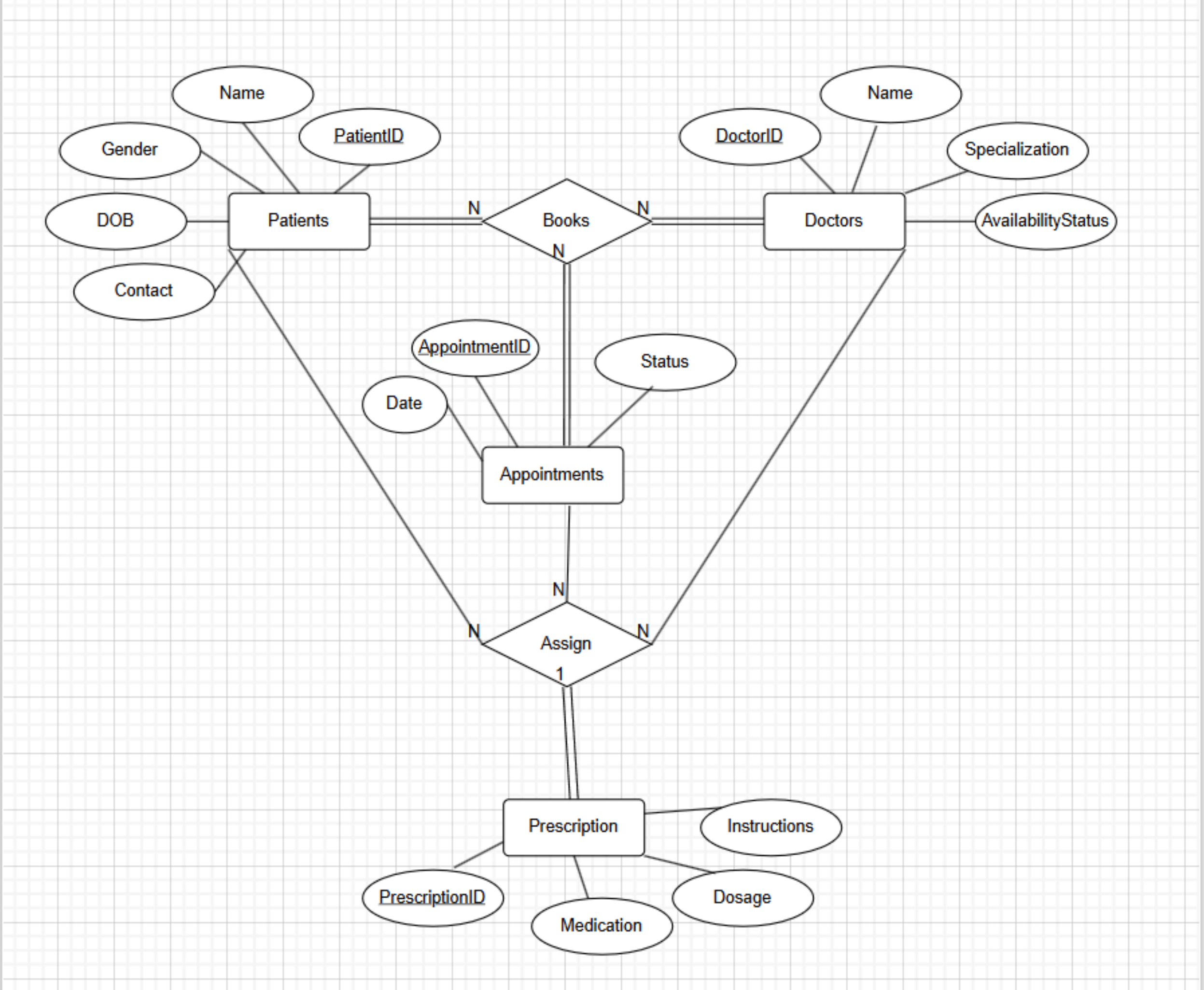
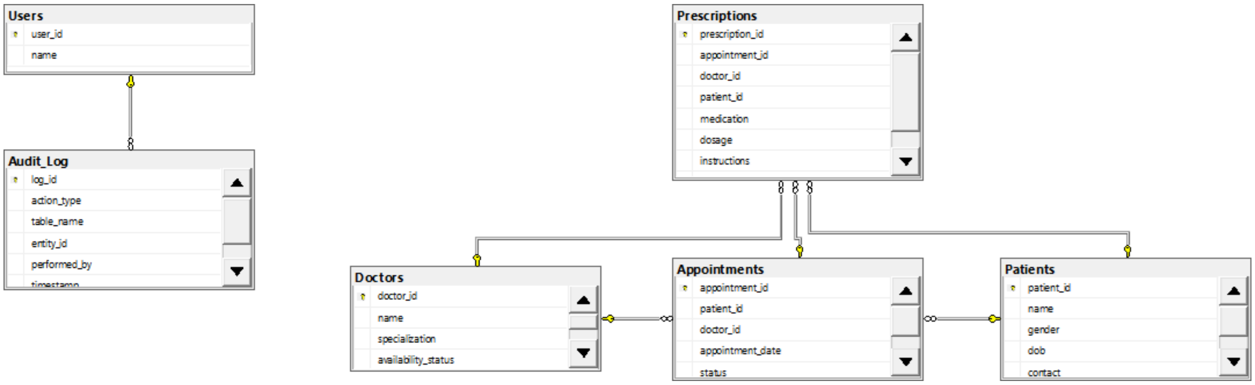
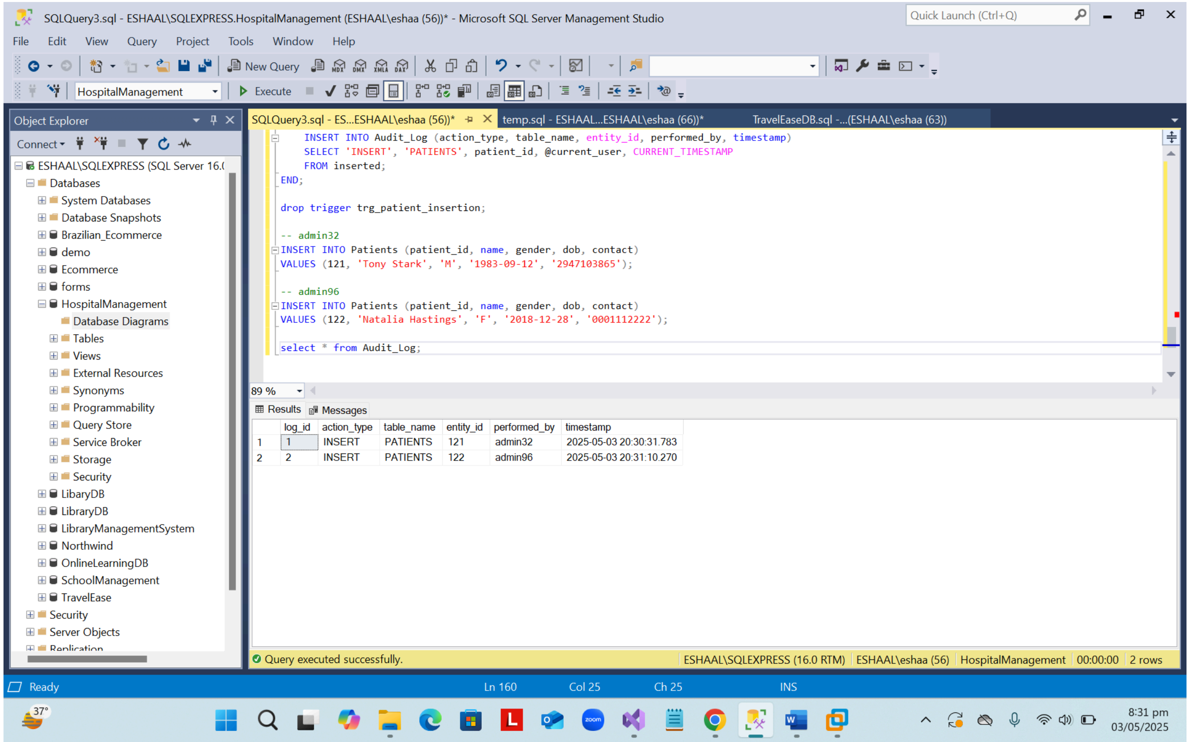
DATABASE SYSTEMS ASSIGNMENT3, 23I-0696 FATIMA ISHTIAQ, 23I-0732 ESHAAL MALIK





