**ПРИЛОЖЕНИЕ Д**

|  |
| --- |
| USE Master\_CarManager  CREATE TRIGGER InsteadOfDeleteUser ON Users INSTEAD OF DELETE AS RETURN  GO  CREATE TRIGGER InsteadOfDeleteCar ON Cars INSTEAD OF DELETE AS  BEGIN  UPDATE CARS SET ACTIVE = 0 WHERE CarId in (SELECT CarId FROM deleted)  END;  GO  CREATE TRIGGER InsteadOfUpdateCarsUsers ON UsersCars INSTEAD OF UPDATE AS RETURN  GO  CREATE TRIGGER InsteadOfInsertDetails ON Details INSTEAD OF INSERT AS  BEGIN  DECLARE @CAR\_ID uniqueidentifier;  DECLARE @DETAIL\_ID uniqueidentifier;  DECLARE @SERVICE\_ID uniqueidentifier;  DECLARE @NAME VARCHAR(100);  DECLARE @TYPE VARCHAR(100);  SELECT @DETAIL\_ID = DetailId, @SERVICE\_ID = idService, @NAME = Name, @TYPE = Type FROM inserted  SET @CAR\_ID = (SELECT TOP(1) idCar FROM CarEvents WHERE EventId IN (SELECT idEvent FROM CarServices WHERE ServiceId IN (SELECT ServiceId FROM inserted)))  INSERT INTO [dbo].[Details]([DetailId],[idCar],[idService],[Name],[Type])  VALUES (@DETAIL\_ID,@CAR\_ID,@SERVICE\_ID,@NAME,@TYPE)  END;  GO  CREATE TRIGGER AuditTrigger\_Users  ON USERS AFTER UPDATE AS  BEGIN  DECLARE @COUNT\_INSERTED INT;  DECLARE @COUNT\_DELETED INT;  DECLARE @ENTITY\_NAME VARCHAR(20);  DECLARE @DATETIME VARCHAR(50);  DECLARE @ACTION VARCHAR(20);  DECLARE @ENTITY\_ID uniqueidentifier;  DECLARE @USER\_ID uniqueidentifier;  SET @ENTITY\_NAME = 'Users';  SET @DATETIME = (SELECT format(getdate(), 'yyyy-MM-dd HH:mm:ss.fff'));  SET @COUNT\_INSERTED = (SELECT COUNT(\*) FROM inserted);  SET @COUNT\_DELETED = (SELECT COUNT(\*) FROM deleted);  SET @ENTITY\_ID = (SELECT U.[UserId] FROM inserted AS U)  SET @USER\_ID = (SELECT U.[UserId] FROM inserted AS U)  SET @ACTION = 'PUT';  INSERT INTO [dbo].[ActionAudit] ([Entity], [EntityId] ,[IdUser] ,[Action] ,[DateUpdate])  VALUES (@ENTITY\_NAME, @ENTITY\_ID, @USER\_ID, @ACTION, @DATETIME);  END  CREATE TRIGGER AuditTrigger\_Cars  ON CARS AFTER UPDATE AS  BEGIN  DECLARE @ENTITY\_NAME VARCHAR(20);  DECLARE @DATETIME VARCHAR(50);  DECLARE @ACTION VARCHAR(20);  DECLARE @ENTITY\_ID uniqueidentifier;  DECLARE @USER\_ID uniqueidentifier;  SET @ENTITY\_NAME = 'Cars';  SET @DATETIME = (SELECT format(getdate(), 'yyyy-MM-dd HH:mm:ss.fff'));  SET @ACTION = 'PUT';  DECLARE USER\_Ids CURSOR LOCAL FORWARD\_ONLY FAST\_FORWARD FOR  SELECT idUser FROM UsersCars WHERE idCar in (SELECT CarId FROM inserted)  SET @ENTITY\_ID = (SELECT CarId FROM inserted)  OPEN USER\_IdS;  FETCH NEXT FROM USER\_IDS INTO @USER\_ID;  WHILE @@FETCH\_STATUS = 0  BEGIN  INSERT INTO [dbo].[ActionAudit] ([Entity], [EntityId] ,[IdUser] ,[Action] ,[DateUpdate])  VALUES (@ENTITY\_NAME, @ENTITY\_ID, @USER\_ID, @ACTION, @DATETIME);  FETCH NEXT FROM USER\_IDS INTO @USER\_ID  END  CLOSE USER\_IdS;  END  GO  CREATE TRIGGER AuditTrigger\_UserCars  ON UsersCars AFTER INSERT, Delete AS  BEGIN  DECLARE @COUNT\_INSERTED INT;  DECLARE @COUNT\_DELETED INT;  DECLARE @ENTITY\_NAME VARCHAR(20);  DECLARE @DATETIME VARCHAR(50);  DECLARE @ACTION VARCHAR(20);  DECLARE @ENTITY\_ID