UNIVERSIDAD
Panamericana

Introducción a la base de datos

Mtra. Celia Marisa Rodríguez Rubio

5to semestre

Base de datos de Netflix

Victor Armando Jaramillo Moreno
María Paredes Gutiérrez de Velasco

Introducción

El objetivo de este proyecto es desarrollar una base de datos para la plataforma de streaming Netflix. La cual, al estar poblada con una cantidad considerable de datos, pueda permitir la obtención de datos relevantes tanto para el usuario como para la empresa. Ejemplos de estas consultas serían, obtener el mejor contenido en la plataforma, contenido disponible, ganancias, entre otros.

Este documento está organizado de la siguiente manera para facilitar la comprensión y el seguimiento del proyecto:

Análisis inicial: se describe la situación actual de Netflix.
Propuesta de solución: se presenta la propuesta de la base de datos, describiendo las entidades, atributos y relaciones que se tendrán.
Diseño de la BD: se muestran todos los pasos hechos para la realización del diseño de la base de datos, la E-R, matriz de relaciones, el modelo relacional, normalización y el diseño final de tablas.
Detalles encontrados al implementar la base de datos: se describen los detalles encontrados al implementar la base de datos en SQLite.
Conclusiones

Análisis inicial

Netflix fue fundada en 1997 por Reed Hastings y Marc Randolph. Netflix comenzó como un servicio de alquiler de DVD en línea, los suscriptores ordenaban películas a través de internet y después las recibían por correo. Su modelo de negocio fue evolucionando con el tiempo, a medida que la tecnología avanzaba. En 2007, Netflix lanzó su servicio de transmisión en línea, lo que permitió a los usuarios poder ver el contenido directamente a través de internet. Esto hizo que pasara de una empresa de alquiler de DVD a convertirse en una empresa líder en el sector del entretenimiento streaming.

Actualmente Netflix ofrece una amplia variedad de contenido audiovisual, películas, series y contenido original, a millones de usuarios en todo el mundo. Ha producido contenido exclusivo en una gran variedad de idiomas y géneros, lo que le ha permitido expandirse a nuevos mercados.

Su plataforma permite a los suscriptores acceder a su contenido de forma instantánea a través de internet. Una de las principales problemáticas que enfrenta Netflix y que resuelve eficientemente mediante un sistema gestor de bases de datos es la gestión masiva de contenido y la personalización de la experiencia del usuario.

Con un extenso catálogo, es esencial organizar y clasificar la información de manera efectiva para que los usuarios puedan encontrar fácilmente el contenido que les interesa.

Propuesta de solución

Para este proyecto se pretende mejorar la arquitectura y diseño de la base de datos de tal forma que sea más sencillo y eficiente realizar consultas de actualización y obtención de datos sobre el contenido de la plataforma y sus usuarios

Para esto, las entidades principales que se tendrán son Película, Serie, Episodio, Usuario, Perfil, Transacción, Restricción y Actor_Director.

En el caso de cada película se guardará su título, la duración en minutos, la calificación, una breve sinopsis, la clasificación que tiene, la fecha de lanzamiento, los géneros del que es la película, también se guardarán los doblajes disponibles al igual de en qué idiomas hay subtítulos disponibles.

La entidad de Serie sería similar a la de Película. Se guardará su título, el número de temporadas que tiene, la fecha de lanzamiento, una sinopsis, su clasificación, calificación, los géneros, y los idiomas y subtítulos en los que la serie está disponible.

La entidad Episodio guardará el nombre del episodio, la duración en minutos, el número de temporada a la que pertenece, y la calificación que tiene.

Usuario guardará el email con el que fue creada la cuenta, el nombre completo del suscriptor, el tipo y costo de su suscripción, y su domicilio.

También se tendrán los Perfiles que tiene cada usuario. Esta entidad guardará el nombre del perfil y la edad del usuario de ese perfil, ya que habrá contenido al que no tendrán acceso dependiendo de la edad.