—---------- INSERT TRIGGER ON PATIENTS TABLE -------------

```
CREATE TRIGGER trg_patient_insertion
ON Patients
AFTER INSERT
AS
BEGIN
   DECLARE @current_user VARCHAR(100);

   -- Assign the current user performing the action
   --SET @current_user = 'admin32';  -- or SYSTEM_USER depending on your needs
         SET @current_user = 'admin96';

   -- Insert into the Audit_Log table
   INSERT INTO Audit_Log (action_type, table_name, entity_id, performed_by, timestamp)
   SELECT 'INSERT', 'PATIENTS', patient_id, @current_user, CURRENT_TIMESTAMP
   FROM inserted;
END;

drop trigger trg_patient_insertion;

-- admin32
INSERT INTO Patients (patient_id, name, gender, dob, contact)
VALUES (121, 'Tony Stark', 'M', '1983-09-12', '2947103865');

-- admin96
INSERT INTO Patients (patient_id, name, gender, dob, contact)
VALUES (122, 'Natalia Hastings', 'F', '2018-12-28', '0001112222');
```



```
    INSERT INTO Audit_Log (action_type, table_name, entity_id, performed_by, timestamp)
    SELECT 'INSERT', 'PATIENTS', patient_id, @current_user, CURRENT_TIMESTAMP
    FROM inserted;
END;

drop trigger trg_patient_insertion;

-- admin32
INSERT INTO Patients (patient_id, name, gender, dob, contact)
VALUES (121, 'Tony Stark', 'M', '1983-09-12', '2947103865');

-- admin96
INSERT INTO Patients (patient_id, name, gender, dob, contact)
VALUES (122, 'Natalia Hastings', 'F', '2018-12-28', '0001112222');

select * from Audit_Log;
```

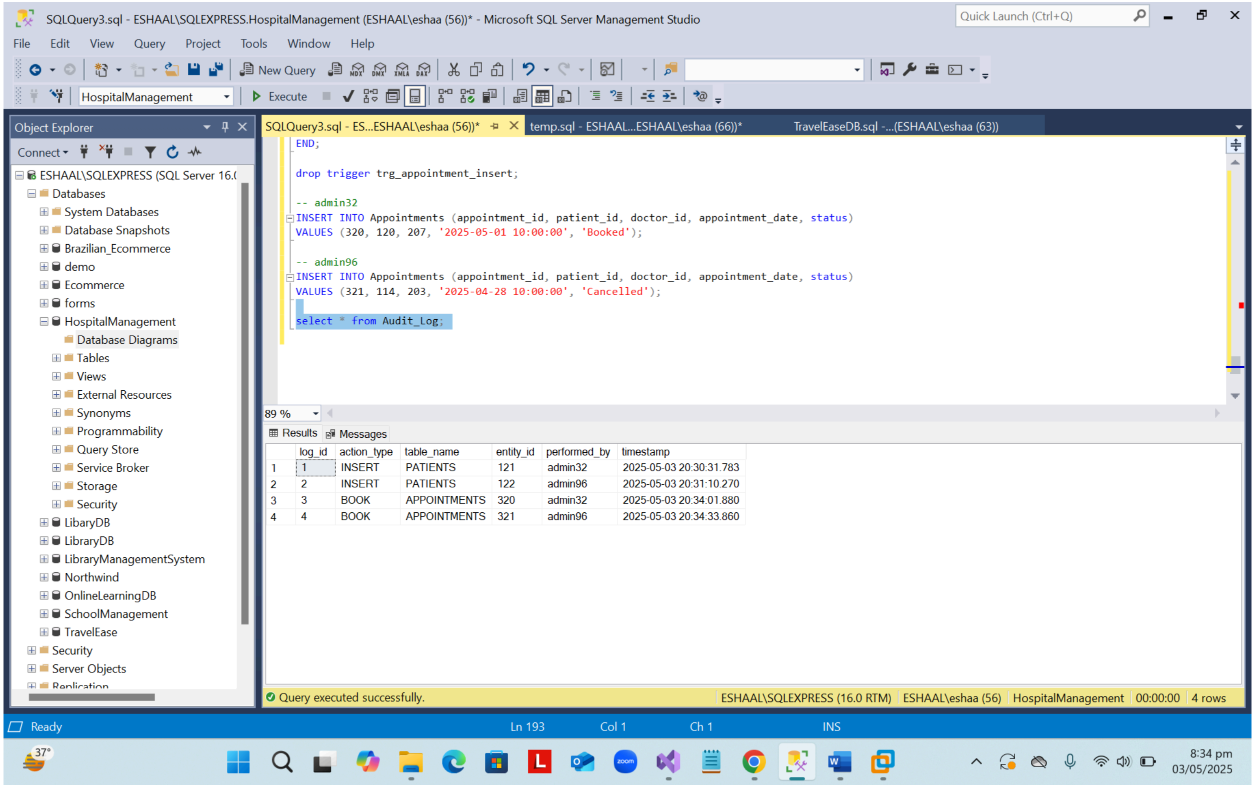
	log_id	action_type	table_name	entity_id	performed_by	timestamp
1	1	INSERT	PATIENTS	121	admin32	2025-05-03 20:30:31.783
2	2	INSERT	PATIENTS	122	admin96	2025-05-03 20:31:10.270

Query executed successfully.

--------- INSERT TRIGGER ON APPOINTMENTS TABLE —-------

```
CREATE TRIGGER trg_appointment_insert
ON Appointments
AFTER INSERT
AS
BEGIN
   DECLARE @current_user VARCHAR(100);

   -- Assign the current user performing the action
   --SET @current_user = 'admin32';
         SET @current_user = 'admin96';

   -- Log the booking action in the Audit_Log table
   INSERT INTO Audit_Log (action_type, table_name, entity_id, performed_by, timestamp)
   SELECT 'BOOK', 'APPOINTMENTS', appointment_id, @current_user,
CURRENT_TIMESTAMP
   FROM inserted;
END;
```

```
-- admin32
INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (320, 120, 207, '2025-05-01 10:00:00', 'Booked');

-- admin96
INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (321, 114, 203, '2025-04-28 10:00:00', 'Cancelled');

select * from Audit_Log;
```

