


Structured Query Language

Proyecto: E-commerce Database



Equipo de desarrollo

Autor: Julián Brezina 

Tutor: Jonatan Canchi

.....

Profesor: Miguel Carrizo

Índice

1. Introducción.....	3
2. Temática: E-commerce de productos informáticos.....	3
3. Problemática.....	3
4.a. Diagrama de Entidad-Relación.....	3
4.b. Reverse Engineering Diagram.....	4
5. Descripción de tablas y tipo de datos.....	5
6. Script de inserción de datos.....	8
7. Vistas.....	8
8. Funciones.....	9
9. Stored Procedures.....	10
10. Triggers.....	11
11. Creación de usuarios.....	11
12. Sublenguaje TCL.....	12
13. Informes.....	13
14. Herramientas y tecnologías utilizadas.....	15
15. Futuras líneas.....	16
16. Script en formato SQL Text.....	16

1. Introducción

En el siguiente proyecto se desarrollará una base de datos funcional a partir de una temática inicial, distinguiendo en ella las tablas dimensionales, de hecho y transaccionales.

Luego a partir de la misma se explorarán las funcionalidades del lenguaje SQL y sus distintos sublenguajes DML, DDL, DCL y TCL, como ejemplo de ello, se crearán vistas, stored procedures, triggers y funciones.

2. Temática: E-commerce de productos informáticos

Se trabajará sobre la temática de E-commerce, en particular, con especialización en productos informáticos. Se trataría de una pequeña empresa que operaría en Buenos Aires a partir de su propia página web. La misma tiene alcance provincial y cuenta con estructura para realizar envíos propios.

3. Problemática

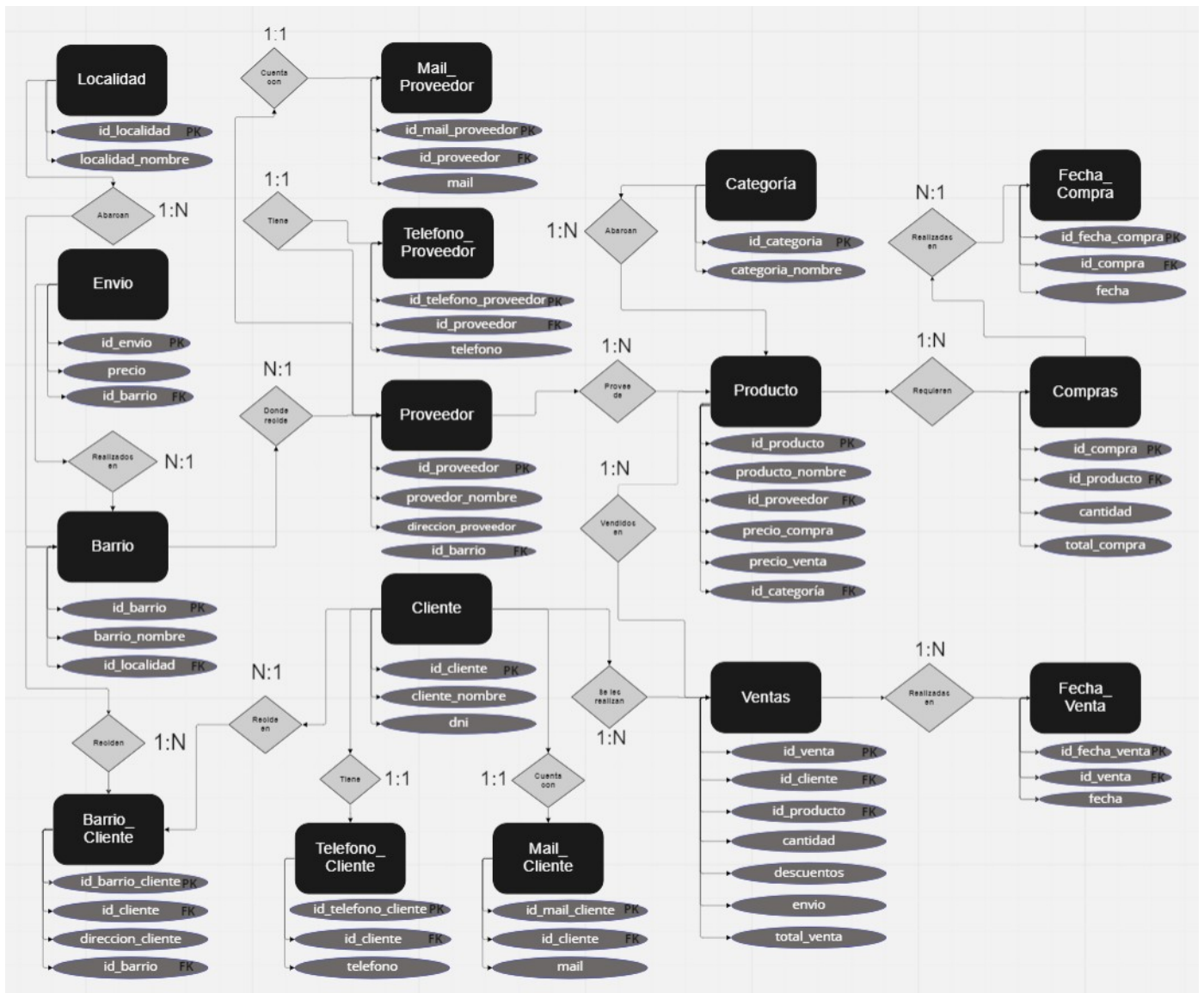
Alcanzar un modelo operativo que simule la base de datos de un E-commerce abarcando las distintas áreas de dificultad que integran al mismo tales como abastecimiento, comercialización y la realización de envíos.

