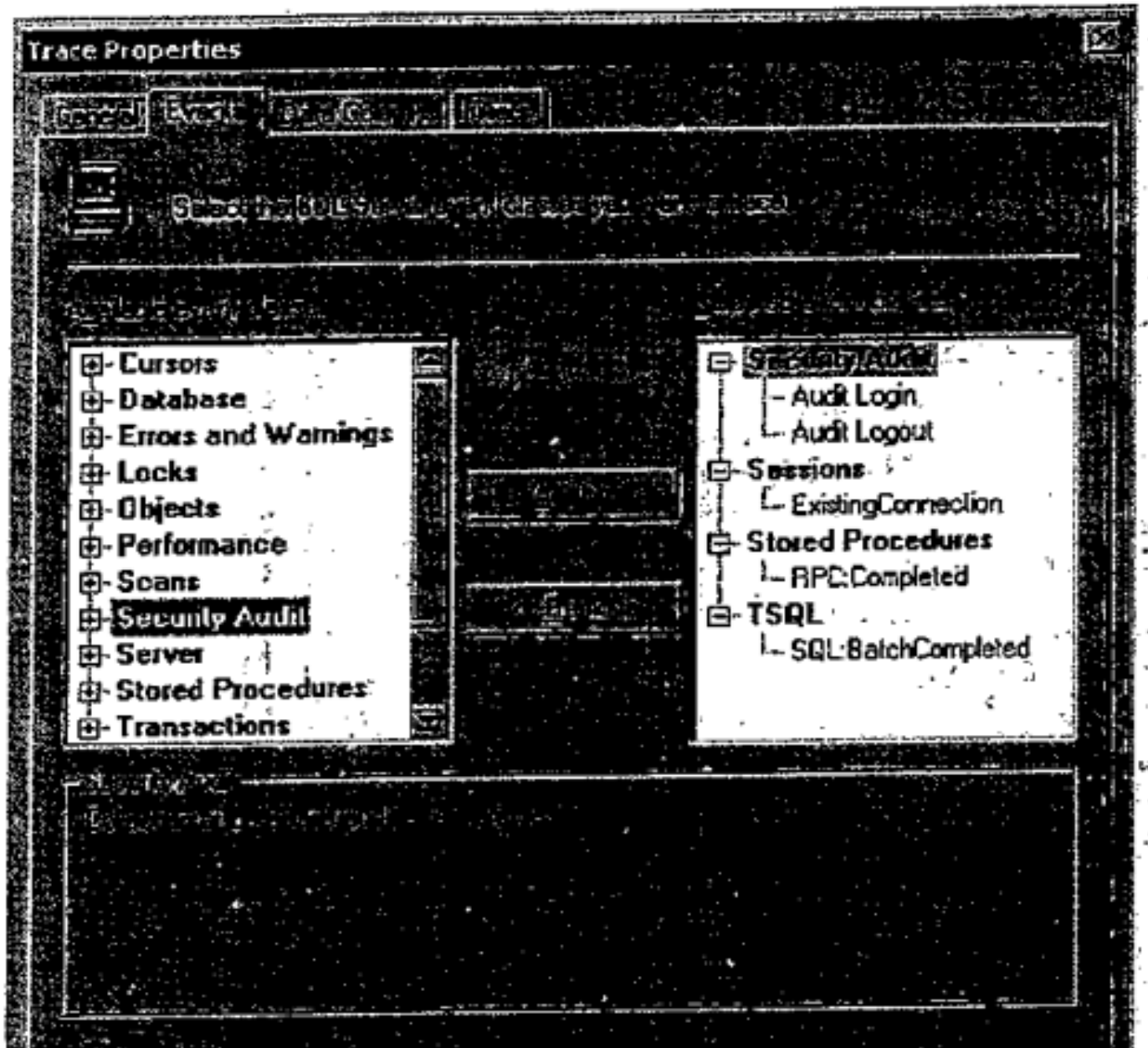
The new trace dialog box appears,
as shown in the figure

Auditing Server Activity with SQL Server 2000 ...

- ✓ On the events tab, you specify events to be audited and in which category they belong
- ✓ As shown in the figure



Auditing Server Activity with SQL Server 2000 ...

