

PROYECTO FINAL - CURSO SQL

COMISIÓN 45230

Negocio de computación CompuHouse

Elaborado por:
Leonardo Risso

Tutor:
Vanina Giselle Aranda

Profesor:
César Aracena

Contenido

Introducción	3
Objetivo	4
Diagrama Entidad Relación (DER)	4
Descripción de tablas y definiciones por tipos de datos	7
Creación de la base de datos en MySQL	10
Insertión de datos en las tablas	11
Creación de Vistas	12
Creación de Funciones	13
Creación de Stored Procedures	14
Creación de Triggers	15
Generación de Outputs y Análisis de Datos	18
Tecnologías utilizadas	21
Conclusiones	21

Introducción

Dado por cumplido el curso de SQL de la plataforma de enseñanza online CODERHOUSE, se presenta el siguiente Proyecto Final en el cual se encuentra todo lo aprendido durante el desarrollo del mismo: Temática seleccionada, elaboración del Diagrama Entidad Relación (DER), descripción de las tablas y definición de tipos de datos, creación de base de datos y tablas en MySQL, inserción de datos en las tablas creadas, diagramación de vistas, funciones, stored procedures y triggers.

La temática seleccionada está relacionada al negocio de computación en venta de hardware y el proyecto tiene el nombre de CompuHouse, la información contenida en la base de datos en parte fue extraída de paginas como mercado libre para tratar de dar un nombre y precio de productos de acuerdo a la actualidad, por otro lado la mayor parte de tablas fueron creadas por criterio propio ya que lo relacionado con computación fue uno de mis hobbies favoritos.

Objetivo

El objetivo de la temática seleccionada es crear una base de datos con los productos de computación de la actualidad y próximos a salir, de los cuales se pueda visualizar cuales son los más vendidos y solicitados para poder atraer más clientes, ver qué empleados son los que realizan más ventas y también saber cuales son los clientes que más compras realizan.

Diagrama Entidad Relación (DER)

Como siguiente se presenta el Diagrama Entidad Relación el cual está conformado por 10 tablas, inicialmente fueron 6 tablas y a medida que pasó el curso y se obtuvieron nuevos conocimientos se fueron agregando nuevas tablas por lo que se modificó varias veces el proyecto hasta llegar a los siguientes DER.

Diagrama Entidad Relación (DER)

