

Alfredo Buzali

Proscaila base de datos y su interacción con SQL

Soporte Proscail

Contenido

1. Introducción

Capítulo 1 Entendiendo la definición de bases de datos

- a. Bases de datos relacionales y SQL.
- b. Que es una base de datos.
- c. Bases de datos relacionales.
- d. Llaves primarias y secundarias
- e. Tipos de datos.
- f. otros tipos de bases de datos.
- g. Lenguaje y lógica.
- h. Definiendo SQL.
- i. Diferentes bases de datos SQL.
- j. Viendo hacia adelante.
- k. Conceptos básicos (resumiendo).

Capítulo 2 Base de datos Proscail

- a. Estructura de la base de datos Proscail.
- b. Principales tablas.
- c. Detalle de campos y contenido.
- d. Modelos de conexión entre tablas.

Capítulo 3 Extracción de datos

- a. Herramientas y mecanismos para la extracción de datos.
- b. SQL Comandos e instrucciones básicas.
- c. Comandos adicionales.
- d. Resumen creando sentencias.
- e. Modos de conexión.
- f. Consultas básicas sin condiciones.
- g. Condicionar una consulta.
- h. Consulta con operadores adicionales.
- i. Combinando operadores.
- j. Condiciones con campos no solicitados.
- k. Agregando funciones como condición.
- l. Usando campos booleanos o con valor numérico.
- m. Haciendo operaciones aritméticas.
- n. Asignar nombre a los campos (alias).
- o. Nombrando tablas (alias).
- p. Cambiando el orden del resultado en consultas.
- q. Creando agrupaciones.
- r. Filtrando después de agrupar.

Capítulo 4 Creando consultas con más de 1 tabla

- a. Banderas de identificación de un documento.
- b. Conectando 2 tablas.
- c. Conectando 3 tablas.
- d. Conectando 4 tablas.
- e. Agregar un contador a una agrupación.
- f. Modelando consultas como reportes de ventas.
- g. Ligando tablas sin llaves primarias.
- h. Varias conexiones a una misma tabla.
- i. Creando reportes de ventas.

Capítulo 5 Herramientas adicionales

- a. Cubo Proscái.
- b. Generar reportes a archivo y querys desde Proscái.

INTRODUCCIÓN

Este libro está pensado para cubrir tres objetivos principales

- 1.- Comprender la estructura de la base de datos de **Proscái** y su diseño para poder explotar la información.
- 2.- Aprender los conceptos básicos de SQL para recuperación de información de una base de datos de **Proscái**.
- 3.- Generar búsquedas de información complejas, donde prácticamente se pueda extraer cualquier dato de la base de datos de **Proscái**.

El presente trabajo esta creado en un formato informal en el cual no estoy cubriendo al máximo los protocolos de escritura ni redacción solamente busco hacerlo más coloquial. Quiero dedicar más mi esfuerzo a que aprendan a utilizar la información de **Proscái** y no al estilo del texto y por el otro lado, para aquellos que me conocen no quiero de dejar de ser yo al escribirlo, ahí la razón de no hacerlo en un formato muy rígido.

No pretendo que este sea un libro donde se cubran todos los aspectos de una base de datos y está enfocado específicamente a la extracción y no a la inserción o actualización de información. Tampoco abordare temas referentes al diseño de la base de datos ya que esto es función de **Proscái**, sin embargo, el entender la estructura de la información permitirá al usuario cuando sea necesario actualizar información (aquella permitida y vía los mecanismos de importación de **Proscái**) en una forma segura.

Quien domine este libro, tendrá las herramientas suficientes para extraer cualquier información de la base de datos de **Proscái** y utilizarla en diferentes formas, desde generar un Query simple en el browser de MySQL, generar cubos o conectar cualquier sistema vía el ODBC de MySQL para generar reportes o consultas en línea para generar y analizar gráficas.

En las siguientes páginas, deseo cubrir los diferentes aspectos del lenguaje SQL de manera natural, fácil y progresiva, y de la misma forma se irá cubriendo la lógica de la estructura de la base de datos de **Proscái**.

Les recomiendo realizar los ejercicios en una Base de Datos de pruebas, ya que al estar ejecutado las sentencias y conociendo los campos, si no se sabe filtrar adecuadamente, por error se puede solicitar gran cantidad de información y ocasionar lentitud en el sistema y el trabajo de los usuarios. Así mismo, consideren usar una base pequeña para realizar las pruebas.

CAPÍTULO 1

Entendiendo la definición de bases de datos

Bases de datos relacionales y SQL

En este capítulo voy a explicar los aspectos básicos de una base de datos, como se guarda la información en registros y campos o renglones y columnas, como se relaciona y los diferentes tipos de datos que se manejan y como todo esto se relaciona al lenguaje de SQL

Que es una base de datos

Es un conjunto de información almacenada de una forma predeterminada, que permitirá buscarla y utilizarla fácilmente. Para ello se agrupa o estructura de forma que la información contenida sea fácil de identificar.

Una base de datos está compuesta por tablas que agruparan información y dentro de las tablas habrá renglones o campos que serán los registros específicos de datos

Bases de datos relacionales

Hablemos ahora de los principios básicos de una base de datos relacional y cómo funciona.

Una base de datos relacional es un conjunto de datos almacenados en diferentes tablas, el término relacional se refiere a la forma en la que las diferentes tablas están ligadas o relacionadas.

Como ejemplo pensemos en una base de datos con dos tablas clientes y documentos, la tabla de clientes es donde se tiene un registro para cada cliente, y en la tabla de documentos están almacenada la información de los documentos que nos deben los diferentes clientes en donde hay un registro para cada documento. Cada tabla contiene una lista de campos donde se define el tipo de información que esta almacena en nuestro ejemplo en la tabla de clientes se almacena información como su nombre, dirección, colonia, etc. y en la de documentos se guarda el número de documento, la fecha, importe, etc.

Llaves primarias y llaves secundarias

Nótese en nuestro ejemplo anterior que el primer dato en cada una de las tablas será un campo identificador del registro de cada una de ellas, en la tabla de clientes lo llamaremos IDcliente y en la de documentos IDdocumento , este dato se conoce como la llave primaria , una llave primaria tiene dos funciones, la primera para identificar un registro específico en una tabla por ejemplo si se desea extraer la información de “Comercializadora ABC” se podrá obtener mediante el campo de IDcliente, las llaves primarias son únicas en cada tabla garantizándonos en nuestro ejemplo que solo hay un cliente con el registro IDcliente=2 inclusive si se tuvieran dos clientes llamados “comercializadora abc” cada uno de ellos tendría un valor diferente en el campo IDcliente, normalmente para generar la llave primaria las bases de datos tienen un campo que incrementa su valor cuando se inserta inicialmente un registro y con esto se garantiza que sea único, a este tipo de campos se les llama secuenciador que es único, irrepetible e inalterable.

Es importante considerar que cada tabla tiene únicamente una llave primaria.

La segunda función de ser de las llaves primarias será para permitir relacionar dos tablas entre sí, la llave primaria de una tabla, se heredará y funcionará como llave secundaria en otra tabla, creando así una relación entre la información de diferentes tablas.

Tipos de datos

Las llaves primarias y secundarias le dan estructura a las tablas de una base de datos, nos proporcionan un canal de accesibilidad a los registros y nos permiten relacionar diferentes tablas, existen otros tipos de datos que se almacenan en los registros y básicamente se pueden resumir en tres: números, caracteres y fechas. Cada uno de ellos pueden tener variantes las cuales no vamos a cubrir en este libro, solo mencionaremos las características más esenciales que son:

Los datos numéricos guardan información cuantificable y con la cual vamos a hacer operaciones aritméticas, dentro de los datos numéricos existen diferentes formatos con decimales, enteros, de punto flotante entre otros, en este tipo de datos se encuentran por ejemplo los campos que guardan los saldos de nuestros clientes, el inventario de los productos, el saldo de los bancos, etc. ya que su visualización siempre es numérica.

Los campos de caracteres son básicamente cadenas de datos alfanuméricos en ellos se puede almacenar cadenas de caracteres incluyendo letras mayúsculas o minúsculas, números y símbolos, prácticamente cualquier carácter que se genere desde nuestro teclado, en ellos es donde se registra información como: nombres, direcciones, direcciones de correo, números telefónicos, RFC entre otros.

En los campos de tipo fecha existen dos variantes, el primero son fechas normales como día, mes, año y en el segundo fechas incluyendo hora, minuto y segundo, en el primer tipo guardamos información como la fecha de alta de un producto o la fecha de vencimiento de un documento, en el segundo se registra información como fecha y la hora en la que se generó un documento específico.

Otros tipos de bases de datos

En la historia de los sistemas han existido diferentes formas de almacenar la información, de lo cual quiero comentar un poco como apoyo en la explicación de los beneficios de una base de datos relacional, como dato curioso todos ellos han sido utilizados en las diferentes versiones de **Proscail** en la evolución desde sus inicios.

Al inicio de los sistemas la información se guardaba en archivos de texto, lo que equivalía a generar un archivo individual de lo que sería cada tabla, cada uno de ellos como un archivo de tipo txt, el manejo de información era complejo y muchas veces se tenía que tener los datos completos en memoria para ser procesados.

Con el surgimiento de las bases de datos jerárquicas se empezaron a usar apuntadores en los registros los cuáles nos indican por ejemplo en un documento quien era su cliente, algo similar a las llaves primarias y de un documento los productos involucrados también se relacionaban con un apuntador a su documento, en este caso el problema principal es que los procesos se tenían que generar un registro a la vez.

En ese momento de la historia, el problema no era como se guardaba la información sino la forma de accederla, el gran cambio fue la llegada de las bases de datos relacionales y junto con la aparición del lenguaje SQL ya que ofrece una forma totalmente diferente de extraer los datos.

A diferencia de las bases de datos antecesoras a SQL, este último permite consultar grandes cantidades de información en un solo evento, en una sola sentencia de SQL se pueden extraer miles de datos de diferentes tablas usando las relaciones existentes entre ellas, con esto se facilita el acceso a la información tanto para los usuarios como para los sistemas ya que no se tiene que procesar cada registro en forma individual, esta es la causa por la cual **Proscail** con SQL es muchas veces más rápido que con las versiones anteriores.

Lenguaje y lógica

Dentro de las principales ventajas que ofrece SQL como manejador de base de datos, es la gran capacidad de mantener cientos de usuarios en línea, almacenar grandes capacidades de información, y la facilidad para que los usuarios puedan extraer información de la misma para fines diversos.

SQL quiere decir por sus siglas en inglés Structured Query Language que significa Lenguaje Estructurado de Consultas.

A través de ciertas funciones y comandos que conforman una sentencia, usamos un lenguaje por medio del cual se puede manipular una base de datos a través de sentencias lógicas, de hecho todo en SQL, desde la creación de la base de datos la inyección de información, el borrado de la misma y la extracción de la misma es a través de sentencias donde le indicamos que es lo que se quiere hacer.

En general las sentencias de este lenguaje son de alguna forma naturales y comprensibles, y llega un momento donde parece ser que estamos hablando o pidiendo mediante palabras concretas que nos diga información, es algo que pareciera un lenguaje casi natural, por ejemplo en la siguiente sentencia:

```
Select nombre, dirección  
From clientes  
Order by nombre
```

Estaríamos pidiendo los nombres y direcciones de nuestros clientes ordenados por nombre, en realidad estamos solicitando lo que queremos, de donde lo queremos y el orden en el que lo queremos, básicamente es una instrucción donde en una forma estructurada indicamos la información y la forma en que la requerimos.

Bajo este ejemplo SQL es un lenguaje con el cual nos vamos a comunicar con una base de datos.

Definiendo SQL

Entonces que es SQL? En pocas palabras es un lenguaje que nos permite interactuar con una base de datos relacional, en nuestro caso es el lenguaje que nos permitirá interactuar con la base de datos de **Proscái**.

SQL es un lenguaje casi universal y ha sido adoptado por un gran número de vendedores de bases de datos como Microsoft y Oracle y prácticamente podríamos decir que se ha convertido en un estándar para el manejo de bases de datos, lo cual permite que haya una gran cantidad de gente que lo conozca, así como productos de terceros para manejo de información y generación de reportes, etc. Por otro lado cada fabricante de bases de datos tiene sus pequeñas o grandes diferencias en el lenguaje y los comandos pueden variar cuando se manejan diferentes bases de datos.

Diferentes bases de datos SQL

Existen diferentes bases de datos que utilizan SQL las más populares son:

MySQL
Microsoft SQL Server
Microsoft Acces
Oracle
PostgresSQL
SQL Anywhere
DB2 de IBM
Informix de IBM

MySQL es una base de datos open source lo que quiere decir que su desarrollo y versiones no pertenece a una empresa específica, sin embargo MySQL fue adquirida en 2008 por SUN microsystems y en 2009 SUN fue adquirida por Oracle.

Por ser una base de datos abierta MySQL existe para diferentes plataformas como Windows, Macintosh y Linux. MySQL se ofrece en su edición Comunitaria como una versión gratuita.

En el caso de **Proscail** la base de datos que se utiliza es MySQL, por lo cual y en adelante, hablaremos y usaremos dicho lenguaje para enfocarnos a través de sus comandos, para realizar ejemplos con una base de datos de **Proscail** y obtener información.

Viendo hacia adelante

En este capítulo hemos podido forjar las bases necesarias para entender una base de datos relacional y hemos visto diferentes aspectos de las bases de datos relacionales como llaves primarias y secundarias y tipos de datos con lo cual estamos listos para conocer nuestra base de datos **Proscail**.

CONCEPTOS BÁSICOS (resumiendo)

Bases de Datos

Es un conjunto de información perteneciente a un mismo contexto, ordenada de modo sistemático, que nace como la necesidad humana de almacenar la información, lo cual permitirá posteriormente su análisis, consulta o transmisión de datos.

La base de datos **Proscail** archiva los datos en tablas separadas que se encuentran conectadas por relaciones definidas, que hacen posible combinar datos de diferentes tablas, en la cual solo es necesario tener identificada claramente la necesidad y el conocimiento de las tablas en las cuales se encuentran los datos que brindarán la información solicitada.

Base Relacional

Una base de datos relacional es una recopilación de elementos de datos con relaciones predefinidas entre ellos. Estos elementos se organizan como un conjunto de tablas con columnas y filas. Este tipo de bases es la más utilizada.

Llave Primaria

Llave Primaria o clave principal, hace referencia a un campo o combinación de campos que identifica de forma única a cada fila de una tabla. No puede haber dos filas en una tabla que tengan la misma llave primaria.

Llave Secundaria

Cuando se establece una relación entre tablas, la relación se hace mediante una copia de la llave primaria, misma que se vuelve llave secundaria en la segunda tabla.

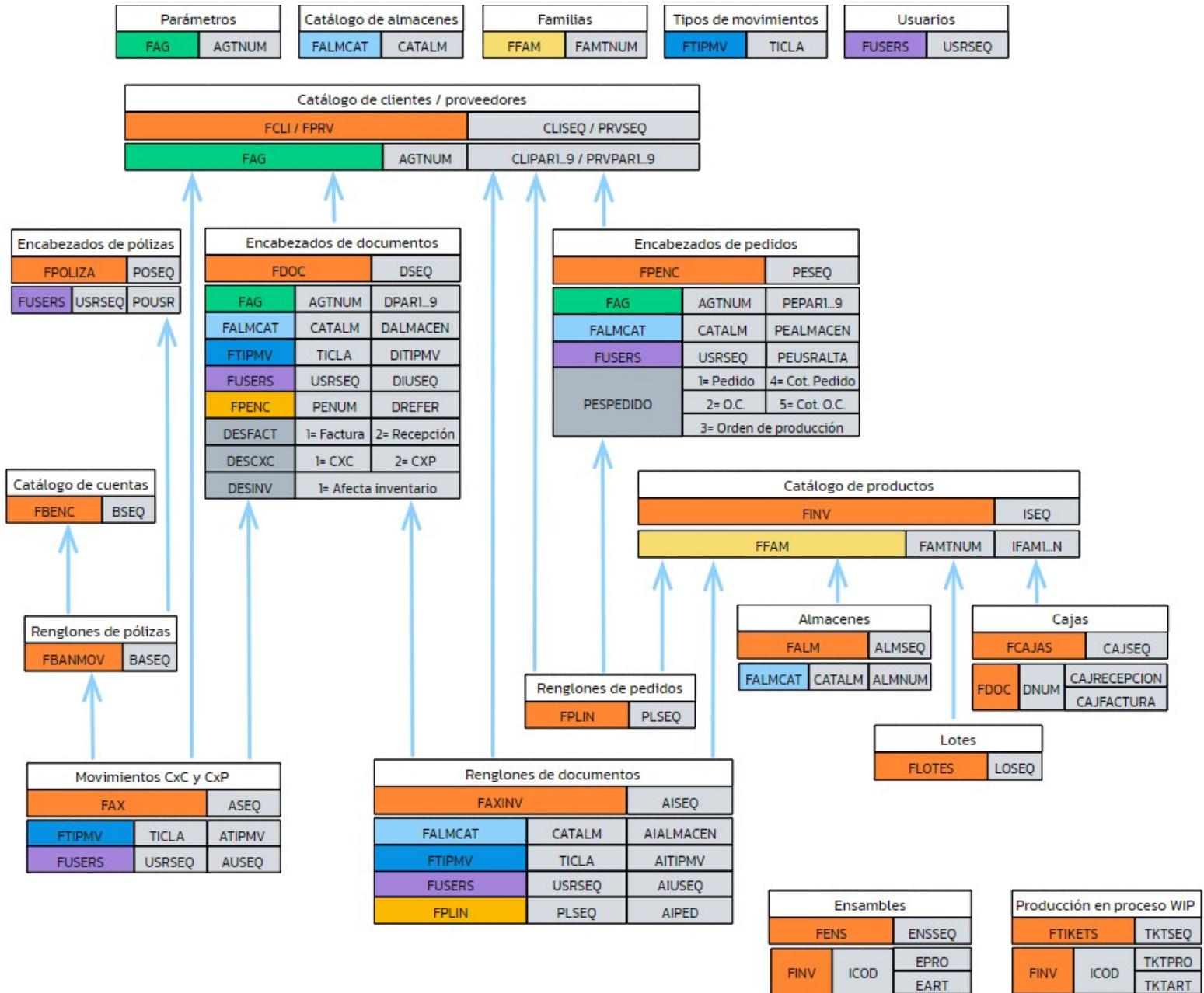
CAPÍTULO 2

Base de datos Proscái.

Estructura de la base de datos Proscái.

En resumen del capítulo anterior definimos qué es una base de datos, qué esta contiene tablas que a su vez contiene registros y en éstos se encuentran los datos distribuidos en una serie de campos, así, cada registro de la tabla guarda la información particular de una unidad o miembro de un mismo grupo.

El siguiente esquema gráfico ejemplifica la estructura y conexión de las principales tablas, después mencionaremos los campos dentro de cada tabla, con los que podrás crear tus propias consultas directas.



Vamos a explicar la relación o vinculación entre tablas y el significado de los diferentes elementos de la gráfica anterior.

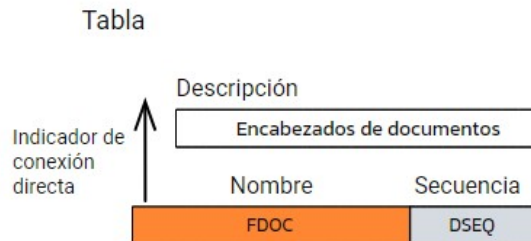
En los siguientes diagramas, describimos la interpretación de las tablas de Proscái, sus relaciones y las banderas que contienen las diferentes tablas, para cada uno de estas relaciones y banderas se encontrara información detallada más adelante

Cada tabla está representada en un campo formado por varios tipos de datos.

En él encontraremos el nombre de la tabla, su secuencia, una descripción breve, sus conexiones directas e indirectas y finalmente, las banderas para poder filtrar los diferentes tipos de documentos que cada tabla puede contener.

Usaremos en este caso la tabla FDOC como ejemplo, ya que es una tabla que utiliza todos los tipos de conexiones.

Encabezados de documentos		
FDOC		DSEQ
FAG	AGTNUM	DPAR1..9
FALMCAT	CATALM	DALMACEN
FTIPMV	TICLA	DITIPMV
FUSERS	USRSEQ	DIUSEQ
FPENC	PENUM	DREFER
DESFACT	1= Factura	2= Recepción
DESCXC	1= CXC	2= CXP
DESINV	1= Afecta inventario	



Conexiones indirectas

Catálogos

Tabla indirecta	Campo de conexión	Campo
FAG	AGTNUM	DPAR1..9
FALMCAT	CATALM	DALMACEN
FTIPMV	TICLA	DITIPMV
FUSERS	USRSEQ	DIUSEQ

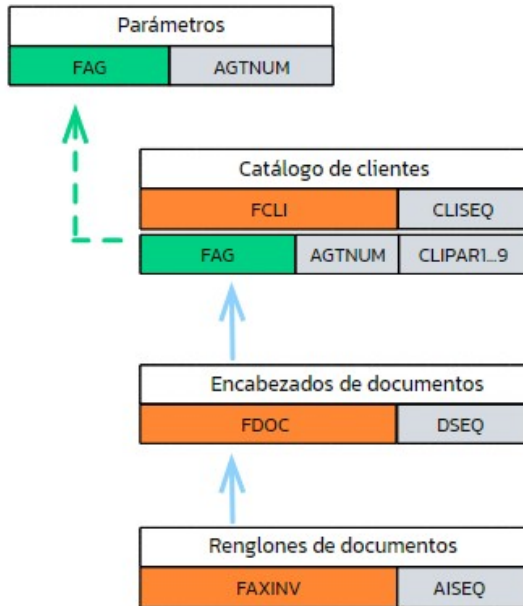
Tablas

Tabla indirecta	Campo de conexión	Campo
FPENC	PENUM	DREFER

Banderas de tipos de documentos

DESFACT	1= Factura	2= Recepción
DESCXC	1= CXC	2= CXP
DESINV	1= Afecta inventario	

La estructura básica de Proscái es de tipo jerárquico. Para entender este concepto de jerarquía, vamos a explicarlo extrayendo una parte de la estructura gráfica inicial.



En el diagrama se puede ver la jerarquía donde, entre más alta es una tabla representa que es de mayor nivel y entre más baja que es de mayor detalle.

Las flechas representan que hay una relación entre las tablas en donde el inicio de la flecha representa una tabla que puede contener varios registros relacionados a la tabla hacia donde la flecha apunta.

- **Padre:** Es aquel que tiene descendientes, conocidos como hijos. Todos los padres están localizados al mismo nivel.
- **Hijo:** Es aquel que depende del segmento anterior. Todos los hijos del mismo padre tienen que estar localizados en el mismo nivel.

Es decir, la jerarquía se da porque un registro de cliente, será el padre de cada documento o factura que a este se le emita y a su vez, esos documentos solo pueden estar ligados y dependen de su padre es decir su cliente, de la misma forma, cada documento será padre de varios renglones que indiquen los productos vendidos al cliente.

Veamos ahora otros elementos que conforman la base de datos Proscái.



Cada cuadro o rectángulo representa una tabla, la cual muestra su nombre y su campo llave que sirve de conexión, este grupo de tablas sirven de soporte para el funcionamiento del sistema en general o para alguna otra tabla.

Ninguna de ellas tiene alguna línea que indique conexión directa hacia otra, pero, muestran un color que nos servirá para identificar que esa tabla podrá ligarse a otra por medio de una referencia

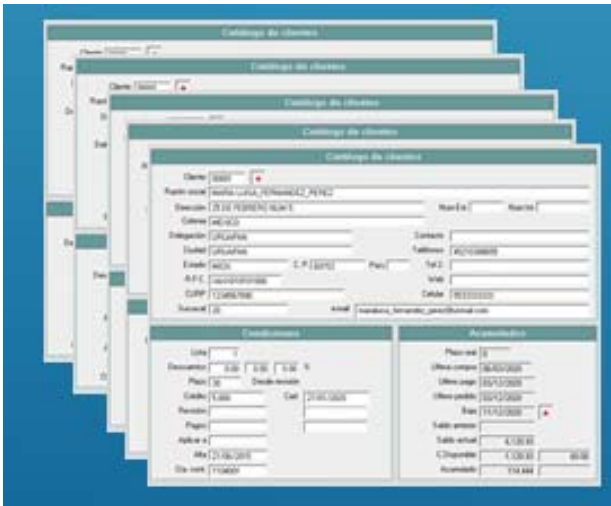
Estructura de datos típicos

En nuestro diagrama inicial, hemos visto que cada tabla está representada por un bloque como los que hemos descrito anteriormente.



Una tabla es una estructura de datos en donde se guarda información de elementos similares, por ejemplo la tabla FCLI contiene la información de los clientes de la empresa.

Una tabla está compuesta de registros en donde un registro es el lugar en donde se guarda la información de cada cliente:



Y finalmente cada registro está compuesto por un grupo de campos que es donde se guarda la información que se requiere de cada uno de los clientes como su nombre, dirección, teléfono, saldo, etc.

Catálogo de clientes	
Ciudad	00001
Razón social	MARIA LUISA_FERNANDEZ_PEREZ
Dirección	25 DE FEBRERO NUM 5
Colonia	MEXICO
Delegación	URUAPAN
Ciudad	URUAPAN
Estado	MICH.
R.F.C.	XAXX-010101000
CURP	1234567890
Sucursal	20
Contacto	
Teléfonos	45210388655
Tel 2	
Web	
Celular	5533333333
e-mail	marialuisa_fernandez_perez@unmail.com
Condiciones	
Lista	1
Descuentos	0.00 0.00 0.00 %
Plazo	30 Desde revisión
Crédito	5,000 Cad 27/01/2020
Revisión	
Pagos	
Aplicar a	
Alta	21/06/2015
Cta. cont.	1104001
Acumulados	
Plazo real	0
Ultima compra	06/03/2020
Ultimo pago	03/12/2020
Ultimo pedido	03/12/2020
Baja	11/12/2020
Saldo anterior	
Saldo actual	6,120.93
C.Disponible	-1,120.93 60.00
Acumulado	114,444

Existen diferentes tipos de datos, donde los principales son de tipo: Carácter, Numérico y Fecha.

Más adelante se detallarán los campos de cada una de las principales tablas de Proscái.

Finalmente me gustaría indicar que una base de datos es el lugar donde viven toda esta información, en conjunto la base de datos contiene todas las tablas que la conforman.

Principales tablas

Dentro de la declaración de tablas que conforman la base de datos de Proscái, existen algunas que de manera principal constituyen el núcleo primordial en contenido y almacenamiento de la información, misma que se genera con cada una de las operaciones que se realizan en el sistema.

Nombre de tabla en Proscái	Descripción de la tabla
FCLI	Catálogo de Clientes
FPRV	Catálogo de Proveedores
FINV	Catálogo de Inventarios por producto
FDOC	Documentos (recepciones / facturas-Tickets)
FAXINV	Líneas de documento o Auxiliar de inventarios
FAX	Auxiliar de Clientes y Proveedores (importe)
FPENC	Encabezados Pedidos y Órdenes de compra
FPLIN	Líneas de pedidos u órdenes de compra
FBENC	Catálogo de cuentas
FPOLIZA	Encabezados Póliza
FBANMOV	Movimientos de Bancos
FAG	Parámetros
FALMCAT	Catálogo de almacenes
FFAM	Familias
FUNIDAD	Unidad de medida
FCAJAS	Rollos y números de serie
FALM	Detalle de almacenes
FENS	Ensamblados de producción

Detalle de campos y contenido

Ahora vamos a conocer el detalle de cada una de las tablas principales, el nombre de cada campo, la descripción del dato que almacena y el tipo de campo o liga con otras tablas, el campo llave de cada tabla se indica con (*)

FCLI (Clientes)

La tabla **FCLI** contiene los registros de los **clientes** que se utilizan para realizar operaciones de venta, y el tipo de información que ahí se registra, es de tipo general es decir, aquí se encuentran los datos básicos, código, nombre o razón social, dirección, colonia, delegación, ciudad, estado, código postal, RFC, etc.

Campo	Descripción	Tipo	Tipo liga
CLISEQ*	Campo llave único	Liga hacia otras tablas	Directa
CLICOD	Código del cliente	Alfanumérico	
CLINOM	Nombre de cliente	Alfanumérico	
CLIDIR	Dirección	Alfanumérico	
CLICD	Ciudad	Alfanumérico	
CLIEDO	Estado	Alfanumérico	
CLITEL	Teléfono	Alfanumérico	

CLICONT	Contacto	Alfanumérico	
CLISTA	Lista de precios	Numérico	
CLICREDIT	Importe límite de crédito	Numérico	
CLIDESC10	Porcentaje descuento 1	Numérico	
CLIDESC20	Porcentaje descuento 2	Numérico	
CLIPLAZO0	Plazo en días para vencimiento	Numérico	
CLIDIREV	Días de revisión	Alfanumérico	
CLIDIPAGO	Días de pago	Alfanumérico	
CLISANT	Salto anterior	Numérico	
CLISACT	Saldo actual	Numérico	
CLIPLAZOR	Plazo real de pago	Numérico	
CLIPAR1	Código de clasificador 1	Referencia a FAG	Indirecta
CLIPAR2	Código de clasificador 2	Referencia a FAG	Indirecta
CLIPAR3	Código de clasificador 3	Referencia a FAG	Indirecta
CLIPAR4	Código de clasificador 4	Referencia a FAG	Indirecta
CLIPAR5	Código de clasificador 5	Referencia a FAG	Indirecta
CLIPAR6	Código de clasificador 6	Referencia a FAG	Indirecta
CLIPAR7	Código de clasificador 7	Referencia a FAG	Indirecta
CLIPAR8	Código de clasificador 8	Referencia a FAG	Indirecta
CLIPAR9	Código de clasificador 9	Referencia a FAG	Indirecta
CLIULTCOMP	Fecha última compra	Fecha	
CLIULTPAG	Fecha último pago	Fecha	
CLICOLONIA	Colonia	Alfanumérico	
CLIFAX	Numero fax	Alfanumérico	
CLICP	Código postal	Numérico	
CLIDESC30	Porcentaje descuento 3	Numérico	
CLIRFC	R.F.C.	Alfanumérico	
CLIDIR2	Dirección enviar a	Alfanumérico	
CLICD2	Ciudad enviar a	Alfanumérico	
CLIEDO2	Estado enviar a	Alfanumérico	
CLIGRUPO0	% Descuento grupo 1	Numérico	
CLIGRUPO1	% Descuento grupo 2	Numérico	
CLIGRUPO2	% Descuento grupo 3	Numérico	
CLIGRUPO3	% Descuento grupo 4	Numérico	
CLIGRUPO4	% Descuento grupo 5	Numérico	
CLIGRUPO5	% Descuento grupo 6	Numérico	
CLIALTA	Fecha alta	Fecha	
CLIBAJA	Fecha bloqueo	Fecha	
CLIOBS	Observaciones	Alfanumérico	
CLINUMPROVEEDOR	Núm. de prov. para cadenas	Alfanumérico	
CLISUCURSAL	Sucursal	Alfanumérico	
CLICTA	Numero cuenta contable	Numérico	
CLITIPOIVA	Tipo IVA	Numérico	
CLIDELEGACION	Delegación	Alfanumérico	
CLICIA	Compañía a la que pertenece	Numérico	
CLIPUNTOS	Puntos lealtad acumulados	Numérico	
CLIEAN	Código EAN o tarjeta lealtad	Alfanumérico	
CLINACIMIENTO	Fecha nacimiento	Fecha	
CLIMONEDA	Moneda de cobro	Numérico	

CLISOLOPOS	Dónde se le puede vender 0=todo1=solo FACT 2=solo POS	Numérico	
CLIMULTICIA	Numero de multicompañía	Numérico	
CLICELULAR	Numero teléfono celular	Numérico	
CLILEALTAD	Estatus lealtad	Numérico	
CLIEMPLEADO	Marca identifica empleados	Numérico	
CLINIVEL	Nivel lealtad	Numérico	

FPRV (Proveedores)

La tabla **FPRV**, contiene los registros de **TODOS** los **proveedores y acreedores** que la empresa maneja, a través del campo PRVTIPO se puede diferenciar a cada uno, permitiendo que en una sola tabla convivan ambos perfectamente identificados.

