

Sistemas en Tiempo Real y Distribuidos

Nombres y Apellidos: Boris Orlando Roque Huanca

C.I.: 9868420 LP

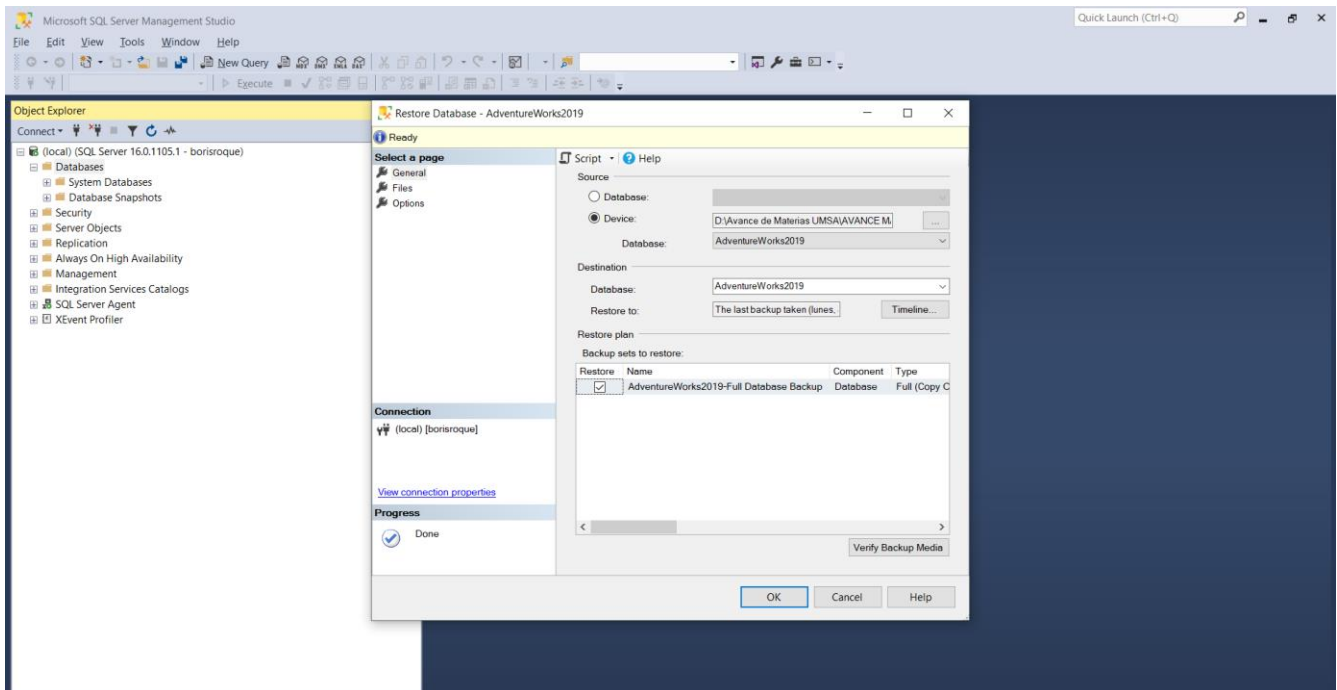
Docente: Ph. D. Moisés Martin Silva Choque

