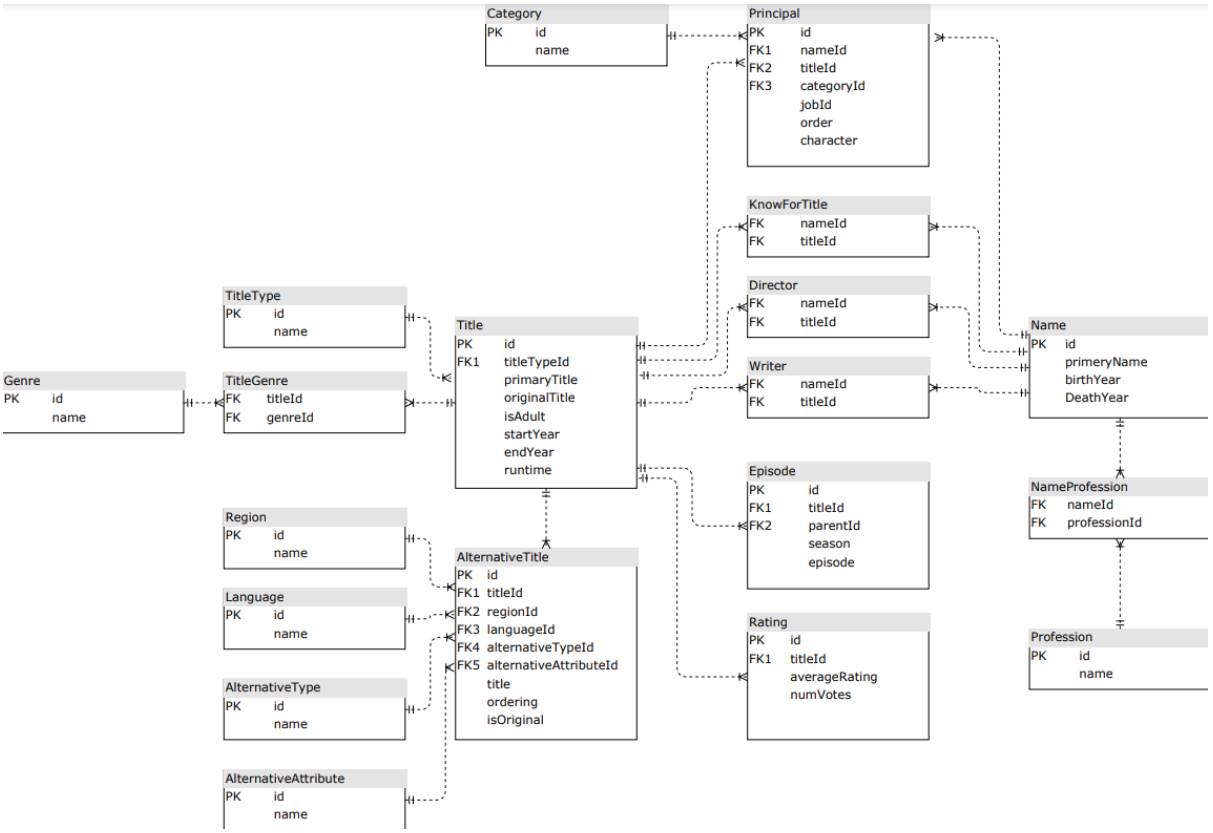
Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Sistemas de bases de datos 2

Manual técnico

Integrantes	Carnet
Erick Alexander Alvarado Guerra	201800546
Jaime Ismael Belloso Garcia	201325557

