

OCTUBRE 7, 2023

CODER HOUSE

Curso SQL - Comisión: 47350

VENTA MAYORISTA DE PRODUCTOS ALIMENTICIOS

Alumno: Mauro Nicolás Rivero

Tutor: Santiago A. González Martín

ÍNDICE

1. Portada	1
2. Índice	2
3. Introducción , situación problemática y tablas	3
3.1 Descripción de Tablas	4
4. Diagramas Entidad-Relación – PPT y MySQL	6
5. Scripts creación de Esquema y Tablas	9
6. Inserción de datos	9
7. Vistas	10
8. Funciones	13
9. Procedimientos de Almacenado	14
10. Triggers	16
11. Ejercicios DCL y TCL	18
12. BackUp	18
13. Herramientas utilizadas	18
14. BI – Gráficos	19
15. Agradecimientos	22

Introducción, Situación Problemática, Objetivo y Tablas

Introducción:

Elegí esta temática porque es un rubro que conozco y es un flujo de información simple con elementos claros y roles bien definidos. El canal mayorista es un vaso comunicante entre las industrias productoras de alimentos y las cadenas minoristas y negocios de cercanía. Posee una estructura física grande para el almacenamiento del stock, una flota de vehículos que se adaptan a los distintos productos comercializados, personal idóneo y una administración acorde al volumen de ventas que maneja.

Situación Problemática:

Como bien dijimos antes, este tipo de negocios tiene en todos sus aspectos un gran tamaño lo que a veces hace difícil su eficiente administración tanto en el personal, stock, control de facturación y mantenimiento de infraestructura.

Objetivo: El objetivo de este trabajo es proveer al mayorista una base de datos robusta y bien administrada, que, junto a informes producidos por el equipo de Business Intelligence aporten valor agregado al negocio como tal y den un panorama más certero a los encargados de las tomas de decisiones dentro del negocio.

Tablas:

Esta lista de tablas que conformarán nuestra base de datos son a nuestro criterio las mínimas e indispensables, eso no quita que, luego vayan incorporándose a medida que vayan surgiendo nuevas necesidades de registro y administración

- Empresa
- Proveedores
- Stock
- Empleados
- Clientes
- Ventas
- Localidad
- Provincias

Descripción de Tablas

Empresa: Esta tabla contendrá en sus campos los datos particulares de la empresa, estos campos llenarán una factura emitida por la empresa, serán usados por ejemplo por los proveedores para facturarle a la empresa y estará en el recibo de sueldo de los empleados.

Nombre Completo	Nombre Abreviado	Tipo de Dato	Clave	Constraint
Número de Cuit	CUIT	Varchar (15)	(PK)	--
Razón Social	Nombre	Varchar (30)		Not null
Dirección Legal	Domicilio	Varchar (50)		Not null
ID Localidad	Loc.	int	(FK)	--
ID Provincia	Prov.	Int	(FK)	--
Teléfono	Tel.	Varchar (20)		Not null

Proveedores: Esta tabla contendrá en sus campos los datos particulares de los proveedores, los mismo pueden ser usados para identificar los productos por ejemplo y en la parte contable para temas impositivos.

Nombre Completo	Nombre Abreviado	Tipo de Dato	Clave	Constraint
ID Proveedor	Proveedor	Int	(PK)	--
Número de Cuit	CUIT	Varchar (15)		Not null
Razón Social	Nombre	Varchar (30)		Not null
Dirección Legal	Domicilio	Varchar (50)		Not null
ID Localidad	Loc.	int	(FK)	--
ID Provincia	Prov.	Int	(FK)	--
Teléfono	Tel.	Varchar (20)		Not null
Productos	Prod.	Varchar (30)		Not null

Stock: Esta tabla contendrá en sus campos los artículos que se encuentran en stock, se usará para controlar el flujo de los productos que comercializa, por ejemplo.

Nombre Completo	Nombre Abreviado	Tipo de Dato	Clave	Constraint
ID Producto	Prod.	Int	(PK)	--
ID Proveedor	Proveedor	Int	(FK)	--
Nombre Completo	Nombre	Varchar (30)		Not null
Cantidad	Cant.	Int		Not null
Ubicación en Depósito	Sector	Varchar (5)		Not null

Empleados: Esta tabla contendrá en sus campos los datos de los empleados de la empresa, se usarán para componer la plantilla actual de la empresa, el rol que desempeña cada uno y acciones comerciales como obsequios por día del cumpleaños, etc.

Nombre Completo	Nombre Abreviado	Tipo de Dato	Clave	Constraint
ID Empleado	Emp.	Int	(PK)	--
Nombre Completo	Nombre	Varchar (30)		Not null
Domicilio	Domicilio	Varchar (50)		Not null
ID Localidad	Loc.	int	(FK)	--
ID Provincia	Prov.	Int	(FK)	--
Teléfono	Tel.	Varchar (20)		Not null
Puesto Laboral	Cargo	Varchar (30)		Not null
Fecha de Nacimiento	Cumpleaños	date		Not null
Fecha de Ingreso	FechaIngreso	Date		Not null

Clientes: Esta tabla contendrá en sus campos los datos de los clientes de la empresa, estos datos se usarán en la facturación de la empresa y también estarán disponibles para acciones comerciales puntuales como saludos por cumpleaños, etc.

Nombre Completo	Nombre Abreviado	Tipo de Dato	Clave	Constraint
ID Cliente	Cliente	Int	(PK)	--
Número de Cuit	CUIT	Varchar (15)		Not null
Nombre Completo	Nombre	Varchar (30)		Not null
Domicilio	Domicilio	Varchar (50)		Not null
ID Localidad	Loc.	Int	(FK)	--
ID Provincia	Prov.	Int	(FK)	--
Teléfono	Tel.	Varchar (30)		Not null
Fecha de Nacimiento	Cumpleaños	date		Not null

Ventas: Esta tabla contendrá en sus campos los datos de las ventas efectuadas por la empresa, desde aquí se harán estimaciones contables, estadísticas, índices, métricas, etc.

Nombre Completo	Nombre Abreviado	Tipo de Dato	Clave	Constraint
Número de Factura	N° Fact.	Varchar(20)	(PK)	Autoincrement
Condición Fiscal	Cond.Fiscal	Varchar(20)		--
ID Producto	Prod.	Int	(FK)	--
ID Clientes	Cliente	Int	(FK)	--
Fecha de Venta	Fecha	Date		Not null
Cantidad	Cant.	Int		Not null
Total Facturado	Monto	Int		Not null

Localidad: Esta tabla contendrá en sus campos las distintas localidades del país para ayudar a normalizar de forma más eficiente las demás tablas que contengan Localidad como campos

Nombre Completo	Nombre Abreviado	Tipo de Dato	Clave	Constraint
ID Localidad	Localidad	Int	(PK)	--
ID Provincia	Prov.	Int	(FK)	--
Nombre Localidad	Nombre	Varchar (30)		Not null
Código Postal	C. P.	Varchar (10)		Not null

Provincia: Esta tabla contendrá en sus campos las distintas provincias del país para ayudar a normalizar de forma más eficiente las demás tablas que contengan Provincia como campos

