

**UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA**

*La Universidad Católica de Loja*

**Entrega Preliminar del Proyecto Final**

**Fundamentos de Base de Datos**

**Autor**:

* Mateo Agustin Valarezo Ojeda

Octubre 2022 – Febrero 2023

Contenido

[1 Introducción 2](#_Toc126797115)

[2 Desarrollo del Componente 3](#_Toc126797116)

[2.1 Diseño y Modelado de la Base de datos 3](#_Toc126797117)

[2.1.1 Diseño Conceptual 3](#_Toc126797118)

[2.1.2 Diseño Lógico 3](#_Toc126797119)

[2.2 Creación del “schema” de la Base de Datos 6](#_Toc126797120)

[2.2.1 Conexión a la base de datos 6](#_Toc126797121)

[2.3 Importación del CSV 7](#_Toc126797122)

[2.4 Creación de Tablas 8](#_Toc126797123)

[2.5 Creación de procedimientos y limpieza de datos 14](#_Toc126797124)

[2.5.1 Ejecución de procedimientos 21](#_Toc126797125)

[3 Conclusiones 22](#_Toc126797126)

# Introducción

En el presente proyecto de la materia Fundamentos de Base de datos, se pretende aplicar los conocimientos obtenidos durante todo el ciclo para de esa forma trabajar con el archivo CSV llamado “movie\_dataset”, el cual fue obtenido de un repositorio de GitHub, el mismo que tendrá que ser, leído, modelado, limpiado y por ultimo explotado utilizando el lenguaje de consulta SQL específicamente en el sistema de gestión de bases de datos relacional MySQL.

# Desarrollo del Componente

## Diseño y Modelado de la Base de datos

### Diseño Conceptual

Al tener la información definida, con la tabla universal, normalizamos los datos usando Entidad – Relación e identificando los atributos que pertenecen a cada una de las relaciones, para eso, hicimos uso de la herramienta draw.io, la cual nos permitió realizar el modelo conceptual.

### Diseño Lógico

A partir del modelo conceptual, se toman los atributos correspondientes a cada tabla y se realiza un prototipo de las tablas en base a las entidades y relaciones identificadas, donde en cada tabla, declaramos una PRIMARY KEY y en caso de que corresponda la FOREIGN KEY, con el fin de que sea una base para desarrollar el esquema de nuestra base de datos.

