# Phase VII: Advanced Database Programming and Auditing

**Project Name**: Movies Management System  
**User Schema**: GRPA\_24918\_SHEJA\_MOVIESMS\_DB

## Objective

The objective of Phase VII is to enhance security and auditing by restricting unauthorized changes to sensitive data (such as ticket records) and by recording all attempted changes. This includes:

Preventing ticket data changes during weekdays and blocked (holiday) dates

Logging every attempt (successful or denied) into an audit table

## Step 1: Create the Blocked Calendar Table

We define a list of dates when any ticket operations should be blocked. These represent public holidays or system-wide blocked days.

### SQL Query

sql

CopyEdit

CREATE TABLE blocked\_calendar (

block\_date DATE PRIMARY KEY,

reason VARCHAR2(100)

);

### Sample Data

sql

CopyEdit

INSERT INTO blocked\_calendar VALUES (TO\_DATE('2025-06-17', 'YYYY-MM-DD'), 'Unity Day');INSERT INTO blocked\_calendar VALUES (TO\_DATE('2025-06-29', 'YYYY-MM-DD'), 'Independence Day');COMMIT;

## Step 2: Create the Audit Table

We log every INSERT, UPDATE, or DELETE attempt on the ticket table — whether it was allowed or blocked.

### SQL Query

sql

CopyEdit

CREATE TABLE ticket\_audit (

audit\_id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

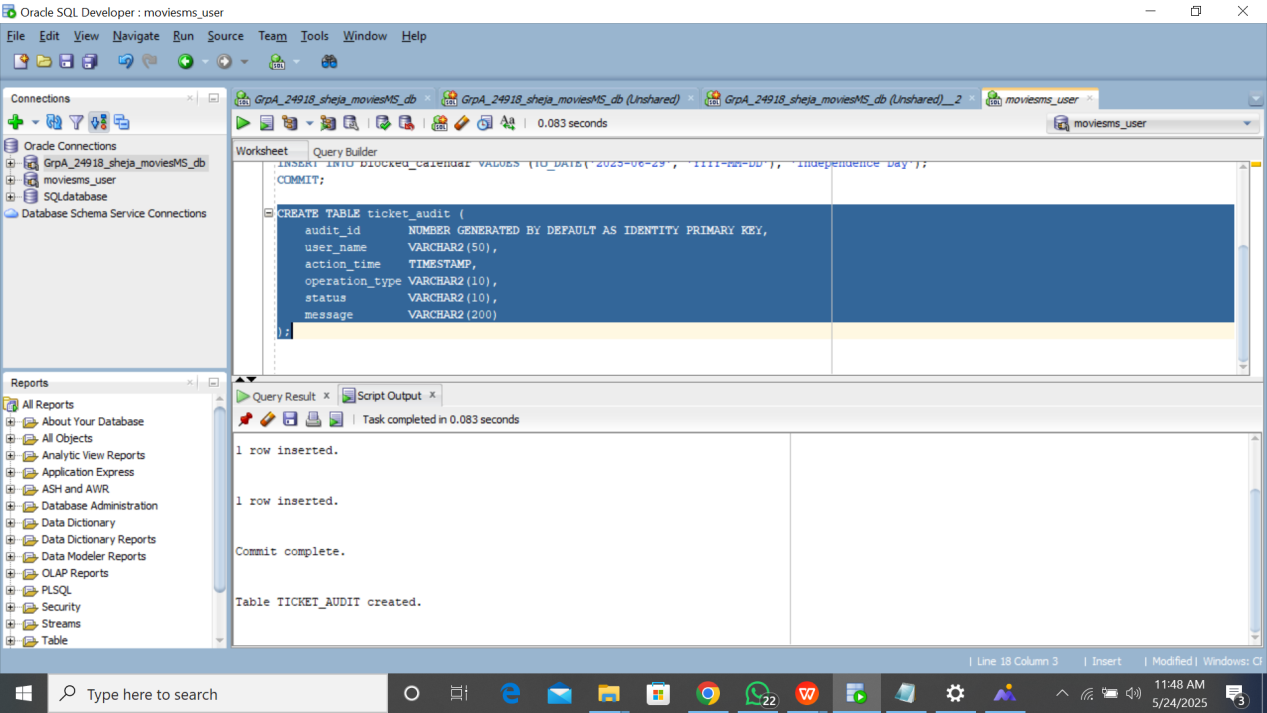
user\_name VARCHAR2(50),

action\_time TIMESTAMP,

operation\_type VARCHAR2(10),

status VARCHAR2(10),

message VARCHAR2(200)

