



PROYECTO FINAL

EL GRAN ABARROTERO

Profesor:

M. en I. Gerardo Avilés Rosas

Equipo Alpha

Integrantes:

Fernanda Ameyalli Osorio Morales

Diego Castro Rendón

Ulises Josué Anaya Pérez

Victor Federico Torres Trejo

José Luis López Bautista

CONSIDERACIONES DE DISEÑO AL MOMENTO DE GENERAR EL MODELO ENTIDAD-RELACIÓN Y RELACIONAL

Entidades

Debido a que en el caso de uso se nos pide almacenar información sobre empleados y clientes, consideramos más conveniente hacer una entidad **persona** desde esta hacemos herencia disjunta a las entidades cliente y empleado, ya que decimos que los empleados no puede ser un cliente, su participación es total porque las personas que nos interesan son los empleados y clientes.

La entidad **empleado** tiene los atributos: salario, escolaridad y fecha de contratación. Para identificar a cada empleado tenemos como atributo identificador "IdPersona", se optó por una identificador generado ya que no se tiene suficientes datos para generar el CURP y no podemos usar otro atributo que nos garantice identificar de forma única cada instancia de entidad.

En la entidad empleado también hacemos uso de herencia ya que tenemos tres tipos de empleados **cajero**, **encargado** y **gerente**, se necesita participación total ya que las instancias de empleado necesariamente tienen que ser alguno de esos tipos de empleado, hacemos uso de una herencia disjunta porque un empleado solo puede ser uno de los tres tipos. Respecto a la entidad **cliente** que está únicamente tiene los atributos de la entidad padre persona.

En cuanto a la entidad **producto** de nuevo optamos por una llave generada, porque con los datos solicitados en el caso de uso no contamos con ningún atributo que los identifique de forma única. Y se prefirió la llave generada, en lugar de buscar agregar un atributo para los productos que lo identificara de forma única pero no apartará información, por lo que solo se tiene como atributos nombre, marca y precio. Tenemos herencia total y disjunta porque los productos deben pertenecer a abarroto o electrónico.

Para la entidad **abarroto** se tiene también herencia total disjunta, para los productos **perecederos** y **no perecederos** ambos tiene los atributos de presentación y fecha de preparación, únicamente los perecederos tiene el atributo de fecha de caducidad. Por otro lado para **electrónica** se tiene descripción, categoría y consumo eléctrico.

En cuanto a **sucursal** de forma similar a las entidades anteriores, decidimos usar una llave generada, porque las llaves compuestas que pensamos, como una compuesta por nombre y fecha o nombre y los atributos de la dirección. Serían muy grandes por lo que se prefirió optar por la generación de las llaves con números enteros positivos.

Creamos la entidad **venta** a partir del problema de que necesitábamos almacenar la información de las ventas realizadas en la sucursal, necesitábamos almacenar en que sucursal se realizó la venta. Únicamente tenemos un atributo el cual es la llave primaria, la cual de nuevo es genérica.

Relaciones

Trabajar: la entidad empleado tiene una relación con la entidad sucursal, ya que los empleados necesitan trabajar en alguna sucursal, tenemos participación es total de ambos lados, porque un empleado necesariamente tiene que trabajar en una sucursal y en una sucursal necesariamente trabajan empleados. La cardinalidad es 1:N, porque en una sucursal pueden trabajar muchos empleados, tenemos la cardinalidad "uno" del lado de sucursal y "muchos" del lado de empleado y no contamos con atributo de relación.

Hacer: La entidad cajero está relacionada con venta, porque en nuestro modelo los cajeros serán los encargados de hacer las ventas. La participación de la relación es parcial del lado de cajero, ya que un cajero puede no haber hecho alguna venta, en cambio la participación es total del lado de ventas, porque una venta necesariamente tiene que ser hecha por un cajero. En cuanto a la cardinalidad es 1:N, porque un cajero puede cobrar muchas ventas pero una venta solo está cobrada por un cajero.

Recibir: La entidad venta está relacionada con cliente, con cardinalidad 1:N, porque un cliente hace muchas compras pero una compra solo puede estar asociada a un cliente, y es total de ambos lados. El atributo que tiene la relación es la forma de pago para tener registrado la forma en la que se realizó el pago en cada una de las ventas.

Realizar: La entidad venta está relacionada con sucursal, la cardinalidad es 1:N, porque una sucursal puede realizar muchas ventas, pero una venta solo puede ser realizada por una sucursal. La participación total de venta, ya que estas siempre son hechas por una sucursal en cambio decidimos hacerla parcial del lado de sucursal, debido a que una sucursal pudo no haber vendido nada, no tenemos atributo de relación.

Contener: Entre Venta y Producto, para poder tener un registro de lo que se está comprando, así como poder calcular el costo total de la venta usando como atributos tendremos la cantidad de productos que se compraron y el precio total. La cardinalidad es N:M puesto que cada producto podrá estar asociado varias venta y una venta podrá tener muchos productos. Un producto no tiene que estar contenido en una venta pero si una venta tiene que contener productos.

Contar: Entre Sucursal y Producto con el atributo Cantidad, para poder manejar el número de productos que se tenga en Stock, la cardinalidad es N:M puesto que una sucursal cuenta con muchos productos y muchos productos están en muchas sucursales. La participación es parcial del lado de Producto puesto que no podremos encontrar todas las instancias de producto en todas las sucursales, y total del lado de Sucursal porque no puede existir una sucursal sin productos.

DEPENDENCIAS FUNCIONALES

- cliente: (idPersona→ nombre, Apaterno, Amaterno, FechaNacimiento, ciudad, Cp, calle, numero, genero)
- contenerElectronico: (idVenta,idProducto→cantidad)
- contenerPerecedero: (idVenta,idProducto→cantidad)
- contenerNoPerecedero: (idVenta,idProducto→cantidad)
- sucursal: (idSucursal → nombre, municipio .cp ,ciudad ,calle ,numero.colonia.fechaApertura)
- perecedero: (idProducto→ nombre, marca, precio, presentacion, fechaPreparacion, fechaCaducidad)
- noPerecedero: (idProducto→ nombre, marca, precio, presentacion, fechaPreparacion)
- electronico: (idProducto→ nombre, marca, precio, descripcion, categoria, consumoElectrico)
- gerente: (idPersona → nombre, idSucursal, Apaterno, Amaterno, FechaNacimiento, ciudad, Cp, calle, numero, genero, salario, fechaContratacion, Escolaridad)
- cajero: (idPersona → nombre, idSucursal, Apaterno, Amaterno, FechaNacimiento, ciudad, Cp, calle, numero, genero, salario, fechaContratacion, Escolaridad)
- encargado: (idPersona → nombre, idSucursal, Apaterno, Amaterno, FechaNacimiento, ciudad, Cp, calle, numero, genero, salario, fechaContratacion, Escolaridad)
- telefonoSucursal: (idSucursal, telefono→idSucursal, telefono)
- telefonoCliente: (idPersona, telefono→idPersona, telefono)
- telefonoCajero: (idPersona, telefono→idPersona, telefono)
- telefonoEncargado: (idPersona, telefono→idPersona, telefono)
- telefonoGerente: (idPersona, telefono→idPersona, telefono)
- correoCliente: (idPersona, correo→idPersona, correo)
- correoCajero: (idPersona, correo→idPersona, correo)
- correoEncargado: (idPersona, correo→idPersona, correo)
- correoGerente: (idPersona, correo→idPersona, correo)
- contarElectronico: (idSucursal,idProducto → cantidad)
- contarPerecedero: (idSucursal,idProducto → cantidad)
- contarNoPerecedero: (idSucursal,idProducto → cantidad)
- venta: (idVenta →idSucursal, idCliente, idCajero, formaPago)

CONSIDERACIONES AL MOMENTO DE GENERAR EL DDL

Siguiendo la lógica que ya se había planteado en el Modelo E/R y posteriormente en el modelo Relacional, al momento de generar el esquema de nuestra base de datos se realizó la definición de tabla por tabla, prestando especial atención al tipo de dato que contendría cada columna.

Asimismo, dentro de cada una de dichas tablas incluimos una sección con restricciones de dominio en donde nos aseguramos que las cadenas no fueran vacías, las llaves fueran únicas y SET NOT NULL para aquellos atributos que de ninguna manera podían quedarse sin datos; las restricciones que consideramos más importantes fueron a las que les otorgamos un nombre, para que en caso de que posteriormente se requiriera modificar algo en ellas se pudieran encontrar fácilmente.



CONSIDERACIONES AL MOMENTO DE POBLAR LAS TABLAS

Para la realización de esta práctica principalmente se usó la herramienta Monckaroo, ya que esta herramienta nos ayuda a generar los datos necesarios para poblar las tablas de nuestra base de datos, para evitar problemas primero en Monckaroo generamos archivos CSV para después pasarlos a formato SQL en este link (<https://www.convertcsv.com/csv-to-sql.htm>) y de ahí poder juntar los INSERT's en un archivo DML.

En Monckaroo se generaron los datos para cada una de nuestras tablas de acuerdo a lo siguiente:

En la **tabla sucursal** usamos los tipos Row number, Custom list para los nombres de sucursal y de las calles, colonia, municipio, Street number y tipo Datetime para la fecha de apertura; usamos Custom list ya que las sucursales son en la República Mexicana.

Para **cliente** se usaron los tipos Row number, First name, Last name, Datetime, Number, custom list para ciudad, Street name, Street number y Gender abreviado.

Para **cajero** se usaron los tipos Row number, First name, Last name, Datetime, City, Number, custom list para ciudad, Street name, Street number y Gender abreviado, number, Datetime y una Custom List para escolaridad con los siguientes grados: Primaria, Secundaria, Bachillerato, Licenciatura, Posgrado.

Para la **tabla venta** se usaron los tipos Row number para los id's, se importaron dataset de las tablas para las llaves foráneas, los dataset fueron de las tablas sucursal, cliente, cajero; en forma de pago usamos una custom list con: Efectivo, Tarjeta credito, Tarjeta debito.

Para las **tablas de los productos**, las llaves se usó del tipo number sin decimales y los números que fueran valores válidos dentro de los ya insertados previamente.

Para el caso de **electrónicos** se creó una custom list con nombres reales de aparatos electrónicos, marcas reales y descripciones y a partir de ahí combinarlas para generar los diferentes valores insertados, para asegurarnos que no hubiese repetidos se creó en nombres una lista bastante grande para que cuando resultara combinada con las marcas no hubieran repetidos, en los demás atributos se generan valores válidos y coherentes que puedan ser insertados.

Para la realización de las siguientes **tablas CorreoSucursal, Telefono Cajero, CorreoCajero, Telefono cliente, CorreoCliente** usamos 3 datasets importantes que fueron cajero, cliente y sucursal. Al poblar estas tablas usamos mockaroo (<https://www.mockaroo.com>), en el que hicimos y subimos los datasets principales para después poderlos usar para todas las demás tablas, para esto nos fuimos a la sección de schemas en la parte de create. Después agregamos todos los atributos necesarios para su población y adjuntamos el dataset que requirieron y llenamos cada campo con sus respectivas características para que a la hora de ejecutarlo no tengamos problemas para poblarlas y poder interactuar con estas. Una vez hecho esto le damos en generate data para que se nos importe y descargue con formato csv y después usamos un conversor de csv a sql (<https://www.convertcsv.io>) para usarlo en nuestro DML.

Para el diseño de los **productos perecederos y noPerecederos** optamos por establecer la id como números enteros, para los nombre creamos listas de los productos más comunes, para cada uno de forma que los datos sean más concordantes con las bases de datos. Para el precio se optó por usar números enteros con precios entre 1 y 500. Con la representación solo se decidió tener una lista con las opción de lata, bolsa y caja. Para las fechas se decidió emplear el formato 'YYYY-MM-DD'.