Campo	Descripción	Tipo	Tipo liga
PRVSEQ*	Campo llave único	Liga a otras tablas	Directa
PRVCOD	Código Proveedor	Alfanumérico	
PRVNOM	Nombre	Alfanumérico	
PRVDIR	Dirección	Alfanumérico	
PRVCD	Ciudad	Alfanumérico	
PRVEDO	Estado	Alfanumérico	
PRVTEL	Teléfono	Alfanumérico	
PRVCONT	Contacto	Alfanumérico	
PRVLISTA	Lista de precios	Numérico	
PRVCREDIT	Límite de crédito	Numérico	
PRVDESC1	Porcentaje descuento 1	Numérico	
PRVDESC2	Porcentaje descuento 2	Numérico	
PRVPLAZO	Días plazo para pago	Numérico	
PRVDIREV	Día de revisión	Alfanumérico	
PRVDIPAGO	Día de pago	Alfanumérico	
PRVSANT	Saldo anterior acumulado	Numérico	
PRVSACT	Saldo actual	Numérico	
PRVPLAZOR	Días de plazo real de pago	Numérico	
PRVPAR1	Código de clasificador 1	Referencia liga a FAG	Indirecta
PRVPAR2	Código de clasificador 2	Referencia liga a FAG	Indirecta
PRVPAR3	Código de clasificador 3	Referencia liga a FAG	Indirecta
PRVPAR4	Código de clasificador 4	Referencia liga a FAG	Indirecta
PRVPAR5	Código de clasificador 5	Referencia liga a FAG	Indirecta
PRVPAR6	Código de clasificador 6	Referencia liga a FAG	Indirecta
PRVPAR7	Código de clasificador 7	Referencia liga a FAG	Indirecta
PRVPAR8	Código de clasificador 8	Referencia liga a FAG	Indirecta
PRVPAR9	Código de clasificador 9	Referencia liga a FAG	Indirecta
PRVULTCOM	Fecha última compra	Fecha	
PRVULTPAG	Fecha último pago	Fecha	
PRVCOLONIA	Colonia del proveedor	Alfanumérico	
PRVFAX	Teléfono de fax	Alfanumérico	
PRVCP	Código postal	Numérico	

PRVRFC	R.F.C.	Alfanumérico	
PRVCTA	Cuenta Mayor de contabilidad	Numérico	
PRVMONEDA	Moneda de pago	Numérico	
PRVTIPO	0=Proveedor, 1=Acreedor, 2=Deudor	Numérico	
PRVTIPOIVA	Tipo de iva que causa	Numérico	
PRVTIEMPO	Días de entrega	Numérico	
PRVALTA	Fecha alta	Fecha	
PRVEXPRESS	Marca para proveedor express	Numérico	

FINV (Inventarios)

La tabla **FINV** contiene los registros de TODOS los **productos, servicios o conceptos** que se utilizan para cualquier operación de compra o venta dentro del sistema. A través del campo ITIPO se puede identificar si el producto es una materia prima, producto terminado, juego, ensamble, servicio, gasto, etc.

Campo	Descripción	Tipo	Tipo liga
ISEQ*	Campo llave único	Liga a otras tablas	Directa
ICOD	Código	Alfanumérico	
IDESCR	Descripción	Alfanumérico	
ITIPO	Tipo	Numérico	
ILISTA1	Lista 1	Numérico	
ILISTA2	Lista 2	Numérico	
ILISTA3	Lista 3	Numérico	
ILISTA4	Lista 4	Numérico	
ILISTA5	Lista 5	Numérico	
IMINIMO	Stock Mínimo	Numérico	
IMAXIMO	Stock Máximo	Numérico	
ITIEMPO	Tiempo de proceso	Numérico	
ILISTA6	Lista 6	Numérico	
ILISTA7	Lista 7	Numérico	
ILISTA8	Lista 8	Numérico	
ILISTA9	Lista 9	Numérico	
IDESCGRUPO	Grupo de descuento	Numérico	
IFAM1	Código familia 1	Referencia liga a FFAM	Indirecta
IFAM2	Código familia 2	Referencia liga a FFAM	Indirecta
IFAM3	Código familia 3	Referencia liga a FFAM	Indirecta
IFAM4	Código familia 4	Referencia liga a FFAM	Indirecta
IFAM5	Código familia 5	Referencia liga a FFAM	Indirecta
IFAM6	Código familia 6	Referencia liga a FFAM	Indirecta
IFAM7	Código familia 7	Referencia liga a FFAM	Indirecta
IFAM8	Código familia 8	Referencia liga a FFAM	Indirecta
IFAM9	Código familia 9	Referencia liga a FFAM	Indirecta
IFAMA	Código familia A	Referencia liga a FFAM	Indirecta
IFAMB	Código familia B	Referencia liga a FFAM	Indirecta
IFAMC	Código familia C	Referencia liga a FFAM	Indirecta

IFAMD	Código familia D	Referencia liga a FFAM	Indirecta
ICT	Marca color y talla	Numérico	
IEAN	Código EAN	Alfanumérico	
ISTKACT	Existencia actual	Numérico	
IOFERDESDE	Inicio de oferta	Fecha	
IOFERHASTA	Fin de oferta	Fecha	
IPORCIEPES	Porcentaje IEPS	Numérico	
IUM	Unidad de medida	Alfanumérico	
ILUGAR	Lugar de origen	Alfanumérico	
ILISTA11	Precio F.O.B.	Numérico	
ICODPRV	Código interno del proveedor	Alfanumérico	
ICTA	Cuenta de contabilidad	Numérico	Indirecta
ICTADEV	Cuenta de contab. devolución	Numérico	Indirecta
ILOCALIZ	Localización en el almacén	Alfanumérico	
ICOLOR	Código color	Alfanumérico	
ICOMPOS	Composición	Alfanumérico	
ICONTROLPZAS	Marca control por piezas	Numérico	
IPRV	Proveedor a quien se compra	Referencia liga FPRV	Indirecta
ISEGUNDAS	Código de Segundas	Alfanumérico	
ITERCERAS	Código de Terceras	Alfanumérico	
ICTA3	Cuenta costo de venta	Numérico	
IMINIMOHASTA	Fecha para calculo mínimos	Fecha	
IPLANLISTA1	ILISTA 1 Planeada para POS	Numérico	
IMESA	Mesa para promociones	Numérico	
ICODSAT	Código SAT	Numérico	
IDESCATOMON	% Descuento en monedero	Numérico	
UCOD*	Clave unidad medida	Referencia liga a FUNIDAD	Directa

FDOC (Documentos)

La tabla **FDOC** contiene TODOS los **documentos** que se derivan de cualquier operación de compra, venta, inventarios físicos, traspasos, etc.

Esta tabla representa el encabezado de la operación y contiene la información general como es: fecha, importe, valor de los impuestos, descuentos, saldo, etc.

Cada registro es perfectamente distinguible ya que cuenta con el campo DITIPMV, que guarda los dígitos que identifican la operación que representa, por ejemplo las facturas se identifican con la letra F.

Esta tabla permite que todas las operaciones de control administrativo y de afectación monetaria convivan juntas, a través de una serie de 'banderas (**)' que identifican diferentes estatus del documento y se puede determinar con exactitud de qué tipo de registro se trata.

Esta tabla hereda la llave única de clientes o proveedores según corresponda a quien participe en la operación y crea una liga de forma natural++

Campo	Contenido	Tipo	Tipo liga
DSEQ*	Campo llave único	Liga a otras tablas	Directa
DNUM	Número / Folio	Alfanumérico	
DCANT	Saldo actual	Numérico	
DFECHA	Fecha	Fecha	
DVENCE	Vencimiento	Fecha	
DIVA	IVA (importe)	Numérico	
DDESC	Descuento (importe)	Numérico	
DBRUTO	Importe bruto	Numérico	
DCANTF	Importe final	Numérico	
DREFER	Referencia	Referencia liga a FPENC	Indirecta
DREFERELLOS	Número pedido de ellos	Alfanumérico	
DESCXC**	Bandera de saldo	Identificación saldo**	
DESINV**	Bandera de inventarios	Identificación afectación**	
DEFACT**	Bandera de documento	Identificación tipo doc.**	
DPAR1	Código clasificador 1	Referencia liga a FAG	Indirecta
DPAR2	Código clasificador 2	Referencia liga a FAG	Indirecta
DPAR3	Código clasificador 3	Referencia liga a FAG	Indirecta
DPAR4	Código clasificador 4	Referencia liga a FAG	Indirecta
DPAR5	Código clasificador 5	Referencia liga a FAG	Indirecta
DPAR6	Código clasificador 6	Referencia liga a FAG	Indirecta
DPAR7	Código clasificador 7	Referencia liga a FAG	Indirecta
DPAR8	Código clasificador 8	Referencia liga a FAG	Indirecta
DPAR9	Código clasificador 9	Referencia liga a FAG	Indirecta
DCOMI	Comisión por venta	Numérico	
DPZAS	Piezas	Numérico	
DALMACEN	Almacén	Referencia a FALMCAT	Indirecta
DPUNTOS	Puntos	Numérico	
DSTATUSCFD**	Bandera fiscal	Identificación estatus CFDI**	
DITIPMV	Identificador de transacción	Referencia a tipo mov.	Indirecta
DCANCELADA**	Estatus de cancelación	Bandera estatus	
CLISEQ++	Llave heredada de FCLI	Liga natural a FCLI++	Directa
PRVSEQ++	Llave heredada de FPRV	Liga natural a FPRV++	Directa

Vamos a explicar con más detalle los campos que denominamos **Banderas****, ya que su principal objetivo es identificar un registro de acuerdo con su valor y significado.

El uso adecuado de las banderas como filtros o condiciones, nos permitirá delimitar correctamente qué tipo de información estamos solicitando, estos campos o banderas serán primordialmente necesarios cuando pretendamos realizar consultas que nos devuelvan información sobre operaciones de venta, compra, inventarios, cuentas por cobrar o por pagar, entre algunas otras.

La siguiente tabla nos muestra el nombre de cada campo (**bandera**) el valor que puede tener y el resultado o interpretación de dicho valor:

Nombre campo	Si tiene valor:	Significado o interpretación
DESFACT	0	Es un documento que no se generó en ningún módulo de ventas, compras o producción.
	1	Es un documento generado desde el módulo facturación
	2	Es un documento generado desde el módulo recepciones
	3	Es un documento generado desde el módulo producción para pago a maquileros
	4	Es un documento generado desde el módulo de producción para el consumo de materiales
	8	Es un documento de complemento de pago timbrado
DESCXC	0	Es un documento que ya no tiene saldo pendiente
	1	Es un documento que tiene saldo pendiente de cobro a clientes
	2	Es un documento que tiene saldo pendiente de pago a proveedores
DESINV	0	Es un documento que no afectó el módulo de inventarios
	1	Es un documento que si afectó el módulo de inventarios
DCANCELADA	0	No está cancelado
	1	Si está cancelado
DSTATUSCFD	0	No es documento CFDI
	1	Documento de ventas pendiente timbre CFDI
	3	Documento de ventas timbrado y validado CFDI
	-1	Documento de compras pendiente de ligar a su XML
	-3	Documento de compras ya ligado a su XML
DITIPMV		El valor corresponderá al tipo de movimiento utilizado en la operación

Es importante explicar que un solo registro combina todas las banderas, y los valores contenidos nos dirán origen y afectación del mismo dentro del sistema, veamos los siguientes ejemplos que detallan como cada registro utiliza simultáneamente los diferentes valores que nos permiten determinar el 'estatus' de un documento dentro del sistema.

Ejemplo 1:

DNUM	DESFAC	DESCXC	DESINV	DCANCELADA	DSTATUSCFD	DITIPMV
C00000799	1	0	1	0	3	C

Veamos el significado de los valores que cada bandera tiene:

C00000799	DESFAC=1	Emitida desde modulo facturación
	DESCXC=0	No tiene saldo pendiente por aplicar ya fue aplicada
	DESINV=1	Afecta inventario posible devolución de cliente
	DCANCELADA=0	No está cancelada
	DSTATUSCFD=3	Está timbrada y validada fiscalmente
	DITIPMV='C'	El tipo de movimiento utilizado fue C

Interpretación del registro: “Es nota de crédito, emitida desde facturación, no tiene saldo pendiente, afecto al inventario, no está cancelada, ya fue timbrada fiscalmente”

Ejemplo 2:

DNUM	DESFAC	DESCXC	DESINV	DCANCELADA	DSTATUSCFD	DITIPMV
F00001	1	0	1	0	0	F

Veamos el significado de los valores que cada bandera tiene:

F00001	DESFAC=1	Emitida desde módulo facturación
	DESCXC=0	No tiene saldo pendiente por cobrar
	DESINV=1	Afecta a inventarios
	DCANCELADA=0	No está cancelada
	DSTATUSCFD=0	No es un documento para timbrar y validar fiscalmente
	DITIPMV='F'	El tipo de movimiento utilizado fue F

Interpretación del registro: “Es factura, emitida desde facturación, no tiene saldo pendiente, afecto al inventario, no está cancelada, no requiere timbre fiscal”

Ejemplo 3:

DNUM	DESFAC	DESCXC	DESINV	DCANCELADA	DSTATUSCFD	DITIPMV
F00001452	1	0	1	0	3	F

Veamos el significado de los valores que cada bandera tiene:

F00001452	DESFAC=1	Emitida desde módulo facturación
	DESCXC=0	No tiene saldo pendiente por cobrar
	DESINV=1	Afecta a inventarios
	DCANCELADA=0	No esta cancelada
	DSTATUSCFD=3	Está timbrada y validada fiscalmente
	DITIPMV='F'	El tipo de movimiento utilizado fue F

Interpretación del registro: “Es factura, emitida desde facturación, no tiene saldo pendiente, afecto al inventario, no está cancelada, ya fue timbrada fiscalmente”

Ejemplo 4:

DNUM	DESFAC	DESCXC	DESINV	DCANCELADA	DSTATUSCFD	DITIPMV
F00001459	1	0	0	1	0	F

Veamos el significado de los valores que cada bandera tiene:

F00001459	DESFAC=1	Emitida desde módulo facturación
	DESCXC=0	No tiene saldo pendiente por cobrar
	DESINV=1	Afecta a inventarios
	DCANCELADA=1	Está cancelada
	DSTATUSCFD=0	No es un documento para timbrar y validar fiscalmente
	DITIPMV='F'	El tipo de movimiento utilizado fue F

Interpretación del registro: “Es factura, emitida desde facturación, no tiene saldo pendiente, afecto al inventario, está cancelada, no está timbrada”

Ejemplo 5:

DNUM	DESFAC	DESCXC	DESINV	DCANCELADA	DSTATUSCFD	DITIPMV
L01460	1	1	1	0	0	L

Veamos el significado de los valores que cada bandera tiene:

L01460	DESFAC=1	Emitida desde módulo facturación
	DESCXC=1	Tiene saldo pendiente para cobrar a clientes
	DESINV=1	Afecta a inventarios
	DCANCELADA=0	No está cancelada
	DSTATUSCFD=0	No es un documento para timbrar y validar fiscalmente
	DITIPMV='L'	El tipo de movimiento utilizado fue L

Interpretación del registro: “Es remisión, emitida desde facturación, tiene saldo pendiente de cobro, afecto al inventario, no está cancelada, no requiere timbre”

Ejemplo 6:

DNUM	DESFAC	DESCXC	DESINV	DCANCELADA	DSTATUSCFD	DITIPMV
F00001543	1	1	1	0	3	F

Veamos el significado de los valores que cada bandera tiene:

F00001543	DESFAC=1	Emitida desde módulo facturación
	DESCXC=1	Tiene saldo pendiente para cobrar a clientes
	DESINV=1	Afecta a inventarios
	DCANCELADA=0	No está cancelada
	DSTATUSCFD=3	Está timbrada y validada fiscalmente
	DITIPMV='F'	El tipo de movimiento utilizado fue F

Interpretación del registro: “Es factura, emitida desde facturación, tiene saldo pendiente de cobro, afecto al inventario, no está cancelada, está timbrada y validada fiscalmente”

Ejemplo 7:

DNUM	DESFAC	DESCXC	DESINV	DCANCELADA	DSTATUSCFD	DITIPMV
F00001597	1	0	0	1	3	F

Veamos el significado de los valores que cada bandera tiene:

F00001597	DESFAC=1	Emitida desde módulo facturación
	DESCXC=0	No tiene saldo pendiente por cobrar
	DESINV=0	No afecto a inventario
	DCANCELADA=1	Está cancelada
	DSTATUSCFD=3	Fue timbrada y validada fiscalmente
	DITIPMV='F'	El tipo de movimiento utilizado fue F

Interpretación del registro: “Es factura, emitida desde facturación, no tiene saldo pendiente de cobro, no afecto al inventario, está cancelada, está timbrada y validada fiscalmente”

Ejemplo 8:

DNUM	DESFAC	DESCXC	DESINV	DCANCELADA	DSTATUSCFD	DITIPMV
F00002	0	1	0	0	0	F

Veamos el significado de los valores que cada bandera tiene:

F00002	DESFAC=0	No fue emitida desde módulo facturación
	DESCXC=1	Tiene saldo pendiente por cobrar a clientes
	DESINV=0	No afecto a inventario
	DCANCELADA=0	No está cancelada
	DSTATUSCFD=0	No es un documento para timbrar
	DITIPMV='F'	El tipo de movimiento utilizado fue F

Interpretación del registro: “Documento de cobro, no fue emitida desde facturación, tiene saldo pendiente de cobro, no afecto al inventario, no está cancelada, no requiere timbre fiscal”

Ejemplo 9:

DNUM	DESFAC	DESCXC	DESINV	DCANCELADA	DSTATUSCFD	DITIPMV
R00001	2	2	1	0	0	R

Veamos el significado de los valores que cada bandera tiene:

R00001	DESFAC=2	Emitida desde modulo recepciones
	DESCXC=2	Tiene saldo pendiente para pagar a proveedores
	DESINV=1	Afecto a inventarios
	DCANCELADA=0	No está cancelada
	DSTATUSCFD=0	No requiere comprobante XML
	DITIPMV='R'	El tipo de movimiento utilizado fue R

Interpretación del registro: “Documento de compras, fue emitida desde recepciones, tiene saldo pendiente de pago a proveedores, afecto al inventario, no está cancelada, no requiere timbre fiscal”

Ejemplo 10:

DNUM	DESFAC	DESCXC	DESINV	DCANCELADA	DSTATUSCFD	DITIPMV
R00006	0	2	0	0	0	R

Veamos el significado de los valores que cada bandera tiene:

R00006	DESFAC=0	No fue emitida en módulo recepciones
	DESCXC=2	Tiene saldo pendiente para pagar a proveedores
	DESINV=0	No afecto a inventarios
	DCANCELADA=0	No está cancelada
	DSTATUSCFD=0	No requiere comprobante XML
	DITIPMV='R'	El tipo de movimiento utilizado fue R

Interpretación del registro: “Documento para pagar a proveedores, no fue emitida en recepciones, tiene saldo pendiente de pago a proveedores, no afecto al inventario, no está cancelada, no requiere timbre fiscal”

Ejemplo 11:

DNUM	DESFAC	DESCXC	DESINV	DCANCELADA	DSTATUSCFD	DITIPMV
R00011	2	0	1	0	-3	R

Veamos el significado de los valores que cada bandera tiene:

R00011	DESFAC=2	Emitida desde módulo recepciones
	DESCXC=0	No tiene saldo pendiente de pago
	DESINV=1	Afecto a inventarios
	DCANCELADA=0	No está cancelada
	DSTATUSCFD=-3	Ya está ligada a su XML de compra
	DITIPMV='R'	El tipo de movimiento utilizado fue R

Interpretación del registro: “Documento de compras, fue emitida en recepciones, no tiene saldo pendiente de pago, afecto al inventario, no está cancelada, ya fue ligada a su comprobante XML”

Ejemplo 12:

DNUM	DESFAC	DESCXC	DESINV	DCANCELADA	DSTATUSCFD	DITIPMV
R00007	2	0	0	1	0	R

Veamos el significado de los valores que cada bandera tiene:

R00007	DESFAC=2	Emitida desde módulo recepciones
	DESCXC=0	No tiene saldo pendiente para pagar
	DESINV=0	No afecto a inventarios
	DCANCELADA=1	Está cancelada
	DSTATUSCFD=0	No requiere archivo XML
	DITIPMV='R'	El tipo de movimiento utilizado fue R

Interpretación del registro: “Documento de compras, fue emitida en recepciones, no tiene saldo pendiente de pago, no afecto al inventario, está cancelada, no requiere comprobante XML”

Ejemplo 13:

DNUM	DESFAC	DESCXC	DESINV	DCANCELADA	DSTATUSCFD	DITIPMV
R00015	2	2	1	0	-1	R

Veamos el significado de los valores que cada bandera tiene:

R00015	DESFAC=2	Emitida desde módulo recepciones
	DESCXC=2	Tiene saldo pendiente de pago a proveedores
	DESINV=1	Afecto a inventarios
	DCANCELADA=0	No está cancelada
	DSTATUSCFD=-1	Está pendiente de ligarse a su XML
	DITIPMV='R'	El tipo de movimiento utilizado fue R

Interpretación del registro: "Documento de compras, fue emitida en recepciones, tiene saldo pendiente de pago, afecto al inventario, no está cancelada, está pendiente de ligarse a su comprobante XML"

Ejemplo 14:

DNUM	DESFAC	DESCXC	DESINV	DCANCELADA	DSTATUSCFD	DITIPMV
IZ0000546	8	0	0	0	3	IZ

Veamos el significado de los valores que cada bandera tiene:

IZ0000546	DESFAC=8	Comprobante de pago de clientes
	DESCXC=0	No tiene saldo pendiente de cobro
	DESINV=0	No afecto a inventarios
	DCANCELADA=0	No está cancelada
	DSTATUSCFD=3	Fue timbrada y validada fiscalmente
	DITIPMV='IZ'	El tipo de movimiento utilizado fue IZ

Interpretación del registro: "Comprobante de pago, timbrada en facturación, no tiene saldo pendiente, no afecto al inventario, no está cancelada, ya fue timbrada y validada fiscalmente"

Como hemos podido observar, cada registro contenido en la tabla FDOC puede combinar diferentes valores en cada una de las banderas y con ello tener una interpretación diferente.

Algunas de las banderas nacen con un valor predeterminado y cambian por efecto de algún proceso operativo, el ejemplo es la bandera DESCXC que cuando una factura se emite, se genera con valor 1 y al ser aplicado el cobro cambia a cero. En el caso de ser una recepción de compra, nace con valor 2 y al realizarse el pago al proveedor, cambia a cero.

Recordemos que la estructura de la base Proscail es de tipo jerárquico como lo mencionamos anteriormente.

Cada documento que se genera dentro de la base de datos, heredará la llave del cliente o proveedor involucrado, y el campo CLISEQ o PRVSEQ indicará la llave que lo liga, un documento solo puede tener una llave heredada, es decir, un documento jamás podrá estar ligado de forma simultánea a un cliente y a un proveedor a la vez.

En la siguiente tabla juntamos todos los ejemplos anteriores y podemos observar claramente que las llaves naturales++ que se heredan y ligan al documento con su cliente o proveedor respectivo de forma directa, solo una de ellas mostrara el valor correspondiente, ya que un documento solo puede tener una liga a la vez.

DNUM	DESFAC	DESCXC	DESINV	DCANCELADA	DSTATUSCFD	DITIPMV	CLISEQ++	PRVSEQ++
F00001452	1	0	1	0	3	F	3357	0
R00001	2	2	1	0	0	R	0	524
C00000799	1	0	1	0	3	C	1452	0
F00001	1	0	1	0	0	F	548	0
R00015	2	2	1	0	-1	R	0	143
R00006	0	2	0	0	0	R	0	231
IZ0000546	8	0	0	0	3	IZ	75	0
R00011	2	0	1	0	-3	R	0	5
F00001597	1	0	0	1	3	F	241	0
F00001543	1	1	1	0	3	F	88	0
R00007	2	0	0	1	0	R	0	931

FAXINV (Auxiliar de Inventarios)

La tabla **FAXINV** contiene TODOS los **registros que detallan o componen el historial de cada producto o servicio** que fue utilizado en las operaciones de compra o venta.

En esta tabla existe el campo AITIPMV que sirve para identificar el tipo de operación que afecto al producto o servicio, dicho campo es de tipo alfanumérico y típicamente podremos observar letras que se asocian al documento u operación realizada.

Campo	Descripción	Tipo	Tipo liga
AISEQ*	Campo llave único	Liga hacia otras tablas	
AITIPMV	Identificador de tipo de movimiento	Alfanumérico	
AIPRECIO	Precio neto	Numérico	
AICOSTO	Costo	Numérico	
AIPREBR	Precio bruto	Numérico	
AIPZAS	Campo de uso múltiple por partida	Alfanumérico	
AIDESCTO	Descuento por línea	Numérico	
AICANTF	Cantidad de venta	Numérico	
AIMES	Mes	Numérico	
AIALMACEN	Almacén	Referencia a FALMCAT	Indirecta
AIPED	Secuencia de FPLIN	Referencia a tabla FPLIN	Directa
AUISEQ++	Llave heredada de FUSERS	Referencia a catalogo usuarios++	Directa
ISEQ++	Llave heredada de FINV	Liga natural al producto++	Directa
CLISEQ++	Llave heredada de FCLI	Liga natural al cliente++	Directa
PRVSEQ++	Llave heredada de FPRV	Liga natural al proveedor++	Directa
DSEQ++	Llave heredada de FDOC	Liga natural al documento++	Directa

FAX (Auxiliar de documentos de clientes o proveedores)

La tabla **FAX** contiene TODOS los **registros que detallan o componen el historial de cada documento** que se ha generado con los clientes o proveedores y que implican un valor monetario en la operación.

Existe el campo AITIPMV que sirve para identificar el tipo de operación que se realizó, dicho campo es de tipo alfanumérico y típicamente podremos observar letras que se asocian al documento o la operación realizada.

Campo	Descripción	Tipo	Tipo liga
ASEQ	Campo llave único	Liga a otras tablas	
ACANT	Importe del movimiento	Numérico	
AITIPMV	Tipo de movimiento	Alfanumérico	
AFECHA	Fecha del movimiento	Fecha	
AREFPAG	Referencia de pago	Alfanumérico	
AMES	Este mes	Numérico	
AUISEQ++	Llave usuario que capturo	Referencia a catalogo usuarios	Directa
CLISEQ++	Llave heredada de FCLI	Liga natural al cliente++	Directa
PRVSEQ++	Llave heredada de FPRV	Liga natural al proveedor++	Directa
DSEQ++	Llave heredada de FDOC	Liga natural al documento++	Directa
BASEQ++	Llave heredada de FBANMOV	Liga natural a mov. contable++	Directa

FPENC (*Encabezados de pedidos y órdenes de compra*)

La tabla **FPENC** contiene TODOS los registros de **pedidos u órdenes de compra con clientes y proveedores** respectivamente, estos registros son el encabezado de dicha operación y detallan la información básica como es: folio o número, fecha de emisión, fecha de entrega y vencimiento, importe, descuentos, valor de los impuestos, valor total, etc.

Dentro de esta tabla, conviven todas las operaciones que representan un promesa de venta o compra y a través del campo PESPEDIDO que es una **bandera**** nos permite identificar de qué tipo de operación se trata o en que estatus se encuentra.

Campo	Descripción	Tipo	Tipo liga
PESEQ*	Campo llave único	Liga a otras tablas	Directa
PENUM	Número de pedido	Alfanumérico	
PENUMELLOS	Núm. de pedido del cliente	Alfanumérico	
PEFECHA	Fecha captura	Fecha	
PEDESDE	Fecha entrega	Fecha	
PEVENCE	Fecha vencimiento	Fecha	
PEBRUTO	Importe bruto	Numérico	
PEDESC	Importe descuento	Numérico	
PEIVA	Importe IVA	Numérico	
PECANT	Importe total	Numérico	
PEPAR1	Código de clasificador 1	Referencia liga a FAG	Indirecta
PEPAR2	Código de clasificador 2	Referencia liga a FAG	Indirecta
PEPAR3	Código de clasificador 3	Referencia liga a FAG	Indirecta
PEPAR4	Código de clasificador 4	Referencia liga a FAG	Indirecta
PEPAR5	Código de clasificador 5	Referencia liga a FAG	Indirecta
PEPAR6	Código de clasificador 6	Referencia liga a FAG	Indirecta
PEPAR7	Código de clasificador 7	Referencia liga a FAG	Indirecta
PEPAR8	Código de clasificador 8	Referencia liga a FAG	Indirecta
PEPAR9	Código de clasificador 9	Referencia liga a FAG	Indirecta
PESPEDIDO**	Bandera tipo pedido	Bandera de identificación	
PEUSRALTA	Código usuario que capturo	Referencia liga a FUSERS	Indirecta
CLISEQ++	Llave heredada de FCLI	Liga natural al cliente++	Directa
PRVSEQ++	Llave heredada de FPRV	Liga natural al proveedor++	Directa

Para poder identificar con total exactitud qué tipo de registro se trata siempre se debe usar la bandera PESPEDIDO para cualquier filtro y de esta forma obtener adecuadamente la información respectiva.

La siguiente tabla nos muestra el nombre del campo (**bandera****) el valor que puede tener y el resultado o interpretación de dicho valor:

Nombre campo	Si tiene valor:	Significado o interpretación
PESPEDIDO	1	Es un PEDIDO DE CLIENTES
	2	Es una ORDEN DE COMPRA para PROVEEDORES
	3	Es una ORDEN DE FABRICACION O PRODUCCIÓN interna
	4	Es un PEDIDO DE CLIENTES en estatus COTIZADO
	5	Es una ORDEN DE COMPRA para PROVEEDORES en estatus COTIZADA

FPLIN (Detalle de líneas de pedidos y órdenes de compra)

La tabla **FPLIN** contiene los **renglones que detallan los pedidos u órdenes de compra** que se encuentran en proceso de surtido o recibo respectivamente.

Campo	Descripción	Tipo	Tipo Liga
PLSEQ*	Campo llave único	Liga para otras tablas	
PLCANT	Cantidad pedida por línea	Numérico	
PLPRECI	Precio unitario por línea	Numérico	
PLSURT	Cantidad surtida por línea	Numérico	
PLDESC	Descuento	Numérico	
PLASIGNADO	Cantidad asignada	Numérico	
PLSUC	Numero sucursal destino	Numérico	Indirecta
PESEQ++	Llave heredada de FPENC	Liga natural al pedido++	Directa
CLISEQ++	Llave heredada de FCLI	Liga natural al cliente++	Directa
PRVSEQ++	Llave heredada de FPRV	Liga natural al proveedor++	Directa
ISEQ++	Llave heredada de FINV	Liga natural al producto++	Directa

FBENC (Catalogo de cuentas contables y bancarias)

La tabla **FBENC** contiene TODOS los registros de las **cuentas contables y bancarias** del sistema. Y que son utilizados en la realización de pólizas contables.

