

**UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA**

**Alumna:**

Lady Lilibeth Puchaicela Calva

**Carrera:**

Ciencias de la Computación

**Docente:**

Nelson Piedra

**Ciclo:**

Tercer Ciclo

**Tema:**

Catedra Integradora

**Introducción**

Hoy en día la base de datos forma parte fundamental en el entorno informático y tecnológico debido a que su aplicación es practica en su totalidad de sus roles en cualquier campo y en toda disciplina o área donde exista la necesidad de gestionar datos, ya sean estos datos mínimos o voluminosos.

El presente informe se basa en el tipo de base de datos relacional debido a que es el más utilizado en la actualidad, el cual permite o da solución a problemas de base de datos utilizando un esquema basado en tablas, por medio del que nos resulta sencillo y fácil de comprender y utilizar para cualquier análisis o consulta de datos.

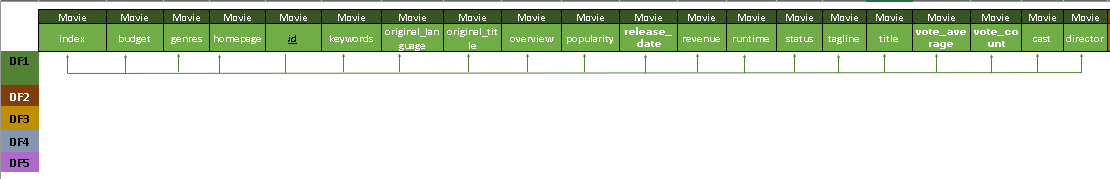
Se desarrolla el estudio previo de la tabla general del cual se procede a identificar cada dependencia funcional existente, y la determinación de cada campo, luego de esto se implementa la normalización en la tabla general siguiendo las tres primeras formas normales con el fin de eliminar redundancia en la información de los datos, optimizar almacenamiento del ordenador y eliminar dependencias funcionales que sean incoherentes, así para llegar a un único resultado, y proceder luego con el armado del esquema entidad – relación y modelo relacional.

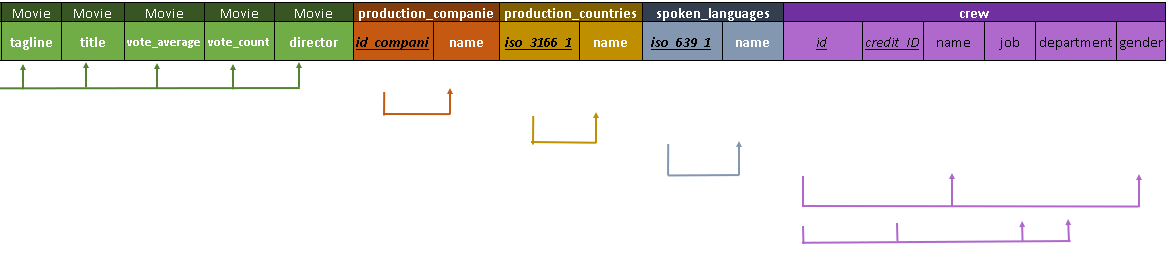
Una vez determinado cada campo, tabla, relación, normalización total es conveniente desarrollar la importación del csv a MySQL en el gestor de base de datos en el que se está trabajando (DataGrip), y proseguir con la creación del DDL, DML y carga de datos.

***Desarrollo***

Desde el archivo csv se encontró diferentes campos de los cuales se pudo identificar sus campos, sus atributos primos y no primos, así como también la dependencia funcional de cada tabla, definiendo los campos que dependen de otros, para luego normalizar las tablas tal como se muestran a continuación:

# Dependencia Funcional

******

******

Como se observa en el grafico expuesto, se muestran todas las dependencias funcionales de la tabla general con todos los campos, seguidamente se describe de una forma mas detallada cada dependencia funcional.

* **IdMovie** 🡪 {keyword, index, homepage, genres, budgets, original\_languaje, original\_title, overview, popularity, director, vote\_count, vote\_average, tagline, status, cast}
* **Id\_company** 🡪 name
* **Iso\_639\_1** 🡪 name
* **Iso\_3166\_1** 🡪 name
* **Credit\_id** 🡪 job, department
* **Id\_Crew** 🡪 name, gender
* **Id, Credit\_Id** → Name, Gender, Job, Department

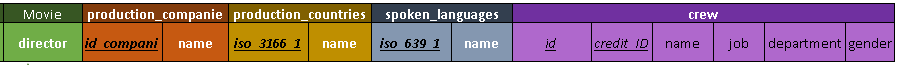
# **Normalización**

La normalización es un proceso el cual permite organizar los datos, lo que incluye el análisis y creación de tablas, así como establecer sus relaciones entre las mismas, con el fin de eliminar redundancia, dependencias incoherentes y optimizar el uso de memoria del ordenador.

Para realizar la normalización de la tabla movie, se siguió las siguientes formas:

## Primera forma normal

En esta forma se analiza toda la tabla, con el fin de identificar que sus campos sean atómicos, es decir en esta tabla no debe existir campos que sean compuestos o multivaluados.

Con el análisis de cada atributo de la figura anterior se ha encontrado los siguientes campos tales como:

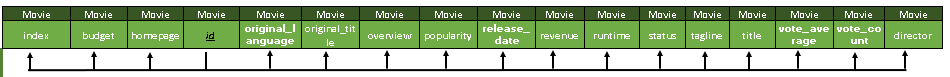
**Multivaluados:**

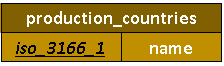
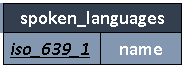
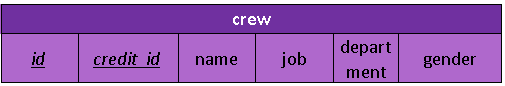
Genres, keywords, y cast.

**Multivaluados y compuestos:**

Production\_companies, production\_countries, spoken\_languages y crew.

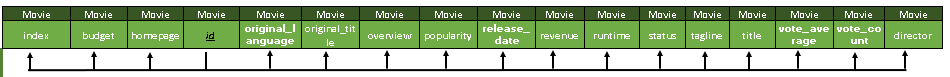
Por lo cual estos campos son separados en otras tablas para cumplir con esta forma normal, posteriormente se muestra gráficamente la normalización de esta primera forma.

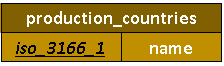
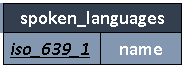
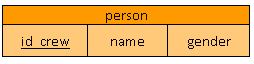


## SEGUNDA forma normal

Una vez que la tabla cumpla la primera forma normal se analiza si alguna de las tablas ya obtenidas tiene dependencia compuesta, por lo contrario, se ha observado que algunas tablas luego de cumplir con la 1FN también cumplen la 2FN en excepción de la tabla crew la cual tiene dependencias parciales donde separadas quedaría que; el id de la tabla person determina el name y gender, por otro lado credit\_id determina job, y department y tiene como clave foránea el idPerson, por consiguiente se tiene la tabla de relación que esta compuesta por el id de la tabla movio y id de la tabla credit.