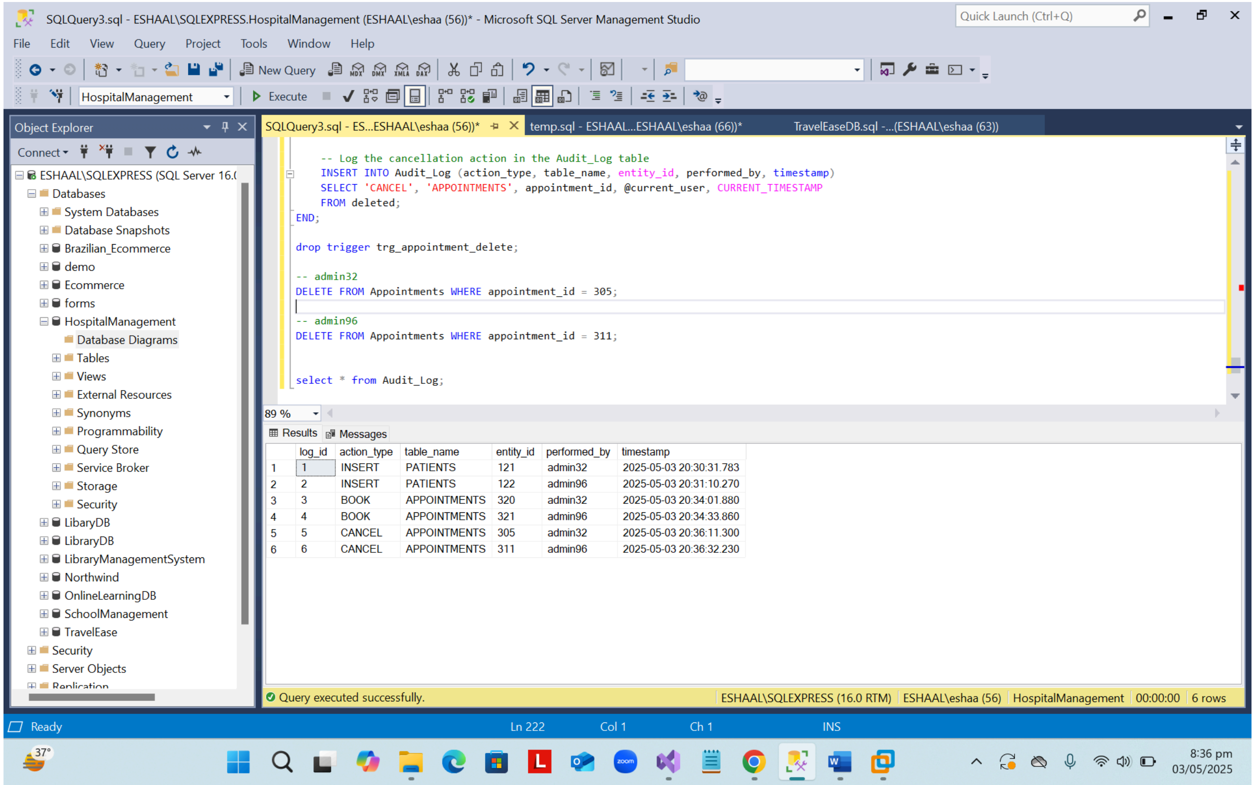


```
END;

drop trigger trg_appointment_insert;

-- admin32
INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (320, 120, 207, '2025-05-01 10:00:00', 'Booked');

-- admin96
INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (321, 114, 203, '2025-04-28 10:00:00', 'Cancelled');

select * from Audit_Log;
```

	log_id	action_type	table_name	entity_id	performed_by	timestamp
1	1	INSERT	PATIENTS	121	admin32	2025-05-03 20:30:31.783
2	2	INSERT	PATIENTS	122	admin96	2025-05-03 20:31:10.270
3	3	BOOK	APPOINTMENTS	320	admin32	2025-05-03 20:34:01.880
4	4	BOOK	APPOINTMENTS	321	admin96	2025-05-03 20:34:33.860

Query executed successfully.

```
---------------------------------------------------------------------------------
--------- DELETE TRIGGER ON APPOINTMENTS TABLE --------
---------------------------------------------------------------------------------
CREATE TRIGGER trg_appointment_delete
ON Appointments
AFTER DELETE
AS
BEGIN
    -- Declare the current_user variable for this trigger only
    DECLARE @current_user VARCHAR(100);
    -- Assign the current user performing the action
    --SET @current_user = 'admin32';
        SET @current_user = 'admin96';

    -- Log the cancellation action in the Audit_Log table
    INSERT INTO Audit_Log (action_type, table_name, entity_id, performed_by, timestamp)
    SELECT 'CANCEL', 'APPOINTMENTS', appointment_id, @current_user,
CURRENT_TIMESTAMP
    FROM deleted;
END;


-- admin32
DELETE FROM Appointments WHERE appointment_id = 305;

-- admin96
DELETE FROM Appointments WHERE appointment_id = 311;


select * from Audit_Log;
```



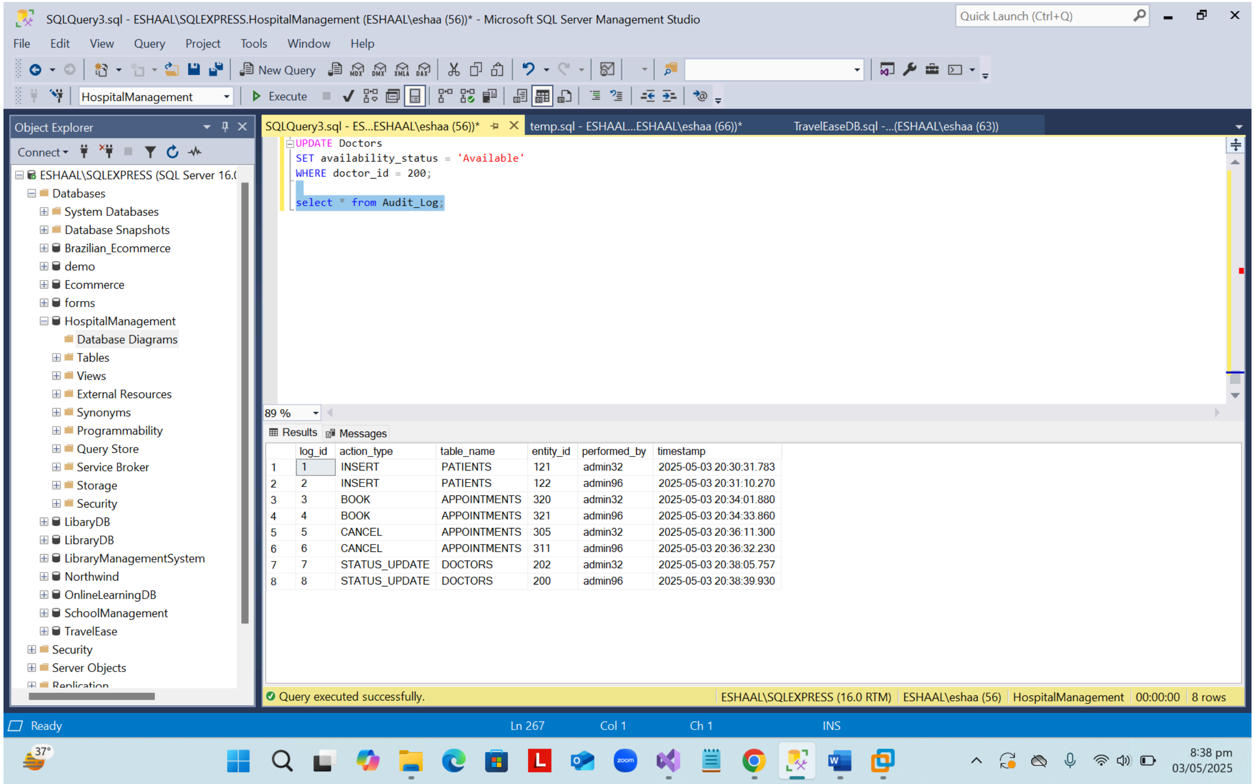
SQLQuery3.sql - ESHAAL\SQLEXPRESS.HospitalManagement (ESHAAL\eshaa (56))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
HospitalManagement
Execute
Object Explorer
SQLQuery3.sql - ES...ESHAAL\eshaa (56))*
temp.sql - ESHAAL...ESHAAL\eshaa (66))*
TravelEaseDB.sql -...(ESHAAL\eshaa (63))
89 %
Results
Messages

```
    -- Log the cancellation action in the Audit_Log table
    INSERT INTO Audit_Log (action_type, table_name, entity_id, performed_by, timestamp)
    SELECT 'CANCEL', 'APPOINTMENTS', appointment_id, @current_user, CURRENT_TIMESTAMP
    FROM deleted;
END;

drop trigger trg_appointment_delete;

-- admin32
DELETE FROM Appointments WHERE appointment_id = 305;

-- admin96
DELETE FROM Appointments WHERE appointment_id = 311;


select * from Audit_Log;
```

	log_id	action_type	table_name	entity_id	performed_by	timestamp
1	1	INSERT	PATIENTS	121	admin32	2025-05-03 20:30:31.783
2	2	INSERT	PATIENTS	122	admin96	2025-05-03 20:31:10.270
3	3	BOOK	APPOINTMENTS	320	admin32	2025-05-03 20:34:01.880
4	4	BOOK	APPOINTMENTS	321	admin96	2025-05-03 20:34:33.860
5	5	CANCEL	APPOINTMENTS	305	admin32	2025-05-03 20:36:11.300
6	6	CANCEL	APPOINTMENTS	311	admin96	2025-05-03 20:36:32.230