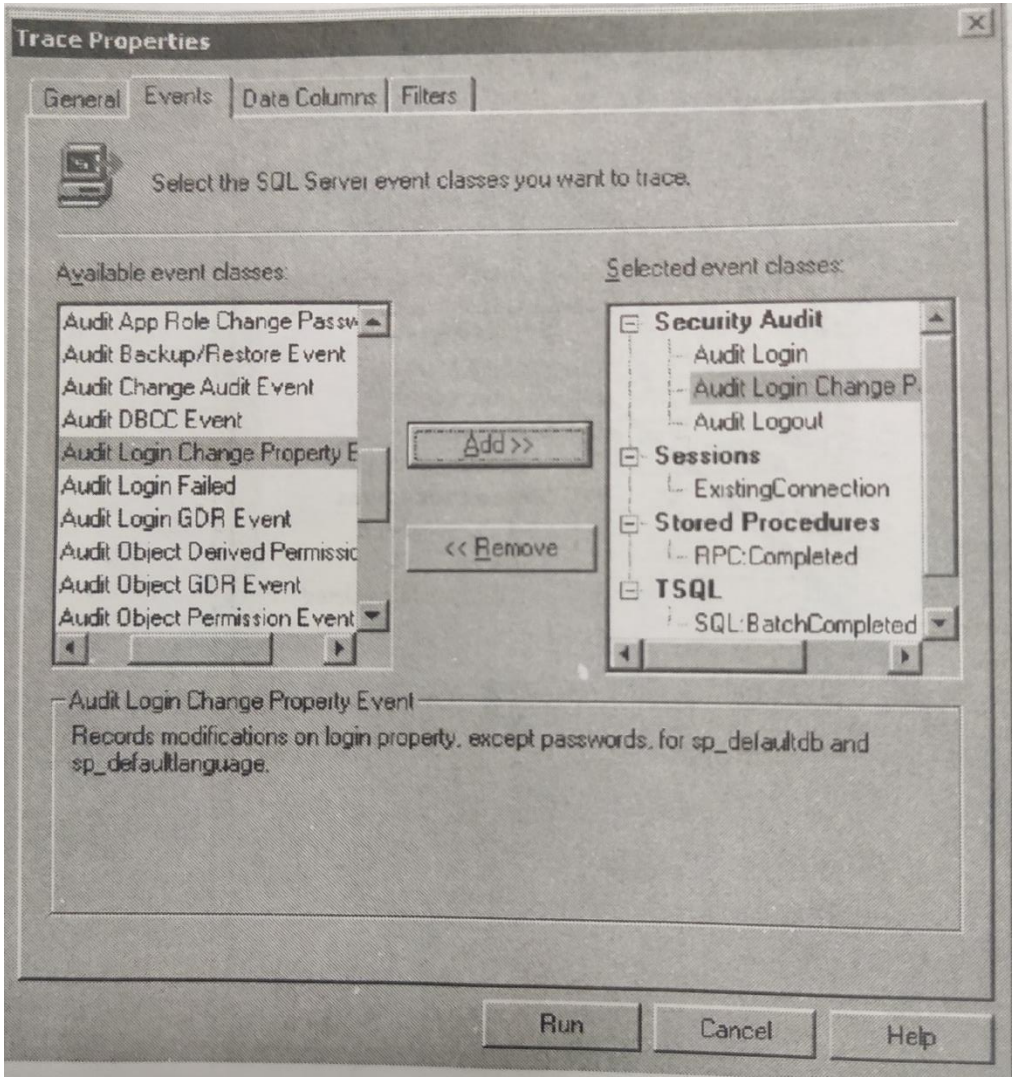


Auditing Server Activity with SQL Server 2000 ...

Add the Login Change Password security event to the trace by performing following steps

- ✓ Expand the **Security Audit** node under Available event classes
- ✓ Click **Audit Login Change Password Event**
- ✓ Click the **Add button**