Para las tablas **Encargado y Gerente** se usaron los tipos de datos como First Name, Last name, street name, street number, para sus respectivos Id's se usó Row Number procurando que no se repitieran unos con otros, se obtuvo el idSucursal del dataset Sucursal para cumplir con las restricciones de llave foránea, el sueldo se estableció con números entero entre un rango determinado, tanto la fechaNacimiento como la fechaContratacion se establecieron con el tipo de dato datetime en un rango viable; en cuanto a la escolaridad se usó una custom list con los ítems: Primaria, Secundaria, Bachillerato, Licenciatura y Posgrado.

Para las tablas **correoEncargado y correoGerente** se obtuvo el id de la tabla encargado y gerente respectivamente y únicamente se generaron los correos con el dato email.

Para las tablas **telefonoEncargado y correoGerente** se obtuvo el id de la tabla encargado y gerente respectivamente y únicamente se generaron los teléfonos con el tipo de dato phone number especificando que queríamos el formato de los 10 dígitos sin guiones.



INFORMACIÓN ACERCA DE LAS TABLAS

cliente:

En esta tabla guardaremos los datos de cada persona que sea un cliente en las sucursales del gran abarrotero.

Atributos.

- ☐ **idPersona:** Atributo que funcionará como el identificador de cada persona, de forma que sea irrepetible. **Tipo de dato:** SERIAL. **Rango del dominio:** Números enteros consecutivos.
- ☐ **nombre:** Atributo que guarda el nombre del cliente. **Tipo de dato:** varchar(30). **Rango del dominio:** Cualquier cadena de texto de longitud máxima 30.
- ☐ **aPaterno:** Atributo que guarda el apellido paterno del cliente. **Tipo de dato:** varchar(15). **Rango del dominio:** Cualquier cadena de texto de longitud máximo 15.
- ☐ **aMaterno:** Atributo que guarda el apellido materno del cliente. **Tipo de dato:** varchar(15). **Rango del dominio:** Cualquier cadena de texto de longitud máxima 15.
- ☐ **fechaNacimiento:** Atributo que guarda la fecha de nacimiento del encargado. **Tipo de dato:** date. **Rango del dominio:** Cualquier fecha.
- ☐ **ciudad:** Atributo que guarda la ciudad de origen del cliente. **Tipo de dato:** varchar(15). **Rango del dominio:** Cualquier cadena de texto de longitud máxima de 15.
- ☐ **cp:** Atributo que guarda el código postal del cliente. **Tipo de dato:** int. **Rango del dominio:** Cualquier número entero.
- ☐ **calle:** Atributo que guarda la calle donde vive el cliente. **Tipo de dato:** varchar(15). **Rango del dominio:** Cualquier cadena de texto de longitud máxima de 15.
- ☐ **número:** Atributo que guarda el número donde vive el cliente. **Tipo de dato:** int. **Rango del dominio:** Cualquier número entero.
- ☐ **género:** Atributo que guarda el género del cliente. **Tipo de dato:** char(1). **Rango del dominio:** 'M' o 'F'.

cajero:

En esta tabla guardaremos los datos de cada persona que esté en el puesto de cajero dentro de alguna sucursal.

Atributos

- ☐ **idPersona:** Atributo que funcionará como el identificador de cada persona, de forma que sea irrepetible. **Tipo de dato:** SERIAL. **Rango del dominio:** Números enteros consecutivos.
- ☐ **idSucursal:** Atributo que funcionará como el identificador de cada sucursal. **Tipo de dato:** int. **Rango del dominio:** Puede tomar prácticamente cualquier número entero.
- ☐ **nombre:** Atributo que guarda el nombre del encargado. **Tipo de dato:** varchar(30). **Rango del dominio:** Cualquier cadena de texto de longitud máxima 30.
- ☐ **aPaterno:** Atributo que guarda el apellido paterno del cajero. **Tipo de dato:** varchar(15). **Rango del dominio:** Cualquier cadena de texto de longitud máximo 15.
- ☐ **aMaterno:** Atributo que guarda el apellido materno del cajero. **Tipo de dato:** varchar(15). **Rango del dominio:** Cualquier cadena de texto de longitud máxima 15.

- ☐ **fechaNacimiento:** Atributo que guarda la fecha de nacimiento del cajero. **Tipo de dato:** date. **Rango del dominio:** Cualquier fecha.
- ☐ **ciudad:** Atributo que guarda la ciudad de origen del cajero. **Tipo de dato:** varchar(15). **Rango del dominio:** Cualquier cadena de texto de longitud máxima de 15.
- ☐ **cp:** Atributo que guarda el código postal del cajero. **Tipo de dato:** int. **Rango del dominio:** Cualquier número entero.
- ☐ **calle:** Atributo que guarda la calle donde vive el cajero. **Tipo de dato:** varchar(15). **Rango del dominio:** Cualquier cadena de texto de longitud máxima de 15.
- ☐ **número:** Atributo que guarda el número donde vive el cajero. **Tipo de dato:** int. **Rango del dominio:** Cualquier número entero.
- ☐ **género:** Atributo que guarda el género del cajero. **Tipo de dato:** char(1). **Rango del dominio:** 'M' o 'F'.
- ☐ **salario:** Atributo que guarda el salario del cajero. **Tipo de dato:** int. **Rango del dominio:** Cualquier número entero.
- ☐ **fechaContratacion:** Atributo que guarda la fecha de contratación del cajero. **Tipo de dato:** date. **Rango del dominio:** Fechas después del año 2000.
- ☐ **escolaridad:** Atributo que guarda la escolaridad del cajero. **Tipo de dato:** varchar(15). **Rango del dominio:** Las cadenas 'Primaria', 'Secundaria', 'Bachillerato', 'Licenciatura' y 'Posgrado'.

encargado:

En esta tabla guardaremos los datos de cada persona que esté en el puesto de encargado dentro de alguna sucursal.

Atributos

- ☐ **idPersona:** Atributo que funcionará como el identificador de cada persona, de forma que sea irrepetible. **Tipo de dato:** SERIAL. **Rango del dominio:** Números enteros consecutivos.
- ☐ **idSucursal:** Atributo que funcionará como el identificador de cada sucursal. **Tipo de dato:** int. **Rango del dominio:** Puede tomar prácticamente cualquier número entero.
- ☐ **nombre:** Atributo que guarda el nombre del encargado. **Tipo de dato:** varchar(30). **Rango del dominio:** Cualquier cadena de texto de longitud máxima 30.
- ☐ **aPaterno:** Atributo que guarda el apellido paterno del encargado. **Tipo de dato:** varchar(15). **Rango del dominio:** Cualquier cadena de texto de longitud máximo 15.
- ☐ **aMaterno:** Atributo que guarda el apellido materno del encargado. **Tipo de dato:** varchar(15). **Rango del dominio:** Cualquier cadena de texto de longitud máxima 15.
- ☐ **fechaNacimiento:** Atributo que guarda la fecha de nacimiento del encargado. **Tipo de dato:** date. **Rango del dominio:** Cualquier fecha.
- ☐ **ciudad:** Atributo que guarda la ciudad de origen del encargado. **Tipo de dato:** varchar(15). **Rango del dominio:** Cualquier cadena de texto de longitud máxima de 15.
- ☐ **cp:** Atributo que guarda el código postal del encargado. **Tipo de dato:** int. **Rango del dominio:** Cualquier número entero.

- ☐ **calle:** Atributo que guarda la calle donde vive el encargado. **Tipo de dato:** varchar(15). **Rango del dominio:** Cualquier cadena de texto de longitud máxima de 15.
- ☐ **número:** Atributo que guarda el número donde vive el encargado. **Tipo de dato:** int. **Rango del dominio:** Cualquier número entero.
- ☐ **género:** Atributo que guarda el género del encargado. **Tipo de dato:** char(1). **Rango del dominio:** 'M' o 'F'.
- ☐ **salario:** Atributo que guarda el salario del encargado. **Tipo de dato:** int. **Rango del dominio:** Cualquier número entero.
- ☐ **fechaContratacion:** Atributo que guarda la fecha de contratación del encargado. **Tipo de dato:** date. **Rango del dominio:** Fechas después del año 2000.
- ☐ **escolaridad:** Atributo que guarda la escolaridad del encargado. **Tipo de dato:** varchar(15). **Rango del dominio:** Las cadenas 'Primaria', 'Secundaria', 'Bachillerato', 'Licenciatura' y 'Posgrado'.

gerente:

En esta tabla guardaremos los datos de cada persona que esté en el puesto de gerente dentro de alguna sucursal.

Atributos

- ☐ **idPersona:** Atributo que funcionará como el identificador de cada persona, de forma que sea irrepetible. **Tipo de dato:** SERIAL. **Rango del dominio:** Números enteros consecutivos.
- ☐ **idSucursal:** Atributo que funcionará como el identificador de cada sucursal. **Tipo de dato:** int. **Rango del dominio:** Puede tomar prácticamente cualquier número entero.
- ☐ **nombre:** Atributo que guarda el nombre del gerente. **Tipo de dato:** varchar(30). **Rango del dominio:** Cualquier cadena de texto de longitud máxima 30.
- ☐ **aPaterno:** Atributo que guarda el apellido paterno del gerente. **Tipo de dato:** varchar(15). **Rango del dominio:** Cualquier cadena de texto de longitud máximo 15.
- ☐ **aMaterno:** Atributo que guarda el apellido materno del gerente. **Tipo de dato:** varchar(15). **Rango del dominio:** Cualquier cadena de texto de longitud máxima 15.
- ☐ **fechaNacimiento:** Atributo que guarda la fecha de nacimiento del gerente. **Tipo de dato:** date. **Rango del dominio:** Cualquier fecha.
- ☐ **ciudad:** Atributo que guarda la ciudad de origen del gerente. **Tipo de dato:** varchar(15). **Rango del dominio:** Cualquier cadena de texto de longitud máxima de 15.
- ☐ **cp:** Atributo que guarda el código postal del gerente. **Tipo de dato:** int. **Rango del dominio:** Cualquier número entero.
- ☐ **calle:** Atributo que guarda la calle donde vive el gerente. **Tipo de dato:** varchar(15). **Rango del dominio:** Cualquier cadena de texto de longitud máxima de 15.
- ☐ **número:** Atributo que guarda el número donde vive el gerente. **Tipo de dato:** int. **Rango del dominio:** Cualquier número entero.
- ☐ **género:** Atributo que guarda el género del gerente. **Tipo de dato:** char(1). **Rango del dominio:** 'M' o 'F'.

- ☐ **salario:**Atributo que guarda el salario del gerente. **Tipo de dato:** int.**Rango del dominio:** Cualquier número entero.
- ☐ **fechaContratacion:** Atributo que guarda la fecha de contratación del gerente. **Tipo de dato:** date .**Rango del dominio:** Fechas después del año 2000.
- ☐ **escolaridad:**Atributo que guarda la escolaridad del gerente. **Tipo de dato:** varchar(15).**Rango del dominio:** Las cadenas 'Primaria', 'Secundaria', 'Bachillerato','Licenciatura' y 'Posgrado'.

perecedero:

Tabla que se hace cargo del almacenamiento de los datos de los productos de tipo perecedero que se venden en las sucursales.

