```
USE MEI_TRAB


if exists (select * from sys.triggers  where Name = 'TR_Encomenda_I_U')
Begin
  DROP TRIGGER TR_Encomenda_I_U
End
GO

-- Trigger associado a Insert e Update. O Delete de encomendas é tratado à parte.

    CREATE TRIGGER TR_Encomenda_I_U
    ON Encomenda
    FOR INSERT, UPDATE
    AS
    DECLARE @NumReg int
    DECLARE @EventType Char(1)

    DECLARE @DELETEDCOUNT INT
    DECLARE @INSERTEDCOUNT INT

    DECLARE @EncId int
    DECLARE @Objecto     varchar(30)
    DECLARE @Valor       varchar(100)
    DECLARE @Referencia  varchar(100)

    SELECT @DELETEDCOUNT = COUNT(*) FROM deleted
    SELECT @INSERTEDCOUNT = COUNT(*) FROM inserted

    -- Event type
    IF (@DELETEDCOUNT > 0) AND (@INSERTEDCOUNT > 0 )
      Set @EventType = 'U'
    ELSE IF (@INSERTEDCOUNT > 0 )
      Set @EventType = 'I'

    Set @Objecto = 'Encomenda'  -- Tabela encomenda
    Set @Referencia = NULL      -- Para estas operações não é usada a referência

    DECLARE fh_cursor CURSOR FOR
    SELECT EncId FROM inserted ORDER by EncId

    OPEN fh_cursor

    FETCH NEXT FROM fh_cursor INTO @EncId

    WHILE (@@fetch_status = 0)
    BEGIN

      Set @Valor = CAST(@EncId as nvarchar(100))  -- Convert int to varchar

      INSERT INTO LogOperations (EventType, Objecto, Valor, Referencia)
      Values (@EventType, @Objecto, @Valor, @Referencia)


      FETCH NEXT FROM fh_cursor INTO @EncId
    END

    IF  (@@fetch_status = -1)
      Begin
        CLOSE fh_cursor
        DEALLOCATE fh_cursor
        RETURN
      End

    on_error:
     CLOSE fh_cursor
     DEALLOCATE fh_cursor


      RAISERROR('Can''t insert the record',16,1 )
      ROLLBACK TRANSACTION
```

```sql
GO



if exists (select * from sys.triggers  where Name = 'TR_Encomenda_D')
Begin
  DROP TRIGGER TR_Encomenda_D
End
GO


    CREATE TRIGGER TR_Encomenda_D
    ON Encomenda
    FOR DELETE
    AS
    DECLARE @NumReg int
    DECLARE @EventType Char(1)

    DECLARE @EncId int
    DECLARE @ClienteID int
    DECLARE @Nome nvarchar(30)
    DECLARE @Morada nvarchar(30)

    DECLARE @Objecto     varchar(30)
    DECLARE @Valor       varchar(100)
    DECLARE @Referencia  varchar(100)

    Set @EventType = 'D'

    Set @Objecto = 'Encomenda'  -- Tabela encomenda
    Set @Referencia = NULL      -- Para estas operações não é usada a referência

--
-- with a cursor we can iterate the deleted rows
--


    DECLARE fh_cursor CURSOR FOR
    SELECT EncId, ClienteId, Nome, Morada FROM Deleted ORDER by EncId

    OPEN fh_cursor

    FETCH NEXT FROM fh_cursor INTO @EncId, @ClienteId, @Nome, @Morada

    WHILE (@@fetch_status = 0)
    BEGIN
      Set @Valor = CAST(@EncId as nvarchar(10)) + '|'+ @Nome + '|' + @Morada

      INSERT INTO LogOperations (EventType, Objecto, Valor, Referencia)
      Values (@EventType, @Objecto, @Valor, @Referencia)

      FETCH NEXT FROM fh_cursor INTO @EncId, @ClienteId, @Nome, @Morada
    END

    IF  (@@fetch_status = -1)
      Begin
        CLOSE fh_cursor
        DEALLOCATE fh_cursor
        RETURN
      End

    on_error:
     CLOSE fh_cursor
     DEALLOCATE fh_cursor


      RAISERROR('Can''t insert the record',16,1 )
      ROLLBACK TRANSACTION
GO

if exists (select * from sys.triggers  where Name = 'TR_EncLinha_I_U')