Para ello, se presentaran en los apartados porterioros todos los pasos realizados para alcanzar una estructura funcional.

4.a. Diagrama de Entidad-Relación

A continuación, se presenta el del diagrama de entidad-relación creado, donde se representan las relaciones entre las distintas tablas que componen la base de datos. Con el siguiente ícono, se tendrá acceso a Miro App donde se podrá ver el DER con mayor detalle.

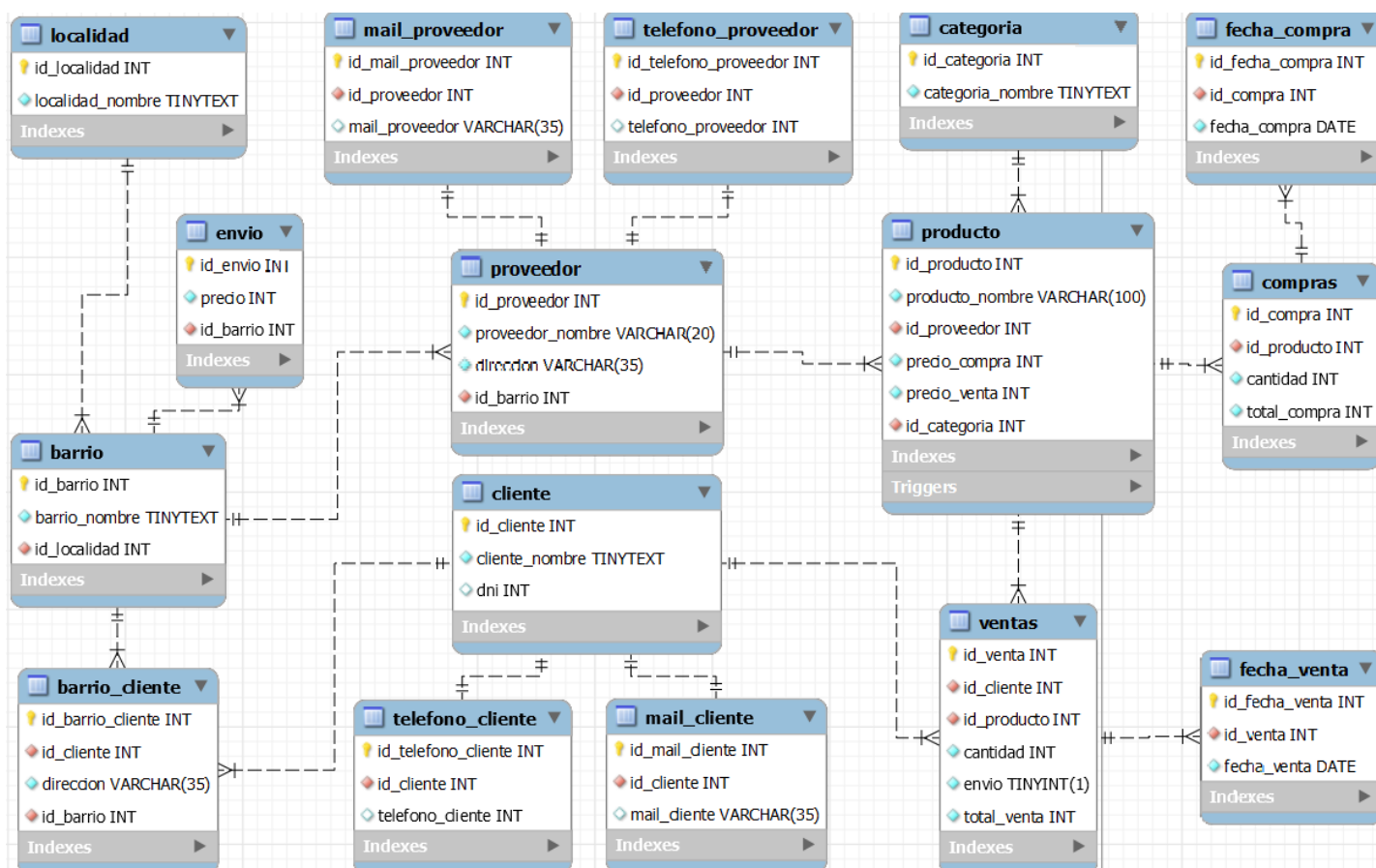




4.b. Reverse Engineering Diagram

Es el modelo de entidad relación realizado automáticamente por MySQL Workbench, lo cual cuenta con utilidad a la hora de confirmar las interconexiones entre las distintas tablas creadas que proyectamos en el DER del paso anterior.





En aras de conseguir nuestro modelo funcional, se debieron crear una serie de tablas las cuales representarían las distintas secciones con flujo de datos que compondrían a la empresa.

De querer acceder al script en formato PDF, por favor ingresar haciendo clic en el ícono que se encuentra acto seguido.



E-commerce Database

Tabla 1: Categoría

Descripción: Tabla de dimensión donde toman lugar las distintas categorías que engloban a los productos vendidos.

Tipo de clave	Campo	Tipo de Campo	N/NN	Increment	Notes
PK – index	id_categoria	INT	NOT NULL	Auto_Increment	campo único de la categoría
