Introduction

This phase of my project focuses on extending the Student Housing and Maintenance Management System by incorporating automated database operations through triggers and stored procedures.
The main goal here is to ensure that frequently occurring operations, consistency rules, and administrative tasks are handled automatically at the database level which enhances database automation.

A trigger is a code that executes when a database event such as INSERT, UPDATE, or DELETE occurs. Triggers increase data integrity, enforce business rules, and eliminate human error by reacting instantly to changes in the database.

A stored procedure is a group of SQL codes stored together that can be reused over and over. Stored procedures help encapsulate repetitive operations into a single function. They help improve performance, by reducing code duplication, and make the system easier to maintain.

In this phase, one trigger and one stored procedure were implemented to automate maintenance request assignment and to generate dorm occupancy reports.

Trigger

Trigger Name: trg_request_auto_assign

Table: MaintenanceRequest

Type: BEFORE INSERT Trigger

Purpose and Functionality:

This trigger ensures that every maintenance request recorded in the system has an assigned staff member, even if the user submitting the request does not specify one. Students usually do not know which technician should handle an issue, and leaving a request unassigned would result in delays and inconsistent data.

The trigger executes before a row is inserted into the MaintenanceRequest table.
If the assignee_staff_id value of the new record is NULL, the trigger automatically assigns the default technician:

- **Staff ID:** 501
- **Name:** Ahmet Usta
- **Role:** Electrician

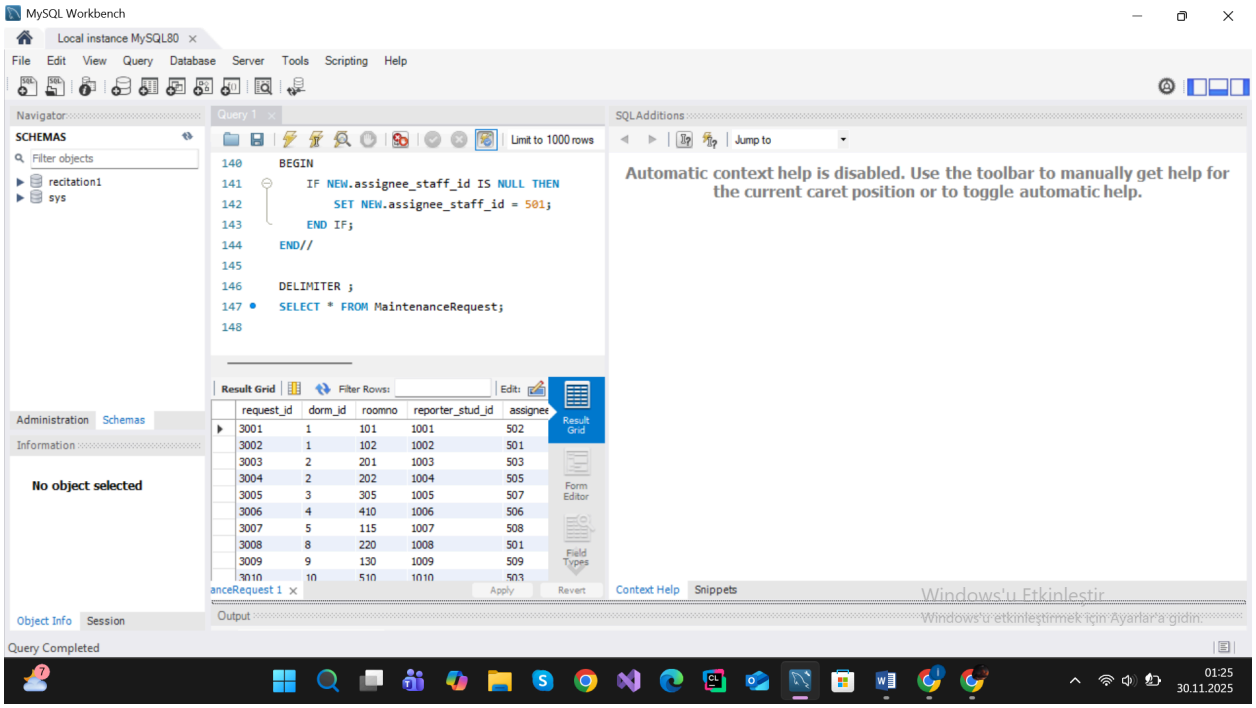
This approach guarantees seamless processing of maintenance requests and prevents incomplete or invalid records from being stored.