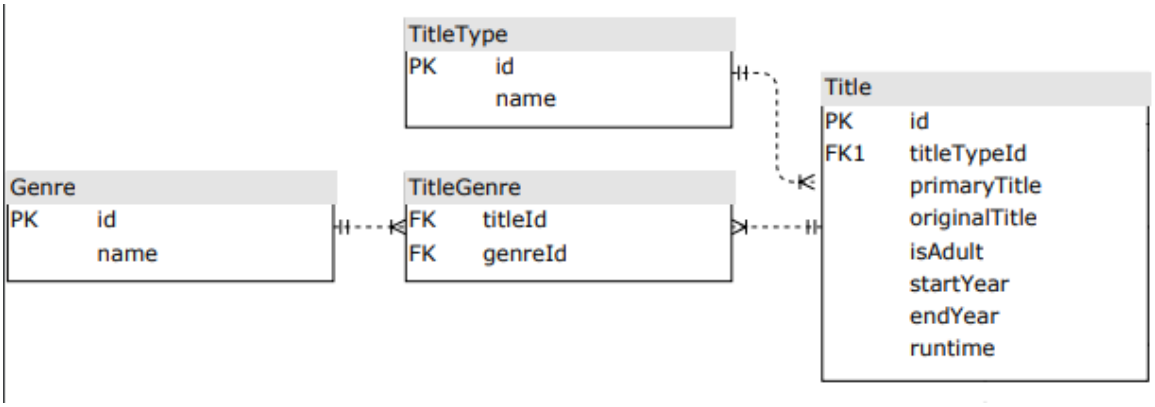
Justificación

Modelo IMDB

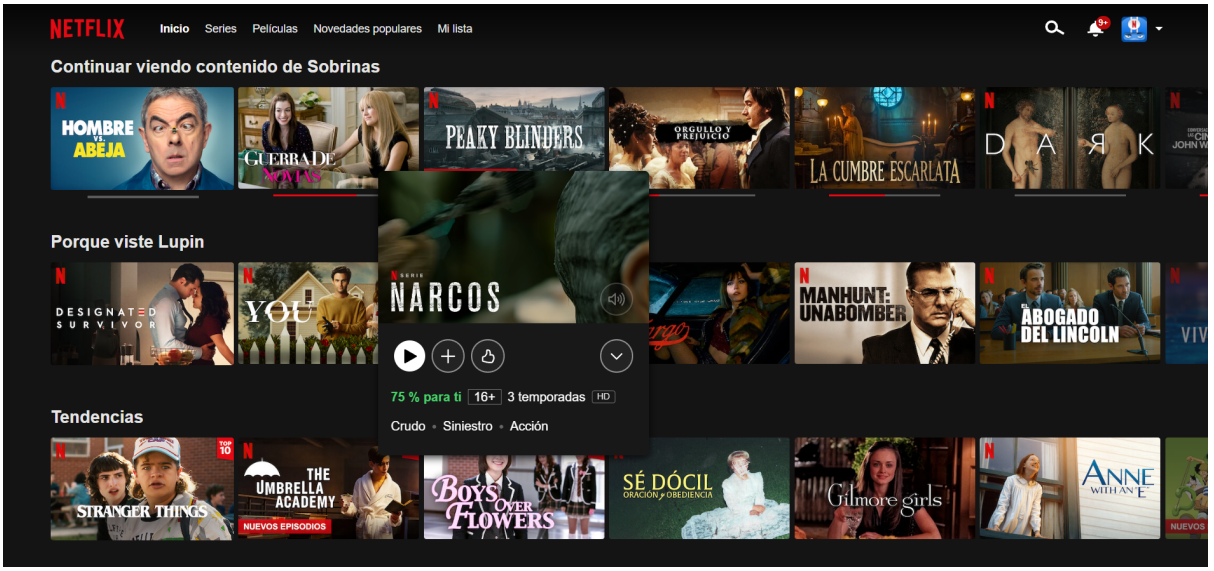


Modelo Netflix

Para el modelo de netflix se hará uso de las tablas más primordiales, siendo estas el título, el género y el tipo de contenido (serie o película). No se considera necesario el obtener los episodios directamente de IMDB ya que Netflix comúnmente solo posee permisos para reproducir ciertas temporadas y capítulos específicos en distintas regiones. Asimismo el rating generado de IMDB no es considerado por Netflix, ya que se maneja un sistema propio.



- Página referencia



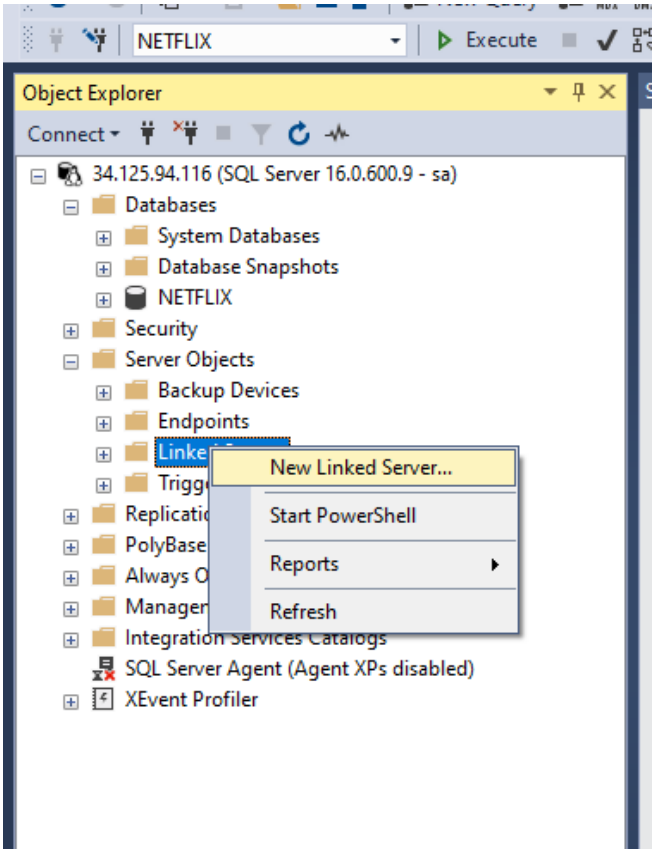
Modelo MongoDB

En esta base de datos debe almacenarse solamente la información que sea comúnmente accedida por usuarios, se determinó que la tabla títulos es la más adecuada, ya que siempre al acceder al inicio de netflix se obtienen todos los títulos almacenados. El formato de documento por utilizar es el siguiente:

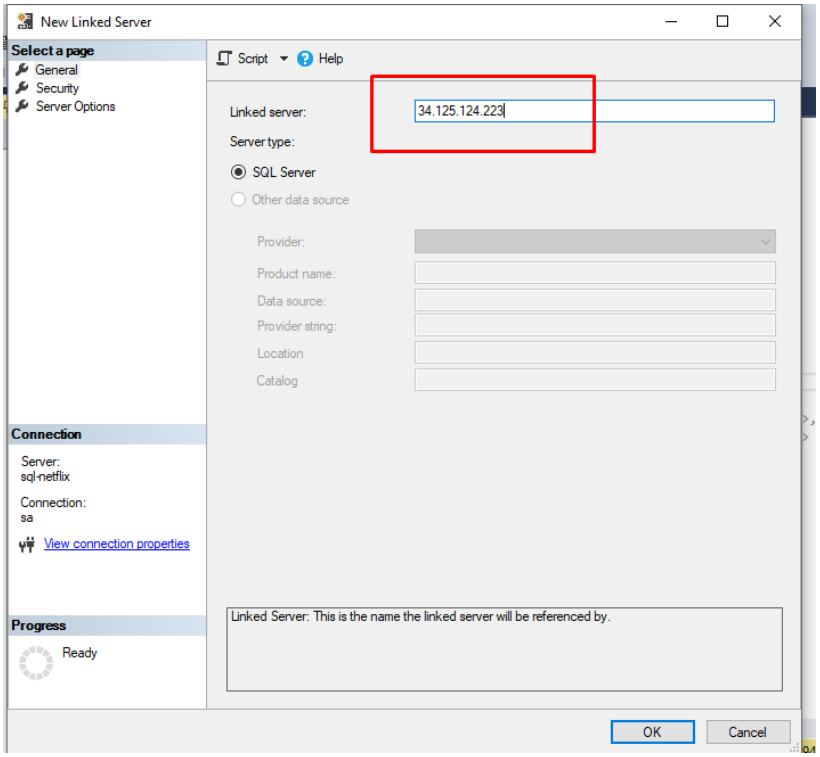
```
_id: ObjectId('62bdf3176a46e883ec88a9f2')
id: 0
primaryTitle: "Matrix"
originalTitle: "Default"
isAdult: "d"
startYear: "10/10/2004"
endYear: "10/10/2005"
runTime: "02:00:00"
```

Pasos para crear Linked Server

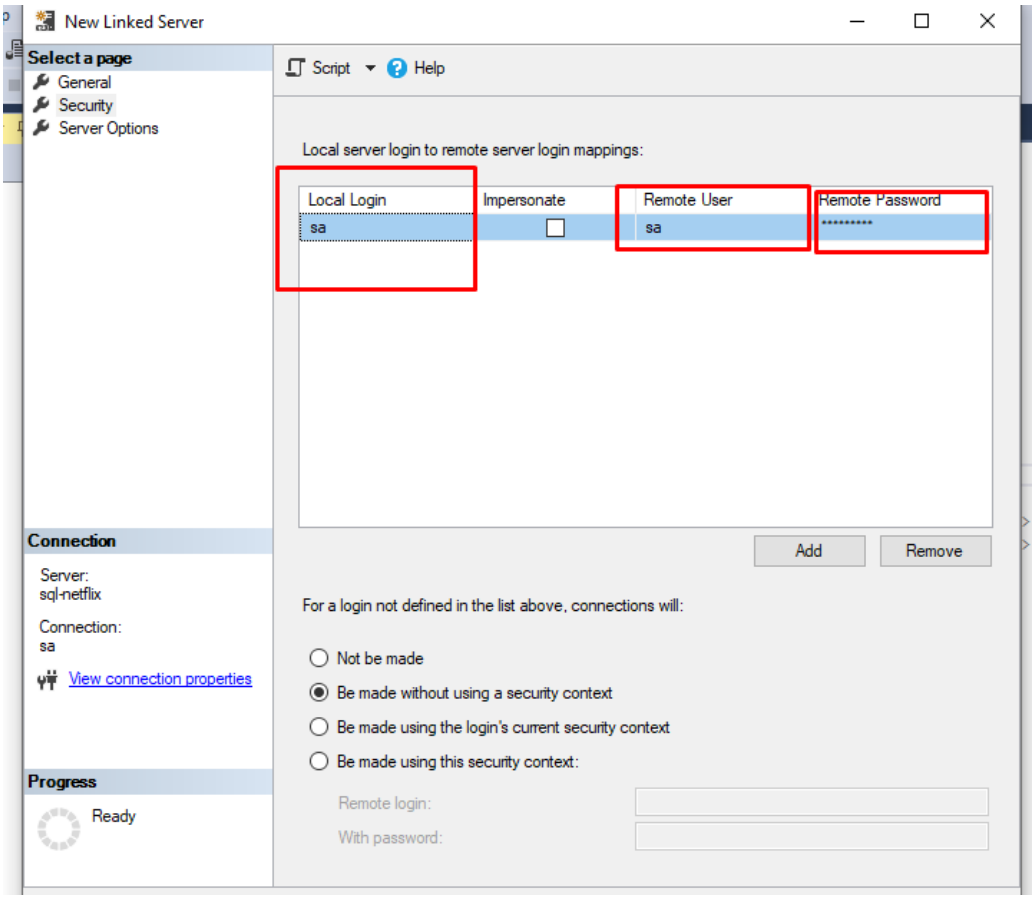
- Dirigirse a Server Objects -> click derecho sobre linked server -> New Linked Server