-	categoria_nombre	TEXT(30)	NOT NULL	-	nombre de la categoría

Tabla 2: Localidad

Descripción: Tabla de dimensión que hace referencia a las distintas localidades que integran a los diferentes barrios en la provincia de Buenos Aires.

Tipo de clave	Campo	Tipo de Campo	N/NN	Increment	Notes
PK – index	id_localidad	INT	NOT NULL	Auto_Increment	campo único de la localidad
-	localidad_nombre	TEXT(30)	NOT NULL	-	nombre de la localidad

Tabla 3: Barrio

Descripción: Tabla de dimensión que hace referencia a los distintos barrios que componen los partidos de Buenos Aires.

Tipo de clave	Campo	Tipo de Campo	N/NN	Increment	Notes
PK – index	id_barrio	INT	NOT NULL	Auto_Increment	campo único para cada barrio
-	barrio_nombre	TEXT(30)	NOT NULL	-	nombre del barrio
FK	id_localidad	INT	NOT NULL	-	campo foráneo de la localidad

Tabla 4: Envío

Descripción: Tabla de dimensión que refiere a los distintos precios en función de los barrios para los envíos realizados en las distintas ventas.

Tipo de clave	Campo	Tipo de Campo	N/NN	Increment	Notes
PK – index	id_envio	INT	NOT NULL	Auto_Increment	campo único para cada envío
FK	id_barrio	INT	NOT NULL	-	campo único para cada barrio
-	precio	DECIMAL(4,2)	NOT NULL	-	precio del envío

Tabla 5: Cliente

Descripción: Tabla de dimensión que se compone por los datos de los clientes que realizan las compras al E-commerce.