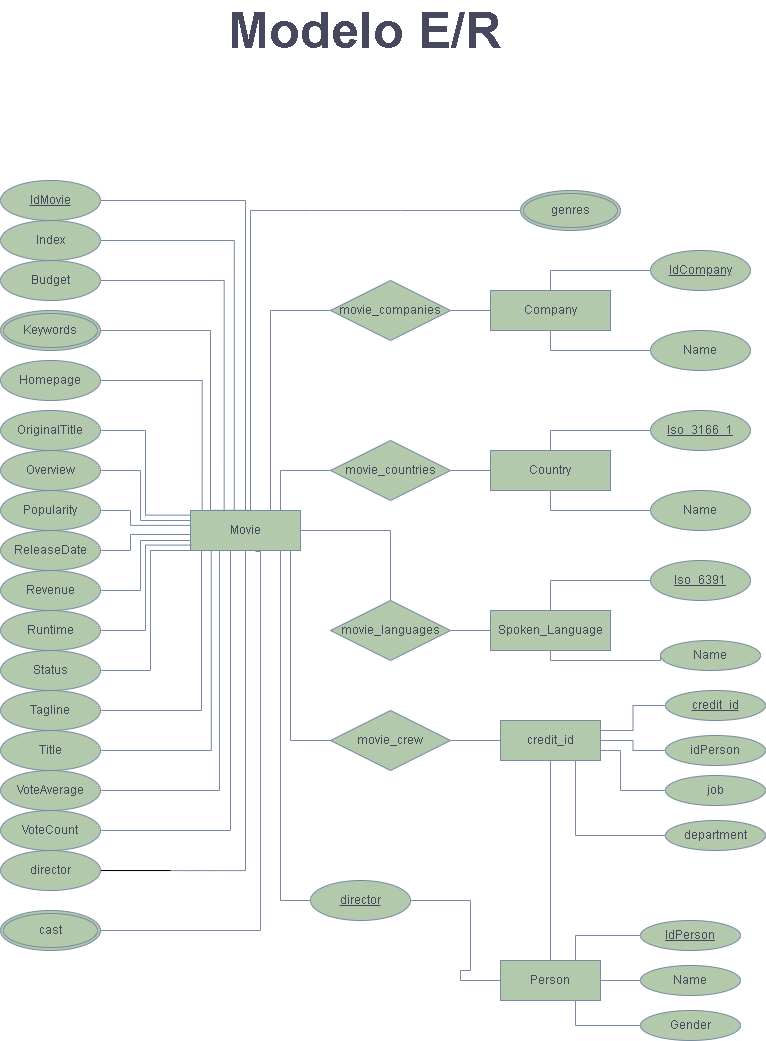
## Tercera forma normal

Luego de determinar las dos primeras formas normales, se procede con la 3FN en la cual se determina si existe alguna dependencia transitiva en las tablas antes obtenidas, sin embargo, con la realización de estas tablas ya se cumplen con las 3 formas normas

***Creación del Esquema Entidad-Relación***

Una vez identificado las entidades, atributos y aplicado la normalización de los campos que se encuentran en el CSV, se procede a diseñar el esquema (E/R) que podemos apreciar en la Figura

# Esquema entidad-relación

Luego de analizar las entidades se prosigue con la creación del esquema donde se define la cardinalidad entre cada tabla. 

# Modelado desde el Enfoque Relacional

Para facilitar la comprensión el realizar las tablas donde se indican las entidades, campos o atributos, así como también el definir de cada entidad o tabla son las claves candidatas y cuáles serían las claves primarias, por consiguiente, reconocer si los atributos de cada entidad son multivaluados y de dominio son.

En la tabla 1 se muestra a la entidad “movie”, la cual es la tabla principal o central de nuestra base de datos, de donde se parte para encontrar la relación, cardinalidad, normalización, que se encuentra entre todas las entidades

Tabla 1

*Modelo de la tabla movie*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Property | Dominio | Optional | MultiValuated | Type | Comment |
| Movie | Director | Varchar (30) | Not null | False |  | Director de la película |
| Movie | Overview | Varchar (500) |  | False |  | Información de la película |
| Movie | Index | Int (5) | Not null | False |  | indice |
| Movie | Id | Int (11) | Not null | False | Primary key | Identificador de la película |
| Movie | Homepage | Varchar (150) |  | False |  | Página principal |
| Movie | Revenue | Int (10) | Not null | False |  | Ingresos |
| Movie | Title | Varchar (200) | Not null | False |  | Título de la película |
| Movie | RunTime | Double (5,5) | Not null | False |  | Tiempo de ejecución |
| Movie | Budget | Int (10) | Not null | False |  | Presupuesto para la película |
|  |  |  |  |  |  |  |
| Movie | Original\_Languages | Varchar (2) | Not null | False |  | Lenguaje original de la película |
| Movie | popularity | Double (6,5) | Not null | False |  | Popularidad de la película |
| Movie | Vote\_count | Int (10) | Not null | False |  | Contador de votos |
| Movie | Vote\_Average | Double (2,3) | Not null | False |  | Promedio de votos |
| Movie | Original\_title | Varchar (150) | Not null | False |  | Titulo original |
| Movie | Tagline | Varchar (100) |  | False |  | Lema de la película |
| Movie | status | Varchar (8) | Not null | False |  | Estado |
| Movie | Release\_date | Date | Not null | False |  | Fecha de realización |
| Movie | director | Varchar(100) | Not null | False | Foreign key | Nombre del director |

En la tabla 2,3,4,5 se muestra las entidades con sus atributos correspondientes, las cuales tienen una relación con la entidad movie, siendo su cardinalidad de muchos a muchos (N : M)

Tabla 2

*Modelo de la tabla production\_companies*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Property | Dominio | Optional | Multivaluated | Type | Comment |
| production\_companies | Id\_Company | Int (6) | Not null | False | Primary key | Identificador de la compañía de producción |
| production\_companies | Name | Varchar (50) | Not null | False |  | Nombre de la compañía de producción |

Tabla 3

*Modelo de la tabla movie\_companies*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Property | Dominio | Optional | Multivaluated | Type | Comment |
| movie\_companies | Id\_Company | Int (6) | Not null | False | Primary key | Identificador de la compañía de producción |
| Movie | Id | Int (11) | Not null | False | Primary key | Identificador de la película |

Tabla 4

*Modelo de la tabla production\_countries*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Property | Dominio | Optional | MultiValuated | Type | Comment |
| production\_countries | Iso\_3166\_1 | Varchar (2) | Not null | False | Primary key | Identificador de la ciudad de producción |
| production\_countries | name | Varchar (30) | Not null | False |  | Nombre de la ciudad de producción |

Tabla 5

*Modelo de la tabla movie\_countries*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Property | Dominio | Optional | MultiValuated | Type | Comment |
| production\_countries | Iso\_3166\_1 | Varchar (2) | Not null | False | Primary key | Identificador de la ciudad de producción |
| Movie | Id | Int (11) | Not null | False | Primary key | Identificador de la película |

Tabla 6

*Modelo de la tabla spoken languages*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Property | Dominio | Optional | MultiValuated | Type | Comment |
| Spoken\_Languages | Iso\_639\_1 | Varchar (2) | Not null | False | Primary key | Identificador de lenguaje |
| Spoken\_Languages | name | Varchar (12) | Not null | False |  | Nombre del lenguaje |

Tabla 7

*Modelo de la tabla languages*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Property | Dominio | Optional | MultiValuated | Type | Comment |
| Spoken\_Languages | Iso\_639\_1 | Varchar (2) | Not null | False | Primary key | Identificador de lenguaje |
| Movie | Id | Int (11) | Not null | False | Primary key | Identificador de la película |

Tabla 8

*Modelo de la tabla crew*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Property | Dominio | Optional | Multivaluated | Type | Comment |
| Crew | idCrew | Int(5) | Not null | False | Primary key | Identificador de tripulación |
| Crew | Credit\_ID | Varchar (45) | Not null | False | Primary key | Identificador de crédito de tripulación |
| Crew | Name | Varchar (25) | Not null | False |  | Nombre de la tripulación |
| crew | gender | Varchar (1) | Not null | False |  | Genero de la tripulación |
| crew | department | Varchar (20) | Not null | False |  | Departamento de cada tripulación |
| crew | job | Varchar (30) | Not null | False |  | Trabajo de cada tripulación |

A continuación, se muestra la entidad department, la cual tiene una relación con la entidad crew, siendo su cardinalidad de muchos a 1 (M: 1).