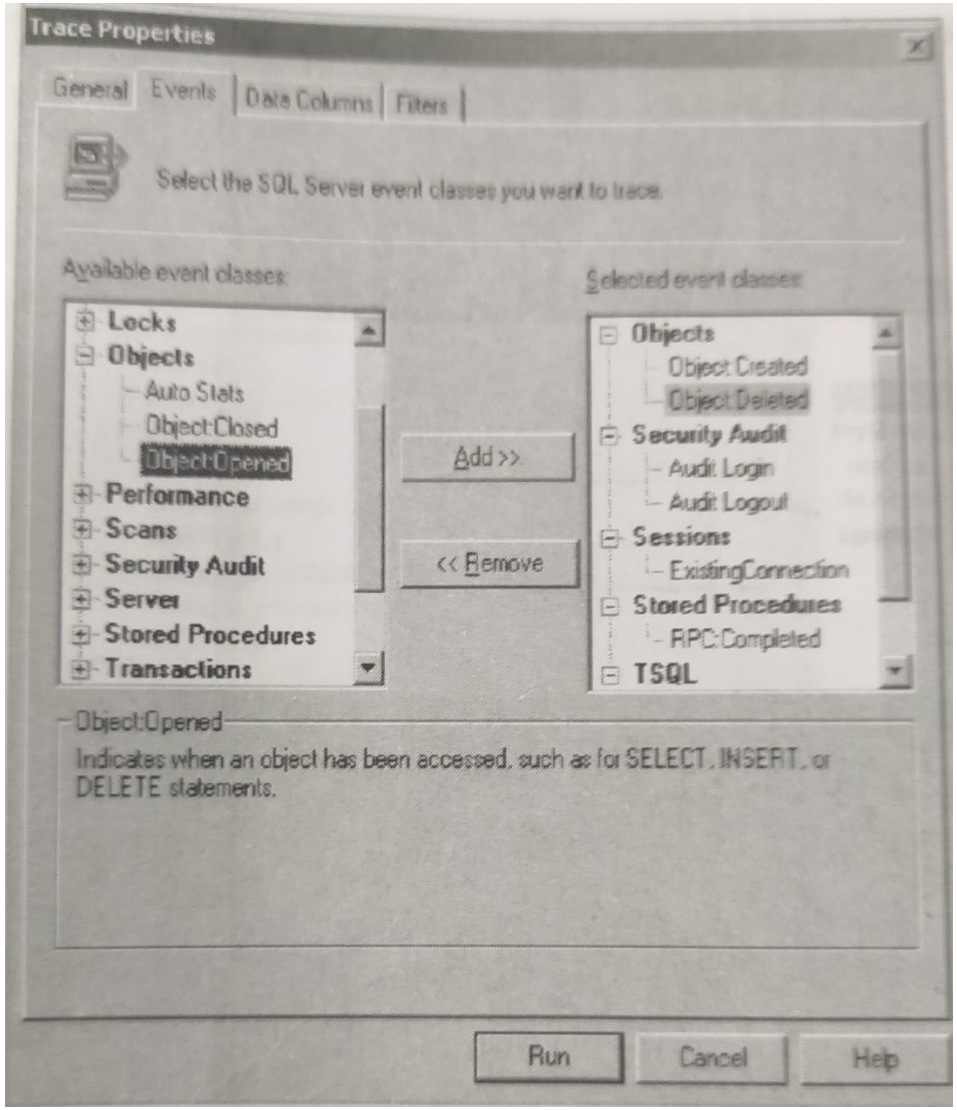
Audit Login Change Password Event should now appear under security Audit in Selected event classes, as shown in the figure



Auditing Server Activity with SQL Server 2000 ...