Query executed successfully. ESHAAL\SQLEXPRESS (16.0 RTM) ESHAAL\eshaa (56) HospitalManagement 00:00:00 6 rows

```
--------------------------------------------------------------------------------
------------ UPDATE TRIGGER ON DOCTORS TABLE -------------
--------------------------------------------------------------------------------
CREATE TRIGGER trg_doctor_update
ON Doctors
AFTER UPDATE
AS
BEGIN
   DECLARE @current_user VARCHAR(100);

   -- Assign the current user performing the action
   --SET @current_user = 'admin32';
         SET @current_user = 'admin96';

   -- Directly log the change if availability_status is updated
   INSERT INTO Audit_Log (action_type, table_name, entity_id, performed_by, timestamp)
   SELECT 'STATUS_UPDATE', 'DOCTORS', doctor_id, @current_user,
CURRENT_TIMESTAMP
   FROM inserted
   WHERE EXISTS (
      SELECT 1
      FROM deleted
      WHERE deleted.availability_status != inserted.availability_status
      AND inserted.doctor_id = deleted.doctor_id
   );
END;


-- admin32, 202 originally unavailable, changing it to available
UPDATE Doctors
SET availability_status = 'Available'
WHERE doctor_id = 202;

-- admin96, 200 originally unavailable, changing it to available
UPDATE Doctors
SET availability_status = 'Available'
WHERE doctor_id = 200;

select * from Audit_Log;
```



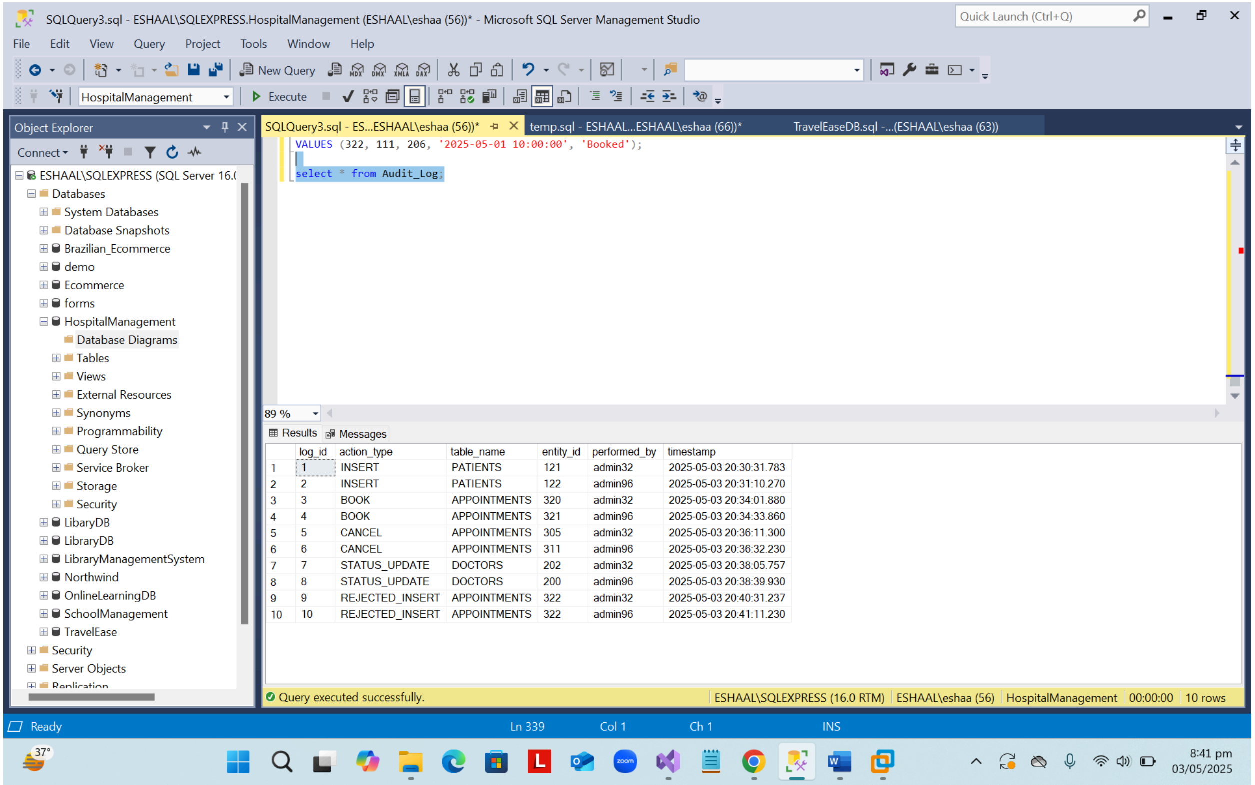
```
UPDATE Doctors
SET availability_status = 'Available'
WHERE doctor_id = 200;

select * from Audit_Log;
```

	log_id	action_type	table_name	entity_id	performed_by	timestamp
1	1	INSERT	PATIENTS	121	admin32	2025-05-03 20:30:31.783
2	2	INSERT	PATIENTS	122	admin96	2025-05-03 20:31:10.270
3	3	BOOK	APPOINTMENTS	320	admin32	2025-05-03 20:34:01.880
4	4	BOOK	APPOINTMENTS	321	admin96	2025-05-03 20:34:33.860
5	5	CANCEL	APPOINTMENTS	305	admin32	2025-05-03 20:36:11.300
6	6	CANCEL	APPOINTMENTS	311	admin96	2025-05-03 20:36:32.230
7	7	STATUS_UPDATE	DOCTORS	202	admin32	2025-05-03 20:38:05.757
8	8	STATUS_UPDATE	DOCTORS	200	admin96	2025-05-03 20:38:39.930

Query executed successfully. ESHAAL\SQLEXPRESS (16.0 RTM) ESHAAL\eshaa (56) HospitalManagement 00:00:00 8 rows

```
-------------------------------------------------------------------------------
—-- CONDITIONAL TRIGGER WITH STATUS VALIDATION —--
-------------------------------------------------------------------------------
CREATE TRIGGER trg_validate_appointment_insert
ON Appointments
AFTER INSERT
AS
BEGIN
    DECLARE @current_user VARCHAR(100);

    -- Assign the current user performing the action
    --SET @current_user = 'admin32';
          SET @current_user = 'admin96';

    -- Log if the insert is rejected due to doctor being unavailable
    INSERT INTO Audit_Log (action_type, table_name, entity_id, performed_by, timestamp)
    SELECT 'REJECTED_INSERT', 'APPOINTMENTS', appointment_id, @current_user,
CURRENT_TIMESTAMP
    FROM inserted
    WHERE EXISTS (
        SELECT 1
        FROM Doctors
        WHERE doctor_id = inserted.doctor_id
          AND availability_status = 'Unavailable'
    );

    -- Log if the insert is rejected due to patient having an active appointment on the same day
    INSERT INTO Audit_Log (action_type, table_name, entity_id, performed_by, timestamp)
    SELECT 'REJECTED_INSERT', 'APPOINTMENTS', appointment_id, @current_user,
CURRENT_TIMESTAMP
    FROM inserted
    WHERE EXISTS (
        SELECT 1
        FROM Appointments
        WHERE patient_id = inserted.patient_id
          AND status = 'Booked'
          AND appointment_date = inserted.appointment_date
    );

    -- Prevent the insert if any condition is met by signaling an error
   DELETE FROM Appointments
    WHERE appointment_id IN (SELECT appointment_id FROM inserted
        WHERE EXISTS (
            SELECT 1
            FROM Doctors
            WHERE doctor_id = inserted.doctor_id
              AND availability_status = 'Unavailable'
        ));

    DELETE FROM Appointments
    WHERE appointment_id IN (SELECT appointment_id FROM inserted
        WHERE EXISTS (
            SELECT 1
            FROM Appointments
            WHERE patient_id = inserted.patient_id
              AND status = 'Booked'
              AND appointment_date = inserted.appointment_date
        ));
END;


-- admin32
INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (322, 106, 209, '2025-05-01 10:00:00', 'Booked');

-- admin96
```

```
INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (322, 111, 206, '2025-05-01 10:00:00', 'Booked');

select * from Audit_Log
```



