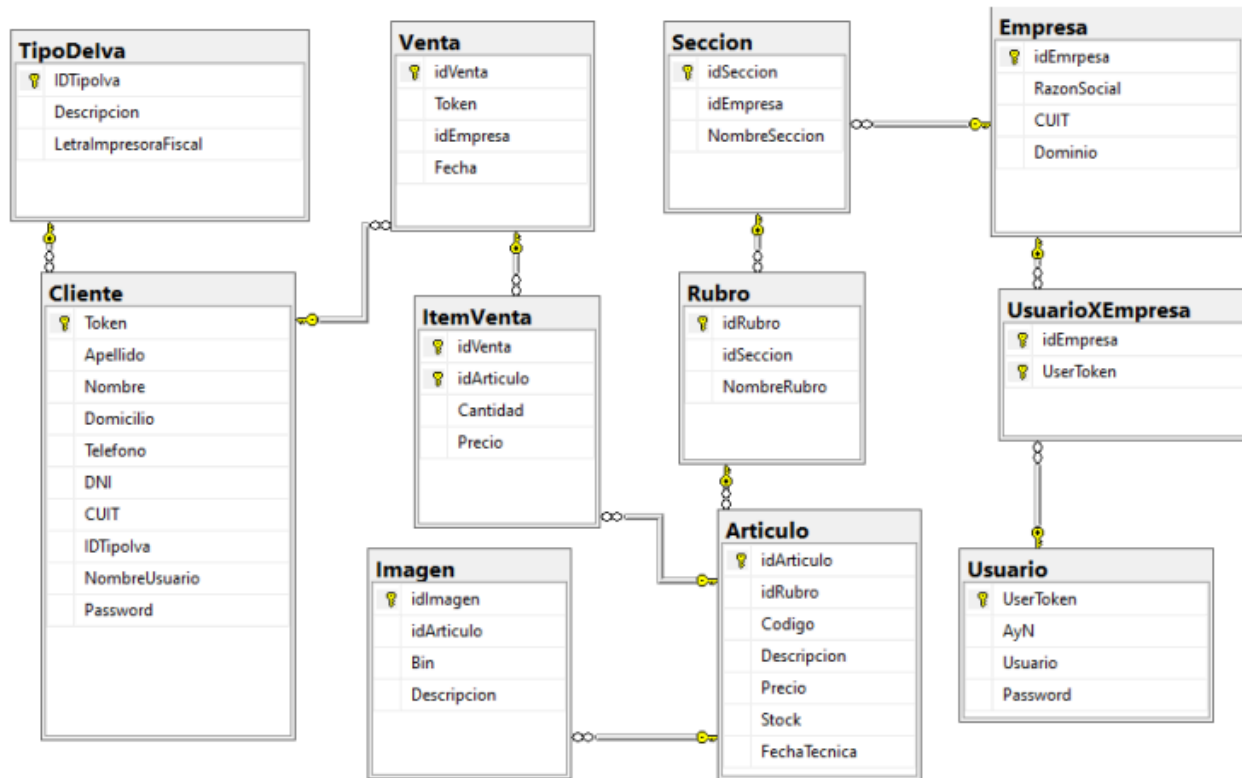


Trabajo Práctico Nro. 9 – Bases de datos (Diagramación y consultas SQL)

Tener en cuenta que las prácticas que siguen, las trabajaremos sobre el siguiente esquema de base de datos:



El esquema representa, un sistema básico de registro de ventas.

Dicho sistema podrá ser utilizado por diferentes empresas, las cuales tendrán uno o más usuarios. Se contempla la posibilidad de que un usuario, pueda pertenecer a más de una empresa. Los usuarios, tendrán diferentes funciones, las que se determinarán a medida que el sistema comience a funcionar.

Cada empresa, podrá tener una cantidad indeterminada de productos que comercializa.

Una de las características de este sistema, será que cada empresa deberá registrar y publicar, los artículos determinados en grupos, que se llaman sección y rubro. Sección, determinará el tipo de productos que vende la empresa. Y dentro de cada uno de los rubros que la empresa comercializa, los productos de cada empresa, deberán estar rubricados para facilitar la búsqueda y agrupamiento de la información.

Una venta realizada por una empresa, quedará identificada por la empresa y el cliente que la realiza.

Estas, combinadas con un idVenta, lo que hará que cada venta quede registrada a una empresa. A su vez, un cliente podrá comprarle a cualquier empresa, pero siempre deberá hacerlo de a una empresa a la vez.

Ejercicio 1: Mostrar toda la información de los clientes.

```
select * from Cliente
```

Ejercicio 2: Listar solamente el “apellido y nombres” (orden descendente) de todos los clientes.

```
select RazonSocial from Cliente order by RazonSocial desc
```

Ejercicio 3: Realizar una selección que le permita mostrar la CANTIDAD total de clientes por tipo de IVA existente.

```
select T.Descripcion, Cantidad = count(C.Token)
from Cliente C, TipoDeIva T
where C.IDTipoIva = T.IDTipoIva
group by T.Descripcion
```

Ejercicio 4: Realizar una selección que le permita listar toda la información pertinente de aquellos clientes que tienen el apellido Martinez.

```
select *
from Cliente C
where C.RazonSocial like 'MARTINEZ%'
```

Ejercicio 5: Muestre un informe que brinde como resultado: la cantidad de clientes que tienen un mismo nombre. Debe indicar Nombre, Cantidad.

```
select RazonSocial, Cantidad = count(Token)
from Cliente
group by RazonSocial
order by Cantidad desc
```