Data Definition Auditing

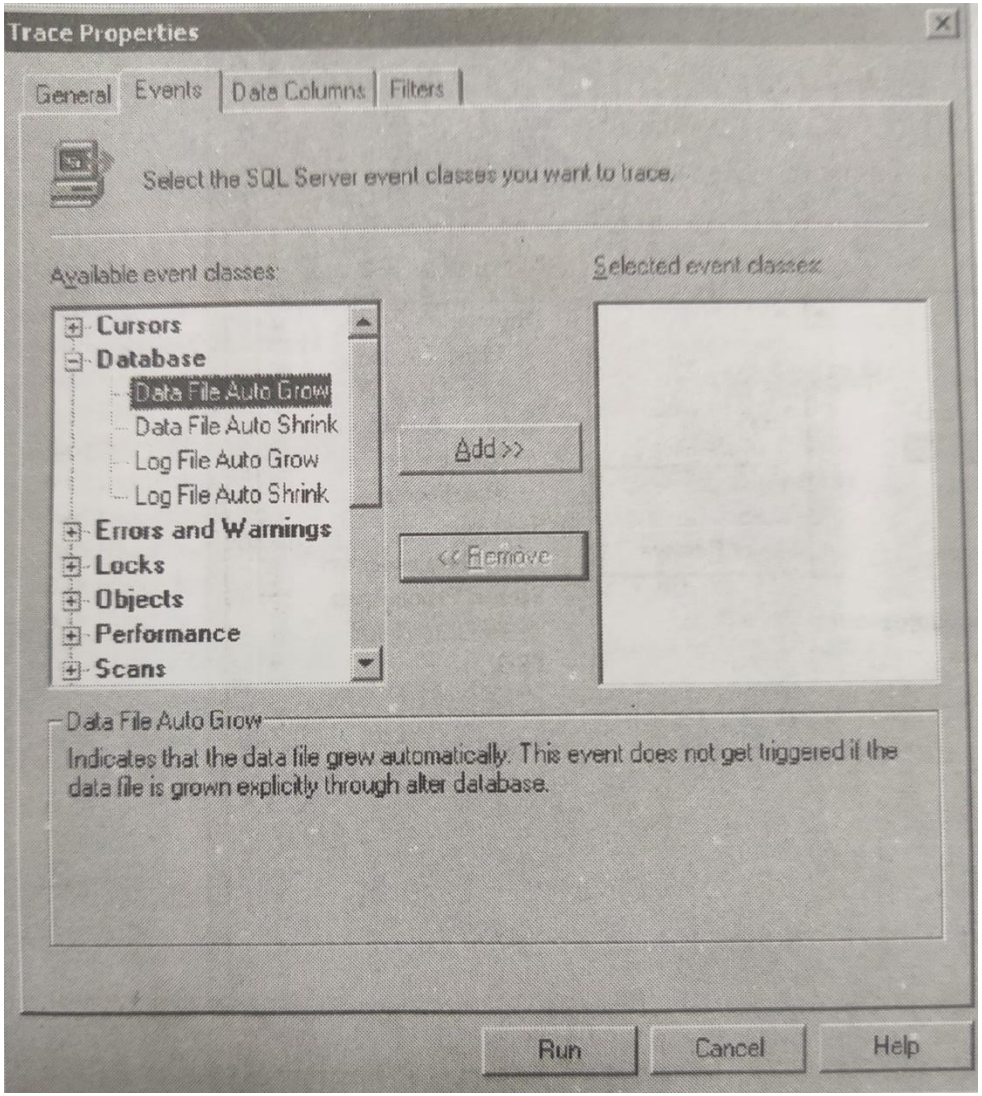
- ✓ To audit DDL statements, on the Events tab of your trace, you select **Object:Created** and **Object:Deleted** under the **objects** Category
- ✓ These two events audit all CREATE and DROP statements.
- ✓ It is shown in the figure



Auditing Server Activity with SQL Server 2000 ...

Database Auditing with SQL Server

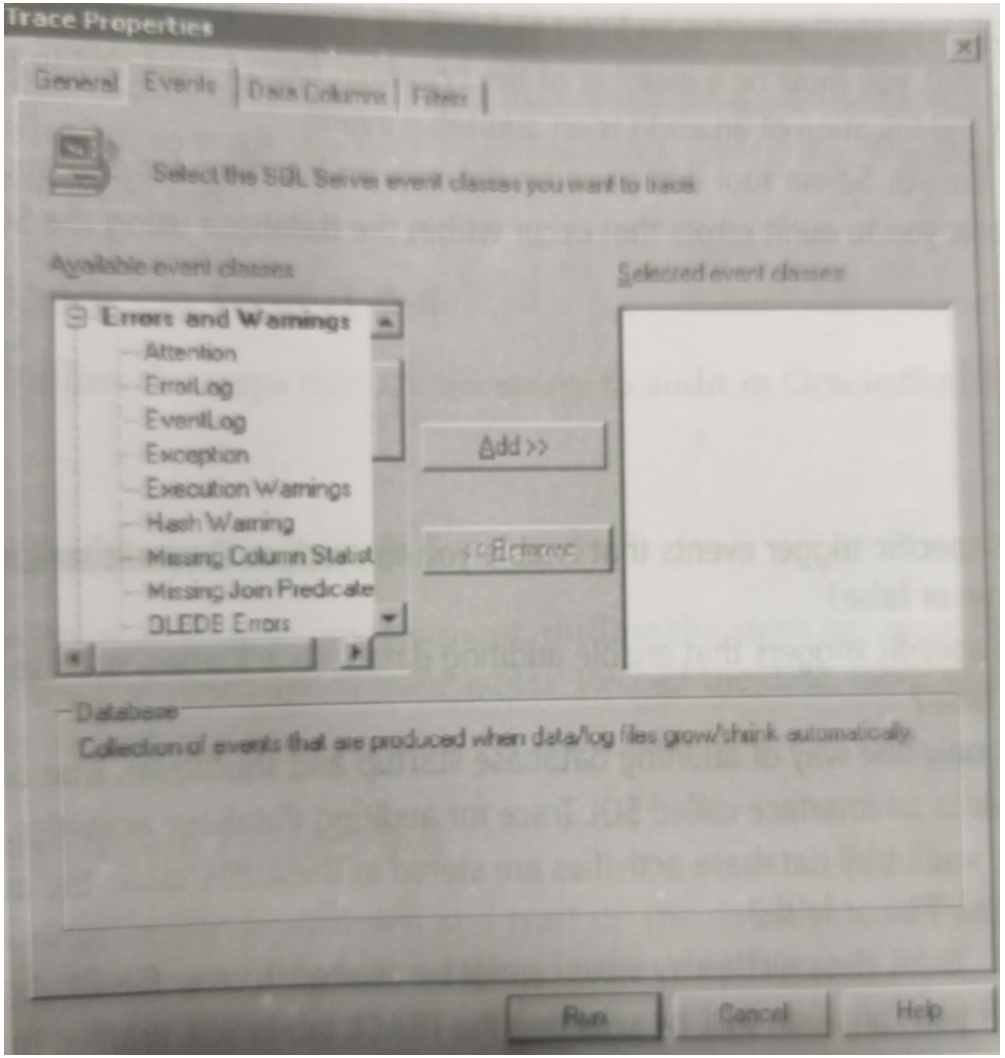
- ✓ To audit operations to the database files, select events under the Database category as shown in the figure