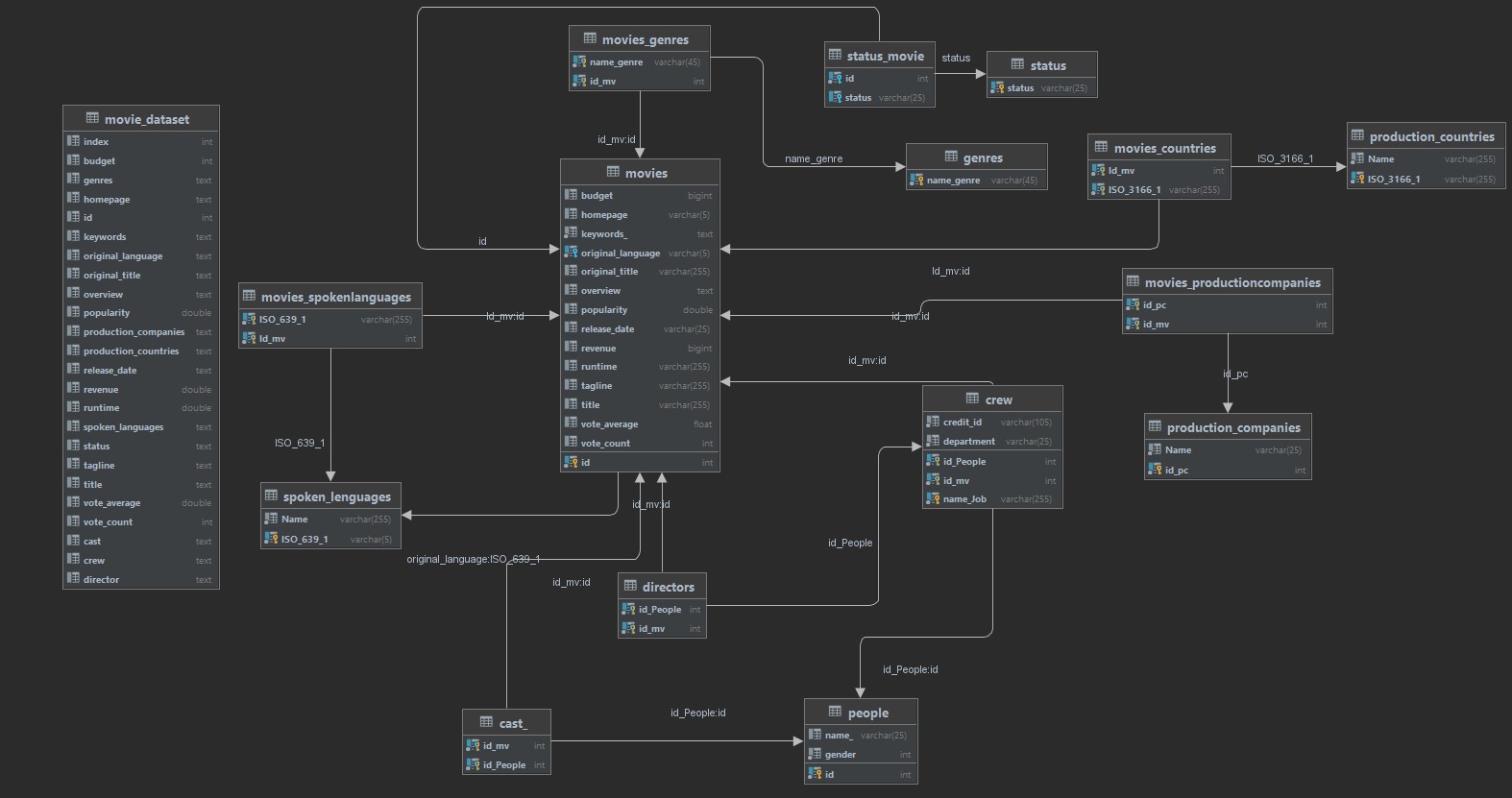


Figure 2.- Modelo Lógico

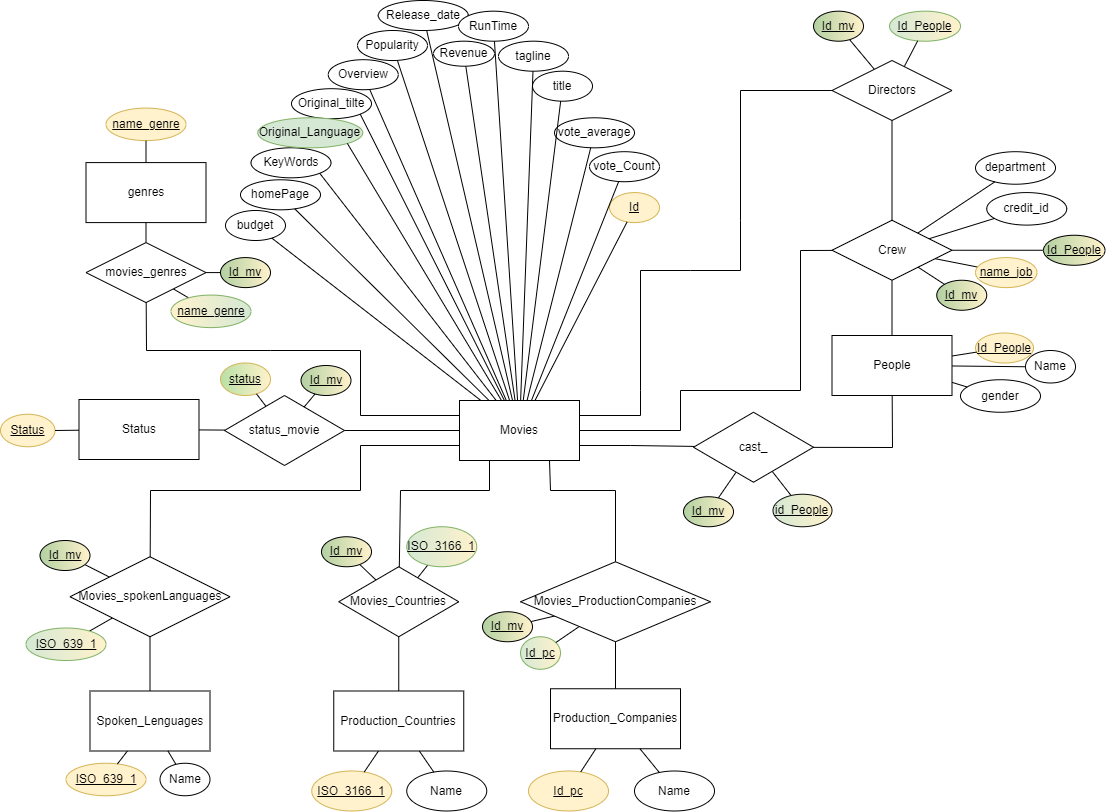


Figure 2.- Modelo Conceptual

## Creación del “schema” de la Base de Datos

Definimos la estructura de la base de datos, incluyendo las tablas, relaciones entre ellas, y los atributos de cada tabla. El "schema" actúa como un plan o esqueleto para la organización y almacenamiento de datos.

