

1a

The screenshot shows a SQL Server Enterprise Manager window with two tabs: 'V4_2.sql - THE-MA...IN-MAN\Erop (71))' and 'V4_1.sql - THE-MA...IN-MAN\Erop (57))*'. The active tab contains a T-SQL script. The script starts with a comment in Russian: '/* a) Создайте таблицу Production.ProductModelHst, которая будет хранить информацию об изменениях в таблице Production.ProductModel. Обязательные поля, которые должны присутствовать в таблице: ID – первичный ключ IDENTITY(1,1); Action – совершенное действие (insert, update или delete); ModifiedDate – дата и время, когда была совершена операция; SourceID – первичный ключ исходной таблицы; UserName – имя пользователя, совершившего операцию. Создайте другие поля, если считаете их нужными. */'. This is followed by a 'CREATE TABLE' statement for 'Production.ProductModelHst' with columns: 'ID INT IDENTITY(1, 1)', 'Action NVARCHAR(6) NOT NULL CHECK (Action IN('insert', 'update', 'delete'))', 'ModifiedDate DATETIME NOT NULL DEFAULT GETDATE()', 'SourceID NCHAR(3) NOT NULL', and 'UserName Name NOT NULL'. The script ends with 'GO'. Below the script editor, a status bar shows '110 %' zoom. A 'Сообщения' (Messages) pane at the bottom displays the message 'Выполнение команд успешно завершено.' (Command execution successfully completed.) and the execution time 'Время выполнения: 2020-11-05T19:19:33.4479301+03:00'.

```

/* a) Создайте таблицу Production.ProductModelHst,
   которая будет хранить информацию об изменениях в таблице Production.ProductModel.
   Обязательные поля, которые должны присутствовать в таблице:
   ID – первичный ключ IDENTITY(1,1);
   Action – совершенное действие (insert, update или delete);
   ModifiedDate – дата и время, когда была совершена операция;
   SourceID – первичный ключ исходной таблицы;
   UserName – имя пользователя, совершившего операцию.
   Создайте другие поля, если считаете их нужными.
   */

CREATE TABLE Production.ProductModelHst
(
    ID INT IDENTITY(1, 1),
    Action NVARCHAR(6) NOT NULL CHECK (Action IN('insert', 'update', 'delete')),
    ModifiedDate DATETIME NOT NULL DEFAULT GETDATE(),
    SourceID NCHAR(3) NOT NULL,
    UserName Name NOT NULL
);
GO

```

Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2020-11-05T19:19:33.4479301+03:00

1b

The screenshot shows the same SQL Server Enterprise Manager window. The active tab now contains a T-SQL script for creating a trigger. The comment in Russian reads: '/* b) Создайте один AFTER триггер для трех операций INSERT, UPDATE, DELETE для таблицы Production.ProductModel. Триггер должен заполнять таблицу Production.ProductModelHst с указанием типа операции в поле Action в зависимости от оператора, вызвавшего триггер. */'. The script is a 'CREATE TRIGGER' statement for 'Production.OnProductModelInsert' on the 'Production.ProductModel' table. It is an 'AFTER' trigger for 'INSERT' operations. The trigger body consists of an 'INSERT INTO' statement into 'Production.ProductModelHst' with columns 'Action', 'ModifiedDate', 'SourceID', and 'UserName', followed by a 'SELECT' statement that inserts 'insert', 'INSERTED.ModifiedDate', and 'ProductModelID' into the corresponding columns of the target table. The script ends with 'GO'. The status bar shows '110 %' zoom. The 'Сообщения' pane displays 'Выполнение команд успешно завершено.' and the execution time 'Время выполнения: 2020-11-05T19:21:16.9464932+03:00'.

```

/* b) Создайте один AFTER триггер для трех операций INSERT, UPDATE, DELETE для таблицы Production.ProductModel.
   Триггер должен заполнять таблицу Production.ProductModelHst с указанием типа операции в поле Action
   в зависимости от оператора, вызвавшего триггер.
   */

CREATE TRIGGER Production.OnProductModelInsert
ON Production.ProductModel
AFTER
INSERT
AS
INSERT INTO Production.ProductModelHst
(
    Action,
    ModifiedDate,
    SourceID,
    UserName
)
SELECT
    'insert',
    INSERTED.ModifiedDate,
    ProductModelID,

```

Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2020-11-05T19:21:16.9464932+03:00