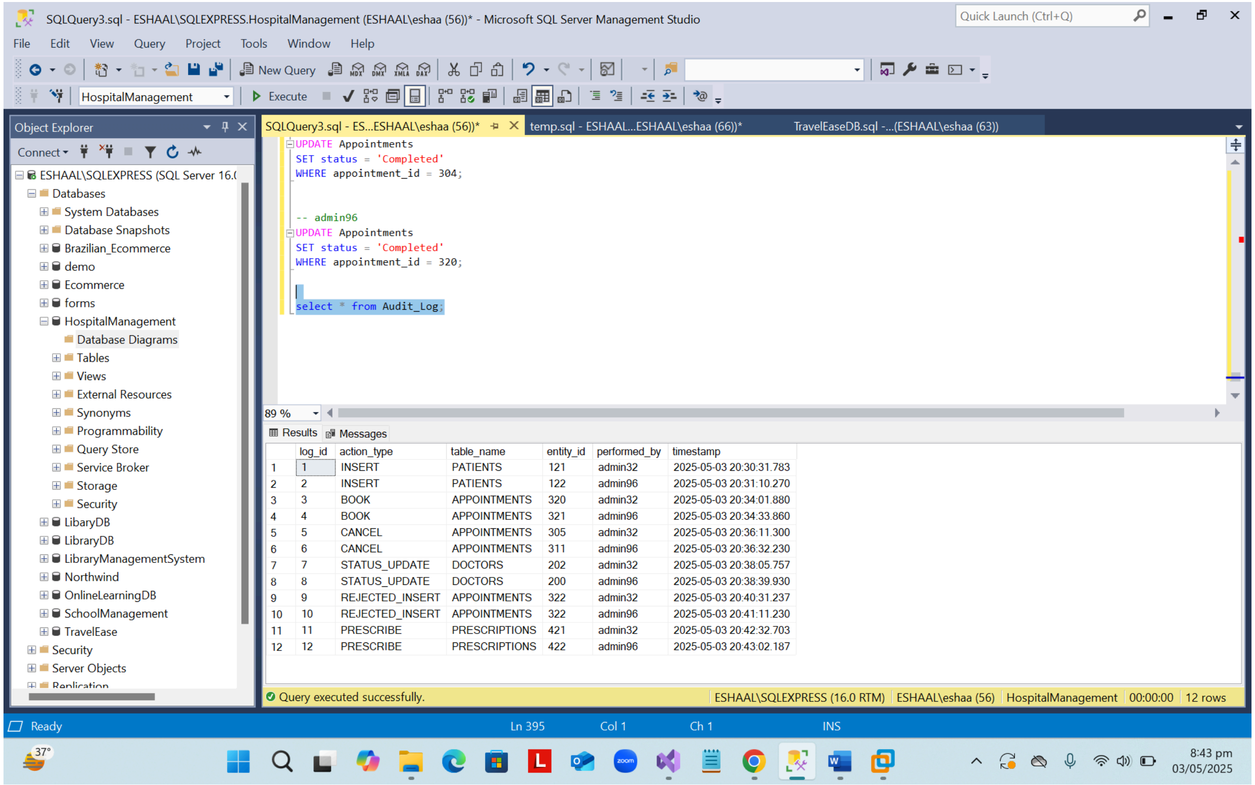
```
VALUES (322, 111, 206, '2025-05-01 10:00:00', 'Booked');

select * from Audit_Log;
```

	log_id	action_type	table_name	entity_id	performed_by	timestamp
1	1	INSERT	PATIENTS	121	admin32	2025-05-03 20:30:31.783
2	2	INSERT	PATIENTS	122	admin96	2025-05-03 20:31:10.270
3	3	BOOK	APPOINTMENTS	320	admin32	2025-05-03 20:34:01.880
4	4	BOOK	APPOINTMENTS	321	admin96	2025-05-03 20:34:33.860
5	5	CANCEL	APPOINTMENTS	305	admin32	2025-05-03 20:36:11.300
6	6	CANCEL	APPOINTMENTS	311	admin96	2025-05-03 20:36:32.230
7	7	STATUS_UPDATE	DOCTORS	202	admin32	2025-05-03 20:38:05.757
8	8	STATUS_UPDATE	DOCTORS	200	admin96	2025-05-03 20:38:39.930
9	9	REJECTED_INSERT	APPOINTMENTS	322	admin32	2025-05-03 20:40:31.237
10	10	REJECTED_INSERT	APPOINTMENTS	322	admin96	2025-05-03 20:41:11.230

Query executed successfully.

```
-------------------------------------------------------------------------------
----- TRIGGER FOR AUTOMATIC PRESCRIPTION ISSUE  -----
-------------------------------------------------------------------------------
CREATE TRIGGER trg_auto_prescription
ON Appointments
AFTER UPDATE
AS
BEGIN
    DECLARE @current_user VARCHAR(100);
    DECLARE @new_prescription_id INT;

    -- Assign the current user performing the action
    --SET @current_user = 'admin32';
          SET @current_user = 'admin96';

    -- Check if the appointment status has been updated to 'Completed'
```

```
  IF EXISTS (
    SELECT 1
    FROM inserted
    WHERE status = 'Completed'
  )
  BEGIN
    -- Get the maximum prescription_id and increment by 1
    SELECT @new_prescription_id = MAX(prescription_id) + 1
    FROM Prescriptions;

    -- Insert a placeholder prescription when the appointment is marked as 'Completed'
    INSERT INTO Prescriptions (prescription_id, appointment_id, doctor_id, patient_id,
medication, dosage, instructions, date_issued)
    SELECT @new_prescription_id, appointment_id, doctor_id, patient_id, 'Diagnosis
Pending', NULL, NULL, CURRENT_TIMESTAMP
    FROM inserted;

    -- Log the creation of the prescription in the Audit_Log table
    INSERT INTO Audit_Log (action_type, table_name, entity_id, performed_by, timestamp)
    SELECT 'PRESCRIBE', 'PRESCRIPTIONS', @new_prescription_id, @current_user,
CURRENT_TIMESTAMP
    FROM inserted;
  END
END;



-- admin32
UPDATE Appointments
SET status = 'Completed'
WHERE appointment_id = 304;


-- admin96
UPDATE Appointments
SET status = 'Completed'
WHERE appointment_id = 320;


select * from Audit_Log;
```



```
UPDATE Appointments
SET status = 'Completed'
WHERE appointment_id = 304;


-- admin96
UPDATE Appointments
SET status = 'Completed'
WHERE appointment_id = 320;


select * from Audit_Log;
```

	log_id	action_type	table_name	entity_id	performed_by	timestamp
1	1	INSERT	PATIENTS	121	admin32	2025-05-03 20:30:31.783
2	2	INSERT	PATIENTS	122	admin96	2025-05-03 20:31:10.270
3	3	BOOK	APPOINTMENTS	320	admin32	2025-05-03 20:34:01.880
4	4	BOOK	APPOINTMENTS	321	admin96	2025-05-03 20:34:33.860
5	5	CANCEL	APPOINTMENTS	305	admin32	2025-05-03 20:36:11.300
6	6	CANCEL	APPOINTMENTS	311	admin96	2025-05-03 20:36:32.230
7	7	STATUS_UPDATE	DOCTORS	202	admin32	2025-05-03 20:38:05.757
8	8	STATUS_UPDATE	DOCTORS	200	admin96	2025-05-03 20:38:39.930
9	9	REJECTED_INSERT	APPOINTMENTS	322	admin32	2025-05-03 20:40:31.237
10	10	REJECTED_INSERT	APPOINTMENTS	322	admin96	2025-05-03 20:41:11.230
11	11	PRESCRIBE	PRESCRIPTIONS	421	admin32	2025-05-03 20:42:32.703
12	12	PRESCRIBE	PRESCRIPTIONS	422	admin96	2025-05-03 20:43:02.187