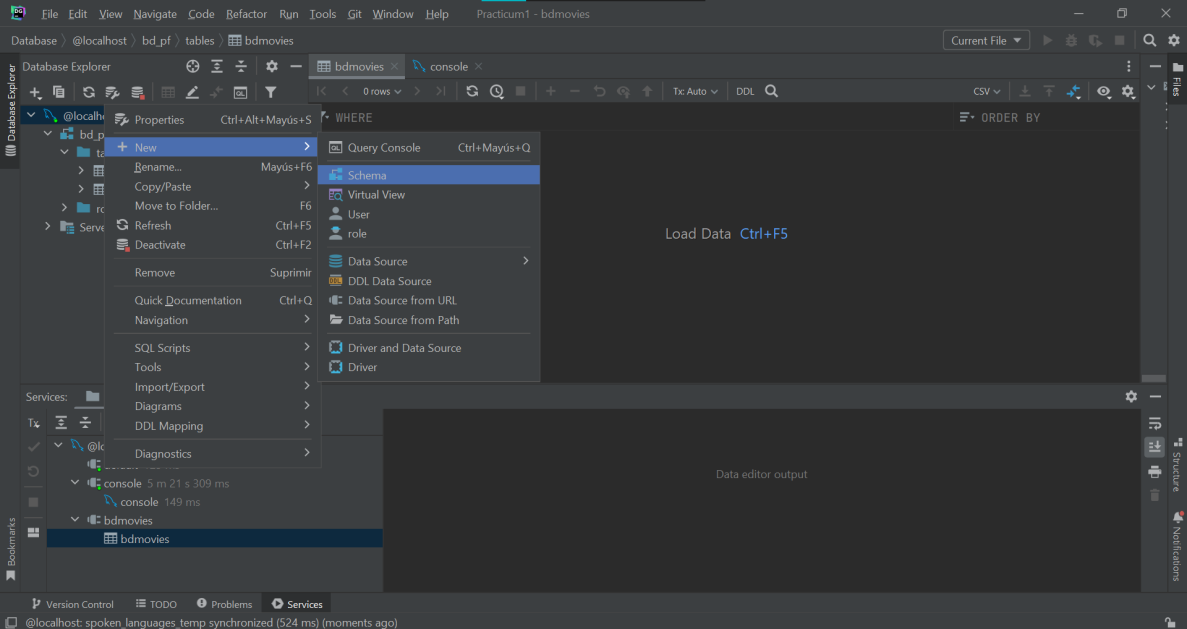


Figure 3.- Creación del "schema"

### Conexión a la base de datos

Establecemos una comunicación a la base de datos utilizando MySQL como lenguaje base y DataGrip como DBMS para así poder acceder y manipular los datos que están almacenados en ella.

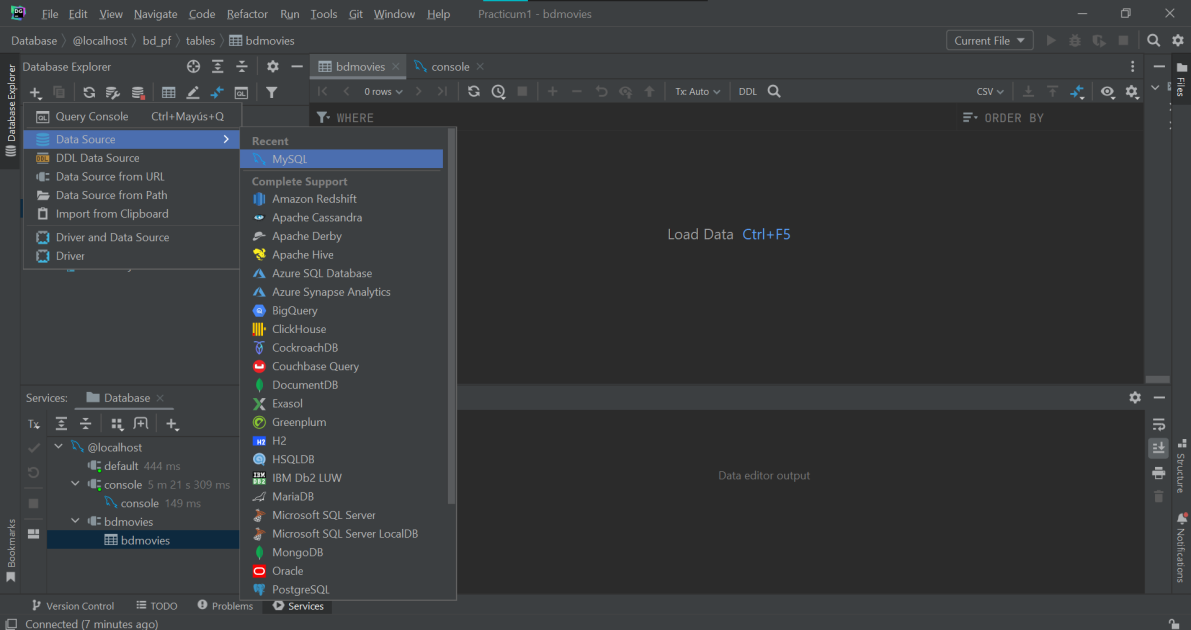


Figure 4.- Conexión a la Base de Datos

## Importación del CSV

Con el archivo CSV descargado llamado “movie\_dataset”, el cual fue obtenido de un repositorio de GitHub, entramos a DataGrip y seleccionamos el schema al cual se quiere importar el CSV.