Tabla 9

*Modelo de la tabla person*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Property | Dominio | Optional | Multivaluated | Type | Comment |
| Crew | idPerson | Int(5) | Not null | False | Primary key | Identificador de tripulación |
| Crew | Name | Varchar (25) | Not null | False |  | Nombre de la tripulación |
| crew | gender | Varchar (1) | Not null | False |  | Genero de la tripulación |

Tabla 10

*Modelo de la tabla credit*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Property | Dominio | Optional | Multivaluated | Type | Comment |
| Crew | idPerson | Int(5) | Not null | False | Foreign key | Identificador de tripulación |
| Crew | department | Varchar (20) | Not null | False |  | Departamento de cada tripulación |
| Crew | job | Varchar (30) | Not null | False |  | Trabajo de cada tripulación |
| Crew | Credit\_ID | Varchar (45) | Not null | False | Primary key | Identificador de crédito de tripulación |

Tabla 11

*Modelo de la tabla movie\_crew*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Property | Dominio | Optional | Multivaluated | Type | Comment |
| Crew | Credit\_id | Varchar (45) | Not null | False | Primary key | Identificador de crédito de tripulación |
| Movie | Id | Int (11) | Not null | False | Primary key | Identificador de la película |

Tabla 12

*Modelo de la tabla movie\_genres*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Movie | genre | Varchar (150) | Not null | True |  | Genero de la película |
| Movie | Id | Int (11) | Not null | False | Primary key | Identificador de la película |

Tabla 13

*Modelo de la tabla cast*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Property | Dominio | Optional | Multivaluated | Type | Comment |
| Cast | cas\_id | Varchar (50) | Not null | False | Primary key | Identificador de actor |
| Cast | actor | Varchar (100) | Not null | False |  | Nombre de actor |

Tabla 14

*Modelo de la tabla movie\_cast*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Property | Dominio | Optional | Multivaluated | Type | Comment |
| Cast | cas\_id | Varchar (50) | Not null | False | Primary key | Identificador de actor |
| Movie | IdMovie | Int (11) | Not null | False | Primary key | Identificador de la película |

Tabla 15

*Modelo de la tabla keywords*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Property | Dominio | Optional | Multivaluated | Type | Comment |
| keywords | keyword\_id | Varchar (50) | Not null | False | Primary key | Identificador de actor |
| keywords | word | Varchar (100) | Not null | False |  | Palabra clave |

Tabla 16

*Modelo de la tabla movie\_keywords*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Property | Dominio | Optional | Multivaluated | Type | Comment |
| keywords | keyword\_id | Varchar (50) | Not null | False | Primary key | Identificador de actor |
| Movie | IdMovie | Int (11) | Not null | False | Primary key | Identificador de la película |

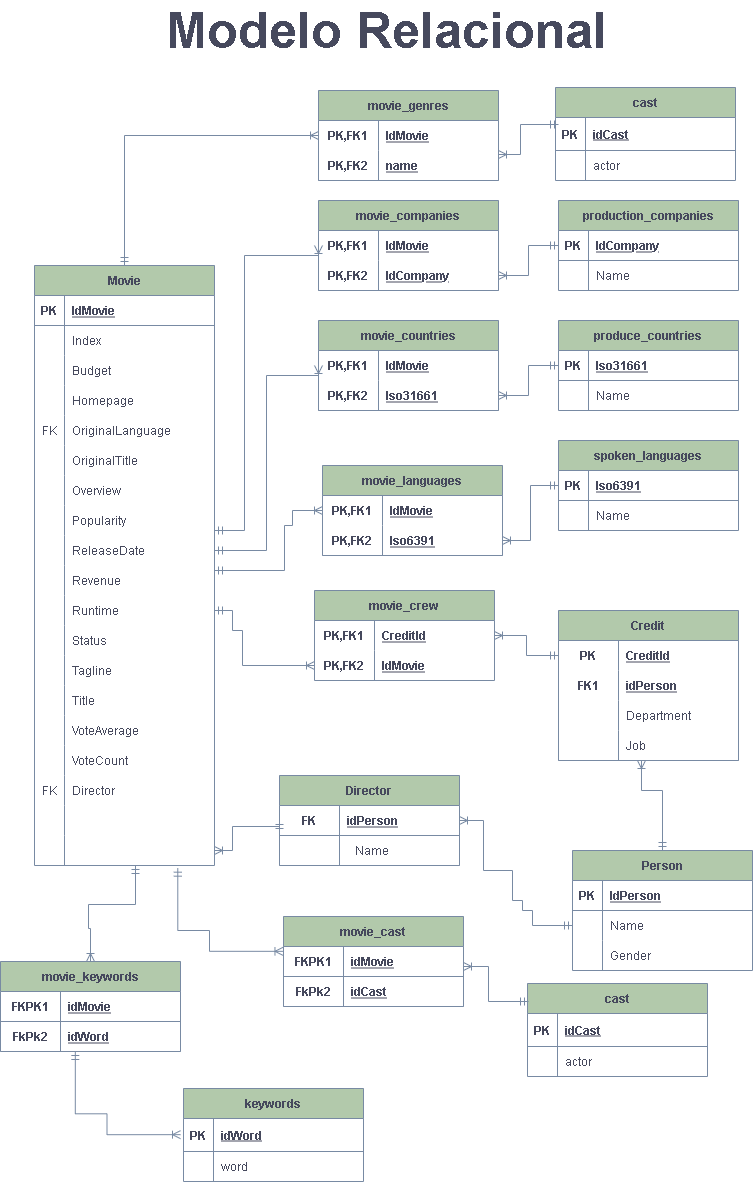
# Creación del Esquema Relacional

Detallando la relación que une las entidades del modelado, como por ejemplo la tabla 2 production\_company donde una película (movie) es producidas por muchas compañías y una compañía puede producir muchas películas, así mismo con las demás tablas.

Cada tabla posee los atributos expuestos que representan valores que se desea almacenar en cada entidad. Los campos marcados con “PK” indican aquellos que son claves primarias, es decir que se identifican de manera única a cada entidad, el cual será un valor de tipo entero. Los campos marcados como “FK” son claves foráneas o claves externas, las cuales indican campos que van a almacenar claves primarias de otras tablas de modo que se puedan relacionar con la tabla actual.

Todo el diseño conceptual de la base de datos para manejar la información que contiene el archivo CSV y tomando de referencia el modelado de este, se lo plasma en el esquema relacional como se muestra en la siguiente figura.

# Esquema relacional

**

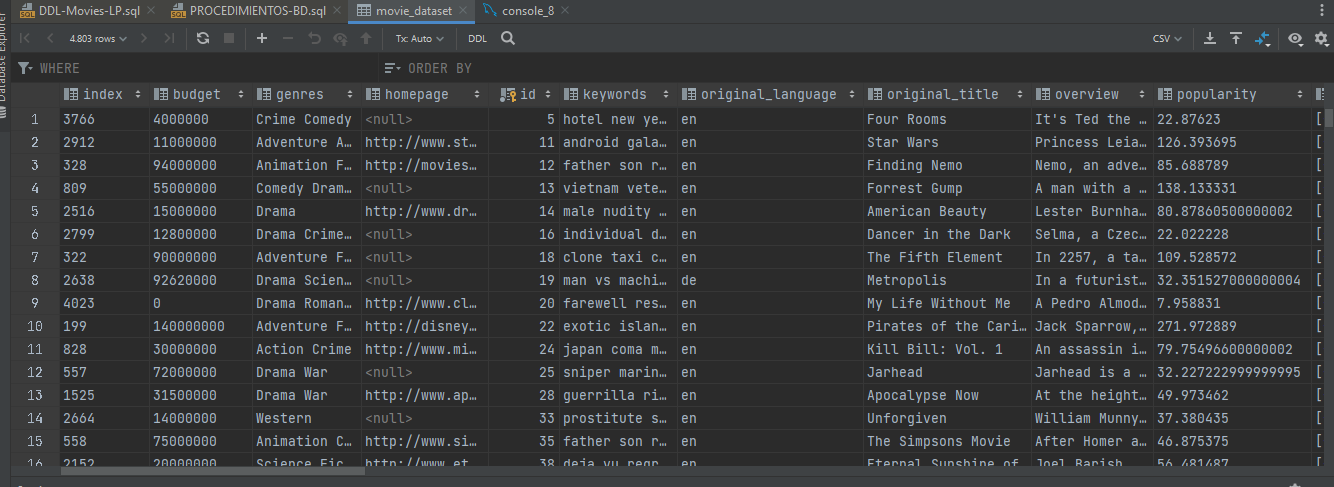
*Nota.* Una vez que tenemos claro el modelo E-R ya se puede traducir a un modelo lógico directamente en el propio sistema gestor de bases de datos.

# Crear database

Una vez realizado lo antes descrito, el primer paso es crear la base de datos y seleccionarla para así trabajar en ella empezando con la creación de las tablas.



Luego de haber creado la base de datos se realiza la importación del CSV a MySQL,



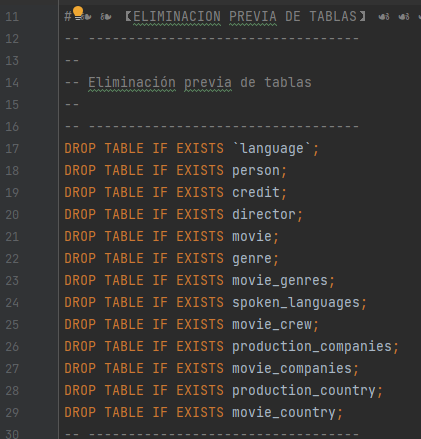
Conforme se analiza la data se observa que hay datos que mediante la importación no se han importado completamente, de tal forma que se realizo inserción manual de esos datos, y al final se puede hacer uso y trabajar con el CSV completo, tomando en cuenta todos los argumentos y acciones necesarias en cual fuera el caso.