Nombre Completo	Nombre Abreviado	Tipo de Dato	Clave	Constraint
ID Provincia	Prov.	Int	(PK)	--
Nombre Provincia	Nombre	Varchar (30)		Not null

Diagrama Entidad-Relación

El diagrama entidad-relación es una representación gráfica de la estructura de una base de datos donde se pueden apreciar 4 elementos destacados que nos ayudan a entenderlo. Los Rectángulos representan las entidades que componen la base de datos, ósea la tablas, estas a su vez tienen atributos que son representados por Elipses. Cada entidad está unida a otra por Líneas que indican la cardinalidad y relación entre estas. Por último los Rombos identifican las acciones que involucran a 2 entidades

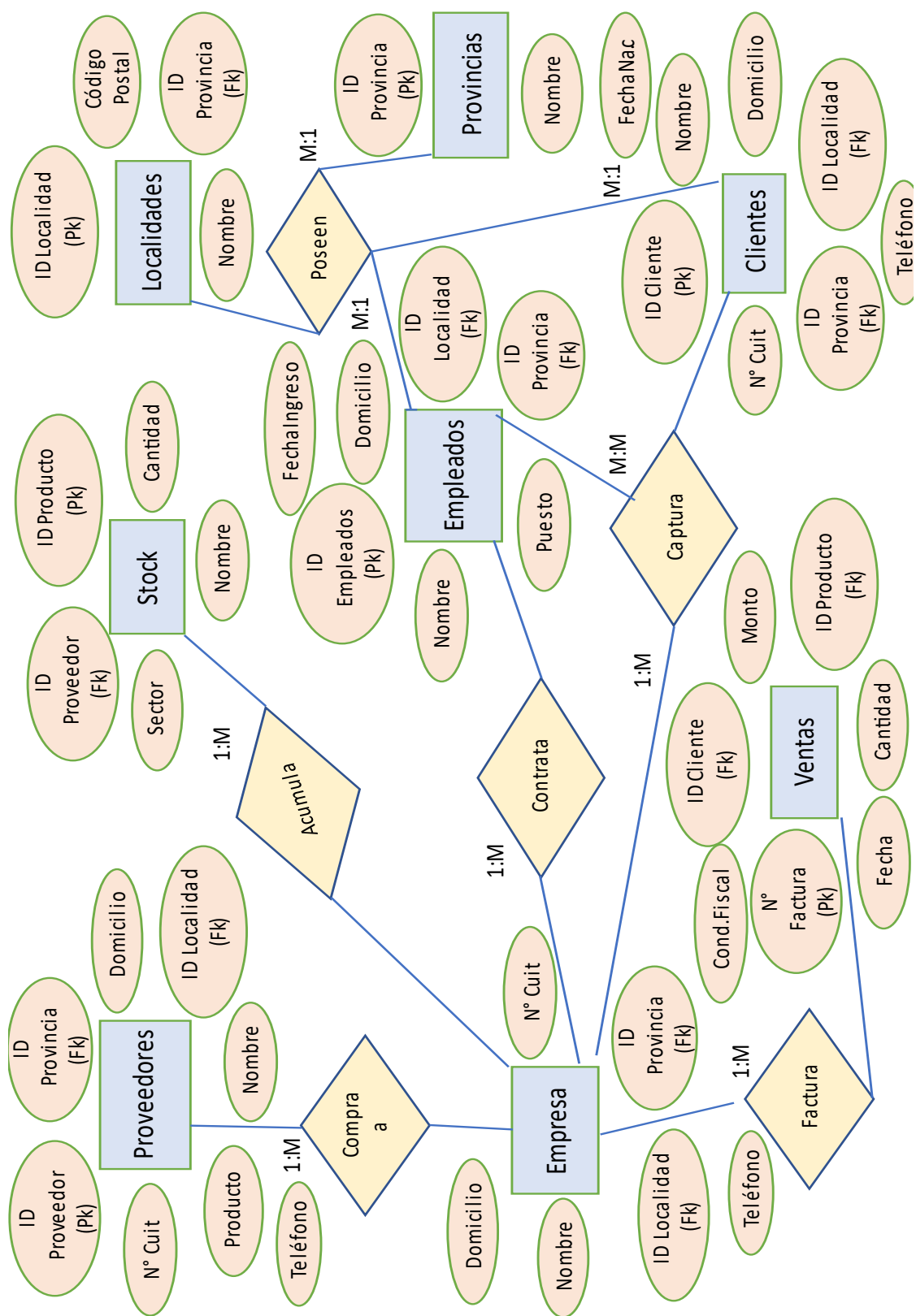
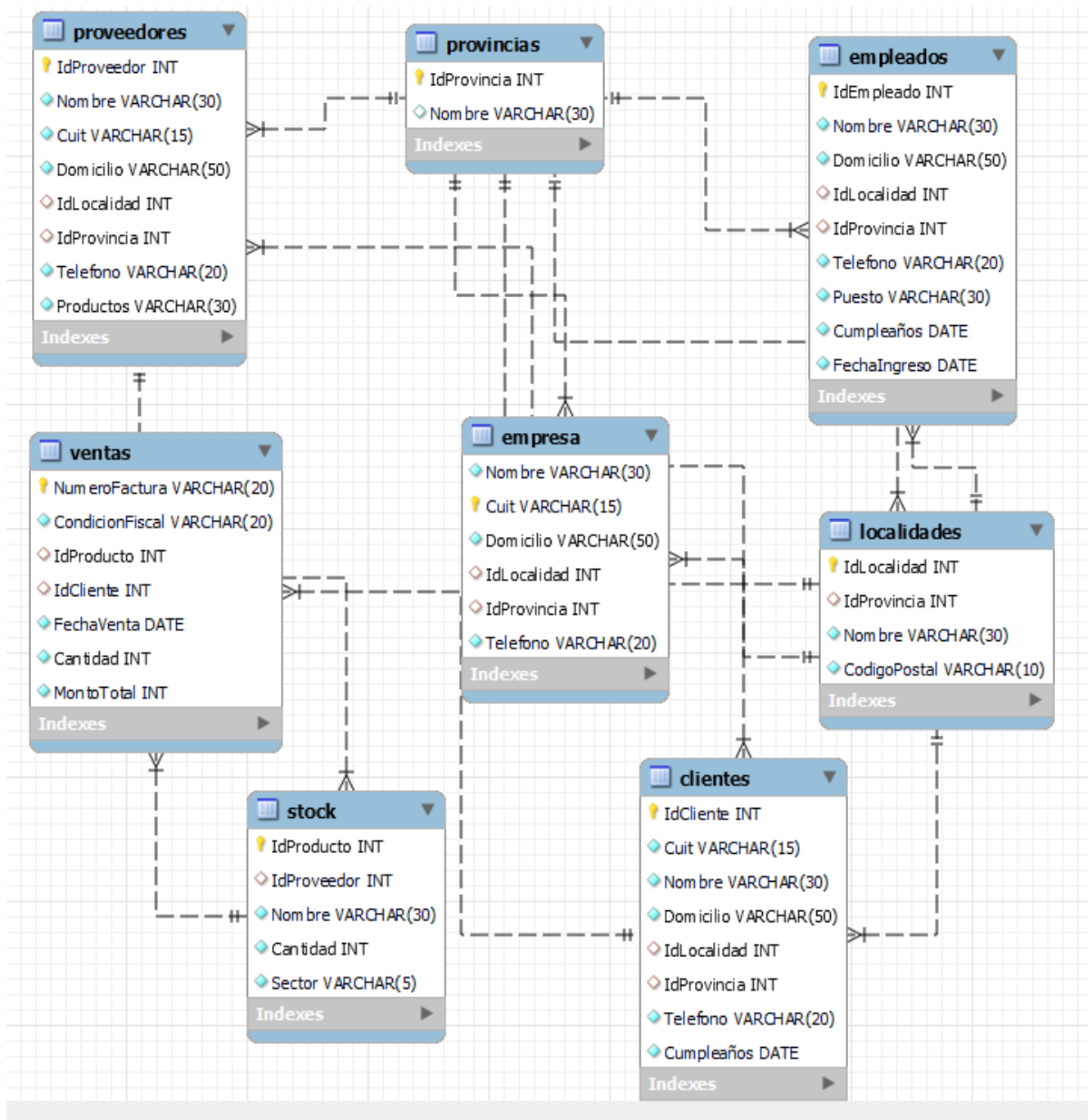


