Challenge:

In the tasks shown below, my assignment was to establish an automated procedure for recording modifications to a database table, and encrypt some of the data in the table, primarily as a security precaution. I was presented with the option of either utilizing external software tools or constructing a solution from scratch. The accompanying screenshot highlights the assignments I successfully fulfilled, encompassing the encryption of data within the table and the establishment of a trigger to monitor changes to the database table.

**Task: 1**

* 1. Showing two encrypted **columns of data.** Graphical user interface, text, application

     Description automatically generated
  2. **Other users with no permission cannot decrypt data in Student table.**

Graphical user interface, text, application, email

Description automatically generated

**Task: 2**

3-1: Showing the creation of TRIGGER named TRIG\_ClassDELETUPDATE

Graphical user interface, text, application, email

Description automatically generated

3-2: Validating**Graphical user interface, text, application

Description automatically generated**

4-1 Validating with an insert eventGraphical user interface, text, application

Description automatically generated

**Task: 3**

4-1: Creating a table called StudentAudit with DATE, AuditID added attributes added to it.Graphical user interface, text, application, email

Description automatically generated

4-2 Showing the TRIGGER for changes on the student table, with then the logs will be stored in the new table.

Graphical user interface, text, application, email

Description automatically generated

4-3: Validate the results by performing an update event.

Graphical user interface, text, application

Description automatically generated

Extra row of data is showing because I accidentally stored “Select \* From StudentAudit” which is showing the entire table.

**Thought:**

Automatic logging and the strategic implementation of triggers play pivotal roles in enhancing incident response capabilities within an organization. Firstly, automatic logging provides a continuous and comprehensive record of all activities occurring within a system or database. By meticulously tracking user interactions, data modifications, and system events, incident response teams gain real-time visibility into potential security breaches or irregularities. This detailed audit trail not only aids in rapid identification of suspicious activities but also enables forensic analysis to reconstruct the sequence of events leading up to an incident. Automatic logging, when intelligently configured, forms an invaluable resource for uncovering the root cause of security breaches, ensuring accountability, and facilitating effective decision-making during the incident response process.

In parallel, creating triggers within a database offers an active and responsive mechanism for incident detection and containment. Triggers, once set up to monitor specific conditions or events, can instantly initiate predefined actions upon occurrence. For incident response, triggers can act as an early warning system by immediately notifying security teams about unauthorized access attempts, data modifications, or anomalous patterns. These real-time alerts empower incident responders to swiftly intervene, isolate affected systems, and initiate the appropriate investigative procedures. Additionally, triggers can be harnessed to automatically trigger mitigation measures such as disabling compromised accounts or blocking malicious IP addresses. By proactively automating key aspects of incident detection and containment, triggers significantly reduce the time gap between breach occurrence and response initiation, bolstering an organization's resilience against cyber threats