```

```
Begin
  DROP TRIGGER TR_EncLinha_I_U
End
GO


    CREATE TRIGGER TR_EncLinha_I_U
    ON EncLinha
    FOR INSERT, UPDATE
    AS
    DECLARE @EventType Char(1)

    DECLARE @EncId int
    DECLARE @ProdutoID int

    DECLARE @Objecto     varchar(30)
    DECLARE @Valor       varchar(100)
    DECLARE @Referencia  varchar(100)

    DECLARE @DELETEDCOUNT INT
    DECLARE @INSERTEDCOUNT INT

    SELECT @DELETEDCOUNT = COUNT(*) FROM deleted
    SELECT @INSERTEDCOUNT = COUNT(*) FROM inserted

    -- Event type
    IF (@DELETEDCOUNT > 0) AND (@INSERTEDCOUNT > 0 )
      Set @EventType = 'U'
    ELSE IF (@INSERTEDCOUNT > 0 )
      Set @EventType = 'I'

    Set @Objecto = 'EncLinha'  -- Tabela encomenda


    DECLARE fh_cursor CURSOR FOR
    SELECT EncId, ProdutoId FROM inserted ORDER by EncId

    OPEN fh_cursor

    FETCH NEXT FROM fh_cursor INTO @EncId, @ProdutoId

    WHILE (@@fetch_status = 0)
    BEGIN


      Set @Valor = CAST(@EncId as varchar(100))             -- Convert int to varchar
      Set @Referencia = CAST(@ProdutoId as varchar(100))    -- Convert int to varchar

      INSERT INTO LogOperations (EventType, Objecto, Valor, Referencia)
      Values (@EventType, @Objecto, @Valor, @Referencia)


      FETCH NEXT FROM fh_cursor INTO @EncId, @ProdutoId
    END

    IF  (@@fetch_status = -1)
      Begin
        CLOSE fh_cursor
        DEALLOCATE fh_cursor
        RETURN
      End

    on_error:
     CLOSE fh_cursor
     DEALLOCATE fh_cursor


      RAISERROR('Can''t insert the record',16,1 )
      ROLLBACK TRANSACTION
GO

```

```sql
if exists (select * from sys.triggers  where Name = 'TR_EncLinha_D')
Begin
  DROP TRIGGER TR_EncLinha_D
End
GO


    CREATE TRIGGER TR_EncLinha_D
    ON EncLinha
    FOR Delete
    AS
    DECLARE @EventType Char(1)

    DECLARE @DELETEDCOUNT INT
    DECLARE @INSERTEDCOUNT INT

    DECLARE @EncId int
    DECLARE @ProdutoID int
    DECLARE @Designacao nvarchar(50)
    DECLARE @Preco Decimal(10,2)
    DECLARE @Qtd Decimal(10,2)

    DECLARE @Objecto     varchar(30)
    DECLARE @Valor       varchar(100)
    DECLARE @Referencia  varchar(100)

    Set @EventType = 'D'

    Set @Objecto = 'EncLinha'  -- Tabela encomenda
    Set @Referencia = NULL      -- Para estas operações não é usada a referência


    DECLARE fh_cursor CURSOR FOR
    SELECT EncId, ProdutoId, Designacao, Preco, Qtd FROM Deleted ORDER by EncId

    OPEN fh_cursor

    FETCH NEXT FROM fh_cursor INTO @EncId, @ProdutoId, @Designacao, @Preco, @Qtd

    WHILE (@@fetch_status = 0)
    BEGIN
      Set @Valor = CAST(@EncId as nvarchar(10)) + '|'+ CAST(@ProdutoId as nvarchar(10))
      Set @Referencia = @Designacao + '|'+ CAST(@Preco as nvarchar(10)) + '|'+ CAST(@Qtd as nvarchar(10))

      INSERT INTO LogOperations (EventType, Objecto, Valor, Referencia)
      Values  (@EventType, @Objecto, @Valor, @Referencia)


      FETCH NEXT FROM fh_cursor INTO @EncId, @ProdutoId, @Designacao, @Preco, @Qtd
    END

    IF  (@@fetch_status = -1)
      Begin
        CLOSE fh_cursor
        DEALLOCATE fh_cursor
        RETURN
      End

    on_error:
     CLOSE fh_cursor
     DEALLOCATE fh_cursor


      RAISERROR('Can''t insert the record',16,1 )
      ROLLBACK TRANSACTION
GO
```