DATABASE SYSTEM LAB 09

Lab Tasks:

DML Trigger Task:

- Create a BEFORE UPDATE trigger that prevents any employee's salary from being reduced in the employees table. If the new salary is lower than the current one, display an error message.
- Create an AFTER UPDATE trigger that records any changes made to the salary column of the employees table in a separate table called salary_history. Include the employee_id, old_salary, new_salary, and the update_time.
- Create a trigger that fires BEFORE DELETE on the departments table. Insert the deleted records' information (department_id, name, and deletion time) into a table called deleted_departments.
- Write a BEFORE INSERT trigger on the employees table that prevents the insertion of a record if the email column is NULL or empty.

DDL Trigger Tasks:

- Create a BEFORE DROP trigger at the schema level that prevents the dropping of critical tables like employees or audit_log and display an error message.
- Write an AFTER CREATE trigger that logs every new table creation in the schema to a ddl_log table. Include the table_name, creation_date, and user who created the table.
- Create a BEFORE ALTER trigger that restricts any ALTER statements on tables between 6 PM and 8 AM. If the time is outside of business hours, display an error message.
- Create a BEFORE DROP trigger that logs every time an index is dropped in the schema, recording the index_name, table_name, and drop_time into an index_log table.

System/Database Trigger Task:

- Create an AFTER LOGON trigger at the database level that records every user's login time and username in a user_login_log table.
- Write a BEFORE SELECT trigger at the system level that logs whenever a sensitive table (like employees) is queried. Log the username, query_time, and table_name.
- Create a BEFORE LOGON trigger that prevents any users from logging into the database during system maintenance hours (e.g., from 2 AM to 4 AM) and display an error message.
- Create an AFTER DDL trigger that logs every DDL change (e.g., table creation, alteration, or drop) made in the schema. Store the action, object_type, object_name, and execution_time in a schema_audit_log table.

Instead of Trigger Task:

- Create an INSTEAD OF INSERT trigger on a view called active_employees_view that represents the employees table but only shows active employees. Ensure that insertions into the view are properly mapped to the underlying employees table.
- Create an INSTEAD OF UPDATE trigger on a emp_dep_view that allows updates to employees information in the view, and ensure the updates are propagated correctly to both the employees and departments tables.