Atributos

- ☐ **idProducto:** Atributo que funcionará como el identificador de cada producto. **Tipo de dato:** SERIAL . **Rango de dominio:** números enteros consecutivos.
- ☐ **nombre:** Atributo que guarda el nombre del producto. **Tipo de dato:** varchar(50) **Rango de dominio:** cadena de texto de longitud máxima 50.
- ☐ **marca:** Almacena el nombre de la marca del producto.**Tipo de dato:** varchar(50) **Rango de dominio:** cadena de texto de longitud máxima 50.
- ☐ **precio:** Representa el precio en el que el producto se venderá en las sucursales.**Tipo de dato:** int. **Rango de dominio:** (0, 2147483647]
- ☐ **presentacion:** Atributo que guarda la forma en que se almacena el producto.**Tipo de dato:** varchar(50) **Rango de dominio:** cadena de texto de longitud máxima 50, con las opción de ser en lata, caja o bolsa.
- ☐ **fechaPreparacion:** Atributo que guarda la fecha en la que se creó el producto. **Tipo de dato:** date. **Rango de dominio:** Mayor a "2000-01-01"
- ☐ **fechaCaducidad:** Atributo que guarda la fecha que indica el momento hasta el cual el alimento puede consumirse de forma segura. **Tipo de dato:** date. **Rango de dominio:** Mayor a "2023-01-01".

noPerecedero:

Tabla que se hace cargo del almacenamiento de los datos de los productos de tipo no perecedero que se venden en las sucursales.

Atributos

- ☐ **idProducto:** Atributo que funcionará como el identificador de cada producto. **Tipo de dato:** SERIAL . **Rango de dominio:** números enteros consecutivos.
- ☐ **nombre:** Atributo que guarda el nombre del producto. **Tipo de dato:** varchar(50) **Rango de dominio:** cadena de texto de longitud máxima 50.
- ☐ **marca:** Almacena el nombre de la marca del producto.**Tipo de dato:** varchar(50) **Rango de dominio:** cadena de texto de longitud máxima 50.
- ☐ **precio:** Representa el precio en el que el producto se venderá en las sucursales.**Tipo de dato:** int. **Rango de dominio:** (0, 2147483647]
- ☐ **presentacion:** Atributo que guarda la forma en que se almacena el producto.**Tipo de dato:** varchar(50) **Rango de dominio:** cadena de texto de longitud máxima 50, con las opción de ser en lata, caja o bolsa.
- ☐ **fechaPreparacion:** Atributo que guarda la fecha en la que se creó el producto. **Tipo de dato:** date. **Rango de dominio:** Mayor a "2000-01-01"

electronico:

Tabla que se hace cargo del almacenamiento de los datos de los productos de tipo electronico que se venden en las sucursales.

Atributos

- ☐ **idProducto:** Atributo que funcionará como el identificador de cada producto. **Tipo de dato:** SERIAL . **Rango de dominio:** números enteros consecutivos.
- ☐ **nombre:** Atributo que guarda el nombre del producto. **Tipo de dato:** varchar(70) **Rango de dominio:** cadena de texto de longitud máxima 70.
- ☐ **marca:** Almacena el nombre de la marca del producto. **Tipo de dato:** varchar(50) **Rango de dominio:** cadena de texto de longitud máxima 50.
- ☐ **precio:** Representa el precio en el que el producto se venderá en las sucursales. **Tipo de dato:** int. **Rango de dominio:** (0, 2147483647]
- ☐ **descripción:** Atributo que guarda la descripción del producto. **Tipo de dato:** varchar(50) **Rango de dominio:** cadena de texto de longitud máxima 50.
- ☐ **descripción:** Atributo que guarda la descripción del producto. **Tipo de dato:** varchar(50) **Rango de dominio:** cadena de texto de longitud máxima 50, con las opción de ser en lata, caja o bolsa.
- ☐ **categoría:** Atributo que guarda la categoría del producto. **Tipo de dato:** varchar(30) **Rango de dominio:** cadena de texto de longitud máxima 30
- ☐ **consumoElectrico:** Atributo que guarda el consumo electrico del producto. **Tipo de dato:** int. **Rango de dominio:** 1-100

sucursal

En esta tabla guardamos la información de cada una de las sucursales del gran abarrotero.

Atributos

- ☐ **idSucursal:** Atributo que funcionará como el identificador de cada sucursal. **Tipo de dato:** SERIAL . **Rango de dominio:** numeros enteros consecutivos.
- ☐ **nombre:** Atributo que guarda el nombre de la sucursal. **Tipo de dato:** varchar(50) **Rango de dominio:** cadena de texto de longitud máxima 50.
- ☐ **municipio:** Atributo que guarda el nombre del municipio de la sucursal. **Tipo de dato:** varchar(50) **Rango de dominio:** cadena de texto de longitud máxima 50.
- ☐ **cp:** Atributo que guarda el código postal de la sucursal. **Tipo de dato:** int. **Rango de dominio:** número entero de 5 dígitos.
- ☐ **ciudad:** Atributo que guarda la ciudad en donde se encuentra la sucursal. **Tipo de dato:** varchar(30) **Rango de dominio:** cadena de texto de longitud máxima 30.
- ☐ **calle:** Atributo que guarda la calle donde se encuentra la sucursal. **Tipo de dato:** varchar(30) **Rango de dominio:** cadena de texto de longitud máxima 30.
- ☐ **numero:** Atributo que guarda el número donde se encuentra localizada la sucursal. **Tipo de dato:** int. **Rango de dominio:** cualquier número entero.
- ☐ **colonia:** Atributo que guarda la colonia de la sucursal. **Tipo de dato:** varchar(30) **Rango de dominio:** cadena de texto de longitud máxima 30.
- ☐ **fechaApertura:** Atributo que guarda la fecha de apertura de la sucursal. **Tipo de dato:** date. **Rango de dominio:** cualquier fecha.

telefonoSucursal

Tabla que guarda la información de los teléfonos de la sucursal.

Atributos

- ☐ **idSucursal:** Atributo que funcionará como el identificador de cada sucursal. **Tipo de dato:** int. **Rango de dominio:** cualquier número entero.
- ☐ **teléfono:** Atributo en donde se guardará el número de teléfono del gerente. **Tipo de dato:** bigint. **Rango del dominio:** Números enteros dentro de -2^{63} a $2^{63}-1$.

correoSucursal

Tabla que guarda la información del correo de la sucursal.

Atributos

- ☐ **idSucursal:** Atributo que funcionará como el identificador de cada sucursal. **Tipo de dato:** int. **Rango de dominio:** cualquier número entero.
- ☐ **correo:** Atributo en donde se guardará el correo de la sucursal. **Tipo de dato:** varchar(30). **Rango del dominio:** Cadenas de texto de longitud máxima de 30.

telefonoGerente:

Tabla que guarda la información de los teléfonos del gerente.

Atributos

- ☐ **idPersona:** Atributo que funcionará como el identificador de cada persona, de forma que sea irrepetible. **Tipo de dato:** SERIAL. **Rango del dominio:** Números enteros consecutivos.
- ☐ **teléfono:** Atributo en donde se guardará el número de teléfono del gerente. **Tipo de dato:** bigint. **Rango del dominio:** Números enteros dentro de -2^{63} a $2^{63}-1$.

telefonoCliente:

Tabla que guarda la información de los teléfonos del cliente.

Atributos

- ☐ **idPersona:** Atributo que funcionará como el identificador de cada persona, de forma que sea irrepetible. **Tipo de dato:** SERIAL. **Rango del dominio:** Números enteros consecutivos.
- ☐ **teléfono:** Atributo en donde se guardará el número de teléfono del gerente. **Tipo de dato:** bigint. **Rango del dominio:** Números enteros dentro de -2^{63} a $2^{63}-1$.

telefonoEncargado:

Atributos

Tabla que guarda la información de los teléfonos del encargado.

- ☐ **idPersona:** Atributo que funcionará como el identificador de cada persona, de forma que sea irrepetible. **Tipo de dato:** SERIAL. **Rango del dominio:** Números enteros consecutivos.
- ☐ **teléfono:** Atributo en donde se guardará el número de teléfono del encargado. **Tipo de dato:** bigint. **Rango del dominio:** Números enteros dentro de -2^{63} a $2^{63}-1$.

telefonoCajero:

Tabla que guarda la información de los teléfonos del cajero.

Atributos

- ☐ **idPersona:** Atributo que funcionará como el identificador de cada persona, de forma que sea irrepetible. **Tipo de dato:** SERIAL. **Rango del dominio:** Números enteros consecutivos.
- ☐ **teléfono:** Atributo en donde se guardará el número de teléfono del encargado. **Tipo de dato:** bigint. **Rango del dominio:** Números enteros dentro de -2^{63} a $2^{63}-1$.

correoCajero:

Tabla que guarda la información del correo de Cajero.

Atributos

- ☐ **idPersona:** Atributo que funcionará como el identificador de cada persona, de forma que sea irrepetible. **Tipo de dato:** SERIAL. **Rango del dominio:** Números enteros consecutivos.
- ☐ **correo:** Atributo en donde se guardará el correo del gerente. **Tipo de dato:** varchar(30). **Rango del dominio:** Cadenas de texto de longitud máxima de 30.

correoGerente:

Atributos

Tabla que guarda la información del correo del gerente.

- ☐ **idPersona:** Atributo que funcionará como el identificador de cada persona, de forma que sea irrepetible. **Tipo de dato:** SERIAL. **Rango del dominio:** Números enteros consecutivos.
- ☐ **correo:** Atributo en donde se guardará el correo del gerente. **Tipo de dato:** varchar(30). **Rango del dominio:** Cadenas de texto de longitud máxima de 30.

correoCliente:

Atributos

Tabla que guarda la información del correo del cliente.

- ☐ **idPersona:** Atributo que funcionará como el identificador de cada persona, de forma que sea irrepetible. **Tipo de dato:** SERIAL. **Rango del dominio:** Números enteros consecutivos.
- ☐ **correo:** Atributo en donde se guardará el correo del gerente. **Tipo de dato:** varchar(30). **Rango del dominio:** Cadenas de texto de longitud máxima de 30.

correoEncargado:

Atributos

Tabla que guarda la información del correo del encargado.

- ☐ **idPersona:** Atributo que funcionará como el identificador de cada persona, de forma que sea irrepetible. **Tipo de dato:** SERIAL. **Rango del dominio:** Números enteros consecutivos.
- ☐ **correo:** Atributo en donde se guardará el correo del encargado. **Tipo de dato:** varchar(30). **Rango del dominio:** Cadenas de texto de longitud máxima de 30.

venta:

Tabla que guarda la información de la venta de productos.

Atributos

- ☐ **idVenta:** Atributo que guarda la información del identificador de la venta. **Tipo de dato:** SERIAL. **Rango de dominio:** números enteros consecutivos .
- ☐ **idSucursal:** Atributo que funcionará como el identificador de cada sucursal. **Tipo de dato:** int. **Rango de dominio:** Puede tomar prácticamente cualquier número entero.
- ☐ **idCliente:** Atributo que funcionará como el identificador de cada persona que sea cliente, de forma que sea irrepensible. **Tipo de dato:** int. **Rango del dominio:** cualquier número entero.
- ☐ **idCajero:** Atributo que funcionará como el identificador de cada persona que sea cajero, de forma que sea irrepensible. **Tipo de dato:** int. **Rango del dominio:** cualquier número entero.
- ☐ **formaPago:** Atributo que guarda la información de la forma de pago de la venta. **Tipo de dato:** varchar(30). **Rango de dominio:** cadena de texto 'Efectivo', 'Tarjeta de credito' 'Tarjeta de debito' .

contarPerecedero:

Tabla que se hace cargo del almacenamiento de los datos de la cantidad de productos de tipo perecedero que se tienen en las sucursales.