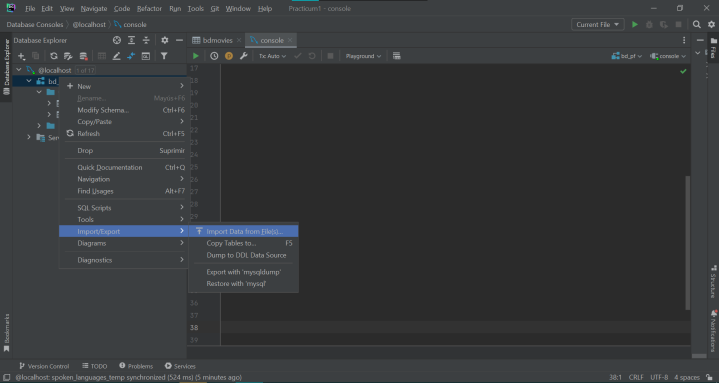


Figure 5.- Importación del CSV

DataGrip nos muestra un preview de la manera que se va a importar el CSV, donde define el tipo de dato que posee cada columna, en este caso en algunas columnas que tenían TEXT, pero su contenido era JSON tales como Crew, Production Companies, Production Countries, Spoken\_Languages, se le modifico para que el tipo de dato sea JSON.

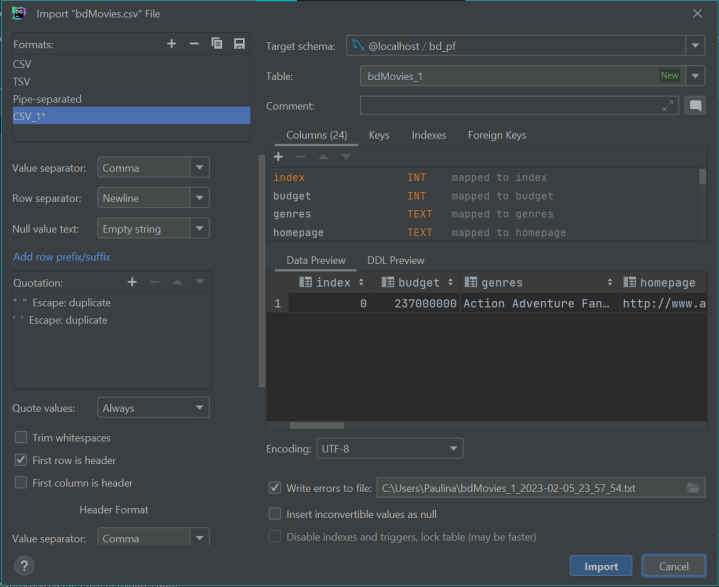


Figure 6.- Preview de Importación CSV

## Creación de Tablas

A partir de la tabla universal que se obtiene al importar el CSV, se genera varias tablas siguiendo el proceso de normalización y partiendo de los modelos realizados anteriormente, los cuales nos dan una vista panorámica de lo que se debe hacer en el DDL a continuación, se observara las tablas creadas de este proceso.

En caso de los JSON se ocuparán tablas temporales, de las cuales obtendremos todos los datos de la columna específicamente, pero en este caso algunos datos saldrían duplicados. Las tablas temporales son muy útiles al momento de realizar las definitivas.

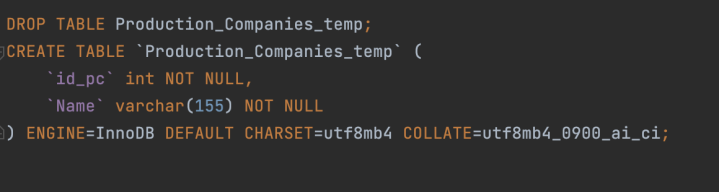


Figure 7.- Creación de la Tabla temporal Production Companies

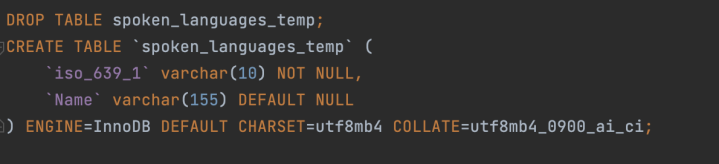


Figure 8.- Creación de la Tabla temporal Spoken Languages

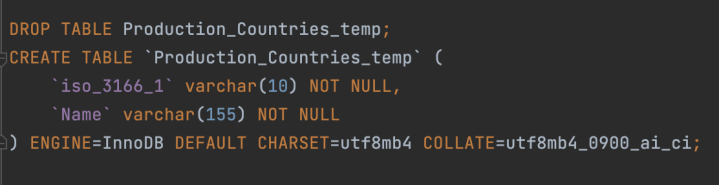


Figure 9.- Creación de la Tabla Temporal Production Countries

Con respecto a las tablas definitivas, para las columnas JSON usamos la tabla temporal, pero modificándola con un DISTINCT para obtener los datos únicos, en algunos casos para poder poner la PRIMARY KEY tendremos que hacer un ALTER TABLE. Como en los siguientes casos:

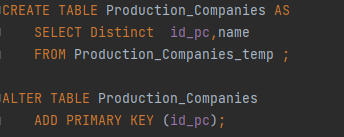


Figure 10.- Tabla Definitiva Production Companies

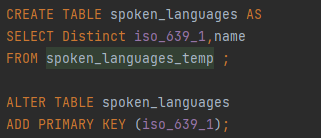


Figure 11.- Tabla Definitiva Spoken Languages

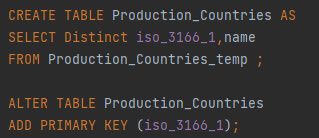


Figure 12.- Tabla Definitiva Production Countries

Para las columnas que no fueron JSON, se procede a hacer la creación de tablas normalmente.