);  
  
  


## Step 3: Create the Blocking and Logging Trigger

This trigger will:

Automatically run before any INSERT, UPDATE, or DELETE on ticket

Block operations on weekdays or blocked dates

Log the attempt with a message and user identity

### SQL Query

sql

CopyEdit

CREATE OR REPLACE TRIGGER trg\_block\_ticket\_dml

BEFORE INSERT OR UPDATE OR DELETE ON ticketFOR EACH ROWDECLARE

v\_day\_name VARCHAR2(10);

v\_user VARCHAR2(50);

v\_is\_blocked NUMBER := 0;

v\_operation VARCHAR2(10);BEGIN

SELECT TO\_CHAR(SYSDATE, 'DAY'), USER

INTO v\_day\_name, v\_user

FROM dual;

SELECT COUNT(\*) INTO v\_is\_blocked

FROM blocked\_calendar

WHERE block\_date = TRUNC(SYSDATE);

IF INSERTING THEN

v\_operation := 'INSERT';

ELSIF UPDATING THEN

v\_operation := 'UPDATE';

ELSIF DELETING THEN

v\_operation := 'DELETE';

END IF;

IF v\_day\_name IN ('MONDAY ', 'TUESDAY ', 'WEDNESDAY', 'THURSDAY ', 'FRIDAY ')

OR v\_is\_blocked > 0 THEN

INSERT INTO ticket\_audit(user\_name, action\_time, operation\_type, status, message)