# Creación de tablas DDL

En la creación de las tablas se especificarán cada campo con su tipo de valor correspondiente, en estos campos se guardará toda una serie de datos que son necesarios para rellenar cada tabla con la data propuesta, que será desglosada del CSV de lo cual se detallará más adelante, conforme sea necesario y siguiendo el esquema y modela antes definido.

Una vez definido el nombre de los campos y sus tipos de datos, se define también cuáles serán sus claves primarias en cada tabla, de igual manera se describe cuáles serán sus claves foráneas, las cuales nos permitirán crear la relación existente en cada una de estas tablas.

## Eliminaciòn previa de las tablas

******

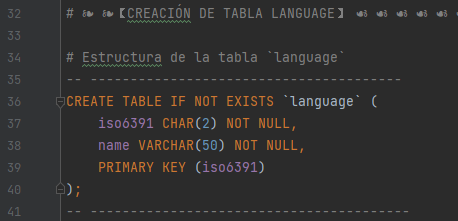
Previo a la creación de cada tabla, se realiza la eliminación de cada una para evitar un error en tablas, si es el caso que alguna de ellas exista.

En cada tabla se hace uso de sintaxis tales como:

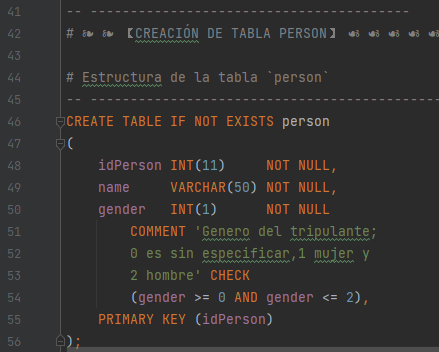
* DROP TABLE: borra una o más tablas.
* CREATE TABLE: crea una tabla con el nombre dado, y se define cada campo con el tipo de valor ya sea, int, varchar, decimal, bigint,etc. Dependiendo de esto se especifica si la columna debe ser NULL o NO NULL, conforme a los datos que se obtienen.
* ALTER TABLE: permite cambiar la estructura de una tabla ya existente, en este caso se puede añadir o borrar columnas, crear o destruir Índices, o a su vez cambiar o renombrar el tipo de columnas existentes en la misma tabla.
* CONSTRAINT: se utiliza en las instrucciones ALTER TABLE y CREATE TABLE para crear o eliminar restricciones en varios campos.

De esta manera se realiza la creación con cada una de las tablas que se encuentran expuestas a continuación;