Campo	Descripción	Tipo	Tipo liga
BSEQ*	Campo llave único	Liga a otras tablas	Directa
BCOD	Código de la cuenta	Numérico	
BCTA	Número de la cuenta	Alfanumérico	
BNOMBRE	Nombre de la cuenta	Alfanumérico	
BGERENTE	Nombre del Gerente	Alfanumérico	
BTELEFONO	Teléfono del banco	Alfanumérico	
BTIPO	Tipo (Banco, Gasto, Otro)	Numérico	
BNATUR	Bandera de naturaleza	Numérico	
BNOPOLIZA	No poder utilizar en pólizas	Numérico	
BPRESUP	Presupuestable	Numérico	
BDEPOSITOS	Poder hacer Depósitos	Numérico	
BPAGOS	Poder hacer Pagos	Numérico	
BFISCALINFLA	Para ajuste anual 0=No, 1= Crédito, 2= Débito	Numérico	
BDEDUCIETU	Deducible IETU	Numérico	
BNODEDUCIIVA	No deducible IVA	Numérico	
BMAYOR	Cuenta agrupadora	Numérico	
BSATGRUPO	Clave agrupación SAT	Alfanumérico	
BSATBANCO	Clave banco SAT	Alfanumérico	
BSATBCOEXT	Nombre banco extranjero SAT	Alfanumérico	
BSATMETPAG	Clave método pago SAT	Alfanumérico	
BSATMONEDA	Clave moneda SAT	Alfanumérico	
BRFC	RFC banco SAT	Alfanumérico	

FPOLIZA (Encabezados de pólizas contables)

La tabla **FPOLIZA** contiene los encabezados de las pólizas contables y que con el primer dígito de su número permiten saber tipo de póliza es (diario, ingresos o egresos), así como fecha de captura y su estatus de aplicación, a través del campo POAPLICADA se puede identificar si tiene afectación en la contabilidad.

Campo	Descripción	Tipo	Tipo Liga
POSEQ*	Campo llave único	Liga a otras tablas	Directa
PONUM	Número de la póliza	Alfanumérico	
PODESCR	Descripción de la póliza	Alfanumérico	
POFECHA	Fecha de la póliza	Fecha	
POAPLICADA	Bandera de aplicación	Numérico	
POBENEF	Beneficiario	Alfanumérico	
POCHEQUE	Número de Cheque	Alfanumérico	
POMES	Mes de generación de la póliza	Numérico	
POYEAR	Año	Numérico	

FBANMOV (Detalle de movimientos de pólizas contables)

La tabla **FBANMOV** contiene los renglones que detallan las afectaciones contables que se realizan en las pólizas.

Campo	Descripción	Tipo	Tipo liga
BASEQ*	Campo llave único	Liga para otras tablas	Directa
BATIP	Tipo de operación	Alfanumérico	
BAIMPOR	Importe de cargo	Numérico	
BABENEF	Beneficiario	Alfanumérico	
BAIMPORNEG	Importe de abono	Numérico	
BAOK	Marca de movimiento conciliado	Booleano	
BACENCOS	centro de costo afectado	Referencia a FALMCAT	Indirecta
BSEQ++	Llave heredada de FBENC	Liga natural a FBENC++	Directa
POSEQ++	Llave heredada de FPOLIZA	Liga natural a FPOLIZA++	Directa

FAG (Parámetros de clasificación)

La tabla **FAG** contiene TODOS los **parámetros de clasificación** que se definen para las diferentes áreas operativas del sistema como: Ventas, compras, contabilidad, punto de venta, etc. Dichos parámetros se asocian dentro de las operaciones los módulos respectivos a los que van enfocados y se usan para en la creación de consultas o análisis de información.

Campo	Descripción	Tipo	Tipo liga
AGTNUM*	Campo llave único	Referencia para otras tablas	Indirecta
AGDESCR	Descripción	Alfanumérico	
AGT	Grupo del parámetro	Numérico	
AGNUM	Código parámetro clasificador	Alfanumérico	

FALMCAT (Catalogo de almacenes)

La tabla **FALMCAT** contiene registros que se usan como **almacenes o centros de costo** y que se usan para registrar operaciones en otros módulos del sistema y se considera una tabla de soporte general.

Campo	Descripción	Tipo	Tipo liga
CATALM*	Campo llave único	Referencia para otras tablas	Indirecta
CATDESCR	Descripción	Alfanumérico	
CATCOD	Código para centros de costo	Alfanumérico	

FFAM (Catalogo de clasificador para familias de productos)

La tabla **FFAM** contiene los **códigos de clasificación que sirven para identificar los productos en sus diferentes categorizaciones** y que se utilizan para creación de reportes y análisis, siendo una tabla de soporte general.

Campo	Descripción	Tipo	Tipo liga
FAMTNUM*	Campo llave único	Referencia para otras tablas	Indirecta
FAMNUM	Código parámetro	Alfanumérico	
FAMDESCR	Descripción	Alfanumérico	
FAMT	Numero de clasificador	Numérico	
FAMOBS	Observaciones	Alfanumérico	

FUNIDAD (Unidades de medida)

La tabla **FUNIDAD** contiene los **códigos de las unidades de medida y/o diferentes tipos de tallas** que sirven para definir las unidades de control básico para los productos y para la ropa respectivamente, se considera una tabla de soporte general

Campo	Descripción	Tipo	Tipo liga
UCOD*	Campo llave único	Referencia para otras tablas	Indirecta
UDESCR	Descripción	Alfanumérico	

FCAJAS (Control de series o rollos)

La tabla **FCAJAS** contiene los registros que sirven para **control de series o rollos en el manejo de inventarios** y conocer su disponibilidad y trazabilidad.

Campo	Descripción	Tipo	Tipo liga
CAJCANT	Cantidad	Numérico	
CAJSERIE	Número	Numérico	
CAJPEDIDO	Pedido	Referencia a tabla FPENC	Indirecta
CAJFACTURA	Factura	Referencia a tabla FDOC	Indirecta
CAJALM	Almacén	Referencia a tabla FALMCAT	Indirecta
CAJRECEPCION	Recepción	Referencia a tabla FDOC	Indirecta

FALM (Existencias por almacén)

La tabla FALM contiene TODOS los registros que detallan las **existencias de cada producto en los diferentes almacenes donde este se encuentra.**

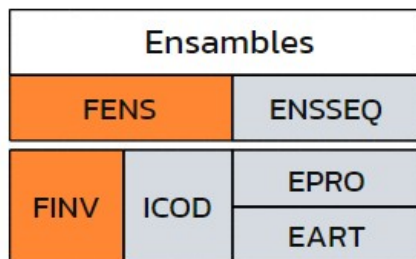
Campo	Descripción	Tipo	Tipo liga
ALMKEY*	Llave interna de producto y almacén	Llave interna	
ALMCANT	Existencia	Numérico	
ALMNUM	Número de bodega	Referencia a tabla FALMCAT	Indirecta
ALMMINIMO	Mínimo	Numérico	
ALMMAXIMO	Máximo	Numérico	
ALMDETDAS	Sugerencia de traspaso	Numérico	
ALMTRANSITO	Tránsito	Numérico	
ISEQ++	Llave heredada de tabla FINV	Liga natural a FINV	Directa

FENS (Ensamblés)

La tabla FENS contiene los registros de los **componentes de fabricación para un producto,** indicando el material necesario y las cantidades requeridas.

Campo	Descripción	Tipo	Tipo liga
EPRO*	Código de producto a fabricar	Referencia a tabla FINV	Indirecta
EART*	Código del componente o material	Referencia a tabla FINV	Indirecta
ECANT	Cantidad requerida	Numérico	
EORDEN	Orden del Proceso	Numérico	
EUSO	Lugar en donde se usa	Alfanumérico	

Esta tabla se considera una tabla de soporte ya que no tiene liga directa con ninguna tabla, usando los campos EPRO que es una referencia del código de producto terminado y EART que es la referencia del código de materia prima, realiza una doble conexión indirecta a la tabla FINV para poder obtener la descripción de cada producto.



Un ensamble está relacionado con la tabla FINV, en este caso 2 veces, una al producto terminado y la otra al material que se usa; por medio de este registro podemos indicar que es una camisa hecha de tela.

Modelos de conexión entre tablas

Después de haber explicado las principales tablas de la base de datos Proscail y algunos de sus campos, basados en el modelo gráfico de la base de datos que vimos al inicio de este capítulo, vamos a ejemplificar que tablas se ligan durante algunos de los procesos para obtener información del sistema.

Recordemos que la conexión o ligas entre tablas, se da en dos formas principalmente:

Conexión directa: significa que de acuerdo a su jerarquía y de forma natural utilizando las llaves primarias que se heredan entre tablas, estas se pueden ligar.

Conexión indirecta: se realiza mediante un campo que contiene una referencia que puede usarse para poder ligar a otra tabla.

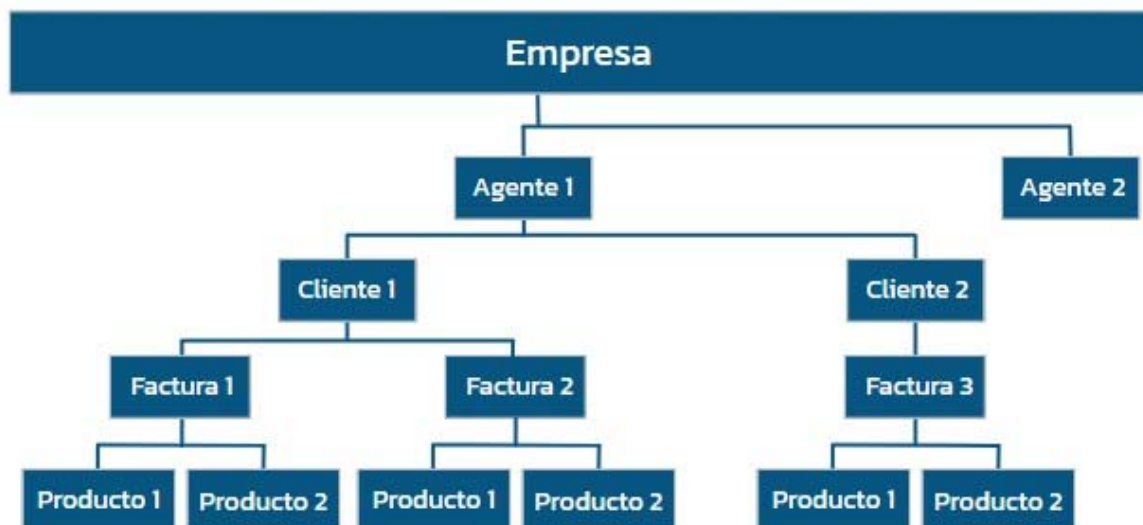
Cualquiera de los dos tipos de conexión que se realice, permitirá usar esa misma conexión para definir filtros o condiciones dentro de una consulta.

A continuación voy a explicar lo que son las conexiones y posteriormente como se hace una conexión directa e indirecta.

Para pensar en la lógica de las conexiones entre las diferentes tablas pensemos en un ejemplo de una empresa:

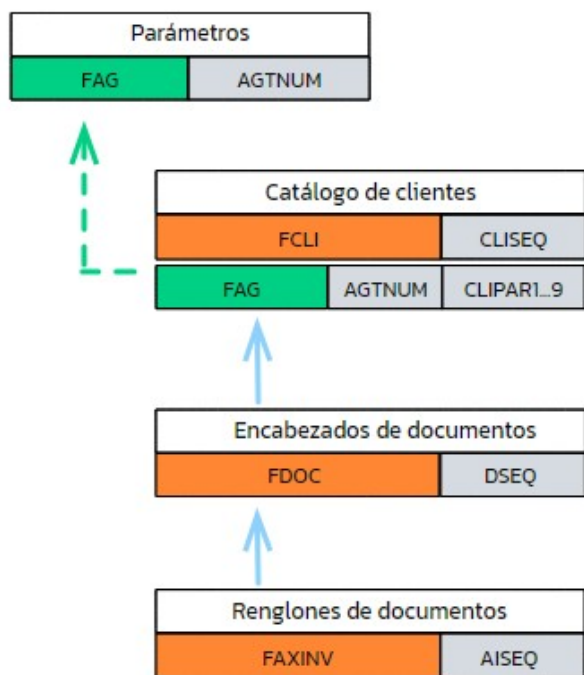
Una empresa puede tener varios vendedores, cada vendedor atender a varios clientes, cada cliente puede tener varias facturas y cada factura puede tener varios productos.

Esto se puede ejemplificar en un diagrama de tipo jerárquico como el siguiente:



En este diagrama se pueden ver los diferentes datos y como se relacionan entre si según lo describimos en el párrafo anterior.

A continuación voy a explicar cómo se ve esta estructura en nuestro diagrama de Datos de Proscat



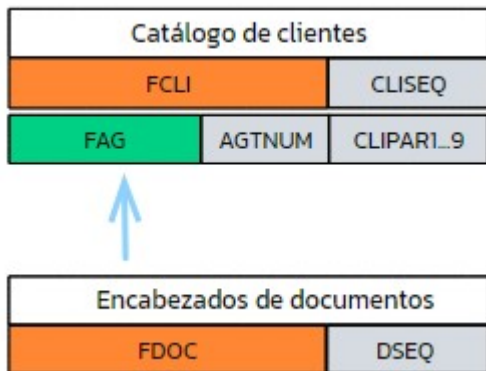
En el diagrama se puede ver la jerarquía donde entre más arriba está una tabla, representa que es de mayor nivel y entre más abajo, que es de mayor detalle.

Las flechas representan que hay una relación entre las tablas, en donde el inicio de la flecha representa una tabla que puede contener varios registros relacionados a la tabla hacia donde la flecha apunta.

Ahora veamos los tipos de conexiones:

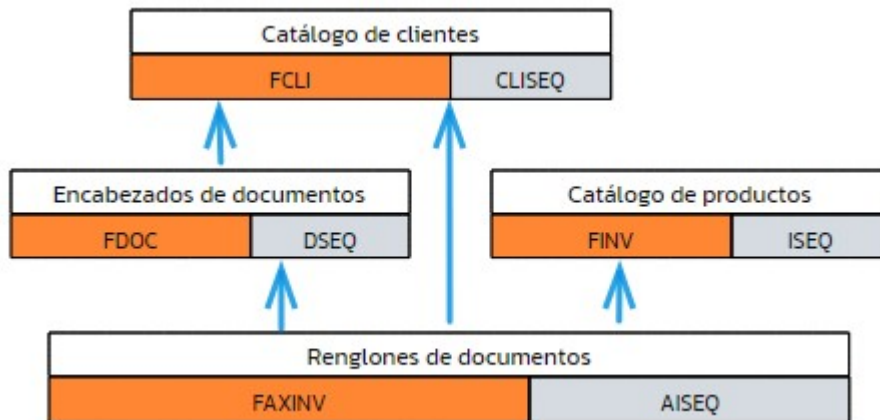
CONEXIÓN DIRECTA

Están representadas en el diagrama por las flechas azules.



La forma en la que se hacen estas conexiones es a través de la secuencia de la tabla padre, siempre la tabla hijo hereda la secuencia de su tabla padre.

En este caso lo ejemplificamos con las tablas FCLI y FDOC, donde la conexión será a través del campo CLISEQ que es la secuencia de la tabla FCLI. Cada registro de la tabla FDOC heredará la secuencia de la tabla FCLI, esta información en la tabla FDOC nos permitirá posteriormente identificar quién es el cliente de cualquier documento.



Una tabla hijo puede estar conectada a varias tablas de jerarquía superior, pero únicamente a un registro de la tabla superior (padre).

Ejemplo:

Una factura puede tener varios renglones y cada renglón pertenece a un cliente y a un producto. Por lo tanto, el registro donde se guarda un renglón está ligado a un cliente, a un documento y a un producto.

Para lograr esto la tabla FAXINV, que es donde se guardan los renglones de una factura, hereda las secuencias CLISEQ, DSEQ, ISEQ para identificar al cliente al que pertenece, al documento y al producto.

CONEXIÓN INDIRECTA

La conexión indirecta entre tablas no se realiza usando la secuencia de la tabla padre, sino a través de otro dato almacenado en el registro de la tabla Hijo.

Las tablas que permiten conexiones indirectas son:

Parámetros		Tipos de movimientos		Catálogo de almacenes		Usuarios		Familias	
FAG	AGTNUM	FTIPMV	TICLA	FALMCAT	CATALM	FUSERS	USRSEQ	FFAM	FAMTNUM

Todas estas tablas son de tipo catálogos y generalmente tienen información más detallada del dato que contiene el registro, por ejemplo:

La tabla de agentes tiene el código de los agentes y también los nombres de los mismos, la tabla de almacenes contiene toda la relación de cada uno de los almacenes.

Cada una de estas tablas tiene el nombre de ella en un campo con un color diferente, ya que usaremos esta guía de colores para identificarlas rápidamente, del lado derecho encontramos el nombre del campo que usaremos para relacionarla con otras tablas.

Por ejemplo, para localizar un registro de la tabla FTIPMV usaremos el campo TICLA

Para ejemplificar este caso tomemos las tablas FCLI y la tabla FAG:

La tabla FCLI guarda los clientes.

La tabla FAG guarda los vendedores (y muchas cosas más que veremos más adelante).

Si queremos identificar al vendedor que atiende a un cliente, podemos ver que la tabla FCLI tiene en el diagrama a la tabla FAG con el campo AGTNUM y el dato CLIPAR1...

Catálogo de clientes / proveedores		
FCLI / FPRV	CLISEQ / PRVSEQ	
FAG	AGTNUM	CLIPAR1..9

En este caso, sabiendo que el código de los agentes se guarda en el campo CLIPAR1 podemos identificar al vendedor correspondiente en la tabla FAG



Este tipo de tablas son ATRIBUTOS QUE SIRVEN PARA AGRUPAR O PARA FILTRAR
Son la base de poderle dar inteligencia a la información

Veamos otro ejemplo de una conexión indirecta ahora usando la tabla de productos y la conexión a las familias, esta se hace usando los campos: IFAM1, IFAM2....IFAMK de la tabla FINV y se conectan a la tabla FFAM con el campo FAMTNUM



Al igual que con las conexiones directas, una tabla puede estar conectada (relacionada) con varias tablas en forma indirecta.

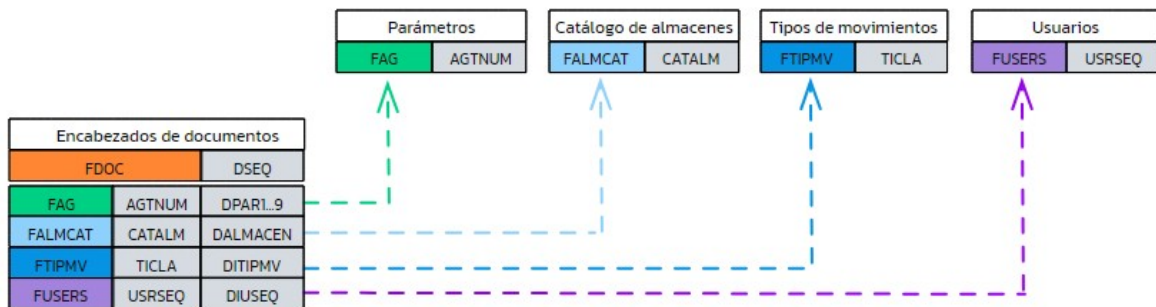
Por ejemplo:

Una factura puede estar relacionada a un vendedor, un almacén.

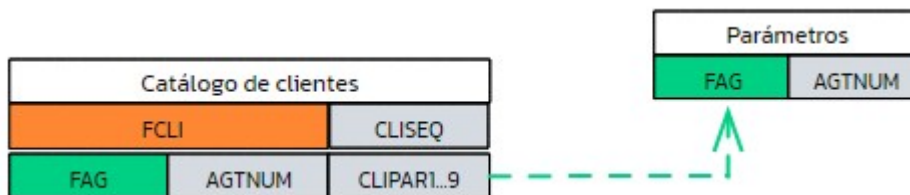
Ser de un tipo (Factura, Nota de crédito), y haber sido hecha por un usuario.

Cuando esto sucede en la tabla aparecerán las diferentes tablas a las que puede estar conectada.

En este ejemplo la tabla FDOC, está relacionada con las tablas FAG, FALMCAT, FTIPMV, FUSERS



En algunos casos, podemos ver que una tabla tiene varios campos de conexión por ejemplo en el diagrama principal, la tabla de FCLI está representada de la siguiente manera para mostrar su conexión múltiple a la tabla FAG.

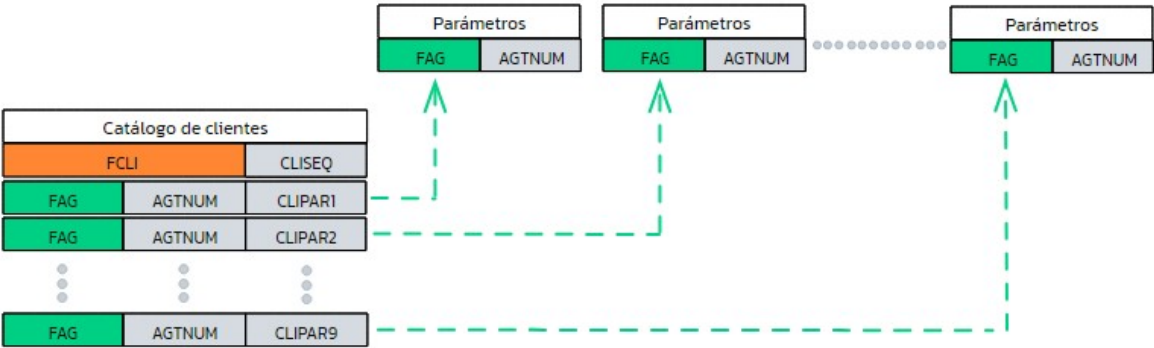


Como podemos observar, los campos de conexión de la tabla FCLI hacia la tabla FAG están marcados como CLIPAR1...9, esto significa que se pueden usar como conexiones los campos: CLIPAR1, CLIPAR2, CLIPAR3 hasta el 9 que se pueden usar para ligar al agente del cliente, la ciudad y el estado del cliente.

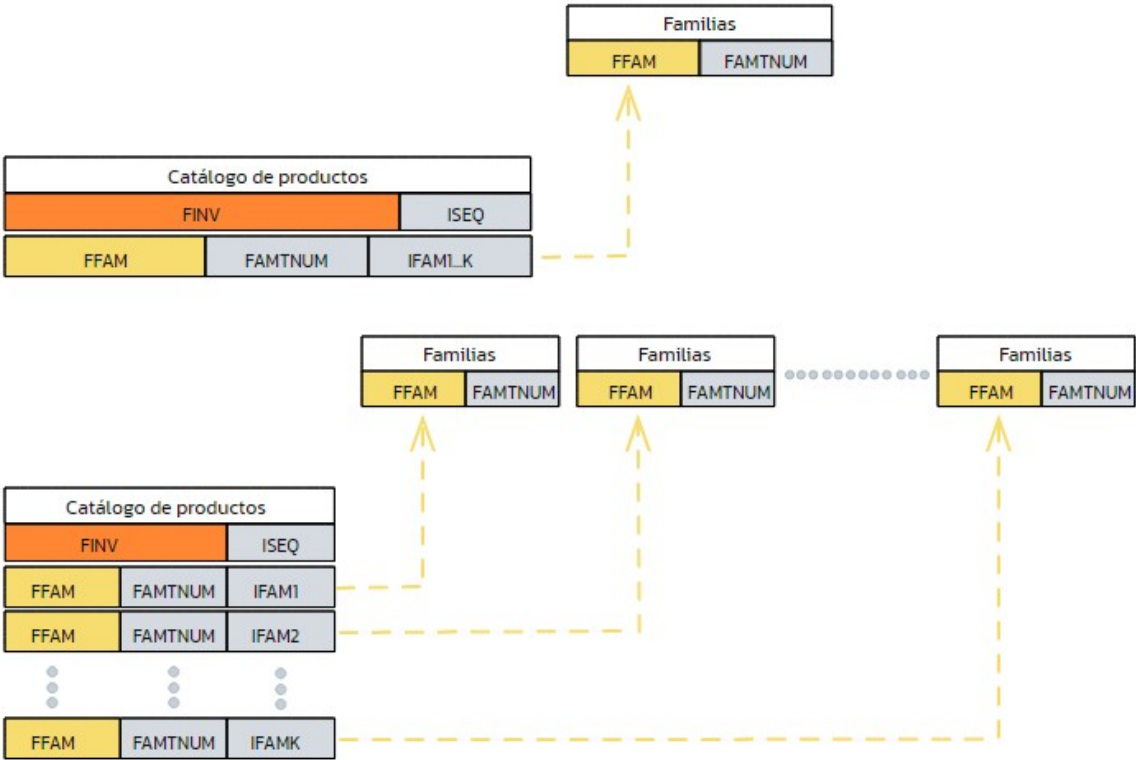
Si queremos generar una consulta en donde tengamos el nombre del cliente con el nombre del vendedor, su ciudad y estado, deberemos de hacer más de una conexión a la tabla FAG que es donde se guarda el catálogo de clasificadores de clientes.

En este caso, debemos pensar que la tabla de clientes tiene nueve conexiones a la tabla FAG, una por cada CLIPAR...

Esta relación la podríamos ver de la siguiente manera:



Encontramos otro caso similar de múltiples conexiones entre la tabla FINV que tiene las familias 1 a la 9 y la A a la K, en este caso los nombres de las familias están en la tabla FFAM.



Dentro de las conexiones indirectas hay un segundo tipo de conexión a otras tablas pero en este caso que no son catálogos como en los ejemplos anteriores.

Por ejemplo una factura puede estar relacionada con un pedido. Este tipo de relación las representamos con un recuadro naranja con las letras en blanco.

En este recuadro aparece el nombre de la tabla, en seguida el nombre del campo de la tabla y por último el nombre del campo de la tabla con el cual se hace la relación con la primera tabla, como lo muestra la siguiente gráfica:



En este ejemplo vemos como una factura (FDQC) está relacionada a su pedido (FPENC) usando los campos DREFER y PENUM de cada tabla respectivamente.

Otros ejemplos de este tipo de conexión se pueden ver en las tablas de ensambles (FENS) donde un ensamble está relacionado con la tabla FINV, en este caso dos veces, una al producto terminado y la otra al material que se usa por medio de este registro podemos indicar que una camisa está hecha de tela.

Ensambls		
FENS		ENSSEQ
FINV	ICOD	EPRO
FINV	ICOD	EART

Un caso más de uso de este tipo de conexión lo encontramos en el manejo de piezas donde una pieza de tela tiene una recepción de compra y una factura con su venta, en este caso la tabla FCAJAS tiene dos conexiones a la tabla FDOC uno relacionando la compra con el campo CAJRECEPCION y el otro relacionándolo con la factura con el campo CAJFACTURA, como lo podemos ver en la gráfica siguiente:

Cajas		
FCAJAS		CAJSEQ
FDOC	DNUM	CAJRECEPCION
FDOC	DNUM	CAJFACTURA

Finalmente en nuestro diagrama principal, encontramos los campos tipo bandera que representan en cada tabla los diferentes tipos de documentos que en ellas se pueden guardar, por ejemplo en la tabla FDOC y sus tablas conectadas se guardan facturas de clientes y también recepciones de proveedores, entonces surge la pregunta de ¿cómo podemos distinguir los registros de un tipo y del otro?, para esto están los campos que hemos denominado banderas que son indicadores del tipo de dato almacenado en un registro.

En nuestro diagrama los campos de este tipo están representados por los recuadros de color gris. Estos campos nos servirán básicamente como filtros para extraer la información que requerimos de la base de datos.

Por ejemplo: todos los registros de la tabla FDOC cuyo campo DESFACT tenga valor 1, representan todos los documentos que contienen una factura y los registros de la misma tabla en donde el campo sea DESCXC con valor 2, representan todos los documentos que son una cuenta por pagar.

Encabezados de documentos		
FDOC		DSEQ
DESFACT	1= Factura	2= Recepción
DESCXC	1= CXC	2= CXP
DESINV	1= Afecta inventario	

Encabezados de pedidos		
FPENC		PESEQ
PESPEDIDO	1= Pedido	4= Cot. Pedido
	2= O.C.	5= Cot. O.C.
	3= Orden de producción	

Las dos tablas que tienen campos tipo banderas son la tabla FDOC y la tabla FPENC y estos serán descritos más adelante.

En esta sección veremos cómo interactúan las diferentes tablas de la base de datos para representar a los diferentes tipos de catálogos, documentos y transacciones de Proscái.

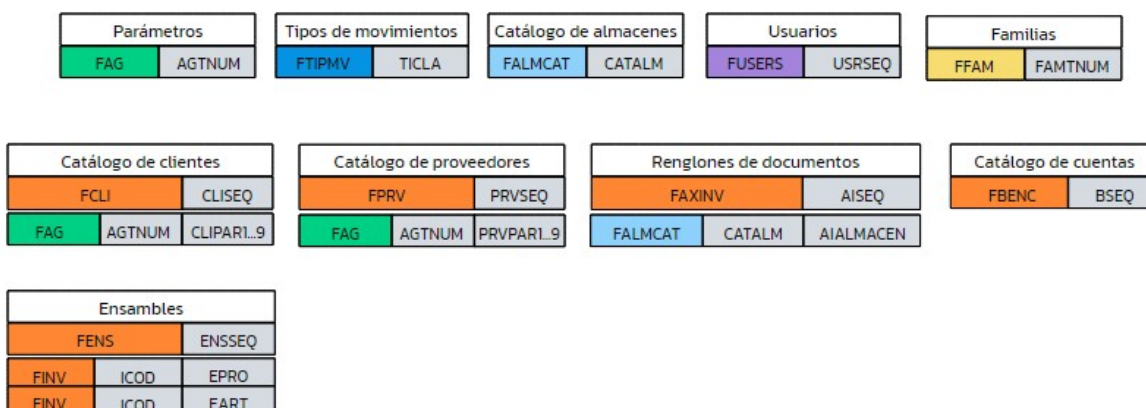
La información que se procesa en una empresa es compleja y no siempre se puede guardar en una sola tabla, sino que en muchas ocasiones se requiere de varias tablas para representar algún documento.

En esta sección, veremos con esa perspectiva como se guardan los diferentes tipos de catálogos (clientes, proveedores, productos, familias, etc.) y los diferentes tipos de documentos (facturas, recepciones, pólizas, pedidos, etc.) de Proscái.

Para este caso dividiremos las estructuras en tres: Catálogos, documentos y almacenamiento de datos.

Catálogos

Los Principales catálogos que se almacenan en la base de datos de Proscái son los siguientes:



En cada uno de ellos se guarda la información pertinente al tipo de dato que se requiere, muchos de estos catálogos solo requieren de una tabla y algunos otros están relacionados a otras, como en el caso del catálogo de productos, se ve una relación al catálogo de familias FFAM, ya que un producto, puede estar ligado a una o varias familias, lo mismo sucede con los catálogos de clientes y Proveedores ya que están ligados a sus clasificadores.

La estructura del catálogo de ensamblados es un poco más compleja ya que en ella se guardan los componentes de un producto en donde un producto está formado por varias materias primas, es por esto que en su estructura se guardan las relaciones hacia el catálogo de productos como producto y como materia prima, cabe mencionar que para guardar la estructura de un producto se requiere un registro para cada material que se utiliza para fabricarlo.