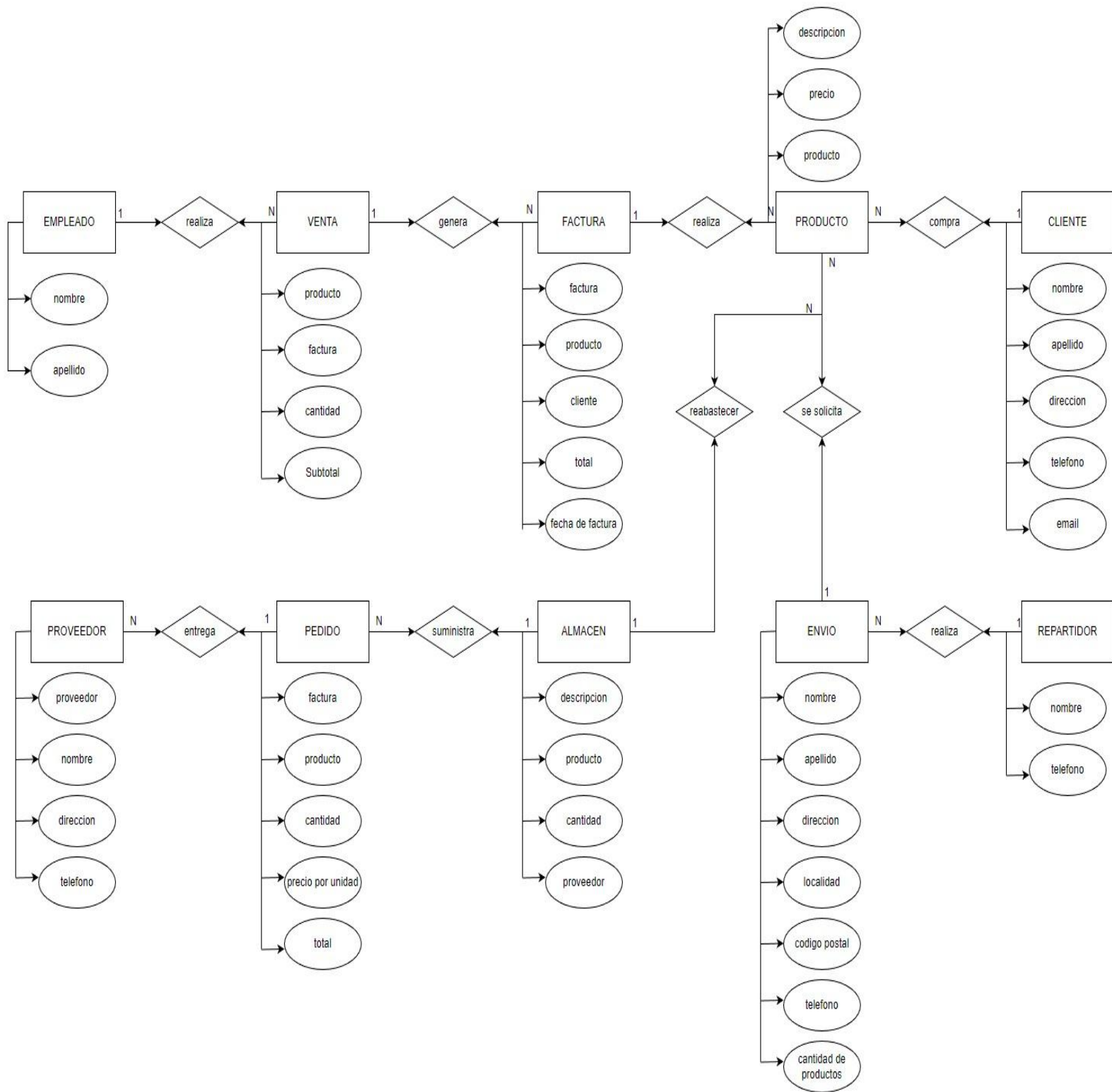
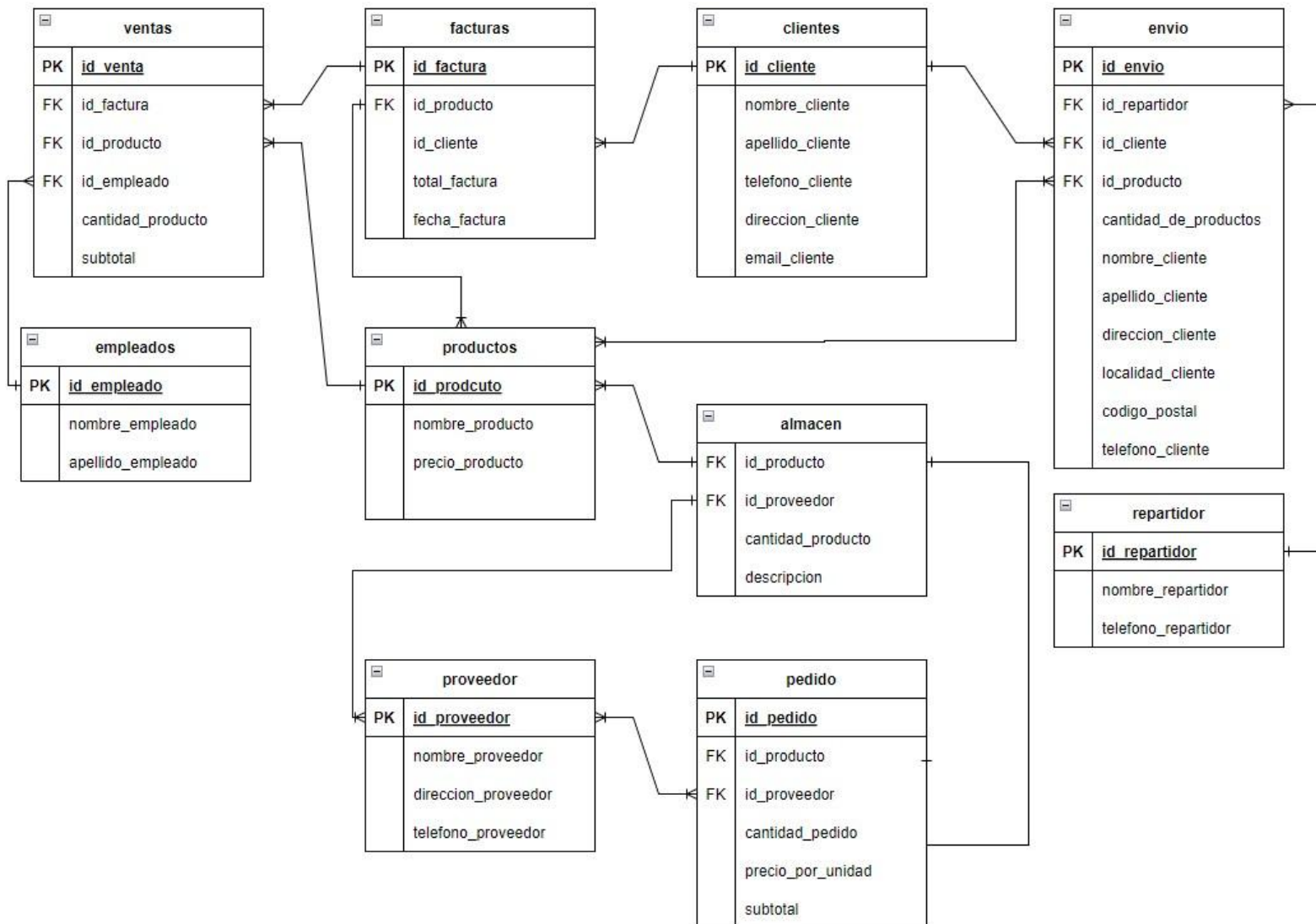


Diagrama Entidad Relación (DER)



TABLAS PARA TRIGGERS

clientes_nuevos_tr	actualizacion_de_clientes_tr	productos_nuevos_tr	actualizacion_de_productos_tr
FK id_cliente	FK id_cliente	FK id_productos	FK id_productos
nombre_cliente	nombre_cliente	nombre_producto	nombre_producto
apellido_cliente	apellido_cliente	precio_producto	precio_producto
telefono_cliente	telefono_cliente	usuario	usuario
direccion_cliente	direccion_cliente	fecha_hora	fecha_hora
email_cliente	email_cliente	db	db
usuario	usuario	version	version
fecha_hora	fecha_hora		
db	db		
version	version		

Descripción de tablas y definiciones por tipo de datos

Tabla	clientes						
Descripción	Es la tabla que almacena los clientes del local con sus datos descriptivos						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	AUTO_INCREMENT	NOTES
P.K	id_cliente	int		X	X	X	ID designado a cada cliente
	nombre_cliente	varchar	30	X			
	apellido_cliente	varchar	30	X			
	telefono_cliente	varchar	30				
	direccion_cliente	varchar	80				
	email_cliente	varchar	60				