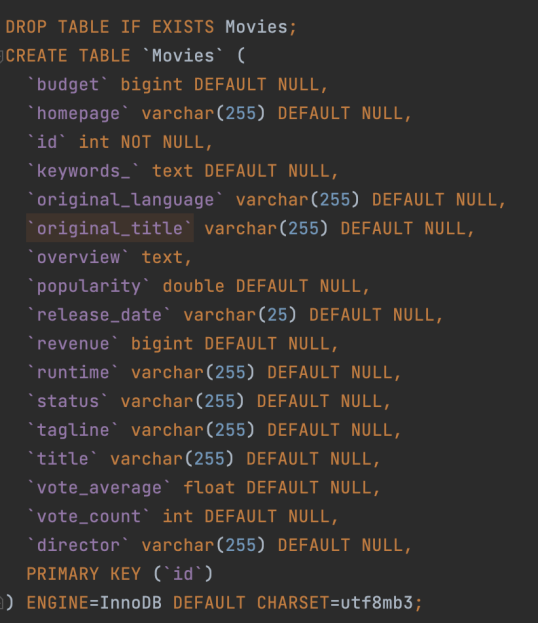


Figure 13.- Creación de la Tabla Movies con los atributos atómicos

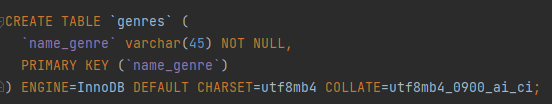


Figure 14.- Creación de la Tabla Genres

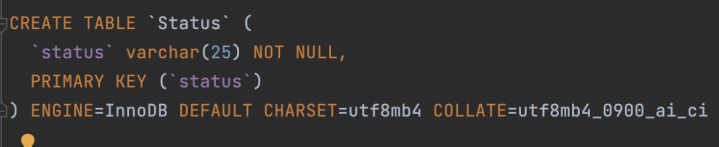


Figure 15.- Creación de Tabla Status

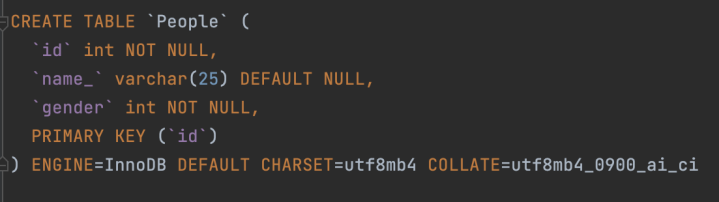


Figure 16.- Creación de la Tabla People

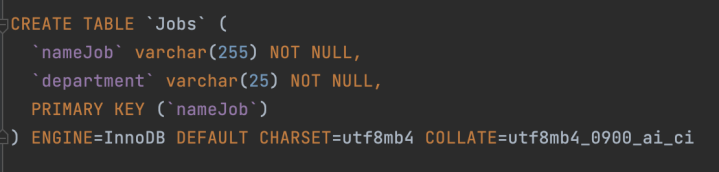


Figure 17.- Creación de la Tabla Jobs

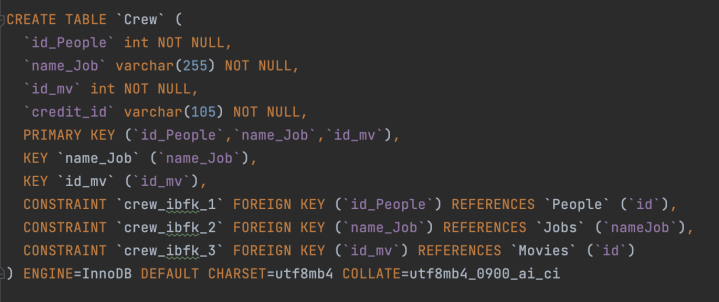


Figure 18.- Creación de la Tabla Crew



Figure 19.- Creación de la Tabla Directors

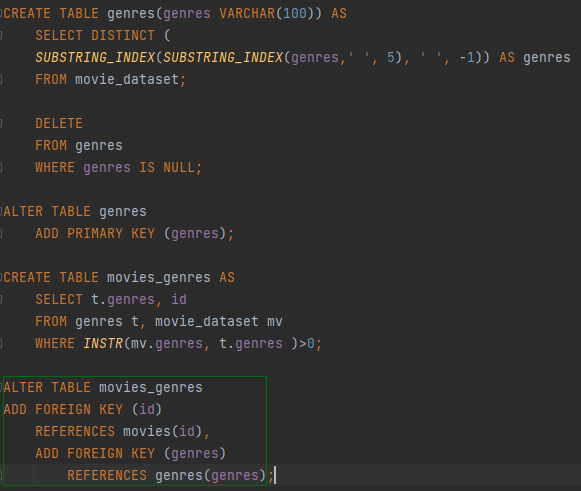


Figure 20.- Creación de la Tabla Genres

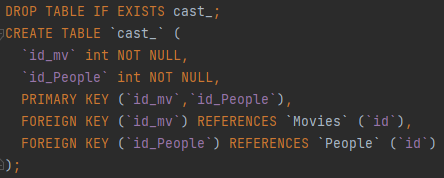


Figure 21.- Creación Tabla Cast

Las tablas de relación, las utilizaremos en relaciones de muchos a muchos, en este caso Movies que es la tabla madre, con todas las que mantiene relación, de la cuales, se sacaran las PRIMARY KEY tanto de Movies como de la otra tabla para establecer la relación.