Diagrama Entidad-Relación (Workbench)



Scripts creación de Esquema y Tablas

Los scripts de creación del Esquema y las diferentes Tablas del proyecto final están alojados en mi repositorio personal de GitHub, a continuación, adjunto el link para que accedan a los archivos. Por favor leer el archivo Readme.md para interiorizarse sobre su contenido y uso.

Link: <https://github.com/MauroNicolasRivero/Idea-Rivero-SQL>

Inserción de Datos

La inserción de datos se realizó utilizando la función “Table Data Import Wizard” de **MySQL Workbench**, el instructivo paso a paso está alojado en mi repositorio personal de GitHub, así como los archivos en formato plano .CSV, a continuación, adjunto el link para que accedan al archivo. Por favor leer el archivo Readme.md para interiorizarse sobre su contenido y uso.

Link: <https://github.com/MauroNicolasRivero/Idea-Rivero-SQL>

Vistas

- Cant_empl_puesto
- Minima_cant_stock
- Procedencia_clientes
- Proveedores_locales
- Top_five_clientes

Cant_empl_puesto: Esta vista trae información de la tabla “Empleados”. Su objetivo es conocer a través de una consulta sencilla la cantidad de empleados por puesto que la empresa tiene actualmente en su plantilla laboral.

Esta vista está construida sobre la columna “Puesto”, la misma es usada en dos ocasiones. La primera es haciendo un “Count()”, función de agregación, para obtener la cantidad de empleados que serán agrupados ,con la sentencia “Group By”, por categoría. La segunda ocasión que recurrimos a la columna “Puesto” es para obtener las etiquetas de cada puesto para identificar los números obtenidos anteriormente.

A continuación, adjuntamos como se ve la tabla usando el “MYSQL - Workbench”



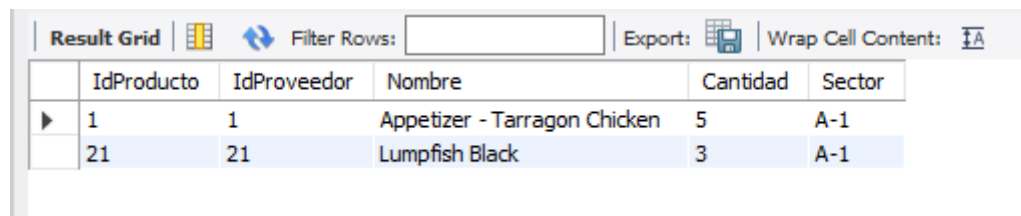
The screenshot shows the MySQL Workbench interface with the 'Result Grid' tab selected. It displays the output of a query, which is the 'Cant_Empleados_Puesto' view. The table has two columns: 'Puesto' and 'Cant_Empleados_Puesto'. The data is as follows:

Puesto	Cant_Empleados_Puesto
Recepción	9
Atención al Cliente	9
Ventas	27
Caja	18
Post Venta	9
Maestranza	8
Depósito	8
Administración	8
Jefe del Personal	1
Gerente	2
Director	1

Minima_cant_stock: Esta vista trae información de la tabla “Stock”. Su objetivo es conocer a través de una consulta sencilla que productos actualmente tienen una cantidad de unidades igual o inferior a 5(número puesto de forma aleatoria, se puede ajustar al modelo de negocio)

Esta vista está construida sobre todas las columnas de la tabla, pero usamos un filtro sobre la columna “Cantidad” en la sentencia “Where” donde le pedimos a la consulta que solamente nos devuelva los productos cuya cantidad sea igual o menor al número indicado.

A continuación, adjuntamos como se ve la tabla usando el “MYSQL - Workbench”

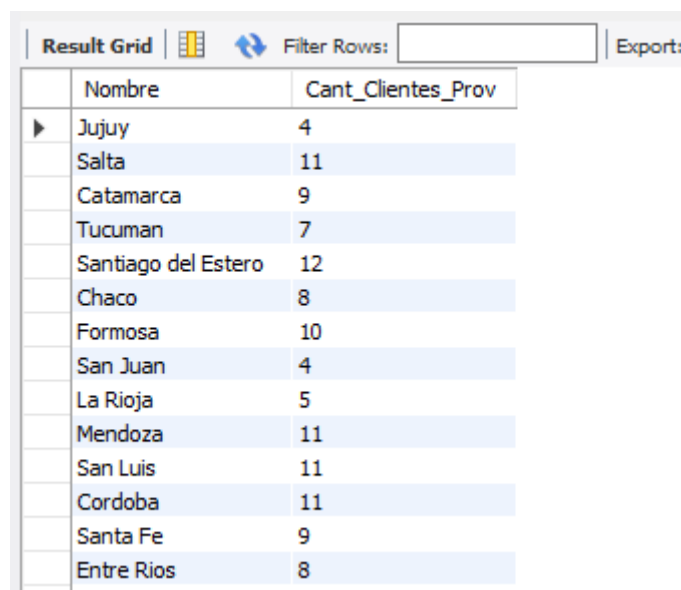


	IdProducto	IdProveedor	Nombre	Cantidad	Sector
▶	1	1	Appetizer - Tarragon Chicken	5	A-1
	21	21	Lumpfish Black	3	A-1

Procedencia_clientes: Esta vista trae información de las tablas “Clientes” y “Provincias”. Su objetivo es conocer a través de una consulta sencilla, la cantidad de Clientes por Provincia que la empresa tiene actualmente en su plantilla de clientes.

Esta vista está construida sobre las columnas “Nombre” e “idProvincia” de la tabla “Provincias” y de la columna “idProvincia” de la tabla “Clientes”. Esta es una consulta de complejidad media ya que debemos recurrir a la palabra reservada “Join”, para unir información de dos tablas, y así lograr el objetivo. Usamos la función de agregación “Count()” para contar la cantidad de clientes que tengan el mismo “idProvincia” y luego a través de la sentencia “Group By” agrupamos el resultado bajo el Nombre de la Provincia correspondiente.