Tabla	pedido						
Descripción	Es la tabla que se encarga de los pedidos realizados a proveedores						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	AUTO_INCREMENT	NOTES
P.K	id_pedido	int		X	X	X	ID que se designa a cada pedido
F.K	id_producto	int		X			
F.K	id_proveedor	int		X			
	cantidad_pedido	int		X			
	precio_por_unidad	float		X			
	subtotal	float		X			

Tabla	productos						
Descripción	Muestra los distintos productos y precio que posee el local						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	AUTO_INCREMENT	NOTES
P.K	id_producto	int		X	X	X	ID de para identificar cada producto
	nombre_producto	varchar	120	X			
	precio_producto	float		X			

Tabla	empleados						
Descripción	Los distintos empleados destinados a la atencion y venta de productos						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	AUTO_INCREMENT	NOTES
P.K	id_empleado	int		X	X	X	ID que se le asigna a cada empleado
	nombre_empleado	varchar	30	X			
	apellido_empleado	varchar	30	X			

Tabla	almacen						
Descripción	Es la tabla que se encarga de almacenar la cantidad de productos disponibles en el local						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	AUTO_INCREMENT	NOTES
F.K	id_producto	int		X			ID de la tabla productos
F.K	id_proveedor	int		X			ID de la tabla proveedores
	cantidad_producto	int		X			
	descripcion	varchar	200				

Tabla	Proveedores						
Descripción	Tabla que se encarga de los distintos proveedores a los cuales se realizan los pedidos						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	AUTO_INCREMENT	NOTES
P.K	id_proveedor	int		X	X	X	ID para identificar cada proveedor
	nombre_proveedor	varchar	30	X			
	direccion_proveedor	varchar	80				
	telefono_proveedor	varchar	30				

Tabla	facturas						
Descripción	Tabla en la cual se generan las facturas de las compras realizadas						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	AUTO_INCREMENT	NOTES
P.K	id_factura	int		X	X	X	ID de cada factura generada
F.K	id_cliente	int		X			ID de la tabla cliente
FK	id_producto	int		X			ID de la tabla producto
	total_factura	float		X			
	fecha_factura	date		X			

Tabla	ventas						
Descripción	Venta realizada por cada empleado para determinado cliente						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	AUTO_INCREMENT	NOTES
P.K	id_venta	int		X	X	X	ID de cada venta realizada
F.K	id_factura	int		X			
F.K	id_producto	int		X			
F.K	id_empleado	int		X			
	cantidad_producto	int		X			
	subtotal	float		X			

Tabla	repartidor						
Descripción	Muestra los datos del repartidor que va a realizar el envío						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	AUTO_INCREMENT	NOTES
P.K	id_repartidor	int		X	X	X	ID de cada venta realizada
	nombre_repartidor	varchar	30	X			
	telefono_repartidor	varchar	30	X			

Tabla	envio						
Descripción	muestra los envios solicitados por los clientes						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	AUTO_INCREMENT	NOTES
P.K	id_envio	int		X	X	X	ID de cada venta realizada
	id_repartidor	int		X			ID del repartidor que realizara el envio
	id_cliente	int		X			ID del cliente que recibira el producto
	id_producto	int		X			ID del producto que compro el cliente
	nombre_cliente	varchar	30	X			
	apellido_cliente	varchar	30	X			
	direccion_cliente	varchar	50	X			
	localidad_cliente	varchar	50	X			
	codigo_postal	varchar	20	X			
	telefono_cliente	varchar	35	X			

TABLAS PARA TRIGGERS

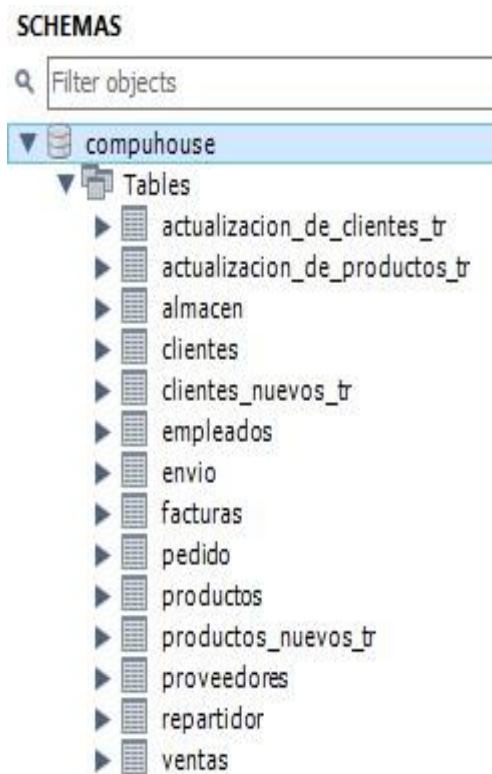
Tabla	clientes_nuevos_tr						
Descripción	Aqui se registran los clientes nuevos que se insertan en la tabla clientes						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	AUTO_INCREMENT	NOTES
F.K	id_cliente	int					id de cliente tomado de la tabla clientes
	nombre_cliente	varchar	30				nombre tomado de la tabla clientes
	apeliido_cliente	varchar	30				apellido tomado de la tabla clientes
	telefono_cliente	varchar	30				telefono tomado de la tabla clientes
	direccion_cliente	varchar	80				direccion tomada de la tabla clientes
	email_cliente	varchar	80				email tomado de la tabla clientes
	usuario	varchar	200				usuario que inserto el nuevo cliente
	fecha_hora	timestamp					fecha que se realizo el cambio
	db	varchar	200				base de datos en la cual se realizo
	version	varchar	100				version del programa