Atributos

- ☐ **idSucursal:** Atributo que funcionará como el identificador de cada sucursal. **Tipo de dato:** int. **Rango de dominio:** Puede tomar prácticamente cualquier número entero.
- ☐ **idProducto:** Atributo que funcionará como el identificador de cada producto. **Tipo de dato:** SERIAL . **Rango de dominio:** números enteros consecutivos.
- ☐ **cantidad:** Representa el número de producto que se tienen en las sucursales. **Tipo de dato:** int. **Rango de dominio:** $0 \leq \text{cantidad}$.

contenerPerecedero:

Tabla que se hace cargo del almacenamiento de los datos del número de productos de tipo perecedero que se vendieron en alguna venta.

Atributos

- ☐ **idVenta:** Atributo que guarda la información del identificador de la venta. **Tipo de dato:** SERIAL. **Rango de dominio:** números enteros consecutivos .
- ☐ **idProducto:** Atributo que funcionará como el identificador de cada producto. **Tipo de dato:** SERIAL . **Rango de dominio:** números enteros consecutivos.
- ☐ **cantidad:** Representa el número de producto que se vendieron. **Tipo de dato:** int. **Rango de dominio:** $0 < \text{cantidad}$.

contarNoPerecedero:

Tabla que se hace cargo del almacenamiento de los datos de la cantidad de productos de tipo no perecedero que se tienen en las sucursales.

Atributos

- ☐ **idSucursal:** Atributo que funcionará como el identificador de cada sucursal. **Tipo de dato:** int. **Rango de dominio:** Puede tomar prácticamente cualquier número entero.

- ☐ **idProducto:** Atributo que funcionará como el identificador de cada producto. **Tipo de dato:** SERIAL . **Rango de dominio:** números enteros consecutivos.
- ☐ **cantidad:** Representa el número de producto que se tienen en las sucursales. **Tipo de dato:** int. **Rango de dominio:** $0 \leq \text{cantidad}$.

contenerNoPerecedero:

Tabla que se hace cargo del almacenamiento de los datos del número de productos de tipo no perecedero que se vendieron en alguna venta.

Atributos

- ☐ **idVenta:** Atributo que guarda la información del identificador de la venta. **Tipo de dato:** SERIAL. **Rango de dominio:** números enteros consecutivos .
- ☐ **idProducto:** Atributo que funcionará como el identificador de cada producto. **Tipo de dato:** SERIAL . **Rango de dominio:** números enteros consecutivos.
- ☐ **cantidad:** Representa el número de producto que se vendieron. **Tipo de dato:** int. **Rango de dominio:** $0 < \text{cantidad}$.

contarElectronico:

Tabla que se hace cargo del almacenamiento de los datos de la cantidad de productos de tipo electronica que se tienen en las sucursales.

Atributos

- ☐ **idSucursal:** Atributo que funcionará como el identificador de cada sucursal. **Tipo de dato:** int. **Rango de dominio:** Puede tomar prácticamente cualquier número entero.
- ☐ **idProducto:** Atributo que funcionará como el identificador de cada producto. **Tipo de dato:** SERIAL . **Rango de dominio:** números enteros consecutivos.
- ☐ **cantidad:** Representa el número de producto que se tienen en las sucursales. **Tipo de dato:** int. **Rango de dominio:** $0 \leq \text{cantidad}$.

contenerElectronico:

Tabla que se hace cargo del almacenamiento de los datos del número de productos de tipo electronico que se vendieron en alguna venta.

Atributos

- ☐ **idVenta:** Atributo que guarda la información del identificador de la venta. **Tipo de dato:** SERIAL. **Rango de dominio:** números enteros consecutivos .
- ☐ **idProducto:** Atributo que funcionará como el identificador de cada producto. **Tipo de dato:** SERIAL . **Rango de dominio:** números enteros consecutivos.
- ☐ **cantidad:** Representa el número de producto que se vendieron. **Tipo de dato:** int. **Rango de dominio:** $0 < \text{cantidad}$.

CONSULTAS

En la parte de consultas decidimos llevarlas a cabo dentro de distintos grados de dificultad, con ello logramos probar el funcionamiento de nuestra base de datos, asimismo buscamos probar su integridad al momento de llevar a cabo una manipulación de los datos.

A continuación enumeramos cada una de ellas así como mostramos los datos que arrojan al momento de ejecutarlas, cabe mencionar que las consultas 1 a la 5 son las que habíamos efectuado en la Práctica 7, de ahí en adelante son las correspondientes al proyecto.

1. Empleados cuyo nombre empiece con la letra C

```
-- 1. Empleados cuyo nombre empiece con la letra C.
select *
from cajero
where nombre like 'C%';
```

Resultados:

| | Idpersona [PK] integer | Idsucursal integer | nombre character varying | apaterno character varying | amaterno character varying | fechanacimiento date |
|----|---------------------------|-----------------------|-----------------------------|-------------------------------|-------------------------------|-------------------------|
| 1 | 1006 | 9 | Cobb | Bigglestone | Mattacks | 1981-06-07 |
| 2 | 1008 | 5 | Carter | Tarply | Whether | 1976-10-31 |
| 3 | 1021 | 14 | Cross | Greatham | Dingwall | 1991-06-02 |
| 4 | 1030 | 12 | Clevey | Botler | Doubleday | 1965-11-16 |
| 5 | 1034 | 10 | Camella | Stott | Avramov | 2001-05-21 |
| 6 | 1051 | 6 | Chickie | Johnes | Jagoe | 1985-12-21 |
| 7 | 1054 | 9 | Cornie | Warwicker | Luker | 1966-10-27 |
| 8 | 1056 | 8 | Culley | Decort | Parmeter | 1991-03-28 |
| 9 | 1116 | 15 | Christos | Mehmet | Akers | 1972-11-23 |
| 10 | 1118 | 13 | Cordula | Danford | Latus | 1961-10-29 |
| 11 | 1121 | 2 | Clerc | Saunt | Shillito | 1972-05-09 |
| 12 | 1135 | 15 | Ciel | Dumbell | Akehurst | 1981-01-13 |
| 13 | 1152 | 2 | Chilton | Adran | Larrat | 1998-04-01 |
| 14 | 1153 | 2 | Carey | Christophersen | Leblanc | 1995-07-25 |
| 15 | 1174 | 15 | Carlie | Eddins | MacComiskey | 1972-05-13 |
| 16 | 1179 | 12 | Cyndy | Crielly | Burgliss | 1996-06-01 |

2. Clientes que hayan nacido en el mes Junio

```
-- 2. Clientes que hayan nacido en el mes de Junio.
select *
from cliente
where to_char(fechanacimiento,'YYYY-MM-DD') like '%-06-%';
```

Resultados:

| | idpersona [PK] integer | nombre character varying | apaterno character varying | amaterno character varying | fechanacimiento date | cp integer | ciudad character varying |
|----|---------------------------|-----------------------------|-------------------------------|-------------------------------|-------------------------|---------------|-----------------------------|
| 1 | 36 | Gare | Storrie | MacGillavery | 1981-06-05 | 12956 | Fresnillo |
| 2 | 53 | Cathe | Brew | Crossgrove | 1963-06-24 | 76224 | Culiacán |
| 3 | 54 | Red | Poznanski | Janous | 2001-06-06 | 45158 | Papantla de Olarte |
| 4 | 66 | Morie | Georgi | Heintsch | 2001-06-26 | 67736 | Ciudad de México |
| 5 | 72 | Verina | Holehouse | Fidilis | 1984-06-23 | 48528 | Aguascalientes |
| 6 | 105 | Federico | Aleksandrov | O'Lyhane | 1965-06-06 | 59269 | Guadalajara |
| 7 | 106 | Jillian | Vanyard | Pleasaunce | 1982-06-19 | 52280 | San Juan del Río |
| 8 | 120 | Krystle | Hempel | Greenman | 1993-06-21 | 71838 | Ciudad Obregón |
| 9 | 136 | Alethea | Caudelier | Targetter | 1991-06-11 | 10232 | Zacatecas |
| 10 | 210 | Audrey | Lamb-shine | Hegg | 1991-06-19 | 82516 | Matamoros |
| 11 | 247 | Nickolas | Willatt | Kelwick | 1973-06-02 | 94536 | La Paz |
| 12 | 264 | Jerri | Sherrum | Harner | 1987-06-04 | 61794 | Piedras Negras |
| 13 | 289 | Philip | Nusche | Myatt | 1971-06-07 | 34018 | Monterrey |
| 14 | 294 | Donny | Rickesles | Jachimczak | 1979-06-20 | 59226 | Chilpancingo de los |
| 15 | 298 | Maxie | McMurty | Poulton | 1995-06-18 | 38672 | Mérida |
| 16 | 302 | Creigh | Toyer | Burling | 1970-06-21 | 17850 | Matamoros |



3. Productos cuya fecha de caducidad esté entre el 1 de enero del 2023 y del 7 de mayo del 2023

```
-- 3. Productos cuya fecha de caducidad este entre el 1 de enero del 2023 y del 7 de mayo del 2023.
select *
from perecedero
where fechacaducidad >= '2023-01-01' AND fechacaducidad <= '2023-05-15';
```

Resultados:

| | Idproducto [PK] integer | nombre character varying | marca character varying | precio integer | presentacion character varying | fechapreparacion date | fechacaducidad date |
|----|----------------------------|-----------------------------|----------------------------|-------------------|-----------------------------------|--------------------------|------------------------|
| 1 | 2 | Res | Pete and Gerry's | 367 | Bolsa | 2023-02-01 | 2023-01-16 |
| 2 | 3 | Rosquillas | ident Seafoods | 233 | Caja | 2023-01-09 | 2023-02-16 |
| 3 | 4 | Tilapia | Fage | 232 | Bolsa | 2023-03-02 | 2023-04-18 |
| 4 | 5 | Sushi | Land O'Lakes | 434 | Botella | 2023-02-25 | 2023-03-23 |
| 5 | 6 | Guisantes | Nature's Own | 456 | Lata | 2023-03-06 | 2023-04-07 |
| 6 | 7 | Pepperoni | Challenge | 374 | Botella | 2023-05-04 | 2023-04-29 |
| 7 | 8 | Trucha | Oikos | 441 | Bolsa | 2023-02-25 | 2023-04-26 |
| 8 | 9 | Pato | Challenge | 96 | Caja | 2023-01-04 | 2023-02-07 |
| 9 | 11 | Melisa | Challenge | 380 | Caja | 2023-03-07 | 2023-05-11 |
| 10 | 13 | Laurel | President | 426 | Lata | 2023-02-13 | 2023-01-17 |
| 11 | 14 | Pan de pasas | Sargento | 289 | Lata | 2023-05-06 | 2023-03-16 |
| 12 | 15 | Hojas de curry | Nature's Own | 183 | Lata | 2023-05-06 | 2023-03-10 |
| 13 | 16 | Espárragos | Yoplait | 268 | Lata | 2023-03-26 | 2023-01-31 |
| 14 | 17 | Zanahorias | Organic Valley | 458 | Lata | 2023-01-20 | 2023-03-22 |
| 15 | 18 | Patatas | Perdue | 449 | Botella | 2023-01-02 | 2023-04-27 |
| 16 | 19 | Remolachas | Smithfield | 168 | Botella | 2023-04-16 | 2023-04-05 |

ALPHA

4. Productos cuya presentación sea en lata

```
-- 4. Productos cuya presentación sea en lata.
select *
from perecedero
where presentacion = 'Lata'
```