A continuación, adjuntamos como se ve la tabla usando el “MYSQL - Workbench”



	Nombre	Cant_Clientes_Prov
▶	Jujuy	4
	Salta	11
	Catamarca	9
	Tucuman	7
	Santiago del Estero	12
	Chaco	8
	Formosa	10
	San Juan	4
	La Rioja	5
	Mendoza	11
	San Luis	11
	Cordoba	11
	Santa Fe	9
	Entre Rios	8

Proveedores_locales: Esta vista trae información de las tablas “Proveedores” y “Empresa”. Su objetivo es conocer a través de una consulta sencilla que proveedores son oriundos de la misma Provincia de la cual es la empresa.

Esta vista está construida sobre todas las columnas de la tabla “Proveedores” y sobre la columna “idProvincia” de la tabla “Empresa”. Es una consulta que tiene en su condición una subconsulta desde la cual obtenemos el número de “idProvincia” de la empresa que usaremos como filtro en la cláusula “Where” para conocer solamente aquellos proveedores cuyo “idProvincia” sea igual al “idProvincia” de la empresa

A continuación, adjuntamos como se ve la tabla usando el “MYSQL - Workbench”

Result Grid						
Filter Rows: <input type="text"/>						
Export: <input type="text"/>						
Wrap Cell Content: <input type="text"/>						
	IdProveedor	Nombre	Cuit	Domicilio	IdLocalidad	IdProvincia
▶	14	Wehner, Welch and Saterfield	30-33742017-2	3432 Anhalt Terrace	52	12
	26	Johnston Inc	30-55933174-0	263 Michigan Circle	92	12
	48	Feil Group	30-43678533-1	78693 Melody Circle	20	12

Top_five_clientes: Esta vista trae información de las tablas “Clientes” y “Ventas”. Su objetivo es conocer a los cinco Clientes que más veces han efectuado compras a la empresa, ordenándolos de manera descendente.

Esta vista está construida sobre las columnas “idClientes” y “Nombre” de la tabla “Clientes” y de la columna “idclientes” de la tabla “Ventas”. Es una consulta de complejidad media ya que utiliza varias cláusulas en su composición.

Primero seleccionamos de la tabla “Clientes” todos los idClientes y sus respectivos nombres, luego a través de la palabra reservada “Join” y usando la función de agregación “Count()” y la cláusula “Group By”, para contar y agrupar los idClientes, unimos una columna que traerá la cantidad de compras hechas por los idclientes identificados en la tabla “Ventas”. La sentencia “Order By” nos ayuda a ordenar de forma descendente el resultado de ese count y con la palabra reservada “Limit” le indicamos que nos devuelva solamente los primeros 5 registros.

A continuación, adjuntamos como se ve la tabla usando el “MYSQL - Workbench”

Result Grid			
Filter Rows: <input type="text"/>			
Export: <input type="text"/>			
	IdCliente	Nombre	Cantidad_Compras
▶	161	Anna-diana Aylmore	13
	47	Lurette Negal	11
	56	Hephzibah Ghidoli	11
	99	Phebe Gratland	10
	15	Levin Hatherley	10

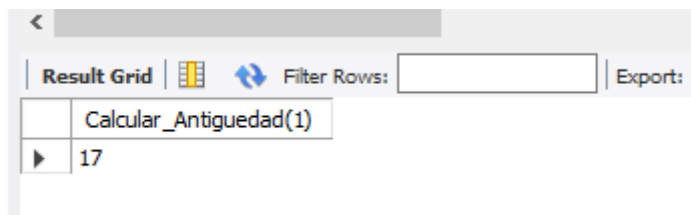
Funciones

- Calcular_Antigüedad
- Nombre_ClientesByCuit

Calcular_Antigüedad: Esta función consume información de la tabla “Empleados”. Su objetivo es calcular de manera sencilla la antigüedad de un empleado.

Es una función con un parámetro de entrada, el id que tiene asignado el empleado en su legajo. A partir de la identificación del mismo, se hace una consulta a la tabla empleados donde utilizando la función de agregación “datediff()” se calcula la fecha actual menos la fecha de incorporación a la empresa, a ese resultado se le aplica otra función para redondear su valor (round()) y se lo almacena dentro de la variable que hemos declarado dentro del cuerpo de la función llamada “Antigüedad”. Para invocar la función debemos usar la palabra reservada “Select” seguida del nombre de la función y el parámetro indicado al principio

A continuación, adjuntamos como se ve la tabla usando el “MYSQL - Workbench”



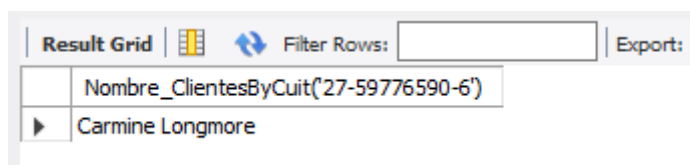
The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with 'Result Grid', a grid icon, a 'Filter Rows' button with a funnel icon, a text input field, and an 'Export' button. Below this, a table displays the results of a query. The first row has a column header 'Calcular_Antigüedad(1)' and a single data row with the value '17'.

Calcular_Antigüedad(1)
17

Nombre_clientesbycuit: Esta función consume información de la tabla “Clientes”. Su objetivo es conocer de manera sencilla el nombre del cliente que tenemos almacenado en nuestra base de datos.

Es una función con un parámetro de entrada, el cuit que tiene asignado el cliente. A partir del ingreso del mismo, se hace una consulta a la tabla clientes donde utilizamos el valor del parámetro como filtro y ese resultado lo almacenamos dentro de la variable que hemos declarado dentro del cuerpo de la función llamada “Nombre_Completo”. Para invocar la función debemos usar la palabra reservada “Select” seguida del nombre de la función y el parámetro indicado al principio

A continuación, adjuntamos como se ve la tabla usando el “MYSQL - Workbench”



The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with 'Result Grid', a grid icon, a 'Filter Rows' button with a funnel icon, a text input field, and an 'Export' button. Below this, a table displays the results of a query. The first row has a column header 'Nombre_ClientesByCuit('27-59776590-6')' and a single data row with the value 'Carmine Longmore'.

Nombre_ClientesByCuit('27-59776590-6')
Carmine Longmore

Procedimientos de Almacenado

- Info_Empleados
- Ingresar_Nva_Loc

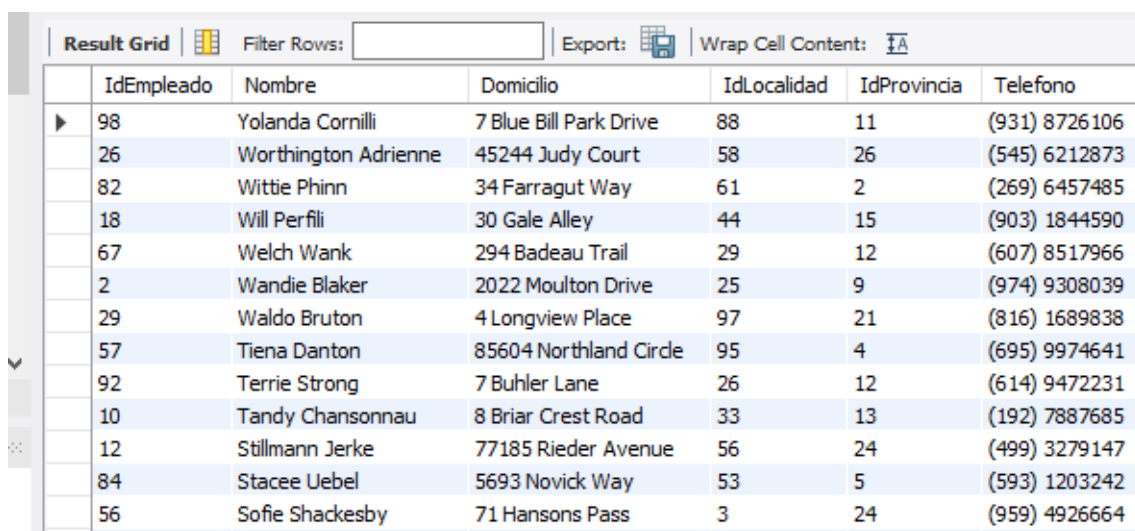
Info_Empleados: Este stored procedure manipula información de la tabla “Empleados”. Su objetivo es poder consumir información de manera sencilla de los distintos campos que componen la tabla, así como también, el orden en el cual se quiere presentar.