Documentos

Los Principales documentos que se almacenan en la base de datos de Proscái son los siguientes:

Documento	Descripción	Tipo
Movimientos de inventarios	Cualquier documento que afecte la existencia de productos	Facturas, Recepciones, movs. de inventario, entradas de producción, salidas de producción, inventarios físicos
Movimientos de cuentas por cobrar o cuentas por pagar	Cualquier documento que incremente la deuda de un cliente, un proveedor o un acreedor	Facturas, recepciones, pago a proveedores, depósitos de clientes, movimientos cxc, cxp
Facturas	Todos los documentos del módulo de facturación	facturas, notas de crédito, devoluciones de clientes
Recepciones	Todos los documentos del módulo de recepciones	recepciones, notas de crédito, devoluciones de proveedores
Pedidos	Los pedidos de clientes y cotizaciones a los mismos	Promesa de venta
Órdenes de compra	Las órdenes de compra a proveedores y las cotizaciones a los mismos	Promesa de compra
Pólizas	Todas las pólizas de contabilidad	Póliza contable

Por su naturaleza estos documentos o transacciones no se guardan en una solo tabla de Proscái sino que se guardan en diferentes tablas.

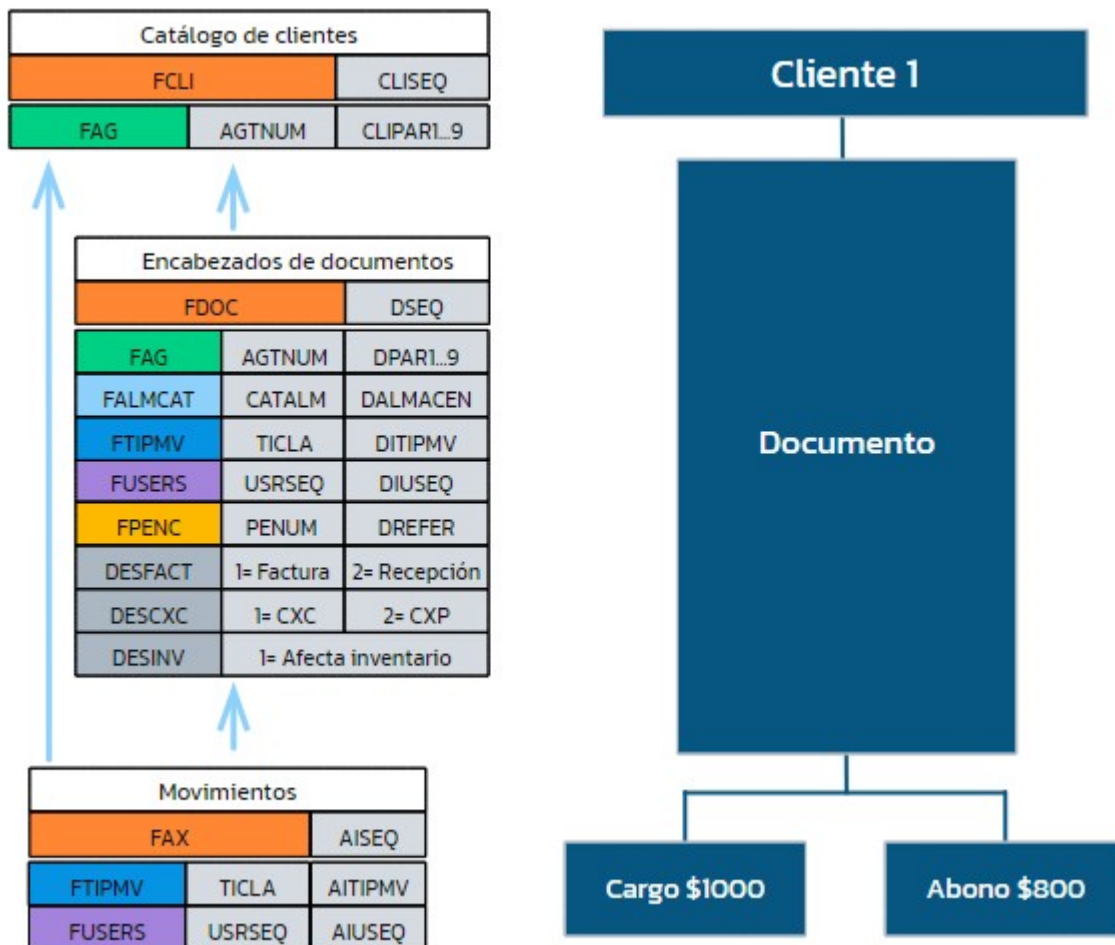
Pongamos como ejemplo una factura, una factura tiene un encabezado donde va la fecha de la misma, el cliente, el vendedor, la fecha de vencimiento, pero la misma puede tener varios renglones de productos, estar relacionada a un pedido y generar una cuenta por cobrar cuando vendemos a crédito, esto hace que al generar una factura se generen varios registros relacionados entre sí para poder ser consultados.

En la siguiente sección veremos algunas de las diferentes operaciones y como se relacionan los diferentes documentos de Proscái.

Ejemplo 1: Movimientos de CXC

Posiblemente la estructura más sencilla de una transacción de Proscail.

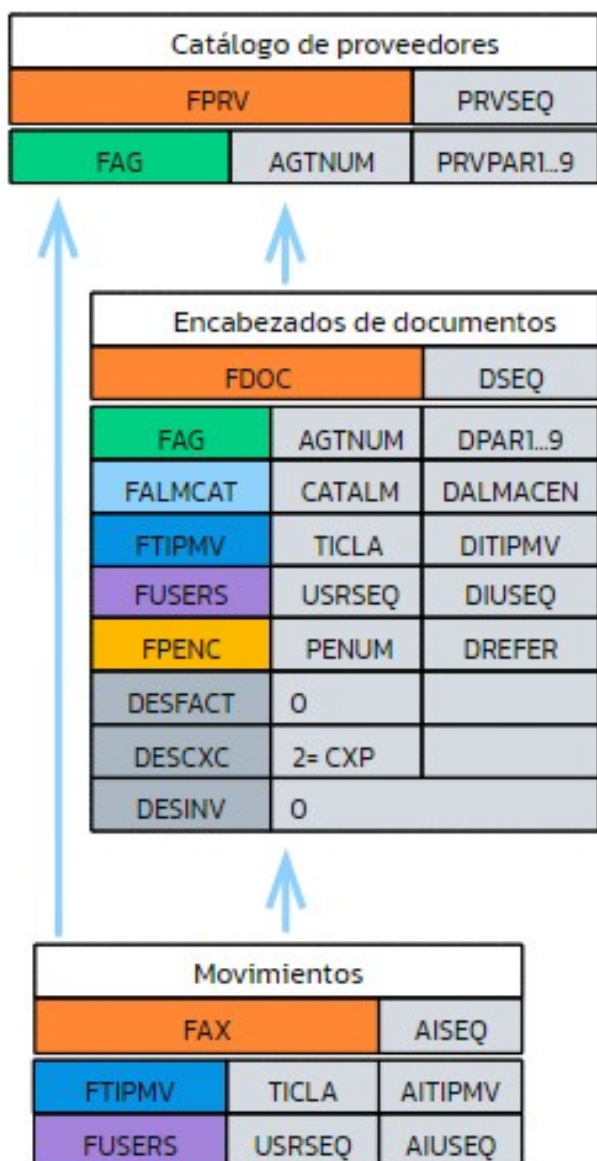
La jerarquía de las tablas nos indica que un cliente puede tener varios documentos **y un documento varios movimientos**, imaginemos un cliente nos debe un documento que originalmente valía \$1000 y tenemos un pago parcial de \$800, esto genera en nuestro documento dos movimientos uno por \$1000 que es el cargo original y uno por (\$800) que es el abono del pago, esto hace que nuestro documento tenga dos movimientos en la tabla de movimientos FAX. Como se muestra en el siguiente diagrama:



Por último, es importante para definir en este ejemplo que los datos bandera que definen el tipo de documento están con valores DESFACT=0 ya que no estamos representando una factura, el campo DESCXC=1 por ser un documento de crédito de cliente, así como el campo DESINV=0 ya que no se trata de un documento que haya afectado el inventario de algún producto

Ejemplo 2: Movimientos de CXP

Ahora vamos a definir la estructura de un documento que represente el adeudo a un proveedor, ósea una cuenta por pagar, la estructura es la siguiente:



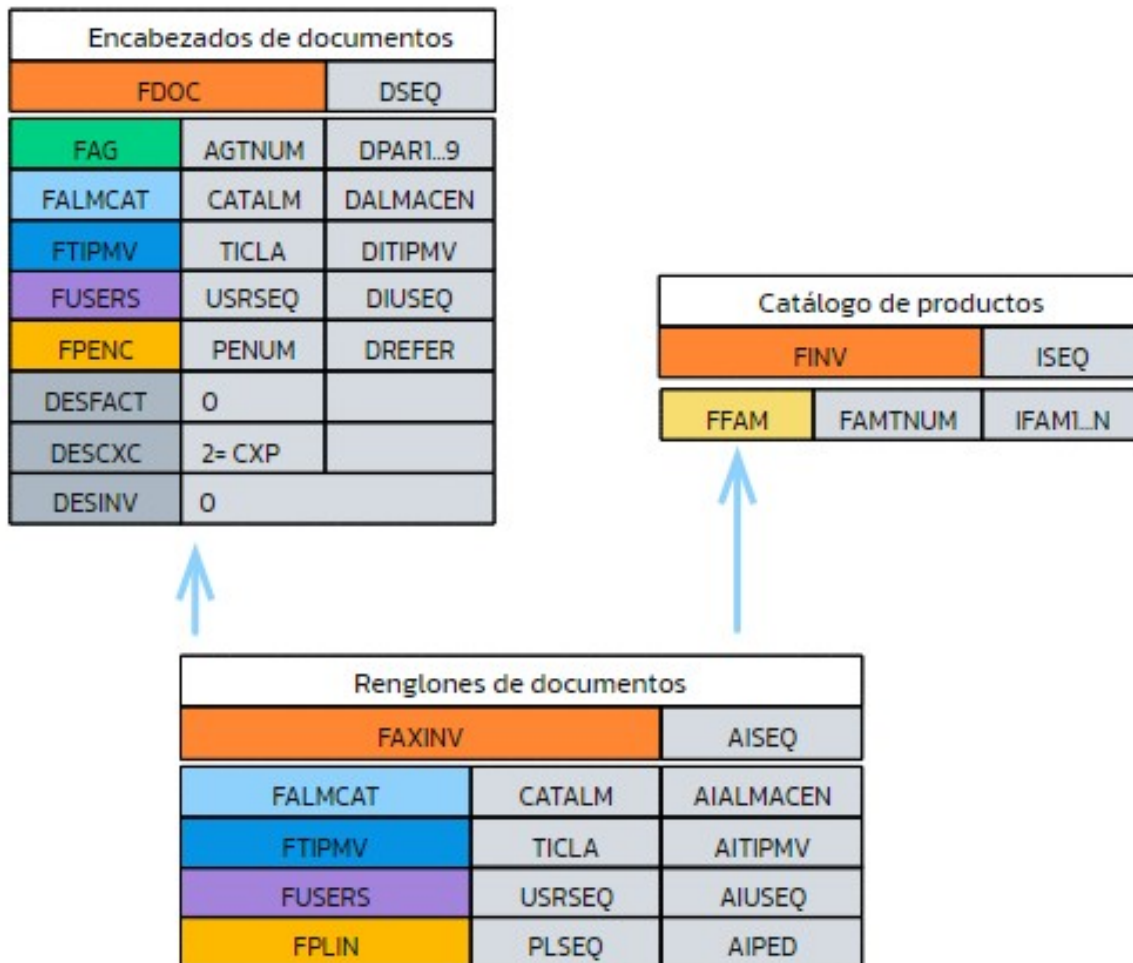
Como se puede ver la estructura es idéntica a la anterior con algunos pequeños cambios, la tabla superior es la de proveedores (FPRV) y la bandera DESCXC=2 para indicar que es una cuenta por pagar.

Este tipo de similitudes en estructuras las veremos en los diferentes documentos, ya que el solo hecho de cambiar una bandera, puede cambiar por completo el contexto de los datos.

Ejemplo 2: Movimientos de INVENTARIOS

Los movimientos de inventarios, requieren de tres tablas para ser representados, un documento que representa la entrada o salida, los renglones de los movimientos en donde se guardan las cantidades del movimiento y una liga a la tabla de productos para saber a qué productos se les están afectando sus existencias.

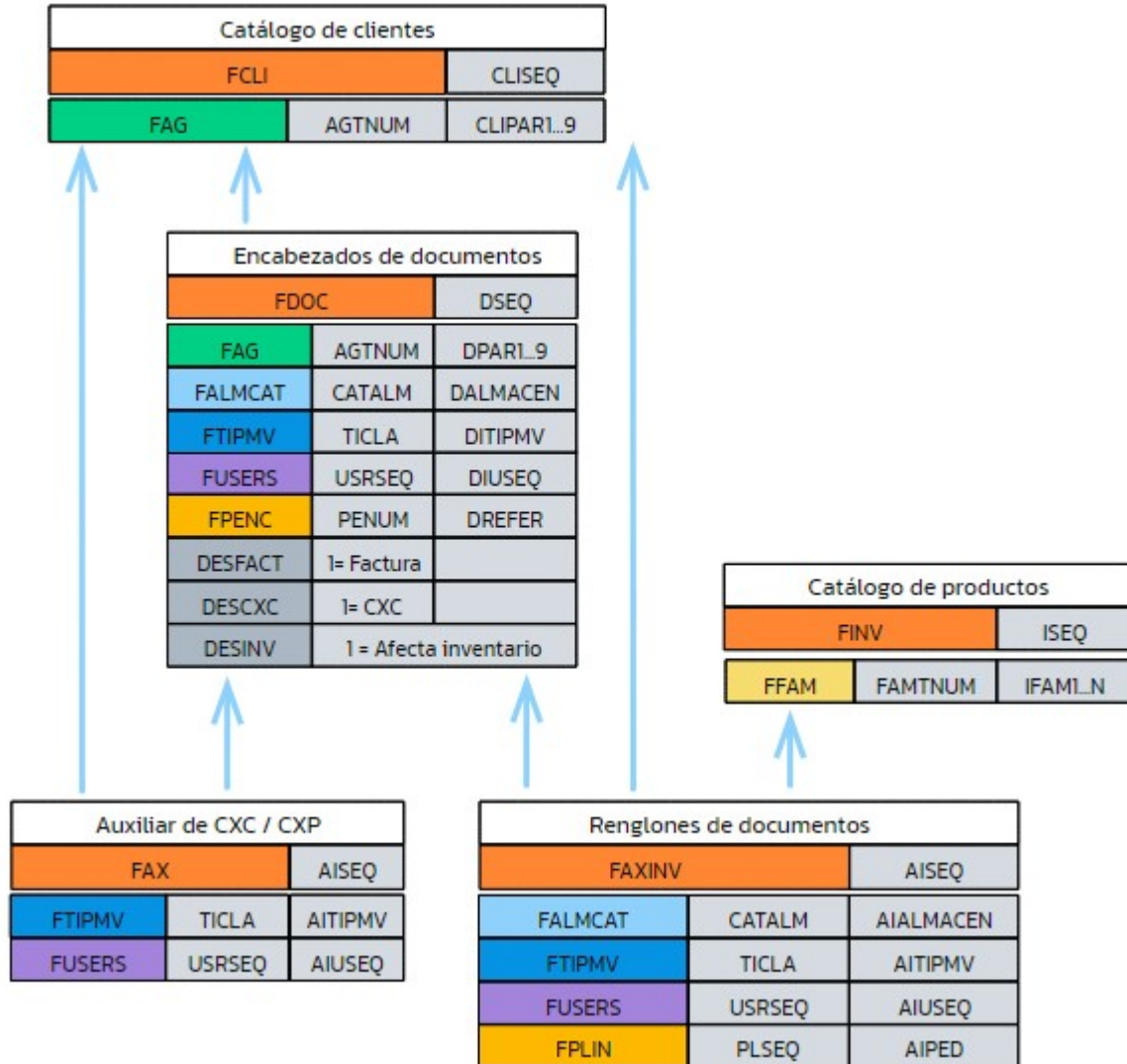
Nótese como en las banderas de la tabla FDOC únicamente esta prendida la bandera DESINV=1 ya que es un documento que está afectando inventarios, pero no representa afectación de cuentas por cobrar ni es una factura por sí mismo.



Ejemplo 3: Facturación

La representación de una factura es uno de los documentos más complejos de Proscái (solo una orden de producción mantiene una estructura más compleja).

En una factura encontramos todas las tablas que se usan en un movimiento de cxc y en un movimiento de inventario, de hecho si lo pensamos una factura es un documento de salida de inventario con una afectación a las cuentas por cobrar, sin embargo parte del diseño elegante de Proscái es que no requerimos de dos registros en la tabla de FDOC sino solo un registro y solamente se afectan las banderas necesarias, en este caso DESFACT=1 ya que es una factura, DESCXC=1 ya que está generando un cargo al cliente y DESINV=1 Para indicar que está haciendo un movimiento de inventario, como lo podemos ver en el diagrama siguiente:



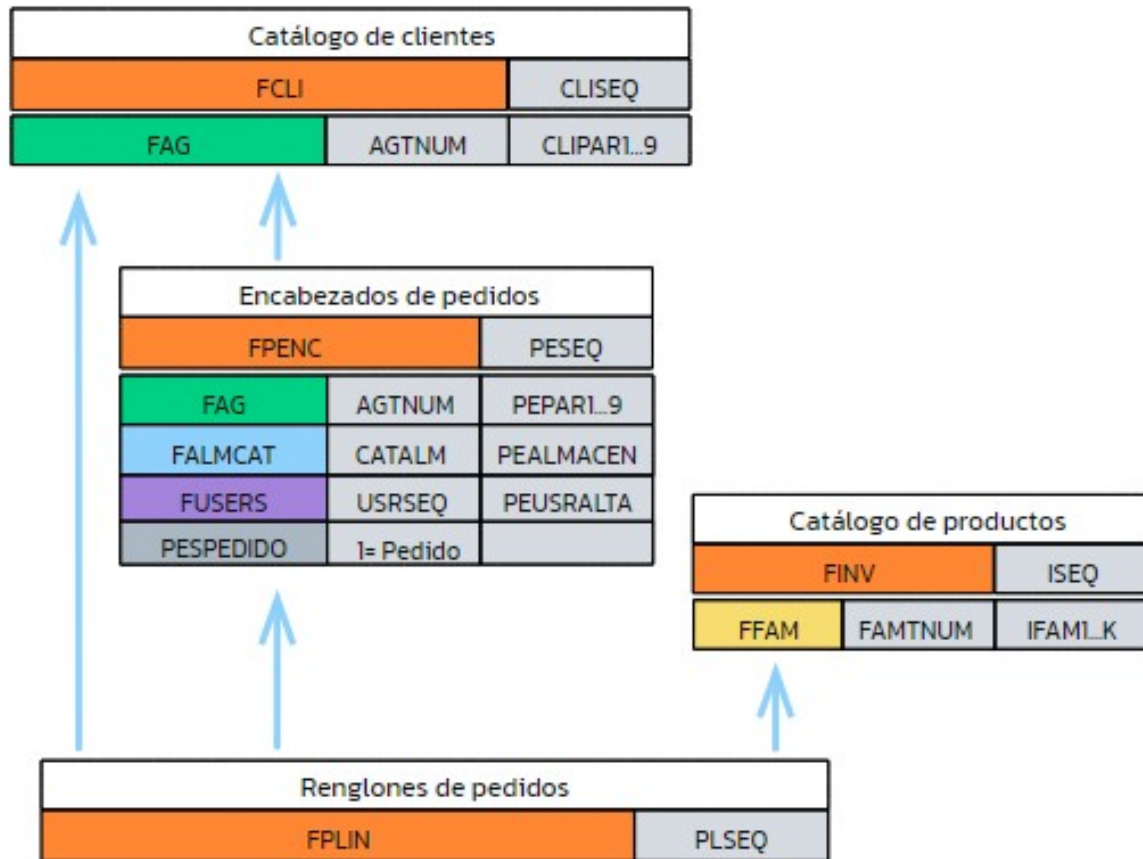
La estructura de una recepción es la misma únicamente cambiando la tabla de clientes (FCLI), por la tabla de proveedores (FPRV) y las banderas a DESFACT=2 DESINV=1 Y DESCXC=2.

Por favor analice los ejemplos anteriores de la estructura de un movimiento de inventarios y la de un movimiento de cxc y compárelos con el de la factura para que se familiarice con la estructura.

Ejemplo 4: Pedidos

La representación de un pedido es similar a la de las facturas pero más sencilla ya que un pedido no afecta cuentas por cobrar ni por pagar, por lo tanto no toca la tabla de movimientos FAX.

Para representar un pedido se requiere de un cliente (FCLI), un documento (el pedido (FPENC)) y los renglones del mismo, como lo vemos en el siguiente diagrama:



Las banderas de PESPEDIDO vale 1 indicando que es un pedido de clientes, cuando un pedido se convierte en cotización esta bandera cambia a 4.

La representación de una orden de compra es similar únicamente cambiando la tabla de clientes por la de proveedores (FPRV) y la bandera PESPEDIDO=2.

Por favor haga una pausa en este momento y vea la similitud en estructura (no en tablas) de una factura y de un pedido y vera que son similares (únicamente quitando la tabla FAX).

Esto es lógico ya que los dos documentos representan un listado de productos ligados a un cliente, la única diferencia es que el pedido es una promesa de compra y una factura es la compra ya realizada por el cliente pero estructuralmente a nivel base de datos son muy similares

Ejemplo 5: Pólizas

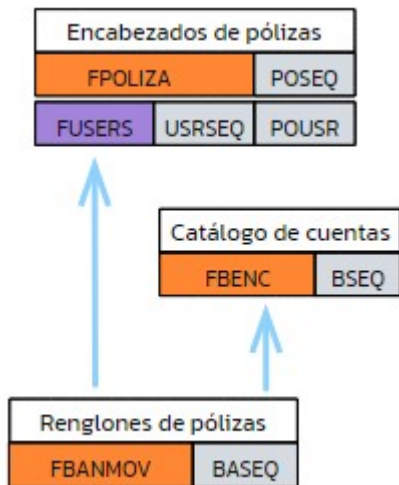
La representación de una póliza consta de tres tablas:

FPOLIZA representando la póliza per-se

FBANMOV representa los asientos o renglones de la misma

FBENC únicamente para conocer las cuentas que han sido afectadas en cada línea de la póliza

El siguiente diagrama nos muestra cómo se ejemplifica una póliza:



Como dato curioso vea la similitud entre esta estructura y la relación entre las tablas FDOC, FAXINV y FINV, también vea la similitud en las tablas de un pedido FPENC, FPLIN y FINV

Esta estructura típica se usa cuando tenemos un documento que contiene varios renglones en el.

Almacenamiento de datos

En esta sección veremos el último tipo de estructura que existe en Proscái y es la que denominamos almacenamiento de datos.

En varias ocasiones se requiere guardar información sin que esta represente un catálogo o un documento complejo como los anteriores que hemos visto, en este caso se usan tablas sin conexiones o tablas conectadas a otras simplemente para guardar información indispensable para Proscái.

Las principales tablas en donde se guarda información en Proscái son:

Almacenes			Lotes		Cajas			Producción en proceso WIP		
FALM	ALMSEQ		FLOTES	LOSEQ	FCAJAS	CAJSEQ	FTIKETS	TKTSEQ		
FALMCAT	CATALM	ALMNUM			FDOC	DNUM	CAJRECEPCION	FINV	ICOD	TKTPRO
					FDOC	DNUM	CAJFACTURA	FINV	ICOD	TKTART

Tabla	Contenido
FALM	Guarda información de los inventarios en cada uno de los almacenes.
FLOTES	Guarda información de las capas de productos cuando se usan capas UEPS/PEPS
FCAJAS	Guarda información de piezas o productos que requieren números de serie o rollos
FTIKETS	FTIKETS que es la tabla que guarda la información de producción en proceso (WIP)

CAPÍTULO 3 EXTRACCIÓN DE DATOS

En este capítulo iniciaremos con la parte más importante del objetivo del libro que es la extracción de información de la base de datos de **Proscái**.

La petición más grande de los usuarios al área de sistemas es la requisición de un informe, normalmente tenemos a los usuarios capturando y generando información y llega el momento de utilizarla para diferentes funciones en la empresa.

Es importante recordar que el objetivo de este libro no es el cómo se guarda la información sino como se extrae. Con el conocimiento que se desarrollara en el trascurso como se vaya leyendo.

Al consultar y extraer datos se estará ayudando a la empresa a explotar la información que se encuentra en ella y seguramente a descubrir muchísimas cosas de gran valor que en los datos se encuentran, y tener la facilidad de en pocos segundos saber quién es el mejor cliente, qué productos compra, con qué frecuencia nos visita, qué vendedor lo atiende o cuantas piezas de un producto vendimos en un periodo, quién es el proveedor que nos las vende, qué existencia y pedidos tenemos del mismo.

Los ejercicios que empezaremos a realizar y que a continuación describiremos, están creados de forma progresiva permitiendo ir desde la forma más simple en la extracción de información y gradualmente incorporar los comandos que mencionamos en el capítulo anterior y algunas otras funciones que SQL permite, con el fin de poder validar y dominar el uso de cada comando.

El objetivo es que tú mismo puedas realizar cada ejercicio y validar su funcionamiento, además de ir familiarizándote con los términos y forma de crear las consultas.

Para la realización de dichos ejercicios, es necesario contar la base de datos denominada '**Ejercicios_SQL**', esta base está especialmente preparada para ese efecto, si no cuentas con ella, contacta a nuestra área de soporte para que te la proporcionen.

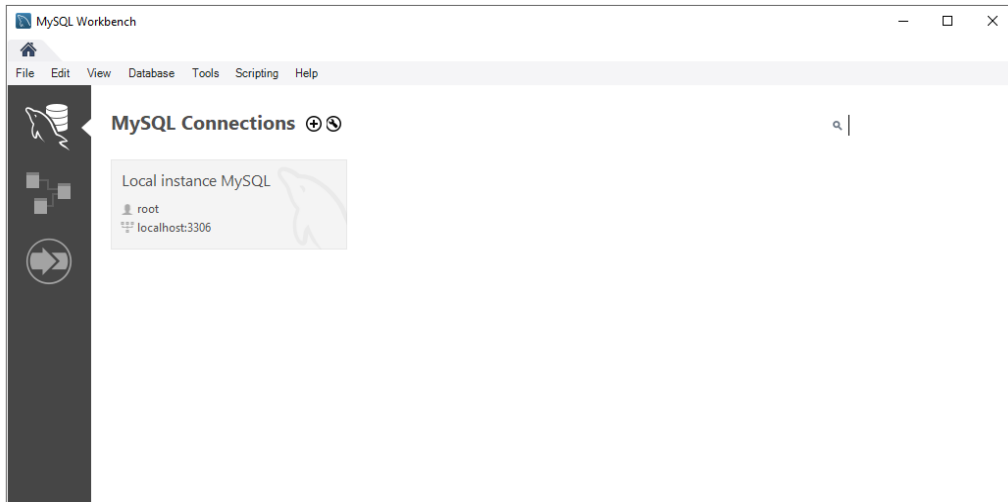
Nunca debes ejecutar estos ejercicios o cualquier otra consulta sobre tu base real operativa, ya que por la cantidad de información que puede contener, se puede afectar el funcionamiento operativo de los demás usuarios, si alguno de los ejercicios quieres usarlo sobre tu información real, es recomendable hacerlo sobre una base de respaldo.


Herramienta y mecanismo para la extracción de datos.

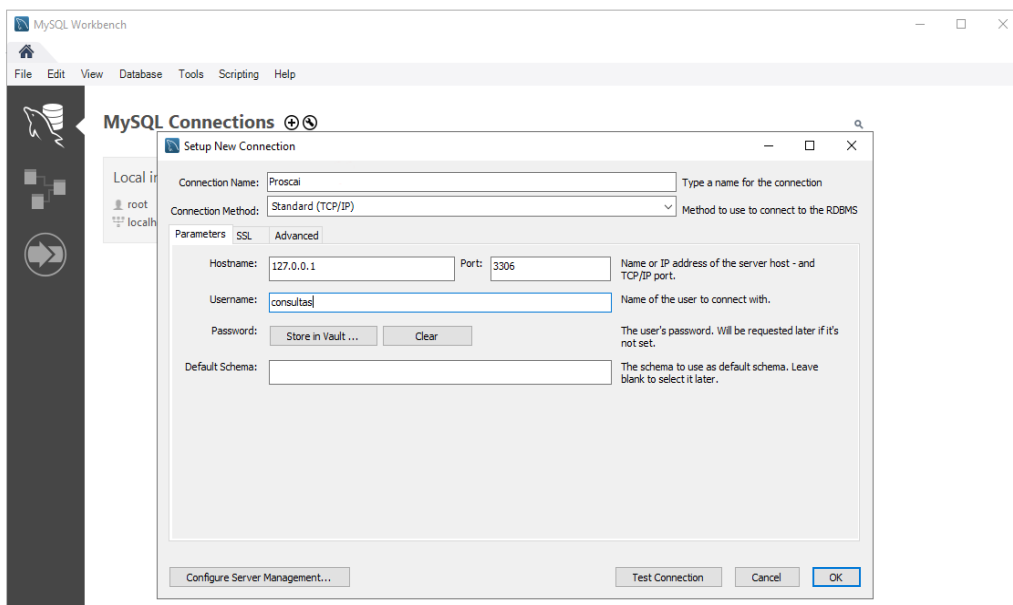
La herramienta por utilizar para la creación y ejecución de las consultas y que debes tener instalada es MySQL Workbench.

Existen dos escenarios de configuración: bajo servidores propios y conexión remota a Galio.

1. MySQL Workbench configuración local:




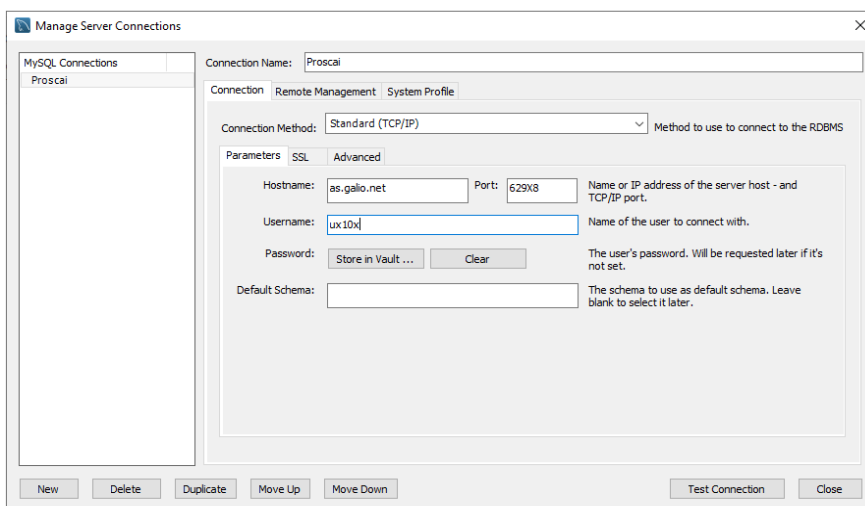
Para definir los datos de configuración debes dar clic sobre el icono  y registrar los datos que corresponden a Hostname que será la dirección IP de tu servidor de datos; así como el puerto de conexión que por default aparece típicamente, en caso de que hayas cambiado el puerto, deberás indicarlo; el Usuario será consultas y dando clic al botón Store in Vault deberás definir la contraseña, que típicamente es consultas.



2. MySQL Workbench configuración Galio:



Para definir los datos de configuración debes dar clic sobre el icono  y registrar los datos que deberás solicitar a Galio y corresponden a Hostname que será la dirección IP remota del servidor de datos; así como el puerto de conexión que ellos señalen; el Usuario será el definido por ellos y dando clic al botón Store in Vault deberás definir la contraseña que ellos te proporcionen.