Tabla	actualizacion_de_clientes_tr						
Descripción	Aqui se registran los datos que se actualizan en la tabla clientes						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	AUTO_INCREMENT	NOTES
F.K	id_cliente	int					id de cliente tomado de la tabla clientes
	nombre_cliente	varchar	30				nombre tomado de la tabla clientes
	apeliido_cliente	varchar	30				apellido tomado de la tabla clientes
	telefono_cliente	varchar	30				telefono tomado de la tabla clientes
	direccion_cliente	varchar	80				direccion tomada de la tabla clientes
	email_cliente	varchar	80				email tomado de la tabla clientes
	usuario	varchar	200				usuario que actualizo los datos del cliente
	fecha_hora	timestamp					fecha que se realizo el cambio
	db	varchar	200				base de datos en la cual se realizo
	version	varchar	100				version del programa

Tabla	productos_nuevos_tr						
Descripción	Aqui se registran los productos nuevos que se insertan en la tabla productos						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	AUTO_INCREMENT	NOTES
F.K	id_producto	int					id de producto tomado de la tabla productos
	nombre_producto	varchar	120				nombre tomado de la tabla productos
	precio_producto	float					precio tomado de la tabla productos
	usuario	varchar	200				usuario que inserto el producto nuevo
	fecha_hora	timestamp					fecha que se realizo el cambio
	db	varchar	200				base de datos en la cual se realizo
	version	varchar	100				version del programa

Tabla	actualizacion_de_productos_tr						
Descripción	Aqui se registran las actualizaciones de datos de la tabla productos						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	AUTO_INCREMENT	NOTES
F.K	id_producto	int					id de producto tomado de la tabla productos
	nombre_producto	varchar	120				nombre tomado de la tabla productos
	precio_producto	float					precio tomado de la tabla productos
	usuario	varchar	200				usuario que realizo el cambio
	fecha_hora	timestamp					fecha que se realizo el cambio
	db	varchar	200				base de datos en la cual se realizo
	version	varchar	100				version del programa

Creación de la base de datos MySQL



Las tablas fueron elaboradas en MySQL Workbench inicialmente con el nombre de CasaComputación pero luego como se visualiza en la imagen se cambió a CompuHouse y consta de las siguientes tablas:

- almacen
- clientes
- empleados
- envio
- facturas
- pedido
- productos
- proveedores
- repartidor
- ventas

Las tablas fueron creadas en un orden específico para evitar errores en la relación entre las Primary Key y Foreign Keys.

Las tablas actualizacion_de_clientes_tr, actualizacion_de_productos_tr, clientes_nuevos_tr y productos_nuevos_tr fueron creadas para el funcionamiento de los triggers que se describen posteriormente.

Inserción de datos en las tablas

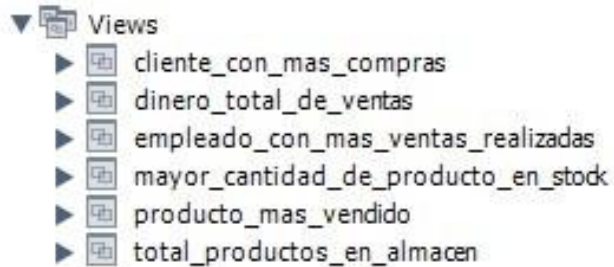
La inserción de datos se realizó con el siguiente comando:

```
INSERT INTO nombre_tabla (Campos) VALUES ('xxxxxx')
```

Cantidad de datos insertados:

- **Tabla almacen:** 30 registros
- **Tabla clientes:** 20 registros
- **Tabla empleados:** 5 registros
- **Tabla envio:** 10 registros
- **Tabla facturas:** 37 registros
- **Tabla pedido:** 9 registros
- **Tabla productos:** 30 registros
- **Tabla proveedores:** 4 registros
- **Tabla repartidor:** 3 registros
- **Tabla ventas:** 37 registros

Creación de Vistas



Se crearon 6 vistas de las cuales se explicará su función y servirán para realizar gráficos de estadísticas posteriormente.

- **cliente_con_mas_compras**

La vista fue creada para visualizar el cliente que más compras realizó, de la cual se utilizan las tablas facturas y clientes.

- **dinero_total_de_ventas**

La vista fue creada para visualizar el total de dinero recaudado de ventas de la tabla ventas.

- **empleado_con_mas_ventas_realizadas**

La vista fue creada para ver el empleado que más ventas realizó con un conteo de cada venta realizada, lo cual une las tablas ventas y empleados.

- **mayor_cantidad_de_producto_en_stock**

La vista fue creada para visualizar y calcular de mayor a menor el total de cada producto en la tabla almacén se une con la tabla productos para visualizar el nombre de cada producto.

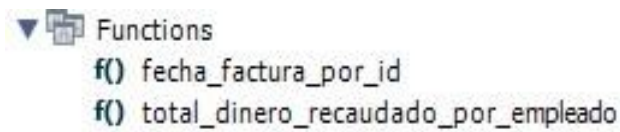
- **producto_mas_vendido**

La vista fue creada para visualizar el producto más vendido en la tabla ventas lo cual realiza un conteo de los productos iguales.

- **total_productos_en_almacen**

La vista fue creada para ver el total de productos en la tabla almacén, lo cual realiza una sumatoria total de los productos.