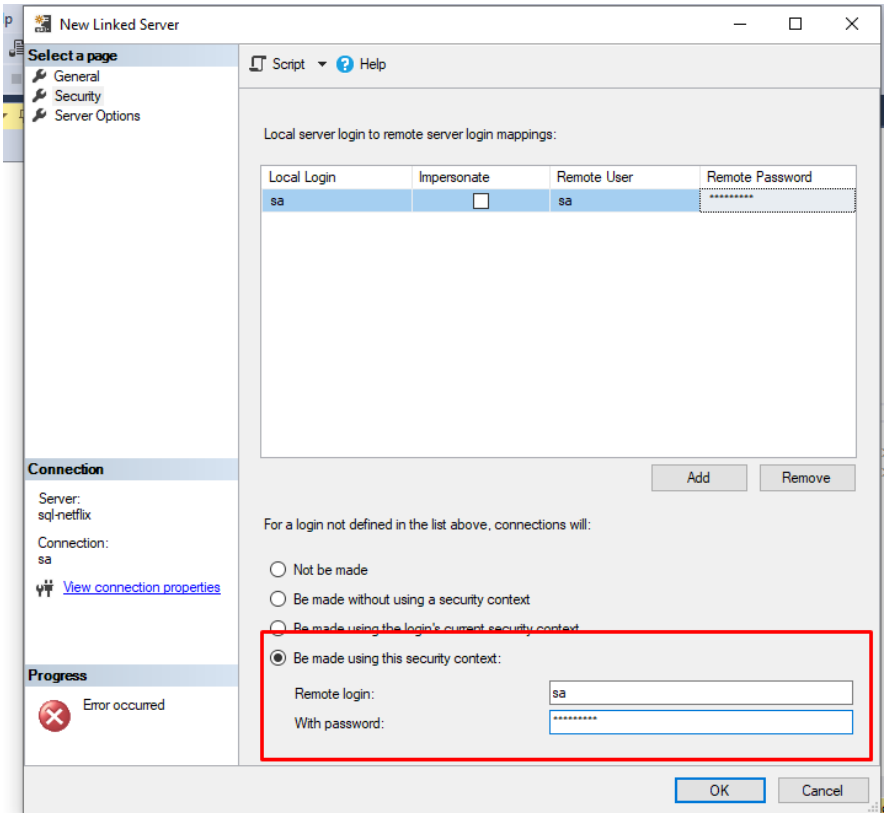
- Configurar servidor externo



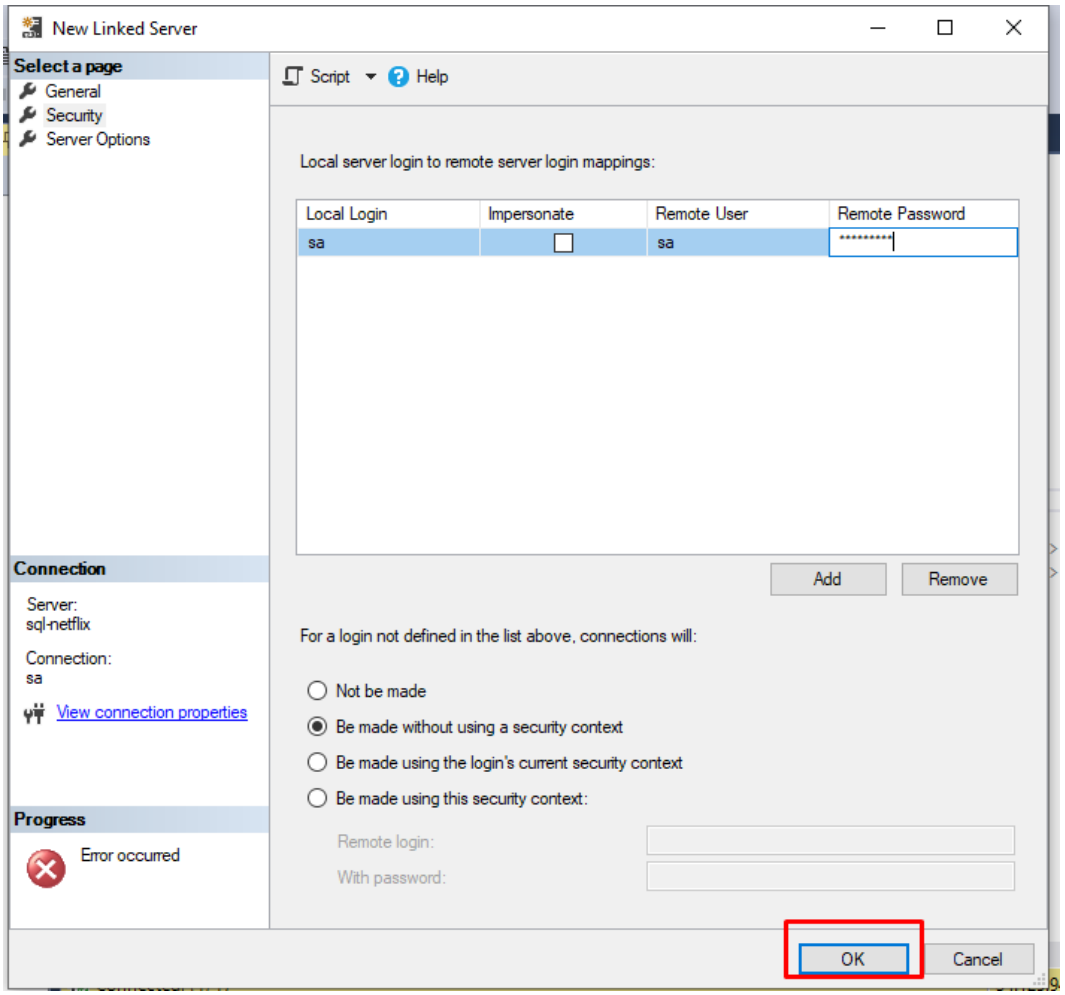
- Agregar usuario de autentificación

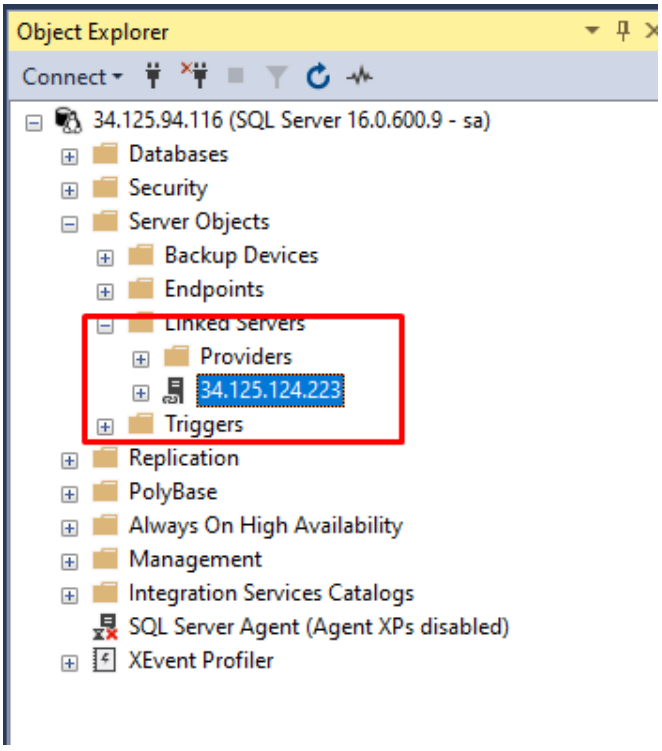


- Agregar credenciales de autenticación externas



- Click en OK

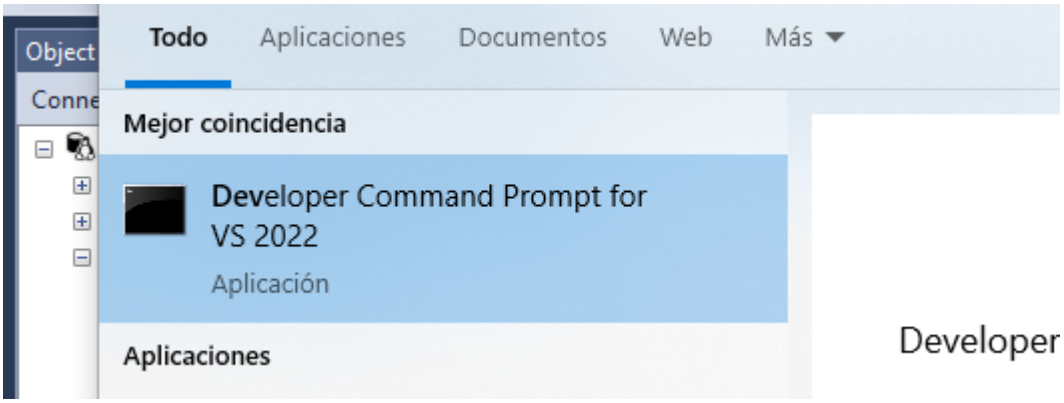

New Linked Server
Select a page
General
Security
Server Options
Script
Help
Local server login to remote server login mappings:
Local Login
Impersonate
Remote User
Remote Password
sa
sa
Add
Remove
Connection
Server:
sql-netflix
Connection:
sa
View connection properties
For a login not defined in the list above, connections will:
Not be made
Be made without using a security context
Be made using the login's current security context
Be made using this security context:
Remote login:
With password:
Progress
Error occurred
OK
Cancel


Object Explorer
Connect
34.125.94.116 (SQL Server 16.0.600.9 - sa)
Databases
Security
Server Objects
Backup Devices
Endpoints
Linked Servers
Providers
34.125.124.223
Triggers
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)
XEvent Profiler

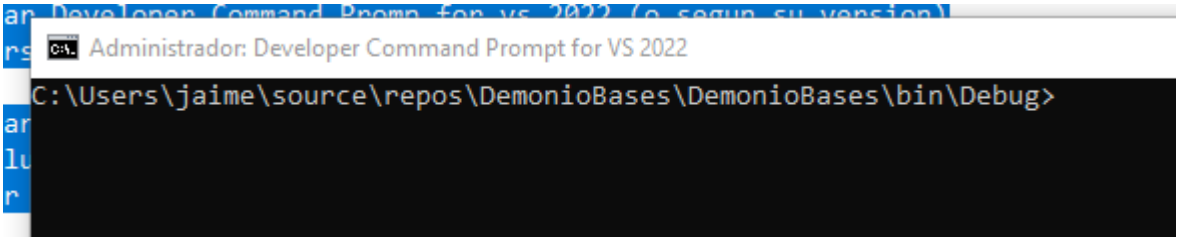
Demonio SQL Server - Mono
El demonio consiste en un servicio de windows que se ejecuta cada cierto tiempo, este lee los datos de la base de datos de Netflix y los sincroniza hacia mongoDB

Pasos para instalar:

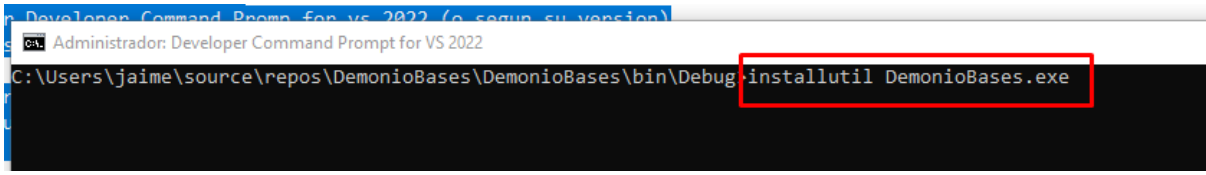
- Descomprimir carpeta Demonio
- Ejecutar Developer Command Prompt for vs 2022 (o según su versión)



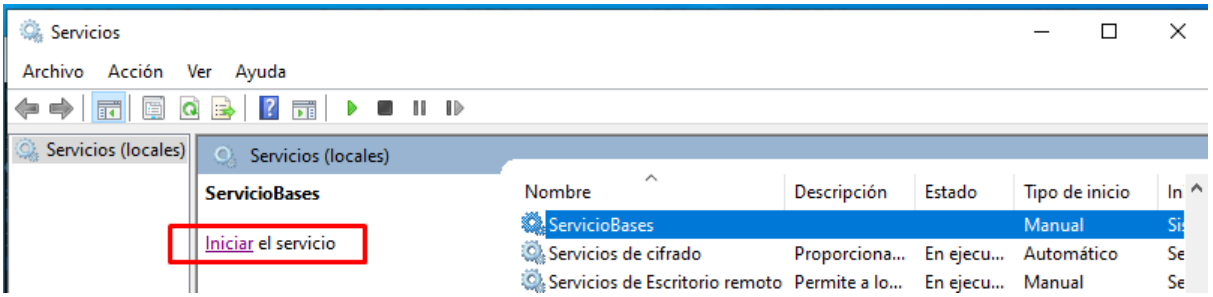
- colocarse en la carpeta descomprimida Ejemplo CD C:/Demonio/