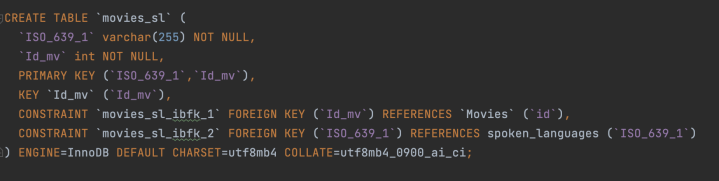


Figure 22.- Creación de la Tabla Relación (Movies- Spoken Languages)

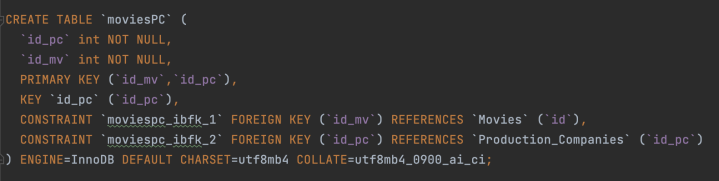


Figure 23.- Creación de la Tabla Relacional (Movies - Production Companies)

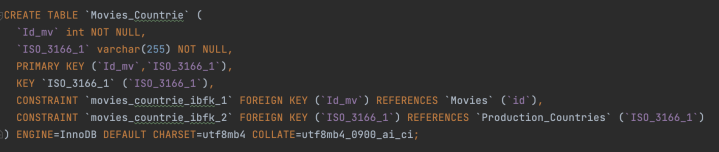


Figure 24.- Creación de la Tabla Relacional (Movies - Production Countries)

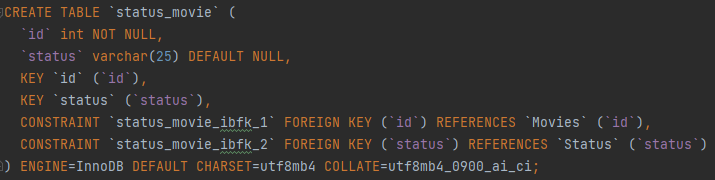


Figure 25.- Creación de la Tabla Relacional (Status - Movie)

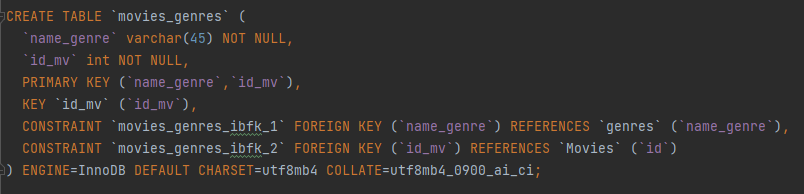


Figure 26.- Creación de la Tabla Relacional (Movies - Genres)

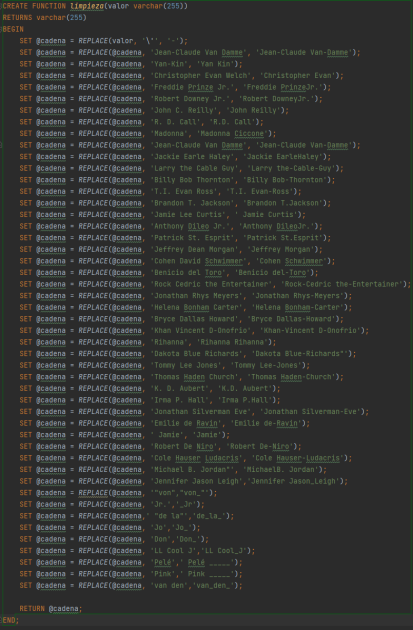


Figure 27.- Limpieza de Cast

## Creación de procedimientos y limpieza de datos

Un procedure es un proceso cuyo objetivo es que sea llamado en múltiples ocasiones en nuestro código, esto otorga ciertas ventajas relacionadas a rendimiento y optimización.