VALUES (

v\_user,

CURRENT\_TIMESTAMP,

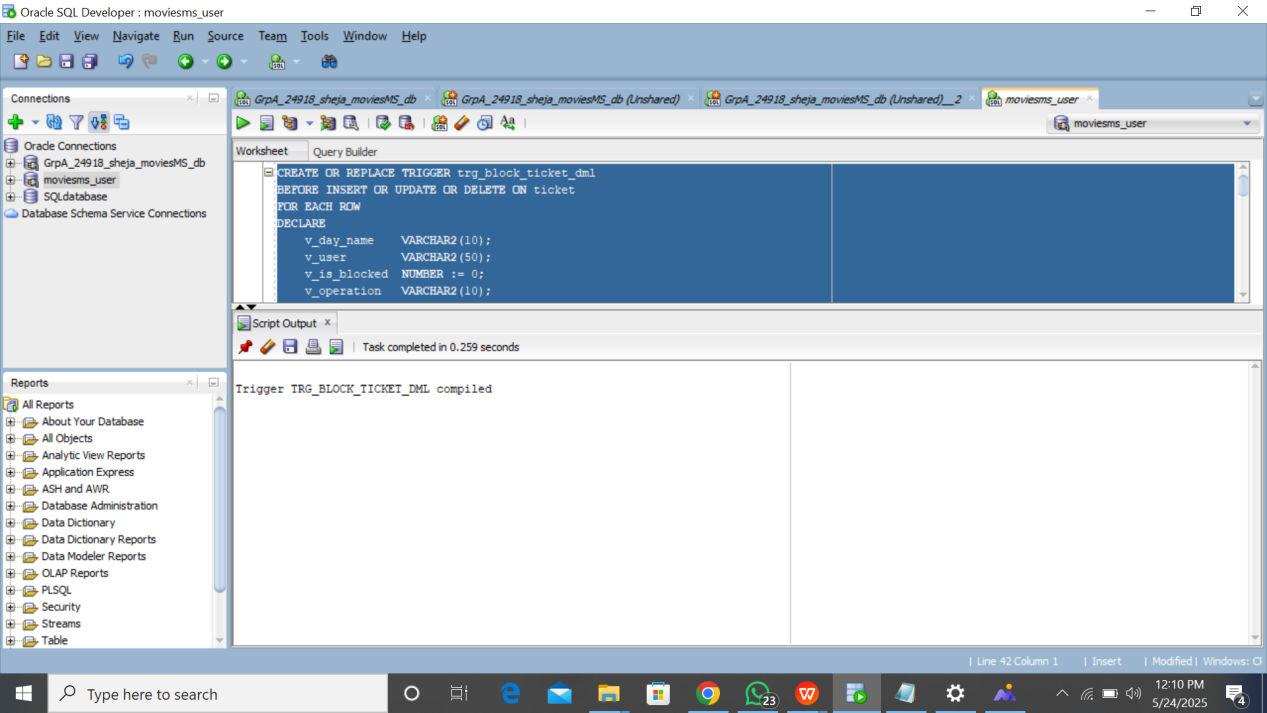
v\_operation,

'denied',

'DML operation blocked due to weekday or blocked date.'

);

RAISE\_APPLICATION\_ERROR(-20001, 'DML operations are blocked on weekdays or blocked dates.');

END IF;END;/  
  


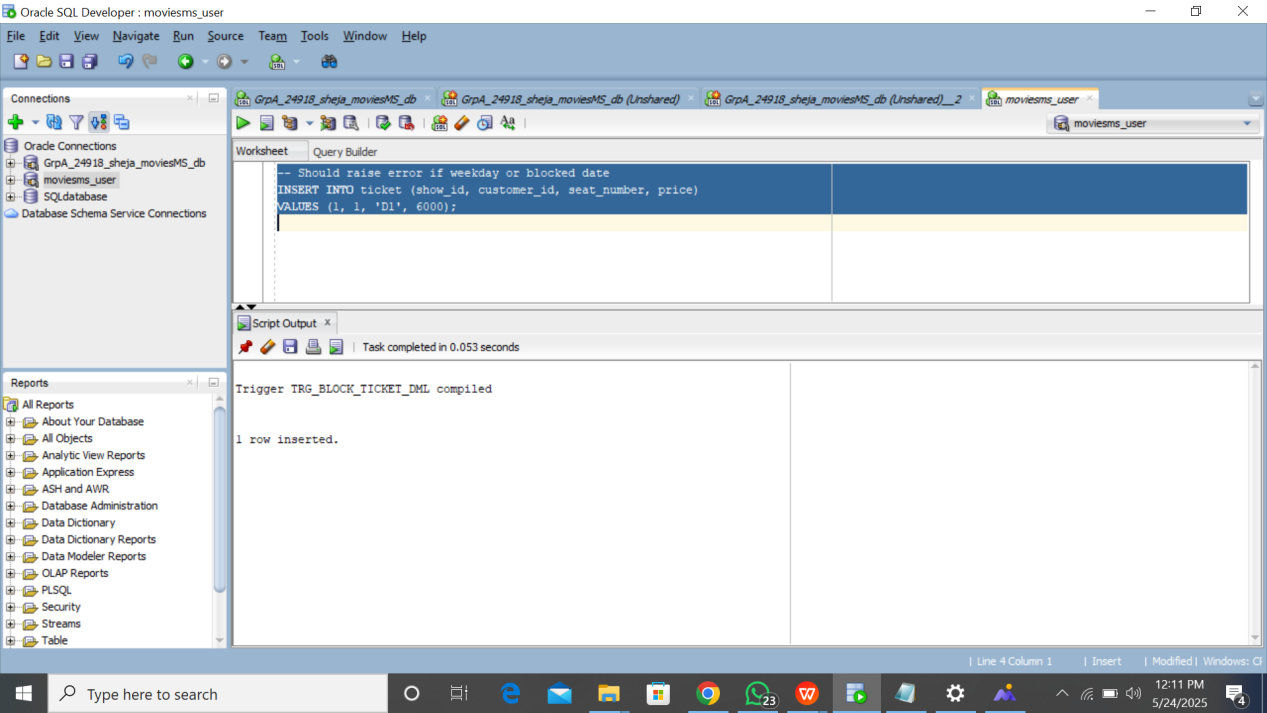
## Step 4: Testing and Validation

### Positive Case

Test inserting a ticket on a non-blocked **weekend**:

sql

CopyEdit

INSERT INTO ticket (show\_id, customer\_id, seat\_number, price)VALUES (1, 1, 'B1', 5000);  
  


Expected Result: ✅ Success, and an entry is added in ticket\_audit with status allowed.