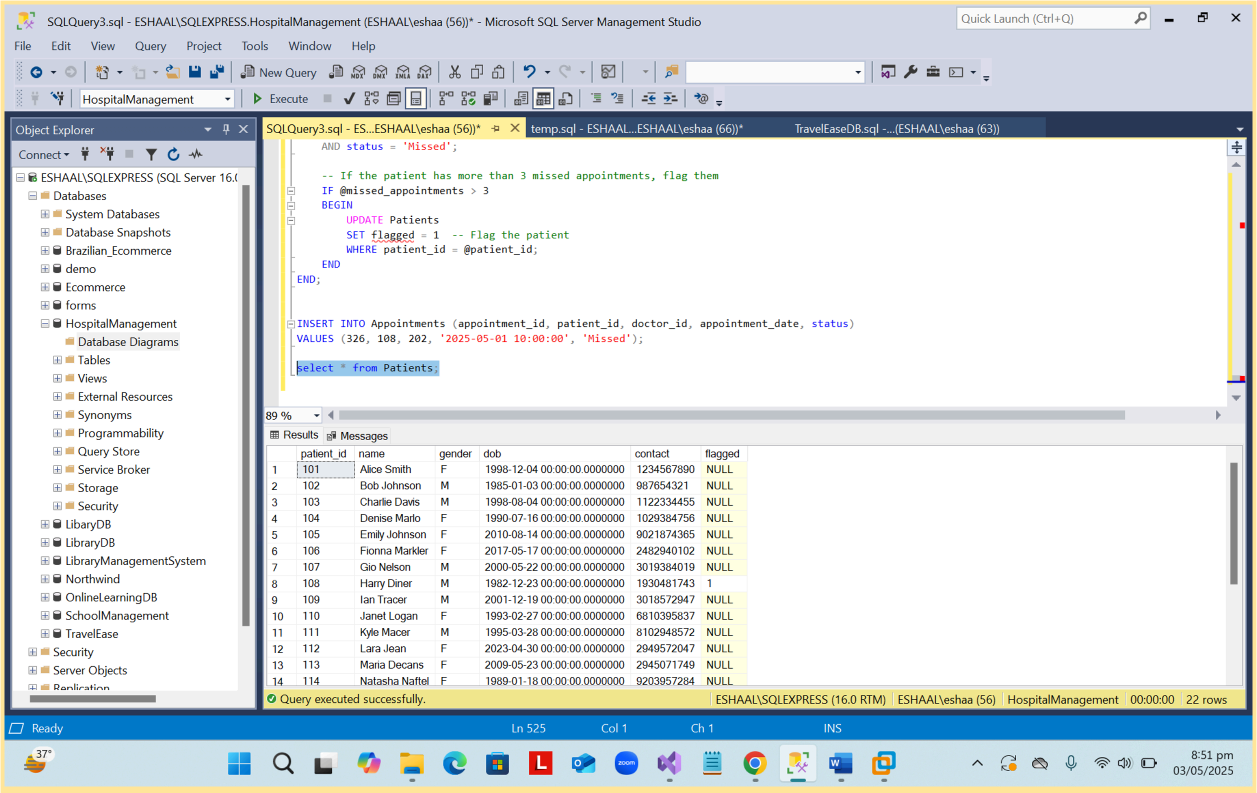
Query executed successfully. | ESHAAL\SQLEXPRESS (16.0 RTM) | ESHAAL\eshaa (56) | HospitalManagement | 00:00:00 | 12 rows

```
--------------------------------------------------------------------------------
—- TRIGGER TO PREVENT MULTIPLE APPOINTMENTS -----
--------------------------------------------------------------------------------

CREATE TRIGGER trg_prevent_multiple_appointments
ON Appointments
INSTEAD OF INSERT
AS
BEGIN
    DECLARE @patient_id INT;
    DECLARE @appointment_date DATETIME;
    DECLARE @current_user VARCHAR(100);

            SET @current_user = 'admin32';
            --SET @current_user = 'admin96';

    -- Get the values from the inserted table
    SELECT @patient_id = patient_id, @appointment_date = appointment_date FROM inserted;

    -- Check if the patient already has an active appointment on the same date
    IF EXISTS (
        SELECT 1
        FROM Appointments
        WHERE patient_id = @patient_id
        AND status = 'Booked'  -- Ensure status is 'Booked'
        AND appointment_date = @appointment_date
    )
    BEGIN
        -- Log the blocked attempt in the Audit_Log table
        INSERT INTO Audit_Log (action_type, table_name, entity_id, performed_by, timestamp)
        SELECT 'REJECTED_INSERT', 'APPOINTMENTS', appointment_id, @current_user,
CURRENT_TIMESTAMP
        FROM inserted;

        -- Prevent the insert and raise an error
        RAISERROR ('Patient already has an active appointment on this date.', 16, 1);
    END
    ELSE
    BEGIN
        -- If no active appointment exists, insert the new appointment
        INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date,
status)
        SELECT appointment_id, patient_id, doctor_id, appointment_date, status
        FROM inserted;
    END
END;

select * from Appointments;

-- admin32
INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (322, 119, 206, '2025-01-08 10:00:00', 'Booked');

select * from Audit_Log;

CREATE TRIGGER trg_log_appointment_insertion
ON Appointments
AFTER INSERT
AS
BEGIN
    DECLARE @current_user VARCHAR(100);
    DECLARE @new_appointment_id INT;

    -- Assign the current user performing the action
    SET @current_user = 'admin32';
        --SET @current_user = 'admin96';
    -- Log the new appointment insertion in the Audit_Log
    SELECT @new_appointment_id = appointment_id FROM inserted;

    -- Log the operation in Audit_Log (new values)
    INSERT INTO Audit_Log (action_type, table_name, entity_id, performed_by, timestamp)
    VALUES ('INSERT', 'APPOINTMENTS', @new_appointment_id, @current_user,
CURRENT_TIMESTAMP);
END;
```

SIMPLE LOGGING AND LOGGING PREVENTION

```
--------------------------------------------------------------------------------------------
------ CASCADING TRIGGERS WITH LOGIC CONFLICT  -------
--------------------------------------------------------------------------------------------

--------- 1 ---------
select * from Appointments;

select * from Doctors;

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (323, 108, 202, '2025-03-08 10:00:00', 'Missed');


INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (324, 108, 206, '2025-02-08 10:00:00', 'Missed');

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (325, 108, 200, '2025-01-08 10:00:00', 'Missed');


CREATE TRIGGER trg_flag_missed_appointments
ON Appointments
AFTER INSERT, UPDATE
AS
BEGIN
    DECLARE @patient_id INT;
    DECLARE @missed_appointments INT;
    -- Get the patient_id from the inserted or updated row
    SELECT @patient_id = patient_id FROM inserted;

    -- Count the number of missed appointments for the patient
    SELECT @missed_appointments = COUNT(*)
    FROM Appointments
    WHERE patient_id = @patient_id
    AND status = 'Missed';

    -- If the patient has more than 3 missed appointments, flag them
    IF @missed_appointments > 3
    BEGIN
        UPDATE Patients
        SET flagged = 1  -- Flag the patient
        WHERE patient_id = @patient_id;
    END
END;


INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (326, 108, 202, '2025-05-01 10:00:00', 'Missed');

select * from Patients;
```