Tipo de clave	Campo	Tipo de Campo	N/NN	Increment	Notes
PK- index	id_cliente	INT	NOT NULL	Auto_Increment	campo único de cada cliente
-	cliente_nombre	TEXT(30)	NOT NULL	-	nombre del cliente
-	dni	INT	NULL	-	dni del cliente

Tabla 6: Telefono_Cliente

Descripción: Tabla de dimensión donde se encuentran el teléfono de los distintos clientes.

Tipo de clave	Campo	Tipo de Campo	N/NN	Increment	Notes
PK- index	id_telefono_cliente	INT	NOT NULL	Auto_Increment	campo único del telefono de cada
FK	id_cliente	INT	NOT NULL	-	campo único de cada cliente
-	telefono_cliente	INT	NULL	-	telefono del cliente

Tabla 7: Mail_Cliente

Descripción: Tabla de dimensión donde se encuentra el mail de los distintos clientes.

Tipo de clave	Campo	Tipo de Campo	N/NN	Increment	Notes
PK- index	id_mail_cliente	INT	NOT NULL	Auto_Increment	campo único del mail de cada cliente
FK	id_cliente	INT	NOT NULL	-	campo único de cada cliente
-	mail_cliente	VARCHAR(35)	NULL	-	mail del cliente

Tabla 8: Barrio_Cliente					
Descripción: Tabla de transaccional que enlaza las tablas Barrio y Cliente para adjudicar un precio de envío.					
Tipo de clave	Campo	Tipo de Campo	N/NN	Increment	Notes
PK- index	id_barrio_cliente	INT	NOT NULL	Auto_Increment	campo único del cliente
FK	id_cliente	INT	NOT NULL	-	campo único del cliente
-	direccion_cliente	VARCHAR(35)	NOT NULL	-	direccion del cliente
FK	id_barrio	INT	NOT NULL	-	campo único para cada barrio

Tabla 9: Proveedor					
Descripción: Tabla de dimensión que hace referencia a quienes proveen los productos para su posterior venta.					
Tipo de clave	Campo	Tipo de Campo	N/NN	Increment	Notes
PK – index	id_proveedor	INT	NOT NULL	Auto_Increment	campo único del proveedor
-	proveedor_nombre	VARCHAR(20)	NOT NULL	-	nombre del proveedor
-	direccion	VARCHAR(35)	NOT NULL	-	dirección del proveedor
FK	id_barrio	INT	NOT NULL	-	campo único del barrio

Tabla 10: Telefono_Proveedor					
Descripción: Tabla de dimensión donde se encuentra el teléfono de los distintos proveedores.					
Tipo de clave	Campo	Tipo de Campo	N/NN	Increment	Notes
PK – index	id_telefono_proveedor	INT	NOT NULL	Auto_Increment	campo único para cada teléfono de los proveedores
FK	id_proveedor	INT	NOT NULL	-	campo único del proveedor
-	telefono_proveedor	INT	NOT NULL	-	teléfono del proveedor

Tabla 11: Mail_Proveedor					
Descripción: Tabla de dimensión donde se encuentra el mail de los distintos proveedores.					
Tipo de clave	Campo	Tipo de Campo	N/NN	Increment	Notes
PK – index	id_mail_proveedor	INT	NOT NULL	Auto_Increment	campo único para cada mail de los proveedores
FK	id_proveedor	INT	NOT NULL	-	campo único del proveedor
-	mail_proveedor	VARCHAR(35)	NOT NULL	-	mail del proveedor

Tabla 12: Producto					
Descripción: Tabla de dimensión en donde toman lugar las variables relacionadas a los productos comercializados.					
Tipo de clave	Campo	Tipo de Campo	N/NN	Increment	Notes
PK – index	id_producto	INT	NOT NULL	Auto_Increment	campo único de cada producto
-	producto_nombre	VARCHAR(40)	NOT NULL	-	nombre del producto
FK	id_proveedor	INT	NOT NULL	-	campo foráneo del proveedor
-	precio_compra	DECIMAL(7,2)	NOT NULL	-	precio de compra del producto
-	precio_venta	DECIMAL(7,2)	NOT NULL	-	precio de venta del producto
FK	id_categoria	INT	NOT NULL	-	campo foráneo del categoria