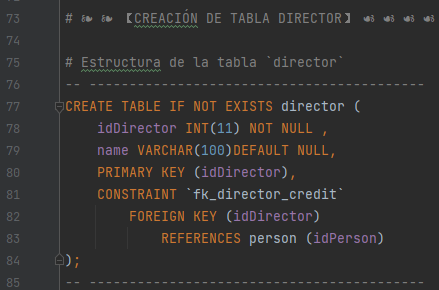
## tabla language

******

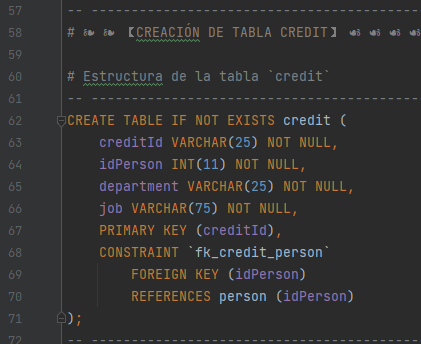
## tabla person

******

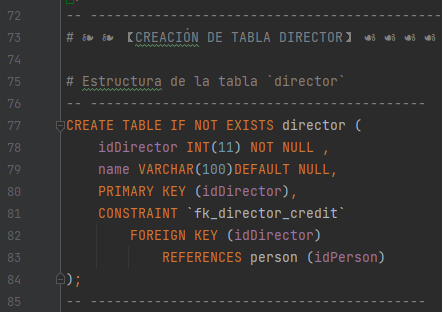
## TABLA DIRECTOR

******

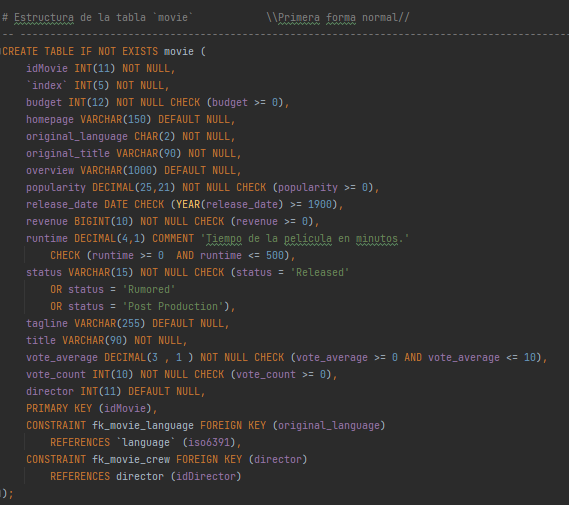
## tabla credit

******

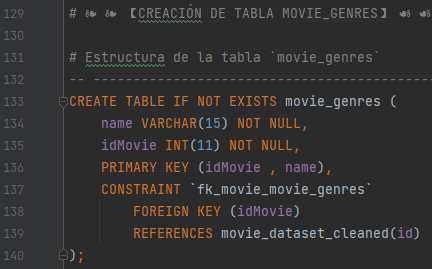
## tabla director

******

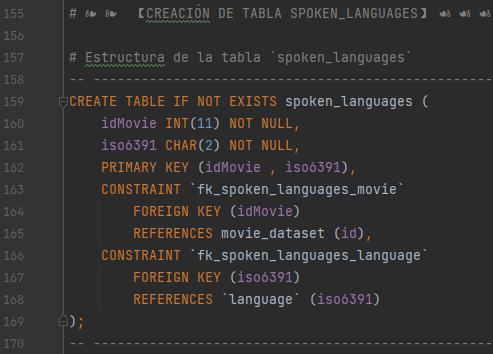
## tabla movie



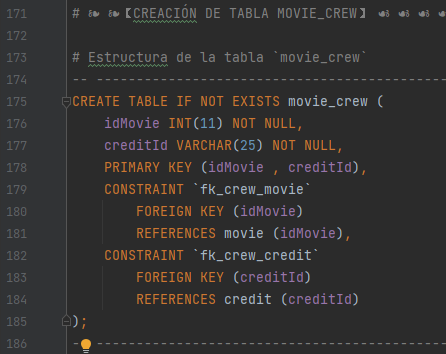
## tabla moviE\_genres

******

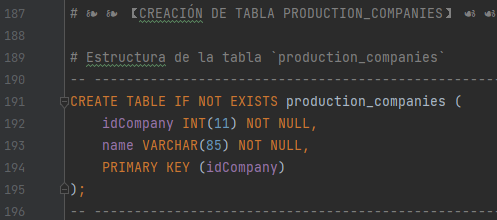
## Tabla SPOKEN\_LANGUAGE

******

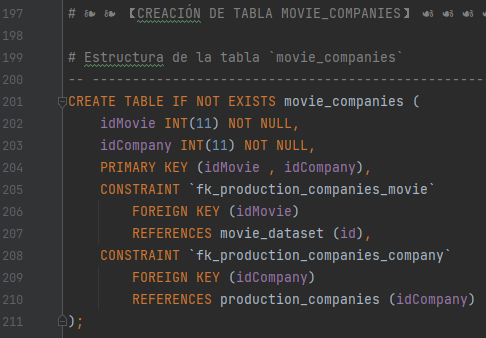
## Tabla movie\_crew

******

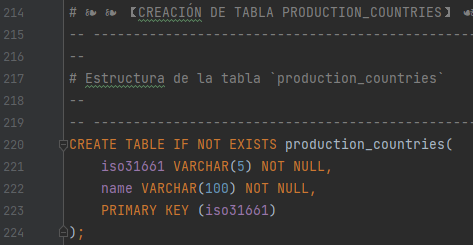
## Tabla production\_company

******

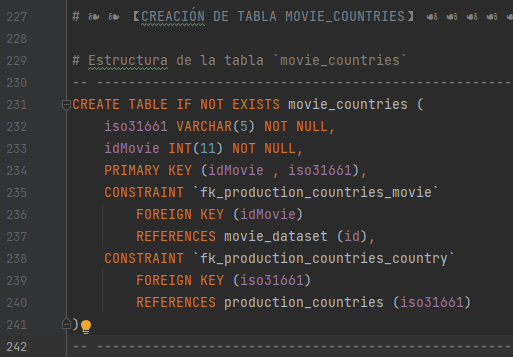
## Tabla movie\_companies

******

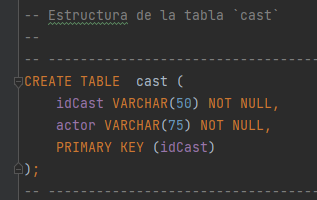
## Tabla production\_countries

******

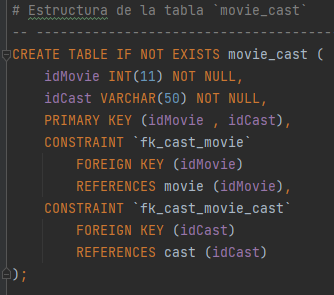
## Tabla movie\_countries

******

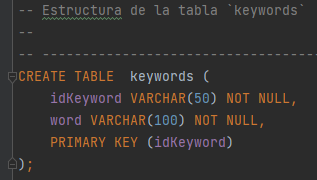
## Tabla CAST

******

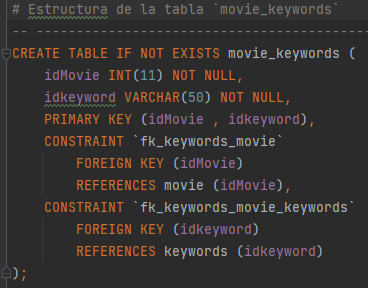
## Tabla movie\_CAST

******

## Tabla KEYWORDS

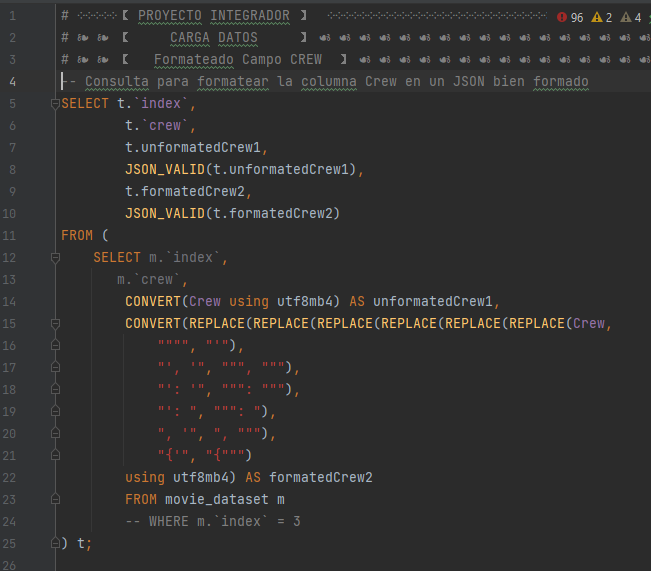
******

## Tabla MOVIE\_KEYWORDS

******

# DML

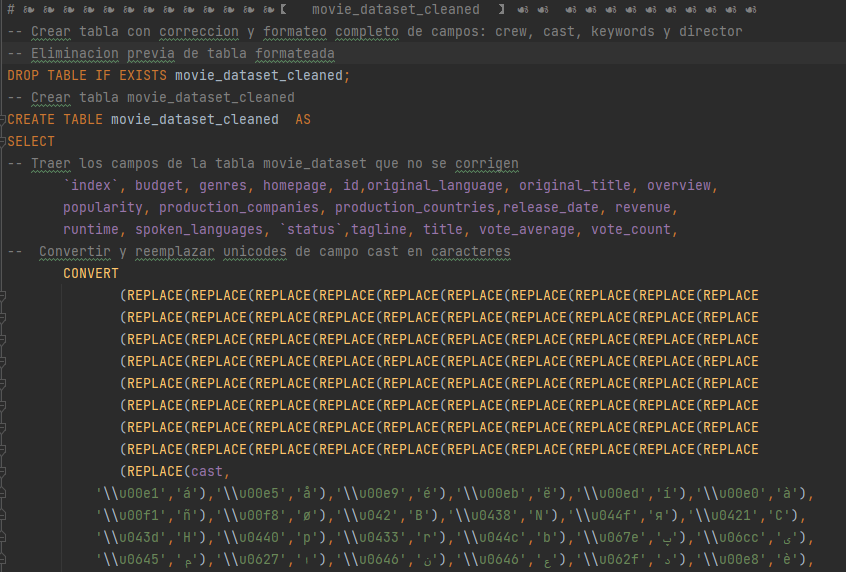
## CORREXIÓN CAMPO CREW

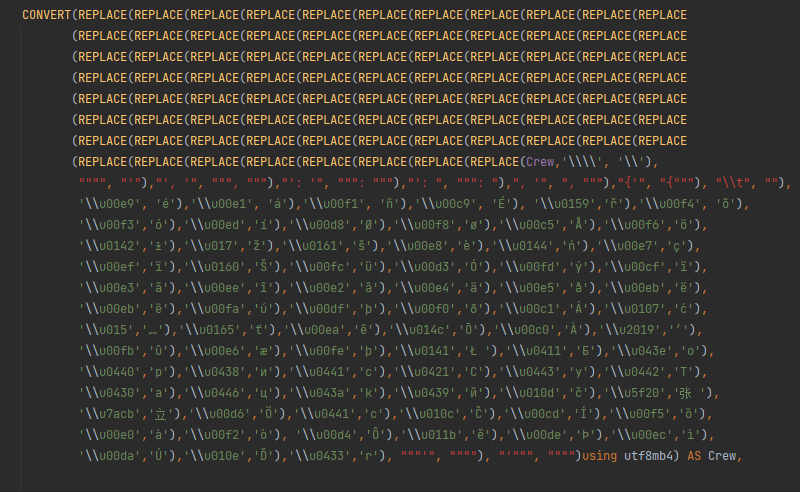
******

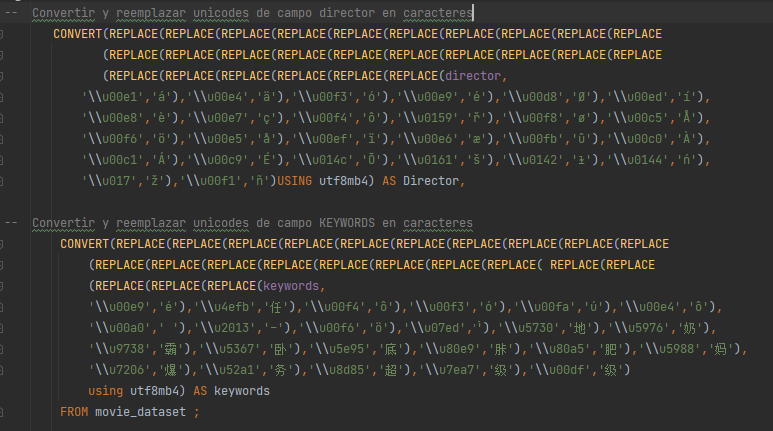
# FORMATEADO COMPLETO CAMPOS (CAST, CREW, Y DIRECTOR)

Debido al análisis realizado en la data, se observo los errores que contenían la misma, desde los campos incompletos al momento de la importación del CSV, luego de delimitar cada inconveniente se obtuvo que los campos, cast, keywords, director y crew era necesario realizar con el uso de funciones como convert y replace; comillas, simples, dobles, códigos de caracteres especiales como la diéresis (¨), tildes (`,´), letras etc.

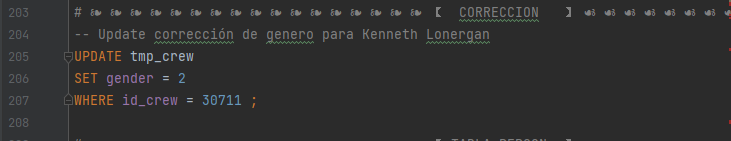
Procediendo se crea una tabla en la cual llamaremos los campos que están de forma correcta e iremos añadiendo los campos con sus debidas correcciones.