La entidad Transacción será por cada pago que hace el usuario de su suscripción. Se guardará el monto, la fecha de la transacción y el banco y tipo de tarjeta con el que se paga.

Restricción, en este caso se guardará el país. Esto va a servir para saber qué contenido no está disponible en qué países. Ya que el catálogo de Netflix puede variar de país en país.

Actor_Director, en esta entidad se guardará el nombre, se guardará si está persona se refiere a un actor o director, su edad, de que ciudad y país es la persona, y los premios que ha ganado.

Ahora se describirán las relaciones que estás entidades van a tener.

Cada usuario puede hacer varias transacciones para pagar su suscripción, pero cada transacción solo puede ser hecha por un usuario.

Cada cuenta del usuario puede tener varios perfiles, pero cada perfil puede pertenecer a un solo usuario (entiéndase aquí a usuario como la cuenta de Netflix).

Cada perfil puede ver varias películas y series, y a su vez cada película y serie puede ser vista por muchos perfiles.

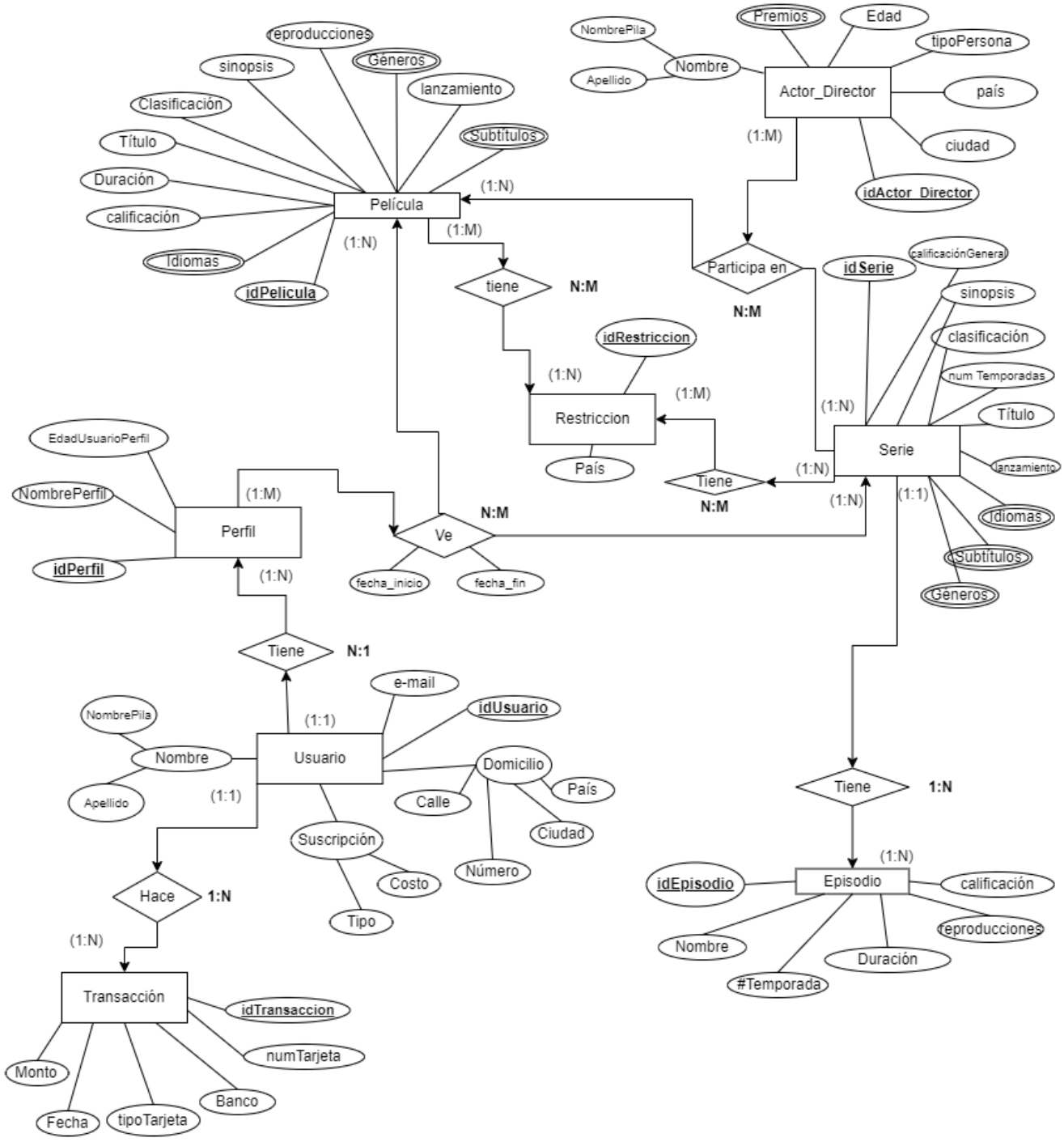
Cada serie puede tener muchos episodios, pero cada episodio solo puede pertenecer a una serie.