Este stored procedure está compuesto por dos parámetros de entrada, el primero es el nombre de la columna, de la tabla empleados que sea de interés, y el segundo el orden con el cual quiero presentar la misma, las opciones son “Asc” y “Desc”.

Dentro del cuerpo del SP. incluimos un condicional que declara una variable llamada “data_orden” que concatenará la información almacenada en caso que NO estén vacíos los parámetros. Luego se declara una segunda variable que concatena una consulta “Select” a la tabla empleados, más el valor de la primera variable. Por último, usando las palabras reservadas “Prepare”, “Execute” y “Deallocate Prepare” ejecutamos la sentencia para obtener los valores de interés.

Para “correr” este SP. debemos invocar la palabra reservada “CALL” seguida del nombre del SP y los parámetros

A continuación, adjuntamos como se ve la tabla usando el “MYSQL – Workbench”



The screenshot shows the MySQL Workbench interface with the 'Result Grid' tab active. It displays a table with 7 columns: IdEmpleado, Nombre, Domicilio, IdLocalidad, IdProvincia, and Telefono. The data is sorted by IdEmpleado in ascending order. The interface includes a 'Filter Rows' search bar, an 'Export' button, and a 'Wrap Cell Content' toggle.

	IdEmpleado	Nombre	Domicilio	IdLocalidad	IdProvincia	Telefono
▶	98	Yolanda Cornilli	7 Blue Bill Park Drive	88	11	(931) 8726106
	26	Worthington Adrienne	45244 Judy Court	58	26	(545) 6212873
	82	Wittie Phinn	34 Farragut Way	61	2	(269) 6457485
	18	Will Perfili	30 Gale Alley	44	15	(903) 1844590
	67	Welch Wank	294 Badeau Trail	29	12	(607) 8517966
	2	Wandie Blaker	2022 Moulton Drive	25	9	(974) 9308039
	29	Waldo Bruton	4 Longview Place	97	21	(816) 1689838
	57	Tiena Danton	85604 Northland Circle	95	4	(695) 9974641
	92	Terrie Strong	7 Buhler Lane	26	12	(614) 9472231
	10	Tandy Chansonau	8 Briar Crest Road	33	13	(192) 7887685
	12	Stillmann Jerke	77185 Rieder Avenue	56	24	(499) 3279147
	84	Stacee Uebel	5693 Novick Way	53	5	(593) 1203242
	56	Sofie Shackesby	71 Hansons Pass	3	24	(959) 4926664

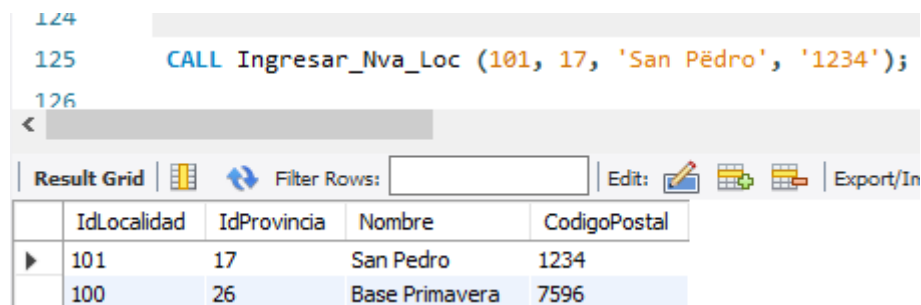
Ingresar_Nva_Loc: Este stored procedure manipula información de la tabla “Localidades”. Su objetivo es poder facilitar el ingreso de una nueva Localidad a la tabla.

Este stored procedure está compuesto por cuatro parámetros de entrada, el primero es el id de la Localidad, el segundo el id de la Provincia, el tercero es el nombre de la Localidad y el cuarto es el Código Postal.

Dentro del cuerpo del SP. incluimos un condicional que, en caso que TODOS los parámetros tengan un valor válido, se ejecute la sentencia “Insert Into” a la tabla “Localidades” y se registre la nueva localidad. Para conocer si el resultado fue exitoso, inmediatamente después se hace una consulta “Select” para ver la variación de la misma.

En caso que algún parámetro llegue vacío, nos mostrará por pantalla un error

A continuación, adjuntamos como se ve la tabla usando el “MYSQL – Workbench”



The screenshot shows the MySQL Workbench interface. At the top, a SQL query is entered in the editor:

```
124  
125 CALL Ingresar_Nva_Loc (101, 17, 'San Pedro', '1234');  
126
```

Below the editor, the 'Result Grid' tab is active, displaying the results of the query. The grid has four columns: IdLocalidad, IdProvincia, Nombre, and CodigoPostal. It shows two rows of data:

	IdLocalidad	IdProvincia	Nombre	CodigoPostal
▶	101	17	San Pedro	1234
	100	26	Base Primavera	7596

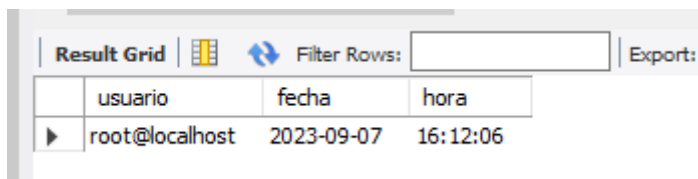
Triggers

- Update_Tabla_Clientes
- Save_Clientes_BK
- Insert_Vtas_BK
- Update_Tabla_Ventas

Update_Tabla_Clientes: Este triggers pertenece a la tabla “Clientes” e interactúa con la tabla de logs creada para esta ocasión llamada “Auditoria_Clientes”. Su objetivo es que ANTES que se actualice un valor en la tabla principal se registre en la tabla auxiliar datos de interés para la empresa, por ejemplo, Quien y Cuando hizo esa actualización.

Su funcionamiento es el siguiente, se crea el trigger dentro de la tabla “Clientes” y se lo condiciona a un evento futuro, en este caso cuando se Actualice la tabla clientes ANTES va a llenar los campos de interés, que nosotros le indicamos dentro del cuerpo del trigger, en la tabla Auditoria_Clientes, en este caso el Usuario, Fecha y Hora de la actualización hecha.