Auditing Server Activity with SQL Server 2000 ...

Database errors auditing with SQL Server

- ✓ To audit errors that occur within the database, select the events under the Errors and Warnings category on the Events tab of your trace, as shown in the figure



Security and Auditing Project Case Study

Introduction

- ✓ A DB developer is assigned to new database application project and is asked to develop an auditing scheme to comply with the industry standards
- ✓ Developers often face this problem
- ✓ DBA are often asked to provide an effective data security and auditing design
- ✓ The case studies follow require you to use these concepts, methods, and techniques to solve data accessibility
- ✓ This cases can be implemented in either ORACLE or SQL Server

Security and Auditing Project Case Study

CASE 1 : Developing an Online Database

- ✓ A new dot-com has decided to launch an affiliated Web site, specifically for individuals interested in database issues.
- ✓ The main mission of the Web site is to provide a forum for database technical tips, issues, and scripts.
- ✓ The CIO and his technical team held a meeting to draft the requirements for the new web site and decided that it would include the following.
  - Technical documents
  - A forum where members can exchange ideas and share experiences
  - Online access
  - A tips section
  - Technical support for error messages

Security and Auditing Project Case Study

- ✓ Immediately after the meeting, the newly appointed project manager asks you to implement security for the site.
- ✓ The manager mentions that the security of a public database is so important that the CIO himself / herself has outlined the security requirements, as follows
- ✓ The online DB will have 10 public host database accounts that allow multiple sessions
- ✓ The password of a public host account must be reset to its original setting whenever disconnects or logoffs occur
- ✓ The maximum duration for a session is 45 minutes
- ✓ Allocations will be set on memory and CPU

Security and Auditing Project Case Study

- ✓ Storage for each public host account must be limited to 1 MB
- ✓ The public host accounts will have privileges to create the most common database objects
- ✓ All newly created database objects must be removed before logoff
- ✓ The database must have the default human resources user account enabled.
- ✓ When number of logs onto the database, all session information, such as IP address, terminal , user session information must be recorded for future analysis.

Note : You may add other security auditing features, as long as you do not overlook any of the requirements in this list