uniqueidentifier;  DECLARE @USER\_ID uniqueidentifier;    SET @COUNT\_INSERTED = (SELECT COUNT(\*) FROM inserted);  SET @COUNT\_DELETED = (SELECT COUNT(\*) FROM deleted);  SET @ENTITY\_NAME = 'Cars';  SET @DATETIME = (SELECT format(getdate(), 'yyyy-MM-dd HH:mm:ss.fff'));  IF(@COUNT\_DELETED = 1 AND @COUNT\_INSERTED = 0)  BEGIN  SET @ENTITY\_ID = (SELECT idCar FROM deleted)  SET @USER\_ID = (SELECT idUser FROM deleted)  SET @ACTION = 'DELETE';  END  IF(@COUNT\_DELETED = 0 AND @COUNT\_INSERTED = 1)  BEGIN  SET @ENTITY\_ID = (SELECT idCar FROM inserted)  SET @USER\_ID = (SELECT idUser FROM inserted)  SET @ACTION = 'POST';  END  INSERT INTO [dbo].[ActionAudit] ([Entity], [EntityId] ,[IdUser] ,[Action] ,[DateUpdate])  VALUES (@ENTITY\_NAME, @ENTITY\_ID, @USER\_ID, @ACTION, @DATETIME);  END  GO  CREATE TRIGGER AuditTrigger\_CarEvents  ON CarEvents AFTER INSERT, Update, Delete AS  BEGIN  DECLARE @COUNT\_INSERTED INT;  DECLARE @COUNT\_DELETED INT;  DECLARE @ENTITY\_NAME VARCHAR(20);  DECLARE @DATETIME VARCHAR(50);  DECLARE @ACTION VARCHAR(20);  DECLARE @ENTITY\_ID uniqueidentifier;  DECLARE @USER\_ID uniqueidentifier;  SET @COUNT\_INSERTED = (SELECT COUNT(\*) FROM inserted);  SET @COUNT\_DELETED = (SELECT COUNT(\*) FROM deleted);  SET @ENTITY\_NAME = 'EVENTS';  SET @DATETIME = (SELECT format(getdate(), 'yyyy-MM-dd HH:mm:ss.fff'));  IF(@COUNT\_DELETED = 1 AND @COUNT\_INSERTED = 0)  BEGIN  SET @ENTITY\_ID = (SELECT EventId FROM deleted)  SET @ACTION = 'DELETE';  DECLARE USER\_Ids CURSOR LOCAL FORWARD\_ONLY FAST\_FORWARD FOR  SELECT idUser FROM UsersCars where idCar in (SELECT idCar FROM deleted)  OPEN USER\_IdS;  FETCH NEXT FROM USER\_IDS INTO @USER\_ID;  WHILE @@FETCH\_STATUS = 0  BEGIN  INSERT INTO [dbo].[ActionAudit] ([Entity], [EntityId] ,[IdUser] ,[Action] ,[DateUpdate])  VALUES (@ENTITY\_NAME, @ENTITY\_ID, @USER\_ID, @ACTION, @DATETIME);  FETCH NEXT FROM USER\_IDS INTO @USER\_ID  END  CLOSE USER\_IdS;  END  ELSE  BEGIN  SET @ENTITY\_ID = (SELECT EventId FROM inserted)  IF(@COUNT\_DELETED = 0 AND @COUNT\_INSERTED = 1)  BEGIN  SET @ACTION = 'POST';  END  IF(@COUNT\_DELETED = 1 AND @COUNT\_INSERTED = 1)  BEGIN  SET @ACTION = 'PUT';  END  DECLARE USER\_Ids CURSOR LOCAL FORWARD\_ONLY FAST\_FORWARD FOR  SELECT idUser FROM UsersCars where idCar in (SELECT idCar FROM inserted)  OPEN USER\_IdS;  FETCH NEXT FROM USER\_IDS INTO @USER\_ID;  WHILE @@FETCH\_STATUS = 0  BEGIN  INSERT INTO [dbo].[ActionAudit] ([Entity], [EntityId] ,[IdUser] ,[Action] ,[DateUpdate])  VALUES (@ENTITY\_NAME, @ENTITY\_ID, @USER\_ID, @ACTION, @DATETIME);  FETCH NEXT FROM USER\_IDS INTO @USER\_ID  END  CLOSE USER\_IdS;  END  END  CREATE TRIGGER AuditTrigger\_CarServices  ON CarServices AFTER Update AS  BEGIN  DECLARE @COUNT\_INSERTED INT;  DECLARE @COUNT\_DELETED INT;  DECLARE @ENTITY\_NAME VARCHAR(20);  DECLARE @DATETIME VARCHAR(50);  DECLARE @ACTION VARCHAR(20);  DECLARE @ENTITY\_ID uniqueidentifier;  DECLARE @USER\_ID uniqueidentifier;  SET @COUNT\_INSERTED = (SELECT COUNT(\*) FROM