Resultados:

| | Idproducto [PK] integer | nombre character varying | marca character varying | precio integer | presentacion character varying | fechapreparacion date | fechacaducida date |
|----|----------------------------|-----------------------------|----------------------------|-------------------|-----------------------------------|--------------------------|-----------------------|
| 1 | 6 | Guisantes | Nature's Own | 456 | Lata | 2023-03-06 | 2023-04-07 |
| 2 | 13 | Laurel | President | 426 | Lata | 2023-02-13 | 2023-01-17 |
| 3 | 14 | Pan de pasas | Sargento | 289 | Lata | 2023-05-06 | 2023-03-16 |
| 4 | 15 | Hojas de curry | Nature's Own | 183 | Lata | 2023-05-06 | 2023-03-10 |
| 5 | 16 | Espárragos | Yoplait | 268 | Lata | 2023-03-26 | 2023-01-31 |
| 6 | 17 | Zanahorias | Organic Valley | 458 | Lata | 2023-01-20 | 2023-03-22 |
| 7 | 20 | Pan de centeno | King Oscar | 405 | Lata | 2023-05-08 | 2023-02-23 |
| 8 | 30 | Peras | Organic Valley | 493 | Lata | 2023-04-27 | 2023-03-05 |
| 9 | 31 | Coles de Bruselas | Pepperidge Farm | 118 | Lata | 2023-02-13 | 2023-02-02 |
| 10 | 37 | Patatas | Sara Lee | 163 | Lata | 2023-03-16 | 2023-02-18 |
| 11 | 42 | Avestruz | Hormel | 270 | Lata | 2023-04-11 | 2023-03-22 |
| 12 | 44 | Papayas | Yoplait | 432 | Lata | 2023-05-02 | 2023-03-10 |
| 13 | 55 | Pretzels | Organic Valley | 293 | Lata | 2023-01-02 | 2023-01-29 |
| 14 | 57 | Ciruelas | Tyson | 265 | Lata | 2023-03-17 | 2023-01-21 |
| 15 | 58 | Chorizo | BelGioioso | 182 | Lata | 2023-03-30 | 2023-04-08 |



5. Todos los clientes registrados en el Gran Abarrotero

```
-- 5. Todos los Clientes registrados en El Gran Abarrotero.
select *
from cliente;
```

Resultados:

| | idpersona [PK] integer | nombre character varying | apaterno character varying | amaterno character varying | fechanacimiento date | cp integer | ciudad character varying |
|----|---------------------------|-----------------------------|-------------------------------|-------------------------------|-------------------------|---------------|-----------------------------|
| 1 | 1 | Harriot | Fallis | Piet | 1969-09-22 | 68286 | Matamoros |
| 2 | 2 | Weber | Stutely | Chetter | 1971-09-11 | 30578 | Teziutlán |
| 3 | 3 | Pincus | Bohman | Borthwick | 1971-09-16 | 16349 | Saltillo |
| 4 | 4 | Laurie | Tremayne | Mitro | 1961-02-20 | 48359 | Hermosillo |
| 5 | 5 | Osbert | Farens | Gibke | 1998-03-11 | 59302 | Ciudad Obregón |
| 6 | 6 | Meagan | Flaunders | Lainton | 1985-01-20 | 63700 | Zacatecas |
| 7 | 7 | Gayelord | Kerrod | Tellenbrook | 1998-05-02 | 99963 | Mexicali |
| 8 | 8 | Mufinella | Hearsey | Rodder | 1994-11-02 | 32899 | Saltillo |
| 9 | 9 | Hedwiga | Eck | Terris | 1962-02-16 | 13477 | Guaymas |
| 10 | 10 | Phillis | Stroyan | Fells | 1972-04-14 | 34281 | Zacatecas |
| 11 | 11 | Sherwynd | Ritchard | Bisgrove | 1979-03-05 | 24263 | Fresnillo |
| 12 | 12 | Van | Schrir | Siemens | 1975-03-31 | 51478 | Fresnillo |
| 13 | 13 | Sunny | Saffell | Klainman | 1977-02-01 | 79052 | Monterrey |
| 14 | 14 | Kirby | Dawtre | Cullingworth | 1997-12-22 | 89580 | San Luis Potosí |
| 15 | 15 | Bertine | Senior | Deal | 1971-05-21 | 73591 | Papantla de Olarte |
| 16 | 16 | Merrill | Demann | Heffernon | 1982-08-19 | 77427 | Papantla de Olarte |



6. Gerentes que fueron contratados entre marzo y junio de 2016

```
-- 6. Consultar quienes son los gerentes que fueron contratados entre marzo y junio de 2016
select nombre, ciudad, fechaNacimiento, fechaContratacion
from gerente
where fechaContratacion between '2016-03-01' and '2016-06-30';
```

Resultados:


| | nombre character varying | ciudad character varying | fechanacimiento date | fechacontratacion date |
|----|-----------------------------|-----------------------------|-------------------------|---------------------------|
| 1 | Andrey | Villa Allende | 1998-12-04 | 2016-05-22 |
| 2 | Homer | Aylmer | 1998-04-28 | 2016-05-15 |
| 3 | Ingamar | Boissevain | 1987-02-19 | 2016-03-23 |
| 4 | Humphrey | Xiapu | 1978-03-27 | 2016-06-02 |
| 5 | Staford | Arvika | 1997-12-22 | 2016-05-19 |
| 6 | Thomasa | Törva | 1975-05-26 | 2016-06-19 |
| 7 | Rakel | Teruel | 1976-05-01 | 2016-03-15 |
| 8 | Abbe | Curridabat | 1997-10-06 | 2016-05-18 |
| 9 | Benedikta | Daoukro | 1985-02-09 | 2016-04-21 |
| 10 | Rochette | Mashah | 2001-05-24 | 2016-05-06 |



7. Lista de productos perecederos más vendidos en todas las sucursales ordenados de forma descendente

```
-- 7. Lista de productos perecederos más vendidos en todas las sucursales ordenados de
-- forma descendente
select perecedero.idproducto, perecedero.nombre, count(contarperecedero.idproducto)
as total_ventas
from perecedero
join contarperecedero on perecedero.idproducto = contarperecedero.idproducto
join venta on contarperecedero.idproducto = venta.idventa
where venta.idsucursal between 1 and 15
group by perecedero.idproducto, perecedero.nombre
order by total_ventas desc;
```

Resultados:



| | idproducto [PK] integer | nombre character varying | total_ventas bigint |
|----|----------------------------|-----------------------------|------------------------|
| 1 | 175 | Frambuesas | 6 |
| 2 | 177 | Bisonte | 5 |
| 3 | 481 | Damascos | 5 |
| 4 | 236 | Queso cottage | 5 |
| 5 | 544 | Pastrami | 5 |
| 6 | 231 | Cebollino | 4 |
| 7 | 771 | Sushi | 4 |
| 8 | 711 | Orégano | 4 |
| 9 | 311 | Cerdo | 4 |
| 10 | 284 | Galletas | 4 |
| 11 | 873 | Espárragos | 4 |
| 12 | 41 | Leche | 4 |
| 13 | 879 | Bacalao | 4 |
| 14 | 14 | Pan de pasas | 4 |
| 15 | 215 | Mantequilla | 4 |
| 16 | 315 | Sardinas | 3 |

8. Clientes cuyo apellido paterno inicie con F y sean mujeres.

```
-- 8. Los clientes cuyo apellido paterno inicie con F y sean mujeres
select nombre,aPaterno,aMaterno
from cliente
where aPaterno like 'F%' and genero = 'F';
```

Resultados:

| | nombre character varying | apaterno character varying | amaterno character varying |
|----|-----------------------------|-------------------------------|-------------------------------|
| 1 | Harriot | Fallis | Piet |
| 2 | Meagan | Flaunders | Lainton |
| 3 | Linn | Folca | Cousen |
| 4 | Muffinella | Falconbridge | Gilding |
| 5 | Conchita | Fray | Maps |
| 6 | Lily | Faraker | Whitelock |
| 7 | Salome | Fruen | Pomeroy |
| 8 | Matty | Fernehough | Vasilenko |
| 9 | Marybeth | Fitkin | Straine |
| 10 | Daphene | Folshom | Goodayle |
| 11 | Alisha | Forsyth | Brisset |
| 12 | Alberta | Fynn | Glasner |
| 13 | Alexandra | Face | Dudeney |
| 14 | Leontyne | Flattman | Glassopp |
| 15 | Gloria | Father | Vasilchenko |
| 16 | Otha | Francom | Cardenoso |



9. Nombre y marca de productos cuya presentación sea en caja de precio mayor o igual a 200 y que se caduquen antes del 30 de mayo del 2023.

```
-- 9. Nombre y marca de productos cuya presentación sea en caja, de precio mayor  
-- o igual a 200 y que se caduquen antes del 30 de mayo del 2023  
select nombre,marca  
from perecedero  
where presentacion = 'Caja' and precio >= 200 and fechaCaducidad < '2023-05-30';
```

Resultados:

| | nombre character varying | marca character varying |
|----|-----------------------------|----------------------------|
| 1 | Rosquillas | rident Seafoods |
| 2 | Zanahorias | Plugrá |
| 3 | Melisa | Challenge |
| 4 | Jamón serrano | Chobani |
| 5 | Ajedrea | Bumble Bee |
| 6 | Leche | Organic Valley |
| 7 | Coppa | Parmalat |
| 8 | Rábanos | Organic Valley |
| 9 | Pollo | Tillamook |
| 10 | Perejil | Kerrygold |
| 11 | Queso | Chobani |
| 12 | Patatas | Fage |
| 13 | Salami | Fage |
| 14 | Coles de Bruselas | Smithfield |
| 15 | Cebollas | Wonder Bread |
| 16 | Rosquillas | Plugrá |

10. Gerentes con salario entre 15,000 y 60,000 que sean de las sucursales 2,6 o 12.

```
--10. Gerentes con salario entre 15,000 y 60,000 pesos y que sean de las sucursales 2,6 o 12
SELECT *
FROM gerente
WHERE salario BETWEEN 15000 AND 60000 AND idSucursal IN (6,2,12);
```

Resultados:

| | fecha_nacimiento date | ciudad character varying | cp integer | calle character varying | numero integer | genero character | salario integer | fecha_contratacion date | escolaridad character varying |
|----|--------------------------|-----------------------------|---------------|----------------------------|-------------------|---------------------|--------------------|----------------------------|----------------------------------|
| 1 | 1996-10-03 | Cikuda | 28155 | Crowley | 847 | M | 33467 | 2018-12-12 | Posgrado |
| 2 | 2003-06-16 | Tiraspol | 69739 | Carberry | 6211 | F | 49632 | 2010-05-24 | Secundaria |
| 3 | 1993-05-22 | Genet | 60679 | Barnett | 46214 | F | 26154 | 2013-01-26 | Posgrado |
| 4 | 1995-07-28 | Morávka | 74812 | Morning | 8 | M | 22717 | 2009-02-21 | Secundaria |
| 5 | 2003-11-15 | Sutukung | 58381 | Mosinee | 181 | F | 57125 | 2003-10-05 | Secundaria |
| 6 | 1996-12-11 | Sulçcin | 51795 | Old Gate | 5465 | M | 54864 | 2003-07-26 | Posgrado |
| 7 | 2001-06-12 | Kasturi | 60847 | Superior | 395 | M | 32149 | 2011-04-07 | Bachillerato |
| 8 | 1981-05-04 | Liozon | 63111 | Eggendart | 56002 | F | 38477 | 2012-03-15 | Secundaria |
| 9 | 1985-11-30 | Huertas | 25266 | West | 6122 | F | 36392 | 2015-10-08 | Posgrado |
| 10 | 1987-10-19 | Jampang Tengah | 45486 | Emmet | 9969 | M | 34990 | 2018-01-30 | Secundaria |
| 11 | 1993-05-23 | Guilherand-Granges | 97328 | Jana | 15619 | F | 25127 | 2007-01-27 | Primaria |
| 12 | 1997-11-28 | Vargön | 99566 | Schmedeman | 247 | M | 45222 | 2001-03-07 | Licenciatura |
| 13 | 1998-09-04 | Talisay | 97530 | Bay | 6121 | F | 23493 | 2002-07-17 | Bachillerato |
| 14 | 1980-06-19 | Nouâdhibou | 98881 | Di Loreto | 74 | M | 59401 | 2012-01-03 | Bachillerato |
| 15 | 1984-10-22 | Cipicung | 44642 | Ridgeview | 675 | M | 57335 | 2001-12-17 | Primaria |
| 16 | 1973-05-23 | Kaduseeng | 73781 | Armistice | 583 | F | 23505 | 2007-08-21 | Primaria |