Una vez que hayas configurado MySQL Workbench podrás continuar con los siguientes pasos.

SQL COMANDOS E INSTRUCCIONES BÁSICAS

Comandos básicos de una sentencia de consulta

Antes de consultar y obtener datos, conozcamos y expliquemos las instrucciones y funciones básicas que nos permitirán crear sentencias; Como mencionamos capítulos atrás, las sentencias que se utilizan en el lenguaje SQL resultan ser fácilmente comprensibles acorde a la función que realizan en la obtención de información.

Empecemos conociendo los comandos que conforman la estructura básica de una consulta y que se denomina Query.

La sentencia necesita de 3 comandos base:

SELECT

Es el primer comando forzoso dentro de la instrucción básica que se utiliza para obtener la información de la base de datos.

Con **SELECT** podemos seleccionar una o varias columnas o campos específicos de datos.

El resultado de una consulta **SELECT** nos devuelve una tabla lógica. Es decir, los resultados son una relación de datos, que tiene filas/registros, con una serie de campos/columnas. Igual que cualquier tabla de la base de datos. Sin embargo esta tabla está en memoria mientras la utilizemos, y luego se descarta. Cada vez que ejecutamos la consulta se vuelve a calcular el resultado.

En pocas palabras es el comando utilizado para traer información desde una tabla

Existen muchas cláusulas asociadas a la sentencia **SELECT** (GROUP BY, ORDER, **HAVING**) y que servirán para modificar los resultados obtenidos de una consulta, más adelante los usaremos y observaremos su funcionamiento.

FROM

Este es el segundo comando forzoso al crear una consulta y sirve para establecer la fuente de datos. Se utiliza para seleccionar una o varias tablas que almacenan las columnas previamente indicadas.

WHERE

Es el tercer comando básico en la estructura y es lo que conocemos como filtro, sirve para condicionar las consultas, para establecer filtros o realizar consultas condicionantes.

***Normalmente se sugiere cambiar de renglón con cada palabra clave que se utilice para darle estructura a la sentencia y hacerla más fácil de leer.**

La sentencia completa quedaría de esta forma:

```
SELECT campo1,campo2
FROM 'X_tabla'
WHERE campo1='X';
```

Observemos que los 3 comandos básicos se encuentran en un orden lógico y estructurado y fácilmente se puede entender su funcionamiento.

A partir de esta simple estructura agregaremos más comandos que nos permitirán ir acrecentando el nivel de conocimiento y sofisticando el grado de consultas.

Resumiendo lo anteriormente explicado, veamos cómo quedaría un query simple:

```
SELECT clicod,clinom  
FROM FCLI  
WHERE clicod='00011'
```

Con la palabra **SELECT** indicamos a SQL que vamos a extraer información, enseguida se declaran los campos deseados a obtener, en este caso clicod,clinom cada campo se separa con una coma y con la palabra **FROM** le indicamos de que tabla queremos extraerla en nuestro caso FCLI que es la tabla de clientes y con el comando **WHERE** condicionamos el código del cliente que queremos extraer. Este query extraerá el código y nombre de la base del cliente indicado.

Notas de sintaxis:

SQL no es sensible a las mayúsculas y minúsculas, lo que quiere decir que se pueden utilizar indistintamente, así mismo se pueden escribir uno o varios renglones sin alterar su resultado por ejemplo:

```
SELECT clicod,clinom  
FROM fcli
```

O de igual forma puede ser:

```
select CLICOD,clinom  
FROM FCLI
```

Es importante recordar que cada dato o campo a solicitar, debe ir separado por comas.

Si queremos ver todos los campos de la tabla será necesario indicarlo con un '*' ejemplo:

```
SELECT *  
FROM FCLI
```

Esta sentencia nos mostrará todos los campos de la tabla FCLI o de la tabla que deseemos, pero **no es recomendable su uso**, ya que si se solicitan todos los campos de alguna tabla que puede contener demasiados registros existe el riesgo de volcar el servidor y afectar el trabajo operativo de los demás usuarios.

Comandos Adicionales

A continuación explicaremos una serie de comandos adicionales y su función.

AND

Para realizar consultas con más de una condición a cumplir.

OPERADORES

Se puede hacer uso de diferentes operadores para filtrar o condicionar la consulta

Algunos operadores;

- <>, Diferente
- <, menor que
- <=, menor que o igual
- >, mayor que
- >=, mayor que o igual

LEFT JOIN

La función **LEFT JOIN**, se utiliza en caso de que exista más de una tabla, es decir cuando se requiere realizar una 'conexión' o una "consulta combinada" en la cual se busca en la tabla superior la liga con la tabla principal

GROUP BY

En esta cláusula se establece los grupos de datos por los que se quiere obtener totales, es decir como queremos agrupar los datos

ORDER BY

Con **ORDER BY** se puede ordenar el resultado por diferentes columnas, y en sentido ascendente o descendente, antes de ser presentado.

LIMIT

Esta función siempre la deberemos utilizar para restringir el número de datos a obtener, es decir limitar el número de filas (registros/resultados) devueltas en las consultas **SELECT**. **Es importante su uso para evitar desbordamientos o caídas del servidor al procesar tantas líneas de información.**

Ejemplo:

```
SELECT CLICOD,CLINOM,CLITEL ,CLICD,CLISACT  
FROM FCLI  
LIMIT 10
```

La consulta se limitará a presentar únicamente los 10 primeros registros de la tabla.

CLICOD	CLINOM	CLITEL	CLICD	CLISACT
01	CENTRO	(55)-24776658	D.F.	-181.70
05	SATELITE	(55)-96774775	D.F.	0.00
00121	EMMA GABRIELA RODRIGUEZ COLORADO	(55)-41 3253710	CHIMALHUACÁN	1041.70
00339	MARIA CAROLINA SANCHEZ GERARDO	(55)-44935272	D.F.	0.00
00509	GABRIELA LEONOR ESCOBAR BADILLO	(55)-52653231	D.F.	0.00
01783	YANETH MACEDA MACATIA	(55)-76383445	D.F.	0.00
07868	SANDY VAZQUEZ ELIZALDE	(55)-76648772	CIUDAD NEZAHUALCOYOTL	0.00
08585	RUTH ROCHA GONZALEZ	(55)-72385966	D.F.	0.00
37167	ISIDRO CAMPUZANO ANGELES	(55)-22524133	D.F.	666.59
03	INSURGENTES	(55)-68187324	D.F.	0.00

HAVING

La cláusula **HAVING** sirve para complementar el condicionante WHERE, ya que con WHERE no se permite utilizar funciones de agregación como SUM, MAX, MIN o AVG, es decir, con un condicionante WHERE no podemos crear consultas válidas que sean capaz de devolvernos los datos donde el condicionante sea un cálculo aritmético, por ejemplo si requerimos obtener los datos de los clientes que compraron más de 1000 productos durante un año.

Resumen creando sentencias

Ahora consideremos que para construir una consulta SQL es necesario hacernos como mínimo tres preguntas:

1. **¿qué datos nos están pidiendo?**
Por ejemplo: el Código y Nombre de los clientes
2. **¿dónde están esos datos?**
Por ejemplo: están en la tabla Clientes.
3. **¿qué condición deben cumplir los registros?**
Por ejemplo: quienes tienen un saldo mayor o igual a 4000 pesos

Sintaxis SQL	Para Proscái
<code>select CAMPOS(separados por comas) from TABLA where CONDICIÓN</code>	<code>select CLICOD,CLINOM from FCLI where CLISACT >=4000</code>

Si requieres conocer características más detalladas de los campos de cualquiera de las tablas, en SQL existe el comando "**Describe**", el cual nos permite obtener una lista de cualquier tabla y todos sus campos, así como un detalle completo sobre las características de cada uno.

Es así, que una vez que conocemos la base de datos de **Proscái**, sus tablas y campos, y revisamos algunos conceptos básicos, vamos a empezar viendo las consultas simples, basadas en una sola tabla. Veremos cómo obtener filas y columnas de una tabla en el orden en que nos haga falta.

Lo que veremos son apenas algunas ideas de lo que se puede hacer con la información que esta guardada en la base de datos de **Proscái**.

Siguiendo nuestros ejemplos gráficos, vamos a ver como se visualiza un query básico con nuestra tabla:

Select campo1,campo2

From

FCLI

Where campo1=x

Catálogo de clientes	
FCLI	CLISEQ

Recordemos que la tabla FCLI almacena todos los clientes de la base de datos.

Modos de conexión:

A continuación, describiremos de forma teórica las formas de conexión entre las diferentes tablas.

Hay estructuras de Proscái que requieren de la información de varias tablas, por ejemplo una factura, en la siguiente sección veremos cómo relacionar información de varias tablas para generar queries con información de dos o más tablas.

Por favor, no trate de realizar ninguno de los siguientes queries, solo los usaremos como parte básica para describir y comprender la relación y conexión entre las tablas.

Más adelante en la sección de ejercicios prácticos, encontrará muchos queries que podrá realizar con su base de datos.

En capítulos anteriores hemos visto que existen dos tipos de conexiones entre las tablas de Proscái, las directas y las indirectas y dentro de las indirectas están, las que se conectan a un catálogo y las que se conectan a otras tablas que no son necesariamente catálogos.

En cualquier tipo de conexión existen dos tablas y un conector:

Para poder hacer esta conexión en nuestra sentencia de SQL debemos de incluir la notación correcta para que se haga esta conexión.

La forma general de un select en donde se unen dos tablas es:

```
Select campo1,campo2,campo3....  
From TABLA2  
Left join TABLA1 ON TABLA1.CAMPOC2=TABLA2.CAMPOC2  
WHERE CAMPO1=XXX AND CAMPO4=YYY
```

Vamos a explicar paso a paso la conformación del query de ejemplo anterior:

Las instrucciones select y from son las mismas que hemos mencionado anteriormente y son parte de los comandos básicos para armar un query. La única diferencia es que en los campos a seleccionar pueden ser campos de cualquiera de las dos tablas y la instrucción from debe de ser de la tabla de más BAJA jerarquía de las dos tablas a relacionar.

La instrucción "left join" que nos servirá para realizar la conexión, la vamos a partir en tres partes para comprender su funcionamiento:

Parte:	
Left join	Instrucción o comando
TABLA2	Tabla a conectar, LA DE MAS ALTA JERARQUIA
TABLA1.CAMPOC1=TABLA2.CAMPOC2	conexión

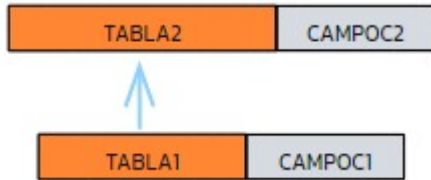
En donde:

TABLA1 es la tabla de jerarquía MAS BAJA de nuestra conexión (de donde SALE la flecha) a esta tabla la llamamos MAIN FILE

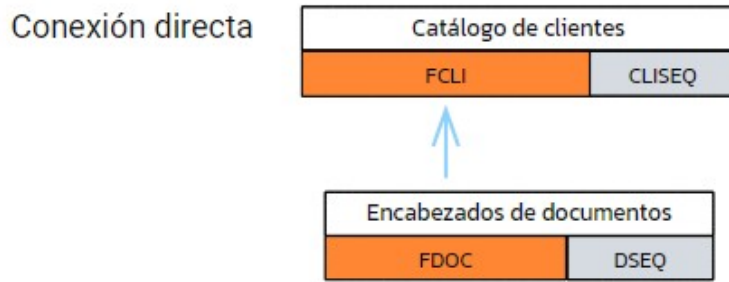
TABLA2 es la tabla de MAYOR jerarquía en la conexión (A donde LLEGA la flecha)

CAMPOC2 es el campo de la TABLA2 a donde se va a hacer la conexión

CAMPOC1 es el campo de la TABLA1 de donde se hace la conexión

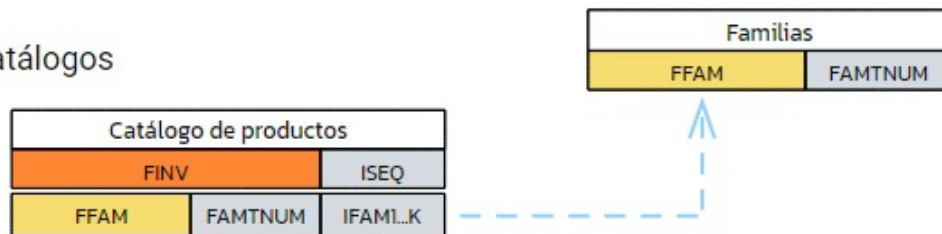


Compara los diagramas y podrás notar que sin importar el tipo de conexión, la lógica es la misma:

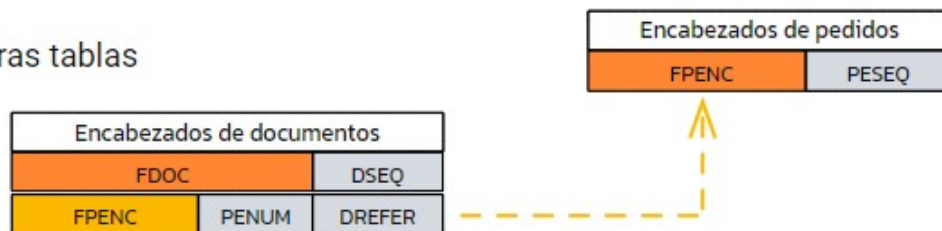


Conexiones indirectas

A Catálogos

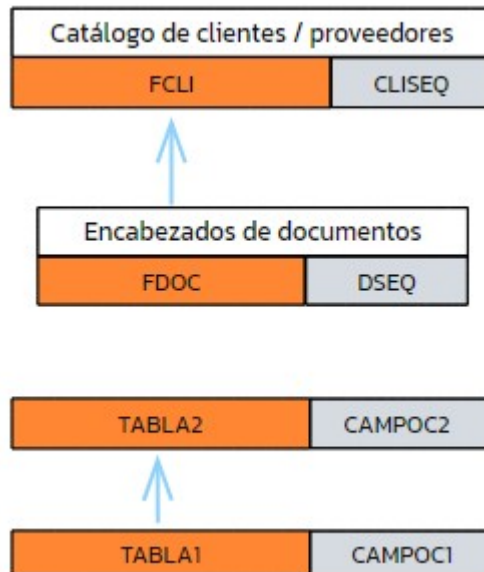


A otras tablas



Con la información anterior, veamos cómo se realizarán las conexiones a nuestros tres tipos de conexiones y como quedarían los queries:

Conexión directa



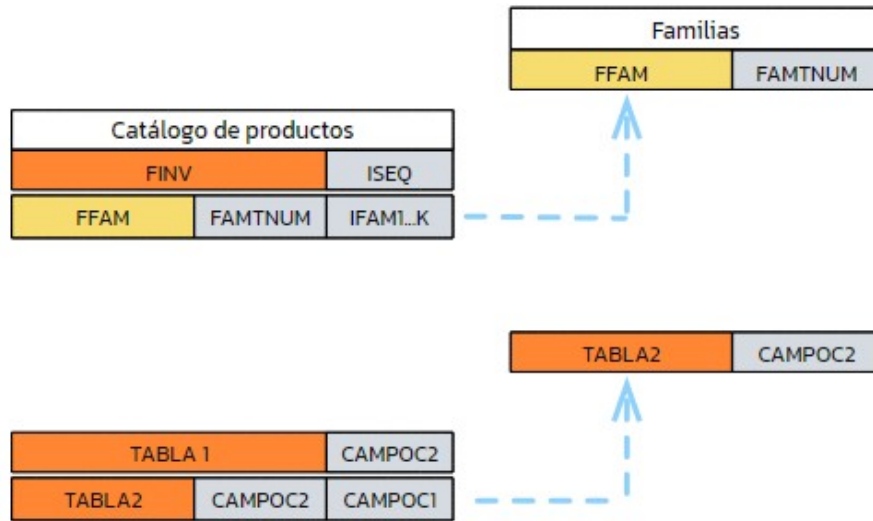
```
Select CLICOD,CLINOM,DNUM  
From FDOC  
Left join FCLI on FCLI.CLISEQ=FDOC.CLISEQ
```

En este tipo de conexión, los campos de liga CAMPOC1 Y CAMPOC2 (equivalente a CLISEQ) son los mismos y deberá de usarse el campo de secuencia de la TABLA2 en este caso CAMPOC2

```
Select campo1,campo2,campo3....  
From TABLA1  
Left join TABLA2 ON TABLA2.CAMPOC2=TABLA1.CAMPOC1
```

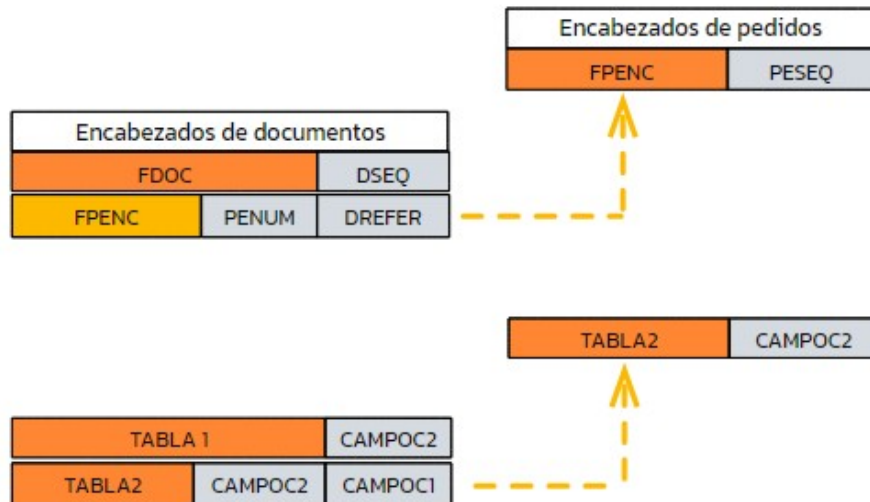
Conexiones indirectas

A catálogos:



```
Select ICOD,IDESCR,FAMDESCR  
From FINV  
Left join FFAM on FINV.IFAM1=FFAM.FAMTNUM
```

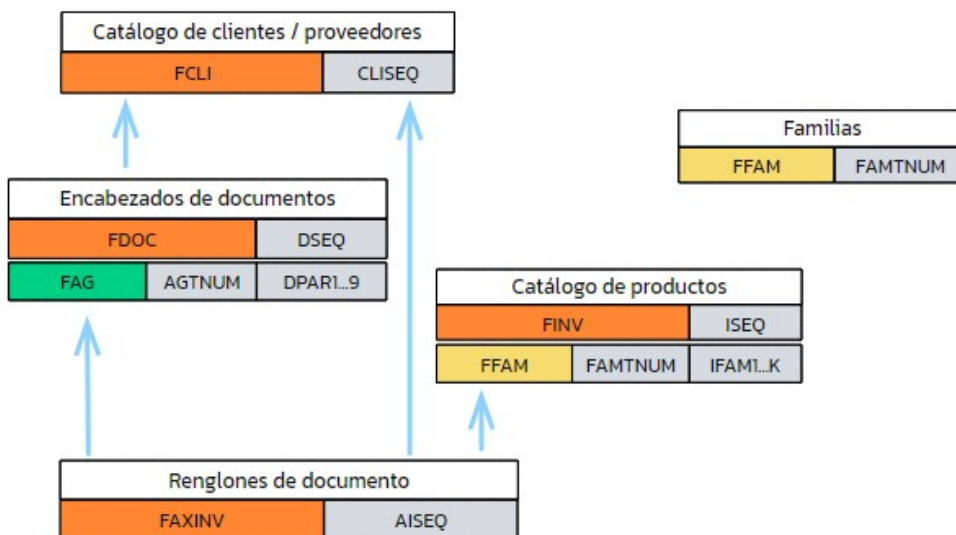
A otras tablas:



```
Select DNUM,PENUM  
From FDOC  
Left join FPENC on FPENC.PENUM=FDOC.DREFER
```

Como podemos ver, en los tres ejemplos las instrucciones son similares y lo único que cambia son los nombres de los campos de conexión (CAMPOC1 y CAMPOC2), esta es la razón por la cual incluimos los tipos de conexión y los campos en el diagrama principal de Proscail ya que con esta información será muy fácil realizar cualquier conexión entre las diferentes tablas.

Como hemos visto anteriormente, una tabla puede estar conectada a varias tablas de jerarquía superior, en el siguiente diagrama se muestra la conexión de la tabla FAXINV a las tablas FDOC, FINV y FCLI.



A continuación mostraremos como hacer la relación para ligar varias tablas, comencemos como base con el query que liga las tablas FAXINV y FINV.

```
Select ICOD,IDESCR,AICANTF
From FAXINV
Left join FINV on FINV.ISEQ=FAXINV.ISEQ
WHERE XXXX
```

Es importante recordar que siempre la tabla donde se declara en el comando FROM... debe de ser la tabla de menor jerarquía, en este caso FAXINV.

Podemos ahora agregar datos de la tabla FCLI agregando el comando LEFT JOIN correspondiente

```
Select ICOD,IDESCR,AICANTF
From FAXINV
Left join FINV on FINV.ISEQ=FAXINV.ISEQ
Left join FDOC on FDOC.DSEQ=FAXINV.DSEQ
Left join FCLI on FCLI.CLISEQ=FAXINV.CLISEQ
WHERE XXXX
```

En este caso hemos hecho la conexión hacia la tabla FCLI directo de la tabla FAXINV, pero si observamos la tabla FCLI también está ligada a la tabla FDOC y como ya hemos hecho previamente la conexión a la tabla FDOC con la instrucción:

```
Left join FDOC on FDOC.DSEQ=FAXINV.DSEQ
```

Podríamos hacer la conexión a la tabla FCLI vía la tabla FDOC, entonces nuestro query quedaría de la siguiente manera:

```
Select ICOD,IDESCR,AICANTF
From FAXINV
Left join FINV on FINV.ISEQ=FAXINV.ISEQ
Left join FDOC on FDOC.DSEQ=FAXINV.DSEQ
Left join FCLI on FCLI.CLISEQ=FDOC.CLISEQ
WHERE XXXX
```

Cuando queramos hacer una conexión de esta forma es importante que el orden de las instrucciones LEFT JOIN, sea de las jerarquías menores a las mayores, en este caso sería un error el siguiente query:

```
Select ICOD,IDESCR,AICANTF
From FAXINV
Left join FINV on FINV.ISEQ=FAXINV.ISEQ
Left join FCLI on FCLI.CLISEQ=FDOC.CLISEQ
Left join FDOC on FDOC.DSEQ=FAXINV.DSEQ
WHERE XXXX
```

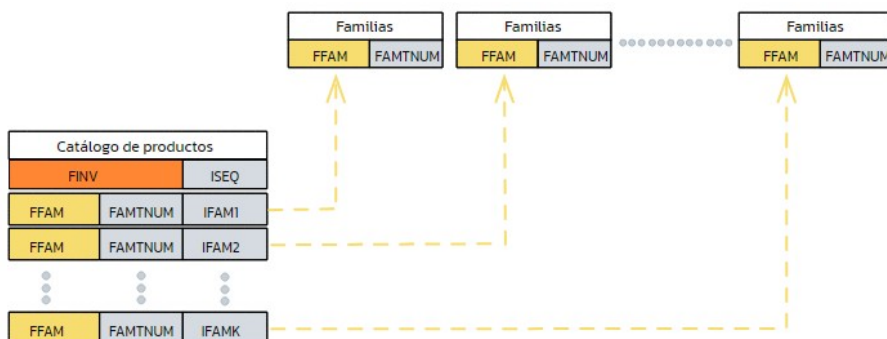
Ahora hagamos un query ligando las tablas FAXINV, FINV y ligar la tabla indirecta FFAM, nuestro query quedaría de la siguiente forma:

```
Select ICOD,IDESCR,FAMDESCR
From FAXINV
Left join FINV on FINV.ISEQ=FAXINV.ISEQ
LEFT JOIN FFAM ON FFAM.FAMTNUM=FINV.IFAM1
WHERE XXXX
```

Con esto podemos ver que se pueden hacer conexiones simultáneas a tablas directas e indirectas simplemente respetando las reglas de conexión.

Conexiones a la misma tabla

En ocasiones es necesario hacer dos conexiones a una misma tabla, esto sucede en diferentes tipos de conexiones.



Nuestro diagrama nos muestra como la tabla FINV utilizando los diferentes campos de familia, podría ligarse a la tabla FFAM.

Usemos como ejemplo el caso de la tabla FINV con sus múltiples conexiones a la tabla FFAM, pensemos que tenemos en nuestro catálogo de productos dos familias: Departamento y Temporada en las familias 2 y 4 respectivamente, el query necesario para hacer esta doble conexión sería:

```
Select ICOD,IDESCR,FAM2.FAMDESCR,FAM4.FAMDESCR
From FINV
Left join FFAM AS FAM2 ON FAM2.FAMTNUM=IFAM.IFAM2
Left join FFAM AS FAM4 ON FAM4.FAMTNUM=IFAM.IFAM4
```

En este caso, estamos generando 2 conexiones a la misma tabla, para poder distinguir en las conexiones una conexión de otra, usamos el atributo 'AS' indicando que le estamos dando un nombre diferente a cada tabla para poder referir a cada una de ellas de forma independiente, note como el 'ALIAS' lo usamos tanto en el nombre de los campos como en las conexiones.

Usaremos la misma regla cuando queramos conectar dos veces a la misma tabla en conexiones indirectas que no sean catálogos, este caso se da en las tablas FENS, FCAJAS y FTIKETS, donde cada una de ellas tiene dos conexiones a las tablas FINV y FDOC respectivamente.

En este caso la regla de los querys serían las siguientes:

FENS

Ensamblajes		
FENS		ENSSEQ
FINV	ICOD	EPRO
FINV	ICOD	EART

```
Select PRODUCTO.IDESCR,MATERIAL, IDESCR,ENSCANT
From FENS
Left join FINV AS PRODUCTO ON PRODUCTO.ICOD=FENS.EPRO
Left join FINV AS MATERIAL ON MATERIAL.ICOD=FENS.EART
```

FCAJAS

Cajas		
FCAJAS		CAJSEQ
FDOC	DNUM	CAJRECEPCION
FDOC	DNUM	CAJFACTURA

```
Select RECEPCION.DNUM,FACTURA.DNUM,CANJUM
From FCAJAS
Left join FDOC AS RECEPCION ON RECEPCION.DNUM=FCAJAS.CAJFACTURA
Left join FDOC AS FACTURA ON FACTURA.DNUM=FCAJAS.CAJRECEPCION
```

FTIKETS

Producción en proceso WIP		
FTIKETS		TKTSEQ
FINV	ICOD	TKTPRO
FINV	ICOD	TKTART

```
Select PRODUCTO.IDESCR,PROCESO.IDESCR,TATCANT,TKTSURT
From FENS
Left join FINV AS PRODUCTO ON PRODUCTO.ICOD=FTIKETS.TKTPRO
Left join FINV AS PROCESO ON PROCESO.ICOD=FTIKETS.TKTART
```

Ejercicios Prácticos

A partir de este punto, encontraras ejercicios que podrás realizar en tu base de datos, la dinámica de los ejercicios está diseñada en forma ascendente para ayudarte a desarrollar los ejercicios agregando comandos y funciones de forma gradual, así como crear combinaciones en el manejo de tablas, estoy seguro que al analizar y practicar cada uno de los siguientes ejercicios podrás desarrollar tus habilidades en la creación de consultas.

Consultas básicas sin condiciones

El primer ejercicio que vamos a realizar es usando una sola tabla y sin condiciones que delimiten el resultado.

Select básico inventarios

```
SELECT ICOD,IDESCR,ISTKACT  
FROM FINV
```

Esta sentencia nos mostrará los campos: código de producto, la descripción y el stock actual y la imagen siguiente nos muestra el resultado.

ICOD	IDESCR	ISTKACT
FGF0012	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	0.000
FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	42.000
FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	54.000
FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	46.000
FGF0012VARXGR	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	43.000
FJB0307AZU070	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000
FJB0307AZU090	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000
FJB0307AZU110	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000
FJB0307AZU130	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000
FJB0307AZU150	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000
FJB0311BLA300	PANTALON CABALLERO ARENA GABARDINA -MOD P...	18.000
FJB0311BLA330	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000
FJB0311BLA360	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000
FJB0311BLA400	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000
FJB0311CAF320	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000
FJB0311CAF340	PANTALON CABALLERO ARENA GABARDINA -MOD P...	8.000
FJB0311CAF380	PANTALON CABALLERO ARENA GABARDINA -MOD P...	8.000
GFJ0189BEI010	PLAYERA CABALLERO ARENA SPANDEX -MOD PEGA...	0.000
GIC0015BEICHI	CONTROL DAMA NUDE SPANDEX -MOD 801	19.000

Observa que el titulo o encabezado de cada columna, muestra el nombre del campo solicitado y cada renglón es un registro existente en la base de datos.

Select básico proveedores

```
SELECT PRVCOD,PRVNOM,PRVCOLONIA,PRVCD,PRVEDO  
FROM FPRV
```

Esta sentencia nos mostrará los campos: código de proveedor, nombre, colonia, ciudad y estado, es decir datos básicos de cada proveedor y observamos el resultado.

PRVCOD	PRVNOM	PRVCOLONIA	PRVCD	PRVEDO
AAE	NIKOLA TESLA	JAIME NUNO	LEON DE LOS ALDAMA (LEON)	GTO.
AAF	FEDERICO FELLINI			
AAH	KATE GOSSELIN	AGUA AZUL	GUADALAJARA	JAL.
AAI	JON GOSSELIN	TOLIMAN		JAL.
AAJ	ROBERT MORSE	SANTA MARIA DE CEMENTOS	LEON DE LOS ALDAMA (LEON)	GTO.
ABA	GEORGE GERSHWIN	LOS LIMONES	LEON DE LOS ALDAMA (LEON)	GTO.
ABB	LESLEY STAHL			
ABC	PAUL ROBESON	LA CAMPIÑA	LEON DE LOS ALDAMA (LEON)	GTO.
ABH	MACDONALD CAREY			
ABI	ROBERT CUMMINGS			
ABJ	MARGARET THATCHER	SAN MIGUEL	LEON DE LOS ALDAMA (LEON)	GTO.
ACB	FRANK ABAGNALE			
ACD	NAPOLEON BONEPART			
ACE	WILLIAM FAULKNER	SAN FRANCISCO DEL RINCÓN CENTRO	SAN FRANCISCO DEL RINCÓN	GTO.
ACG	SHAWN MICHAELS	ISSFAM		D.F.
ACH	CHARLES DARWIN	EL PAISAJE	LEON DE LOS ALDAMA (LEON)	GTO.
ACI	SHARON OSBOURNE			
ACJ	STEVEN SPIELBURG	SALAS	PURÍSIMA DE BUSTOS	GTO.
ADA	SHOLEM ALEICHEM	LOS LIMONES	LEON DE LOS ALDAMA (LEON)	GTO.
ADB	WOODROW WILSON	VILLANUEVA	LEON DE LOS ALDAMA (LEON)	GTO.

En este resultado observamos que algunos campos se muestran vacíos, significa que el proveedor respectivo, carece de esa información en el campo indicado.

Usando where para condicionar una consulta.

Ahora, sabiendo cómo hacer una consulta básica, agregaremos el comando **WHERE**, que permite añadir una condición como filtro y limitar el resultado a obtener.

Select clientes sin saldo pendiente

```
SELECT CLICOD,CLINOM,CLITEL ,CLICD,CLISACT  
FROM FCLI  
WHERE CLISACT=0
```

Esta sentencia nos mostrará los campos de: código de cliente, nombre, teléfono, ciudad y saldo actual, que además este último deberá cumplir la condición de estar en cero.