## CORRECIÓN DE GENERO KENNETH LONERGAN

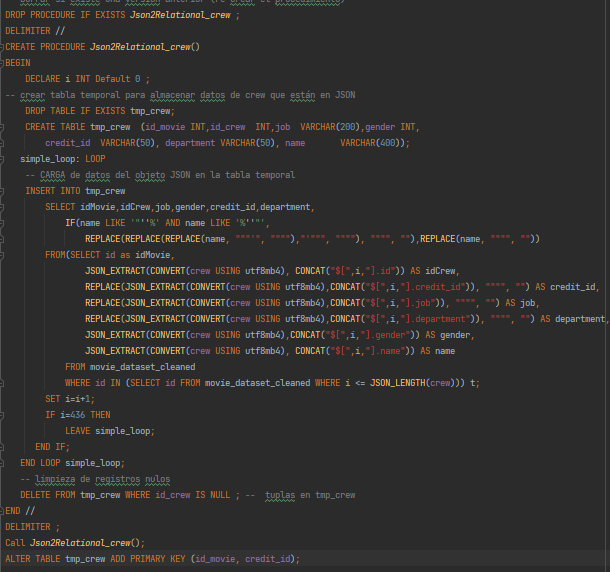
******

# Procedimientos

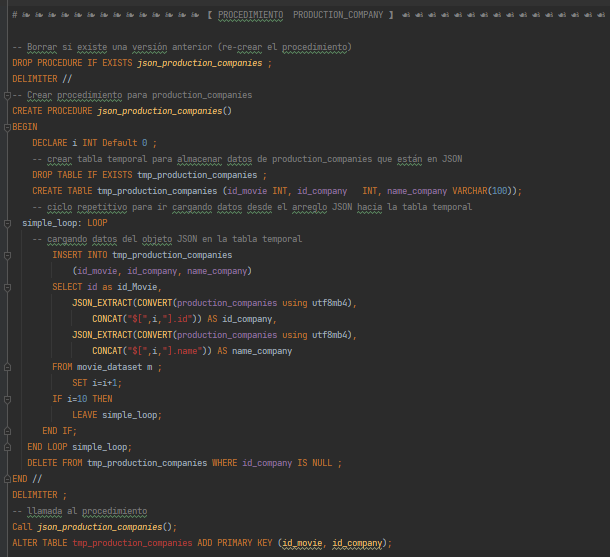
Luego de crear las tablas oficiales con sus respectivos atributos y su relación correspondiente, para realizar la carga de datos en cada una de estas tablas se necesito sacar los datos de la tabla original con los 24 campos, de esta manera para realizar cada acción se procedió a crear procedimientos para cada campo multivaluado y compuesto, de tal manera que para production\_companies, production\_countries,spoken\_languages,casr, crew y keywords.Tales que luego de estos procedimientos en los cuales mediante un ciclo se recorre cada dato y se extrae conforme a la condición que se describa dentro del mismo, por consiguiente se hace un llamado a cada procedimiento para la carga de datos a cada tabla temporal creada en el mismo.

Estas tablas temporales contienen los datos con los que procederemos a llenar cada tabla original antes creadas, cada una con sus respectivos datos. Cabe recalcar que en el campo cast y keyword se analizó, debido a que es un campo multivaluado lo que se delimita a separar cada dato contenido por fila en tokens dependiendo de su respectiva sintaxis.

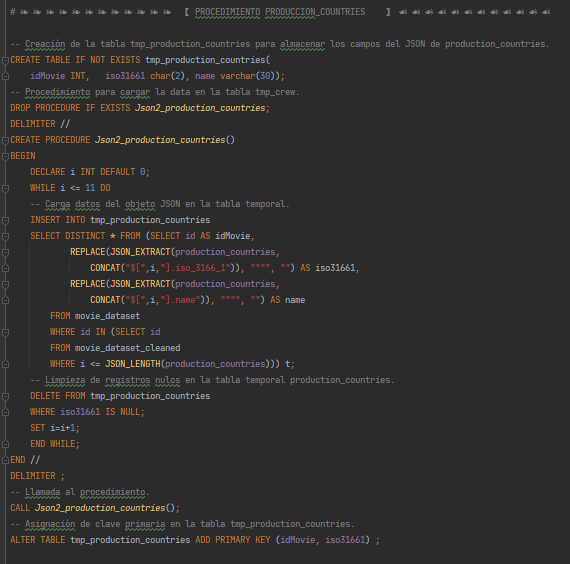
## PROCEDIMIENTO PARA CREAR TABLA TEMPORAL CREW

