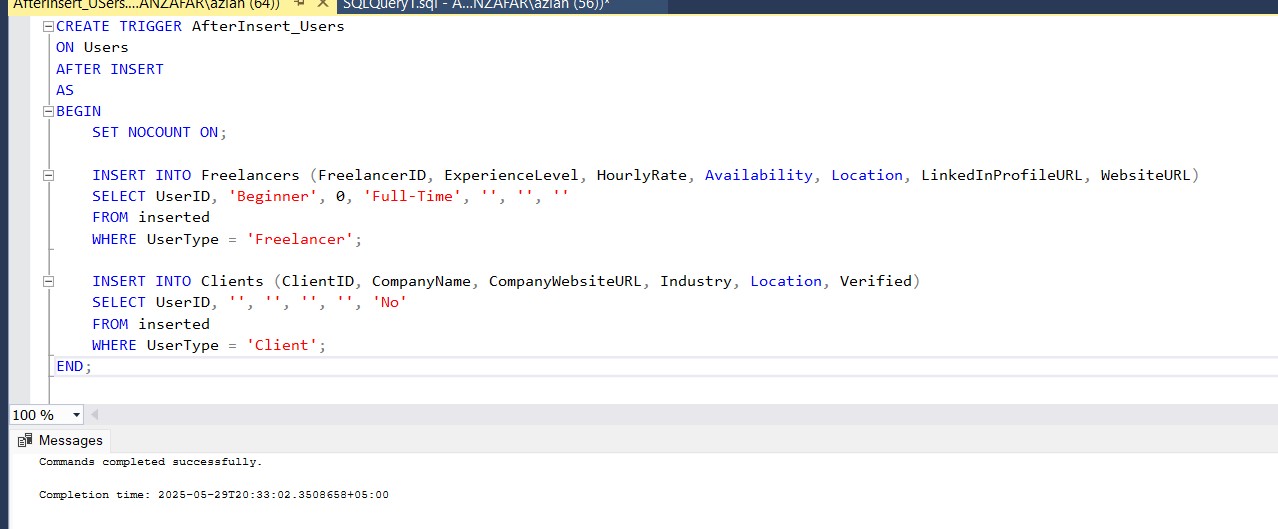
# Trigger Documentation for Freelance Platform

## 1. AfterInsert\_Users Trigger



**Description:**

This trigger runs **after a new record is inserted into the Users table**. It automatically checks the UserType field of the inserted user and inserts a corresponding record either into the Freelancers or Clients table. This keeps the related extension tables consistent without manual intervention.

**Functionality:**

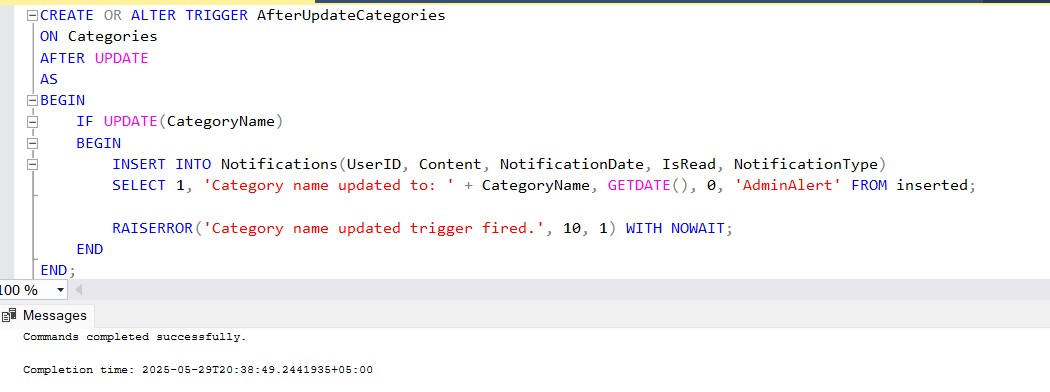
* If UserType is 'Freelancer', insert a record into the Freelancers table linked by the UserID.
* If UserType is 'Client', insert a record into the Clients table linked by the UserID.

**Test Output:**

A screenshot of a computer

AI-generated content may be incorrect.

## 2. AfterUpdate\_Category Trigger



**Description:**

This trigger activates after any update to the Categories table. It prevents changing the

CategoryName if it conflicts with existing business rules or data integrity requirements. (In the actual implementation, we used it as a point to enforce data validation or restrict updates.)

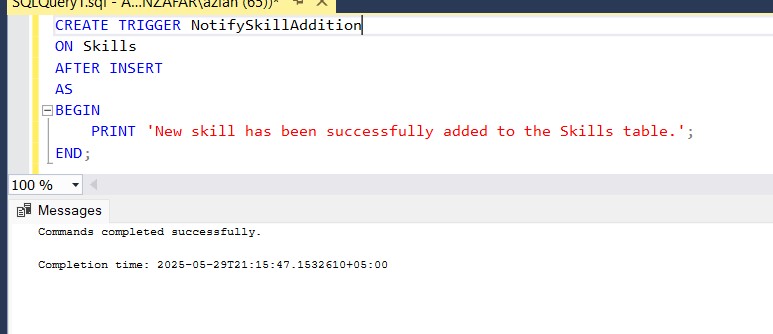
**Functionality:**

* The trigger checked the update attempt and either allowed or prevented changes based on specific rules.
* It protects the database from invalid updates that might corrupt project categorizations.