NEEDED TO ADD A FLAG TO THE PATIENTS TABLE SO THAT I COULD SET THE FLAG WHEN A PATIENT MISSED MORE THAN 3 APPOINTMENTS.



```
    AND status = 'Missed';

    -- If the patient has more than 3 missed appointments, flag them
    IF @missed_appointments > 3
    BEGIN
        UPDATE Patients
        SET flagged = 1  -- Flag the patient
        WHERE patient_id = @patient_id;
    END
END;


INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (326, 108, 202, '2025-05-01 10:00:00', 'Missed');

select * from Patients;
```

	patient_id	name	gender	dob	contact	flagged
1	101	Alice Smith	F	1998-12-04 00:00:00.0000000	1234567890	NULL
2	102	Bob Johnson	M	1985-01-03 00:00:00.0000000	987654321	NULL
3	103	Charlie Davis	M	1998-08-04 00:00:00.0000000	1122334455	NULL
4	104	Denise Marlo	F	1990-07-16 00:00:00.0000000	1029384756	NULL
5	105	Emily Johnson	F	2010-08-14 00:00:00.0000000	9021874365	NULL
6	106	Fionna Markler	F	2017-05-17 00:00:00.0000000	2482940102	NULL
7	107	Gio Nelson	M	2000-05-22 00:00:00.0000000	3019384019	NULL
8	108	Harry Diner	M	1982-12-23 00:00:00.0000000	1930481743	1
9	109	Ian Tracer	M	2001-12-19 00:00:00.0000000	3018572947	NULL
10	110	Janet Logan	F	1993-02-27 00:00:00.0000000	6810395837	NULL
11	111	Kyle Macer	M	1995-03-28 00:00:00.0000000	8102948572	NULL
12	112	Lara Jean	F	2023-04-30 00:00:00.0000000	2949572047	NULL
13	113	Maria Decans	F	2009-05-23 00:00:00.0000000	2945071749	NULL
14	114	Natasha Naftel	F	1989-01-18 00:00:00.0000000	9203957284	NULL

Query executed successfully. ESHAAL\SQLEXPRESS (16.0 RTM) | ESHAAL\eshaa (56) | HospitalManagement | 00:00:00 | 22 rows

----------- 2 -----------

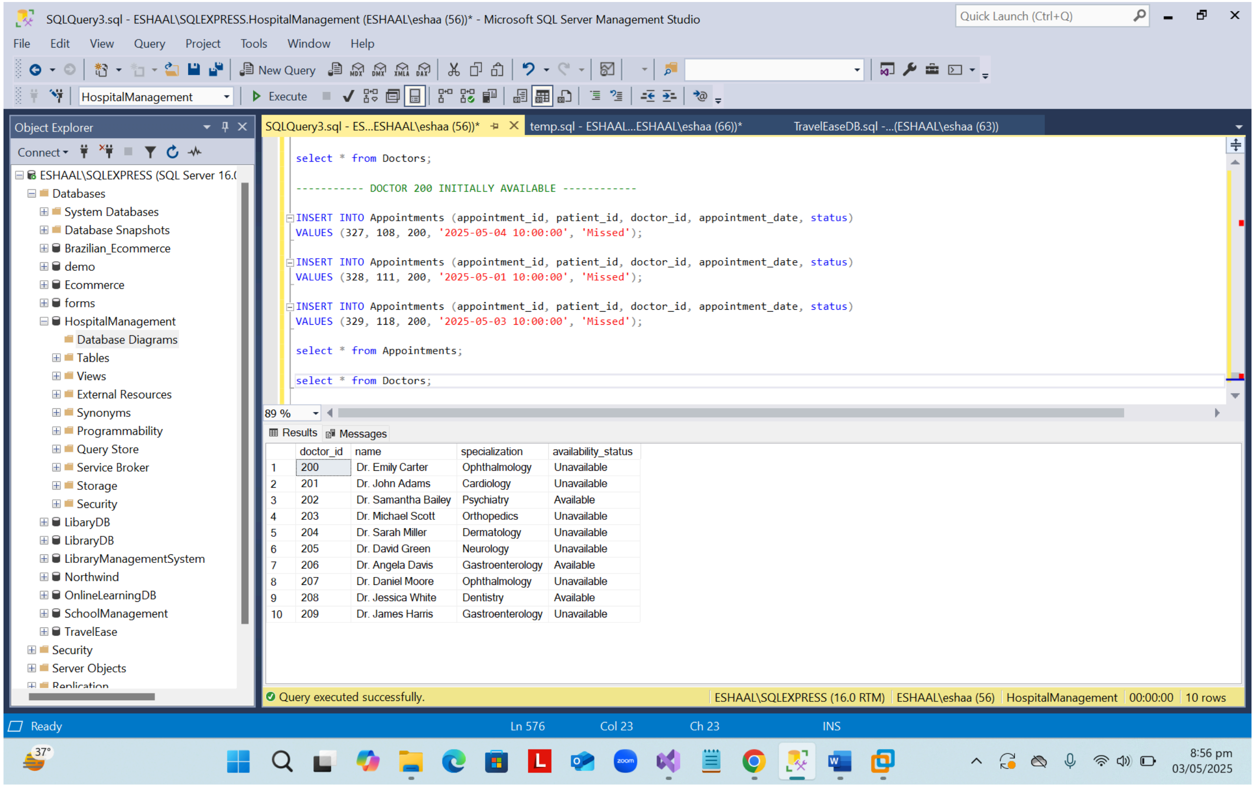
select * from Appointments;

CREATE TRIGGER trg_mark_doctor_unavailable
ON Appointments
AFTER INSERT, UPDATE
AS
BEGIN
    DECLARE @doctor_id INT;
    DECLARE @missed_appointments INT;

    -- Get the doctor_id from the inserted or updated row
    SELECT @doctor_id = doctor_id FROM inserted;

    -- Count the number of missed appointments for the doctor in the past 7 days
    SELECT @missed_appointments = COUNT(*)
    FROM Appointments
    WHERE doctor_id = @doctor_id
    AND status = 'Missed'

```
   AND appointment_date >= DATEADD(DAY, -7, GETDATE());  -- Appointments in the last 7 days

   -- If the doctor has 3 or more missed appointments in a week, mark them as 'Unavailable'
   IF @missed_appointments >= 3
   BEGIN
      UPDATE Doctors
      SET availability_status = 'Unavailable'
      WHERE doctor_id = @doctor_id;
   END
END;

select * from Doctors;

----------- DOCTOR 200 INITIALLY AVAILABLE ------------

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (326, 108, 200, '2025-05-04 10:00:00', 'Missed');

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (327, 111, 200, '2025-05-01 10:00:00', 'Missed');

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (328, 118, 200, '2025-05-03 10:00:00', 'Missed');

select * from Appointments;


----------- DOCTOR 200 NOW UNAVAILABLE ------------
```


SQLQuery3.sql - ESHAAL\SQLEXPRESS.HospitalManagement (ESHAAL\eshaa (56))* - Microsoft SQL Server Management Studio
File
Edit
View
Query
Project
Tools
Window
Help
New Query
Execute
HospitalManagement
Object Explorer
Connect
SQLQuery3.sql - ES...ESHAAL\eshaa (56))*
temp.sql - ESHAAL...ESHAAL\eshaa (66))*
TravelEaseDB.sql -...(ESHAAL\eshaa (63))
89 %
Results
Messages
Ready
Ln 576
Col 23
Ch 23
INS

```
select * from Doctors;

----------- DOCTOR 200 INITIALLY AVAILABLE ------------

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (327, 108, 200, '2025-05-04 10:00:00', 'Missed');

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (328, 111, 200, '2025-05-01 10:00:00', 'Missed');

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (329, 118, 200, '2025-05-03 10:00:00', 'Missed');

select * from Appointments;

select * from Doctors;
```

	doctor_id	name	specialization	availability_status
1	200	Dr. Emily Carter	Ophthalmology	Unavailable
2	201	Dr. John Adams	Cardiology	Unavailable
3	202	Dr. Samantha Bailey	Psychiatry	Available
4	203	Dr. Michael Scott	Orthopedics	Unavailable
5	204	Dr. Sarah Miller	Dermatology	Unavailable
6	205	Dr. David Green	Neurology	Unavailable
7	206	Dr. Angela Davis	Gastroenterology	Available
8	207	Dr. Daniel Moore	Ophthalmology	Unavailable
9	208	Dr. Jessica White	Dentistry	Available
10	209	Dr. James Harris	Gastroenterology	Unavailable

Query executed successfully.