Trigger Script:

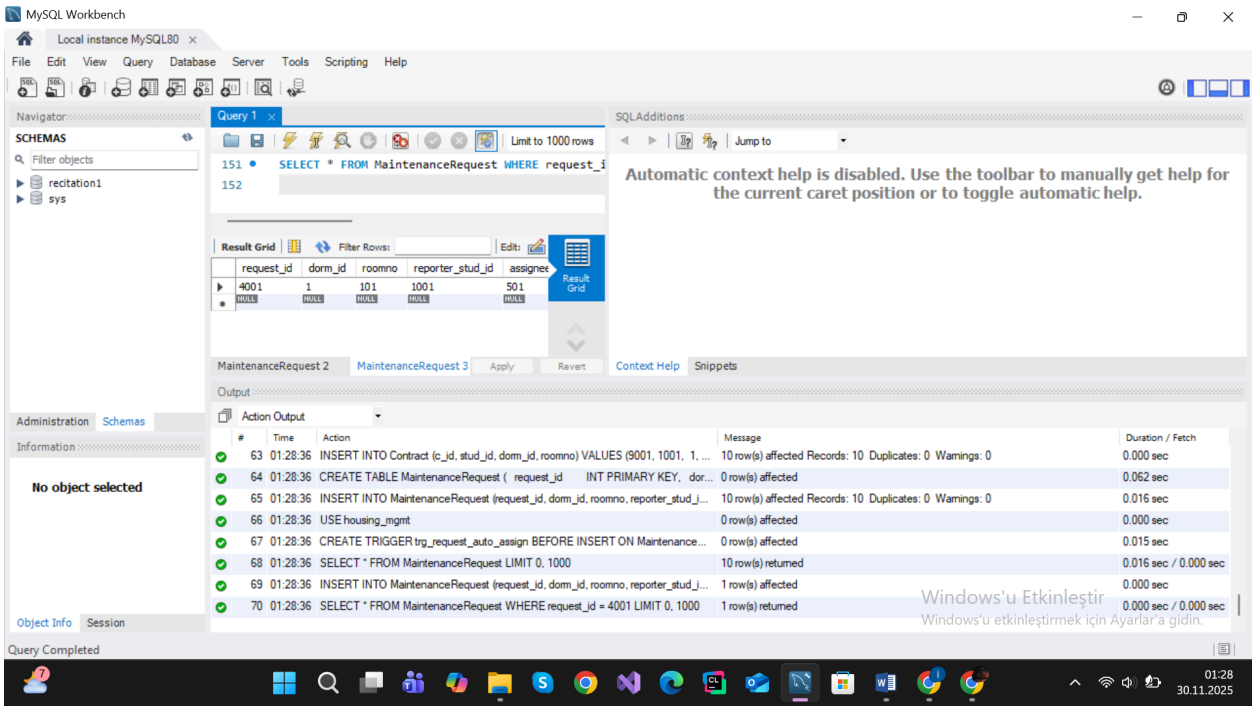
```
USE housing_mgmt;

DELIMITER //

CREATE TRIGGER trg_request_auto_assign

BEFORE INSERT ON MaintenanceRequest

FOR EACH ROW

BEGIN

   IF NEW.assignee_staff_id IS NULL THEN

      SET NEW.assignee_staff_id = 501;

   END IF;

END//

DELIMITER ;
```

Screenshots:

Before Trigger:



After Trigger:



Stored Procedure

Name: sp_room_occupancy_report

Purpose:

This stored procedure provides an administrative report showing room usage for a given dormitory. Dorm management frequently needs to know how many students are currently living in each room, how many beds are free, and whether rooms are fully occupied.

The procedure takes a dorm ID as input and returns:

- The room numbers in that dorm
- Each room's assigned capacity
- The number of students currently assigned through the Contract table
- The number of free beds remaining

Procedure Script:

```
USE housing_mgmt;

DELIMITER //

CREATE PROCEDURE sp_room_occupancy_report(IN p_dorm_id INT)
BEGIN
   SELECT
      r.dorm_id,
      r.roomno,
      r.capacity,
      COUNT(c.c_id) AS current_occupancy,
      (r.capacity - COUNT(c.c_id)) AS free_beds
   FROM Room r
   LEFT JOIN Contract c
     ON r.dorm_id = c.dorm_id
    AND r.roomno  = c.roomno
```

```
    WHERE r.dorm_id = p_dorm_id
    GROUP BY r.dorm_id, r.roomno, r.capacity
    ORDER BY r.roomno;
END//

DELIMITER ;
```