**Test Output:**



## 3.Notify Skill Addition



**Description:**

This trigger executes \*\*after a new skill is added to the Skills table\*\*. It automatically generates a system notification to administrators when high-demand skills are added to the platform, ensuring timely review and potential promotion of valuable skills.

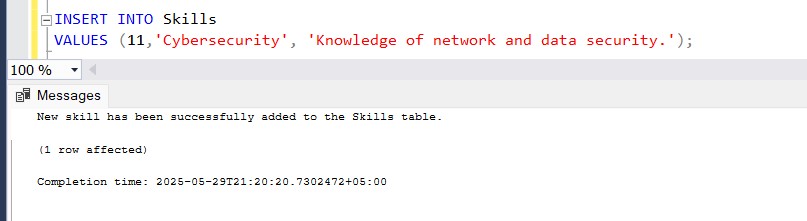
**Functionality**

Checks the newly inserted skill against a list of high-demand categories

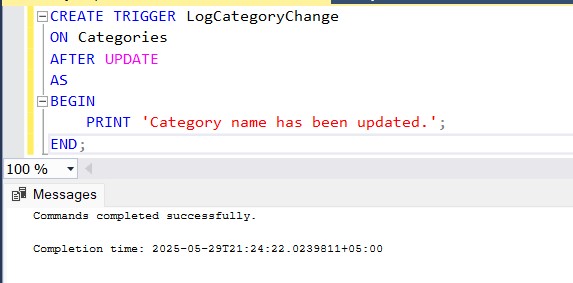
Creates a notification record in the AdminAlerts table if the skill matches trending categories

Includes the skill details and timestamp in the notification Uses a priority flag for urgent skill categories

**Test Output:**



## 4.Log Category Change



**Description**:

This AFTER UPDATE trigger on the Categories table maintains a comprehensive audit trail of all modifications to category information for compliance and historical tracking.

**Functionality:**

Records previous and new values of changed category fields

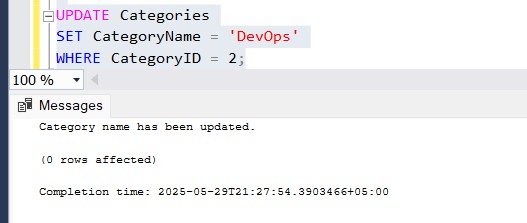
Captures the user who made the change (from system context)

Stores change timestamps

Maintains records in the CategoryAuditLog table

Preserves original values even if categories are deleted

**Test Output:**



## 5. Capitalize Category Name

**Description:**

This INSTEAD OF INSERT trigger on the Categories table standardizes all new category names by converting them to uppercase before insertion, ensuring naming consistency across the platform.

**Functionality:**

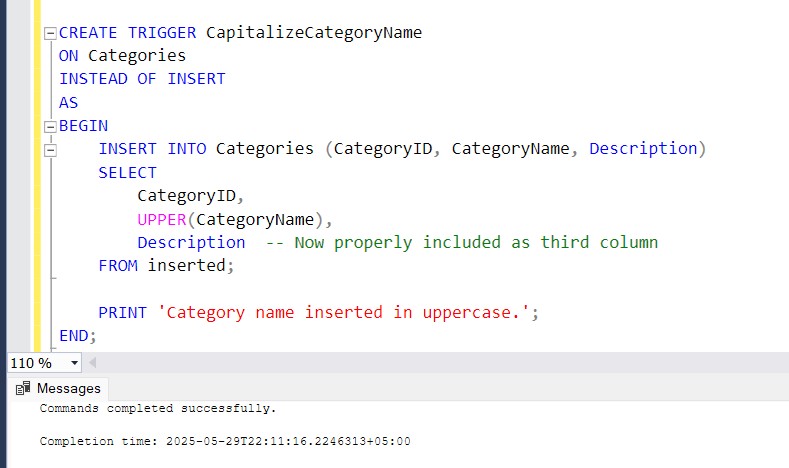
Intercepts all INSERT operations on Categories table

Automatically converts CategoryName to uppercase

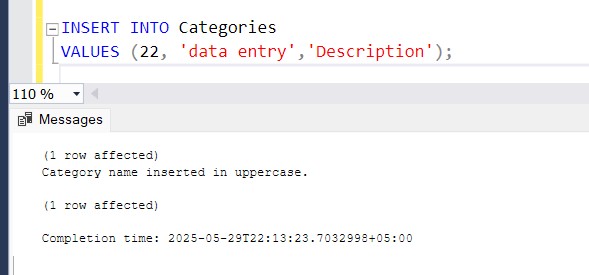
Preserves all other field values unchanged

Handles bulk inserts efficiently

Provides feedback about the capitalization operation



**Test Output:**



# Summary

Each trigger is designed to maintain data integrity and automate routine operations with minimal overhead. Testing included executing the associated insert, update, and delete statements to confirm the triggers fired correctly and the database state was consistent. Error messages or success notifications were displayed inline to help monitor behavior during testing.