Tabla 13: Compras					
Descripción: Tabla de dimensión que se compone por las variables que toman lugar en la compra de productos.					
Tipo de clave	Campo	Tipo de Campo	N/NN	Increment	Notes
PK – index	id_compra	INT	NOT NULL	Auto_Increment	campo único de cada compra
FK	id_producto	INT	NOT NULL	-	campo foráneo de cada producto
-	cantidad	INT	NOT NULL	-	indica cantidad del producto comprado
-	total_compra	DECIMAL(7,2)	NOT NULL	-	indica el importe total de la compra

Tabla 14: Fecha_Compras					
Descripción: Tabla de dimensión donde se encuentran las fechas de las distintas compras					
Tipo de clave	Campo	Tipo de Campo	N/NN	Increment	Notes
PK - index	id_fecha_compra	INT	NOT NULL	Auto_Increment	campo único de cada fecha de compra
FK	id_compra	INT	NOT NULL	Auto_Increment	campo único de cada compra
-	fecha_compra	DATETIME	NOT NULL	-	indica fecha de la compra

Tabla 15: Ventas					
Descripción: Tabla de hecho donde se encuentran las variables involucradas con las ventas de productos del E-commerce.					
Tipo de clave	Campo	Tipo de Campo	N/NN	Default	Notes
PK - index	id_venta	INT	NOT NULL	Auto_Increment	campo único de cada venta
FK	id_cliente	INT	NOT NULL	-	campo foráneo de cada cliente
FK	id_producto	INT	NOT NULL	-	campo foráneo de cada producto
-	cantidad	INT	NOT NULL	-	indica cantidad del producto vendido
-	envio	BOLEEN	NOT NULL	-	indica el importe del envío
-	total_venta	DECIMAL(7,2)	NOT NULL	-	indica el importe total de la venta

Tabla 16: Fecha_Venta					
Descripción: Tabla de dimensión donde se encuentran las fechas de las distintas ventas.					
Tipo de clave	Campo	Tipo de Campo	N/NN	Default	Notes
PK - index	id_fecha_venta	INT	NOT NULL	Auto_Increment	campo único de cada fecha de venta
FK	id_venta	INT	NOT NULL	-	campo único de cada venta
-	fecha_venta	DATETIME	NOT NULL	-	indica fecha de la venta

6. Script de inserción de datos

Seguidamente a la creación de tablas, se procedió con la inserción manual de datos en las mismas.

De querer visualizar el script en formato PDF, por favor hacer clic en el siguiente ícono.



7. Vistas

En el presente apartado se presentan las Vistas creadas en la base de datos.

Las mismas son consultas SQL almacenadas que permiten la visualización de tablas virtuales, ahorrando con ello tiempo, y, adicionalmente, las mismas se brindan a distintos usuarios para que estos tengan un acceso selectivo de nuestra base de datos, incrementando por tanto el nivel de seguridad.

Las vistas creadas fueron las siguientes:

- **compras_grandes:** Cumple la función de devolver una tabla compuesta por compras superiores a los \$550,000.00 para identificar las fuentes más relevantes del costo de las operaciones del E-commerce.
- **venta_cliente:** Su función radica en permitir la visualización del total recaudado por cliente.
- **iva_credito:** La funcionalidad de la misma radica en brindar el IVA crédito resultante de cada compra realizada.
- **iva_debito:** Su función radica en brindar el IVA débito resultante de cada venta realizada.
- **costo_envio_venta:** La funcionalidad de la misma radica en brindar el costo de envío involucrado en las distintas ventas en función del barrio del cliente.

De querer visualizar el script utilizado, presentado en formato PDF, por favor acceder al siguiente ícono.



8. Funciones

Las mismas son consultas SQL almacenadas en la base de datos que procesan parámetros ingresados por el usuario devolviendo un determinado resultado. Ahorrando por tanto tiempo e incrementando la eficiencia por la automatización y la reducción de error.

A continuación se brindan las funciones creadas en nuestra base de datos:

- **calcular_iva_a_pagar:** Función no relacionada a tablas existentes, cuya operatividad radica en simplificar al usuario la obtención del total de IVA a pagar en función del ingreso de variables de IVA débito e IVA crédito.

- `producto_size`: La función ofrece la funcionalidad de otorgarle una categoría a los productos en función de su precio unitario de compra, permitiendo por tanto clasificarlos para dar pie a realizar operaciones específicas sobre los mismos.