- Ejecutar comando para instalar
  installutil DemonioBases.exe



- iniciar servicio



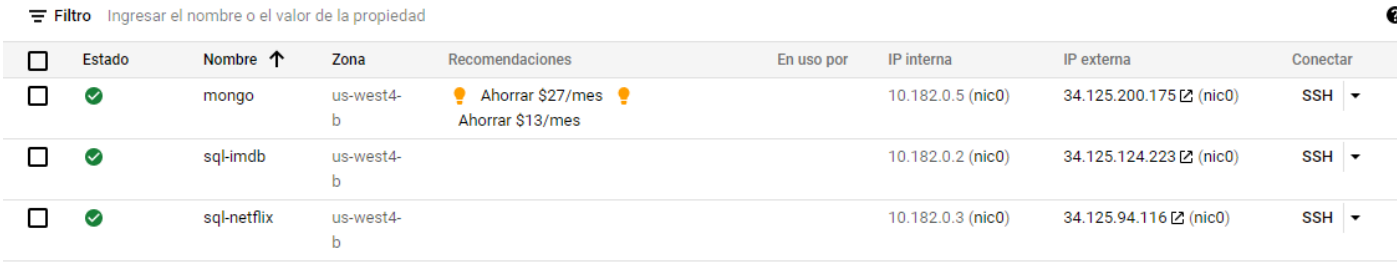
Nota: Para detener el servicio se debe de

1. Detener el servicio
2. Ejecutar el comando “installutil /u DemonioBases.exe”

Servidores de Bases de Datos:
Los servidores de base de datos fueron creados en Máquinas Virtuales de GCP, cada uno en diferente máquina virtual.

Direcciones de Servidores:

- MongoDB: 34.125.200.175:27017
- SQL Server Netflix: 34.125.94.116
- SQL Server IMDB: 34.125.124.223



Procedimiento almacenado para inserción de datos en Netflix.

```
CREATE PROCEDURE INSERTAR_PELICULA @ID INT
	,@primaryTitle VARCHAR(100)
	,@originalTitle VARCHAR(100)
	,@isAdult VARCHAR(10)
	,@startYear DATE
	,@endYear DATE
	,@runtime TIME(7)
	,@TitleTypeId INT
	,@genreId INT
AS
BEGIN
	DECLARE @ID_IMDB INT

	SELECT @ID_IMDB = [id]
	FROM [34.125.124.223].[IMDB].[dbo].[Title]
	WHERE [primaryTitle] = @primaryTitle

	SET NOCOUNT ON;

	IF @ID_IMDB > 0
	BEGIN
		INSERT INTO Title
		VALUES (
			@ID
			,@primaryTitle
			,@originalTitle
			,@isAdult
			,@startYear
			,@endYear
```

```
                    ,@runtime
                    ,@TitleTypeId
                    ,@ID_IMDB
                    ,0
                    );

                INSERT INTO TitleGenre
                VALUES (
                    @ID
                    ,@genreId
                    );
        END
END
GO
```

Ejemplo de Ejecución

```
EXEC INSERTAR_PELICULA 1,'Matrix','Matrix','no','11/12/2005','11/12/2005','03:00:00',1,2;
```