ESHAAL\SQLEXPRESS (16.0 RTM) | ESHAAL\eshaa (56) | HospitalManagement | 00:00:00 | 10 rows

Object Explorer tree: ESHAAL\SQLEXPRESS (SQL Server 16.(; Databases; System Databases; Database Snapshots; Brazilian_Ecommerce; demo; Ecommerce; forms; HospitalManagement; Database Diagrams; Tables; Views; External Resources; Synonyms; Programmability; Query Store; Service Broker; Storage; Security; LibaryDB; LibraryDB; LibraryManagementSystem; Northwind; OnlineLearningDB; SchoolManagement; TravelEase; Security; Server Objects; Replication

37°

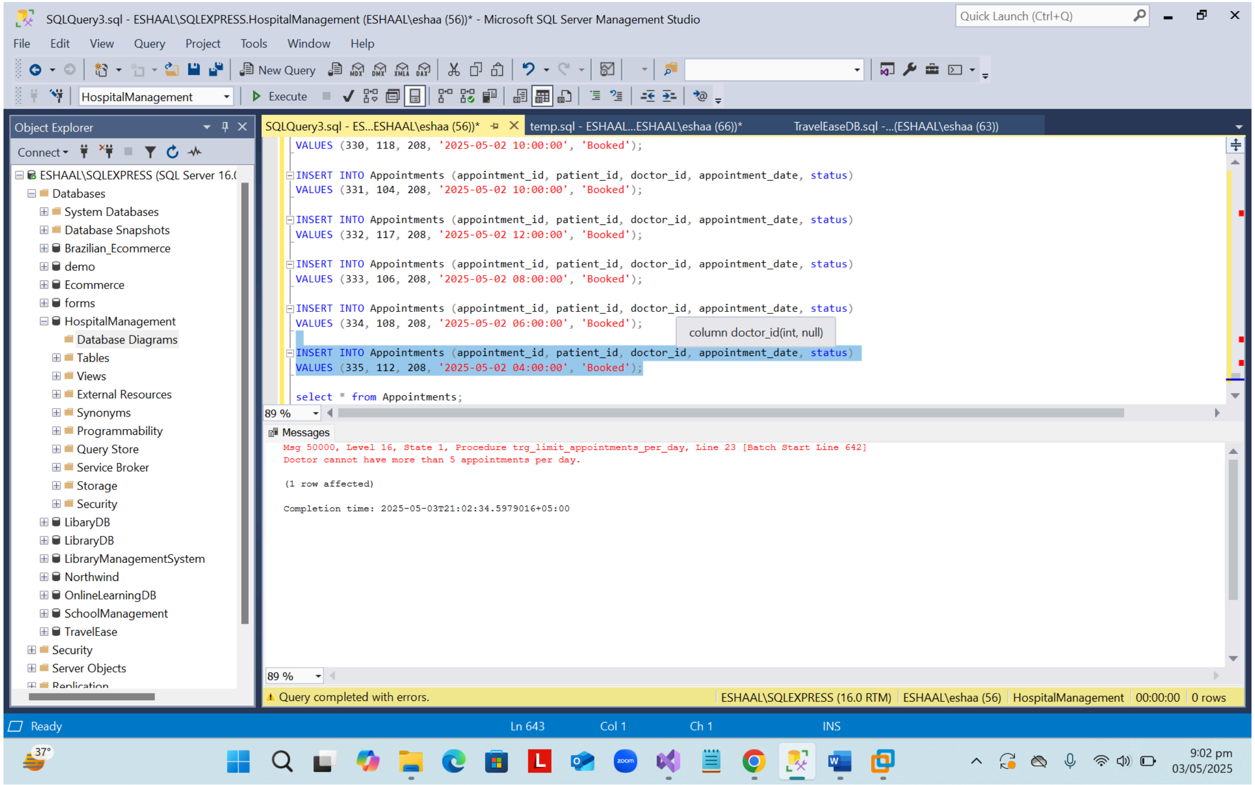
8:56 pm
03/05/2025

---------- 3 -----------

```
CREATE OR ALTER TRIGGER trg_limit_appointments_per_day
ON Appointments
INSTEAD OF INSERT
AS
BEGIN
    DECLARE @doctor_id INT;
    DECLARE @appointment_date DATETIME;
    DECLARE @existing_appointments INT;

    -- Get the doctor_id and appointment_date from the inserted row
    SELECT @doctor_id = doctor_id, @appointment_date = appointment_date FROM inserted;

    -- Count how many appointments the doctor already has on the same day
    SELECT @existing_appointments = COUNT(*)
    FROM Appointments
    WHERE doctor_id = @doctor_id
    AND CAST(appointment_date AS DATE) = CAST(@appointment_date AS DATE);  --
Compare only the date part
```

```
    -- If the doctor already has 5 appointments for that day, prevent the insert
    IF @existing_appointments >= 5
    BEGIN
        -- Raise an error and block the insert
        RAISERROR ('Doctor cannot have more than 5 appointments per day.', 16, 1);
    END
    ELSE
    BEGIN
        -- If the limit is not reached, insert the new appointment
        INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date,
status)
        SELECT appointment_id, patient_id, doctor_id, appointment_date, status
        FROM inserted;
    END
END;


select * from Doctors;

-- Insert 5 appointments for doctor 1 on the same day

------------- DOCTOR 208 WILL SEE AT LEAST 5 PATIENTS A DAY ---------------

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (330, 118, 208, '2025-05-02 10:00:00', 'Missed');

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (331, 104, 208, '2025-05-02 10:00:00', 'Completed');

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (332, 117, 208, '2025-05-02 12:00:00', 'Completed');

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (333, 106, 208, '2025-05-02 08:00:00', 'Completed');

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (334, 108, 208, '2025-05-02 06:00:00', 'Completed');

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (335, 112, 208, '2025-05-02 04:00:00', 'Booked');

select * from Appointments;

select * from Audit_Log;
```

DATABASE SYSTEMS ASSIGNMENT3, 23I-0696 FATIMA ISHTIAQ, 23I-0732 ESHAAL MALIK



```sql
VALUES (330, 118, 208, '2025-05-02 10:00:00', 'Booked');

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (331, 104, 208, '2025-05-02 10:00:00', 'Booked');

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (332, 117, 208, '2025-05-02 12:00:00', 'Booked');

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (333, 106, 208, '2025-05-02 08:00:00', 'Booked');

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (334, 108, 208, '2025-05-02 06:00:00', 'Booked');

INSERT INTO Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
VALUES (335, 112, 208, '2025-05-02 04:00:00', 'Booked');

select * from Appointments;
```

```
Msg 50000, Level 16, State 1, Procedure trg_limit_appointments_per_day, Line 23 [Batch Start Line 642]
Doctor cannot have more than 5 appointments per day.

(1 row affected)

Completion time: 2025-05-03T21:02:34.5979016+05:00
```