1c

The screenshot shows the same SQL Server Enterprise Manager window. The active tab contains a T-SQL script for creating a view. The comment in Russian reads: '/* c) Создайте представление VIEW, отображающее все поля таблицы Production.ProductModel. */'. The script is a 'CREATE VIEW' statement for 'Production.ProductModelView'. The view definition is 'AS SELECT * FROM Production.ProductModel;'. The script ends with 'GO'. The status bar shows '110 %' zoom. The 'Сообщения' pane displays 'Выполнение команд успешно завершено.' and the execution time 'Время выполнения: 2020-11-05T19:22:13.0617788+03:00'.

```

/* c) Создайте представление VIEW, отображающее все поля таблицы Production.ProductModel.
   */

CREATE VIEW
    Production.ProductModelView
AS
SELECT *
FROM Production.ProductModel;
GO

```

Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2020-11-05T19:22:13.0617788+03:00

```

/* d) Вставьте новую строку в Production.ProductModel через представление.
Обновите вставленную строку. Удалите вставленную строку.
Убедитесь, что все три операции отображены в Production.ProductModelHst. */

```

```

INSERT INTO Production.ProductModel (Name)
VALUES ('The Main Man');
GO

```

```

UPDATE Production.ProductModel
SET Name = 'Yorn Parsil'
WHERE Name = 'The Main Man';
GO

```

```

DELETE FROM Production.ProductModel
WHERE Name = 'Yorn Parsil';
GO

```

110 %

Сообщения

(затронута одна строка)

(затронута одна строка)

(затронута одна строка)

(затронута одна строка)

(затронута одна строка)

(затронута одна строка)

Время выполнения: 2020-11-05T19:22:50.7536303+03:00

V4_2.sql - THE-MA...IN-MAN\Erop (71)

V4_1.sql - THE-MA...IN-MAN\Erop (57)*

```

/* d) Вставьте новую строку в Production.ProductModel через представление.
Обновите вставленную строку. Удалите вставленную строку.
Убедитесь, что все три операции отображены в Production.ProductModelHst. */

```

```

INSERT INTO Production.ProductModel (Name)
VALUES ('The Main Man');
GO

```

```

UPDATE Production.ProductModel
SET Name = 'Yorn Parsil'
WHERE Name = 'The Main Man';
GO

```

```

DELETE FROM Production.ProductModel
WHERE Name = 'Yorn Parsil';
GO

```

```

SELECT * FROM Production.ProductModelHst;
GO

```

110 %

Результаты

Сообщения

	ID	Action	ModifiedDate	SourceID	UserName
1	1	insert	2020-11-05 19:22:50.547	129	dbo
2	2	update	2020-11-05 19:22:50.547	129	dbo
3	3	delete	2020-11-05 19:22:50.547	129	dbo

2a

```

V4_2.sql - THE-MA...IN-MAN\Erop (71))*  V4_1.sql - THE-MA...IN-MAN\Erop (57))*
/* a) Создайте представление VIEW, отображающее данные из таблиц Production.ProductModel,
   Production.ProductModelProductDescriptionCulture, Production.Culture и Production.ProductDescription.
   Сделайте невозможным просмотр исходного кода представления.
   Создайте уникальный кластерный индекс в представлении по полям ProductModelID,CultureID. */

CREATE VIEW Production.ProductModelView
WITH ENCRYPTION, SCHEMABINDING
AS
SELECT
    Cult.CultureID,
    Cult.Name AS CultureName,
    Cult.ModifiedDate AS CultureModifiedDate,
    PM.CatalogDescription,
    PM.Instructions,
    PM.Name AS ProductModelName,
    PM.ProductModelID,
    PM.ModifiedDate AS ProductModelModifiedDate,
    PD.[Description],
    PD.ProductDescriptionID,
    PD.rowguid,
    PD.ModifiedDate AS ProductDescriptionModifiedDate,
    PMPDC.ModifiedDate AS PMPDC_ModifiedDate
FROM Production.ProductModel AS PM
JOIN Production.ProductModelProductDescriptionCulture AS PMPDC
    ON PM.ProductModelID = PMPDC.ProductModelID
JOIN Production.Culture AS Cult
    ON Cult.CultureID = PMPDC.CultureID
JOIN Production.ProductDescription AS PD
    ON PD.ProductDescriptionID = PMPDC.ProductDescriptionID;

```

110 %

Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2020-11-05T19:34:51.3888546+03:00

