

## SQL – Pre entrega 2 – Santiago Duré

### DESCRIPCION DE LA BASE DE DATOS:

#### supermercado\_coderhouse

Se trata de la base de datos de un supermercado tanto mayorista, como minorista, que albergará toda la estructura organizacional del negocio, partiendo desde el árbol jerárquico, hasta la compra de productos y la venta de los mismos, sea a otros comerciantes como a consumidores finales.

INFORMACIÓN IMPORTANTE: Esta base de datos está pensada actualmente para funcionar con una sola sucursal cuyo id SIEMPRE será 1, nadie estará autorizado en un futuro a modificar esto, sin embargo, en caso de escalar, solo con un par de modificaciones podría utilizarse para múltiples sucursales.

### LISTADO DE TABLAS:

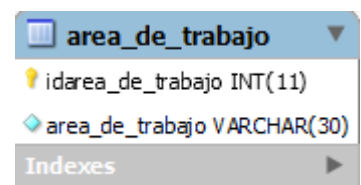
- area\_de\_trabajo
- cargo
- producto
- categoria\_producto
- cliente
- empleado
- proveedor
- sucursal
- venta
- tipo\_de\_venta
- línea\_de\_producto
- nacionalidad

DER:

Ver en <https://github.com/SantiDure/SQL-Pre-entrega-2-Santiago-Dure>

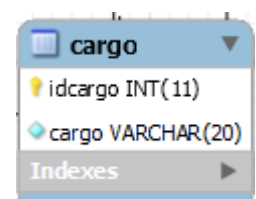
### Tablas:

**area\_de\_trabajo:** Define las diferentes áreas que tiene el supermercado, su pk servirá como fk de las tablas empleado (para saber en qué sector le corresponde trabajar) y proveedor (para saber a qué área corresponden los productos que ingrese ese proveedor al supermercado).



area_de_trabajo	
idarea_de_trabajo	INT(11)
area_de_trabajo	VARCHAR(30)
Indexes	

**cargo:** se relaciona con la tabla empleado, que ya obtiene su area de trabajo con la tabla anterior, ahora además, obtendrá su cargo, definido más tarde por el árbol jerárquico del supermercado.



cargo	
idcargo	INT(11)
cargo	VARCHAR(20)
Indexes	

**producto:** contiene información útil para el inventario, como precio y marca, además, una fk con la información del proveedor que trajo ese producto al supermercado, y otra, con su respectiva categoría (ej. Limpieza, comida, etc). Se agrego un campo para registrar el ultimo momento en el que fue actualizado el producto, esto, será registrado mediante el disparo de un trigger.

producto	
idproducto	INT(11)
precio	VARCHAR(45)
marca	VARCHAR(45)
fk_idCategoriaProducto	INT(11)
fk_idProveedor	INT(11)
nombre	VARCHAR(45)
stock	INT(11)
ultima_actualizacion	DATETIME
Indexes	
Triggers	

**categoria\_producto:** almacenará las categorías mencionadas anteriormente.

categoria_producto	
idcategoria_producto	INT(11)
categoria	VARCHAR(45)
Indexes	

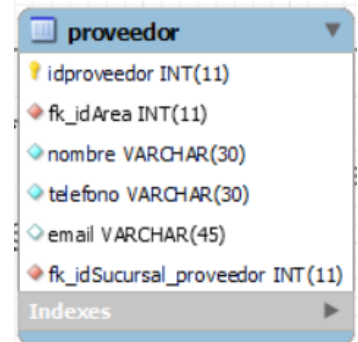
**cliente:** contiene información útil del cliente que realizo al menos 1 compra en la sucursal 1 (por el momento, será la unica sucursal en funcionamiento).

cliente	
idcliente	INT(11)
nombre	VARCHAR(45)
apellido	VARCHAR(45)
nacionalidad	VARCHAR(45)
dni	VARCHAR(45)
pasaporte	VARCHAR(45)
fk_idSucursal	INT(11)
Indexes	

**empleado:** contiene información de un empleado de la sucursal 1(para eso esta la fk id sucursal a su respectiva tabla), como se mencionó antes, también contiene las fk para su cargo, su área de trabajo y su nacionalidad, y además, su información personal, que fue solicitada al momento de su contratación.