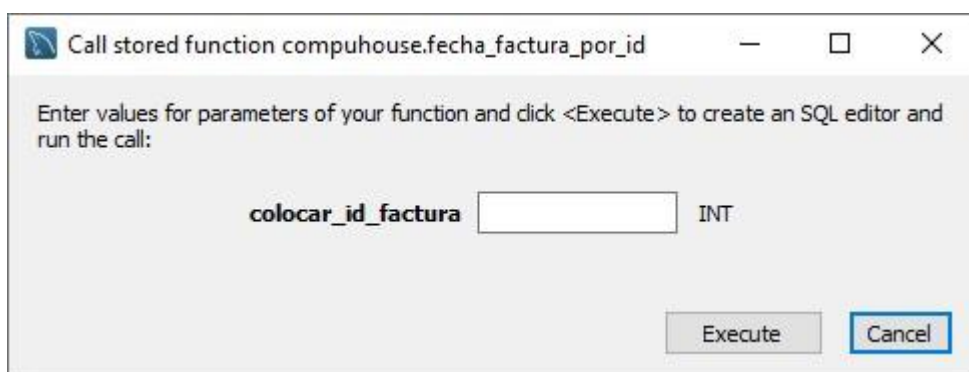
Creación de Funciones



Las funciones son las siguientes:

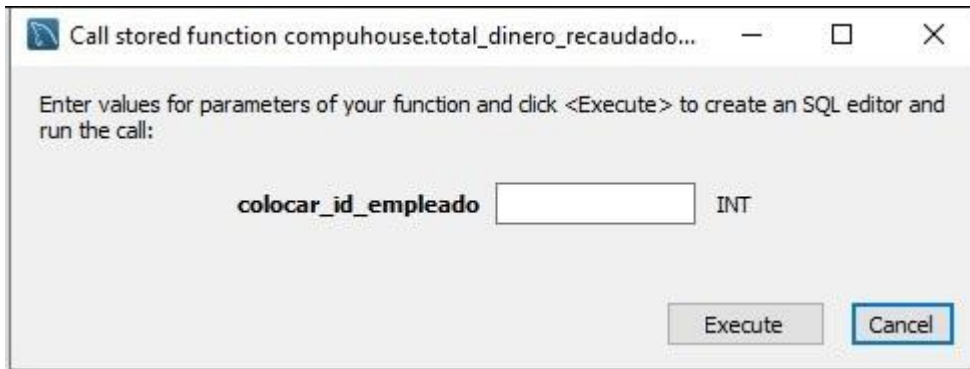
- **fecha_factura_por_id**

La función fue creada para que al poner un id de factura, busque la fecha que se realizó en la tabla facturas.

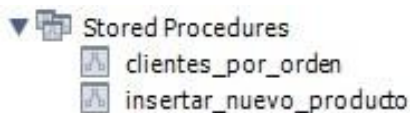


- **total_dinero_recaudado_por_empleado**

La función fue creada para que al poner el id de empleado realice la suma total del dinero recaudado de ventas que realizó ese empleado, se utilizan las tablas ventas y empleados.



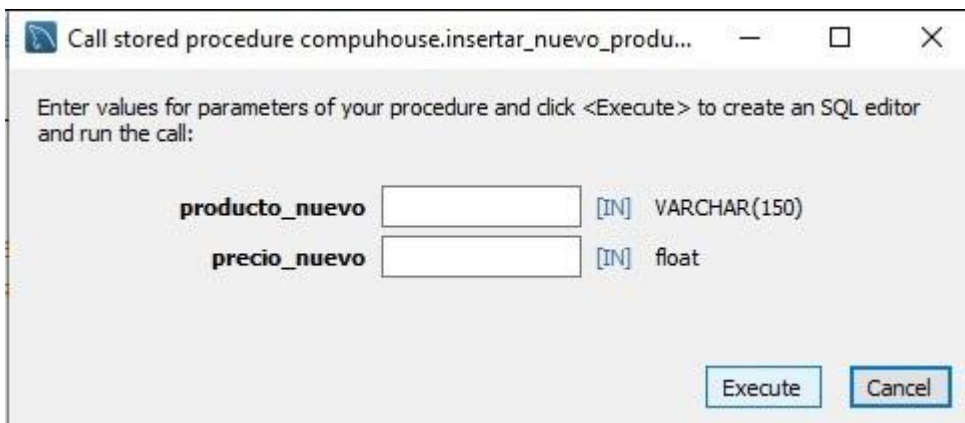
Creación de Stored Procedures



Los stored Procedures creados son los siguientes:

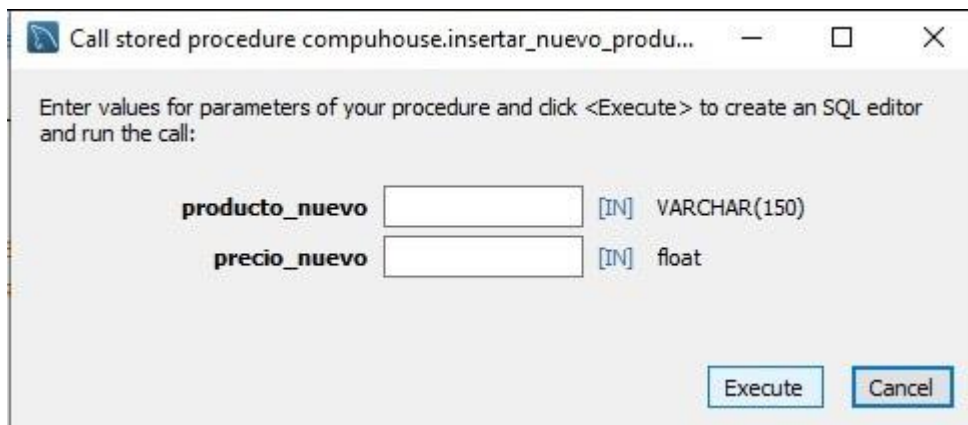
- **insertar_nuevo_producto**

Este stored procedure fue creado para insertar un nombre nuevo de producto y precio en la tabla productos.



• clientes_por_orden

El stored procedure creado se utiliza para que en el primer campo se ingrese el nombre de la columna de clientes y en el segundo en el orden que se desea asc (ascendente) o desc (descendente).



Call stored procedure compuhouse.insertar_nuevo_produ...

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

producto_nuevo [IN] VARCHAR(150)

precio_nuevo [IN] float

Execute Cancel

Creación de Triggers

• clientes_nuevos_tr

El trigger fue creado para guardar en la tabla clientes_nuevos_tr los nuevos datos insertados en la tabla clientes, así también como los registros de: usuario (usuario que inserte el cliente nuevo), fecha_hora (fecha y hora que se inserte el cliente nuevo), db (database que se insertó el dato) y versión (versión que se utilizó).

```
1 • SELECT * FROM compuhouse.clientes_nuevos_tr;
```

	id_cliente	nombre_cliente	apellido_cliente	telefono_cliente	direccion_cliente	email_cliente	usuario	fecha_hora	db	version
▶	1	German	Freigedo	4521-3528	Siempre viva 742	chueko@gmail.com	root@localhost	2023-01-17 16:13:04	compuhouse	8.0.31
	2	Fabrizio	Galizia	4235-4586	av,jujuy 580	fabry@gmail.com	root@localhost	2023-01-17 16:13:04	compuhouse	8.0.31
	3	Daiana	Patao	1165288563	Tokio 2233	Day22@hotmail.com	root@localhost	2023-01-17 16:13:04	compuhouse	8.0.31
	4	Pablo	Segundo	1152364587	Espora 2685	pablo2@gmail.com	root@localhost	2023-01-17 16:13:04	compuhouse	8.0.31

● actualizacion_de_clientes_tr

El trigger fue creado para que al actualizar cualquier dato en la tabla clientes al mismo tiempo quede registrado en la tabla trigger actualizacion_de_clientes_tr en la cual del mismo modo que la anterior están los registros de: usuario (usuario que inserte el cliente nuevo), fecha_hora (fecha y hora que se inserte el cliente nuevo), db (database que se insertó el dato) y versión (versión que se utilizó).

id_cliente	nombre_cliente	apellido_cliente	telefono_cliente	direccion_cliente	email_cliente	<<<< CLIENTE ACTUALIZADO			
1	German	prueba	4521-3528	Siempre viva 742	chueko@gmail.com				

```
1 • SELECT * FROM compuhouse.actualizacion_de_clientes_tr;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	id_cliente	nombre_cliente	apellido_cliente	telefono_cliente	direccion_cliente	email_cliente	usuario	fecha_hora	db	version
▶	1	German	Freijedo	4521-3528	Siempre viva 742	chueko@gmail.com	root@localhost	2023-01-31 15:21:43	compuhouse	8.0.31

● productos_nuevos_tr

El trigger fue creado para guardar en la tabla productos_nuevos_tr los nuevos datos insertados en la tabla productos, así también como los registros de: usuario (usuario que inserte el cliente nuevo), fecha_hora (fecha y hora que se inserte el cliente nuevo), db (database que se insertó el dato) y versión (versión que se utilizó).

```
1 • SELECT * FROM compuhouse.productos_nuevos_tr;
```



Result Grid							
Filter Rows: <input type="text"/>							
Export: Wrap Cell Content:							
	id_producto	nombre_producto	precio_producto	usuario	fecha_hora	db	version
▶	1	Placa de video AMD PowerColor Fighter Radeon...	95000	root@localhost	2023-01-17 16:13:04	compuhouse	8.0.31
	2	Placa de video AMD Gigabyte Gaming Radeon R...	170000	root@localhost	2023-01-17 16:13:04	compuhouse	8.0.31
	3	Placa De Video Powercolor Rx 6600 Fighter 8g ...	130990	root@localhost	2023-01-17 16:13:04	compuhouse	8.0.31
	4	Placa De Video Nvidia Evga Xc Gaming Geforce ...	175054	root@localhost	2023-01-17 16:13:04	compuhouse	8.0.31

● actualizacion_de_productos_tr

El trigger fue creado para que al actualizar cualquier dato en la tabla productos al mismo tiempo quede registrado en la tabla trigger actualizacion_de_productos_tr en la cual del mismo modo que la anterior están los registros de: usuario (usuario que inserte el cliente nuevo), fecha_hora (fecha y hora que se inserte el cliente nuevo), db (database que se insertó el dato) y versión (versión que se utilizó).

id_producto	nombre_producto	precio_producto	<<<< PRODUCTO ACTUALIZADO
1	Placa de video AMD PowerColor Fighter Radeon 6600 Series RX 6600 ...	103020	

1 • `SELECT * FROM compuhouse.actualizacion_de_productos_tr;`

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: IA							
	id_producto	nombre_producto	precio_producto	usuario	fecha_hora	db	version
▶	1	Placa de video AMD PowerColor Fighter Radeon...	95000	root@localhost	2023-01-31 15:28:00	compuhouse	8.0.31

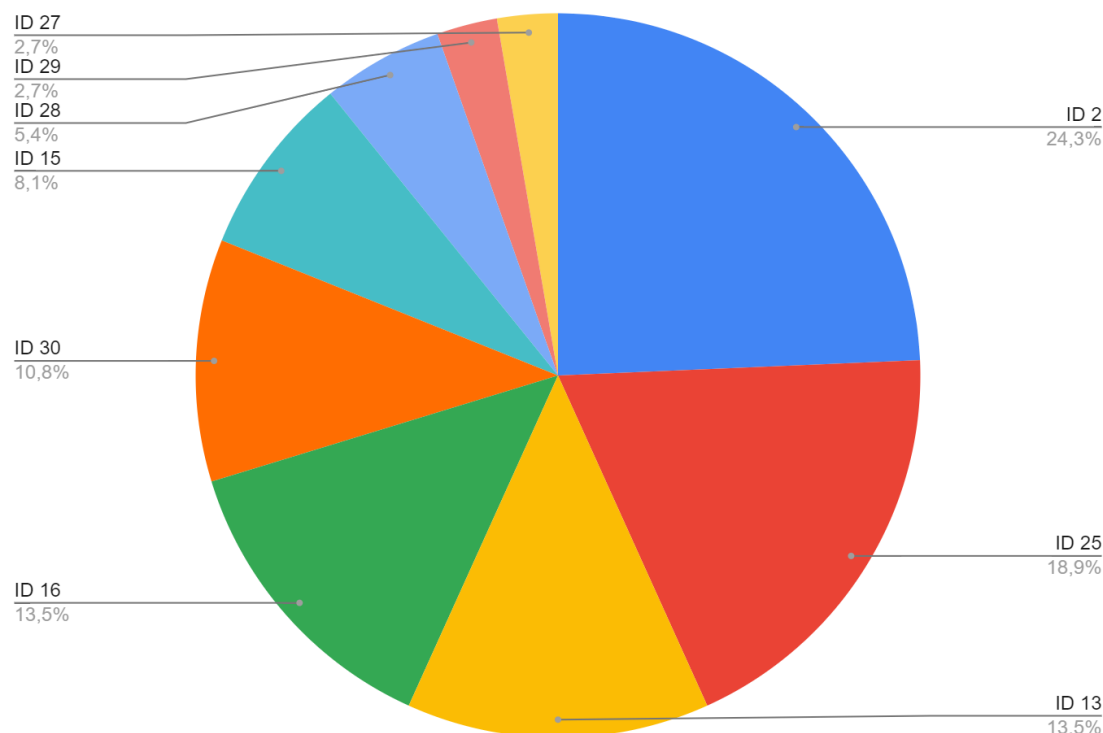
Generación de Outputs y Análisis de Datos

Producto más vendido

De la vista producto_mas_vendio el cual usa las tablas productos y ventas se puede obtener los productos que más se vendieron.

```
1 • SELECT * FROM compuhouse.producto_mas_vendido;
```

	id_producto	nombre_producto	cantidad
▶	2	Placa de video AMD Gigabyte Gaming Radeon RX 5700 Series RX 570...	9
	25	Pendrivel Multilaser Twist PD588 16GB 2.0 negro y plateado	7
	13	Tedado Gaming Cynosa Lite Essential Razer Rz03-02741600	5
	16	Mouse de juego Redragon Cobra Chroma M711 negro	5
	30	Mouse Pad gamer Razer Chroma Goliathus de goma y tela extended ...	4
	15	Mouse de juego Razer DeathAdder Essential negro	3
	28	Disco duro interno Seagate Barracuda ST1000DM010 1TB	2
	29	Disco sólido interno Kingston SA400S37/240G 240GB negro	1
	27	Fan Cooler 120 X120 Cm - Rgb Con Conector A Controlador	1



**El ID 2 es el
producto
más
vendido
con el
24.3%**

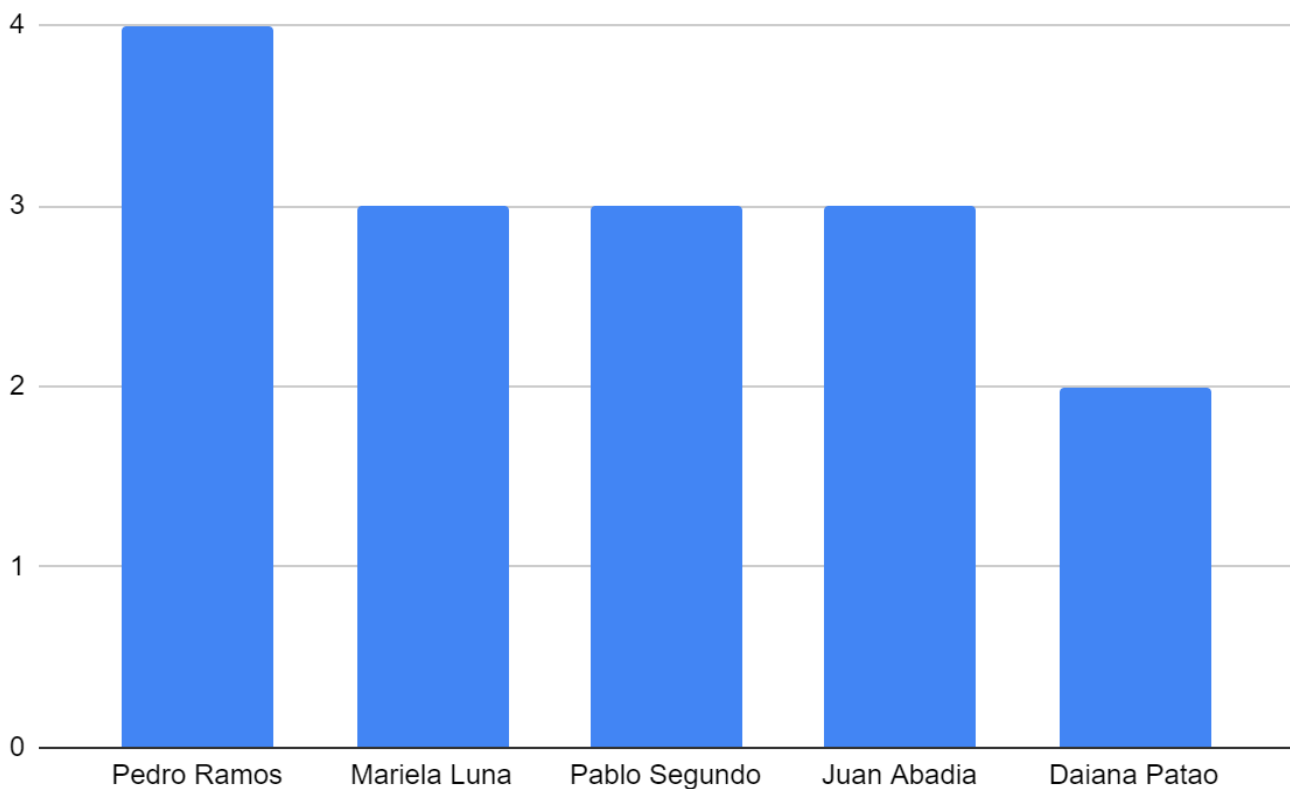
Cliente del mes

Utilizando la vista cliente_con_mas_compras en el cual se utilizan las tablas clientes y facturas para calcular la cantidad de compras que realizó cada cliente.

```
1 • SELECT * FROM compuhouse.cliente_con_mas_compras;
```

Se toma los 5 primeros clientes con más compras.

ID	nombre_cliente	apellido_cliente	cantidad_de_compras
9	Pedro	Ramos	4
15	Mariela	Luna	3
4	Pablo	Segundo	3
5	Juan	Abadia	3
3	Daiana	Patao	2



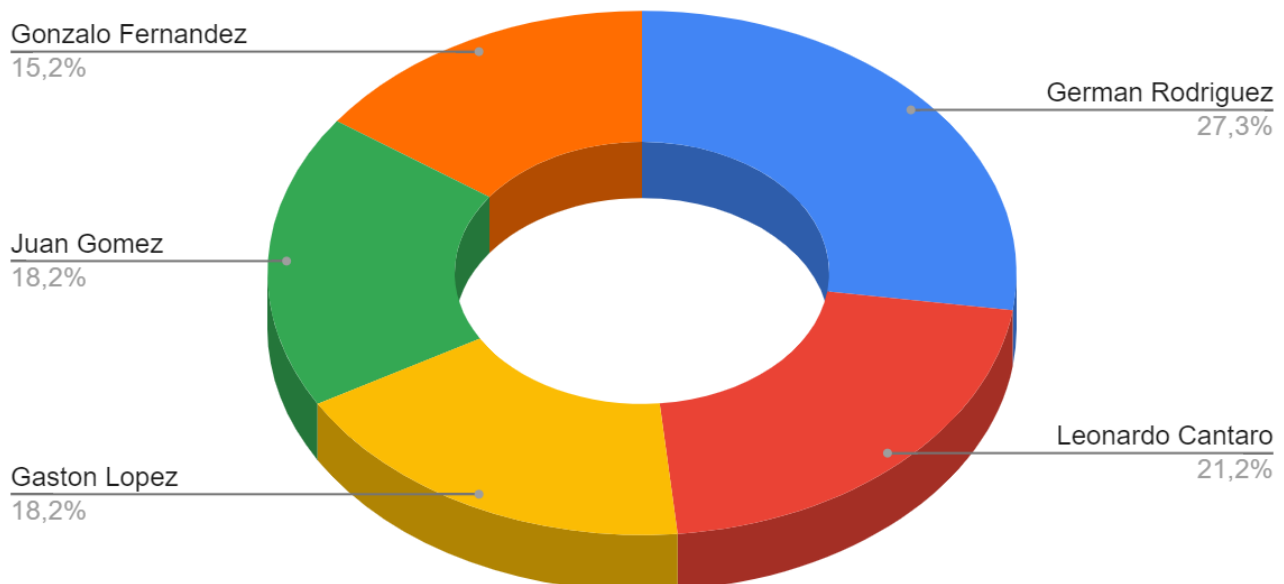
El cliente del mes es Pedro Ramos.

Empleado del mes

Utilizando la vista empleado_con_mas_ventas_realizadas se puede calcular el total de ventas que realizo cada empleado, se utilizan las tablas empleados y ventas.

```
1 • SELECT * FROM compuhouse.empleado_con_mas_ventas_realizadas;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
id_empleado	nombre_empleado	apellido_empleado	cantidad_ventas
3	German	Rodriguez	9
1	Leonardo	Cantaro	7
4	Gaston	Lopez	6
2	Juan	Gomez	6
5	Gonzalo	Fernandez	5



El empleado del mes es German Rodriguez con el 27,3% del total de ventas.

Tecnologías utilizadas

- Microsoft 365: Office Excel y Word
- MySQL Workbench 8.0 CE
- PDF Adobe Acrobat DC

Conclusiones

La base de datos y el diagrama entidad relación (DER) fueron diseñados a medida que se avanzaba y se iban ganando nuevos conocimientos en el curso de SQL de la plataforma Coderhouse.

La base de datos fue modificada varias veces hasta llegar al resultado deseado, fue creada y modificada utilizando el software MySQL Workbench 8.0.

Los datos utilizados en la base de datos fueron extraídos de distintas páginas de venta de hardware de pc, por otro lado se agregaron clientes ficticios para poder poner en funcionamiento las otras tablas y de este modo poder crear la base de datos que contiene: datos, funciones, vistas, stored procedures y triggers.

Gracias a lo aprendido durante el curso entre otros temas las vistas ayudaron a crear gráficos analizando distintos datos que ayuden a progresar en la base de datos y su crecimiento.