******

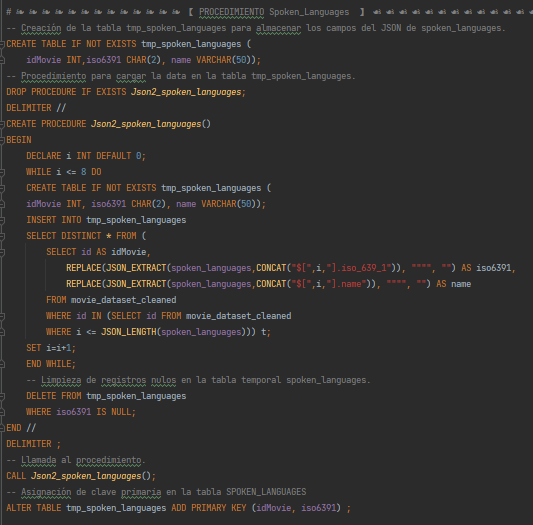
## PROCEDIMIENTO PRODUCTION\_COMPANIES



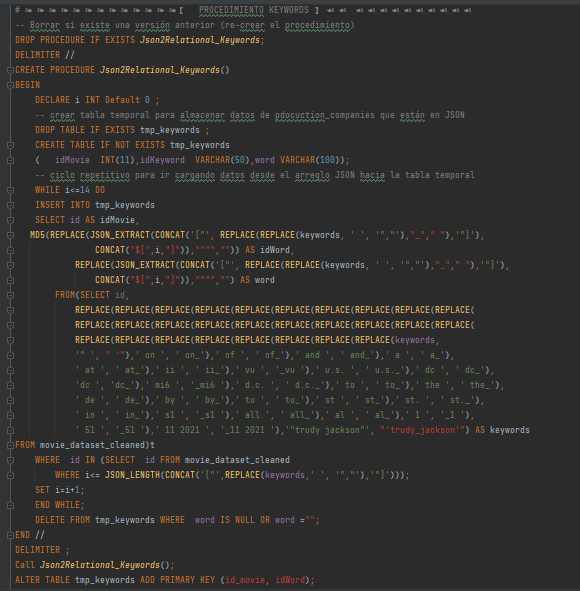
## PROCEDIMIENTO PRODUCTION\_COUNTRIES



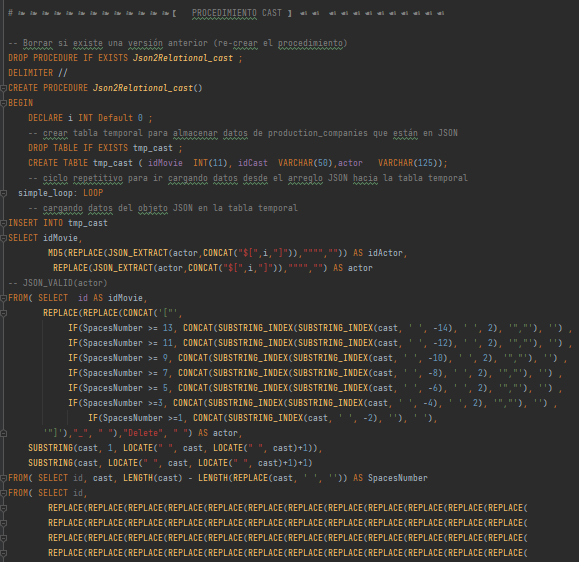
## PROCEDIMIENTO SPOKEN\_LANGUAGES

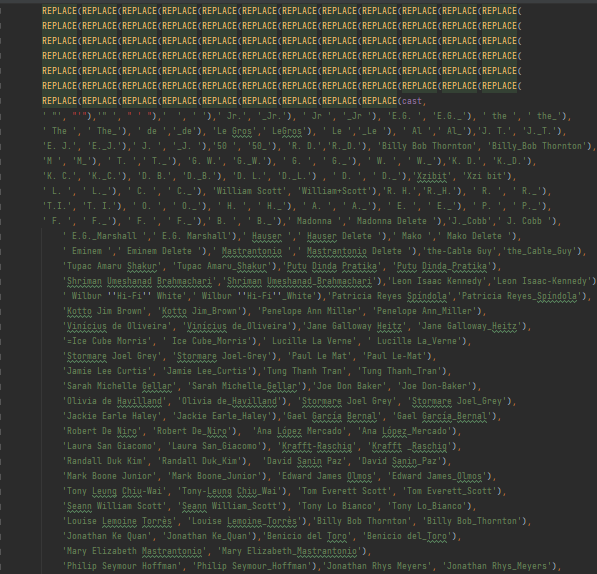
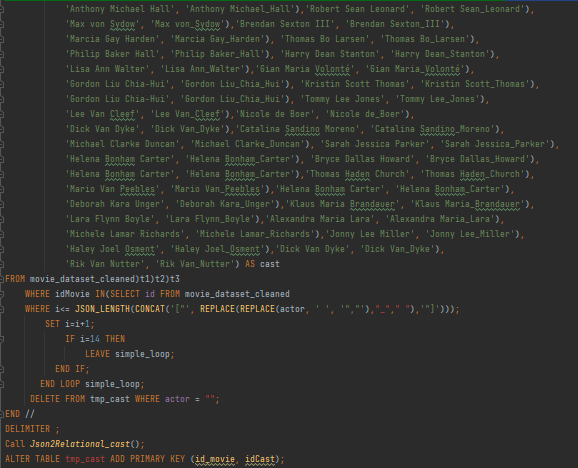


## PROCEDIMIENTO KEYWORDS



## PROCEDIMIENTO CAST

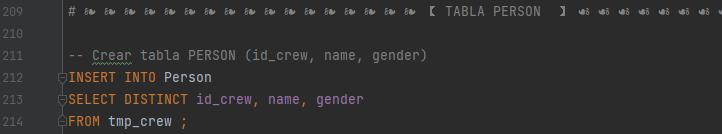




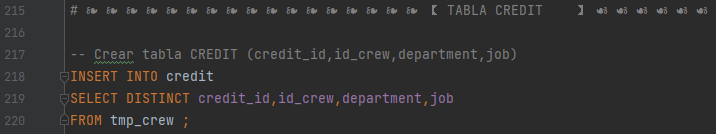
# Carga de datos a tablas originales

De la forma antes mencionada para la carga de datos se realiza mediante sentencias tales como insert into nombre de la tabla a la que se va a realizar la inserción de datos, Select donde selecciona los campos que se desea traer desde la tabla temporal con un From.

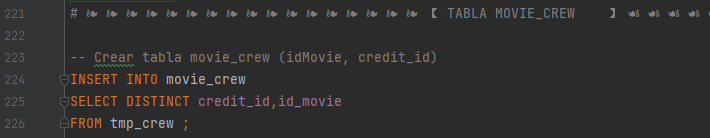
## TABLA PERSON

******

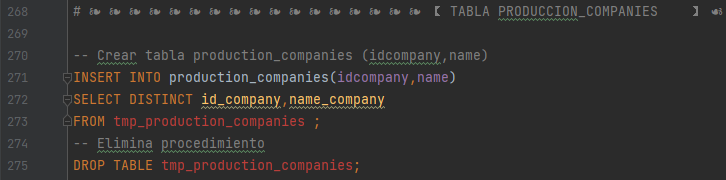
## TABLA CREDIT

******

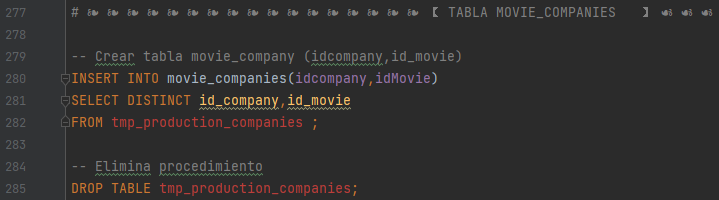
## TABLA MOVIE\_CREW

******

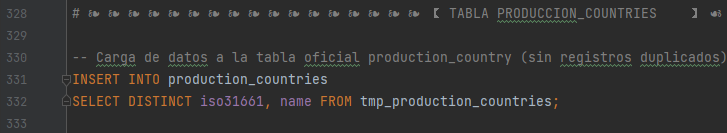
## CARGA DE DATOS A TABLA MOVIE\_COMPANIES

******

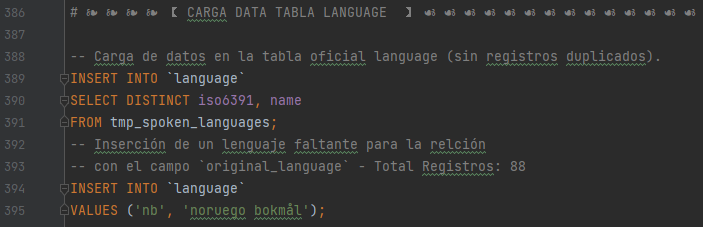
## CARGA DE DATOS TABLA MOVIE\_COMPANIES

******

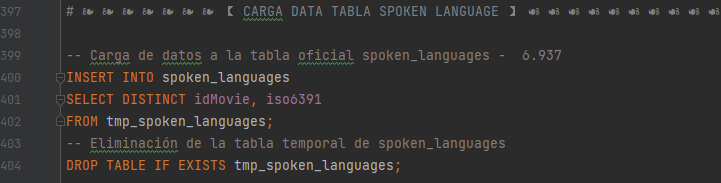
## CARGA DE DATOS A TABLA PRODUCTION\_COUNTRIES