A continuación, adjuntamos como se ve la tabla usando el “MYSQL – Workbench”



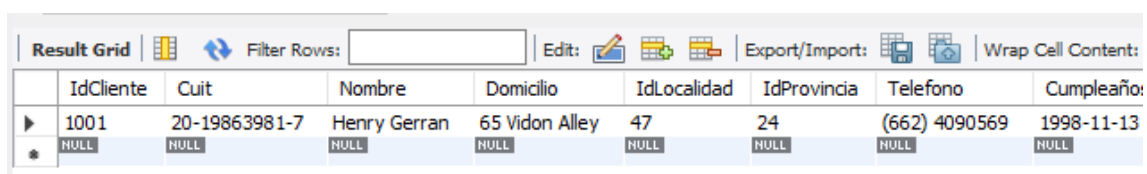
	usuario	fecha	hora
▶	root@localhost	2023-09-07	16:12:06

Save_Clientes_BK: Este Trigger pertenece a la tabla “Clientes” e interactúa con la tabla espejo creada para esta ocasión llamada “Clientes_BK”. Su objetivo es que ANTES que se Elimine un registro de la tabla principal, se guarden los datos originales en la tabla espejo como Back_Up

Su funcionamiento es el siguiente, se crea el Trigger dentro de la tabla “Clientes” y se lo condiciona a un evento futuro, en este caso cuando se quiera Eliminar un registro de la tabla principal ANTES se van a insertar esos datos a eliminar en la tabla de back_up llamada “Clientes_BK” para resguardar su integridad si por error se llegaron a eliminar.

Estos datos se capturan anteponiendo la palabra reservada “OLD” al campo que queremos guardarle su valor original en la tabla de back-Up.

A continuación, adjuntamos como se ve la tabla usando el “MYSQL – Workbench”

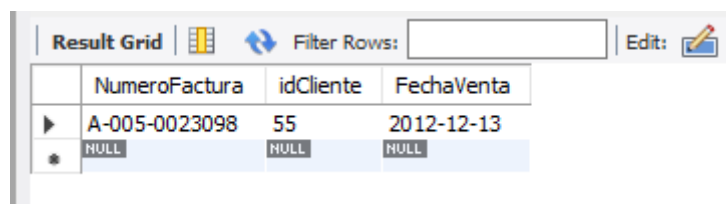


	IdCliente	Cuit	Nombre	Domicilio	IdLocalidad	IdProvincia	Telefono	Cumpleaños
▶	1001	20-19863981-7	Henry Gerran	65 Vidon Alley	47	24	(662) 4090569	1998-11-13
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Insert_Vtas_BK: Este Trigger pertenece a la tabla “Ventas” e interactúa con la tabla de back_up creada para esta ocasión llamada “Ventas_BK”. Su objetivo es que DESPUÉS que se Inserte un registro de la tabla principal, se guarden datos de interés en la tabla de Back_Up.

Su funcionamiento es el siguiente, se crea el Trigger dentro de la tabla “Ventas” y se lo condiciona a un evento futuro, en este caso cuando se Inserte un registro en la tabla principal DESPUÉS de dicha acción se van a insertar también los datos de nuestro interés en la tabla de back_up llamada “Ventas_BK”. Así tendremos un doble control sobre las ventas

A continuación, adjuntamos como se ve la tabla usando el “MYSQL – Workbench”

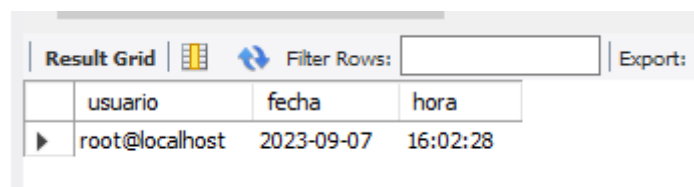


	NumeroFactura	idCliente	FechaVenta
▶	A-005-0023098	55	2012-12-13
✱	NULL	NULL	NULL

Update_Tabla_Ventas: Este triggers pertenece a la tabla “Ventas” e interactúa con la tabla de logs creada para esta ocasión llamada “Auditoria_Ventas”. Su objetivo es que DESPUÉS que se actualice un valor en la tabla principal se registre en la tabla auxiliar datos de interés para la empresa, por ejemplo, Quien y Cuando hizo esa actualización.

Su funcionamiento es el siguiente, se crea el trigger dentro de la tabla “Ventas” y se lo condiciona a un evento futuro, en este caso cuando se Actualice la tabla Ventas, DESPUÉS de dicha acción, va a llenar los campos de interés, que nosotros le indicamos dentro del cuerpo del trigger, en la tabla Auditoria_Ventas, en este caso el Usuario, Fecha y Hora de la actualización hecha.

A continuación, adjuntamos como se ve la tabla usando el “MYSQL – Workbench”



	usuario	fecha	hora
▶	root@localhost	2023-09-07	16:02:28

Ejercicios DCL y TCL

Los ejercicios correspondientes a los Sublenguajes referidos en el título están alojados en mi repositorio personal de GitHub, a continuación, adjunto el link para acceder al archivo. Por favor, leer el archivo Readme.md para interiorizarse sobre su contenido y uso.

Link: <https://github.com/MauroNicolasRivero/Idea-Rivero-SQL>

BackUp

Los ejercicios correspondientes a la confección del Back-Up y su posterior restauración de la base de datos están alojados en mi repositorio personal de GitHub, a continuación, adjunto el link para acceder al archivo. Por favor, leer el archivo Readme.md para interiorizarse sobre su contenido y uso.

Link: <https://github.com/MauroNicolasRivero/Idea-Rivero-SQL>

Herramientas Utilizadas

- MySQL Workbench (Interfaz gráfica para trabajar con SQL)
- Microsoft Word (Confección de informes)
- Microsoft Excel (Manipulación de datos)
- Microsoft Power Point (Creación de Diagrama E-R)
- Microsoft Power BI (Creación de gráficos)
- Google Drive (Almacenamiento de archivos)
- GitHub.com (Repositorio del proyecto final)
- Mockaroo.com (Población de la base de datos)