DDL IMDB

```
CREATE TABLE AlternativeAttribute
   (
    id INTEGER NOT NULL ,
    name VARCHAR (100)
   )
GO

ALTER TABLE AlternativeAttribute ADD CONSTRAINT AlternativeAttribute_PK PRIMARY KEY CLUSTERED (id)
    WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE AlternativeTitle
   (
    id INTEGER NOT NULL ,
    titleId INTEGER NOT NULL ,
    regionId INTEGER NOT NULL ,
    languageId INTEGER NOT NULL ,
    alternativeTypeId INTEGER NOT NULL ,
    alternativeAttributeId INTEGER NOT NULL ,
    title VARCHAR (100) ,
    ordering VARCHAR (100) ,
    isOriginal VARCHAR (100)
   )
GO

ALTER TABLE AlternativeTitle ADD CONSTRAINT AlternativeTitle_PK PRIMARY KEY CLUSTERED (id)
    WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE AlternativeType
   (
    id INTEGER NOT NULL ,
    name VARCHAR (100) NOT NULL
   )
GO

ALTER TABLE AlternativeType ADD CONSTRAINT AlternativeType_PK PRIMARY KEY CLUSTERED (id)
    WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Category
   (
    id INTEGER NOT NULL ,
    name VARCHAR (100)
   )
GO

ALTER TABLE Category ADD CONSTRAINT Category_PK PRIMARY KEY CLUSTERED (id)
    WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Director
   (
    nameId INTEGER NOT NULL ,
    titleId INTEGER NOT NULL
```

```
    )
GO

CREATE TABLE Episode
    (
     id INTEGER NOT NULL ,
     titleId INTEGER NOT NULL ,
     season VARCHAR (100) ,
     episode VARCHAR (100)
    )
GO

ALTER TABLE Episode ADD CONSTRAINT Episode_PK PRIMARY KEY CLUSTERED (id)
     WITH (
     ALLOW_PAGE_LOCKS = ON ,
     ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Genre
    (
     id INTEGER NOT NULL ,
     name VARCHAR (100)
    )
GO

ALTER TABLE Genre ADD CONSTRAINT Genre_PK PRIMARY KEY CLUSTERED (id)
     WITH (
     ALLOW_PAGE_LOCKS = ON ,
     ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE KnowForTitle
    (
     nameId INTEGER NOT NULL ,
     titleId INTEGER NOT NULL
    )
GO

CREATE TABLE Language
    (
     id INTEGER NOT NULL ,
     name VARCHAR (100)
    )
GO

ALTER TABLE Language ADD CONSTRAINT TABLE_4_PK PRIMARY KEY CLUSTERED (id)
     WITH (
     ALLOW_PAGE_LOCKS = ON ,
     ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Name
    (
     id INTEGER NOT NULL ,
     primeryName VARCHAR (100) ,
     birthYear DATE ,
     deathYear DATE
    )
GO

ALTER TABLE Name ADD CONSTRAINT Name_PK PRIMARY KEY CLUSTERED (id)
     WITH (
     ALLOW_PAGE_LOCKS = ON ,
     ALLOW_ROW_LOCKS = ON )
GO
```

```
CREATE TABLE NameProfession
   (
    nameId INTEGER NOT NULL ,
    professionId INTEGER NOT NULL
   )
GO

CREATE TABLE Principal
   (
    id INTEGER NOT NULL ,
    nameId INTEGER NOT NULL ,
    titleId INTEGER NOT NULL ,
    categoryId INTEGER NOT NULL ,
    jobId INTEGER ,
    "order" INTEGER ,
    character VARCHAR (100)
   )
GO

ALTER TABLE Principal ADD CONSTRAINT Principal_PK PRIMARY KEY CLUSTERED (id)
    WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Profession
   (
    id INTEGER NOT NULL ,
    name VARCHAR (100)
   )
GO

ALTER TABLE Profession ADD CONSTRAINT Profession_PK PRIMARY KEY CLUSTERED (id)
    WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Rating
   (
    id INTEGER NOT NULL ,
    titleId INTEGER NOT NULL ,
    averageRating INTEGER ,
    numVotes INTEGER
   )
GO

ALTER TABLE Rating ADD CONSTRAINT Rating_PK PRIMARY KEY CLUSTERED (id)
    WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Region
   (
    id INTEGER NOT NULL ,
    name VARCHAR (100)
   )
GO

ALTER TABLE Region ADD CONSTRAINT Region_PK PRIMARY KEY CLUSTERED (id)
    WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO
```

```
CREATE TABLE Title
   (
    id INTEGER NOT NULL ,
    primaryTitle VARCHAR (100) ,
    originalTitle VARCHAR (100) ,
    isAdult VARCHAR (10) ,
    startYear DATE ,
    endYear DATE ,
    runtime TIME ,
    TitleTypeId INTEGER NOT NULL
   )
GO

ALTER TABLE Title ADD CONSTRAINT Title_PK PRIMARY KEY CLUSTERED (id)
    WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE TitleGenre
   (
    titleId INTEGER NOT NULL ,
    genreId INTEGER NOT NULL
   )
GO

CREATE TABLE TitleType
   (
    id INTEGER NOT NULL ,
    name VARCHAR (100)
   )
GO

ALTER TABLE TitleType ADD CONSTRAINT TitleType_PK PRIMARY KEY CLUSTERED (id)
    WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Writer
   (
    nameId INTEGER NOT NULL ,
    titleId INTEGER NOT NULL
   )
GO

ALTER TABLE AlternativeTitle
   ADD CONSTRAINT alternativeAttributeId FOREIGN KEY
   (
    alternativeAttributeId
   )
   REFERENCES AlternativeAttribute
   (
    id
   )
   ON DELETE NO ACTION
   ON UPDATE NO ACTION
GO

ALTER TABLE AlternativeTitle
   ADD CONSTRAINT alternativeTypeId FOREIGN KEY
   (
    alternativeTypeId
   )
   REFERENCES AlternativeType
   (
```

```
    id
   )
   ON DELETE NO ACTION
   ON UPDATE NO ACTION
GO

ALTER TABLE Principal
   ADD CONSTRAINT categoryId FOREIGN KEY
   (
    categoryId
   )
   REFERENCES Category
   (
    id
   )
   ON DELETE NO ACTION
   ON UPDATE NO ACTION
GO

ALTER TABLE TitleGenre
   ADD CONSTRAINT genreId FOREIGN KEY
   (
    genreId
   )
   REFERENCES Genre
   (
    id
   )
   ON DELETE NO ACTION
   ON UPDATE NO ACTION
GO

ALTER TABLE AlternativeTitle
   ADD CONSTRAINT languageId FOREIGN KEY
   (
    languageId
   )
   REFERENCES Language
   (
    id
   )
   ON DELETE NO ACTION
   ON UPDATE NO ACTION
GO

ALTER TABLE NameProfession
   ADD CONSTRAINT nameId FOREIGN KEY
   (
    nameId
   )
   REFERENCES Name
   (
    id
   )
   ON DELETE NO ACTION
   ON UPDATE NO ACTION
GO

ALTER TABLE Writer
   ADD CONSTRAINT nameIdv1 FOREIGN KEY
   (
    nameId
   )
   REFERENCES Name
   (
    id
```

```
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE Director
    ADD CONSTRAINT nameIdv2 FOREIGN KEY
    (
     nameId
    )
    REFERENCES Name
    (
     id
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE KnowForTitle
    ADD CONSTRAINT nameIdv3 FOREIGN KEY
    (
     nameId
    )
    REFERENCES Name
    (
     id
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE Principal
    ADD CONSTRAINT nameIdv4 FOREIGN KEY
    (
     nameId
    )
    REFERENCES Name
    (
     id
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE NameProfession
    ADD CONSTRAINT professionId FOREIGN KEY
    (
     professionId
    )
    REFERENCES Profession
    (
     id
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE AlternativeTitle
    ADD CONSTRAINT regionId FOREIGN KEY
    (
     regionId
    )
    REFERENCES Region
    (
     id
    )
```

```
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE TitleGenre
    ADD CONSTRAINT titleId FOREIGN KEY
    (
     titleId
    )
    REFERENCES Title
    (
     id
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE AlternativeTitle
    ADD CONSTRAINT titleIdv1 FOREIGN KEY
    (
     titleId
    )
    REFERENCES Title
    (
     id
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE Episode
    ADD CONSTRAINT titleIdv2 FOREIGN KEY
    (
     titleId
    )
    REFERENCES Title
    (
     id
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE Rating
    ADD CONSTRAINT titleIdv3 FOREIGN KEY
    (
     titleId
    )
    REFERENCES Title
    (
     id
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE Writer
    ADD CONSTRAINT titleIdv4 FOREIGN KEY
    (
     titleId
    )
    REFERENCES Title
    (
     id
    )
    ON DELETE NO ACTION
```

```
    ON UPDATE NO ACTION
GO

ALTER TABLE Director
    ADD CONSTRAINT titleIdv5 FOREIGN KEY
    (
     titleId
    )
    REFERENCES Title
    (
     id
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE KnowForTitle
    ADD CONSTRAINT titleIdv6 FOREIGN KEY
    (
     titleId
    )
    REFERENCES Title
    (
     id
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE Principal
    ADD CONSTRAINT titleIdv7 FOREIGN KEY
    (
     titleId
    )
    REFERENCES Title
    (
     id
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE Title
    ADD CONSTRAINT titleTypeId FOREIGN KEY
    (
     TitleTypeId
    )
    REFERENCES TitleType
    (
     id
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO
```