En el caso de querer visualizar el script utilizado y los resultados obtenidos, por favor acceder mediante el ícono que se encuentra a continuación.



9. Stored Procedures

De forma similar a las funciones en cuanto estructura, los stored procedures son consultas SQL almacenadas en nuestra base de datos las cuales al ejecutarlas permiten al usuario realizar operaciones tanto simples como complejas efectuadas en un servidor independiente que no influye en los procesos de la computadora personal.

En el siguiente apartado se brindarán los stored procedures creados para otorgar funcionalidad a la base de datos del E-commerce:

- `order_products`: La funcionalidad de la misma radica en devolver la tabla productos con un orden en específico para permitir la fácil búsqueda sobre la misma o bien crear un reporte a partir de ello, dicho orden puede ser ya sea a partir de columnas numéricas o alfanuméricas.
- `ventas_según_mes`: La funcionalidad del mismo radica en filtrar las ventas según el mes, uniéndolo a columnas de diversas tablas a fin de brindar datos significativos, lo que es de utilidad al momento de generar reportes, tales como el Cash Flow.
- `product_by_name`: La funcionalidad del mismo radica en filtrar productos en función de su nombre en específico, ello es de utilidad para consultar existencia o disponibilidad de productos.

En caso de querer visualizar el script, por favor acceder desde el ícono dispuesto a continuación.



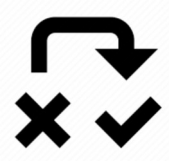
10. Triggers

Los mismos serían un conjunto de sentencias almacenadas en la DB las cuales se ejecutan de forma automática cuando acontece un evento DML en específico.

Para nuestra base de datos, se crearon los siguientes:

- **producto_min_price**: Su funcionalidad radica en volver un mensaje de error cuando se insertan en la tabla producto precios de venta iguales o inferiores a 0.
- **auditoria_producto**: Permite, a partir de la inserción de datos en tabla productos, almacenar qué producto se incorporó, quién lo ha hecho y cuándo, realizado ello en una nueva tabla denominada auditoria_producto.

Para el acceso a el script utilizado, utilizar por favor mediante el siguiente ícono.



11. Creación de usuarios

- La funcionalidad de la creación de usuarios radica moderar el acceso a nuestra DB a partir de asignar a cada uno una serie de permisos determinados en función del área o departamento del empleado o agente externo. Por ejemplo, de solo lectura sobre una determinada tabla.

- En este caso se crearon 2 usuarios, julian@localhost y elena@localhost, en tanto que el primero tiene permisos de solo lectura en todas las tablas, el segundo cuenta con permiso de lectura, modificación e inserción sobre todas las tablas, por otra parte, ninguno cuenta con permisos de eliminación de datos.

Por información adicional respecto al script, acceder por favor al ícono ubicado a continuación



12. Sublenguaje TCL

El mismo sirve para confirmar o restablecer el DML realizado sobre alguna tabla. Por ejemplo, confirmar una inserción de datos.

En nuestro caso, se aplicó en los dos siguientes escenarios.

- **TCL 1 ROLLBACK:** Se procedió a crear una nueva tabla, para en ella, insertar datos, los cuales fueron ulteriormente eliminados, para, por último, reestablecerlos con un ROLLBACK.
- **TCL 2 SAVEPOINT:** En la tabla anterior se insertaron 10 nuevas filas de datos, pero, cada 5 filas insertadas, se colocó un SAVEPOINT, denominados lote_1 y lote_2 respectivamente, lo cual permite confirmar o restablecer lo modificado en una tabla hasta dichos punto.
Por último, se paso a hacer un ROLLBACK hacia el lote 1, por lo tanto, no se consideraron ninguna de las inserciones realizadas posteriores a dicho punto.

Para acceder al script en formato PDF, ingresar por favor mediante el siguiente ícono:



13. Informes

A partir de nuestra DB, se prosiguió con la realización de informes, entendidos estos como insights obtenidos a partir de nuestros datos.