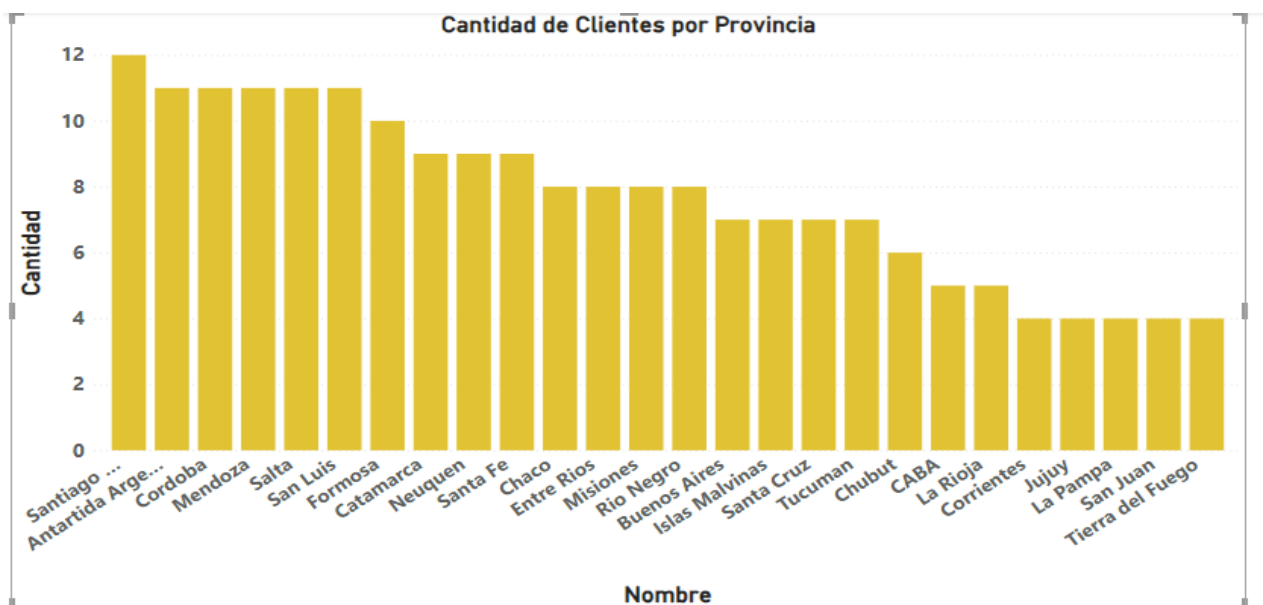
Business Intelligence – Gráficos

En esta sección integraremos la parte de BI al proyecto de creación y administración de la base de datos. La idea es mostrar como un equipo orientado al manejo de herramientas de análisis y visualización de datos puede aportar valor al proyecto donde, a veces, las tablas con muchas filas y columnas no termina de comunicar lo que están representando. Estos gráficos están orientados a la descripción gráfica de algunas de las tablas

Los scripts asociados a cada gráfico están alojados en mi repositorio personal de GitHub, a continuación, adjunto el link para acceder al archivo. Por favor, leer el archivo Readme.md para interiorizarse sobre su contenido y uso.

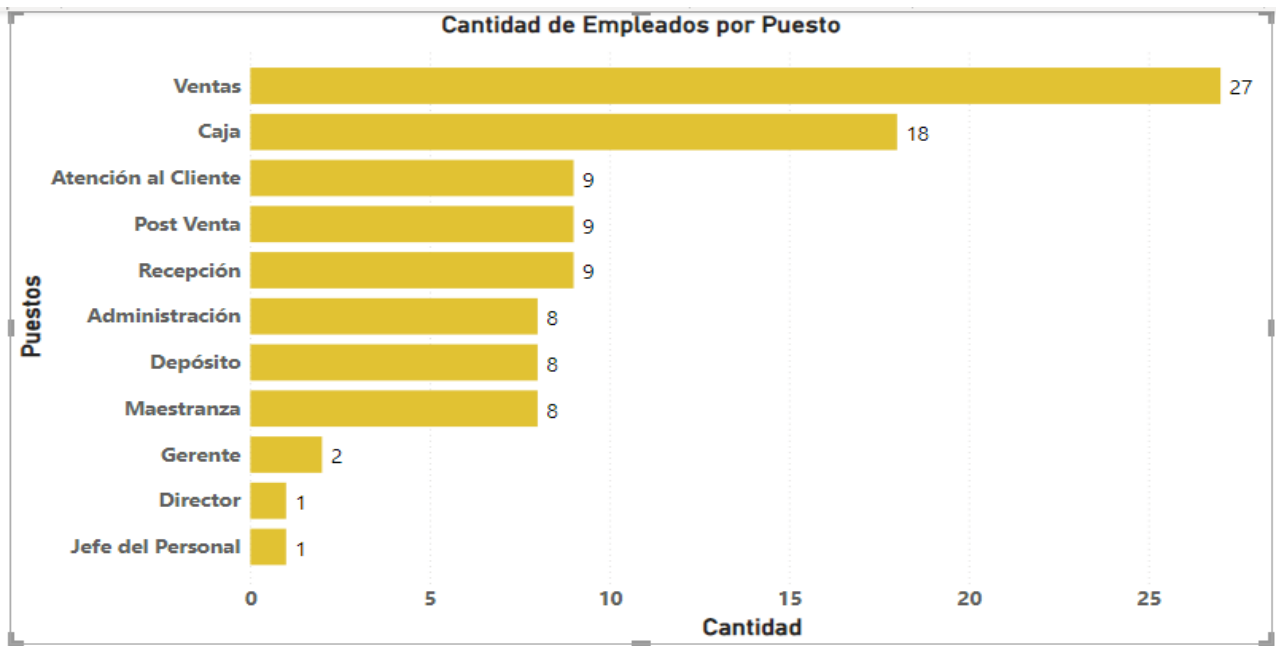
Link: <https://github.com/MauroNicolasRivero/Idea-Rivero-SQL>

Gráfico N° 1



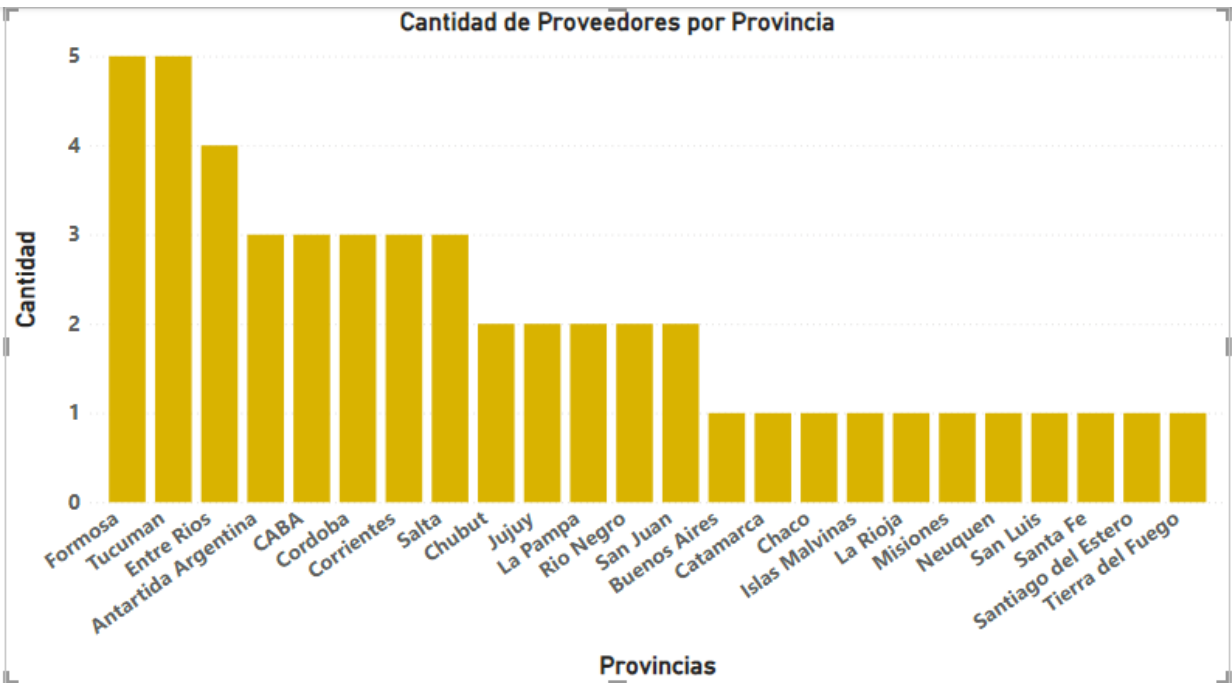
Este gráfico de barras verticales representa la cantidad de Clientes agrupados por Provincias.