Fecha de Entrega: 12/12/2023

2do Examen Parcial - Ejercicio 1

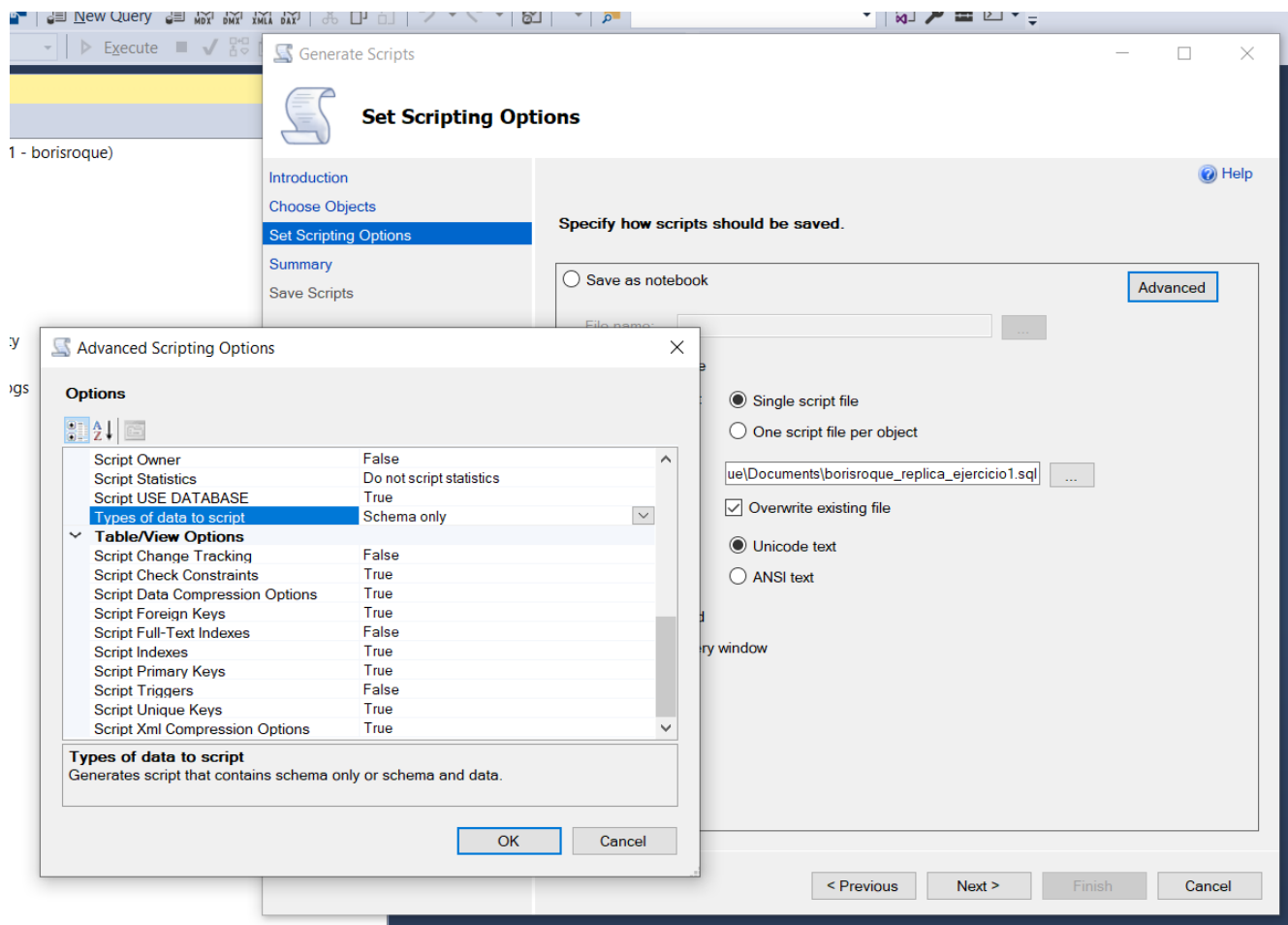
Mediante el uso del Agente genere una réplica de una base de datos del repositorio de Microsoft, de al menos dos tablas

1. La resolución del ejercicio inicia con la Restauración de la base de datos “Adventure Works 2019” descargada de la plataforma Adventure Works Sample Databases de Microsoft.

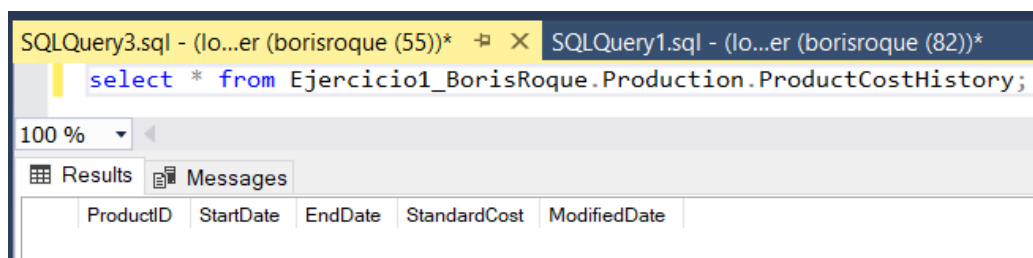


- [-] (local) (SQL Server 16.0.1105.1 - borisroque)
 - [+] Databases
 - [+] System Databases
 - [+] Database Snapshots
 - [+] **AdventureWorks2019**
 - [+] Security
 - [+] Server Objects
 - [+] Replication
 - [+] Always On High Availability
 - [+] Management
 - [+] Integration Services Catalogs
 - [+] SQL Server Agent
 - [+] XEvent Profiler

2. Se extrae el script sin datos de la base de datos “AdventureWorks2019” para crear la base de datos donde se replicará la mencionada base.



3. Para la creación de la base de datos y las tablas donde se replicarán los datos se ejecutan las líneas de código adjuntas al presente documento bajo el nombre: “borisroque_2doparcial_ejercicio1.sql”. Como resultado se tiene la misma estructura, pero ésta se encuentra vacía.

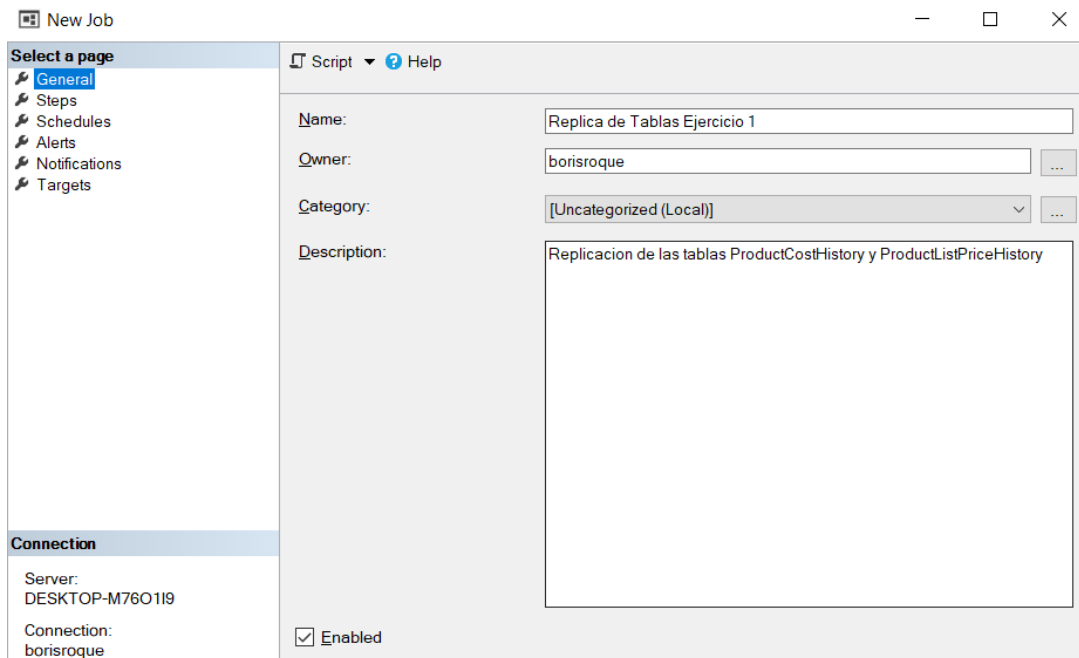


4. Se replicarán las tablas: “Production.ProductCostHistory” y “Production.ProductListPriceHistory”. Para evitar problemas debido a las PK y FK se ejecutarán las siguientes líneas de código.