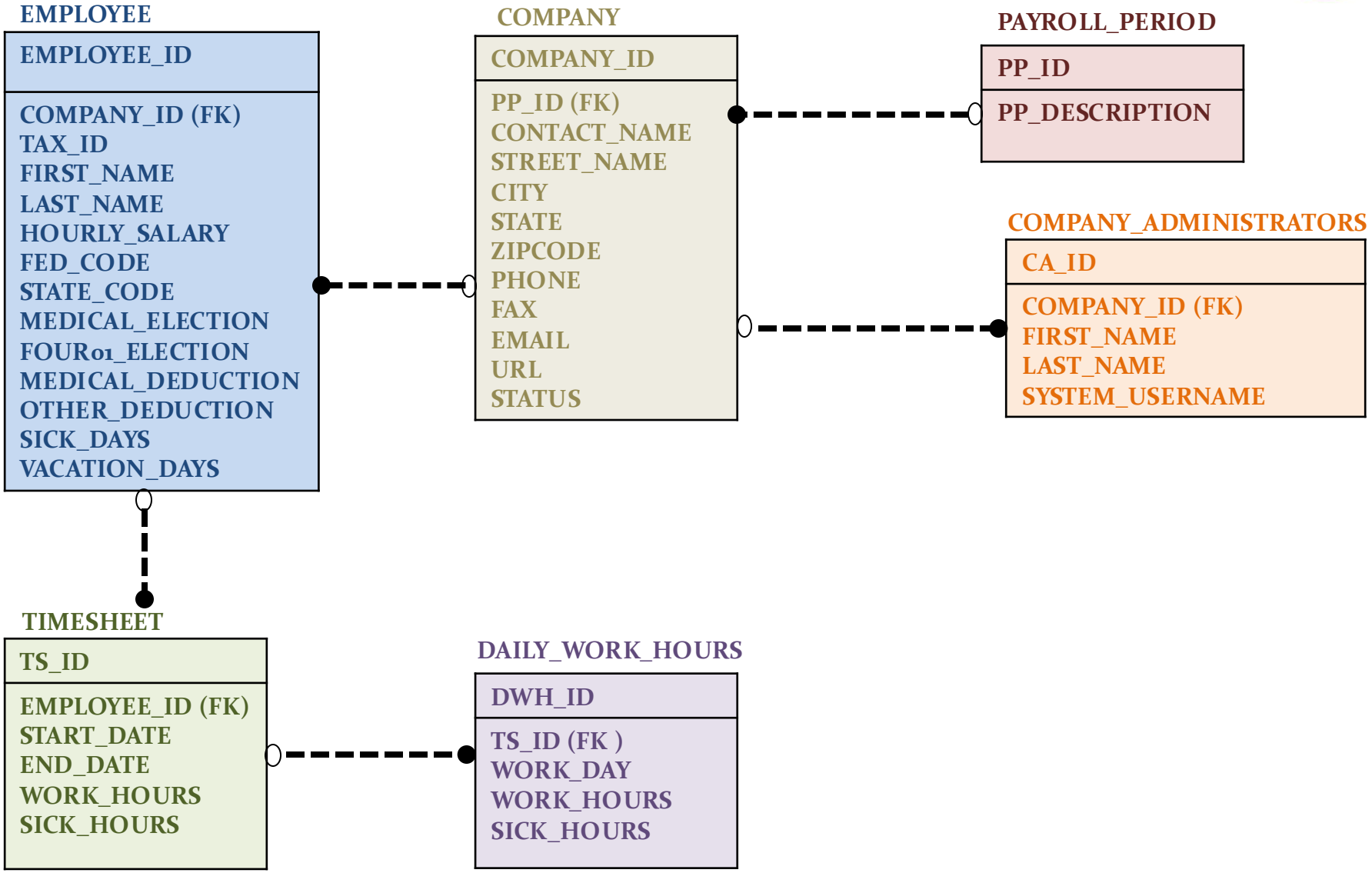
Security and Auditing Project Case Study

Case 2 : Taking Care of Payroll

- ✓ Acme Payroll Systems is a small payroll services company that has been in business for two years and has had only one major customer
- ✓ Suddenly, it lands a contract with another large corporation
- ✓ If the company hired you as Database consultant to design and implement a virtual private database for the existing payroll application.
- ✓ The main objective of the virtual private database feature is allow each client to administer his own payroll data without violating the privacy of other clients.