Cada película y serie pueden tener varias restricciones, en este caso a restricción nos referimos a en qué país no está disponible la película o serie. y a su vez una restricción de cierto país puede aplicar a varias series y películas.

Cada Actor_Director puede participar en varias películas y series, y en cada película y serie pueden participar varios Actor_Director.

Diseño de la BD

E-R



Matriz de relaciones

	Usuario	Perfil	Transacción	Película	Serie	Episodio	Restriccion	Actor_Director
Usuario	-	Tiene 1:N	Hace 1: N	-	-	-	-	-
Perfil	es de 1:1	-	-	Ve 1:N	Ve 1:N	-	-	-
Transacción	La hace 1:1	-	-	-	-	-	-	-
Película	-	Es vista 1:N	-	-	-	-	Tiene 1:N	Tiene 1:N
Serie	-	Es vista 1:N	-	-	-	Tiene 1:N	Tiene 1:N	Tiene 1:N
Episodio	-	-	-	-	Es parte de 1:1	-	-	-
Restricción	-	-	-	Restringe 1:N	Restringe 1:N	-	-	-
Actor_Director	-	-	-	Aparece/ Dirige 1:N	Aparece/ Dirige 1:N	-	-	-

Modelo Relacional antes de la normalización

Usuario(idUsuario, email, nombre(nombrePila,Apellido), domicilio(calle,número,ciudad,país),suscripción(tipo,costo)

Perfil(idPerfil, idUsuario (FK), NombrePerfil,EdadUsuarioPerfil)

Transaccion(idTransaccion,idUsuario (FK), monto, fecha, banco,tipoTarjeta, numTarjeta)

Película(idPelicula, título, duración, lanzamiento, calificación, clasificación, sinopsis, reproducciones, géneros, idiomas, subtítulos)

Serie(idSerie, título, numTemporadas, lanzamiento, calificaciónGeneral, sinopsis, clasificación, géneros, idiomas, subtítulos)

Episodio(idEpisodio, idSerie (FK), Nombre, #Temporada, Duración,calificación, reproducciones)

Restricción(idRestricción, idContenido (FK), tipoContenido, restriccionPaís)

> En este caso, se especificará el tipo de contenido serie o película, y junto con el id proporcionado, se sabrá si se debe referenciar la tabla de Película o Serie.

Actor_Director(idActor_Director, nombre(NombrePila,Apellido), tipoPersona, edad, ciudad, país, premios)

Perfil_ve_Contenido(idPerfil, idContenido (FK), tipoContenido, visualizacionMin, vecesVisto)

Actor_Director_Participa(idActor_Director, idContenido, tipoContenido)

