dbo.trg_Product_Audit

Business

This trigger automatically keeps a historical record of any changes made to products. Whenever a product's details are modified or a product is deleted from the system, the trigger captures what the old values were, what the new values are, who made the change, and exactly when it happened. This is useful for auditing purposes — for example, tracking price changes over time or knowing who deleted a product and when.

Technical

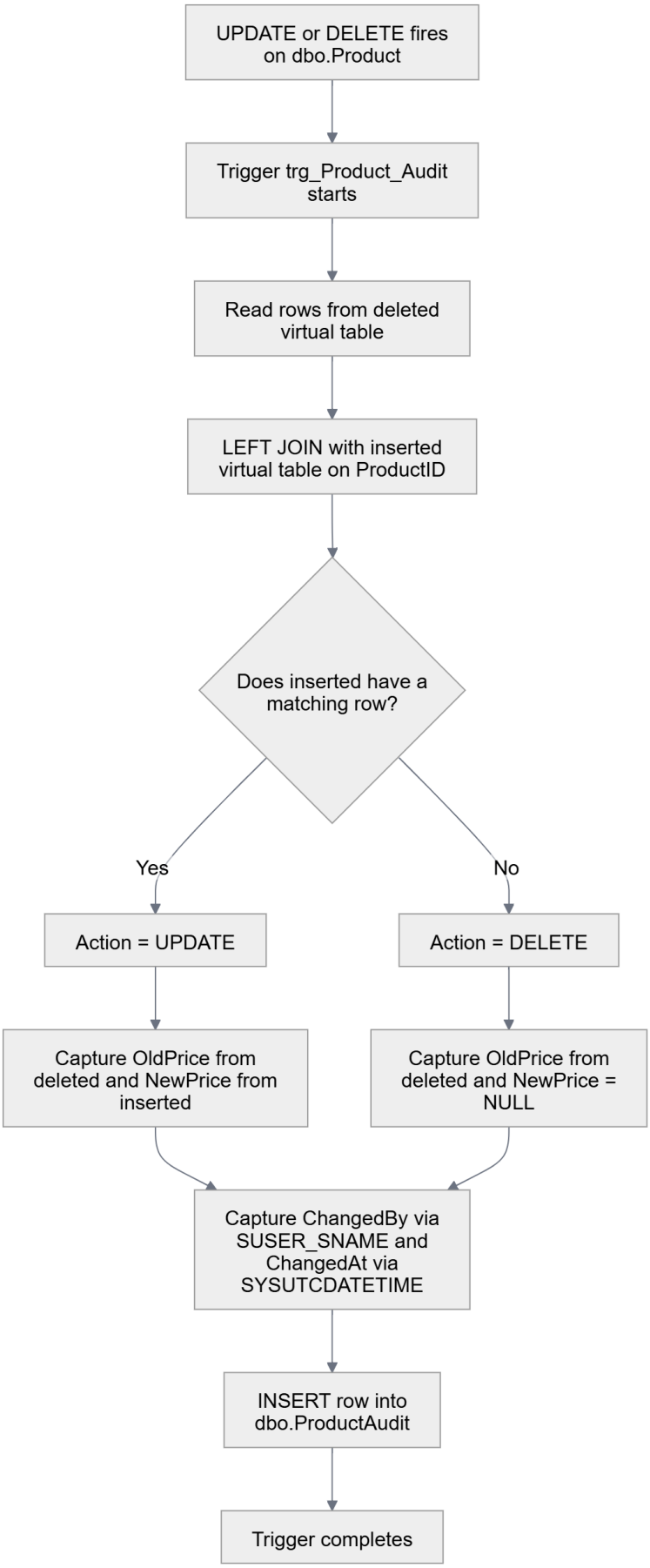
Input parameters:
- none (trigger fires automatically on UPDATE or DELETE on dbo.Product)

Used objects:
- dbo.Product (source table, trigger owner)
- dbo.ProductAudit (audit destination table)
- inserted (virtual trigger table, contains new row versions after UPDATE)
- deleted (virtual trigger table, contains old row versions before UPDATE or DELETE)

Logic:
- Step 1: Trigger fires after any UPDATE or DELETE operation on dbo.Product
- Step 2: SET NOCOUNT ON suppresses row-count messages
- Step 3: A single INSERT into dbo.ProductAudit is executed
- Step 4: Source data is built by joining deleted to inserted on ProductID using LEFT JOIN, so DELETE operations (where inserted has no matching row) are also captured
- Step 5: COALESCE picks ProductID from inserted if available, otherwise from deleted
- Step 6: Action is determined — if no matching row exists in inserted, the action is DELETE, otherwise UPDATE
- Step 7: OldPrice is taken from deleted, NewPrice from inserted (NULL for DELETE)
- Step 8: ChangedBy is captured via SUSER_SNAME() (current login name)
- Step 9: ChangedAt is captured via SYSUTCDATETIME() (UTC timestamp with high precision)

Side effects:
- INSERT into dbo.ProductAudit

Diagram


UPDATE or DELETE fires on dbo.Product
Trigger trg_Product_Audit starts
Read rows from deleted virtual table
LEFT JOIN with inserted virtual table on ProductID
Does inserted have a matching row?
Yes
No
Action = UPDATE
Action = DELETE
Capture OldPrice from deleted and NewPrice from inserted
Capture OldPrice from deleted and NewPrice = NULL
Capture ChangedBy via SUSER_SNAME and ChangedAt via SYSUTCDATETIME
INSERT row into dbo.ProductAudit
Trigger completes