Para ello, se siguieron los siguientes pasos:

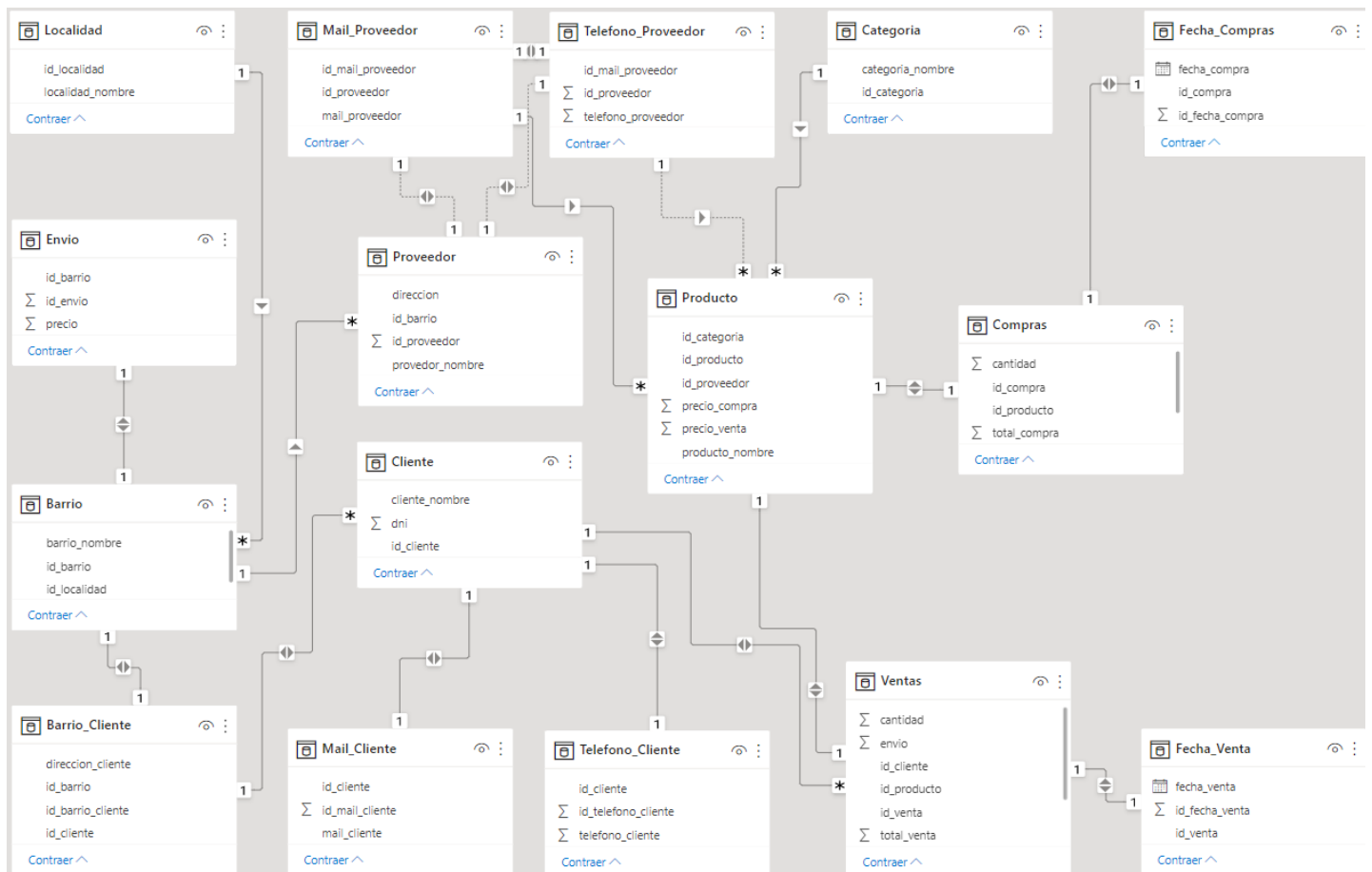
13.a) Para la elaboración de informes se migraron las tablas y datos insertados en MySQL Workbench a un archivo de Excel, al cual, se podrá acceder mediante el siguiente ícono:



13.b) Posteriormente a ello, se conectó Power BI Desktop al archivo Excel creado.

13.c) Luego, se realizó una modificación menor en las relaciones de las tablas a fin de respetar las conexiones realizadas en MySQL. Obteniendo con ello el siguiente Modelo de Entidad Relación.