110 %

Запрос успешно выполнен. THE-MAIN-MAN (15.0 RTM) THE

2b

```

V4_2.sql - THE-MA...IN-MAN\Erop (71))*  V4_1.sql - THE-MA...IN-MAN\Erop (57))*
/* b) Создайте три INSTEAD OF триггера для представления на операции INSERT, UPDATE, DELETE.
   Каждый триггер должен выполнять соответствующие операции в таблицах Production.ProductModel,
   Production.ProductModelProductDescriptionCulture, Production.Culture и Production.ProductDescription.
   Обновление не должно происходить в таблице Production.ProductModelProductDescriptionCulture.
   Удаление строк из таблиц Production.ProductModel, Production.Culture и Production.ProductDescription
   производите только в том случае, если удаляемые строки больше не ссылаются на Production.ProductModelProductDescriptionCulture.*/

CREATE TRIGGER Production.OnInsertIntoProductModelView
ON Production.ProductModelView
INSTEAD OF INSERT
AS
BEGIN
    INSERT INTO Production.ProductModel(Name)
    SELECT ProductModelName
    FROM INSERTED;

    INSERT INTO Production.Culture(CultureID, Name)
    SELECT CultureID, CultureName
    FROM INSERTED;

    INSERT INTO Production.ProductDescription([Description])
    SELECT [Description]
    FROM INSERTED;

    INSERT INTO Production.ProductModelProductDescriptionCulture(CultureID, ProductModelID, ProductDescriptionID)
    VALUES ((SELECT CultureID FROM INSERTED),
            IDENT_CURRENT('Production.ProductModel'),
            IDENT_CURRENT('Production.ProductDescription'));
END;

```

110 %

Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2020-11-05T19:36:38.1772539+03:00

```
V4_2.sql - THE-MA...IN-MAN\Erop (56) [X]
/* с) Вставьте новую строку в представление, указав новые данные для ProductModel, Culture и ProductDescription.
   Триггер должен добавить новые строки в таблицы Production.ProductModel, Production.ProductModelProductDescriptionCulture,
   Production.Culture и Production.ProductDescription. Обновите вставленные строки через представление. Удалите строки.*/

INSERT INTO Production.ProductModelView
(
    ProductModelName,
    CultureID,
    CultureName,
    [Description]
)
VALUES
```

Сообщения

(затронута одна строка)
(затронута одна строка)
(затронута одна строка)
(затронута одна строка)
(затронута одна строка)
(затронута одна строка)
(затронута одна строка)
(затронута одна строка)
(затронута одна строка)
(затронута одна строка)

Время выполнения: 2020-11-05T20:23:01.2881852+03:00

```
SELECT * FROM Production.ProductModelView
WHERE
    CultureID = 'me';
GO
```

110 %

Результаты Сообщения

	CultureID	CultureName	CultureModifiedDate	CatalogDescription	Instructions	ProductModelName	ProductModelID	ProductModelModifiedDate	Description	ProductDescriptionID	rowguid	ProductDescriptionMod
1	me	Mexico	2020-11-06 01:01:26.423	NULL	NULL	Taco	135	2020-11-06 01:01:26.260	Cheese	2012	1746EB90-2776-4FE2-83EA-380164247079	2020-11-06 01:01:26.44

```
UPDATE Production.ProductModelView
SET
    ProductModelName = 'Burito',
    CultureName = 'Mexican',
    [Description] = 'Mexican Shawarma'
WHERE
    CultureID = 'me'
    AND ProductModelID = IDENT_CURRENT('Production.ProductModel')
    AND ProductDescriptionID = IDENT_CURRENT('Production.ProductDescription');
GO
```

110 %

Сообщения

(затронута одна строка)

(затронута одна строка)

(затронута одна строка)

(затронута одна строка)

(затронута одна строка)

Время выполнения: 2020-11-06T01:03:16.4204192+03:00

```
SELECT * FROM Production.ProductModelView  
WHERE  
CultureID = 'me';  
GO
```

110 %

Результаты Сообщения

	CultureID	CultureName	CultureModifiedDate	CatalogDescription	Instructions	ProductModelName	ProductModelID	ProductModelModifiedDate	Description	ProductDescriptionID	rowguid	ProductDescri
1	me	Mexican	2020-11-06 01:03:16.400	NULL	NULL	Burrito	135	2020-11-06 01:03:16.353	Mexican Shawarma	2012	1746EB90-2776-4FE2-83EA-380164247079	2020-11-06 0

```
DELETE FROM Production.ProductModelView  
WHERE  
CultureID = 'me'  
AND ProductModelID = IDENT_CURRENT('Production.ProductModel')  
AND ProductDescriptionID = IDENT_CURRENT('Production.ProductDescription');  
GO
```

110 %

Сообщения

(затронута одна строка)

Время выполнения: 2020-11-06T01:04:21.5630148+03:00