Universidad San Carlos de Guatemala

Curso: Sistemas de bases de datos 2

Catedrático: Marlon Orellana

Auxiliar: Jurgen Ramírez

Sección: N

Imagen que contiene Logotipo

Descripción generada automáticamente

Integrantes:

|  |  |
| --- | --- |
| Nombre | Carnet |
| Josué David Zea Herrera | 201807159 |
| Jorge David Espina Molina | 201403632 |
| Kenni Roberto Martínez Marroquín | 201800457 |

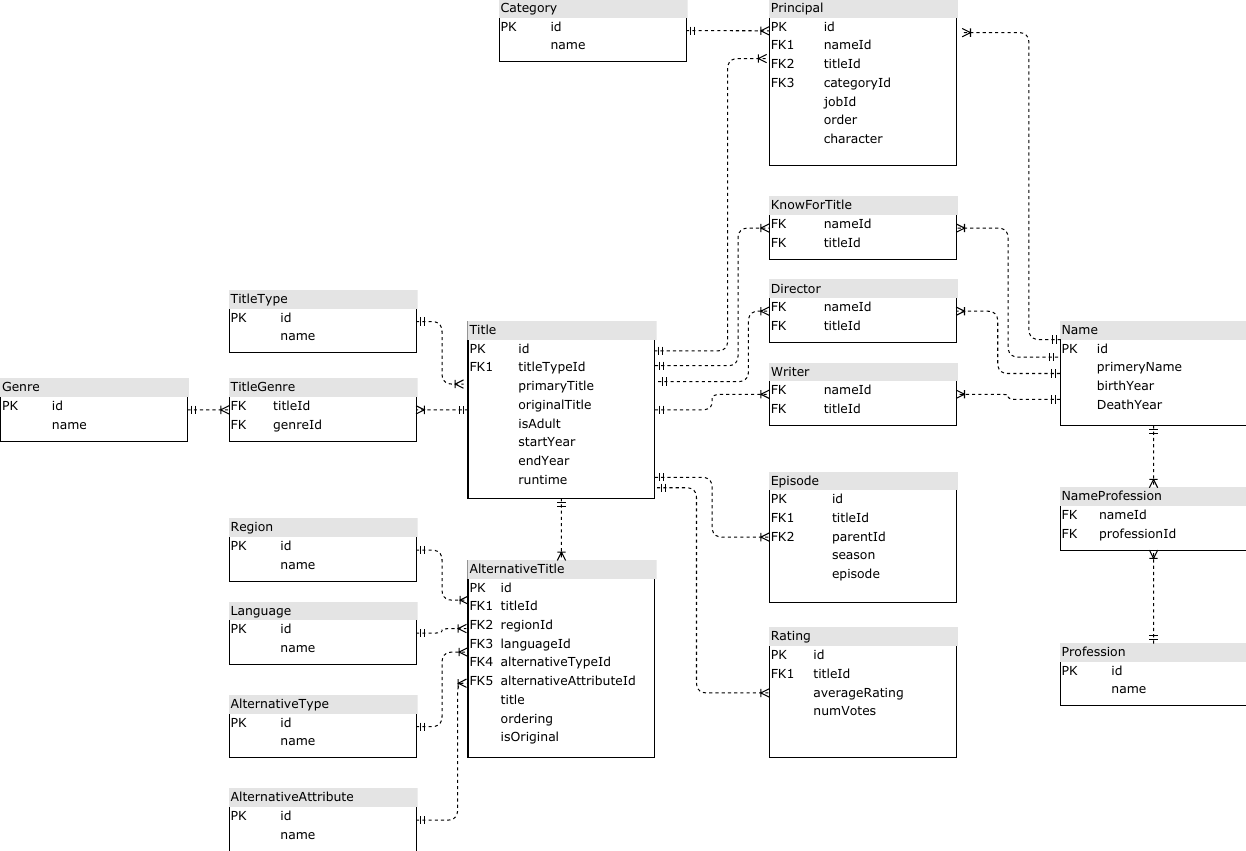
**Arquitectura del sistema**

Interfaz de usuario gráfica, Diagrama, Aplicación

Descripción generada automáticamente

**IMDB**

Esta parte del proyecto se desarrolló en una máquina virtual con el sistema operativo Linux 20.04 LTS, y la base de datos fue implementada en el motor Sql Server, en este se implementó el siguiente modelo relacional:



**Carga de datos**

Este procedimiento necesita de tablas temporales, mismas que almacenarán la información proporcionada para cargar los correspondientes registros en el modelo relacional ya implementado en la base de datos.

Para este ejemplo se realizará la carga de archivos del archivo titlebasics.tsv.

Primero creamos la tabla temporal que almacenará los registros del archivo, este debe contener la misma estructura de datos a almacenar.

Texto

Descripción generada automáticamente

Procedemos a insertar los datos utilizando la siguiente instrucción:

Texto, Carta

Descripción generada automáticamente

Realizar este procedimiento para todos los archivos .tsv a cargar en el sistema.

**Creación del modelo relacional e inserción de los datos**

Creamos el modelo relacional a utilizar para posteriormente cargar los datos utilizando la información contenida en las tablas temporales.

Para este ejemplo usaremos la tabla titletype.

Creamos la tabla con la correspondiente estructura.

Texto

Descripción generada automáticamente

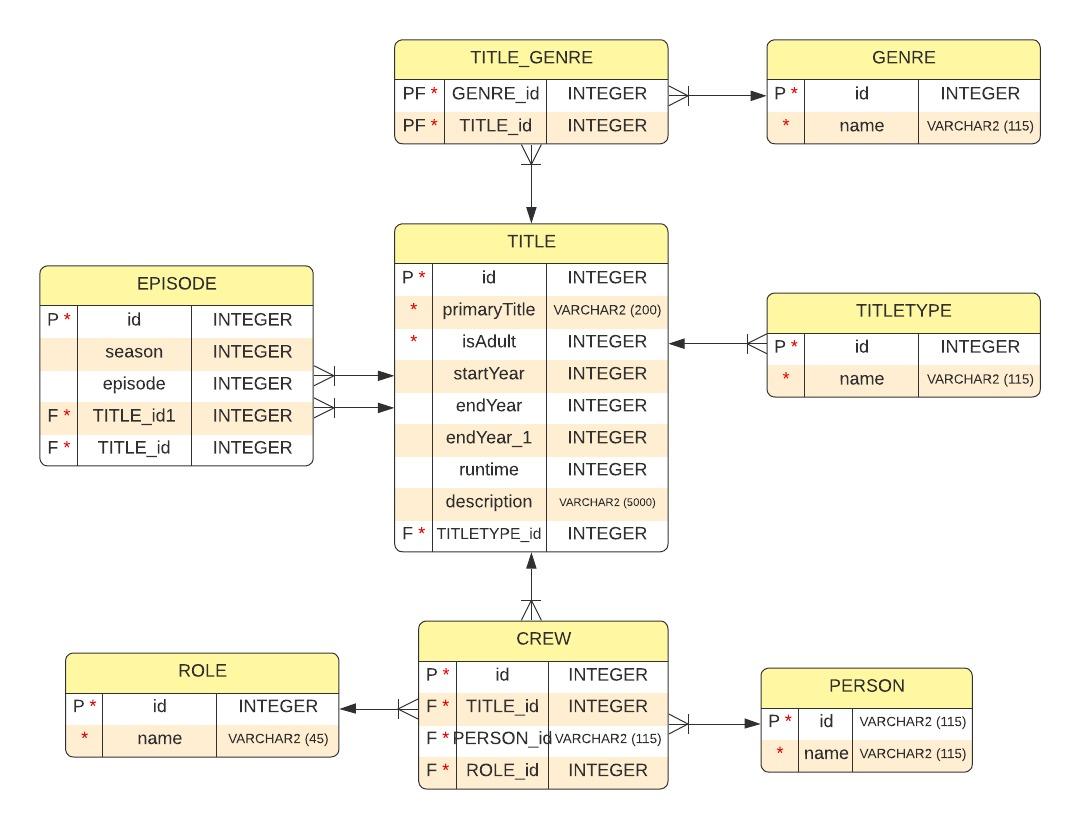
Posteriormente realizamos las correspondientes inserciones, estas se realizan incluyendo la clausula select en el insert, de esta manera se insertarán los registros obtenidos en la consulta, para esta tabla se obtiene la información de la tabla temporal titlebasics.