CLICOD	CLINOM	CLITEL	CLICD	CLISACT
05	SATELITE	(55)-96774775	D.F.	0.00
00339	MARIA CAROLINA SANCHEZ GERARDO	(55)-44935272	D.F.	0.00
00509	GABRIELA LEONOR ESCOBAR BADILLO	(55)-52653231	D.F.	0.00
01783	YANETH MACEDA MACATIA	(55)-76383445	D.F.	0.00
07868	SANDY VAZQUEZ ELIZALDE	(55)-76648772	CIUDAD NEZAHUALCOYOTL	0.00
08585	RUTH ROCHA GONZALEZ	(55)-72385966	D.F.	0.00
03	INSURGENTES	(55)-68187324	D.F.	0.00
01631	YAZMIN AGUILAR AVILA	(55)-42243575	SANTA MARIA TULTEPEC	0.00
05786	SUSANA GARCIA BUENO	(55)-75545470	D.F.	0.00
06817	SILVIA LOPEZ MONTEROLA	(55)-210286473	D.F.	0.00
11121	ROBERTO GARCIA MORALES	(779)-7553574	TIZAYUCA	0.00
11512	RICARDO CONDE ANDRADE	(55)-543510257	D.F.	0.00
13938	PATRICIA SANCHEZ CASTILLO	(712)-4568239		0.00
14077	PATRICIA PEÑA LOPEZ	(55)-210734413	D.F.	0.00
15040	OYUKI BENITEZ ALVARADO	(55)-23851146	CIUDAD NEZAHUALCOYOTL	0.00
15489	OMAR PEREZ MEJIA	(55)-422665106	D.F.	0.00
16060	ODETH MOSCAIRO TORRES	(55)-32667536	D.F.	0.00
16317	NORMA MARTINEZ LOPEZ	(55)-25523217	D.F.	0.00
16421	NORMA BEATRIZ FERRERA	(55)-62566935	D.F.	0.00
16505	NOI MATUS CARRASCO	(55)-57353637	IXTAPALUCA	0.00

Usando where con operadores adicionales

El comando where permite el uso de los diferentes operadores de comparación o llamados operadores relacionales, los más comunes son:

Operador	Comparativa
=	Igual
>	Mayor que
<	Menor que
<>	Diferente a
<=	Menor o igual
>=	Mayor o igual
LIKE	Coincidente a
AND	y

Realicemos ahora un ejercicio usando algunos de los operadores mencionados:

```
SELECT ICOD,IDESCR,ISTKACT
FROM FINV
WHERE ISTKACT>=25
```

Se está solicitando nos presente los códigos que tengan 25 o más piezas en stock actual

ICOD	IDESCR	ISTKACT
FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	42.000
FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	54.000
FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	46.000
FGF0012VARXGR	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	43.000
FJB0311BEI300	JEANS CABALLERO EPIC-CARBON MEZ -MOD POWER	34.000
FII0003MAR010	CALCETIN DAMA MARINO SPANDEX -MOD 1503	60.000
FII0003BEI010	CALCETIN DAMA KHAKI SPANDEX -MOD 1503	71.000
FII0006MAR010	CALCETIN DAMA MARINO SPANDEX -MOD 1505	61.000
FII0006NEG010	CALCETIN DAMA NEGRO SPANDEX -MOD 1505	80.000
FII0006BEI010	CALCETIN DAMA KHAKI SPANDEX -MOD 1505	55.000
FII0013NEG010	CALCETIN CABALLERO NEGRO SPANDEX -MOD 410	68.000
FII0013CAF010	CALCETIN CABALLERO CAFE SPANDEX -MOD 410	52.000
FII0013OXF010	CALCETIN CABALLERO OXFORD SPANDEX -MOD 410	47.000
FII0017NEG010	CALCETIN CABALLERO NEGRO SPANDEX -MOD 1406	41.000

Al observar el resultado, claramente vemos que nos muestra los productos que cumplen y superan la cantidad establecida en la condición.

Combinando operadores

Al utilizar los operadores, estos permiten crear comparativas diversas, ayudándonos para poder agregar varias condiciones en conjunto.

Para poder agregar más de una condición se debe usar “AND” para filtrar la consulta con dos o más condiciones, sin que haya un límite de condiciones u operadores a utilizar.

```
SELECT ICOD,IDESCR,ISTKACT
FROM FINV
WHERE ISTKACT>=25 AND ISTKACT<=1000
```

Se está solicitando nos presente los códigos que tengan 25 o más piezas y al mismo tiempo que sean menores o igual a 1,000 en stock actual

ICOD	IDESCR	ISTKACT
FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	42.000
FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	54.000
FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	46.000
FGF0012VARXGR	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	43.000
FJB0311BEI300	JEANS CABALLERO EPIC-CARBON MEZ -MOD POWER	34.000
FII0003MAR010	CALCETIN DAMA MARINO SPANDEX -MOD 1503	60.000
FII0003BEI010	CALCETIN DAMA KHAKI SPANDEX -MOD 1503	71.000
FII0006MAR010	CALCETIN DAMA MARINO SPANDEX -MOD 1505	61.000
FII0006NEG010	CALCETIN DAMA NEGRO SPANDEX -MOD 1505	80.000
FII0006BEI010	CALCETIN DAMA KHAKI SPANDEX -MOD 1505	55.000
FII0013NEG010	CALCETIN CABALLERO NEGRO SPANDEX -MOD 410	68.000
FII0013CAF010	CALCETIN CABALLERO CAFE SPANDEX -MOD 410	52.000
FII0013OXF010	CALCETIN CABALLERO OXFORD SPANDEX -MOD 410	47.000
FII0017NEG010	CALCETIN CABALLERO NEGRO SPANDEX -MOD 1406	41.000
FII0017OXF010	CALCETIN CABALLERO OXFORD SPANDEX -MOD 1406	28.000
FII0022VAR010	CALCETIN CABALLERO PAQ-3-SURTIDO SPANDE -MO...	56.000

En el ejemplo anterior, el uso del operador ‘AND’ permite crear un rango limitado y muestra los productos que cumplen con la existencia definida en la condición.

Condiciones con campos no solicitados

Hasta este ejercicio, hemos realizado consultas a una sola tabla y hemos incorporado condiciones que permiten delimitar la información obtenida.

Ahora haremos una consulta usando un campo que no está solicitado en el comando SELECT, es decir, usaremos un campo para condicionar sin que el campo sea enlistado en el resultado.

```
SELECT CLICOD,CLINOM,CLISACT  
FROM FCLI  
WHERE CLICD='D.F.'
```

En esta consulta estamos solicitando nos presente el saldo actual de los clientes que en el campo CLICD tienen "D.F.", y en este caso dicho campo no se enlista sólo se usa como condición para filtrar los resultados que deseamos obtener

CLICOD	CLINOM	CLISACT
01	CENTRO	-181.70
05	SATELITE	0.00
00339	MARIA CAROLINA SANCHEZ GERARDO	0.00
00509	GABRIELA LEONOR ESCOBAR BADILLO	0.00
01783	YANETH MACEDA MACATIA	0.00
08585	RUTH ROCHA GONZALEZ	0.00
37167	ISIDRO CAMPUZANO ANGELES	666.59
03	INSURGENTES	0.00
05786	SUSANA GARCIA BUENO	0.00
06817	SILVIA LOPEZ MONTEROLA	0.00
11512	RICARDO CONDE ANDRADE	0.00
14077	PATRICIA PEÑA LOPEZ	0.00
15489	OMAR PEREZ MEJIA	0.00
16060	ODETH MOSCAIRO TORRES	0.00
16317	NORMA MARTINEZ LOPEZ	0.00
16421	NORMA BEATRIZ FERRERA	0.00

Cualquier campo dentro de la tabla que consideremos útil nos puede servir para crear una condición.

Agregando funciones como condición

Existen ciertas funciones que nos permiten modificar visualmente la información y también podemos usarlas como parte de una condición.

La función **MID()** sirve para extraer determinada cantidad de caracteres específicos de un campo

```
SELECT ICOD,IDESCR,ISTKACT
FROM FINV
WHERE MID(ICOD,1,3)='FGF'
```

En esta consulta, estamos solicitando, que solo muestre los productos cuyas primeras 3 letras de su código coincidan con "FGF"

ICOD	IDESCR	ISTKACT
FGF0012	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	0.000
FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	42.000
FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	54.000
FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	46.000
FGF0012VARXGR	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	43.000
FGF0221MARCHI	TRUZA CABALLERO MARINO ALGODON -MOD 31326	19.000
FGF0221MARMED	TRUZA CABALLERO MARINO ALGODON -MOD 31326	5.000
FGF0221MARGRA	TRUZA CABALLERO MARINO ALGODON -MOD 31326	19.000
FGF0221MARXGR	TRUZA CABALLERO MARINO ALGODON -MOD 31326	19.000
FGF0221GRICHI	TRUZA CABALLERO JASPE ALGODON -MOD 31326	18.000
FGF0221GRIMED	TRUZA CABALLERO JASPE ALGODON -MOD 31326	19.000
FGF0221GRIGRA	TRUZA CABALLERO JASPE ALGODON -MOD 31326	22.000
FGF0221GRIXGR	TRUZA CABALLERO JASPE ALGODON -MOD 31326	23.000

Campos con valor numérico o booleano

Existen algunos campos que se guardan en la base de datos con un valor determinado, dicho valor se define en función que el campo este 'seleccionado' o no, un ejemplo lo tenemos en la siguiente imagen donde ubicamos una palomita que activa color y talla, dicho campo se llama ICT, si se encuentra seleccionado su valor será "1" significa que es raíz

Catálogo de productos

Código: ABA2101

Descripción: BOTA CASUAL CUÑA DAMA MELLE TEXTIL -MOD 22-209

Unidad: AD | AD | Familia

Color y talla Foto

M.P. P.T. Juego Ensamble Servicio Gasto POS Prepack

y con valor "0" o no seleccionado significa que es hijo, este tipo de campos se denominan booleano

Catálogo de productos

Código: ABA2101BEI030

Descripción: BOTA CASUAL CUÑA DAMA MELLE TEXTIL -MOD 22-209

Unidad: AD | AD | Familia

Color y talla Foto

M.P. P.T. Juego Ensamble Servicio Gasto POS Prepack

Campos con valor numérico

Existe otro tipo de campos cuyo valor numérico se asigna en función al orden en que se encuentran y cuál de ellos este seleccionado, este ejemplo lo podemos observar en la siguiente imagen que nos muestra una serie de botones que nos indican:

M.P. P.T. Juego Ensamble Servicio Gasto POS Prepack

Este campo se llama ITIPO y siempre guardara un valor numérico dentro de la base de datos, de acuerdo a la imagen anterior, el valor se asigna de la siguiente forma:

MP=0 PT=1 Juego=2 Ensamble=3 Servicio=4 Gastos POS=5 Prepack=6

Sigue siendo el mismo campo pero en función del botón seleccionado corresponderá su valor.

Hagamos un ejercicio usando el campo booleano ICT, utilizando dicho campo como condición.

SELECT ICOD,IDESCR,ISTKACT

FROM FINV

WHERE ICT= 0

ICOD	IDESCR	ISTKACT
FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	42.000
FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	54.000
FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	46.000
FGF0012VARXGR	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	43.000
FJB0307AZU070	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000
FJB0307AZU090	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000
FJB0307AZU110	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000
FJB0307AZU130	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000
FJB0307AZU150	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000
FJB0311BLA300	PANTALON CABALLERO ARENA GABARDINA -MOD P...	18.000
FJB0311BLA330	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000
FJB0311BLA360	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000
FJB0311BLA400	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000
FJB0311CAF320	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000

Haciendo operaciones aritméticas

Realizar operaciones aritméticas usando la información que contiene nuestra base de datos es posible, con lo cual nos permite crear consultas y agregarles valores a aquellos datos donde lo requerimos.

Veamos la opción de generar consultas agregando operaciones, partimos con un select básico;

```
SELECT ICOD,IDESCR,ISTKACT,ILISTA1
FROM FINV
```

ICOD	IDESCR	ISTKACT	ILISTA1
FGF0012	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	0.000	24.0000
FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	42.000	24.0000
FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	54.000	24.0000
FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	46.000	24.0000
FGF0012VARXGR	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	43.000	24.0000
FJB0307AZU070	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000
FJB0307AZU090	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000
FJB0307AZU110	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000
FJB0307AZU130	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000
FJB0307AZU150	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000
FJB0311BLA300	PANTALON CABALLERO ARENA GABARDINA -MOD P...	18.000	419.0000
FJB0311BLA330	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000	419.0000
FJB0311BLA360	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000	419.0000
FJB0311BLA400	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000	419.0000

Vamos a agregar una multiplicación, dentro del comando select, indicando los campos y el operador deseado:

```
SELECT ICOD,IDESCR,ISTKACT,ILISTA1,ISTKACT*ILISTA1
FROM FINV
```

ICOD	IDESCR	ISTKACT	ILISTA1	ISTKACT*ILIST...
FGF0012	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	0.000	24.0000	0.0000000
FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	42.000	24.0000	1008.0000000
FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	54.000	24.0000	1296.0000000
FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	46.000	24.0000	1104.0000000
FGF0012VARXGR	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	43.000	24.0000	1032.0000000
FJB0307AZU070	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0307AZU090	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0307AZU110	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0307AZU130	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0307AZU150	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0311BLA300	PANTALON CABALLERO ARENA GABARDINA -MOD P...	18.000	419.0000	7542.0000000
FJB0311BLA330	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000	419.0000	7961.0000000
FJB0311BLA360	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000	419.0000	7961.0000000
FJB0311BLA400	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000	419.0000	7961.0000000
FJB0311CAF320	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000	419.0000	7961.0000000
FJB0311CAF340	PANTALON CABALLERO ARENA GABARDINA -MOD P...	8.000	419.0000	3352.0000000

Como podemos observar, se agregó una columna que contiene el resultado de la operación que realizamos, es decir, en este ejemplo estamos valorizando la existencia actual de cada producto valuado al precio de venta

También es posible utilizar operaciones aritméticas con el comando where de tal manera que se condicione la consulta con el resultado de una operación.

Usando la consulta anterior, agregamos la condición para obtener productos en existencia cuyo valor total sea menor a 1,000

```
SELECT ICOD,IDESCR,ISTKACT,ILISTA1,ILISTA1*ISTKACT
FROM FINV
WHERE (ISTKACT*ILISTA1)<1000
```

ICOD	IDESCR	ISTKACT	ILISTA1	ILISTA1*ISTKA...
FGF0012	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	0.000	24.0000	0.0000000
FJB0307AZU070	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0307AZU090	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0307AZU110	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0307AZU130	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0307AZU150	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
GFJ0189BEI010	PLAYERA CABALLERO ARENA SPANDEX -MOD PEGA...	0.000	20.0000	0.0000000
FGF0221MARMED	TRUZA CABALLERO MARINO ALGODON -MOD 31326	5.000	54.0000	270.0000000
FGF0221GRICHI	TRUZA CABALLERO JASPE ALGODON -MOD 31326	18.000	54.0000	972.0000000
FGF0221BLACHI	TRUZA CABALLERO BLANCO ALGODON -MOD 31326	18.000	54.0000	972.0000000
FGF0221BLAMED	TRUZA CABALLERO BLANCO ALGODON -MOD 31326	18.000	54.0000	972.0000000
GAD0002	CALCETIN CABALLERO BLANCO ALGODON -MOD 250-P	0.000	10.0000	0.0000000

Recordemos que los resultados de las operaciones nos pueden dar tanto valores positivos como negativos.

Realicemos otro ejemplo agregando un rango de valor derivado de una operación.

Nuestra consulta debe listar los productos en stock con valor total menor a \$1,000 y mayor a \$0, de esta forma, estamos quitando de la consulta todos aquellos en 0 o negativos

```
SELECT ICOD,IDESCR,ISTKACT,ILISTA1,ILISTA1*ISTKACT
FROM FINV
WHERE (ISTKACT*ILISTA1)<1000 AND (ISTKACT*ILISTA1)>0
```

ICOD	IDESCR	ISTKACT	ILISTA1	ILISTA1*ISTKA...
FGF0221MARMED	TRUZA CABALLERO MARINO ALGODON -MOD 31326	5.000	54.0000	270.0000000
FGF0221GRICHI	TRUZA CABALLERO JASPE ALGODON -MOD 31326	18.000	54.0000	972.0000000
FGF0221BLACHI	TRUZA CABALLERO BLANCO ALGODON -MOD 31326	18.000	54.0000	972.0000000
FGF0221BLAMED	TRUZA CABALLERO BLANCO ALGODON -MOD 31326	18.000	54.0000	972.0000000
GIC0064BLAXGR	CONTROL DAMA BLANCO SPANDEX -MOD 802	1.000	90.0000	90.0000000
FGF0221NEGCHI	TRUZA CABALLERO NEGRO ALGODON -MOD 31326	18.000	54.0000	972.0000000
FGF0221NEGMED	TRUZA CABALLERO NEGRO ALGODON -MOD 31326	10.000	54.0000	540.0000000
FGF0221NEGXGR	TRUZA CABALLERO NEGRO ALGODON -MOD 31326	17.000	54.0000	918.0000000
FGF0971AMAGRA	PLAYERA CABALLERO AMARILLO PIQUE -MOD 33912	1.000	20.0000	20.0000000
FGF0971AMAXGR	PLAYERA CABALLERO AMARILLO PIQUE -MOD 33912	1.000	20.0000	20.0000000
HAE0484BLAMED	PLAYERA DAMA BLANCO SPANDEX -MOD 0213	2.000	50.0000	100.0000000
HAE0484BLAGRA	PLAYERA DAMA BLANCO SPANDEX -MOD 0213	1.000	50.0000	50.0000000
HAE0484BLAXGR	PLAYERA DAMA BLANCO SPANDEX -MOD 0213	1.000	50.0000	50.0000000
FJB0616BLA050	JEANS DAMA BLANCO MEZCLILL -MOD YESS	2.000	120.0000	240.0000000
FJB0616BLA070	JEANS DAMA BLANCO MEZCLILL -MOD YESS	2.000	120.0000	240.0000000
FJB0616BLA130	JEANS DAMA BLANCO MEZCLILL -MOD YESS	1.000	120.0000	120.0000000

Asignar nombre a los campos (alias)

Hasta este momento, todas las consultas que hemos creado nos han mostrado como título de cada columna el nombre del campo que hemos solicitado, sin embargo, hay forma de ‘cambiar’ o ‘asignar’ un nombre a cada campo en la consulta obtenida, sin que esto signifique que se está cambiando el nombre dentro de la base de datos.

Se pueden agregar nombres a los campos para identificarlos mejor, para ello a cada campo a nombrar le agregamos el parámetro “as” y el nombre deseado. Usando el query básico del punto anterior, realicemos el ejercicio siguiente:

```
SELECT ICOD AS CODIGO ,IDESCR,ISTKACT,ILISTA1,ISTKACT*ILISTA1 AS IMPORTE  
FROM FINV
```

CODIGO	IDESCR	ISTKACT	ILISTA1	IMPORTE
FGF0012	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	0.000	24.0000	0.0000000
FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	42.000	24.0000	1008.0000000
FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	54.000	24.0000	1296.0000000
FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	46.000	24.0000	1104.0000000
FGF0012VARXGR	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	43.000	24.0000	1032.0000000
FJB0307AZU070	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0307AZU090	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0307AZU110	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0307AZU130	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0307AZU150	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0311BLA300	PANTALON CABALLERO ARENA GABARDINA -MOD P...	18.000	419.0000	7542.0000000
FJB0311BLA330	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000	419.0000	7961.0000000
FJB0311BLA360	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000	419.0000	7961.0000000
FJB0311BLA400	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000	419.0000	7961.0000000
FJB0311CAF320	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000	419.0000	7961.0000000
FJB0311CAF340	PANTALON CABALLERO ARENA GABARDINA -MOD P...	8.000	419.0000	3352.0000000
FJB0311CAF380	PANTALON CABALLERO ARENA GABARDINA -MOD P...	8.000	419.0000	3352.0000000

Podemos observar que la columna que corresponde al valor de la existencia, muestra como título el nombre que se asignó a dicha operación

Nombrando tablas (alias)

También es posible poner nombres a las tablas para referirse a ellas, esto en este momento no es necesario, pero más adelante será útil para relacionar varias tablas y referirnos a ellas.

```
SELECT ICOD AS CODIGO ,IDESCR,ISTKACT,ILISTA1,ISTKACT*ILISTA1 AS IMPORTE FROM  
FINV AS PPRODUCTOS
```

CODIGO	IDESCR	ISTKACT	ILISTA1	IMPORTE
FGF0012	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	0.000	24.0000	0.0000000
FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	42.000	24.0000	1008.0000000
FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	54.000	24.0000	1296.0000000
FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	46.000	24.0000	1104.0000000
FGF0012VARXGR	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	43.000	24.0000	1032.0000000
FJB0307AZU070	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0307AZU090	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0307AZU110	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0307AZU130	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0307AZU150	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	0.000	419.0000	0.0000000
FJB0311BLA300	PANTALON CABALLERO ARENA GABARDINA -MOD P...	18.000	419.0000	7542.0000000
FJB0311BLA330	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000	419.0000	7961.0000000
FJB0311BLA360	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000	419.0000	7961.0000000
FJB0311BLA400	PANTALON CABALLERO ARENA GABARDINA -MOD P...	19.000	419.0000	7961.0000000

El resultado será prácticamente el mismo, ya que nombrar la tabla no afecta en nada a la consulta.

Cambiando el orden del resultado en consultas

De forma predeterminada cada consulta que hacemos es presentada visualmente en el orden en que los registros están grabados dentro de la tabla que los contiene, es decir, se listan en orden secuencial de creación.

Para modificar el orden en el resultado de nuestra consulta, usaremos el comando **ORDER BY**, este comando como claramente se entiende, nos permite definir cuál de los campos solicitados servirá para ordenar la información.

Usamos nuestro select básico anterior en el que no se especifica ningún orden y agregaremos el comando **ORDER BY**, con lo cual modificaremos el resultado pidiendo que se ordene con base al campo al cual le pusimos el nombre, es decir; IMPORTE y observemos el resultado

```
SELECT ICOD,IDESCR,ISTKACT,ILISTA1,ISTKACT*ILISTA1 AS IMPORTE
FROM FINV
WHERE ISTKACT>0
ORDER BY IMPORTE
```

ICOD	IDESCR	ISTKACT	ILISTA1	IMPORTE
FGF1122BEIGRA	PANTALETA DAMA HUESO SPANDEX -MOD SE25956	1.000	5.0000	5.0000000
FGF1122BEIXGR	PANTALETA DAMA HUESO SPANDEX -MOD SE25956	1.000	5.0000	5.0000000
FGF1128BEICHI	PANTALETA DAMA HUESO SPANDEX -MOD SE25303	1.000	5.0000	5.0000000
FGF1128BEIXGR	PANTALETA DAMA HUESO SPANDEX -MOD SE25303	1.000	5.0000	5.0000000
GIC0189BEICHI	PANTALETA DAMA BEIGE SPANDEX -MOD 2388P	1.000	10.0000	10.0000000
GIC0192ROSGRA	PANTALETA DAMA FIUSHA-LUNARES SPAND -MOD 2...	1.000	10.0000	10.0000000
FGF0971AMAGRA	PLAYERA CABALLERO AMARILLO PIQUE -MOD 33912	1.000	20.0000	20.0000000
FGF0971AMAXGR	PLAYERA CABALLERO AMARILLO PIQUE -MOD 33912	1.000	20.0000	20.0000000
FGF0776MARCHI	PLAYERA DAMA MARINO PIQUE -MOD 33811/32626	1.000	20.0000	20.0000000
GIC0228BLAXGR	PANTALETA DAMA BLANCO-FLORES SPANDE -MOD 2...	1.000	20.0000	20.0000000
GEB0468AZUGRA	MALLON DAMA AZUL SPANDEX -MOD 6014	1.000	20.0000	20.0000000
GID0129ROSMED	PANTALETA DAMA ROSA SPANDEX -MOD 110726	1.000	20.0000	20.0000000
GID0129ROSGRA	PANTALETA DAMA ROSA SPANDEX -MOD 110726	1.000	20.0000	20.0000000
GID0129ROSEXGR	PANTALETA DAMA ROSA SPANDEX -MOD 110726	1.000	20.0000	20.0000000
GID0131AZUCHI	PANTALETA DAMA AZUL SPANDEX -MOD 70955	1.000	30.0000	30.0000000
GID0131AZUMED	PANTALETA DAMA AZUL SPANDEX -MOD 70955	1.000	30.0000	30.0000000
GID0131AZUGRA	PANTALETA DAMA AZUL SPANDEX -MOD 70955	1.000	30.0000	30.0000000

Podemos ver que los datos ahora se muestran ordenados por el campo importe y en automático en orden ascendente

Con el comando **ORDER BY** es posible hacer consultas ordenadas por cualquier campo, por automático el orden siempre es ascendente, pero de ser necesario podemos cambiarlo a descendente, para poder aplicar el orden en descendente usaremos la cláusula **DESC**, esta cláusula se agrega después del campo que se quiere definir como ordenador.

Usemos nuestra consulta anterior para agregar la cláusula y veamos el resultado.

```
SELECT ICOD,IDESCR,ISTKACT,ILISTA1,ISTKACT*ILISTA1 AS IMPORTE
FROM FINV
WHERE ISTKACT>0
ORDER BY IMPORTE DESC
```

ICOD	IDESCR	ISTKACT	ILISTA1	IMPORTE
FFG01790XFCHI	SWEATER CABALLERO JASPE/OXFORD ALGODON -...	1146.000	494.0000	566124.00000000
FJB1525MAR110	JEANS DAMA RING-STONE MEZCLILLA -MOD DEMI M...	178.000	419.0000	74582.00000000
FJB1525MAR090	JEANS DAMA RING-STONE MEZCLILLA -MOD DEMI M...	170.000	419.0000	71230.00000000
GDG0337NEG380	TRAJE CABALLERO NEGRO-FANTASIA POLIE -MOD S...	18.000	3000.0000	54000.00000000
GDG0337NEG400	TRAJE CABALLERO NEGRO-FANTASIA POLIE -MOD S...	18.000	3000.0000	54000.00000000
GDG0337NEG440	TRAJE CABALLERO NEGRO-FANTASIA POLIE -MOD S...	18.000	3000.0000	54000.00000000
GDG0337NEG360	TRAJE CABALLERO NEGRO-FANTASIA POLIE -MOD S...	17.000	3000.0000	51000.00000000
GDG0337NEG420	TRAJE CABALLERO NEGRO-FANTASIA POLIE -MOD S...	15.000	3000.0000	45000.00000000
FJB1525NEG110	JEANS DAMA RING-STONE MEZCLILLA -MOD DEMI N...	107.000	419.0000	44833.00000000
FJB1525NEG090	JEANS DAMA RING-STONE MEZCLILLA -MOD DEMI N...	106.000	419.0000	44414.00000000
FJB1525BLA090	JEANS DAMA RING-STONE MEZCLILLA -MOD DEMI B...	99.000	419.0000	41481.00000000
FJB1525BLA110	JEANS DAMA RING-STONE MEZCLILLA -MOD DEMI B...	99.000	419.0000	41481.00000000
FJB1525MAR050	JEANS DAMA RING-STONE MEZCLILLA -MOD DEMI M...	98.000	419.0000	41062.00000000
FJB1525MAR070	JEANS DAMA RING-STONE MEZCLILLA -MOD DEMI M...	85.000	419.0000	35615.00000000
FJB1525MAR130	JEANS DAMA RING-STONE MEZCLILLA -MOD DEMI M...	85.000	419.0000	35615.00000000

Claramente ahora la información se ha ordenado con el campo importe del valor más alto al más bajo.

Ordenando información usando varios campos a la vez

Es posible realizar consultas, solicitando se presente de manera ordenada considerando más de un campo, para hacer esto simplemente debemos nombrar los campos separándolos con coma, sigamos usando nuestro query y apliquemos otro campo al orden y observemos el resultado.

```
SELECT ICOD,IDESCR,ISTKACT,ILISTA1,ISTKACT*ILISTA1 AS IMPORTE
FROM FINV
WHERE ISTKACT>5
ORDER BY ILISTA1,ICOD
```

ICOD	IDESCR	ISTKACT	ILISTA1	IMPORTE
GEB0466VERCHI	MALLON DAMA VERDE SPANDEX -MOD 6013	20.000	20.0000	400.0000000
GEB0466VERMED	MALLON DAMA VERDE SPANDEX -MOD 6013	6.000	20.0000	120.0000000
GEB0468AZUCHI	MALLON DAMA AZUL SPANDEX -MOD 6014	7.000	20.0000	140.0000000
FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	42.000	24.0000	1008.0000000
FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	46.000	24.0000	1104.0000000
FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	54.000	24.0000	1296.0000000
FGF0012VARXGR	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	43.000	24.0000	1032.0000000
FII0003BEI010	CALCETIN DAMA KHAKI SPANDEX -MOD 1503	71.000	34.0000	2414.0000000
FII0003MAR010	CALCETIN DAMA MARINO SPANDEX -MOD 1503	60.000	34.0000	2040.0000000
FII0006BEI010	CALCETIN DAMA KHAKI SPANDEX -MOD 1505	55.000	34.0000	1870.0000000
FII0006MAR010	CALCETIN DAMA MARINO SPANDEX -MOD 1505	61.000	34.0000	2074.0000000
FII0006NEG010	CALCETIN DAMA NEGRO SPANDEX -MOD 1505	80.000	34.0000	2720.0000000
FII0030BEI010	CALCETIN DAMA KHAKI SPANDEX -MOD 1502	59.000	34.0000	2006.0000000

Aunque parece imperceptible, se está aplicando el orden por lista 1 en orden ascendente y después por código, igualmente en orden ascendente.