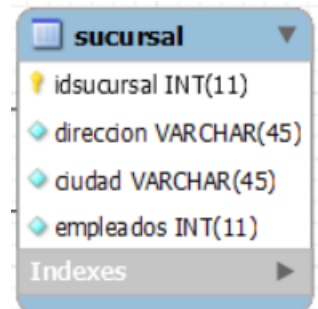
empleado	
idempleado	INT(11)
nombre	VARCHAR(45)
apellido	VARCHAR(45)
fk_nacionalidad	INT(11)
dni	VARCHAR(45)
pasaporte	VARCHAR(45)
fk_idCargo	INT(11)
fk_idAreaDeTrabajo	INT(11)
fk_idSucursal_empleado	INT(11)
Indexes	
Triggers	

**proveedor:** contiene información de contacto de dicho proveedor, además, una fk que apuntará a la tabla area\_de\_trabajo, para saber a que área pertenecen los productos que traerá este proveedor, y otra fk que apunta a la sucursal a la que abastece dicho proveedor (solo funciona para la sucursal 1).



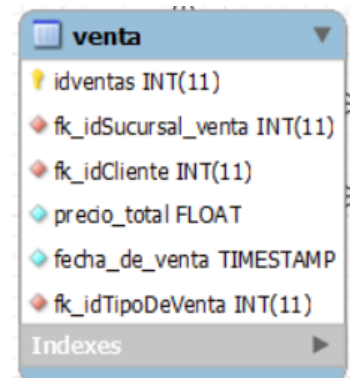
proveedor	
idproveedor	INT(11)
fk_idArea	INT(11)
nombre	VARCHAR(30)
telefono	VARCHAR(30)
email	VARCHAR(45)
fk_idSucursal_proveedor	INT(11)
Indexes	

**sucursal (1):** contiene información sobre la ubicación exacta de dicha sucursal (1), la cantidad de empleados.



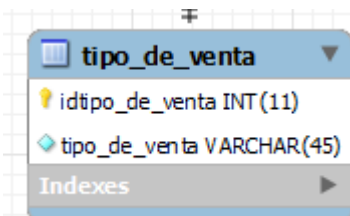
sucursal	
idsucursal	INT(11)
direccion	VARCHAR(45)
ciudad	VARCHAR(45)
empleados	INT(11)
Indexes	

**venta:** contiene información de la sucursal (1), el cliente que realizo la compra, la fecha de la misma, el precio total que deberá pagar el cliente, y una fk, que apunta a una tabla tipo\_de\_venta, definiendo asi si se trata de una venta mayorista o una venta minorista.



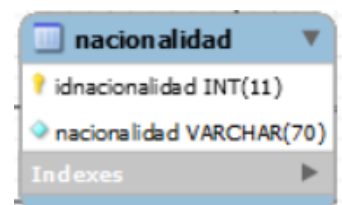
venta	
idventas	INT(11)
fk_idSucursal_venta	INT(11)
fk_idCliente	INT(11)
precio_total	FLOAT
fecha_de_venta	TIMESTAMP
fk_idTipoDeVenta	INT(11)
Indexes	

**tipo\_de\_venta:** identificará una venta como mayorista o minorista, según corresponda.




tipo_de_venta	
idtipo_de_venta	INT(11)
tipo_de_venta	VARCHAR(45)
Indexes	

**nacionalidad:** contiene una lista de nacionalidades, servirá para proveer a la tabla empleado, mediante su respectiva fk, de informacion sobre la nacionalidad de dicho empleado.



nacionalidad	
idnacionalidad	INT(11)
nacionalidad	VARCHAR(70)
Indexes	

**linea\_de\_producto:** esta tabla se abastecerá de datos, en un futuro, en el momento en el que se realice una venta, la idea es que, a medida que se hacen ventas, se almacenen los productos con su respectivo precio unitario y cantidad, relacionados con el id de su respectiva venta, de esta forma, si hay que generar un ticket, se extraerá la información de esta tabla.



linea_de_producto	
idlinea_de_producto	INT(11)
fk_idProducto	INT(11)
cantidad	INT(11)
fk_idVenta	INT(11)
precio_unitario	INT(11)
Indexes	
Triggers	