******

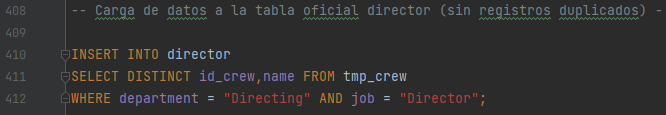
## CARGA DE DATOS EN TABLA LANGUAGE

******

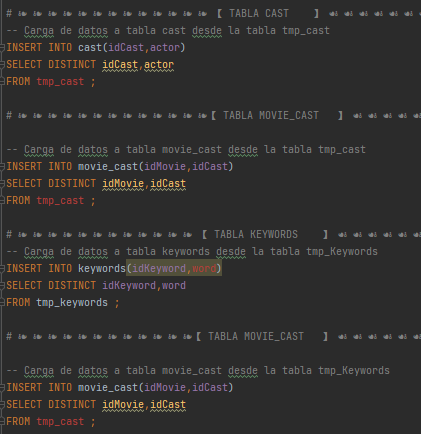
## CARGA DE DATOS TABLA SPOKEN LANGUAGE

******

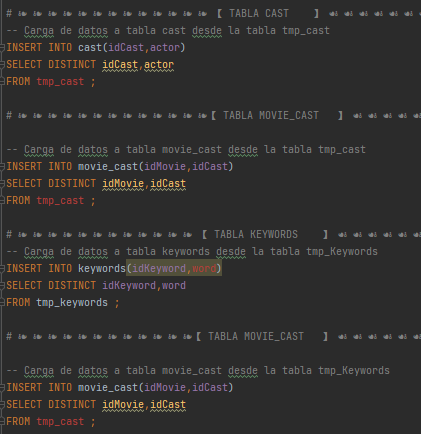
## CARGA DE DATOS TABLA DIRECTOR

******

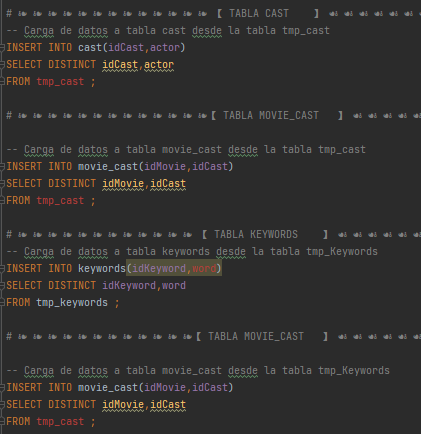
## CARGA DE DATOS TABLA CAST

******

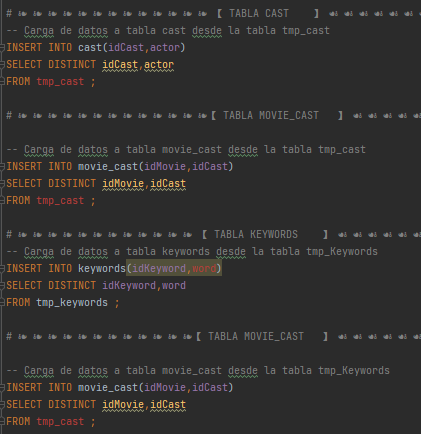
## CARGA DE DATOS TABLA MOVIE\_CAST

******

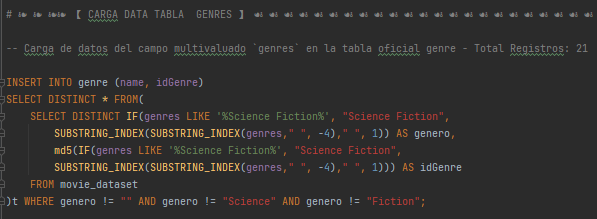
## CARGA DE DATOS TABLA KEYWORDS

******

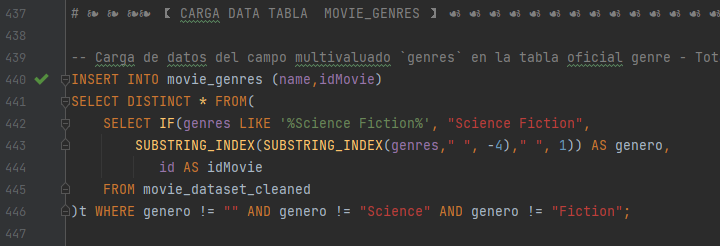
## CARGA DE DATOS TABLA MOVIE\_KEYWORDS

******

## CARGA DE DATOS TABLA GENRES

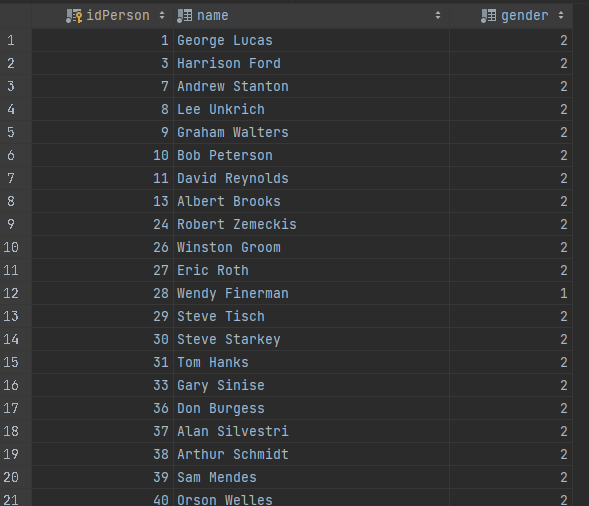
******

## CARGA DE DATOS TABLA MOVIE\_GENRES

******

# Población de tablas originales

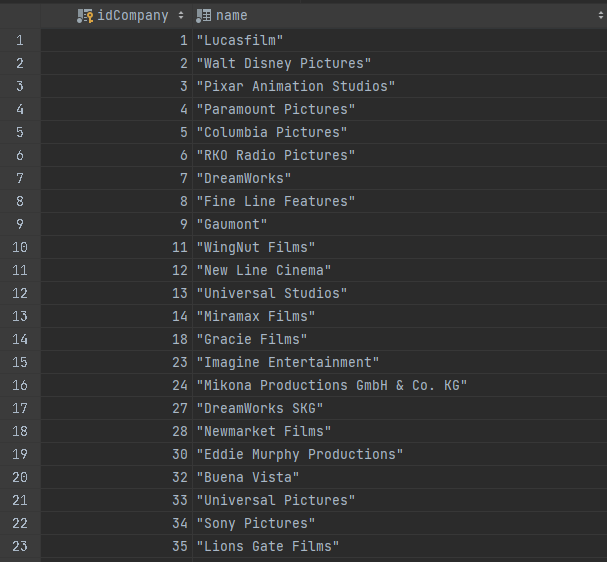
## Tabla Person

******

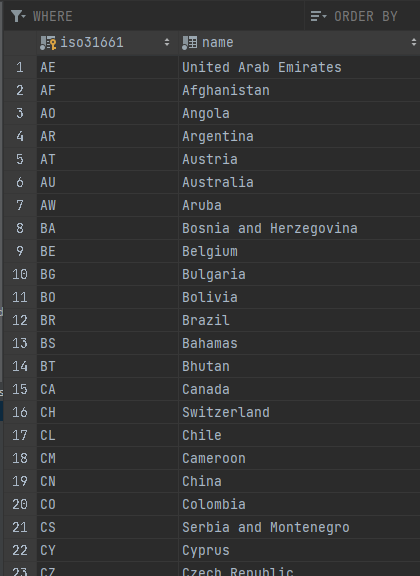
## Tabla movie\_companies

# 

## tabla production\_companies



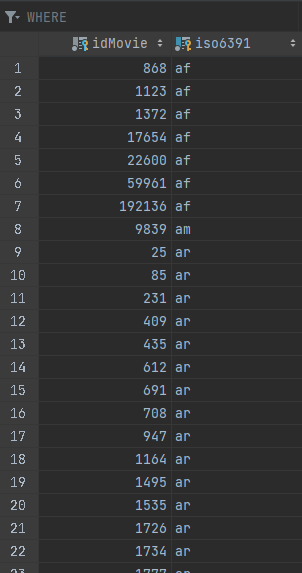
## tabla production\_countries



## tabla movie\_countries

# 

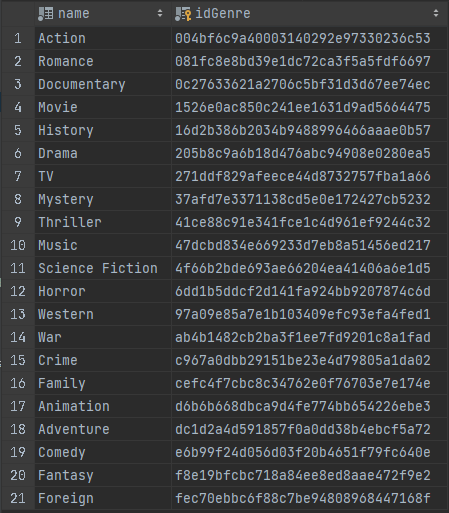
## tabla spoken\_languages



## tabla languages

# 

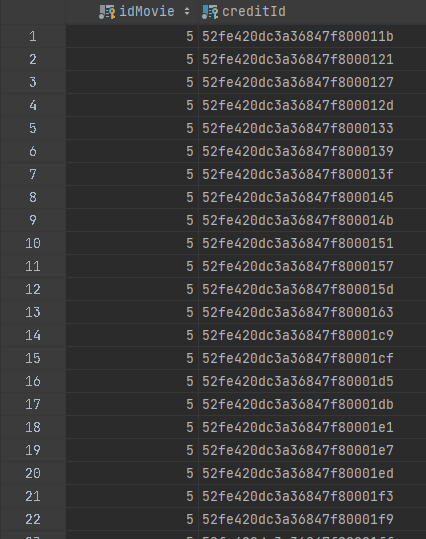
## tabla genre



## tabla movie\_genres

# 

## tabla movie\_crew



# Conclusión

Se concluye que dado el archivo CSV y siguiendo las respectivas normas para el desarrollo del presente proyecto, hasta el momento se ha identificado dependencias funcionales, la creación de tablas, así como la normalización de las mismas y la creación de modelo entidad- relación, así como el esquema relacional.

También se ha puesto en practica los conocimientos adquiridos durante todo el transcurso del ciclo, así como también la indagación extra de temas en los que se necesitaba reforzar para dar solución a inconvenientes que fueron surgiendo conforme al desarrollo del presente proyecto.

Como conclusión conforme se ha ido desarrollando el proyecto se debe tener en cuenta todas las formas de normalización debido a que un paso mal aplicado, el modelo que se obtiene nos lleva a una base de datos mal organizada, con redundancia de datos y dependencias incoherentes, por ende, se debe precautelar cualquier imprevisto.