Normalización

Dependencias funcionales

- Usuario(idUsuario, email, nombre(nombrePila,Apellido), domicilio(calle,número,ciudad,país),suscripción(tipo,costo))
    - {idUsuario, idSuscripcion} -> email, nombre(nombrePila,Apellido), domicilio(calle,número, ciudad, país), suscripción(tipo,costo)
- Perfil(idPerfil, idUsuario (FK), NombrePerfil,EdadUsuarioPerfil)
    - {idPerfil,idUsuario} -> NombrePerfil, EdadUsuarioPerfil
- Transaccion(idTransaccion,idUsuario (FK), monto, fecha, banco, tipoTarjeta, numTarjeta)
    - {idTransaccion, idUsuario} -> monto, fecha, banco, tipoTarjeta, numTarjeta
- Película(idPelicula, título, duración, lanzamiento, calificación, clasificación, sinopsis, reproducciones, géneros, idiomas, subtítulos)
    - {idPelicula} -> , título, duración, lanzamiento, calificación, clasificación, sinopsis, reproducciones, géneros, idiomas, subtítulos
- Serie(idSerie, título, numTemporadas, lanzamiento, calificaciónGeneral, sinopsis, clasificación, géneros, idiomas, subtítulos)
    - {idSerie} -> título, numTemporadas, lanzamiento, calificaciónGeneral, sinopsis, clasificación, géneros, idiomas, subtítulos
- Episodio(idEpisodio, idSerie (FK), Nombre, #Temporada, Duración,calificación, reproducciones)
    - {idEpisodio, idSerie} -> Nombre, #Temporada, Duración, calificación, reproducciones
- Restricción(idRestricción, idContenido (FK), tipoContenido, restriccionPaís)
    - {idRestricción,idContenido,t tipoContenido} -> País
- Actor_Director(idActor_Director, nombre(NombrePila,Apellido), tipoPersona, edad, ciudad, país, premios)
    - {idActor_Director} -> nombre(NombrePila,Apellido), tipoPersona, edad, ciudad, país premios
- Perfil_ve_Contenido(idPerfil, idContenido (FK), tipoContenido, visualizacionMin, vecesVisto)
    - {idPerfil, idContenido,tipoContenido} -> visualizacionMin, vecesVisto
- Actor_Director_Participa(idActor_Director, idContenido, tipoContenido, protagonista, director)
    - {idActor_Director, idContenido, tipoContenido} -> protagonista, director