Security and Auditing Project Case Study

The given figure represents the payroll application model for case 2



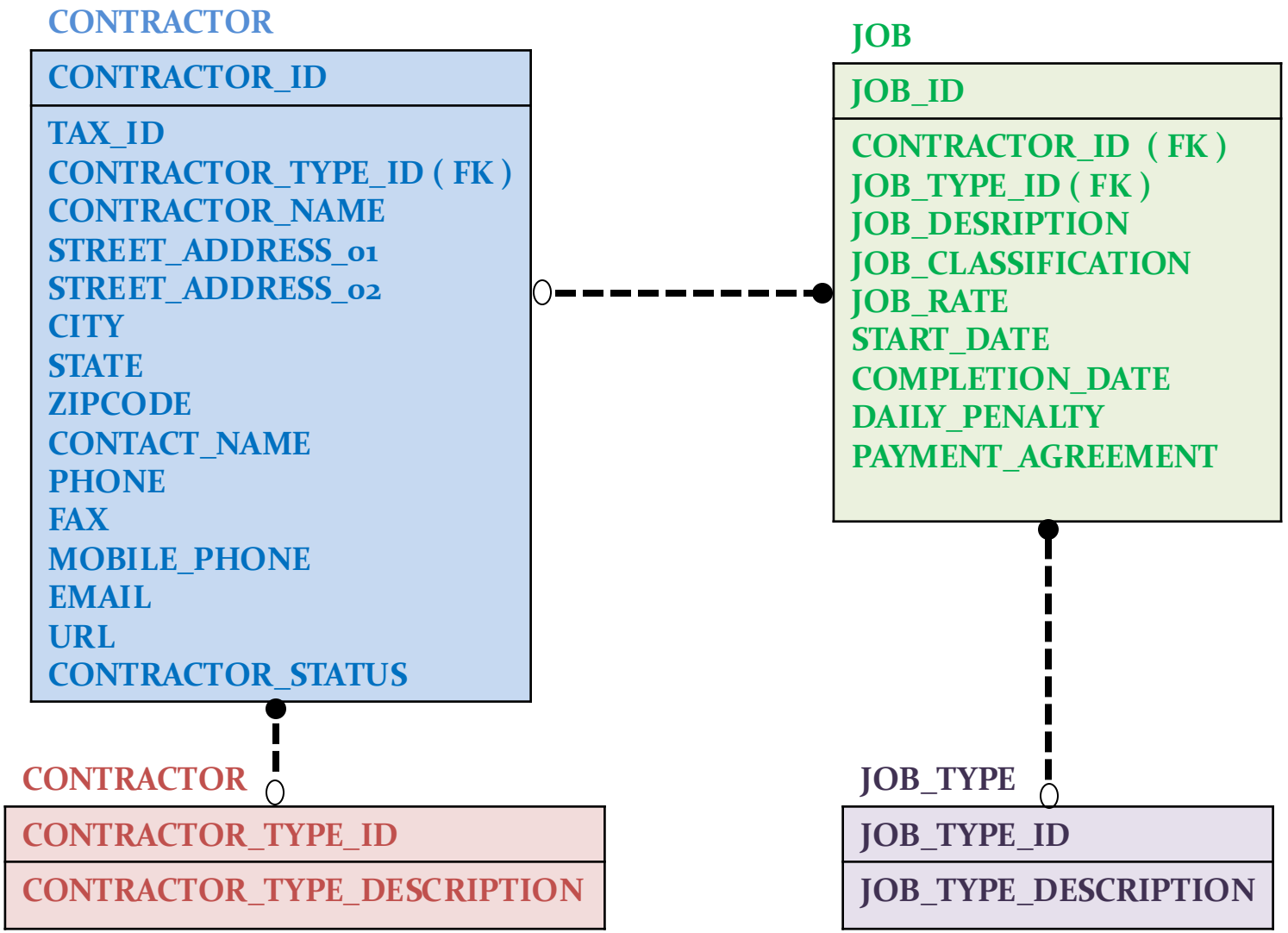
Security and Auditing Project Case Study

Case 3 : Tracking Town Contracts

- ✓ A small town has hired you as a database specialist on contract
- ✓ Your job is to develop a new database application to keep track of the jobs awarded to different contractors
- ✓ All town hall employees will use the application
- ✓ After several interviews with clerks and managers , you found out that a prior attempt at application development by a consulting company resulted in a draft of an entity – relationship (ER) diagram
- ✓ The ER diagram depicts all the required information about the contractors and the awarded jobs.

Security and Auditing Project Case Study

The given figure presents Contractor job data model for case 3



Security and Auditing Project Case Study

- ✓ During your meeting with the project manager for this application , you are asked to design an application with the following capabilities

  - Track all changes made to the application data
  - Obtain the approval of project manager before accepting any contract job for more than $10,000
  - Alert the project manager whenever an awarded job is modified to a value greater than $10,000
  - Implement three levels of security
  - The DEPARTMENT CLERK level allows clerks to add and update records
  - The DEPARTMENT MANAGER level allows clerks to add, update, delete and approve records
  - The EXTERNAL CLERK level allows employees outside the department only to view data.