## Views:

### productos\_sin\_stock:

Devuelve una tabla con los productos cuyo stock es 0.

	idproducto	producto_sin_stock
▶	1	Vinegar - Rice

### productos\_con\_poco\_stock:

Devuelve una tabla con productos cuyo stock esta entre 1 y 10.

	idproducto	nombre	stock
▶	2	Wine - Two Oceans Sauvignon	2
	3	Dried Apple	3
	4	Beef - Top Sirloin - Aaa	4
	5	Coffee - Decaffeinato Coffee	5
	6	Steampan - Half Size Shallow	6
	7	Sugar - Palm	7

### total\_gastado\_por\_cliente:

Devuelve una tabla con la información de todos los clientes registrados y cuanto dinero en total han gastado en la sucursal (solo sucursal 1).

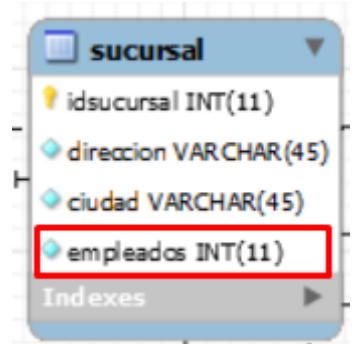
	idcliente	nombre	apellido	total_comprado
▶	1	Reid	Pride	110
	4	Yardley	Westoll	1000

## Triggers:

### - Tabla empleado:

#### empleado\_BEFORE\_INSERT y empleado\_BEFORE\_DELETE:

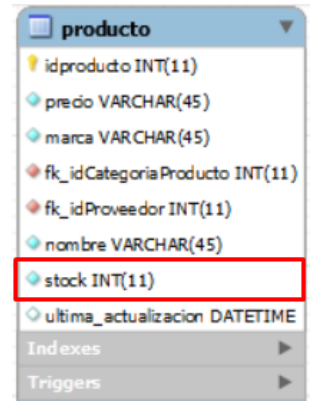
Se encarga de modificar el valor de un campo de la tabla sucursal, que almacena la cantidad de empleados que tiene la sucursal.



### - Tabla linea\_de\_producto:

#### linea\_de\_producto\_AFTER\_INSERT:

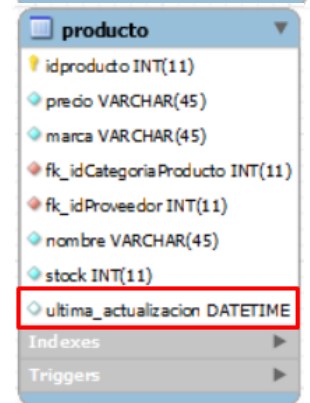
Una vez que se realiza una venta y se agregan los productos con su cantidad a la tabla linea\_de\_producto (el trigger que se encargará de esto aún no está hecho, puede probarse agregando productos a la tabla linea\_de\_producto), este trigger se encargará de actualizar el campo “stock” de la tabla producto buscado según si id.



### - Tabla producto:

#### registrar\_actualizacion\_precio:

Se encarga de actualizar el campo “ultima\_actualizacion” de la misma tabla producto, cada vez que se actualiza el precio de un producto.



## Funciones:

**calcular\_descuento(precio, descuento):** recibe 2 parámetros, “precio” es el monto original por unidad que cuesta el producto, y “descuento” es el porcentaje a descontar de dicho producto.

**nombre\_completo\_cliente(idcliente):** recibe el id del cliente y devuelve su nombre completo.

**total\_gastado\_cliente(idcliente):** recibe le id del cliente y devuelte el total que dicho cliente se ha gastado en la sucursal.

### **Stored Procedures:**

**actualizar\_precio\_producto (idproducto, nuevo\_precio):** recibe el id del producto a actualizar y su nuevo precio (en este punto, al actualizar el precio, se disparará el trigger "registrar\_actualizacion\_precio").

**agregar\_categoria\_de\_producto(categoria):** registra en la base de datos una nueva categoría de producto.

**agregar\_producto(nombre\*, marca, precio, stock, categoría\*, proveedor\*):** crea un registro en su respectiva tabla con la información del nuevo producto. (los parámetros señalados con un asterisco sin NOT NULL y no pueden faltar al momento de agregar un nuevo producto).