De esta manera quedarán insertados los datos en la nueva tabla, realizar el mismo procedimiento para todo el modelo relacional, los archivos necesarios para estos procedimientos puede encontrarlos en el siguiente [enlace](https://github.com/JorgeEspina/BD2_Proyecto2/tree/main/BD%20IMDB).

**Netflix**

Esta parte del proyecto se desarrollo en una máquina virtual con el sistema operativo Linux 20.04 LTS, y la base de datos fue implementada en el motor Sql Server, en este se implementó el siguiente modelo relacional:



Este modelo surgió en base al modelo entidad relación implementado en IMDB sin embargo se vio reducido para la base de datos de Netflix con el motivo de mantener aquí únicamente la información necesaria y así poder optimizar los datos.

**Descripción de las tablas:**

|  |  |
| --- | --- |
| **Tabla** | **Descripción** |
| GENRE | Aquí se almacenarán todos los posibles géneros que tendrá algún título. |
| TITLETYPE | Aquí almacenan los posibles tipos de un título. |
| TITLE | Aquí almacenamos la información relevante de un título, también se hace referencia a los posibles tipos de títulos y géneros que pueda clasificarse un título. El valor de esta llave primaria es igual a la de IMDB para poder vincular su rating. |
| TITLE\_GENRE | En esta tabla se almacenan todos los géneros posibles en los que pueda clasificarse un título. |
| EPISODE | En dado caso el título sea una serie, acá se encuentra un detalle de cada episodio correspondientes a un mismo título. |
| PERSON | Esta tabla almacena todas las personas que pueden verse envueltas en algún título. |
| ROLE | Acá se especifican los posibles roles que las personas pueden ejercer. Éstos pueden ser actor, actriz, director o escritor. |
| CREW | Acá se definen todas las personas y rol que ejercen en un mismo título. |

Este modelo se definió refiriéndose a algunas de las actuales directivas de Netflix, procedemos a cargar el modelo relacional a la base de datos.

**Inserción de datos a Netflix**

Para estar insertando nuevas películas en la base de datos de Netflix, se necesita un procedimiento almacenado y que la película que se vaya insertando cree un vínculo que relacione directamente la base de datos IMDB con la de Netflix, dado que ambas están en diferente servidor, se crea un Linked Server para conectar entre servidores.

**Linked Service**

Texto

Descripción generada automáticamente

Con la anterior instrucción quedan enlazadas las bases de datos, la dirección IP 32.193.226.141 que pertenece al servidor que aloja la base de datos de IMDB.

Procedemos a insertar la información de interés y estática para la base de datos de Netflix.

**Store procedure**

El llenado de las tablas TITLE, EPISODE, TITLE\_GENRE y CREW se hacen mediante el uso de un procedimiento almacenado que se encarga de realizar todas las acciones correspondientes.

Iniciamos asignándole el identificador insert\_title y asignándole los siguientes parámetros:

|  |  |  |
| --- | --- | --- |
| **Parámetro** | **Tipo** | **Descripción** |
| @titulo | VARCHAR | Contiene el nombre del título que se desea almacenar. |
| @anio | INT | Contiene el año de inicio, el año en el que se inició o estrenó el título a almacenar. |
| @tipo\_titulo | INT | Contiene el tipo de título que es (ya sea, tvSerie, show, etc) |
| @desc | VARCHAR | Contiene una breve descripción del título que se va a almacenar |
| @flag | VARCHAR | Parámetro de salida para imprimir el resultado del procedimiento almacenado |

Procedemos a crear el sp con sus parámetros.