Incluso se puede ordenar considerando campos a los que se les asignó un nombre, hagamos el ejemplo con nuestra misma consulta, pero usemos el campo al que asignamos nombre como parte del ordenamiento:

```
SELECT ICOD,IDESCR,ISTKACT,ILISTA1,ISTKACT*ILISTA1 AS IMPORTE
FROM FINV
WHERE ISTKACT>5
ORDER BY ILISTA1,IMPORTE
```

ICOD	IDESCR	ISTKACT	ILISTA1	IMPORTE
GEB0466VERMED	MALLON DAMA VERDE SPANDEX -MOD 6013	6.000	20.0000	120.0000000
GEB0468AZUCHI	MALLON DAMA AZUL SPANDEX -MOD 6014	7.000	20.0000	140.0000000
GEB0466VERCHI	MALLON DAMA VERDE SPANDEX -MOD 6013	20.000	20.0000	400.0000000
FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	42.000	24.0000	1008.0000000
FGF0012VARXGR	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	43.000	24.0000	1032.0000000
FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	46.000	24.0000	1104.0000000
FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	54.000	24.0000	1296.0000000
FII00358LA010	CALCETIN DAMA BLANCO SPANDEX -MOD 1504	42.000	34.0000	1428.0000000
FII0030MAR010	CALCETIN DAMA OXFORD SPANDEX -MOD 1502	43.000	34.0000	1462.0000000
FII00358EI010	CALCETIN DAMA KHAKI SPANDEX -MOD 1504	46.000	34.0000	1564.0000000
FII0030GRI010	CALCETIN DAMA MARINO SPANDEX -MOD 1502	49.000	34.0000	1666.0000000
FII00350XF010	CALCETIN DAMA OXFORD SPANDEX -MOD 1504	51.000	34.0000	1734.0000000
FII0035NEG010	CALCETIN DAMA NEGRO SPANDEX -MOD 1504	52.000	34.0000	1768.0000000
FII00068EI010	CALCETIN DAMA KHAKI SPANDEX -MOD 1505	55.000	34.0000	1870.0000000
FII0030ROS010	CALCETIN DAMA ROSA SPANDEX -MOD 1502	57.000	34.0000	1938.0000000

Vamos ahora a agregar orden descendente utilizando el campo que tiene asignado un nombre

```
SELECT ICOD,IDESCR,ISTKACT,ILISTA1,ISTKACT*ILISTA1 AS IMPORTE
FROM FINV
WHERE ISTKACT>5
ORDER BY ILISTA1,IMPORTE DESC
```

ICOD	IDESCR	ISTKACT	ILISTA1	IMPORTE
GEB0466VERCHI	MALLON DAMA VERDE SPANDEX -MOD 6013	20.000	20.0000	400.0000000
GEB0468AZUCHI	MALLON DAMA AZUL SPANDEX -MOD 6014	7.000	20.0000	140.0000000
GEB0466VERMED	MALLON DAMA VERDE SPANDEX -MOD 6013	6.000	20.0000	120.0000000
FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	54.000	24.0000	1296.0000000
FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	46.000	24.0000	1104.0000000
FGF0012VARXGR	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	43.000	24.0000	1032.0000000
FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	42.000	24.0000	1008.0000000
FII0006NEG010	CALCETIN DAMA NEGRO SPANDEX -MOD 1505	80.000	34.0000	2720.0000000
FII00038EI010	CALCETIN DAMA KHAKI SPANDEX -MOD 1503	71.000	34.0000	2414.0000000
FII00300XF010	CALCETIN DAMA JASPE SPANDEX -MOD 1502	61.000	34.0000	2074.0000000
FII0006MAR010	CALCETIN DAMA MARINO SPANDEX -MOD 1505	61.000	34.0000	2074.0000000
FII0003MAR010	CALCETIN DAMA MARINO SPANDEX -MOD 1503	60.000	34.0000	2040.0000000
FII00308EI010	CALCETIN DAMA KHAKI SPANDEX -MOD 1502	59.000	34.0000	2006.0000000

Creando agrupaciones

Ahora vamos a utilizar el comando **GROUP BY**, que nos permitirá crear agrupaciones. Una agrupación nos sirve para juntar información que tenga el mismo valor en un campo determinado. Es importante señalar que al utilizar este comando, dejaremos de ver la información detallada, ya que justamente se creara un 'grupo' con el campo declarado como agrupador.

Este comando siempre se usa en combinación con la cláusula **SUM()** que se utiliza para sumar campos que almacenan un valor numérico, de esta forma se agrupa la información y además se obtiene una suma. Ambos campos deben residir dentro de la misma tabla.

Usemos un select básico:

```
SELECT IPRV,ICOD,ISTKACT  
FROM FINV
```

IPRV	ICOD	ISTKACT
FGF	FGF0012	0.000
FGF	FGF0012VARCHI	42.000
FGF	FGF0012VARMED	54.000
FGF	FGF0012VARGRA	46.000
FGF	FGF0012VARXGR	43.000
FJB	FJB0307AZU070	0.000
FJB	FJB0307AZU090	0.000
FJB	FJB0307AZU110	0.000
FJB	FJB0307AZU130	0.000
FJB	FJB0307AZU150	0.000
FJB	FJB0311BLA300	18.000
FJB	FJB0311BLA330	19.000
FJB	FJB0311BLA360	19.000
FJB	FJB0311BLA400	19.000
FJB	FJB0311CAF320	19.000
FJB	FJB0311CAF340	8.000
FJB	FJB0311CAF380	8.000
GFJ	GFJ0189BEI010	0.000

La anterior imagen solo muestra una parte del resultado obtenido, en realidad los renglones son muchos más y qué pasa si quisiéramos sumar el inventario de cada proveedor?

Agreguemos el comando y la cláusula de suma y veamos el resultado:

```
SELECT IPRV,ICOD,SUM(ISTKACT)
FROM FINV
GROUP BY IPRV
```

IPRV	ICOD	SUM(ISTKACT)
FGF	FGF0012	698.000
FGG	FGG0033	1281.000
FGH	FGH0507	1658.000
FGJ	FGJ0003	0.000
FHA	FHA0004	997.000
FHE	FHE0216	35.000
FHH	FHH0027	198.000
FIB	FIB0024	460.000
FIC	FIC0056	1418.000
FIE	FIE0006	16.000
FIF	FIF0443	6.000
FIH	FIH0002	62.000
FII	FII0003MAR010	1348.000
FJB	FJB0307AZU070	4866.000
FJC	FJC0010	82.000
FJD	FJD0003	0.000
FJF	FJF0009	14.000

Observa como el código de producto ya no muestra ninguna de las variantes debido a que se han agrupado los datos y además el campo de stock ya muestra la suma.

Si agregamos ahora un comando usado anteriormente podremos cambiar el orden implícito de nuestra consulta. Combinemos los comandos: **GROUP BY, ORDER BY, DESC** y veamos el resultado:

```
SELECT IPRV,SUM(ISTKACT)
FROM FINV
GROUP BY IPRV
ORDER BY SUM(ISTKACT) DESC
```

IPRV	SUM(ISTKACT)
FJB	4866.000
HCJ	4440.000
GCB	4182.000
GEB	3264.000
HAE	2790.000
HAF	2271.000
FCJ	2188.000
GBH	1837.000
GIC	1775.000
GCD	1669.000
FGH	1658.000
GDG	1580.000
FFG	1573.000
HAG	1569.000
GIH	1523.000
GBJ	1522.000
GDB	1466.000

Al aplicar el comando **ORDER BY** y **DESC** se ordenaron en valores descendentes.

Usando 2 agrupadores simultáneos

Simulemos el siguiente ejercicio: tenemos un select básico de proveedores con el proveedor de cada producto y a su vez queremos saber cuál es el stock que se tiene de cada proveedor por departamento, para esto necesitamos un subtotal para cada combinación de proveedor y de Departamento.

```
SELECT IPRV,IFAM2,ISTKACT  
FROM FINV
```

Nuestro select básico incluye ahora el campo IFAM2 que corresponde al departamento de cada producto y nos muestra este resultado:

IPRV	IFAM2	ISTKACT
FGF	2RCA	0.000
FGF	2RCA	42.000
FGF	2RCA	54.000
FGF	2RCA	46.000
FGF	2RCA	43.000
FJB	2RDA	0.000
FJB	2RDA	0.000
FJB	2RDA	0.000
FJB	2RDA	0.000
FJB	2RDA	0.000
FJB	2RCA	18.000
FJB	2RCA	19.000
FJB	2RCA	19.000
FJB	2RCA	19.000
FJB	2RCA	19.000
FJB	2RCA	8.000

Aunque no vemos el código del producto, hay que recordar que cada renglón corresponde a cada uno de los productos de la base de datos.

Usemos el comando de agrupación igual que en los ejercicios anteriores, pero agregamos 2 datos de la tabla, en este caso IPRV,IFAM2

Podremos observar que para el proveedor “FAB” hay varios renglones, uno para cada departamento en los cuales ese PRV tiene stock, de igual forma podemos ver que el orden ascendente aplica sobre el primero de los campos de agrupación, en este caso IPRV

```
SELECT IPRV,IFAM2,SUM(ISTKACT)
FROM FINV
GROUP BY IPRV,IFAM2
```

IPRV	IFAM2	SUM(ISTKACT)
FAB	2RDA	7.000
FAG	2RDA	62.000
FAJ	2RDA	242.000
FBJ	2RCA	624.000
FCB	2RDA	14.000
FCD	2RDA	0.000
FCG	2RCA	1405.000
FCG	2RDA	43.000
FCJ	2RDA	2188.000
FDA	2RDA	998.000
FDC	2RDA	10.000
FEF	2RCA	77.000
FEG	2RCA	767.000

Si se invierte el orden del **GROUP BY**, notemos que el orden de los campos está acomodado igual que el orden definido en el **SELECT**, aunque esto no es estrictamente necesario, sí es una buena práctica, ya que le estamos dando al usuario la información ya ordenada o acomodada.

```
SELECT IFAM2, IPRV,SUM(ISTKACT)
FROM FINV
GROUP BY IFAM2,IPRV
```

IFAM2	IPRV	SUM(ISTKACT)
2RCA	FAB	95.000
2RCA	FBJ	624.000
2RCA	FCG	1405.000
2RCA	FEF	77.000
2RCA	FEG	767.000
2RCA	FFG	1477.000
2RCA	FFH	172.000
2RCA	FGA	1.000
2RCA	FGB	4.000
2RCA	FGF	471.000
2RCA	FGG	1281.000
2RCA	FHE	35.000
2RCA	FHH	148.000
2RCA	FIC	374.000
2RCA	FIE	16.000

Ordenar una agrupación con una columna sumada.

Como ya hemos visto en ejercicios anteriores, es posible definir campos que ordenen el resultado de nuestras consultas, combinemos ahora el comando **ORDER BY** para ordenar por el total de forma descendente

```
SELECT IPRV,IFAM2,SUM(ISTKACT)
FROM FINV
GROUP BY IFAM2,IPRV
ORDER BY SUM(ISTKACT) DESC
```

IPRV	IFAM2	SUM(ISTKACT)
HCJ	2RDA	4440.000
FJB	2RDA	3420.000
GEB	2RDA	3264.000
HAE	2RDA	2790.000
GCB	2RCA	2496.000
FCJ	2RDA	2188.000
HAF	2RCA	1949.000
GBH	2RCA	1837.000
GIC	2RDA	1775.000
GCB	2RDA	1686.000
GCD	2RDA	1669.000
FGH	2RDA	1658.000
GDG	2RCA	1580.000
GIH	2RDA	1523.000
GBJ	2RDA	1522.000

Ahora generemos una consulta que sume el stock por proveedor, sin importar el departamento y asignemos un nombre a dicha sumatoria además de modificar su ordenamiento y veamos el resultado.

```
SELECT IPRV,SUM(ISTKACT) AS TOTAL  
FROM FINV  
GROUP BY IPRV  
ORDER BY TOTAL DESC
```

IPRV	TOTAL
FJB	4866.000
HCJ	4440.000
GCB	4182.000
GEB	3264.000
HAE	2790.000
HAF	2271.000
FCJ	2188.000
GBH	1837.000
GIC	1775.000
GCD	1669.000
FGH	1658.000
GDG	1580.000
FFG	1573.000
HAG	1569.000
GIH	1523.000
GBJ	1522.000

Condicionando una agrupación

Como ya lo hicimos en ejercicios anteriores, agreguemos una condición a nuestra agrupación, recordemos que el comando where nos permite hacerlo:

```
SELECT IPRV,SUM(ISTKACT) AS TOTAL
FROM FINV
WHERE ISTKACT>=200
GROUP BY IPRV
```

IPRV	TOTAL
FFG	1146.000
IBI	420.000

Este Query nos da el inventario total por proveedor, de aquellos productos que tienen más 200 piezas en total.

Vamos a analizar la información y entender la funcionalidad de la condición aplicada, ya que como lo mencionamos al inicio del punto agrupaciones, al agrupar el detalle se omite y poder comprender como está aplicando la condición agregada.

Analicemos el detalle de 1 proveedor (IBI) cuyo total del query anterior es 420, para ello modificaremos el query quitando la agrupación y la condición de mayores a 200 piezas, agregamos el filtro para solo el obtener el proveedor IBI y ordenamos en descendente:

```
SELECT IPRV,ISTKACT
FROM FINV
WHERE IPRV='IBI'
ORDER BY ISTKACT DESC
```

IPRV	ISTKACT
IBI	220.000
IBI	200.000
IBI	26.000
IBI	21.000
IBI	18.000
IBI	14.000
IBI	13.000
IBI	11.000
IBI	11.000
IBI	11.000
IBI	9.000
IBI	9.000
IBI	8.000

Como podemos observar, existen muchos más registros de producto que corresponden al proveedor 'IBI', si se compara con el query agrupado anterior, el resultado de las piezas debiera ser mayor a 420, porque sucede esto?

Para poder entender esta aparente diferencia, debemos recordar que el comando Where opera a nivel registro, es decir por cada producto, y el query agrupado que ejecutamos, al ponerle **WHERE** ISTKACT>=200, primero filtra todos los productos que tienen 200 piezas o más y luego los suma.

Filtrando después de agrupar

Ahora utilizaremos el comando **HAVING** con este comando podremos filtrar un resultado después de haberlo agrupado, es decir, la consulta buscara los registros que cumplan con la condición, después agrupara y finalmente sobre dicho resultado agrupado aplicara el filtrado.

Utilizando el query que hicimos en el punto anterior, usaremos el comando **HAVING** de la siguiente manera: filtraremos únicamente los proveedores con un inventario total igual o superior a 200 piezas.

```
SELECT IPRV,SUM(ISTKACT) AS TOTAL
FROM FINV
GROUP BY IPRV HAVING TOTAL>=200
ORDER BY TOTAL DESC
```

IPRV	TOTAL
FJB	4866.000
HCJ	4440.000
GCB	4182.000
GEB	3264.000
HAE	2790.000
HAF	2271.000
FCJ	2188.000
GBH	1837.000
GIC	1775.000
GCD	1669.000
FGH	1658.000
GDG	1580.000
FFG	1573.000
HAG	1569.000

Como hemos visto es posible combinar comandos dentro de las sentencias, ahora combinaremos el uso de los comandos **WHERE** que si recordamos sirve para agregar condiciones y el comando **HAVING** que permite agregar un filtro adicional sobre una agrupación y para conseguirlo usaremos también los comparadores lógicos “>=”

Realicemos otro ejercicio usando el comando **HAVING** para conseguir otro resultado.

Si queremos todos los proveedores que su letra inicial sea "H" y que su existencia total sea más de 1,000 piezas, debemos ejecutar este Query, en donde la sentencia **WHERE** filtra a los proveedores que inician con "H" y el **HAVING** para filtrar aquellos que su total sea mayor a 1,000.

Nuestro query consta de 3 partes; la primera selecciona todos los productos filtrando únicamente a los proveedores que inician con "H", y en un segundo paso suma las piezas de cada proveedor, para finalmente en un tercer paso filtrar los proveedores que tienen más de 1,000 piezas y veamos el resultado:

```
SELECT IPRV,SUM(ISTKACT) AS TOTAL
FROM FINV
WHERE MID(IPRV,1,1)='H'
GROUP BY IPRV HAVING TOTAL>1000
ORDER BY TOTAL DESC
```

IPRV	TOTAL
HCJ	4440.000
HAE	2790.000
HAF	2271.000
HAG	1569.000
HGE	1252.000
HCA	1222.000
HBG	1139.000

Vamos paso a paso la consulta anterior, para comprobar como cada etapa de nuestro query se va aplicando e identificar los comandos que vamos agregando para conseguirlo.

Paso 1 empecemos por filtrar los productos con stock de los proveedores que empiezan con "H" y veamos el resultado:

```
SELECT IPRV,ISTKACT AS TOTAL
FROM FINV
WHERE MID(IPRV,1,1)='H' AND ISTKACT>0
ORDER BY IPRV
```

IPRV	TOTAL
HAE	2.000
HAE	1.000
HAE	1.000
HAE	20.000
HAE	18.000
HAE	21.000
HAE	17.000
HAE	19.000
HAE	18.000
HAE	18.000
HAE	18.000
HAE	1.000

Paso 2 Ahora suma los productos sub totalizando por proveedor

```
SELECT IPRV,SUM(ISTKACT) AS TOTAL
FROM FINV
WHERE MID(IPRV,1,1)='H'
GROUP BY IPRV
ORDER BY TOTAL DESC
```

IPRV	TOTAL
HCJ	4440.000
HAE	2790.000
HAF	2271.000
HAG	1569.000
HGE	1252.000
HCA	1222.000
HBG	1139.000
HFB	914.000
HII	895.000
HCD	761.000
HFF	616.000

Paso 3 Finalmente filtremos los que no cumplan con la condición del **HAVING**, es decir sólo deja los que tengan más de 1,000 piezas.
En este query también quitamos el **ORDER BY** sólo para mostrar su efecto,

```
SELECT IPRV,SUM(ISTKACT) AS TOTAL
FROM FINV
WHERE MID(IPRV,1,1)='H'
GROUP BY IPRV HAVING TOTAL>1000
ORDER BY TOTAL DESC
```

IPRV	TOTAL
HCJ	4440.000
HAE	2790.000
HAF	2271.000
HAG	1569.000
HGE	1252.000
HCA	1222.000
HBG	1139.000

Así como en el **WHERE** se pueden combinar varias comparaciones, lo mismo se puede hacer con el **HAVING**

Vamos a modificar el query para agregar doble condición de filtros tanto con el comando **WHERE** como dentro de comando **HAVING**

Aquí en el **WHERE** se combinan dos filtros y en el **HAVING** 2 filtros

```
SELECT IPRV,SUM(ISTKACT) AS TOTAL
FROM FINV
WHERE MID(IPRV,1,1)>='A' AND MID(IPRV,1,1)<='H'
GROUP BY IPRV HAVING TOTAL>200 AND TOTAL<1000
ORDER BY TOTAL
```

Y veamos el resultado:

IPRV	TOTAL
FAJ	242.000
FFH	273.000
FJH	432.000
FIB	460.000
FBJ	624.000
FGF	698.000
FEG	767.000
FHA	997.000
FDA	998.000

Hasta este punto hemos partido de ir creando consultas básicas e ir incorporando más comandos que nos permiten modificar los resultados con nuestra información.

Aquí tenemos la estructura básica de un query y los comandos que hemos estado usando para extraer datos de una tabla.

```
SELECT
FROM
WHERE
GROUP BY
HAVING
ORDER BY
```

CAPÍTULO 4 CREANDO CONSULTAS CON MAS DE 1 TABLA

Hasta este momento hemos visto Querys básicos utilizando únicamente 1 tabla, las bases de datos relacionales y en particular la de **Proscail** tiene muchas tablas, donde en cada una de ellas se guardan diferentes tipos de datos.

Todas estas tablas tienen una relación con otras más y esto nos permite poder generar querys relacionando más de una tabla.

Para los ejemplos de este capítulo nos basaremos en usar la tabla FDOC que guarda todos los datos de un documento y la ligaremos hacia otras tablas que se relacionan a ella.

Si recordamos el capítulo 2 y el tema “modelos de conexión entre tablas”, se explicaban diversos casos donde la tabla FDOC se relaciona hacia otras debido a que por naturaleza y principio de control existe un documento que ampara la operación.

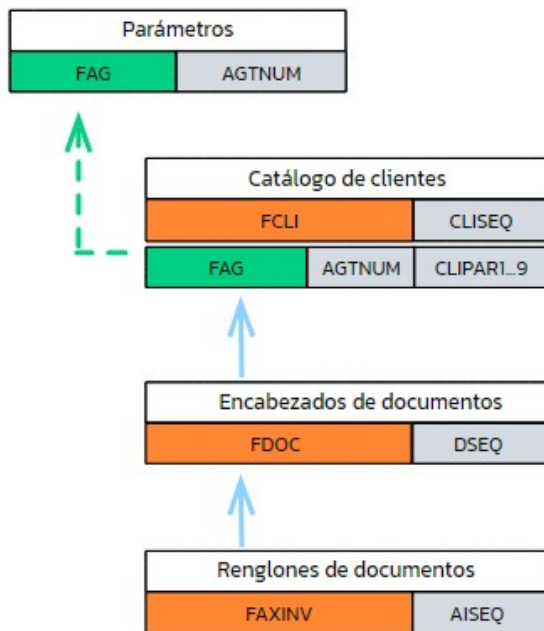
Recordemos que dentro de una base relacional se pueden ligar tablas a través de las conexiones naturales o directas mediante las secuencias que se heredan, y conexiones indirectas que se generan a través de algunos campos que guardan una referencia y que sirve para conectar una tabla no ligada.

La jerarquía de tablas y el detalle de información

Para una correcta explotación de la información al crear consultas, donde requerimos usar más de una tabla, siempre debemos recordar que la estructura jerárquica en la que está definida la base de datos Proscail, nos indica que las tablas padre siempre tienen información a un nivel general, es decir, no cuentan con el detalle máximo de información.

Si una tabla (padre) tiene una liga directa hacia una tabla inferior (hijo), y esta a su vez también tiene una liga directa hacia un nivel inferior (nieto), significa que el detalle máximo de información estará siempre en los niveles más bajos.

Recordemos nuestro esquema que explica la jerarquía y expliquemos un ejemplo:



En el diagrama se puede ver la jerarquía donde entre más arriba está una tabla, representa que es de mayor nivel y entre más abajo, que es de mayor detalle.

Las flechas representan que hay una relación entre las tablas, en donde el inicio de la flecha representa una tabla que puede contener varios registros relacionados a la tabla hacia donde la flecha apunta.

Ubiquemos la tabla FCLI, sabemos que a ese nivel el cliente solo tiene un campo con el monto total que adeuda (CLISACT). Si requerimos saber cuántos y qué documentos específicamente son, debemos bajar un nivel y entrar a la tabla FDOC que hereda de FCLI la llave directa (CLISEQ) que nos permite saber qué documentos pertenecen a él, y a su vez, obtener del documento el monto individual de cada uno, cuya suma corresponderá al importe acumulado en el cliente, pero el documento no contiene las cantidades o precios de los productos involucrados, FDOC tiene una liga directa hacia un nivel inferior con la tabla FAXINV que a su vez hereda de las tablas de niveles superiores a ella, las llaves primarias de cada una, es decir CLISEQ y DSEQ respectivamente.

Por lo tanto, la tabla del nivel más inferior FAXINV contiene el mayor detalle de información. Si deseamos obtener un reporte de ventas debemos preguntarnos ¿a qué nivel? Si solo se desea el importe sin detallar cantidades ni precios, la tabla FDOC es nuestro punto de arranque; si deseamos saber cuánto debe el cliente sin importar qué documentos o de qué fecha, la tabla FCLI tiene el saldo ya acumulado; si por el contrario, requerimos un nivel máximo que involucre desglose de cantidades y precios, documentos y datos de fechas además del cliente, debemos ubicarnos desde el nivel más bajo (FAXINV), y usando las ligas directas heredadas de cada uno de los niveles anteriores (CLISEQ y DSEQ) podremos obtener la información al máximo nivel de detalle y al mismo tiempo utilizar los niveles superiores para complementar el análisis.

En este momento es importante volver a explicar una serie de campos que existen dentro de la tabla FDOC y que permitirán de forma sencilla determinar la naturaleza del tipo de documento que representa un registro, es decir, la tabla FDOC contiene en conjunto todos los documentos que se generan dentro de **Proscái**, pero es necesario definir si es factura, recepción, ticket, mov. De inventarios, con saldo, sin saldo, cancelado, etc; esta identificación nos permitirá crear queries precisos para información de ventas, compras o producción.

A este grupo de campos le llamamos **banderas**, ya que su principal objetivo es identificar un registro y de acuerdo a su valor y significado es que podremos hacerlo. El uso adecuado de las banderas como filtros o condiciones, nos permitirá delimitar adecuadamente que tipo de información estamos solicitando, de esta forma, estos campos o banderas serán primordialmente necesarios cuando pretendamos realizar consultas que nos devuelvan información sensible y que valore operaciones de venta, compra, inventarios, cuentas por cobrar o por pagar, entre algunas otras.

También es importante explicar que un solo registro, combina todas las banderas y los valores contenidos nos dirán origen y afectación del mismo dentro del sistema.

Veamos nuevamente la tabla que nos muestra el nombre del campo (**bandera**) el valor que puede tener y el resultado de dicho valor:

Nombre campo	Si tiene valor:	Significado o interpretación
DESFACT	0	Es un documento que no se generó en ningún módulo de ventas, compras o producción.
	1	Es un documento generado desde el módulo facturación
	2	Es un documento generado desde el módulo recepciones
	3	Es un documento generado desde el módulo producción para el consumo de materiales
	4	Es un documento generado desde el módulo producción para pago a maquileros
	8	Es un documento de complemento de pago timbrado
DESCXC	0	Es un documento que ya no tiene saldo pendiente
	1	Es un documento que tiene saldo pendiente de cobro a clientes
	2	Es un documento que tiene saldo pendiente de pago a proveedores
DESINV	0	Es un documento que no afectó el módulo de inventarios
	1	Es un documento que si afectó el módulo de inventarios
DCANCELADA	0	No está cancelado
	1	Si está cancelado
DSTATUSCFD	0	No es documento CFDI
	1	Documento de ventas pendiente timbre CFDI
	3	Documento de ventas timbrado y validado CFDI
	-1	Documento de compras pendiente de ligar a su XML
	-3	Documento de compras ya ligado a su XML
DITIPMV		El valor corresponderá al tipo de movimiento utilizado en la operación

Cuando pensamos en la conexión entre tablas, siempre debemos recordar que La base natural de **Proscai** es jerárquica, ejemplo: un cliente siempre tendrá “muchos” documentos, lo que lo hace superior, mientras que un documento solo puede pertenecer a un cliente.

Para poder realizar una conexión de tablas, usaremos el comando: **LEFT JOIN**

Por la estructura de tipo jerárquico que maneja la base de datos Proscai, este comando será el utilizado para unir las tablas tanto de forma directa como indirecta.

Es importante mencionar que dentro del lenguaje SQL existe el comando **INNER JOIN** que también se usa para realizar conexiones, sin embargo no se utilizara en ninguno de los ejemplos ya que los resultados que se obtienen en su uso puede omitir algunos registros, pero eso no significa que se limite su uso, si estas familiarizado con el podrás usarlo.

Conectando 2 tablas

Empecemos con un ejercicio básico donde podamos unir 2 tablas.

Hagamos una consulta para obtener todos los documentos que pertenecen a cada cliente y que se encuentren dentro de determinado rango de fechas y que cumpla la característica de haber sido emitidos desde facturación, es decir documentos de venta, esto último mediante el uso de las banderas que explique en el punto anterior, nuestro query quedaría así:

```
SELECT CLICOD,CLINOM,CLIEDO,DNUM,DFECHA,DCANTF
FROM FDOC
LEFT JOIN FCLI ON FCLI.CLISEQ=FDOC.CLISEQ
WHERE DESFACT=1 AND DFECHA>='2018-03-01' AND DFECHA <='2019-03-01'
```

CLICOD	CLINOM	CLIEDO	DNUM	DFECHA	DCANTF
01	CENTRO	D.F.	T01029945	2018-03-01	150.00
21619	MARIO MARTINEZ SEGURA	MEX.	T01029951	2018-03-01	388.00
01	CENTRO	D.F.	T01029952	2018-03-01	60.00
01	CENTRO	D.F.	T01029956	2018-03-01	120.00
01	CENTRO	D.F.	T01029958	2018-03-01	150.00
01	CENTRO	D.F.	T01029959	2018-03-01	300.00
05786	SUSANA GARCIA BUENO	D.F.	T05006250	2018-03-01	513.00
01	CENTRO	D.F.	T01029964	2018-03-01	580.00
01	CENTRO	D.F.	T01029967	2018-03-01	140.00
19740	MERCEDES CASTRO CASTRO	D.F.	T01029969	2018-03-01	0.00
03	INSURGENTES	D.F.	T03015506	2018-03-01	480.00
03	INSURGENTES	D.F.	T03015507	2018-03-01	188.00
01	CENTRO	D.F.	T01029970	2018-03-01	320.00
01	CENTRO	D.F.	T01029972	2018-03-01	120.00
05	SATELITE	D.F.	T05006255	2018-03-01	169.00
35533	JAVIER ALVAREZ CRUZ	MEX.	T05006256	2018-03-01	320.00

Analicemos el query y observa que después del comando **FROM** se declara la tabla FDOC ya que es ahí donde reside la información de documentos, después se agrega el comando **LEFT JOIN** que sirve para conectar los documentos hacia los clientes y la forma de definir la conexión se explica de la siguiente forma:

FCLI.CLISEQ=FDOC.CLISEQ recuerda que en la tabla de clientes cada cliente tiene una llave primaria que lo identifica, dicha llave es el campo CLISEQ y en la tabla FDOC esa misma llave fue heredada, para saber ese documento a que cliente pertenece.

En otras palabras, el comando **LEFT JOIN** conectara a los documentos con su respectivo cliente a través de la llave primaria CLISEQ al que correspondan.

Y dentro del comando **WHERE** estamos usando la bandera DESFACT para buscar los registros que tengan valor 1 que equivale a documentos de venta.

En lenguaje coloquial, este query nos está mostrando todas las ventas que hemos realizado a nuestros clientes en el periodo de un año.

Realicemos otro ejercicio similar, pero haremos unos pequeños cambios para obtener todos los documentos que nos deben pagar los clientes.

Los cambios que debemos aplicar son: cambiar la bandera DESFACT por DESCXC y quitemos el rango de fechas.

```
SELECT CLICOD,CLINOM,DNUM,DFECHA,DCANT
FROM FCLI
LEFT JOIN FDOC ON FCLI.CLISEQ=FDOC.CLISEQ
WHERE DESCXC=1
```

CLICOD	CLINOM	DNUM	DFECHA	DCANT
37167	ISIDRO CAMPUZANO ANGELES	T05008948	2018-05-17	666.59
52672	DOMINGO RAMON BUSTOS	T05014463	2018-11-17	566.50

El resultado obtenido nos dirá cada documento a que cliente corresponde y el valor que el cliente aún debe pagarnos.

En este query, los cambios fueron muy sencillos y la información obtenida ahora nos habla de lo correspondiente a cuentas por cobrar.

Al existir conexión entre 2 tablas, esto permite que podamos usar cualquier campo de ambas tablas tanto para visualizarlos o como condiciones como lo mostraremos en el ejercicio siguiente:

```
SELECT CLICOD,CLINOM,CLIEDO,DNUM,DFECHA,DCANTF
FROM FDOC
LEFT JOIN FCLI ON FCLI.CLISEQ=FDOC.CLISEQ
WHERE DESFACT=1 AND CLIEDO='D.F.'
```

CLICOD	CLINOM	CLIEDO	DNUM	DFECHA	DCANTF
01	CENTRO	D.F.	T01023254	2018-01-01	174.00
01	CENTRO	D.F.	T01023255	2018-01-01	234.00
03	INSURGENTES	D.F.	T03012184	2018-01-01	419.00
01	CENTRO	D.F.	T01023268	2018-01-01	343.00
01	CENTRO	D.F.	T01023274	2018-01-01	259.00
03	INSURGENTES	D.F.	T03012189	2018-01-01	269.00
01	CENTRO	D.F.	T01023280	2018-01-01	206.00
01	CENTRO	D.F.	T01023292	2018-01-01	912.00
01	CENTRO	D.F.	T01023296	2018-01-01	219.00
03	INSURGENTES	D.F.	T03012196	2018-01-01	194.00
16421	NORMA BEATRIZ FERRERA	D.F.	T01023306	2018-01-01	180.00
01	CENTRO	D.F.	T01023314	2018-01-01	359.00
01	CENTRO	D.F.	T01023316	2018-01-01	278.00
19740	MERCEDES CASTRO CASTRO	D.F.	T01023327	2018-01-01	164.00
01	CENTRO	D.F.	T01023332	2018-01-01	54.00

Observa que el comando **WHERE** Condiciona con un campo que reside en cada una de ellas, la conexión entre ambas, permite dicho funcionamiento.

Realicemos otro ejercicio de consulta a 2 tablas simultaneas, pero cambiando las tablas de uso.