inserted);  SET @COUNT\_DELETED = (SELECT COUNT(\*) FROM deleted);  SET @ENTITY\_NAME = 'CarServices';  SET @DATETIME = (SELECT format(getdate(), 'yyyy-MM-dd HH:mm:ss.fff'));  SET @ENTITY\_ID = (SELECT idEvent FROM inserted)  SET @ACTION = 'PUT';  DECLARE USER\_Ids CURSOR LOCAL FORWARD\_ONLY FAST\_FORWARD FOR  SELECT idUser FROM UsersCars where idCar in (SELECT idCar FROM CarEvents WHERE EventId IN (SELECT idEvent FROM inserted))  OPEN USER\_IdS;  FETCH NEXT FROM USER\_IDS INTO @USER\_ID;  WHILE @@FETCH\_STATUS = 0  BEGIN  INSERT INTO [dbo].[ActionAudit] ([Entity], [EntityId] ,[IdUser] ,[Action] ,[DateUpdate])  VALUES (@ENTITY\_NAME, @ENTITY\_ID, @USER\_ID, @ACTION, @DATETIME);  FETCH NEXT FROM USER\_IDS INTO @USER\_ID  END  CLOSE USER\_IdS;  END  GO  CREATE TRIGGER AuditTrigger\_Fuels  ON Fuels AFTER Update AS  BEGIN  DECLARE @COUNT\_INSERTED INT;  DECLARE @COUNT\_DELETED INT;  DECLARE @ENTITY\_NAME VARCHAR(20);  DECLARE @DATETIME VARCHAR(50);  DECLARE @ACTION VARCHAR(20);  DECLARE @ENTITY\_ID uniqueidentifier;  DECLARE @USER\_ID uniqueidentifier;  SET @COUNT\_INSERTED = (SELECT COUNT(\*) FROM inserted);  SET @COUNT\_DELETED = (SELECT COUNT(\*) FROM deleted);  SET @ENTITY\_NAME = 'Fuels';  SET @DATETIME = (SELECT format(getdate(), 'yyyy-MM-dd HH:mm:ss.fff'));  SET @ENTITY\_ID = (SELECT idEvent FROM inserted)  SET @ACTION = 'PUT';  DECLARE USER\_Ids CURSOR LOCAL FORWARD\_ONLY FAST\_FORWARD FOR  SELECT idUser FROM UsersCars where idCar in (SELECT idCar FROM CarEvents WHERE EventId IN (SELECT idEvent FROM inserted))  OPEN USER\_IdS;  FETCH NEXT FROM USER\_IDS INTO @USER\_ID;  WHILE @@FETCH\_STATUS = 0  BEGIN  INSERT INTO [dbo].[ActionAudit] ([Entity], [EntityId] ,[IdUser] ,[Action] ,[DateUpdate])  VALUES (@ENTITY\_NAME, @ENTITY\_ID, @USER\_ID, @ACTION, @DATETIME);  FETCH NEXT FROM USER\_IDS INTO @USER\_ID  END  CLOSE USER\_IdS;  END  GO  CREATE TRIGGER AuditTrigger\_Details  ON Details AFTER Update AS  BEGIN  DECLARE @COUNT\_INSERTED INT;  DECLARE @COUNT\_DELETED INT;  DECLARE @ENTITY\_NAME VARCHAR(20);  DECLARE @DATETIME VARCHAR(50);  DECLARE @ACTION VARCHAR(20);  DECLARE @ENTITY\_ID uniqueidentifier;  DECLARE @USER\_ID uniqueidentifier;  SET @COUNT\_INSERTED = (SELECT COUNT(\*) FROM inserted);  SET @COUNT\_DELETED = (SELECT COUNT(\*) FROM deleted);  SET @ENTITY\_NAME = 'Details';  SET @DATETIME = (SELECT format(getdate(), 'yyyy-MM-dd HH:mm:ss.fff'));  SET @ENTITY\_ID = (SELECT idEvent FROM CarServices WHERE ServiceId = (SELECT ServiceId FROM inserted))  SET @ACTION = 'PUT';  DECLARE USER\_Ids CURSOR LOCAL FORWARD\_ONLY FAST\_FORWARD FOR  SELECT idUser FROM UsersCars where idCar in (SELECT idCar FROM inserted)  OPEN USER\_IdS;  FETCH NEXT FROM USER\_IDS INTO @USER\_ID;  WHILE @@FETCH\_STATUS = 0  BEGIN  INSERT INTO [dbo].[ActionAudit] ([Entity], [EntityId] ,[IdUser] ,[Action] ,[DateUpdate])  VALUES (@ENTITY\_NAME, @ENTITY\_ID, @USER\_ID, @ACTION, @DATETIME);  FETCH NEXT FROM USER\_IDS INTO @USER\_ID  END  CLOSE USER\_IdS;  END |