Texto, Carta

Descripción generada automáticamente

Luego se hace uso de una variable @id\_title la cual almacena el id del título en dado caso ya esté registrado en la base de datos de IMDB.

Texto

Descripción generada automáticamente

Se hace la respectiva verificación si la variable es nula o no (es decir, si el título existe o no). Si no existe, se sale del procedimiento.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

En dado caso ya exista, se inserta en las tablas respectivas que se han mencionado antes, además se vuelve a verificar que no exista el título a agregar en la base de datos de Netflix.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Y si ya ha sido agregado antes, no se agrega nada en la base de datos.

Imagen que contiene nombre de la empresa

Descripción generada automáticamente

Todos estos scripts puede encontrarlos en el siguiente [enlace](https://github.com/JorgeEspina/BD2_Proyecto2/tree/main/BD%20NETFLIX).

**MongoDB**

Para implementar una base de datos de MongoDB en la nube, se hizo uso del servicio proporcionado por la misma compañía: Atlas; seguido se creó un clúster en el que se crearía la base de datos:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

El clúster creado lleva el nombre de **Cluster0**, la base de datos que se utilizará para las consultas será llamada **db2\_mongodb**, y la colección que se utilizará será **quickacess**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

La colección puede ser visualizada desde cloud.mongodb.com, sin embargo, si se desea utilizar con MongoDB Compass, es necesario crear un **Connection String:**

mongodb+srv://db2\_p2:8XdKTBUCq4vi3ODF@cluster0.dfzgh5g.mongodb.net/test

Este String puede variar, dependiendo de la aplicación que se desea utilizar para conectarse a la base de datos.

**Daemon**

Se implementará un Daemon con Python, el cual estará recuperando la información de la base de datos de **NETFLIX.**

La estructura del Daemon será la siguiente:

Texto

Descripción generada automáticamente

El cliente será el que contenga las conexiones a SQL Server y a través de este también se podrán ingresar títulos al catálogo de Netflix y obtener consultas generales desde MongoDB.

mongo.py será el archivo que creará el Daemon, para esto haremos uso de la librería **Daemonize**, que puede ser obtenida por medio de pip:

pip install daemonize

En este archivo también creamos una conexión a MongoDB, haciendo uso de la librería **pymongo**:

Texto

Descripción generada automáticamente

Seleccionamos la Base de datos a utilizar, y luego accedemos a la colección deseada.

Creamos una función que recupere los datos de NETFLIX desde el cliente, y le damos el formato deseado para almacenarlo en nuestra colección:

Texto

Descripción generada automáticamente

Seguido, creamos un método que será el que tome un temporizador para ejecutar la recuperación de datos cada X cantidad de segundos, finalmente, se utiliza Daemonize para iniciar el Daemon ejecutando el método anterior:

Interfaz de usuario gráfica, Texto, Sitio web

Descripción generada automáticamente

**Cliente**

Se cuenta con un sencillo cliente de consola, en él se pueden insertar datos directamente a la base de datos de NETFLIX, haciendo uso del procedimiento almacenado **insert\_title**, el cual busca información del título ingresado en la base de datos IMDB.

Texto

Descripción generada automáticamente

Se pueden buscar datos en la base de datos de MongoDB, especificando el título del proyecto, o retornar múltiples títulos basados en el rating que se desea buscar:

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

La información obtenida de estas dos opciones de búsqueda se realiza a través de MongoDB:

Texto

Descripción generada automáticamente

El Daemon utiliza consultas a NETFLIX para luego insertarlas en MongoDB: estas son unas de las funciones que se utilizan:

Texto

Descripción generada automáticamente