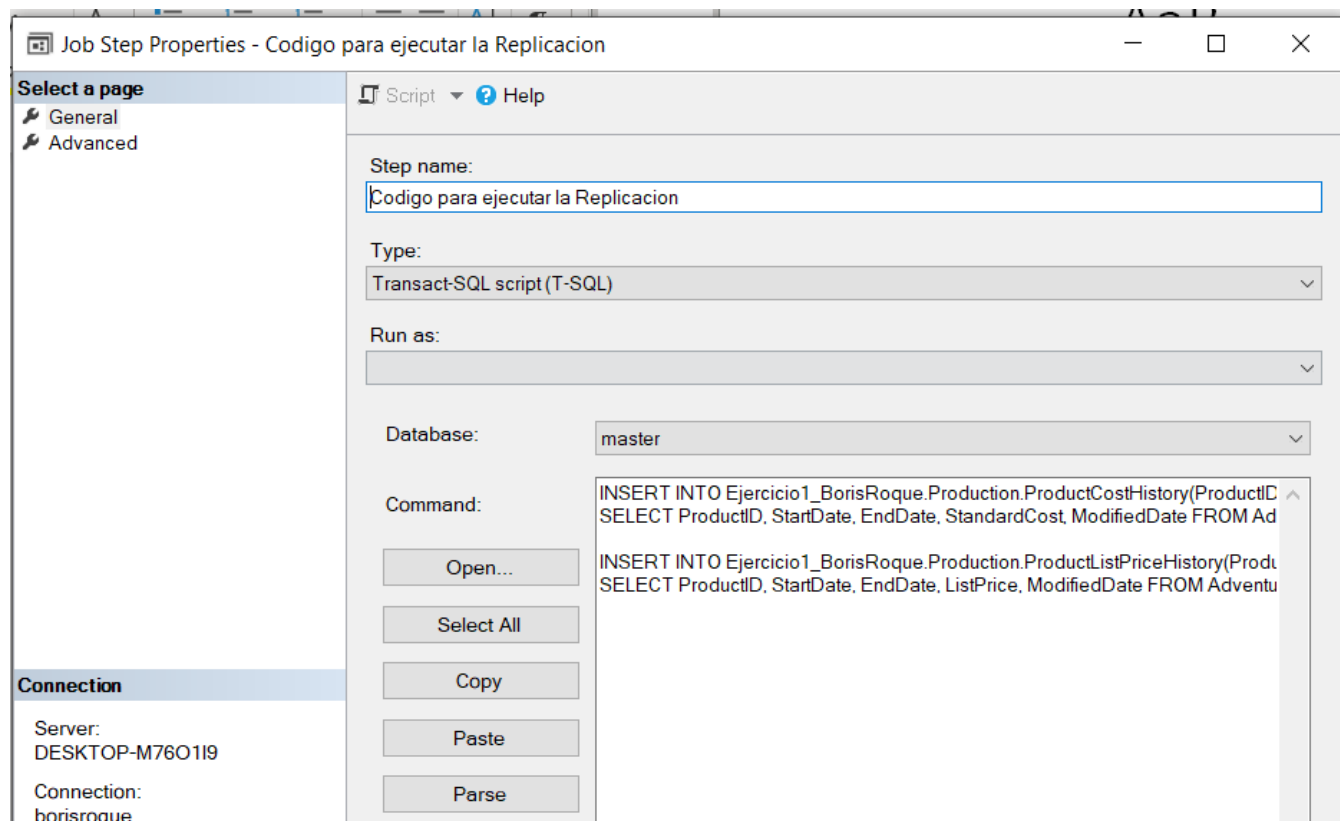
```
ALTER TABLE Ejercicio1_BorisRoque.Production.ProductCostHistory NOCHECK
CONSTRAINT FK_ProductCostHistory_Product_ProductID;
```

ALTER TABLE Ejercicio1_BorisRoque.Production.ProductListPriceHistory **NOCHECK**
CONSTRAINT FK_ProductListPriceHistory_Product_ProductID;

Luego, se inicia el Agente de SQL Server y se crea un nuevo trabajo.



Y se crea un paso para replicar las tablas.



El código implementado para realizar la réplica es el siguiente:

```
INSERT INTO Ejercicio1_BorisRoque.Production.ProductCostHistory(ProductID,
StartDate, EndDate, StandardCost, ModifiedDate)
SELECT ProductID, StartDate, EndDate, StandardCost, ModifiedDate FROM
AdventureWorks2019.Production.ProductCostHistory;
```

```
INSERT INTO
Ejercicio1_BorisRoque.Production.ProductListPriceHistory(ProductID,
StartDate, EndDate, ListPrice, ModifiedDate)
SELECT ProductID, StartDate, EndDate, ListPrice, ModifiedDate FROM
AdventureWorks2019.Production.ProductListPriceHistory;
```

Se crea un Horario para la ejecución de la replicación. Para este caso, se modifica de tal manera que se repita una sola vez.

5. Para verificar la replicación se ejecuta el siguiente código.

```
select * from Ejercicio1_BorisRoque.Production.ProductListPriceHistory;
select * from Ejercicio1_BorisRoque.Production.ProductCostHistory;
```

	ProductID	StartDate	EndDate	ListPrice	ModifiedDate
4	708	2011-05-31 00:00:00.000	2012-05-29 00:00:00.000	33,6442	2012-05-29 00:00:00.000
5	708	2012-05-30 00:00:00.000	2013-05-29 00:00:00.000	33,6442	2013-05-29 00:00:00.000
6	708	2013-05-30 00:00:00.000	NULL	34,99	2013-05-09 00:00:00.000

	ProductID	StartDate	EndDate	StandardCost	ModifiedDate
1	707	2011-05-31 00:00:00.000	2012-05-29 00:00:00.000	12,0278	2012-05-29 00:00:00.000
2	707	2012-05-30 00:00:00.000	2013-05-29 00:00:00.000	13,8782	2013-05-29 00:00:00.000
3	707	2013-05-30 00:00:00.000	NULL	13,0863	2013-05-16 00:00:00.000
4	708	2011-05-31 00:00:00.000	2012-05-29 00:00:00.000	12,0278	2012-05-29 00:00:00.000
5	708	2012-05-30 00:00:00.000	2013-05-29 00:00:00.000	13,8782	2013-05-29 00:00:00.000
6	708	2013-05-30 00:00:00.000	NULL	13,0863	2013-05-16 00:00:00.000