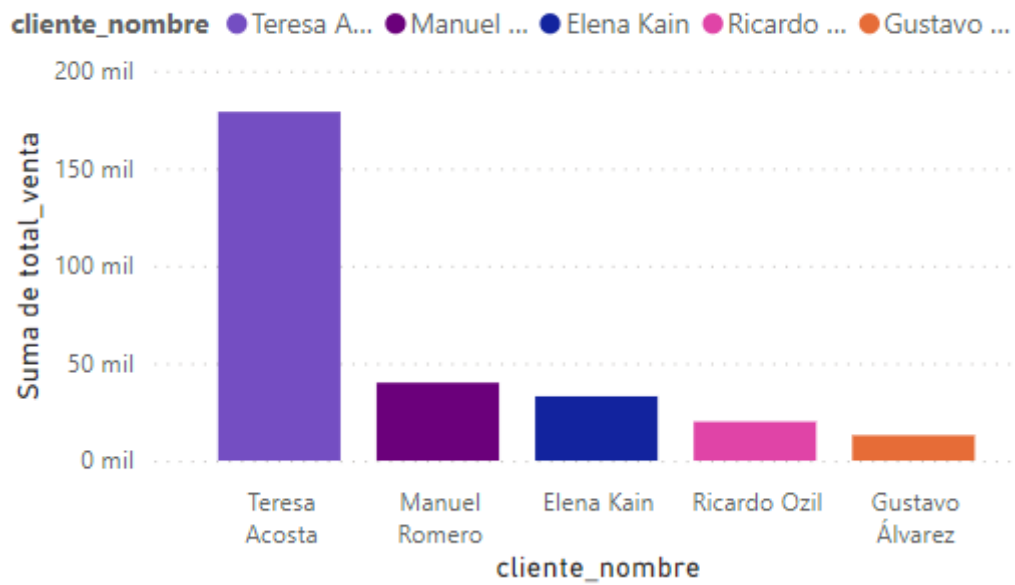


13.d) Ulteriormente, se generaron distintos gráficos para comprobar el alcancé de los datos ingresados. Obteniendo, a modo de ejemplo, un conjunto de KPIs los cuales reflejan el Total de Compras, IVA Crédito, Total de Ventas e IVA Débito.



13.e) Para terminar, se generó un gráfico de columnas para identificar el cliente que mayor gasto hizo en la empresa.

Ventas por cliente



14. Herramientas y tecnologías utilizadas

A continuación se presentan las distintas herramientas utilizadas en el Proyecto:



MySQL Workbench



MiroApp



G. Drive



M. Word.



M. Excel



M. Power BI

- **MySQL Workbench:** Creación de la base de datos, aplicación de ingeniería inversa e implementación de los distintos lenguajes y sublenguajes del SQL.
 - **Miro App:** Generación del DER.
 - **Google Drive:** Almacenamiento en línea.
 - **Microsoft Word:** Recopilación de actividades, presentación final del proyecto.
 - **Microsoft Excel:** Creación de tablas y su contenido.
 - **Microsoft Power BI:** Generación de gráficos.
-

15. Futuras líneas

En un futuro, la estructura de la base de datos puede adaptarse a un modelo de negocio real.

A partir de dicha adaptación, el negocio pasará a generar una cantidad significativa de datos relacionados a la compra y venta de productos, y, ello, abre paso a la generación de informes avanzados, donde, por ejemplo, se podría representar lo siguiente:

- **Segmentación geográfica:** Identificar los barrios y localidades que generan la mayor cantidad de ingresos, y, con ello, aplicar marketing localizado y reducir los costos de transporte en dicha área para potenciar las ventas y atraer nuevos clientes.
 - **Segmentación demográfica:** Identificación etaria y de género respecto a nuestros compradores, dando lugar a personalizar aún más la oferta de marketing, adaptando las distintas publicidades al mercado meta.
 - **Logística:** Identificar productos con mayor y menor cantidad de ventas para mejorar la logística mediante el aumento de stock y la compra al por mayor o la discontinuación del mismo.
 - **Automatización:** A partir de la información ingresada en nuestra base de datos la generación de un Dashboard Power BI en línea.
-

16. Script en formato SQL Text



