

# Proyecto Final de Programación I y Base de datos

## Integrantes:

- Diego Andrés Heredia Ticona
- Israel Gutierrez Lara

## Introducción al Problema:

Nosotros trabajamos en una empresa que realiza comisiones a negocios más pequeños, en este caso la comisión proviene de una tienda que se dedica a vender mangas (Cómics Japoneses) que nos ha solicitado que creamos un diseño de base de datos para su negocio para dejar atrás uno obsoleto, que sea eficiente para poder ejecutarse correctamente durante el transcurso laboral del negocio, además también solicitaron que hagamos un software que implementa dos funciones principales:

1. Migrar la Base de Datos de la Empresa: Esto se refiere a que diseñemos un software que sea capaz de transferir sus datos desde una base de datos en mysql a una base de datos en sql server.
2. Una interfaz que se conecte a la base de datos: Esto se refiere a crear una interfaz que funcione tanto en sql server como en mysql, y que sea capaz de cumplir cuatro funciones que serán detalladas más abajo en el documento.

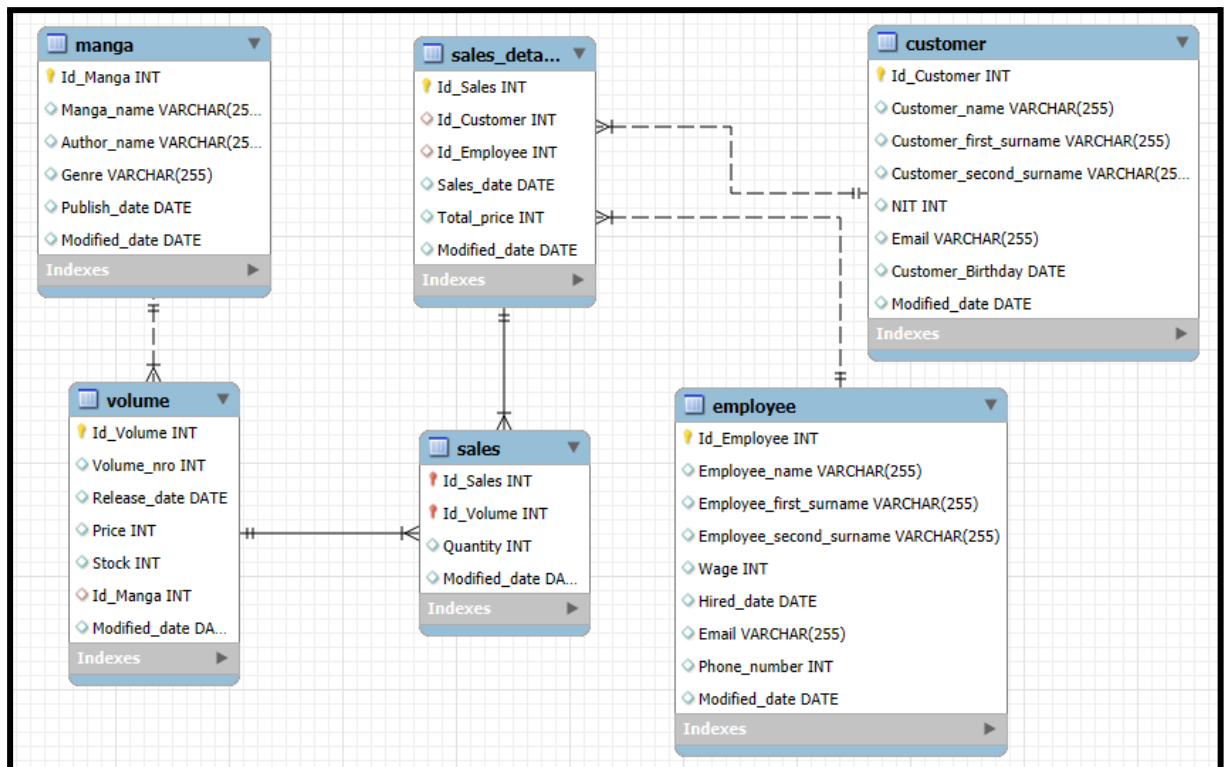
- **Creación de una Base de Datos:**

Para el nuevo modelo de base de datos, tomamos en cuenta lo principal que necesitaría un tipo de negocio como este y lo dividimos en seis tablas esenciales:

1. **Manga:** En este aspecto tomamos manga debido a que necesitamos saber a qué manga pertenece un volumen, para que se entienda mejor el manga es como el nombre de una saga de libros y el volumen es un libro de esta saga, cada manga puede variar entre n cantidad de volúmenes.
2. **Volumen:** El volumen es el producto que busca vender el negocio, para verlo de una manera más simple, el volumen es el producto y el manga una categoría.
3. **Sales (ventas):** Creamos esta tabla porque sería como un intermediario entre volúmenes y detalles de venta, ya que el detalle de venta sería como la factura, y una factura es única respecto a las ventas.
4. **Sales Details (Factura):** Es necesario que se almacenen en una tabla todas las ventas de volúmenes respecto al cliente que lo compró.

5. **Employee (Empleado):** También tomamos en cuenta que necesitamos una tabla de los empleados para ademar de sus datos personales, podemos sacar su desempeño.
6. **Customer (Cliente):** Y por último, pero no menos importante necesitamos cierta información de nuestros clientes y para no estar pidiendo a cada momento información para la compra como el NIT, es mejor guardarlos en una tabla.