Formas Normales

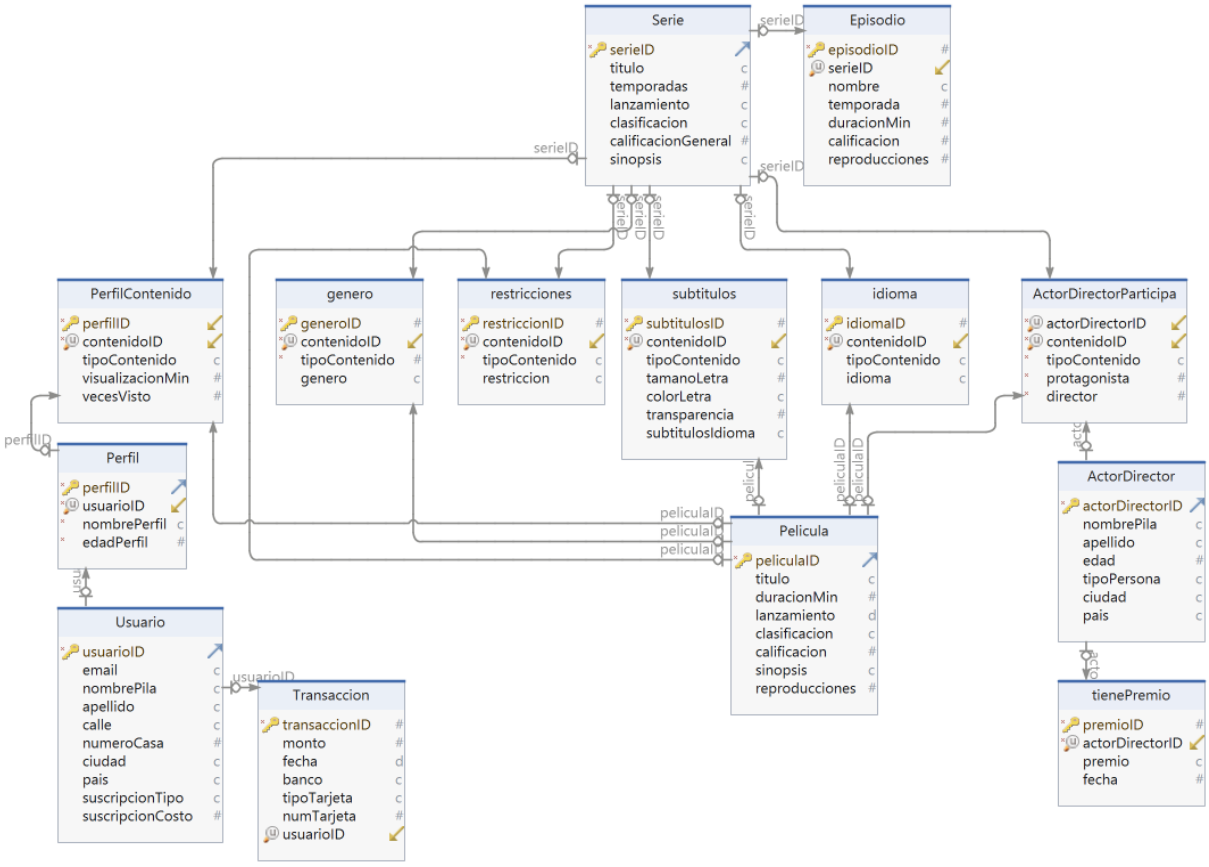
- Usuario(idUsuario, email, nombre(nombrePila,Apellido), domicilio(calle,número,ciudad,país),suscripción(tipo,costo))
    - F1: no tiene campos multivaluados, pero tiene tres campos compuestos
        - Usuario(idUsuario, email, nombrePila,Apellido,calle,número,ciudad,país,suscripciónTipo, suscripcionCosto)
    - F2: los campos no principales dependen de las claves -> Sí cumple
    - F3: No hay dependencias transitivas -> Sí cumple
- Perfil(idPerfil, idUsuario (FK), NombrePerfil,EdadUsuarioPerfil)
    - F1: no tiene campos multivaluados ni compuetos -> Sí cumple
    - F2: los campos no principales dependen de las claves -> Sí cumple
    - F3: No hay dependencias transitivas -> Sí cumple
- Transaccion(idTransaccion,idUsuario (FK), monto, fecha, banco,tipoTarjeta, numTarjeta)
    - F1: no tiene campos multivaluados ni compuestos -> Sí cumple
    - F2: los campos no principales dependen de las claves -> Sí cumple
    - F3: No hay dependencias transitivas -> Sí cumple
- Película(idPelicula, título, duración, lanzamiento, calificación, clasificación, sinopsis, reproducciones, géneros, idiomas, subtítulos)
    - F1: tiene campos multivaluados, pero no tiene campos compuestos
        - Película(idPelicula, título, duración, lanzamiento, calificación, clasificación, sinopsis, reproducciones)
        - Géneros(idGenero, idContenido (FK), tipoContenido, genero)
        - Idioma(idIdioma, idContenido (FK), tipoContenido, idioma)
        - Subtítulos(idSubtitulos, idContenido (FK), tipoContenido, subtituloIdioma, tamañoLetra, colorLetra, transparencia)
    - F2: los campos no principales dependen de las claves -> Sí cumple
    - F3: no hay dependencias transitivas -> sí cumple
- Serie(idSerie, título, numTemporadas, lanzamiento, calificaciónGeneral, sinopsis, clasificación, géneros, idiomas, subtítulos)
    - F1: tiene campos multivaluados, pero no tiene campos compuestos
        - Serie(idSerie, título, numTemporadas, lanzamiento, calificaciónGeneral, sinopsis, clasificación)
        - Para los atributos géneros, idiomas, subtítulos se utilizarán las tablas Géneros, Idioma, Subtítulos. En este caso con la llave foránea {idContenido, tipoContenido} se sabrá si se debe referir a la tabla de Película o Serie.
    - F2: los campos no principales dependen de las claves -> Sí cumple
    - F3: no hay dependencias transitivas -> sí cumple

- Episodio(idEpisodio, idSerie (FK), Nombre, #Temporada, Duración, calificación, reproducciones)
    - F1: no tiene campos multivaluados ni campos compuestos -> Sí cumple
    - F2: los campos no principales dependen de las claves -> Sí cumple
    - F3: no hay dependencias transitivas -> Sí cumple
- Restricción(idRestricción,idContenido (FK), tipoContenido, restriccionPaís)
    - F1: no tiene campos multivaluados ni campos compuestos -> Sí cumple
    - F2: los campos no principales dependen de las claves -> Sí cumple
    - F3: no hay dependencias transitivas -> Sí cumple
- Actor_Director(idActor_Director, nombre(NombrePila,Apellido), tipoPersona, edad, ciudad, país, premios)
    - F1: tiene un campo multivaluado y un campo compuesto
        - Actor_Director(idActor_Director, NombrePila,Apellido, tipoPersona, edad, ciudad, país)
        - tienePremio(idPremio,idActor_Director, premio, fecha)
    - F2: los campos no principales dependen de las claves -> Sí cumple
    - F3: no hay dependencias transitivas -> Sí cumple
- Perfil_ve_Contenido(idPerfil, idContenido (FK), tipoContenido, visualizacionMin, vecesVisto)
    - F1: no tiene campos multivaluados ni campos compuestos -> Sí cumple
    - F2: los campos no principales dependen de las claves -> Sí cumple
    - F3: no hay dependencias transitivas -> Sí cumple
- Actor_Director_Participa(idActor_Director, idContenido, tipoContenido, protagonista, director)
    - F1: no tiene campos multivaluados ni campos compuestos -> Sí cumple
    - F2: los campos no principales dependen de las claves -> Sí cumple
    - F3: no hay dependencias transitivas -> Sí cumple

Modelo relacional después de la normalización

- Usuario(idUsuario, email, nombrePila,Apellido,calle,número,ciudad,país,suscripciónTipo,suscripcionCosto)
- Perfil(idPerfil, idUsuario (FK), NombrePerfil,EdadUsuarioPerfil)
- Transaccion(idTransaccion,idUsuario (FK), monto, fecha, banco,tipoTarjeta, numTarjeta)
- Película(idPelicula, título, duración, lanzamiento, calificación, clasificación, sinopsis, reproducciones)
- Serie(idSerie, título, numTemporadas, lanzamiento, calificaciónGeneral, sinopsis, clasificación)
- Actor_Director(idActor_Director, NombrePila,Apellido, tipoPersona, edad, ciudad, país)
- Género(idGenero, idContenido (FK), tipoContenido, genero)
- tienePremio(idPremio,idActor_Director (FK), premio, fecha)
- Subtítulos(idSubtitulos, idContenido (FK), tipoContenido, subtituloIdioma, tamañoLetra, colorLetra, transparencia)
- Idioma(idIdioma, , idContenido (FK), tipoContenido, idioma)
- Episodio(idEpisodio, idSerie (FK), Nombre, #Temporada, Duración, calificación, reproducciones)
- Restricción(idRestricción, idContenido (FK), tipoContenido, restriccionPaís)
- Perfil_ve_Contenido(idPerfil, idContenido (FK), tipoContenido, visualizacionMin, vecesVisto)
- Actor_Director_Participa(idActor_Director, idContenido, tipoContenido, protagonista, director)

Tablas



Detalles encontrados al implementar la base de datos

Creación de las tablas en la base de datos

Realizar el modelo E-R correctamente facilita la creación de las tablas en la base de datos, ya que se identifican rápidamente las llaves primarias y únicas. Así como las llaves foráneas y los campos que no pueden ser null.

Llaves foráneas

Las tablas de idioma,género,subtítulos,restricciones,perfilContenido y actorDirectorParticipa todas hacen referencia a las tablas Película y Serie. Como estas últimas 2 son tablas independientes, pueden compartir ID, puesto que película con ID 1 es diferente a la Serie con ID 1.