11. Encargados con salario mayor a 10,000 pesos y que hayan sido contratados en el primer trimestre del 2023

```
--11. Encargados con salario mayor a 10,000 pesos y que hayan sido contratados en
-- el primer trimestre de 2023
SELECT *
FROM encargado
WHERE salario >= 10000 AND fechaContratacion >= '2023-01-01' AND fechaContratacion <= '2023-01-31';
```

Resultados:

| | ciudad character varying | cp integer | calle character varying | numero integer | genero character | salario integer | fechacontratacion date | escolaridad character varying |
|----|-----------------------------|---------------|----------------------------|-------------------|---------------------|--------------------|---------------------------|----------------------------------|
| 1 | Hengshanqiao | 57875 | Sugar | 56 | F | 37683 | 2023-01-28 | Bachillerato |
| 2 | Banjar Bucu | 62311 | Truax | 5775 | M | 56480 | 2023-01-24 | Licenciatura |
| 3 | Dimona | 98273 | Pleasure | 8 | F | 16409 | 2023-01-06 | Posgrado |
| 4 | Hyattsville | 73874 | Michigan | 18 | F | 11226 | 2023-01-23 | Posgrado |
| 5 | Miragaia | 77116 | Lunder | 27097 | M | 43473 | 2023-01-04 | Posgrado |
| 6 | Kedungasem | 32998 | Elgar | 2108 | F | 36201 | 2023-01-24 | Secundaria |
| 7 | Kladruby | 24929 | Jay | 32048 | M | 64960 | 2023-01-31 | Licenciatura |
| 8 | Kälvä | 78941 | School | 4881 | M | 17862 | 2023-01-16 | Bachillerato |
| 9 | Uppsala | 90588 | Waubesa | 5 | F | 20878 | 2023-01-05 | Bachillerato |
| 10 | Xiejiatan | 42394 | Sutteridge | 23 | F | 46397 | 2023-01-23 | Licenciatura |
| 11 | Shanhaiguan | 91660 | Delaware | 59394 | F | 34937 | 2023-01-06 | Bachillerato |
| 12 | Sv. Trojica v Slov. Gorica | 34003 | Montana | 6645 | F | 12430 | 2023-01-21 | Primaria |
| 13 | Bakung | 93328 | Dawn | 56 | M | 19188 | 2023-01-04 | Licenciatura |
| 14 | Čajetina | 53081 | Farwell | 249 | M | 48687 | 2023-01-22 | Licenciatura |
| 15 | Žalec | 62320 | Columbus | 272 | M | 33829 | 2023-01-13 | Primaria |



12. Clientes que sean de Monterrey, Aguascalientes, Acapulco o Morelia y cuya fecha de nacimiento sea mayor a 1990

```
--12.Clientes que sean de Monterrey, Aguascalientes, Acapulco o Morelia y cuya
-- fecha de nacimiento sea mayor a 1990
SELECT *
FROM cliente
WHERE ciudad = 'Monterrey' OR ciudad = 'Aguascalientes' OR ciudad = 'Acapulco' OR
ciudad = 'Morelia' AND fechaNacimiento >= '1990-01-01';
```

Resultados:

| | Idpersona [PK] integer | nombre character varying | apaterno character varying | amaterno character varying | fechanacimiento date | cp integer | ciudad character varying | calle character varying |
|----|---------------------------|-----------------------------|-------------------------------|-------------------------------|-------------------------|---------------|-----------------------------|----------------------------|
| 1 | 13 | Sunny | Saffell | Klainman | 1977-02-01 | 79052 | Monterrey | Bellgrove |
| 2 | 43 | Gregorio | Palfery | Eddis | 1973-11-05 | 23521 | Monterrey | Moulton |
| 3 | 44 | Arman | Byles | Maunders | 1988-11-22 | 24990 | Acapulco | Graedel |
| 4 | 61 | Chlo | Mitroshinov | Brum | 1967-09-15 | 62415 | Aguascalientes | Shopko |
| 5 | 62 | Ferrel | Eads | Dorracott | 1977-03-04 | 42126 | Monterrey | Buell |
| 6 | 72 | Verina | Holehouse | Fidilis | 1984-06-23 | 48528 | Aguascalientes | Dryden |
| 7 | 81 | Melamie | Dullingham | Claiton | 1991-07-26 | 88051 | Morelia | Continental |
| 8 | 94 | Wadsworth | Selman | Pickin | 2002-12-06 | 68382 | Monterrey | Prentice |
| 9 | 104 | Nissy | Waudby | Adolphine | 1986-04-08 | 26513 | Acapulco | Tomscot |
| 10 | 111 | Karl | Carson | Boyton | 1987-12-12 | 94785 | Aguascalientes | Cordelia |
| 11 | 127 | Ealasaid | Gianasi | Mackaile | 2001-03-20 | 32223 | Aguascalientes | Shopko |
| 12 | 179 | Dylan | Vedekhov | Pounder | 1962-10-10 | 91348 | Aguascalientes | 5th |
| 13 | 180 | Brier | Lanyon | Amberger | 1967-03-26 | 42414 | Aguascalientes | Mariners Cove |
| 14 | 189 | Sigfrid | Beckmann | Grumble | 1993-04-09 | 26132 | Morelia | Anzinger |
| 15 | 195 | Marybeth | Fitkin | Straine | 2002-09-04 | 45373 | Aguascalientes | Sutteridge |
| 16 | 205 | Rayna | McAree | Hapke | 1961-02-09 | 61810 | Aguascalientes | Forest |



13. Ventas en efectivo efectuadas por cajeros que tengan sueldo menor o igual a 12000 pesos y sean de las sucursales 5,8 o 15.

```
-- 13.Ventas en efectivo efectuadas por cajeros que tengan sueldo menor o igual a
-- 12,000 pesos y sean de las sucursales 5,8 o 15
SELECT *
FROM cajero NATURAL JOIN venta
WHERE salario <= 12000 AND formaPago = 'Efectivo' AND genero = 'F' AND idSucursal IN (5,8,15) ;
```

Resultados:

| | er varying | salario integer | fechacontratacion date | escolaridad character varying | idventa integer | idcliente integer | idcajero integer | formapago character varying |
|----|------------|--------------------|---------------------------|----------------------------------|--------------------|----------------------|---------------------|--------------------------------|
| 1 | | 11234 | 2023-07-29 | Posgrado | 951 | 751 | 1905 | Efectivo |
| 2 | | 11234 | 2023-07-29 | Posgrado | 857 | 659 | 1207 | Efectivo |
| 3 | | 11234 | 2023-07-29 | Posgrado | 753 | 380 | 1956 | Efectivo |
| 4 | | 11234 | 2023-07-29 | Posgrado | 752 | 903 | 1236 | Efectivo |
| 5 | | 11234 | 2023-07-29 | Posgrado | 687 | 575 | 1144 | Efectivo |
| 6 | | 11234 | 2023-07-29 | Posgrado | 676 | 222 | 1625 | Efectivo |
| 7 | | 11234 | 2023-07-29 | Posgrado | 674 | 15 | 1479 | Efectivo |
| 8 | | 11234 | 2023-07-29 | Posgrado | 648 | 261 | 1577 | Efectivo |
| 9 | | 11234 | 2023-07-29 | Posgrado | 629 | 503 | 1766 | Efectivo |
| 10 | | 11234 | 2023-07-29 | Posgrado | 588 | 534 | 1831 | Efectivo |
| 11 | | 11234 | 2023-07-29 | Posgrado | 584 | 833 | 1924 | Efectivo |
| 12 | | 11234 | 2023-07-29 | Posgrado | 544 | 565 | 1071 | Efectivo |
| 13 | | 11234 | 2023-07-29 | Posgrado | 512 | 249 | 1679 | Efectivo |
| 14 | | 11234 | 2023-07-29 | Posgrado | 494 | 544 | 1237 | Efectivo |



14. Toda la información de los clientes que tengan al menos dos compras.

```
--14. Toda la informacion de los clientes que tengan al menos dos compras
SELECT *
FROM cliente
WHERE idPersona IN (
    SELECT idCliente
    FROM venta
    GROUP BY idCliente
    HAVING COUNT(*) >= 2
);
```

Resultados:

| | Idpersona [PK] integer | nombre character varying | apaterno character varying | amaterno character varying | fechanacimiento date | cp integer | ciudad character varying | ca cl |
|----|---------------------------|-----------------------------|-------------------------------|-------------------------------|-------------------------|---------------|-----------------------------|----------|
| 1 | 7 | Gayelord | Kerrod | Tellenbrook | 1998-05-02 | 99963 | Mexicali | M |
| 2 | 11 | Sherwynd | Ritchard | Bisgrove | 1979-03-05 | 24263 | Fresnillo | N |
| 3 | 13 | Sunny | Saffell | Klainman | 1977-02-01 | 79052 | Monterrey | B |
| 4 | 14 | Kirby | Dawtre | Cullingworth | 1997-12-22 | 89580 | San Luis Potosi | 8 |
| 5 | 15 | Bertine | Senior | Deal | 1971-05-21 | 73591 | Papantla de Olarte | U |
| 6 | 16 | Merrill | Demann | Heffernon | 1982-08-19 | 77427 | Papantla de Olarte | S |
| 7 | 18 | Collette | Coatham | Heinl | 1976-09-03 | 41737 | Matamoros | T |
| 8 | 20 | Wynn | Maybury | Bowdler | 1987-04-08 | 48529 | Piedras Negras | N |
| 9 | 23 | Flo | Cottingham | Geraldez | 1978-12-06 | 65749 | Cuernavaca | H |
| 10 | 24 | Carlin | Feacham | Gabbatiss | 1985-11-01 | 28699 | Reynosa | H |
| 11 | 28 | Marthena | Pettyfer | Tesimon | 1962-10-10 | 32931 | Papantla de Olarte | H |
| 12 | 30 | Carlie | Merrell | Goater | 1986-12-28 | 76995 | San Luis Potosi | C |
| 13 | 32 | Clarabelle | Kemster | Kundert | 1987-11-12 | 73125 | Ciudad Madero | C |
| 14 | 38 | Jacobo | Eilhart | Rechert | 1972-09-08 | 88157 | Chihuahua | D |
| 15 | 41 | Desiree | Antoniazzi | Mendonca | 1976-10-19 | 59097 | Orizaba | S |

ALPHA

15. Los 100 cajeros que más ventas han realizado

```
-- 15. Obtener los 100 cajeros que más ventas ha realizado
select idpersona,idsucursal,concat(nombre,' ', apaterno,' ',amaterno) as nombre,
      to_char(fechanacimiento, 'Mon dd yyyy') as fechanacimiento,
      to_char(fechacontratacion, 'Mon dd yyyy') as fechaContratacion,
      escolaridad,count( idpersona) as numVentas
from cajero c natural join venta v
group by idpersona
order by numventas desc
limit 100;
```