Para el proyecto, se crearon distintos procedimientos que nos permitieron extraer los datos de las columnas para poder insertar en las tablas correspondientes, en medio de estos procedimientos se limpian aquellos datos que lo necesiten, ya que algunas partes del CSV vienen mal formadas o con caracteres especiales que no son admitidos en la inserción de los datos.

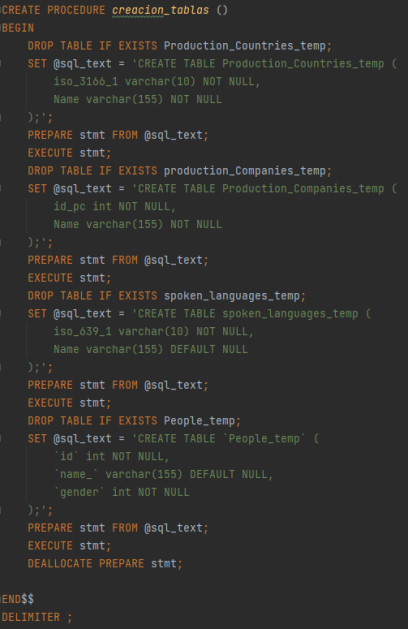


Figure 28.- Creación del Procedimiento "Creación Tablas"

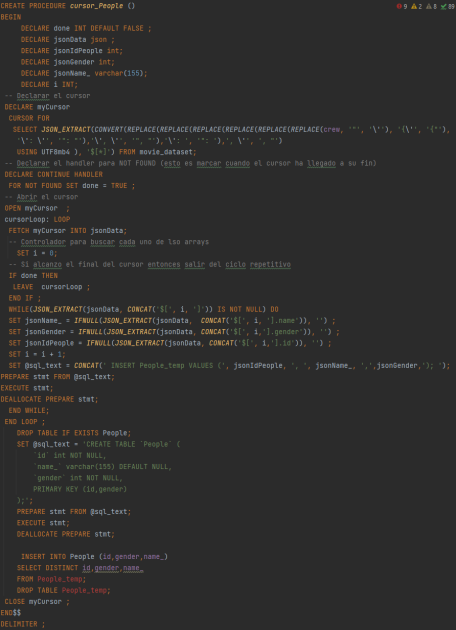


Figure 29.- Creación del Procedimiento "Cursor - People"

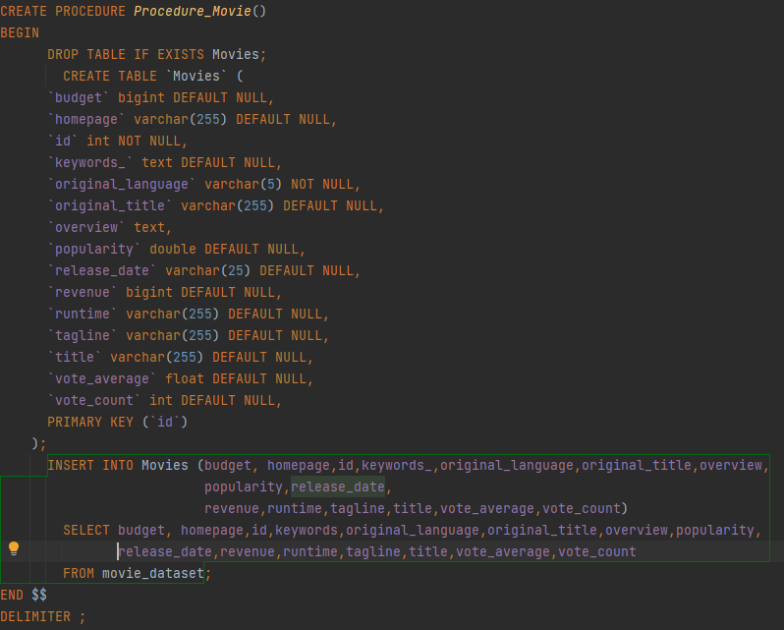


Figure 30.- Creación del Procedimiento "Procedure Movie"

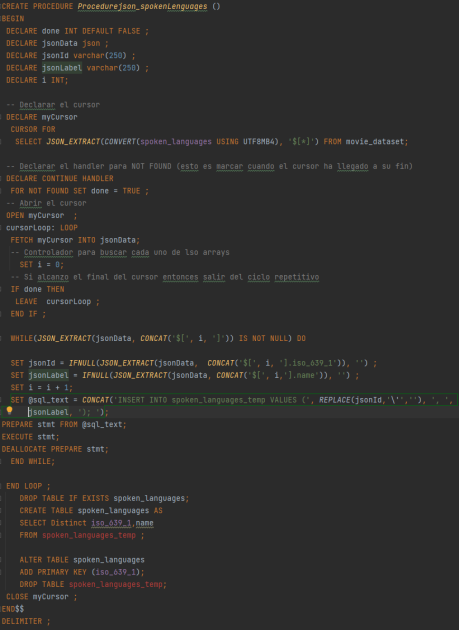


Figure 31.- Creación del Procedimiento " ProcedureJSON Spoken Languages"