Ejercicio 6: Realizar una selección que le permita listar la cantidad de veces que se vendió cada producto en el año actual. Cantidad de veces, no es cantidad de artículos.

```
select A.Descripcion, Cantidad = count(IV.idItem)
from Articulo A
left join ItemVenta IV on A.idArticulo=IV.idArticulo
inner join Venta V on IV.idVenta=V.idVenta
where YEAR(V.Fecha) = 2018
group by A.Descripcion
order by Cantidad desc
```

Ejercicio 7: Realizar una selección que le permita listar solamente el “apellido y nombre” y DNI de todos los clientes ordenados por DNI (ascendente), que no tengan un CUIT registrado.

```
select C.NombreUsuario, C.DNI
from Cliente C
where CUIT is null or CUIT = ' '
order by C.DNI asc
```

Ejercicio 8: Listar los clientes que están registrados como Responsables Inscriptos, pero no tienen registrado su CUIT como es debido.

```
select *
from Cliente C join TipoDeIva TI on C.IDTipoIva=TI.IDTipoIva
where CUIT is null and TI.Descripcion='Resp. Inscripto'
```

Ejercicio 9: Realizar una selección que le permita listar el promedio de veces que ha sido vendido cada artículo en el historial, y ordenar de manera descendente de mayor a menor. De cada artículo incluir el dato de código y descripción.

```
select A.Codigo, A.Descripcion, Promedio = AVG(IV.Cantidad)
from Articulo A
left join ItemVenta IV on A.idArticulo = IV.idArticulo
inner join Venta V on IV.idVenta = V.idVenta
group by A.Codigo, A.Descripcion
order by Promedio desc
```

Ejercicio 10: Devuelva el listado de la cantidad de artículos registrada por cada sección por cada empresa. Ordenar todo por orden alfabético.

```
select E.RazonSocial, S.NombreSeccion, Cantidad = COUNT(A.idArticulo)
from Empresa E join Seccion S on E.idEmpresa = S.idEmpresa
join Rubro R on S.idSeccion = R.idSeccion
join Articulo A on R.idRubro = A.idRubro

group by E.RazonSocial, S.NombreSeccion
order by E.RazonSocial desc, S.NombreSeccion asc
```

Ejercicio 11: Resuelva el listado que devuelva cual fue el monto vendido por empresa en el año actual.

```
--(No salen resultados porque no se realizaron ventas en la fecha actual)
select E.idEmpresa, Monto_Vendido = SUM(IV.Cantidad*IV.Precio)
from Empresa E join Venta V on E.idEmpresa = V.idEmpresa
left join ItemVenta IV on V.idVenta = IV.idVenta
```

```
where YEAR(V.Fecha) = YEAR(GETDATE())
group by E.idEmpresa
```

Ejercicio 12: Realizar una selección de los productos que se contengan la palabra “unipolar”, junto con sus precios.

```
select A.Descripcion, A.Precio
from Articulo A
where Descripcion like '%unipolar%'
```

Ejercicio 13: Generar la consulta que emita la lista de precios más completa de cada empresa.

```
--(Solamente hay datos cargados en la tabla de Seccion de la empresa con ID = 1)
select E.idEmpresa, S.NombreSeccion, R.NombreRubro, A.idArticulo, A.Precio
from Empresa E join Seccion S on S.idEmpresa = E.idEmpresa
join Rubro R on S.idSeccion = R.idSeccion
join Articulo A on R.idRubro = A.idRubro
group by E.idEmpresa, S.NombreSeccion, R.NombreRubro, A.idArticulo, A.Precio
```

Ejercicio 14: Devolver el listado de los Usuarios que estén registrados en más de una empresa. Devolver, Apellido y Nombre, y la cantidad de empresas en las que se encuentra registrado.

```
select U.AyN, Cantidad = COUNT(UXE.idEmpresa)
from Usuario U join UsuarioXEmpresa UXE on U.UserToken = UXE.UserToken
join Empresa E on UXE.idEmpresa = E.idEmpresa
group by U.AyN
```

Ejercicio 15: Listar los artículos que nunca han sido vendidos. Luego listar la cantidad de artículos que jamás fueron vendidos.

```
select * from Articulo where idArticulo not in (select idArticulo from ItemVenta)

select A.Descripcion, Cantidad = COUNT(A.idArticulo)
from Articulo A left join ItemVenta IV on A.idArticulo = IV.idArticulo
group by A.Descripcion having COUNT(A.idArticulo) = 0
```

Ejercicio 16: Listar el monto total vendido por mes en el año 2019 por cada una de las empresas.

```
select E.idEmpresa,V.Fecha,Monto_Vendido = SUM(IV.Precio*IV.Cantidad)
from ItemVenta IV join Venta V on IV.idVenta = V.idVenta
join Empresa E on V.idEmpresa = E.idEmpresa
where YEAR(V.Fecha) = 2019
group by E.idEmpresa, V.Fecha
```

```
order by MONTH(V.Fecha) asc
```

Ejercicio 17: Listar las ventas realizadas por día de la semana en todo el historial de la base de datos, pero, ordenada por número de día.

```
select Dia_de_la_semana = DATEPART(dw, V.Fecha),Cant_Ventas_Realizadas = COUNT(V.idVenta)
from Venta V
group by DATEPART(dw, V.Fecha)
order by DATEPART(dw, V.Fecha) asc
```