Una complicación respecto a este modelo es que no se pueden definir correctamente los campos de contenidoID en el diseño de la base de datos en el gestor de SQL puesto que un solo campo solo puede referenciar a una sola tabla.

Aunque para definir si el contenidoID se refiere a una Película o Serie se utiliza el campo tipoContenido que contiene algún valor entre ‘Pelicula’ o ‘Serie’, aun así fue imposible asignar las llaves foráneas, sin embargo, se tratan a estos campos como llaves foráneas en toda la implementación de la base de datos y en todas las consultas.

Creación de datos para alimentar la base de datos

El objetivo de este proyecto es realizar secuencias complejas que funcionen en poco tiempo bajo cantidades masivas de datos, es por esta razón que necesitamos una forma rápida y automática de generar datos “correctos” en cantidad. Para realizar esto usamos diversas librerías y módulos de python como random,datetime y faker, librerías especializadas en generar variables aleatorias, fechas y datos falsos.

Una complicación encontrada es que los datos debían tener relación, un ejemplo es con los premios obtenidos por actores o director, pues es ilógico que un actor de 21 años actualmente haya recibido un premio en 1989. Los códigos utilizados para generar estos datos de la manera más precisa y con sentido posible los puede encontrar en la carpeta de archivos auxiliares.

Otra situación fue generar las transacciones para cada usuario, porque normalmente las transacciones se hacen cada mes (30 días) por lo que se debían generar fechas aleatorias con una diferencia de 30 teniendo en cuenta que todas las fechas sean anteriores a la fecha de creación de la cuenta.

Además, todos los nombres y apellidos se obtuvieron de listas de datos que se pueden obtener desde el siguiente repositorio https://github.com/VictorAJM/lemarios

Copias de seguridad

Siempre es una buena práctica mantener copias de seguridad de los archivos necesarios o utilizados en la base de datos, ya que al realizar actualizaciones o consultas muy grandes incorrectas, esto puede causar errores y pérdida de información que sin tener una copia de seguridad puede ser perjudicial.

Conclusiones

Realizar este proyecto nos permitió aplicar todo el conocimiento que hemos adquirido a lo largo del semestre. Podemos dividir el proyecto en 3 partes, diseño de la base de datos, creación de la base de datos, y pruebas de la base de datos.

Durante el diseño de la base de datos fue fundamental aplicar y dominar los conocimientos sobre los modelos E-R, ya que estos son los pilares sobre los cuales vamos a crear las tablas y dentro de estas las llaves primarias como las foráneas. De igual forma, notamos la importancia de las formas normales, ya que más adelante nos ayudarían a crear una base de datos en donde priorizamos la eficiencia e integridad de los datos.

Durante la creación de la base de datos, fue necesario identificar todos los campos y sus respectivos tipos. Aunque se nos pide un mínimo de 20 registros por tablas, decidimos ingresar un mayor número de registros para tener una base de datos lo suficientemente poblada para poder identificar y mejorar las consultas realizadas. En base a esto, creamos un total de 70221 registros divididos entre las 14 tablas creadas.

ActorDirector	ActorDirectorParticipa	Episodio	Genero	Idioma	Pelicula	Perfil
100	12430	3587	4990	6643	1577	1432

PerfilContenido	Restricciones	Serie	Subtitulos	tienePremio	Transaccion	Usuario
17212	1918	303	6677	247	12508	597

TOTAL=	70221

Finalmente, cada secuencia implementada responde a algún problema, algunas de ellas como descuentos en los planes de membresía, o mejoras en los tipos de membresía para los clientes con al menos 1 año de antigüedad. Para resolver estas problemáticas fue necesario expandir y complementar el conocimiento adquirido en clase, ya que fue necesario utilizar funciones agregadas no vistas en clases o en las que no profundizamos lo suficiente, como por ejemplo cross join, CAST, los operadores ||, round, limit, entre otros.