### Negative Case

Test inserting a ticket on a **blocked weekday** or **holiday**:

sql

CopyEdit

-- This will fail if run on a blocked dateINSERT INTO ticket (show\_id, customer\_id, seat\_number, price)VALUES (1, 2, 'B2', 5500);

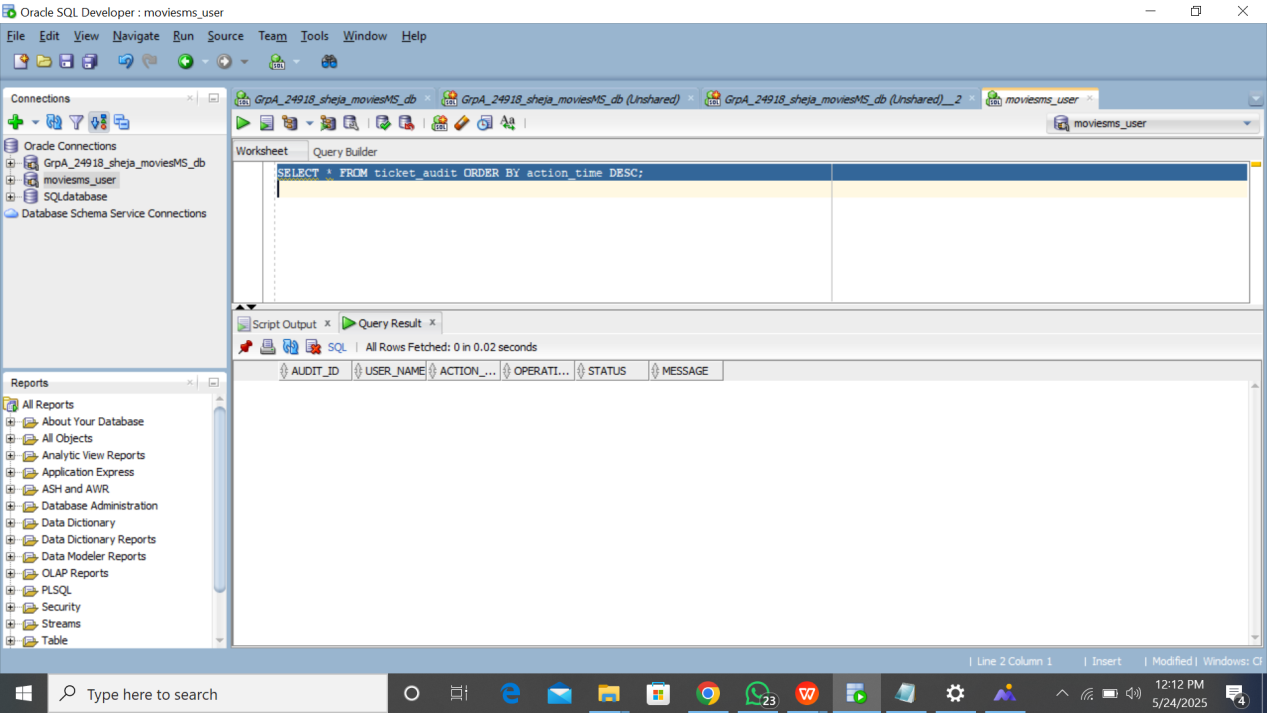
Expected Result: Error ORA-20001, and an entry is added to ticket\_audit with status denied.

## 🔎 View Audit Log

To confirm the auditing works correctly:

sql

CopyEdit

SELECT \* FROM ticket\_audit ORDER BY action\_time DESC;  
  


## Summary

In this phase, we successfully:

Created a list of blocked operational dates

Implemented a trigger that blocks and logs any ticket data changes during those periods

Ensured that all changes are traceable by username, action, and status

Maintained all logic in the schema GRPA\_24918\_SHEJA\_MOVIESMS\_DB