### Modelo Final de la Base de Datos:



(Se adjuntará una carpeta llamada base\_de\_datos en el github donde se subirá los códigos para la creación y inserción de datos de prueba de la base de mysql, además de agregar los diseños anteriores más el diseño final en mejor resolución)

### • Software de Migración:

Respecto al software de migración, se nos solicitó que el código pueda transferir los datos desde una base en mysql a una base en sql server, además se nos impuso algunos criterios o requerimientos para un buen funcionamiento, se mostrará a continuación los criterios y requerimientos:

### • Criterios:

Por órdenes de nuestra empresa y para un mejor manejo de errores se requiere usar los siguientes aspectos:

1. dataclass de la librería dataclasses.

2. Para establecer y cerrar conexiones, deben usar try, except, finally.
3. Que sea Modular.

- **Requerimientos:**

**Funcionamiento:**

1. Tras ejecutar el script, todas las tablas deben aparecer en el nuevo manejador de BD, PostgreSQL.
2. Todas las relaciones entre registros deben mantenerse intactas.
3. La ejecución repetida del script no debe generar registros duplicados en el nuevo manejador de base de datos, por ejemplo en PostgreSQL.
4. Las conexiones deben establecerse una sola vez al inicio del programa y se deben cerrar cuando corresponda.

**Pruebas:**

1. Verificar la integridad de los datos entre cada par de tablas en MySQL y PostgreSQL a través de la cantidad de registros. (Utilizar assert)
2. Verificar el contenido de los registros dentro de cada tabla, que se hayan transferido correctamente. (Utilizar assert)
3. Realizar pruebas en las tablas que estén presentes en su BD.

El software se buscará hacer lo más automático posible por lo que solamente en la verificación y selección de base de datos será necesario insertar valores, este programa se divide en 5 grandes etapas:

1. Conexión y Verificación: Se insertarán los valores como el servidor, el usuario, etc. Para crear las conexiones con las bases de datos y se verificará si existe esa base de datos en sql server.
2. Extracción de Tablas: De manera automática se extraerán los nombres de las tablas, de sus columnas y de su código de creación de mysql.
3. Creación: Se creará la tabla y se limpiará el código de creación de las tablas de mysql para que sea apto para sql server y de forma automática se crearán las tablas y relaciones.
4. Transferencia de datos: Los datos serán extraídos de mysql y serán transferidos de acuerdo a las especificaciones de la tabla a sql server.
5. Pruebas: Una vez finalizada la transferencia se pasará a comparar la cantidad de registros y valores, además de verificar las conexiones a través de un código de sql server.

## Manejo de Errores:

Los principales errores que pueden suceder es que el usuario ingrese mal sus credenciales por lo tanto solamente allí saltarán los errores humanos, si sucede algún error después del ingreso de credenciales, posiblemente será debido a que la base de datos superó la depuración del código de creación de las tablas o que el orden lógico de la creación de tablas se vea afectado por un mal ordenamiento.

## Casos de Uso:

En caso de que la base de datos de mysql ya se encuentre existiendo en la base de datos de sql server, el programa solamente compara las dos bases, revisando la cantidad de registros por tabla y comparando los valores dentro de las tablas

Tendrá un funcionamiento normal siempre que la base a migrar sea creada con formatos normales, uso de auto\_increase, creación de conexiones entre tablas a través de llaves foráneas y en muy pocos casos será capaz de soportar una llave compuesta.

(Se adjuntará un diagrama de clase, el diseño inicial, el diseño final del código en una carpeta llamada mysql\_to\_postgres)

- **Interfaz**

1. **Caso de uso:**

Una interfaz adecuada para interactuar con la base de datos. Se presentan 4 botones para Añadir, Eliminar, Visualizar y Modificar en base a la base de datos. Cada botón presenta todas las tablas de la base de datos para que el usuario pueda escoger.

Para Añadir, se le pide al usuario ingresar los datos para cada columna, excepto aquellos datos que tenga como predeterminado el incremento automático. Finalmente se ingresan los datos a la base de datos.

Para eliminar, se pide al usuario el id, un valor único para identificar qué fila eliminar, una vez se ingresa el id, se elimina tal fila

Para visualizar, se muestra un ventana de la tabla previamente escogida por el usuario con todos los datos por columna

Para modificar, se pide al usuario el id, un valor único, el nombre de la columna y el nuevo valor a ingresar así se puede identificar qué celda o valor modificar.

## 2. Manejo de errores:

Durante cada acción, se verifica que el usuario no deje valores vacíos o que ingrese id inexistentes, también se verifica el ingreso inadecuado del valor de cada dato y el ingreso de valores id repetidos.

## 3. Requisitos:

La interfaz hace uso de 3 archivos .py, uno para llamar a la función principal, y los otros dos para hacer la conexión con mysql o sqlserver además de usar las funciones para que poder manejar la base de datos.

Se usó la librería Tkinter para un uso más dinámico y sencillo, de tal manera que el uso de la base de datos sea fácil por parte del usuario. También se usó la extensión ttk y messagebox para visualización de las tablas y el la creación de mensajes para el usuario

## DIAGRAMA DE FLUJO PARA LA INTERFAZ:

