

Restaurante La Musleria

Hice esta DB de un restaurante en el cual trabaje hace tiempo. Se llamaba La Musleria y hacíamos muslo de pollo en espadas brasileñas y bañábamos en diferentes tipos de salsas.

En este caso escogí este proyecto por ser algo que más o menos conocía del negocio por estar trabajando en mi adolescencia en ese lugar, había varias problemáticas que uno como empleado se daba cuenta las cuáles serán las que expondré más a delante en este mismo proyecto.

Objetivo

El objetivo de este proyecto es demostrar las principales funciones aprendidas de MySQL en el curso de CoderHouse, demostrando habilidad para la creación de DB, cargar datos, utilizar vistas, store procedures, entre muchas otras.

A pesar que este proyecto es algo hipotético, puede ser utilizado para un mejor control de inventarios, mejoras las ventas y ganancias de un restaurante, entre otras muchas cosas para la ayuda de mejora constante que todo negocio debe de poseer.

Situación Problemática.

El Restaurante al intentar elevar sus ventas implemento un sistema de cliente frecuente, pero es necesario que otros procesos se vean mejorados para brindar un mejor servicio y así el sistema de cliente frecuente se vea efectuado de manera óptima; algunos de las problemáticas se enlistan a continuación:

- Dueño desea llevar un mejor control con proveedores.
- Se dese tener un mejor control en inventarios y stock de productos.
- Se desea implementar un sistema de clientes frecuentes.
- Tener mejores ganancias y control de ellas.

Modelo de negocio

Es un restaurante que su especialidad es vender muslos de pollo en espadas brasileñas combinando sabores norteamericanos, mexicanos y brasileños entre algunos otros, también en el menú hay diferentes platos como hamburguesas conservando la temática de comida fusión.

Menú

El restaurante tiene su platillo estrella que son los muslos y sus salsas y 4 tipos de hamburguesas, dos diferentes postres.

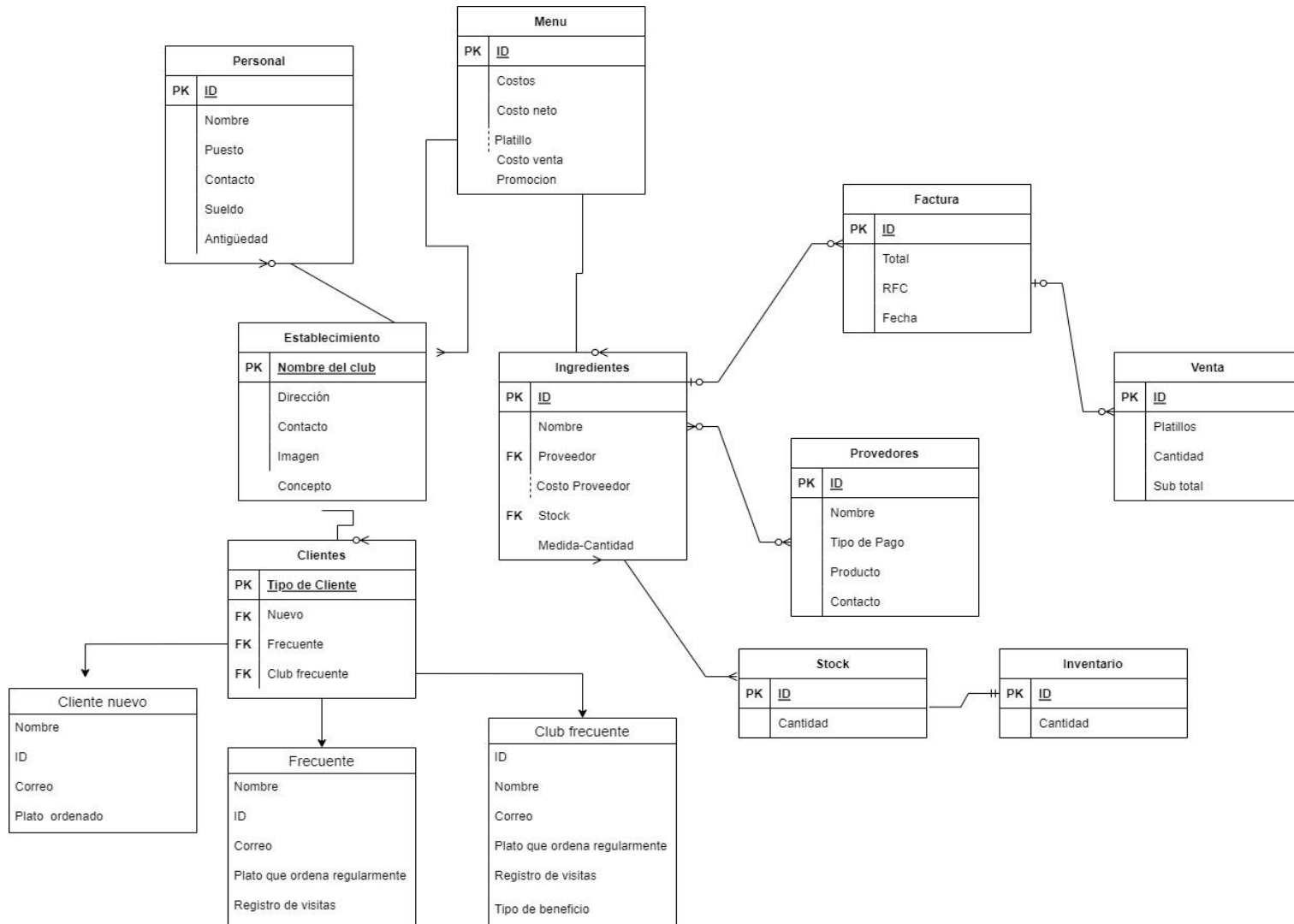
- 10 tipos de salsas

- 4 hamburguesas

- 2 postres

Se trabajará para que no queden mal los proveedores y así siempre tener los platillos del menú e incrementar las ventas del establecimiento.

Diagrama E-R



Tablas

Tabla	ESTABLECIMIENTO						
Descripción	INFO GENERAL DEL LUGAR						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	DEFAULT	AUTO_INCREMENT
PK	NOMBRE	VARCHAR	20	X			
	DIRECCION	VARCHAR	100	X			
	CONTACTO	VARCHAR	100	X			
	CONCEPTO	VARCHAR	150	X			
	IMAGEN	VARCHAR	100	X			

Tabla	PERSONAL						
Descripción	DATOS DEL PERSONAL						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	DEFAULT	AUTO_INCREMENT
PK	ID	INT		X	X		X
	NOMBRE	VARCHAR	50	X			
	CONTACTO	VARCHAR	100	X			
	PUESTO	VARCHAR	20	X			
	SUELDO	INT		X			
	ANTIGÜEDAD	DATESTAMP		X			

Tabla	PROVEDOR						
Descripción	DATOS DE PROVEDORES						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	DEFAULT	AUTO_INCREMENT
PK	ID	INT		X	X		X

NOMBRE	VARCHAR	50	X
TIPO DE PAGO	INT		X
PRODUCTO	VARCHAR	200	X
CONTACTO	VARCHAR	150	X

Tabla		MENU					
Descripción		PLATILLOS DEL LUGAR					
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	DEFAULT	AUTO_INCREMENT
PK	ID	INT		X	X		X
FK	PLATILLO	VARCHAR	50	X			
	COSTO	INT		X			
	PROMOCION	VARCAHR	50	X			
	COSTO NETO	INT		X			

Tabla		INGREDIENTES					
Descripción		INGRENDIENTES QUE LLEVAN LOS PLATILLOS					
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	DEFAULT	AUTO_INCREMENT
PK	ID	INT		X	X		X
FK	PLATILLO	VARCHAR	20	X			
	NOMBRE	VARCHAR	25	X			
	CANTIDAD	VARCHAR	20	X			
	PROVEEDOR	VARCHAR	25	X			
	COSTO						
	PROVEEDOR	VARCHAR	20	X			

Tabla	CLIENTES						
Descripción	DATOS DE LOS CLIENTES QUE NOS VISITAN						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	DEFAULT	AUTO_INCREMENT
PK	TIPO DE CLIENTE	INT		X			
FK	NUEVO	BOOLEAN		X			
FK	FRECUEnte	BOOLEAN		X			
FK	CLUB FRECUEnte	BOOLEAN		X			

Tabla	NUEVO CLIENTE NUEVO						
Descripción	CLIENTE NUEVO						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	DEFAULT	AUTO_INCREMENT
PK	ID	INT		X			X
	NOMBRE	VARCHAR	50	X			
	PLATO ORDENADO	VARCHAR	50	X			
	CORREO	VARCHAR	50	X			

Tabla	FRECUEnte CLIENTE FRECUEnte						
Descripción	CLIENTE FRECUEnte						
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	DEFAULT	AUTO_INCREMENT
PK	ID	INT		X			X
	NOMBRE	VARCHAR	50	X			
	CORREO	VARCHAR	50	X			

	REGISTRO DE VISITAS	VARCHAR	50	X
	PLATILLO QUE ORDENA	VARCHAR	20	X

Tabla	CLUB FRECUENTE				
Descripción	CLIENTE FRECUENTE, PERO CON MEMEBRESIA				
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE DEFAULT AUTO_INCREMENT
PK	ID	INT		X	X
	NOMBRE	VARCHAR	50	X	
	CORREO PLATILLO QUE ORDENA	VARCHAR	50	X	
	REGISTO DE VISITAS	VARCHAR	50	X	
	BENEFICIO	VARCHAR	50	X	

Tabla	FACTURA				
Descripción	CUENTAS DE EL RESTAURANTE				
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE DEFAULT AUTO_INCREMENT
PK	ID	INT		X	X
	TOTAL	INT		X	
	RFC	VARCHAR	15	X	
	FECHA	TIMESTAMP		X	

Tabla	VENTA				
-------	-------	--	--	--	--

Descripción		VENTAS DE EL RESTAURANTE					
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	DEFAULT	AUTO_INCREMENT
PK	ID	INT		X			X
	PLATILLOS	VARCHAR	60	X			
	CANTIDAD	INT		X			
	SUB TOTAL	INT		X			

Tabla		STOCK					
Descripción		STOCK DE EL RESTAURANTE					
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	DEFAULT	AUTO_INCREMENT
PK	ID	INT		X			X
	CANTIDAD	INT		X			

Tabla		INVENTARIO					
Descripción		INVENTARIO DE EL RESTAURANTE					
INDEX	COLUMN	TYPE	LEN	NOT NULL	UNIQUE	DEFAULT	AUTO_INCREMENT
PK	ID	INT		X			X
	CANTIDAD	INT		X			

Script de creación y Scripts de inserción de datos

Estos documentos se encuentran en la carpeta compartida de OneDrive anexo link:

https://drive.google.com/drive/folders/19UwsEwWxFiFHi9A7yAq_x-zyk--LfhwC?usp=sharing

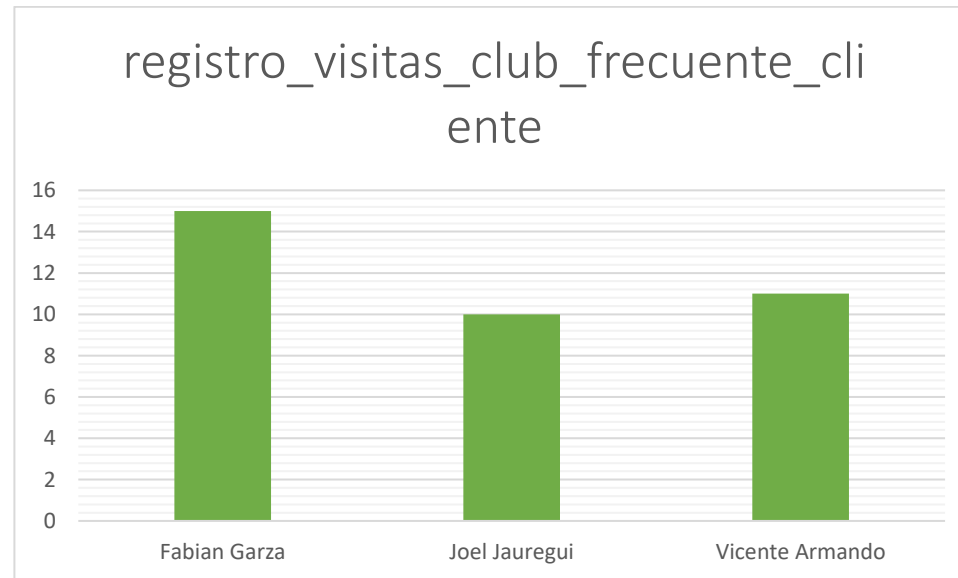
Informes generados con la base de datos.

Vistas

- En las vistas son vistas para los empleados del restaurante puedan ver información de los empleados y clientes sin datos sensibles y de estas maneras poder aplicar las promociones correspondientes a los clientes y/o contactar algún empleado.

-La primera vista llamada "cliente_con_mas_visitas" es solo para observar que cliente es el que tiene más visitas y esta vista la obtenemos usando un MAX para obtener el valor de más alto valor y nos muestre a nuestro cliente con más visitas y poder otorgarle descuentos o productos gratis y así seguir incitando a que nos siga visitando.

	nombre_dub_frecuente_cliente	registro_visitas_dub_frecuente_cliente
►	Fabian Garza	15



-La segunda vista son las facturas que son mayores a mil pesos ya que por ley arriba de esa cantidad es necesario reportar la venta al SAT (hacienda en México) esta vista hace un barrido en la tabla facturas y cuando en "total_factura" detecta que es mayor igual a \$1,000.00 pesos lo manda a esa vista, que seria el gerente de revisarla cada día y dar notificación a autoridades.

	id_factura	total_factura	rfc_factura
►	4	4522	YUJN890514ETF
	5	5632	CVGD0010155HE

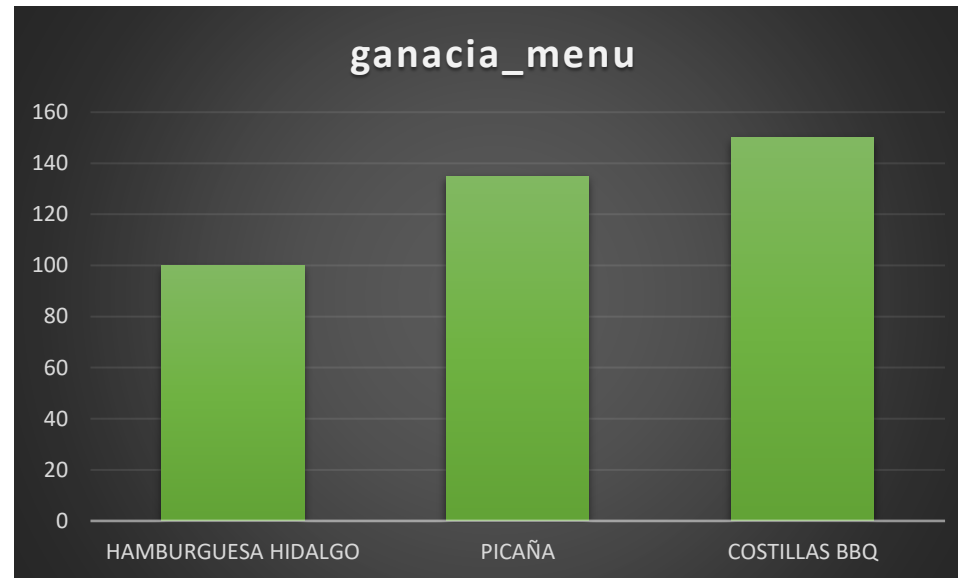
-La tercera vista es para observar información del personal sin sueldo y no comprometer su información, esta tendría utilidad para revisar puestos y gente contratada para el puesto y si fuese necesario hacer

alguna modificación en la descripción de el mismo, esta se obtiene solo pidiendo a la información de la tabla personal sin los sueldos.

	id_personal	nombre_personal	puesto_personal
	1	Frida Corona	Gerente General
	2	Alberto Garza	Dueño
	3	Alma Ayala	Jefe de cocina
	4	Abril Ayala	Cocinera
	5	Rolando Guerrero	Mesero
	6	Hugo Alfaro	Mesero

-La cuarta vista son los platillos que mas ganancia dejan por el momento al restaurante y que los meseros traten de ofrecerlos, esto varia desde por que se compró en promoción algún ingrediente o simplemente es que se tiene una mejor ganancia de el mismo, aquí lo obtenemos teniendo una ganancia mayos de o igual a \$100 pesos y

	platillo_menu	costo_menu	ganacia_menu
	HAMBURGUESA HIDALGO	140	100
	COSTILLAS BBQ	190	150
▶	PICAÑA	165	135



-La quinta vista es con los proveedores que solo aceptan efectivo y no manejan ningún tipo de crédito o concesión con el restaurante, mismos que cuando surtan será necesario tener efectivo en la caja chica y revisar algún trato para conseguir crédito o concesiones y de esa manera tener un flojo óptimo de efectivo, está la obtenemos al solo comprar si es true o false en la tabla de proveedor en la columna de tipo de pago.

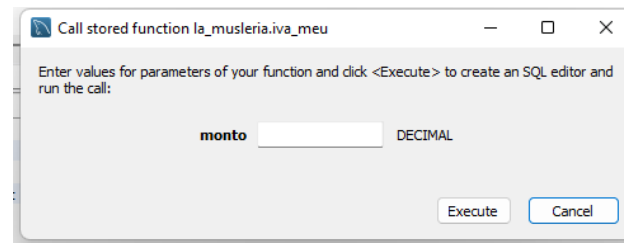
	id_proveedor	nombre_proveedor	contacto_proveedor	tipo_de_pago_proveedor
▶	1	Los pollos hermanos	7894561230	1
	3	Desechable Lopes	6543219877	1
	4	Aderezos buffalo	7893214560	1
	6	Cerveceria Cuauhtemoc	1236547899	1
	152	salsas gonzalez	8225522445	1

Se presentaría así al dueño para que los tuviera en consideración

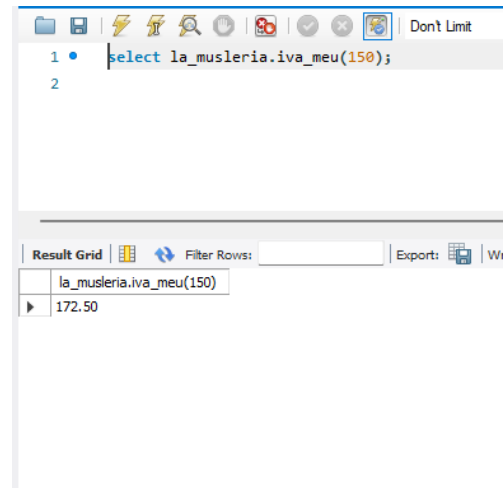
Proveedores a conseguir trato
Salsas González
Los pollos hermanos
Desechable López
Cervecería Cuauhtémoc
Aderezos buffalo

Funciones

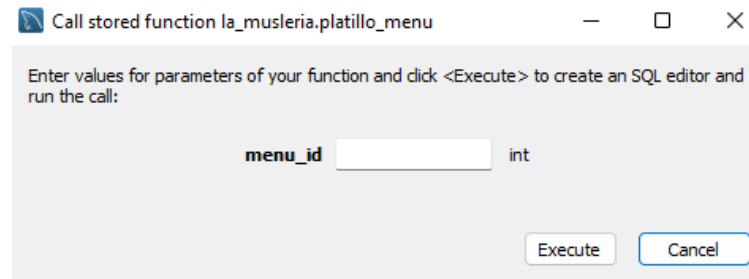
- En las funciones se utilizaron dos funciones sencillas una para obtener el IVA de los productos vendidos usando el 16% como se maneja acá en México, al obtener ya los costos de el platillo y el costo con el cual se estaría vendiendo antes de ponerlo en el menú ponemos el costo total y nos arroja el costo ya con IVA y de esta manera tener un correcto costo de IVA.



Como ejemplo ponemos 150 pesos y nos arroja ya el total de 172.50 que ya sería el precio que se publicaría en menú



- En la segunda función fue para con solo poner el id de los productos arroje que platillo es que se está ordenando.



Esta implementación seria para poder pasar los platillos ordenandos a cocina y ellos pueden ver que platillos están entrando a comanda y puedan ir preparándolos en el orden que los meseros introduzcan los platillos.

```
personal_sin_sureos personal_sin_sureos - view pri
1 • select la_musleria.platillo_menu(2);
2
```

Result Grid | Filter Rows: | Export

la_musleria.platillo_menu(2)
HAMBURGUESA HIDALGO

Stored procedures

- En los stored procedures hicimos los dos que pedían en consignas anteriores el ordenar de manera ascendente o descendente una lista de la bd por medio de un booleano en este caso decidí utilizar la tabla de menú y al introducir el nombre de la columna escoge esta y ya solo es necesario agregar 1 o 0 dependiendo de cómo quieras que aparezca si de forma ascendente o descendente, muestro en un ejemplo a continuación.

Call stored procedure la_musleria.sp_ordeN

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

field	<input type="text"/>	[IN] VARCHAR(20)
orden	<input type="text"/>	[IN] BOOLEAN

Execute Cancel

```
1 • call la_musleria.sp_ordeN('costo_menu', 1);
2
```

Result Grid					
Filter Rows:		Export:		Wrap Cell Content:	
	id_menu	plato_menu	costo_menu	promocion_menu	ganadia_menu
▶	10	REFresco	30	1	15
	9	CERVEZA	40	1	25
	8	PAY DE MANZANA	65	0	45
	6	ESPARAGOS CON TOCINO	70	0	40
	1	MUSLOS	100	1	80
	3	HAMBURGUESA JOYA	120	0	90
	4	HAMBURGUESA CALLEJERA	120	0	90
	2	HAMBURGUESA HIDALGO	140	0	100
	7	PICAÑA	165	0	135

Y se ordenó de menor costo del menú al mayo costo de el mismo.

En el segundo para agregar un dato en alguna tabla que en este caso de utilizo para cuando nos visitara un cliente nuevo al restaurante darlo de alta en la bd y asi irle aplicando promociones dependiendo de las veces que nos visite.

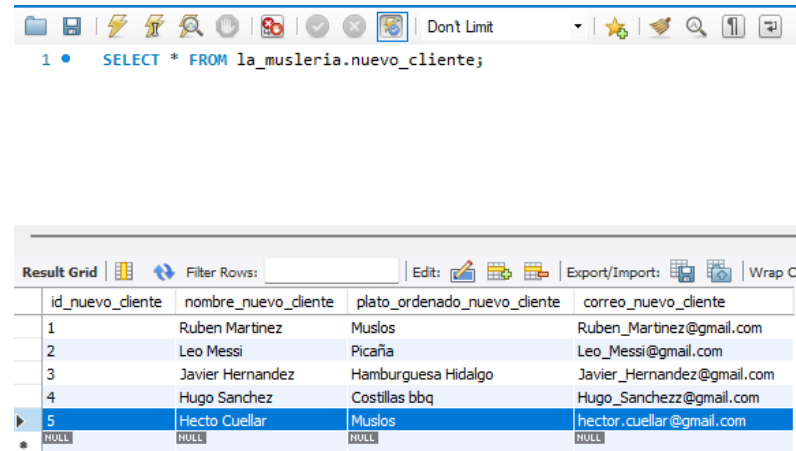
Call stored procedure la_musleria.insert_new_clientE

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

nombre_nuevo_cliente	Hecto Cuellar	[IN]	VARCHAR(50)
plato_ordenado_nuevo_cliente	Muslos	[IN]	VARCHAR(30)
correo_nuevo_cliente	hector.cuellar@gmail	[IN]	VARCHAR(50)

Execute Cancel

Al ejecutar el stored se agrega el nuevo cliente a nuestra base de datos como se muestra a continuación.



The screenshot shows a SQL client window with a toolbar at the top. Below the toolbar, a query is entered: `1 • SELECT * FROM la_musleria.nuevo_cliente;`. Below the query, a 'Result Grid' displays the results of the query. The grid has four columns: `id_nuevo_cliente`, `nombre_nuevo_cliente`, `plato_ordenado_nuevo_cliente`, and `correo_nuevo_cliente`. The results are as follows:

id_nuevo_cliente	nombre_nuevo_cliente	plato_ordenado_nuevo_cliente	correo_nuevo_cliente
1	Ruben Martinez	Muslos	Ruben_Martinez@gmail.com
2	Leo Messi	Picaña	Leo_Messi@gmail.com
3	Javier Hernandez	Hamburguesa Hidalgo	Javier_Hernandez@gmail.com
4	Hugo Sanchez	Costillas bbq	Hugo_Sanchezz@gmail.com
5	Hecto Cuellar	Muslos	hector.cuellar@gmail.com
* NULL	NULL	NULL	NULL

Trigger

- En el trigger estamos agregando un nuevo proveedor con sus datos de contacto y pues como se solicitó se pone el registro de hora usuario y versión de esta manera saber cuándo se dio de alta y por quien se dio; y así llevar un llenado correcto y eficaz de la base de datos.

```
CREATE TRIGGER `tr_add_new_proveedor` AFTER INSERT ON `proveedor` FOR EACH ROW BEGIN
    insert into `la_musleria`.`new_proveedor`(id_proveedor, nombre_proveedor, contacto_proveedor,
    producto_proveedor, usuario, registro, db, version)
    values (new.id_proveedor, new.nombre_proveedor, new.contacto_proveedor, new.producto_proveedor,
    user(), current_timestamp(), database(), version())
END $$
```

```
DELIMITER ;
;
create table new_proveedor(
```

```
    id_proveedor int primary key,  
    nombre_proveedor varchar(100),  
    contacto_proveedor varchar(100),  
    producto_proveedor varchar(100),  
    usuario varchar(200),  
    registro timestamp,  
    db varchar (200),  
    version varchar (100)  
);  
  
insert into proveedor (id_proveedor,  
nombre_proveedor, contacto_proveedor, tipo_de_pago_proveedor, producto_proveedor)  
values (152, 'salsas gonzalez', '8225522445', '1', 'salsas');
```

Tecnologías usadas.

- MySQL
- GitHub (<https://github.com/Alberto9204/SQL>)
- Google Drive
- Word
- Stack Overflow