DDL Netflix

```
CREATE TABLE Episode
   (
    id INTEGER NOT NULL ,
    titleId INTEGER NOT NULL ,
    season VARCHAR (100) ,
    episode VARCHAR (100)
   )
GO

ALTER TABLE Episode ADD CONSTRAINT Episode_PK PRIMARY KEY CLUSTERED (id)
    WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Genre
   (
    id INTEGER NOT NULL ,
    name VARCHAR (100)
   )
GO

ALTER TABLE Genre ADD CONSTRAINT Genre_PK PRIMARY KEY CLUSTERED (id)
    WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO
```

```
CREATE TABLE Title
   (
    id INTEGER NOT NULL ,
    primaryTitle VARCHAR (100) ,
    originalTitle VARCHAR (100) ,
    isAdult VARCHAR (10) ,
    startYear DATE ,
    endYear DATE ,
    runtime TIME ,
    TitleTypeId INTEGER NOT NULL
   )
GO

ALTER TABLE Title ADD CONSTRAINT Title_PK PRIMARY KEY CLUSTERED (id)
    WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO
```

```
CREATE TABLE TitleGenre
   (
    titleId INTEGER NOT NULL ,
    genreId INTEGER NOT NULL
   )
GO

CREATE TABLE TitleType
   (
    id INTEGER NOT NULL ,
    name VARCHAR (100)
   )
GO

ALTER TABLE TitleType ADD CONSTRAINT TitleType_PK PRIMARY KEY CLUSTERED (id)
    WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO


ALTER TABLE TitleGenre
   ADD CONSTRAINT genreId FOREIGN KEY
   (
    genreId
   )
   REFERENCES Genre
   (
    id
   )
   ON DELETE NO ACTION
   ON UPDATE NO ACTION
GO

ALTER TABLE TitleGenre
   ADD CONSTRAINT titleId FOREIGN KEY
   (
    titleId
   )
   REFERENCES Title
   (
    id
   )
   ON DELETE NO ACTION
   ON UPDATE NO ACTION
GO

ALTER TABLE Episode
   ADD CONSTRAINT titleIdv2 FOREIGN KEY
   (
    titleId
   )
   REFERENCES Title
   (
    id
   )
   ON DELETE NO ACTION
```

```
    ON UPDATE NO ACTION
GO


ALTER TABLE Title
    ADD CONSTRAINT titleTypeId FOREIGN KEY
    (
     TitleTypeId
    )
    REFERENCES TitleType
    (
     id
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO
```

Cargar datos ejemplo

```
insert into Genre values (1, 'accion');
insert into Genre values (2, 'misterio');
insert into Genre values (3, 'suspenso');
insert into Genre values (4, 'romance');
insert into Genre values (5, 'comedia');

insert into TitleType values (1, 'pelicula');
insert into TitleType values (2, 'serie');

insert into Title values (1,'El rey leon','The Lion
King','no','12/12/2002','12/12/2003','03:20:00',1);
insert into Title values (2,'El juego del
miedo','Saw','si','10/12/2002','10/12/2003','02:20:00',1);
insert into Title values
(3,'Matrix','Matrix','no','11/12/2005','11/12/2005','03:00:00',1);
insert into Title values
(4,'Pinocho','Pinocho','no','02/02/2010','03/03/2011','01:20:00',1);
insert into Title values (5,'Como conoci a tu madre','How i met your
mother','si','09/10/2010','09/11/2011','01:20:00',2);

insert into TitleGenre values (1,2)
insert into TitleGenre values (1,5)
insert into TitleGenre values (2,2)
insert into TitleGenre values (2,3)
insert into TitleGenre values (3,1)
insert into TitleGenre values (3,2)
insert into TitleGenre values (4,5)
insert into TitleGenre values (5,5)


insert into Episode values (1,5, 1,1)
insert into Episode values (2,5, 1,2)
insert into Episode values (3,5, 1,3)
insert into Episode values (4,5, 1,4)
insert into Episode values (5,5, 1,5)

select * from Episode
```