Resultados:

| | idpersona [PK] integer | idsucursal integer | nombre text | fechanacimiento text | fechacontratacion text | escolaridad character varying | numventas bigint |
|----|---------------------------|-----------------------|-----------------------------|-------------------------|---------------------------|----------------------------------|---------------------|
| 1 | 1344 | 15 | Jacqueline Mallison Baudin | Aug 29 1961 | Mar 31 2023 | Secundaria | 84 |
| 2 | 1237 | 15 | Marita Isaacs Casotti | Aug 10 1963 | May 07 2023 | Bachillerato | 84 |
| 3 | 1135 | 15 | Ciel Dumbell Akehurst | Jan 13 1981 | Aug 28 2023 | Secundaria | 84 |
| 4 | 1949 | 15 | Sansone Lemoir Beckford | Dec 26 1971 | Aug 17 2023 | Posgrado | 84 |
| 5 | 1572 | 15 | Sauncho Linggood Antoshin | Jul 19 1992 | Nov 01 2023 | Licenciatura | 84 |
| 6 | 1744 | 15 | Garrett Villa Lettice | Oct 02 1964 | Feb 27 2023 | Primaria | 84 |
| 7 | 1211 | 15 | Danni Gillion Bostick | Mar 09 1989 | Dec 09 2023 | Secundaria | 84 |
| 8 | 1543 | 15 | Rinaldo Hickinbottom McCook | Jul 11 1980 | Nov 20 2023 | Licenciatura | 84 |
| 9 | 1120 | 15 | Wadsworth Dennant Nyles | Jul 10 1999 | Feb 19 2023 | Bachillerato | 84 |
| 10 | 1229 | 15 | Bevon Girling Lintott | May 30 1975 | Sep 02 2023 | Bachillerato | 84 |
| 11 | 1301 | 15 | Theda Caveney Craze | May 28 1975 | Mar 16 2023 | Secundaria | 84 |
| 12 | 1914 | 15 | Megen Pattinson Goodrich | Apr 07 1984 | Jan 10 2023 | Secundaria | 84 |
| 13 | 1940 | 15 | Katinka McMyler Lowdham | Jan 07 1984 | Feb 21 2023 | Licenciatura | 84 |
| 14 | 1312 | 15 | Dag Surridge Beveridge | Sep 20 1968 | Dec 17 2023 | Licenciatura | 84 |
| 15 | 1504 | 15 | Randa Rennison Mackin | Dec 23 2002 | Nov 12 2023 | Posgrado | 84 |
| 16 | 1116 | 15 | Christos Mehmet Akers | Nov 23 1972 | Jan 20 2023 | Licenciatura | 84 |

16. Productos de caja que vendieron más de 15 unidades

```
-- 16. Productos de caja que vendieron más de 15 unidades
select idsucursal, idproducto, nombre, marca, presentacion, to_char(precio, 'LFM999'),
       max(cantidad) as cantidad
from (perecedero natural join contenerperecedero) natural join venta
where presentacion = 'Caja'
group by idsucursal, idproducto
having max(cantidad) > 15
order by cantidad desc;
```

Resultados:

| | idsucursal integer | idproducto integer | nombre character varying | marca character varying | presentacion character varying | to_char text | cantidad integer |
|----|-----------------------|-----------------------|-----------------------------|----------------------------|-----------------------------------|-----------------|---------------------|
| 1 | 15 | 279 | Jamón serrano | Hormel | Caja | \$453 | 20 |
| 2 | 5 | 240 | Codorniz | Tyson | Caja | \$371 | 20 |
| 3 | 12 | 168 | Frambuesas | Bimbo | Caja | \$105 | 20 |
| 4 | 6 | 385 | Anchoas | BelGioioso | Caja | \$190 | 20 |
| 5 | 14 | 310 | Cilantro | Chobani | Caja | \$163 | 20 |
| 6 | 1 | 573 | Morcilla | Kerrygold | Caja | \$346 | 20 |
| 7 | 6 | 812 | Huevos | Sancor | Caja | \$457 | 20 |
| 8 | 8 | 127 | Comino | Smithfield | Caja | \$332 | 20 |
| 9 | 12 | 270 | Butifarra | Arnold | Caja | \$362 | 20 |
| 10 | 13 | 643 | Gallina Cornish | Sara Lee | Caja | \$498 | 20 |
| 11 | 4 | 900 | Wintergreen | Oikos | Caja | \$183 | 19 |
| 12 | 6 | 181 | Romero | rident Seafoods | Caja | \$110 | 19 |
| 13 | 14 | 816 | Sobrasada | Tyson | Caja | \$421 | 19 |
| 14 | 1 | 872 | Jabalí | Alquería | Caja | \$139 | 19 |
| 15 | 10 | 125 | Rosquillas | Plugrá | Caja | \$435 | 19 |

17. Gerentes con menor sueldo de cada sucursal

```
-- 17. Los Gerentes con menor sueldo de cada sucursal
select idpersona, concat(nombre, ' ', apaterno, ' ', amaterno) as nombre,
       gerente.idsucursal, to_char(salario, 'LFM999,999') as salario,
       to_char(fechacontratacion, 'Mon dd yyyy') as fechaContratacion, genero
from (select idsucursal, min(salario) as minimo
      from gerente g
      group by idsucursal) as saldosMinSucursal, gerente
where concat(saldosMinSucursal.idsucursal, saldosMinSucursal.minimo) =
       concat(gerente.idsucursal, gerente.salario)
order by idsucursal;
```

Resultados:

| | idpersona [PK] integer | nombre text | idsucursal integer | salario text | fechacontratacion text | genero character |
|----|---------------------------|---------------------------|-----------------------|-----------------|---------------------------|---------------------|
| 1 | 3586 | Nonna Iacifo Chipperfi... | 1 | \$11,582 | Nov 04 2000 | F |
| 2 | 3892 | Stearn Ormston Di Biasi | 2 | \$11,121 | May 30 2008 | M |
| 3 | 3201 | Bernadine Caines Holme | 3 | \$12,637 | Oct 16 2002 | F |
| 4 | 3230 | Claudius Thormwell Sw... | 4 | \$12,052 | Oct 05 2017 | M |
| 5 | 3177 | Walsh Burke Yegorkin | 5 | \$10,128 | Aug 26 2008 | M |
| 6 | 3098 | Phineas Noice Silverwo... | 6 | \$10,453 | Aug 12 2021 | M |
| 7 | 3429 | Adlai Osorio Krop | 7 | \$10,052 | Jun 09 2007 | M |
| 8 | 3664 | Maurine Rodway Maco... | 8 | \$10,540 | Oct 27 2014 | F |
| 9 | 3813 | Nellie Casarili Oxtan | 9 | \$10,552 | Jan 08 2016 | F |
| 10 | 3640 | Raff Chaperling House | 10 | \$10,281 | Jul 15 2001 | M |
| 11 | 3688 | Kelli Clow Chandler | 11 | \$13,249 | Aug 15 2022 | F |
| 12 | 3740 | Tuck Cranfield Jerche | 12 | \$11,500 | Dec 10 2010 | M |
| 13 | 3105 | Ward Ivasechko Hadkins | 13 | \$10,619 | Sep 05 2019 | M |
| 14 | 3215 | Hurley Bleakley McCrillis | 14 | \$10,249 | Nov 27 2002 | M |
| 15 | 3754 | Jacky Amar Snadden | 15 | \$11,050 | Aug 16 2009 | F |

18. Nombre, ciudad, fechaNacimiento de los clientes que hayan nacido entre el 21 de febrero de 1961 y el 24 de septiembre de 1987

```
--18. Obtener nombre, ciudad y fechaNacimiento de los clientes
-- que hayan nacido entre el 21 de febrero de 1961 y el 24 de sept de 1987
select nombre,ciudad,fechanacimiento
from cliente
where fechanacimiento > '1961-02-20' and fechanacimiento < '1987-09-25';
```

Resultados:

| | nombre character varying | ciudad character varying | fechanacimiento date |
|----|-----------------------------|-----------------------------|-------------------------|
| 1 | Harriot | Matamoros | 1969-09-22 |
| 2 | Weber | Teziutlán | 1971-09-11 |
| 3 | Pincus | Saltillo | 1971-09-16 |
| 4 | Meagan | Zacatecas | 1985-01-20 |
| 5 | Hedwiga | Guaymas | 1962-02-16 |
| 6 | Phillis | Zacatecas | 1972-04-14 |
| 7 | Sherwynd | Fresnillo | 1979-03-05 |
| 8 | Van | Fresnillo | 1975-03-31 |
| 9 | Sunny | Monterrey | 1977-02-01 |
| 10 | Bertine | Papantla de Olarte | 1971-05-21 |
| 11 | Merrill | Papantla de Olarte | 1982-08-19 |
| 12 | Collette | Matamoros | 1976-09-03 |
| 13 | Wynn | Piedras Negras | 1987-04-08 |
| 14 | Rayna | Toluca | 1987-04-25 |
| 15 | Flo | Cuernavaca | 1978-12-06 |

19. Cliente cuyo nombre inicie con L y vivan en alguna ciudad que comience con C

```
--19. Clientes cuyo nombre inicie con L y vivan en alguna ciudad que comience con C
select *
from cliente
where nombre ilike 'L%' and
      (ciudad ilike 'C%');
```

Resultados:

| | Idpersona [PK] integer | nombre character varying | apaterno character varying | amaterno character varying | fechanacimiento date | cp integer | ciudad character varying |
|----|---------------------------|-----------------------------|-------------------------------|-------------------------------|-------------------------|---------------|-----------------------------|
| 1 | 107 | Layne | Blaker | Gauche | 1970-07-23 | 64598 | Chilpancingo de los Bravo |
| 2 | 164 | Lauren | Smithies | Acome | 1988-05-21 | 41370 | Coatzacoalcos |
| 3 | 272 | Lanna | Oldaker | Oade | 1981-09-01 | 96059 | Chilpancingo de los Bravo |
| 4 | 288 | Leona | Whichelow | Muddle | 1987-10-10 | 68812 | Ciudad Madero |
| 5 | 341 | Lucine | Danaher | Gladdish | 1991-04-27 | 16963 | Coatzacoalcos |
| 6 | 362 | Leo | Searl | Scurrah | 1989-05-07 | 19913 | Ciudad Juárez |
| 7 | 569 | Lucho | Kellaway | Emtage | 1978-08-14 | 63927 | Ciudad de México |
| 8 | 599 | Lillis | Ridgley | Corradi | 1981-02-01 | 66536 | Cancún |
| 9 | 624 | Laurence | Henrion | Heavyside | 1984-03-26 | 24588 | Chihuahua |
| 10 | 651 | Leland | McMonnies | Shillom | 1971-12-24 | 80326 | Chihuahua |
| 11 | 660 | Lesley | Roncelli | Strowther | 1960-07-14 | 66962 | Coatzacoalcos |
| 12 | 679 | Libby | Bohje | Beernaert | 1979-11-08 | 77438 | Ciudad de México |
| 13 | 908 | Lezley | Browne | Cramond | 2002-09-29 | 13469 | Cancún |
| 14 | 976 | Livvy | Iwanicki | Pogosian | 1999-02-22 | 65116 | Ciudad Obregón |



20. Compra de mayor valor de cada cliente

```
-- 20. La compra de mayor valor de cada cliente
select nombre, apaterno, amaterno, maxima from (select idcliente, max(cantidad*precio) as maxima
from contenerperecedero natural join perecedero natural join venta group by idcliente)
as maximos join cliente on cliente.idpersona = maximos.idcliente order by maximos.maxima desc;
```