Gráfico N° 2



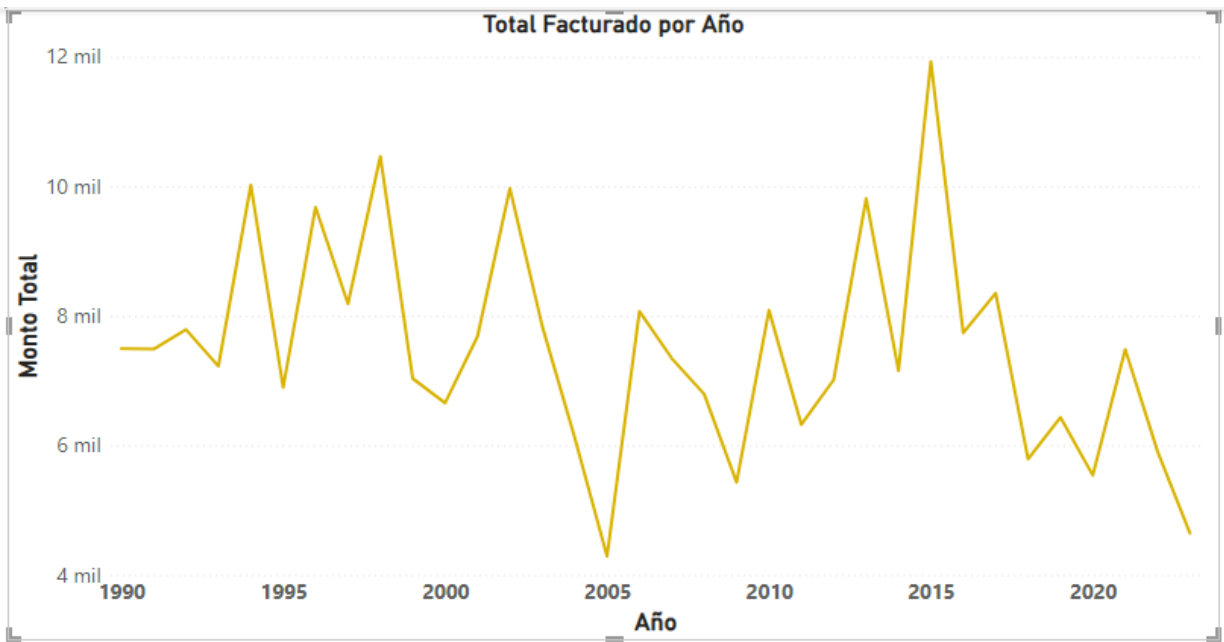
Este gráfico de barras horizontales representa la cantidad de Empleados agrupados por Puesto

Gráfico N° 3



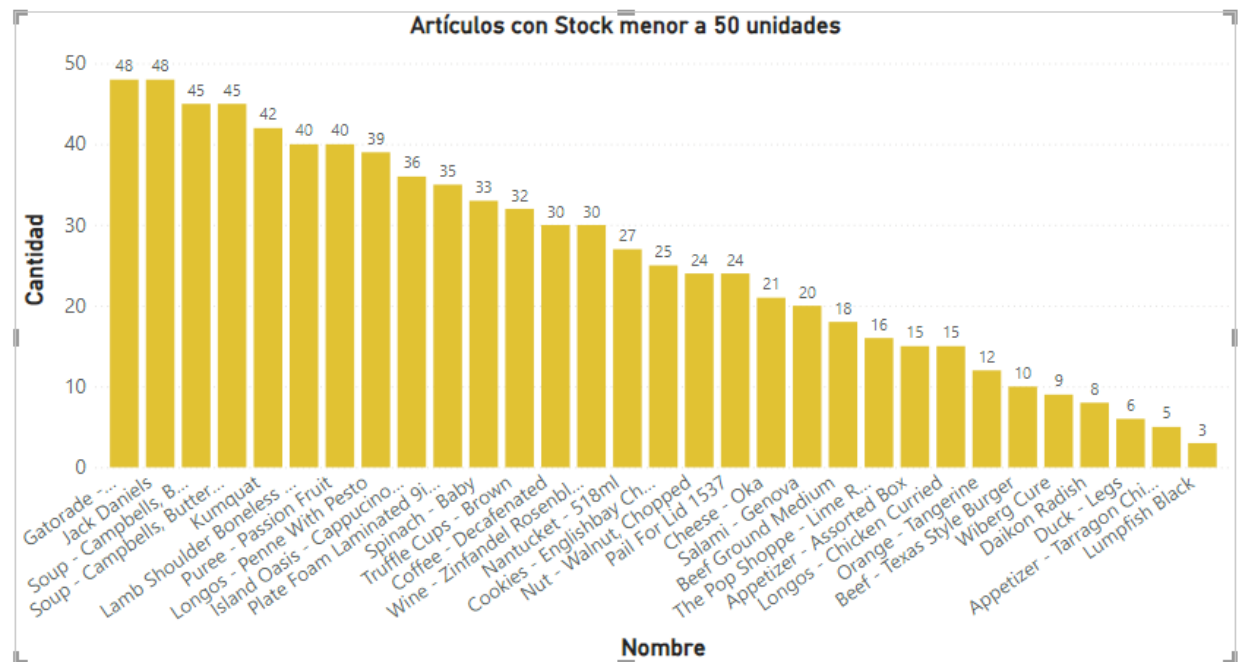
Este gráfico de barras verticales representa la cantidad de Proveedores por Provincia

Gráfico N° 4



Este gráfico de líneas representa la evolución del total facturado a lo largo de los años desde su fundación hasta la actualidad

Gráfico N° 5



Este gráfico de barras verticales representa los productos que actualmente tenemos en stock y su cantidad es inferior a 50 unidades.

Agradecimientos

Quiero agradecer a los profesores Gerardo Tezza en el comienzo del curso y a Miguel Rodas quien hizo un trabajo enorme para recuperar los contenidos de las primeras clases, explicando de forma muy clara y profesional los conceptos, con un manejo excelente de las tecnologías de educación que nos aceleró la curva de aprendizaje. Valoro mucho también su experiencia en la materia, la que supo transmitir en cada contenido, para acercarnos a la realidad del rubro.

Agradecer también al grupo de tutores porque siempre estuvieron apoyándonos, especialmente a Santiago A. González Martín que me acompañó durante toda la cursada aportándome su experiencia y dándome seguridad en los momentos de incertidumbre en los entregables.

Agradecer también a mis compañeros por el respeto y dedicación en cada clase y también porque ellos aportaron conocimiento al curso con sus experiencias.

Por último y no menos importante a todo el equipo de CODERHOUSE por democratizar la educación, dándole oportunidades a muchas personas como en mi caso, de formarse a la distancia y con aranceles accesibles en estos tiempos difíciles en lo económico

” Una vez que sabemos algo, nos resulta difícil
imaginar como era no saberlo”

- Chip Heath

Volver al [Índice](#)