Security and Auditing Project Case Study

Case 4 : Tracking Database Changes

- ✓ A friend recommended you to the company he/she works for
- ✓ The need your help to solve a series of database and application violations
- ✓ When you meet with the hiring manager, he/she explains that there has been a series of inexplicable, suspicious activities on the applications and production databases
- ✓ The company wants to know
  - Who accessed these databases?
  - Who modified data?
  - Who changed the data structure?

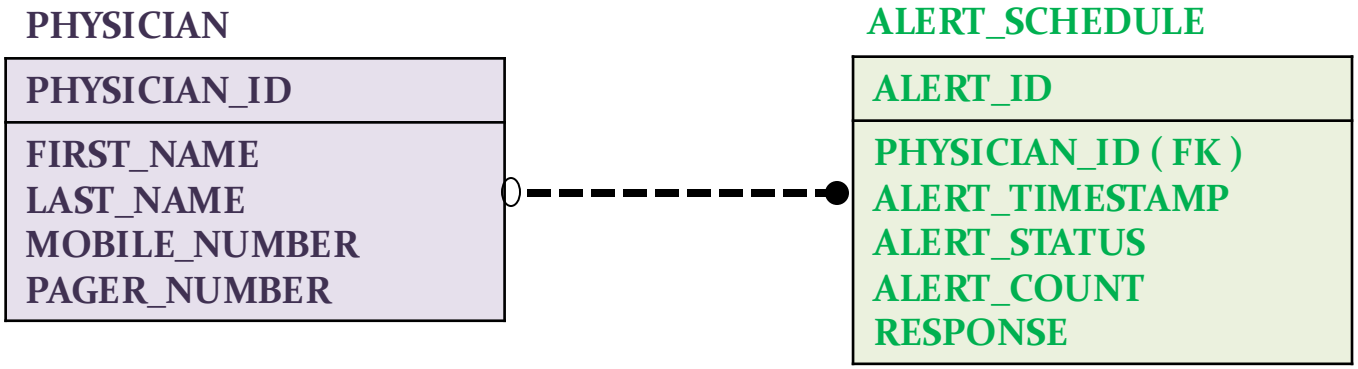
Security and Auditing Project Case Study

- ✓ Also the company want to have an audit trail for all these activities but that company was not interested in historical changes trail
- ✓ As a consultant, your job is to design an audit model to meet these requirements
- ✓ The following is the summary of the project requirements
  - Audit of database connections
  - Audit trail of users that are performing DML operations
  - Audit trail of users that are modifying structures of the application schema tables

Security and Auditing Project Case Study

Sample data model for case 4

- ✓ You may use two tables illustrated in the given figure as sample of application schema tables.



Security and Auditing Project Case Study

Case 5 : Developing a Secured Authorization Repository

- ✓ A small retail company has asked you to provide them with database security services
- ✓ The main requirement of this project is to create a security data model that will be used for by the central authorization module
- ✓ This model should include an auditing repository
- ✓ This model will store
  - Application users
  - Roles
  - Applications
  - Application Modules

Security and Auditing Project Case Study

- ✓ Your mission is to create an authorization data model with a relevant auditing repository
- ✓ The following is a summary of the project security requirements
  - There must be one database user account for the application schema owner
  - Database – assigned roles are not followed
  - There must be application roles only
  - Each application use is assigned to application modules
  - Each application user is assigned a security level that indicates the type of operations the user can perform within the application.
  - Operations are READ,WRITE, DELETE and ADMINISTER
  - Passwords must be stored within the designed security module
  - Each user has a logon identification number to the application
  - The security model should have the flexibility to logically lock, disable and remove accounts
  - Application accounts must have an activation date and expiry date

Security and Auditing Project Case Study

- ✓ The security module must be coupled with an auditing module that meets these auditing requirements
  - It must have an audit trail of the date and time a user connects and disconnects from application
  - It must have an audit trail of application operations that includes the date and time operations were performed by the application user
  - It must have an audit trail of all activities and operations performed on the security module
  - The auditing module must be coupled with the security module

Note : You are provide only a design solution , not an implementation

References :

1) Hassan A. Afyouni, “Database Security and Auditing”, Third Edition, Cengage Learning, 2009
2) Charu C. Aggarwal, Philip S Yu, “Privacy Preserving Data Mining”: Models and Algorithms, Kluwer Academic Publishers, 2008
3) Ron Ben Natan, ”Implementing Database Security and Auditing”, Elsevier Digital Press, 2005.
4) http://adrem.ua.ac.be/sites/adrem.ua.ac.be/files/securitybook.pdf
5) www.docs.oracle.com