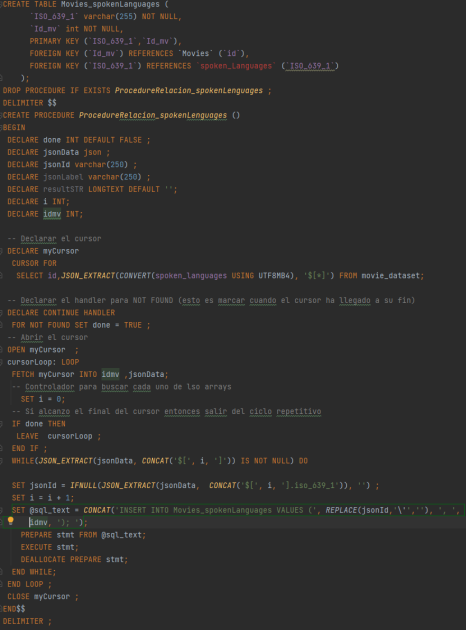


Figure 32.- Creación Procedimiento "Procedure Relacion Spoken Languages"

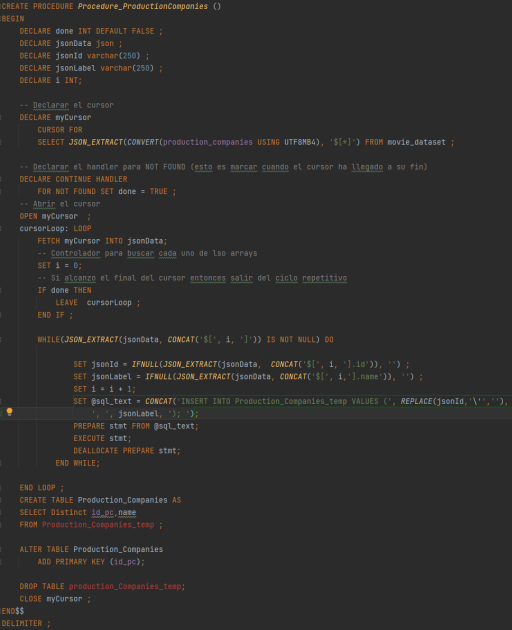


Figure 33.- Creación Procedimiento "Procedure Production Companies"

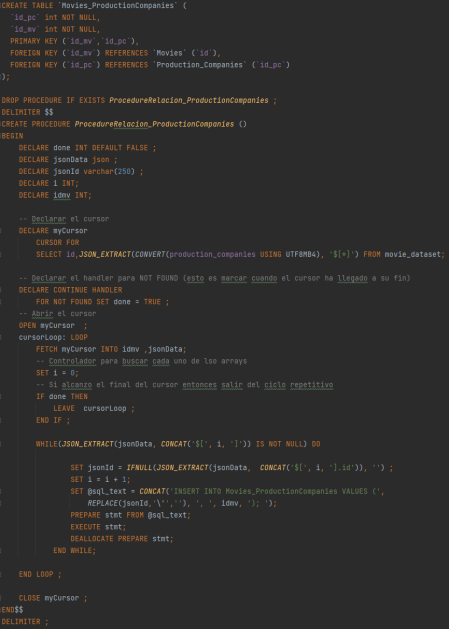


Figure 34.- Creación Procedimiento "Procedure Relacion Production Companies"

### Ejecución de procedimientos

Se hace un llamado a los procedimientos creados anteriormente, los cuales tienen funciones de limpieza e inserción de datos de cada columna respectivamente.

Image

Figure 35.- Llamado al Procedimiento "Creacion tablas"

Image

Figure 36.- Llamado al Procedimiento "Cursor People"

Image

Figure 37.- Llamado al Procedimiento "Procedure Movie"

Image

Figure 38.- Llamado al Procedimiento "Procedure JSON Spoken Languages"

Image

Figure 39.- Llamado al Procedimiento "Procedure Relacion Spoken Languages"

Image

Figure 40.- Llamado al Procedimiento "Procedure Production Companies"

Image

Figure 41.- Llamado al Procedimiento "Procedure Relacion Production Companies"

# Conclusiones

* El lenguaje MySQL es muy importante para llevar a cabo cada una de las actividades de corrección, relación, extracción e ingreso de datos del CSV, el cual, si bien presentaba varios errores en la formación de los JSON o la formación de cadenas en varias columnas, con dichas herramientas fue posible brindar una resolución que nos permita trabajar con dichos datos.
* Existen varias herramientas DBMS, sin embargo, refiriéndonos específicamente a las herramientas a nuestra disposición a lo largo del desarrollo de este proyecto integrador tanto Workbench como DataGrip son dos excelentes herramientas de trabajo, pero DataGrip al contrario de Workbench representa un mayor detalle tanto en la detección de errores, como en las opciones de importación presentadas, este tipo de herramientas facilita el proceso de desarrollo de dicho trabajo al otorgar un mayor detalle respecto a los errores que se comete conforme se avanza en la codificación de las soluciones.