Resultados:

| | nombre character varying | apaterno character varying | amaterno character varying | maxima integer |
|----|-----------------------------|-------------------------------|-------------------------------|-------------------|
| 1 | Cyndy | Cassley | Sawart | 9960 |
| 2 | Kirby | Dawtre | Cullingworth | 9860 |
| 3 | Juanita | Haselwood | Wagner | 9620 |
| 4 | Wallis | Meegin | Grindell | 9600 |
| 5 | Franky | Tilne | Fairbourne | 9560 |
| 6 | Corly | Daouze | Fawloe | 9462 |
| 7 | Flo | Cottingham | Geraldez | 9460 |
| 8 | Chrisse | Sprague | McIlvaney | 9140 |
| 9 | Alethea | Caudelier | Targetter | 9120 |
| 10 | Rowan | Gittus | Studdard | 9082 |
| 11 | Anette | Eastham | Giorgi | 9060 |
| 12 | Corri | Elch | Mackie | 8964 |
| 13 | Grier | Almey | Boggish | 8960 |
| 14 | Liesa | Bacher | Jantot | 8892 |
| 15 | Terese | Dower | Jewise | 8873 |
| 16 | Allison | Gosford | Clapshaw | 8820 |

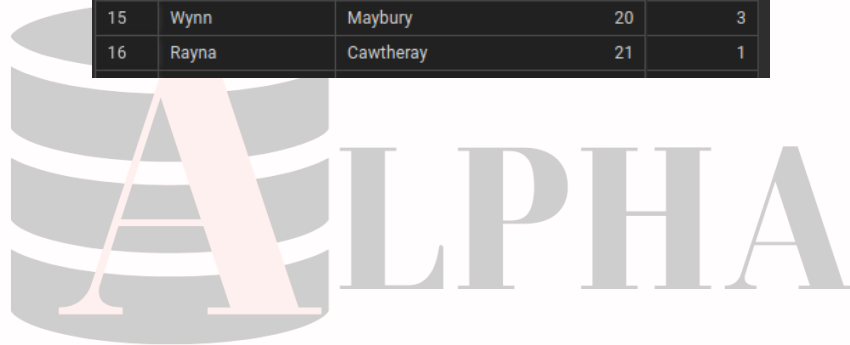


21. Número de veces que ha comprado cada cliente en El Gran Abarrotero

```
-- 21. Número de veces que ha comprado cada cliente en El Gran Abarrotero
select cliente.nombre, cliente.apaterno, cliente.idpersona, count(idventa) as veces from venta join cliente
on cliente.idpersona = venta.idcliente group by cliente.idpersona order by cliente.idpersona;
```

Resultados:

| | nombre character varying | apaterno character varying | idpersona [PK] integer | veces bigint |
|----|-----------------------------|-------------------------------|---------------------------|-----------------|
| 1 | Harriot | Fallis | 1 | 1 |
| 2 | Laurie | Tremayne | 4 | 1 |
| 3 | Meagan | Flaunders | 6 | 1 |
| 4 | Gayelord | Kerrod | 7 | 3 |
| 5 | Mufinella | Hearsey | 8 | 1 |
| 6 | Hedwiga | Eck | 9 | 1 |
| 7 | Phillis | Stroyan | 10 | 1 |
| 8 | Sherwynd | Ritchard | 11 | 2 |
| 9 | Sunny | Saffell | 13 | 2 |
| 10 | Kirby | Dawtre | 14 | 2 |
| 11 | Bertine | Senior | 15 | 2 |
| 12 | Merrill | Demann | 16 | 2 |
| 13 | Collette | Coatham | 18 | 2 |
| 14 | Shepherd | Durtnall | 19 | 1 |
| 15 | Wynn | Maybury | 20 | 3 |
| 16 | Rayna | Cawtheray | 21 | 1 |



22. Número de cajeros que han sido contratados durante el mes de mayo de cada sucursal en orden descendente. (al comienzo de muestra el número mayor de contratados)

```
-- 22. Número de cajeros que han sido contratados durante el mes de mayo en alguna sucursal en
-- orden descendente
select contratados from (
  select cajero.idsucursal, count(idpersona) as contratados
  from cajero
  join sucursal on cajero.idsucursal = sucursal.idsucursal
  where cajero.fechacontratacion between DATE '2023-05-01' and DATE '2023-05-31'
  group by cajero.idsucursal
) as contrataciones
order by contratados desc;
```

Resultados:

| | contratados bigint |
|----|-----------------------|
| 1 | 9 |
| 2 | 8 |
| 3 | 6 |
| 4 | 6 |
| 5 | 6 |
| 6 | 6 |
| 7 | 5 |
| 8 | 5 |
| 9 | 5 |
| 10 | 4 |
| 11 | 3 |
| 12 | 3 |
| 13 | 3 |
| 14 | 2 |
| 15 | 1 |



23. Promedios del sueldo de los cajeros por sucursal.

```
-- 23. Promedio del sueldo de los cajeros por sucursal
select promedio from (
  select sucursal.idsucursal, avg(cajero.sueldo) as promedio
  from cajero
  join sucursal on cajero.idsucursal = sucursal.idsucursal
  group by sucursal.idsucursal
) as promedios;
```

Resultados:

| | promedio numeric |
|----|---------------------|
| 1 | 27198.750000000000 |
| 2 | 26841.555555555556 |
| 3 | 27968.692307692308 |
| 4 | 26542.414285714286 |
| 5 | 27736.676056338028 |
| 6 | 28997.016949152542 |
| 7 | 26982.065573770492 |
| 8 | 28116.452054794521 |
| 9 | 28732.253164556962 |
| 10 | 29911.593750000000 |
| 11 | 26226.656716417910 |
| 12 | 28321.400000000000 |
| 13 | 23781.629032258065 |
| 14 | 28443.750000000000 |
| 15 | 26411.573529411765 |



24. Número de ventas por sucursal

```
-- 24. Número de ventas por sucursal.  
select idsucursal, count(idventa) as ventas from venta natural join sucursal  
group by idsucursal order by idsucursal ;
```

Resultados:

| | idsucursal integer | ventas bigint |
|----|-----------------------|------------------|
| 1 | 1 | 67 |
| 2 | 2 | 64 |
| 3 | 3 | 50 |
| 4 | 4 | 58 |
| 5 | 5 | 74 |
| 6 | 6 | 69 |
| 7 | 7 | 62 |
| 8 | 8 | 73 |
| 9 | 9 | 65 |
| 10 | 10 | 83 |
| 11 | 11 | 68 |
| 12 | 12 | 72 |
| 13 | 13 | 51 |
| 14 | 14 | 60 |
| 15 | 15 | 84 |



FUNCIONES Y DISPARADORES

DISPARADORES

1. Disparador que funciona para modificar las inserciones en la tabla Cliente de tal forma que cuando la ciudad del cliente se escribe como "Ciudad de México" se modificará a CDMX

```
-- 1. Disparador que cambia las inserciones cuando se escribe 'Ciudad de México' a 'CDMX'
-- Crear el trigger
CREATE OR REPLACE FUNCTION actualizar_ciudad_cliente()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.ciudad = 'Ciudad de México' THEN
        NEW.ciudad := 'CDMX';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

-- Asociar el trigger a la tabla "cliente"
CREATE TRIGGER actualizar_ciudad_cliente_disp
BEFORE INSERT OR UPDATE ON cliente
FOR EACH ROW
EXECUTE FUNCTION actualizar_ciudad_cliente();
```

Para probarlo insertamos un nuevo cliente, cuya ciudad en efecto sea "Ciudad de México"

```
VALUES (4001, 'Harriot', 'Fallis', 'Piet', '1969-09-22', 68286, 'Ciudad de México')
```

Luego hacemos la consulta de la información del cliente con el idPersona 4001 y vemos que en efecto, la ciudad se encuentra ahora escrita como "CDMX"

| | id | nombre | apaterno | amaterno | fechanacimiento | cp | ciudad |
|---|---------|-------------------|-------------------|-------------------|-----------------|---------|-------------------|
| | integer | character varying | character varying | character varying | date | integer | character varying |
| 1 | 4001 | Harriot | Fallis | Piet | 1969-09-22 | 68286 | CDMX |

2. Disparador que se encarga de verificar que si los cajeros tienen la cadena "Licenciatura" en su escolaridad entonces su salario no puede ser menor a 2000.00

```
-- 2. Disparador que verifica que el salario mínimo sea de 2000.00
CREATE OR REPLACE FUNCTION verificaSalarioLic() RETURNS TRIGGER
AS $$
DECLARE
    esco varchar(20);
BEGIN
    IF(TG_OP = 'INSERT' or TG_OP = 'UPDATE') THEN
        SELECT escolaridad INTO esco FROM cajero
        WHERE idpersona = new.idpersona;
        IF (esco = 'Licenciatura' AND new.salario < 2000) THEN
            RAISE EXCEPTION 'El saldo minimo en % debe ser $2,000.00', esco;
        END IF;
    END IF;
    RETURN null;
END;
$$
LANGUAGE plpgsql;

-- Asociamos el trigger a la tabla cajero
CREATE TRIGGER saldoLicenciado
AFTER INSERT OR UPDATE ON cajero
FOR EACH ROW
EXECUTE PROCEDURE verificaSalarioLic();
```

Ahora, pongámoslo a prueba intentando insertar un nuevo cajero que gane 1900 con escolaridad= 'Licenciatura', tal como lo esperábamos no nos lo permite y arroja el mensaje acordado.

```
ERROR: El saldo minimo en
Licenciatura debe ser $2,000.00
CONTEXT: PL/pgSQL function
verificasalariolic() line 9 at
RAISE
```

FUNCIONES

1. Función que devuelve el id, nombre y apellidos de los cajeros que se llamen de cierta forma en determinada sucursal.

```
-- 1. Función que devuelve toda la información de determinado empleado
CREATE OR REPLACE FUNCTION trabajadores(nom varchar, num bigint) RETURNS
  TABLE(nombre varchar(70), materno varchar(70), paterno varchar(70), id bigint)
AS
$$
  SELECT cajero.nombre, apaterno, amaterno, idpersona
  FROM cajero JOIN sucursal ON cajero.idsucursal = sucursal.idsucursal
  WHERE num = sucursal.idsucursal
  GROUP BY cajero.nombre, apaterno, amaterno, idpersona
  HAVING cajero.nombre LIKE nom;
$$
LANGUAGE SQL;
```

Veámosla puesta en práctica:

| | nombre character varying 🔒 | materno character varying 🔒 | paterno character varying 🔒 | id bigint 🔒 |
|---|-------------------------------|--------------------------------|--------------------------------|----------------|
| 1 | Berty | Endon | Skrzynski | 1004 |



2. Función que verifica la existencia de algún producto

```
-- 2. Función que verifica la existencia de algún producto
CREATE OR REPLACE FUNCTION existenciaProducto(text,int) RETURNS SETOF "record"
AS $$
DECLARE
    r record;
    tabla text := $1;
BEGIN
    for r in execute 'select nombre,idsucursal, idproducto, cantidad
                    from contar'|| tabla ||' natural join '|| tabla || ' where idproducto = '|| $2
    loop
        return next r;
    end loop;
    return;
END;
$$
LANGUAGE plpgsql;
```

Veámosla puesta en práctica:

| | nombre character varying | idsucursal integer | idproducto integer | cantidad integer |
|---|-----------------------------|-----------------------|-----------------------|---------------------|
| 1 | Frijoles negros chinos | 3 | 1073 | 184 |
| 2 | Frijoles negros chinos | 10 | 1073 | 95 |