Vamos a obtener los productos vendidos, donde cada renglón corresponde a un renglón dentro de un ticket, observa el comando **LEFT JOIN** que conectara las tablas involucradas en nuestro query. Gracias al campo que sirve de filtro (AITIPMV='T') se puede identificar la operación y su efecto en la información, dicho campo se vuelve muy importante en el uso y como parte de las condicionantes al consultar información.

```
SELECT ICOD, IDESCR,AITIPMV,AICANTF
FROM FAXINV
LEFT JOIN FINV ON FINV.ISEQ=FAXINV.ISEQ
WHERE AITIPMV='T'
```

ICOD	IDESCR	AITIPMV	AICANTF
HAF0385NEGCHI	PLAYERA CABALLERO NEGRO SPANDEX -MOD 5505	T	1.000
GBJ0206VERMED	VESTIDO DAMA VERDE SPANDEX -MOD 622	T	1.000
IAH1197BEIMED	SUDADERA CABALLERO BEIGE POLIESTER -MOD 3Y...	T	1.000
HAF0428BLAMED	PLAYERA CABALLERO BLANCO SPANDEX -MOD 5555	T	1.000
GBH0557AZUCHI	PLAYERA CABALLERO PLUMBAGO ALGODON/POL -M...	T	1.000
HDC0129NEGMED	PANTALON DAMA NEGRO-RAYAS SPANDEX -MOD 23...	T	1.000
HBJ0001NEGXGR	BLUSA DAMA NEGRO POLIESTER -MOD 9054	T	1.000
FII0017NEG010	CALCETIN CABALLERO NEGRO SPANDEX -MOD 1406	T	1.000
FII00170XF010	CALCETIN CABALLERO OXFORD SPANDEX -MOD 1406	T	1.000
FGF0221MARMED	TRUZA CABALLERO MARINO ALGODON -MOD 31326	T	1.000
FGF0221NEGXGR	TRUZA CABALLERO NEGRO ALGODON -MOD 31326	T	1.000

Usando comandos de ejercicios anteriores, agregaremos una agrupación y con ello sumar las ventas de cada producto.

```
SELECT ICOD, IDESCR,AITIPMV,SUM(AICANTF)
FROM FAXINV
LEFT JOIN FINV ON FINV.ISEQ=FAXINV.ISEQ
WHERE AITIPMV='T'
GROUP BY ICOD
```

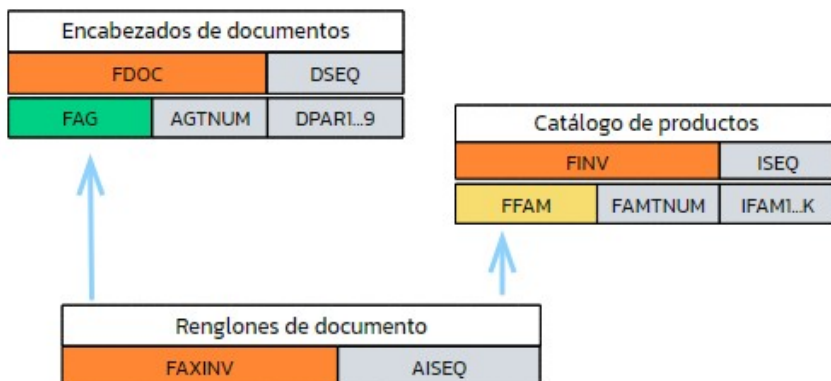
ICOD	IDESCR	AITIPMV	SUM(AICANTF)
FAB0012CAFCHI	CHAMARRA DAMA CAFE POLIESTER -MOD 1499w1-101	T	1.000
FAB0023GRICHI	SUDADERA CABALLERO GRIS POLIESTER -MOD 141...	T	1.000
FAB0023GRIGRA	SUDADERA CABALLERO GRIS POLIESTER -MOD 141...	T	1.000
FAB0023GRIMED	SUDADERA CABALLERO GRIS POLIESTER -MOD 141...	T	1.000
FAB0023GRIXGR	SUDADERA CABALLERO GRIS POLIESTER -MOD 141...	T	2.000
FAB0023NEGGRA	SUDADERA CABALLERO NEGRO POLIESTER -MOD 1...	T	4.000
FAB0023NEGMED	SUDADERA CABALLERO NEGRO POLIESTER -MOD 1...	T	4.000
FAB0023NEGXGR	SUDADERA CABALLERO NEGRO POLIESTER -MOD 1...	T	1.000
FAB0028CAFGRA	CHALECO CABALLERO CAFE ALGODON-POLIEST -MO...	T	3.000
FAB0028CAFGR	CHALECO CABALLERO CAFE ALGODON-POLIEST -MO...	T	2.000
FAB0028GRICHI	CHALECO CABALLERO GRIS ALGODON-POLIEST -MO...	T	2.000
FAB0028GRIGRA	CHALECO CABALLERO GRIS ALGODON-POLIEST -MO...	T	3.000
FAB0028GRIMED	CHALECO CABALLERO GRIS ALGODON-POLIEST -MO...	T	2.000
FAB0028GRIXGR	CHALECO CABALLERO GRIS ALGODON-POLIEST -MO...	T	4.000
FAB0028NEGGRA	CHALECO CABALLERO NEGRO ALGODON-POLIES -M...	T	6.000

El resultado obtenido nos muestra la información acumulada.

Conectando 3 tablas

Pero qué pasa si queremos filtrar ahora un rango de fechas. Podemos agregar una nueva tabla, en este caso FDOC que es quién tiene las fechas de los movimientos, ya que este dato no existe en la tabla FAXINV y poder agregar al filtro una fecha al Query.

De acuerdo a nuestro diagrama las tablas involucradas son: FAXINV, FINV, FDOC



La liga directa se genera porque FAXINV hereda las secuencias de FINV y FDOC, para ligar los movimientos a su producto específico, así como saber los datos de fecha entre otros en que fue generado el documento.

En este Query, cambiamos el filtro, en vez de usar AITIPMV='F' usaremos DESFACT=1, siendo una forma más genérica de extraer los documentos facturados.

Esto va a traer todos los documentos generados en el módulo de facturación incluyendo facturas y notas de crédito.

Es importante notar que la forma de relacionar la tabla FDOC con FAXINV es la misma que se usó en los ejercicios anteriores.

```

SELECT ICOD, IDESCR, AITIPMV, SUM(AICANTF), DFECHA
FROM FAXINV
LEFT JOIN FINV ON FINV.ISEQ=FAXINV.ISEQ
LEFT JOIN FDOC ON FDOC.DSEQ=FAXINV.DSEQ
WHERE DESFACT=1 AND DFECHA='2018-06-01'
GROUP BY ICOD
  
```

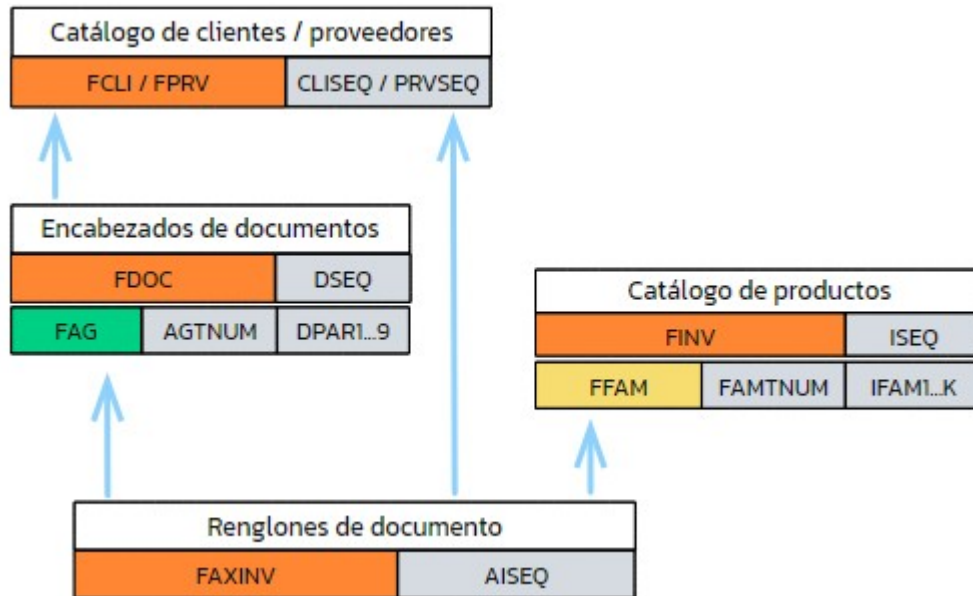
ICOD	IDESCR	AITIPMV	SUM(AICANTF)	DFECHA
FAB0081NEGMED	SUDADERA CABALLERO NEGRO ALGODON-POLIES -...	T	1.000	2018-06-01
FAG0207GRIGRA	FALDA DAMA GRIS-FLORES MEZCLILL -MOD 2344	T	1.000	2018-06-01
FCG0118AMACHI	PLAYERA DAMA PAJA ALGODON-POLIEST -MOD ST-P...	T	1.000	2018-06-01
FCG0118MORCHI	PLAYERA DAMA GRANADA ALGODON-POLI -MOD ST-...	T	1.000	2018-06-01
FCJ0175GRI110	JEANS DAMA STONE MEZCLILLA-STRE -MOD 038-SALV	T	1.000	2018-06-01
FDA0566NEG360	PANTALON DE VESTIR DAMA NEGRO-CUADROS SPA...	T	1.000	2018-06-01
FFG01450XFCHI	CHALECO CABALLERO OXFORD ACRILICO -MOD CH-E...	T	1.000	2018-06-01
FGA0114VAR050	JEANS DAMA BLEANCH MEZCLILLA-ST -MOD D436-MG	T	1.000	2018-06-01
FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	T	1.000	2018-06-01
FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	T	2.000	2018-06-01
FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	T	2.000	2018-06-01

Como podemos observar, para cada conexión de tablas que se requiera hacer es necesario agregar un comando **LEFT JOIN** para cada tabla.

Conectando cuatro tablas

Podemos seguir agregando tablas, en este caso la de clientes, es decir ahora podemos filtrar o agrupar por esta tabla. En este caso agruparemos las ventas por cliente y producto

Nuestro diagrama base ahora incluiría las siguientes tablas:



```

SELECT CLICOD,CLINOM,ICOD, IDESCR,AITIPMV,SUM(AICANTF),DFECHA
FROM FAXINV
LEFT JOIN FINV ON FINV.ISEQ=FAXINV.ISEQ
LEFT JOIN FDOC ON FDOC.DSEQ=FAXINV.DSEQ
LEFT JOIN FCLI ON FCLI.CLISEQ=FAXINV.CLISEQ
WHERE DESFACT=1 AND DFECHA='2018-06-01'
GROUP BY CLICOD,ICOD
  
```

CLICOD	CLINOM	ICOD	IDESCR	AITIPMV	SUM(AICANTF)	DFECHA
01	CENTRO	FAB0081NEGMED	SUDADERA CABALLERO NEGRO ALGODON-POLIES -...	T	1.000	2018-06-01
01	CENTRO	FCG0118AMACHI	PLAYERA DAMA PAJA ALGODON-POLIEST -MOD ST-P...	T	1.000	2018-06-01
01	CENTRO	FCG0118MORCHI	PLAYERA DAMA GRANADA ALGODON-POLI -MOD ST-...	T	1.000	2018-06-01
01	CENTRO	FGG0067GRI360	PANTALON CABALLERO VINO GABARDINA-STRET -M...	T	1.000	2018-06-01
01	CENTRO	FIC0112AZUCHI	PLAYERA CABALLERO AZUL POLIESTER -MOD BARC...	T	1.000	2018-06-01
01	CENTRO	FJB1411BEI070	JEANS DAMA BLACK MEZCLILLA-STRE -MOD ALISSON	T	1.000	2018-06-01
01	CENTRO	FJB1666NEG050	PANTALON CORTO DAMA CORAL GABARDINA-STRE ...	T	1.000	2018-06-01
01	CENTRO	FJC0087BEI010	SWEATER DAMA HUESO ACRILICO -MOD 4-0021	T	1.000	2018-06-01
01	CENTRO	GBH0659ROJMED	PLAYERA CABALLERO ROJO SPANDEX -MOD 5313	T	1.000	2018-06-01
01	CENTRO	GBH0681LIMCHI	PLAYERA CABALLERO MARINO ALGODON/POLIE -M...	T	1.000	2018-06-01
01	CENTRO	GCB3180GRIGRA	PLAYERA CABALLERO 1049569.JASPE ALGOD? -MOD...	T	1.000	2018-06-01

En este ejercicio, observamos que las tablas conectadas son en total 4, 3 de ellas a través del comando **LEFT JOIN** y la cuarta tabla es la tabla base de donde estamos extrayendo la información.

Aunque parece complicado, en realidad hemos estado usando los mismos comandos que empezamos conociendo, solo hemos agregado conexiones a más tablas.

Agregando un contador a una agrupación

Si queremos saber cuántos registros tenemos de cada proveedor podemos agregar la función SUM(1), el comando **GROUP BY** ya lo hemos usado, aprovecharemos la función de agrupación y suma que realiza para obtener una suma.

Esta función lo que hace es sumar 1 a cada registro que se lea y así tener un total de productos por cada proveedor

```
SELECT IPRV,SUM(ISTKACT),SUM(1)
FROM FINV
GROUP BY IPRV
```

IPRV	SUM(ISTKACT)	SUM(1)
FAB	102.000	91
FAG	62.000	90
FAJ	242.000	379
FBJ	624.000	245
FCB	14.000	101
FCD	0.000	10
FCG	1448.000	332
FCJ	2188.000	485
FDA	998.000	592
FDC	10.000	25
FEF	77.000	33
FEG	767.000	55
FEI	110.000	75
FEJ	200.000	118

Modelando consultas como reportes de ventas

Haciendo uso de varios comandos que hemos usado en todos los capítulos anteriores, ahora crearemos consultas que asemejen reportes de ventas por producto, veamos el primer query:

```
SELECT ICOD, IDESCR, SUM(AICANTF), ROUND(SUM(AICANTF*AIPRECIO),2) IMPORTE
FROM FAXINV
LEFT JOIN FINV ON FINV.ISEQ=FAXINV.ISEQ
LEFT JOIN FDOC ON FDOC.DSEQ=FAXINV.DSEQ
WHERE DESFACT=1 AND DFECHA>='2018-06-01' AND DFECHA<='2019-06-30'
GROUP BY ICOD
ORDER BY IMPORTE DESC
```

ICOD	IDESCR	SUM(AICANTF)	IMPORTE
HCJ0033AZUMED	BLUSA DAMA AZUL-RAYAS ALGODON -MOD JR3251	15.000	3000.86
FHH00440XFMED	CHAMARRA CABALLERO OXFORD POLIESTER -MOD ...	4.000	2358.62
FCJ0132CAF050	PANTALON DAMA CAFE GABARDINA-STRET -MOD 02...	6.000	2321.55
FCJ0175GRI110	JEANS DAMA STONE MEZCLILLA-STRE -MOD 038-SALV	6.000	2314.66
GJH0089VERGRA	BLUSA DAMA JADE SPANDEX -MOD 1805	13.000	2013.79
GCB3156NEG320	PANTALON CABALLERO 10200017 NEGRO GABAR -M...	6.000	1983.62
GHJ0491NARMED	VESTIDO DAMA CORAL/HUESO POLIESTE -MOD 17175	9.000	1937.93
FCJ0175GRI090	JEANS DAMA STONE MEZCLILLA-STRE -MOD 038-SALV	5.000	1917.24
FII0022VAR010	CALCETIN CABALLERO PAQ-3-SURTIDO SPANDE -MO...	16.000	1874.14
GDG0223NEG380	TRAJE CABALLERO NEGRO-RAYAS POL -MOD 22938	4.000	1827.59
FCJ0132BLA090	PANTALON DAMA BLANCO GABARDINA-STR -MOD 02...	5.000	1794.83
GDG02890XF400	TRAJE CABALLERO OXFORD/NEGRO-RAYAS P -MOD ...	3.000	1793.10
FHH0044NEGMED	CHAMARRA CABALLERO NEGRO POLIESTER -MOD 3...	3.000	1743.10
FCG0174NEG010	PLAYERA CABALLERO NEGRO SPANDEX -MOD TERL...	13.000	1738.79

Hemos obtenido una consulta de ventas por producto con unidades vendidas e importe además de ordenarla por el importe de ventas en forma descendente.

En el siguiente ejercicio, haremos un pequeño cambio quitando la tabla de FINV y en su lugar usamos FCLI y cambiaremos ICOD e IDESCR, por CLICOD Y CLINOM, para obtener las ventas sumariadas por cliente, para esto hacemos liga; **LEFT JOIN** con la tabla FCLI

```

SELECT CLICOD, CLINOM,SUM(AICANTF),ROUND(SUM(AICANTF*AIPRECIO),2) IMPORTE
FROM FAXINV
LEFT JOIN FCLI ON FCLI.CLISEQ=FAXINV.CLISEQ
LEFT JOIN FDOC ON FDOC.DSEQ=FAXINV.DSEQ
WHERE DESFACT=1 AND DFECHA>='2018-06-01' AND DFECHA<='2018-06-30'
GROUP BY CLICOD
ORDER BY IMPORTE DESC

```

CLICOD	CLINOM	SUM(AICANTF)	IMPORTE
01	CENTRO	770.000	120355.17
03	INSURGENTES	453.000	81817.24
05	SATELITE	29.000	5218.97
35877	JANET MARTINEZ HUERTA	6.000	2426.72
13938	PATRICIA SANCHEZ CASTILLO	9.000	1593.97
16421	NORMA BEATRIZ FERRERA	5.000	1389.66
15489	OMAR PEREZ MEJIA	3.000	1247.41
11121	ROBERTO GARCIA MORALES	5.000	1219.83
01631	YAZMIN AGUILAR AVILA	4.000	1081.03
01783	YANETH MACEDA MACATIA	6.000	1075.00
08585	RUTH ROCHA GONZALEZ	3.000	1041.38
23061	MARIA RICARDEZ ARIAS	4.000	930.17
28704	LINDA ZUÑIGA OLIVARES	7.000	844.83
37660	ISABEL ORTIZ FALCON	5.000	828.45
55010	CRISTINA GOMEZ TORRES	3.000	796.55
16505	NOI MATUS CARRASCO	5.000	639.66
14077	PATRICIA PEÑA LOPEZ	4.000	593.10
16317	NORMA MARTINEZ LOPEZ	3.000	529.31
16060	ODETH MOSCAIRO TORRES	4.000	527.59

Ligando tablas sin llaves primarias

Hasta este momento, hemos visto ligas entre tablas que hemos denominado LLAVES PRIMARIAS o NATURALES, es importante saber que es posible ligar tablas que no tienen conexión directa.

En algunas situaciones es necesario utilizar alguna referencia que sirva para conectar tablas ya que resulta de gran importancia, el uso de algún campo que sirva de liga lo denominaremos CONEXIONES INDIRECTAS.

Parámetros		Tipos de movimientos		Catálogo de almacenes		Usuarios		Familias	
FAG	AGTNUM	FTIPMV	TICLA	FALMCAT	CATALM	FUSERS	USRSEQ	FFAM	FAMTNUM

Recordemos todas las tablas de soporte con que cuenta Proscái y que no tienen liga de conexión directa, y como hemos mencionado antes, existen campos cuya referencia nos permitirá realizar la liga indirecta.

En los capítulos y ejercicios anteriores hemos visto la estructura de un Query básico con llaves primarias, es decir donde siempre la relación entre dos tablas se hace entre las llaves primarias, que en el caso de **Proscái**, son las secuencias de las mismas, en algunas ocasiones se requiere de relacionar dos tablas donde la relación no está dada por secuencias sino por algún otro campo.

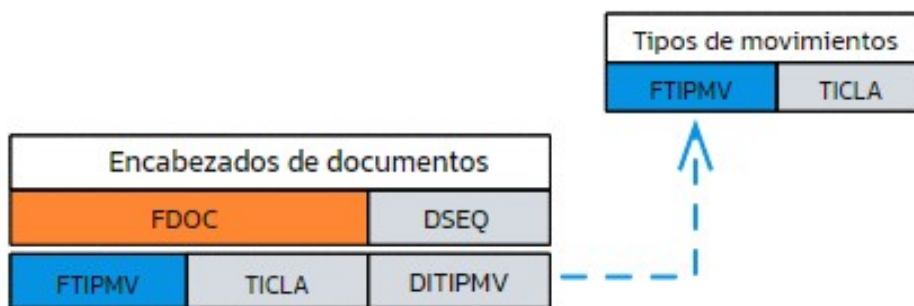
Te recomendamos que de ser necesario vuelvas al capítulo Capítulo 2 y consultes el detalle y descripción de tablas y campos.

Empezaremos con un Query básico donde queremos obtener los documentos generados en el módulo de facturación (DESFACT=1) y queremos saber qué tipo de movimiento es cada uno de ellos, en este caso DITIPMV es el campo que almacena el tipo de movimiento.

```
SELECT CLICOD,CLINOM,DNUM,DCANTF,DITIPMV
FROM FDOC
LEFT JOIN FCLI ON FCLI.CLISEQ=FDOC.CLISEQ
WHERE DESFACT=1
```

CLICOD	CLINOM	DNUM	DCANTF	DITIPMV
01	CENTRO	T01023254	174.00	T
01	CENTRO	T01023255	234.00	T
03	INSURGENTES	T03012184	419.00	T
01	CENTRO	T01023268	343.00	T
01	CENTRO	T01023274	259.00	T
03	INSURGENTES	T03012189	269.00	T
01	CENTRO	T01023280	206.00	T
01	CENTRO	T01023292	912.00	T
01	CENTRO	T01023296	219.00	T
03	INSURGENTES	T03012196	194.00	T
16421	NORMA BEATRIZ FERRERA	T01023306	180.00	T
01	CENTRO	T01023314	359.00	T
01	CENTRO	T01023316	278.00	T
19740	MERCEDES CASTRO CASTRO	T01023327	164.00	T
01	CENTRO	T01023332	54.00	T
01	CENTRO	T01023344	100.00	T
01	CENTRO	T01023355	107.00	T

Podemos ver en el resultado la información y en la última columna el tipo de movimiento, sin embargo no es muy descriptivo, podemos ligar la tabla para obtener la descripción y saber claramente qué tipo de movimiento representa, recordemos que a través del campo DITIPMV podemos realizar una liga indirecta hacia la tabla FTIPMV como lo podemos ver en el siguiente diagrama:



Esta liga indirecta nos permitirá extraer el nombre descriptivo del tipo de movimiento que representa el documento, nuestro query quedaría así:

```
SELECT CLICOD,CLINOM,DNUM,DCANTF,DITIPMV,TIDESCR
FROM FDOC
LEFT JOIN FCLI ON FCLI.CLISEQ=FDOC.CLISEQ
LEFT JOIN FTIPMV ON FTIPMV.TICLA=FDOC.DITIPMV
WHERE DESFACT=1
```

CLICOD	CLINOM	DNUM	DCANTF	DITIPMV	TIDESCR
01	CENTRO	T01023254	174.00	T	TIKET
01	CENTRO	T01023255	234.00	T	TIKET
03	INSURGENTES	T03012184	419.00	T	TIKET
01	CENTRO	T01023268	343.00	T	TIKET
01	CENTRO	T01023274	259.00	T	TIKET
03	INSURGENTES	T03012189	269.00	T	TIKET
01	CENTRO	T01023280	206.00	T	TIKET

Observa que el campo que sirvió como conexión, no es una secuencia

Realicemos más ejercicios usando ligas con conexiones indirectas. Recordemos una referencia puede ser cualquier campo o dato que no necesariamente sea una secuencia.

Ahora queremos obtener los documentos que ha emitido cada tienda pudiendo conocer el nombre además de su número.

```
SELECT DNUM,DCANTF,DFECHA,DALMACEN,CATDESCR
FROM FDOC
LEFT JOIN FALMCAT ON FALMCAT.CATALM=FDOC.DALMACEN
WHERE DESFACT=1 AND DITIPMV='T'
```

DNUM	DCANTF	DFECHA	DALMACEN	CATDESCR
T01023254	174.00	2018-01-01	01	CENTRO
T01023255	234.00	2018-01-01	01	CENTRO
T03012184	419.00	2018-01-01	03	INSURGENTES
T01023268	343.00	2018-01-01	01	CENTRO
T01023274	259.00	2018-01-01	01	CENTRO
T03012189	269.00	2018-01-01	03	INSURGENTES
T01023280	206.00	2018-01-01	01	CENTRO
T01023292	912.00	2018-01-01	01	CENTRO
T01023296	219.00	2018-01-01	01	CENTRO
T03012196	194.00	2018-01-01	03	INSURGENTES
T01023306	180.00	2018-01-01	01	CENTRO
T01023314	359.00	2018-01-01	01	CENTRO
T01023316	278.00	2018-01-01	01	CENTRO
T01023327	164.00	2018-01-01	01	CENTRO
T01023332	54.00	2018-01-01	01	CENTRO

Realicemos ahora un ejercicio para obtener el total de ventas y cuantos documentos hemos generado por cada almacén, pudiendo describir el nombre de la tienda específica.

```
SELECT SUM(1),SUM(DCANTF),DALMACEN,CATDESCR
FROM FDOC
LEFT JOIN FALMCAT ON FALMCAT.CATALM=FDOC.DALMACEN
WHERE DESFACT=1 AND DITIPMV='T'
GROUP BY DALMACEN
```

SUM(1)	SUM(DCANTF)	DALMACEN	CATDESCR
7876	2069457.15	01	CENTRO
5260	1364803.29	03	INSURGENTES
661	196249.89	05	SATELITE

Observa nuevamente que realizamos una conexión hacia tabla FALMCAT usando el campo DALMACEN que típicamente indica el número de tienda, pero hemos “traducido” su número al nombre a través de dicha conexión.

Generemos el siguiente query, para poder obtener una sumatoria de importe y documentos de venta por tienda y día de emisión.

```
SELECT DALMACEN,CATDESCR,DFECHA,SUM(1),SUM(DCANTF)
FROM FDOC
LEFT JOIN FALMCAT ON FALMCAT.CATALM=FDOC.DALMACEN
WHERE DESFACT=1 AND DITIPMV='T'
GROUP BY DALMACEN,DFECHA
```

DALMACEN	CATDESCR	DFECHA	SUM(1)	SUM(DCANTF)
01	CENTRO	2018-01-01	20	5179.00
01	CENTRO	2018-01-02	10	2163.00
01	CENTRO	2018-01-03	17	4256.00
01	CENTRO	2018-01-04	11	2469.00
01	CENTRO	2018-01-05	11	2537.00
01	CENTRO	2018-01-06	12	4974.00
01	CENTRO	2018-01-07	10	3177.00
01	CENTRO	2018-01-08	9	1789.00
01	CENTRO	2018-01-09	7	2383.00
01	CENTRO	2018-01-10	17	3267.00
01	CENTRO	2018-01-11	13	2751.00
01	CENTRO	2018-01-12	17	4157.00
01	CENTRO	2018-01-13	13	2674.00

Si realizamos una pequeña modificación al query anterior, obtenemos las ventas por fechas.

```
SELECT DFECHA,SUM(1),SUM(DCANTF)
FROM FDOC
GROUP BY DFECHA
```

DFECHA	SUM(1)	SUM(DCANTF)
2018-01-01	26	6438.00
2018-01-02	22	4800.00
2018-01-03	25	5914.00
2018-01-04	23	6781.00
2018-01-05	19	3859.31
2018-01-06	18	5858.00
2018-01-07	16	4247.00
2018-01-08	16	2880.00
2018-01-09	9	3551.00
2018-01-10	23	5529.00
2018-01-11	20	4370.00
2018-01-12	28	6221.00
2018-01-13	26	5070.00

Aunque no podemos ver el nombre de la tienda, es importante comparar para entender el beneficio de poder describir información mediante las conexiones de tabla

Vamos a obtener ahora la venta sumada por fecha y tienda, para saber cuántos documentos generó cada tienda y el importe de las ventas.

Podemos notar que el orden de los campos solicitados coincide con el orden de los campos agrupadores, aunque esto no es estrictamente necesario, en una buena práctica ya que le estamos dando al usuario indirectamente el orden para acomodar la información

```
SELECT DFECHA,DALMACEN,CATDESCR,SUM(1),SUM(DCANTF)
FROM FDOC
LEFT JOIN FALMCAT ON FALMCAT.CATALM=FDOC.DALMACEN
WHERE DESFACT=1 AND DITIPMV='T'
GROUP BY DFECHA,DALMACEN
```

DFECHA	DALMACEN	CATDESCR	SUM(1)	SUM(DCANTF)
2018-01-01	01	CENTRO	20	5179.00
2018-01-01	03	INSURGENTES	6	1259.00
2018-01-02	01	CENTRO	10	2163.00
2018-01-02	03	INSURGENTES	9	1273.00
2018-01-02	05	SATELITE	3	1364.00
2018-01-03	01	CENTRO	17	4256.00
2018-01-03	03	INSURGENTES	6	1224.00
2018-01-03	05	SATELITE	2	434.00
2018-01-04	01	CENTRO	11	2469.00
2018-01-04	03	INSURGENTES	9	3743.00
2018-01-04	05	SATELITE	3	569.00
2018-01-05	01	CENTRO	11	2537.00
2018-01-05	03	INSURGENTES	6	816.00
2018-01-05	05	SATELITE	2	506.31
2018-01-06	01	CENTRO	12	4974.00

Dos ligas con llaves indirectas

Al igual que las ligas con llaves primarias, se pueden ligar varias tablas con ligas indirectas, en este caso obtendremos los nombres de los almacenes y la de los cajeros de nuestros tickets de venta del punto de venta.

```
SELECT DALMACEN,CATDESCR,USRCOD,USRNAME,DNUM,DCANTF
FROM FDOC
LEFT JOIN FALMCAT ON FALMCAT.CATALM=FDOC.DALMACEN
LEFT JOIN FUSERS ON FUSERS.USRSEQ=FDOC.DIUSEQ
WHERE DESFACT=1 AND DITIPMV='T'
```

DALMACEN	CATDESCR	USRCOD	USRNAME	DNUM	DCANTF
01	CENTRO	NULL	NULL	T01023254	174.00
01	CENTRO	NULL	NULL	T01023255	234.00
03	INSURGENTES	NULL	NULL	T03012184	419.00
01	CENTRO	C1T1	GUADALUPE RICARDO VERA	T01023268	343.00
01	CENTRO	C1T1	GUADALUPE RICARDO VERA	T01023274	259.00
03	INSURGENTES	NULL	NULL	T03012189	269.00
01	CENTRO	C1T1	GUADALUPE RICARDO VERA	T01023280	206.00
01	CENTRO	NULL	NULL	T01023292	912.00
01	CENTRO	C1T1	GUADALUPE RICARDO VERA	T01023296	219.00
03	INSURGENTES	NULL	NULL	T03012196	194.00
01	CENTRO	C1T1	GUADALUPE RICARDO VERA	T01023306	180.00
01	CENTRO	C1T1	GUADALUPE RICARDO VERA	T01023314	359.00
01	CENTRO	C1T1	GUADALUPE RICARDO VERA	T01023316	278.00
01	CENTRO	C1T1	GUADALUPE RICARDO VERA	T01023327	164.00
01	CENTRO	C1T1	GUADALUPE RICARDO VERA	T01023332	54.00

Vemos en el resultado que existen algunos campos que muestran el valor "NULL", dicho valor se debe entender a que el registro del documento, no cuenta con la referencia necesaria en este caso en el campo DIUSEQ para poder realizar la liga indirecta a la tabla FUSERS por lo tanto, al ejecutar la consulta, el sistema indica "NULL" al no poder buscar un dato que no existe.

Se puede usar el comando "**NOT NULL**" para hacer que el resultado de la consulta no muestre los registros donde no exista resultado que mostrar. Es importante considerar que al usar este comando estamos dejando de ver datos que podrían modificar el resultado de la consulta.

Ahora obtendremos los datos del cajero y vendedor de cada ticket;

```
SELECT USRCOD,USRNAME,DPAR1,AGDESCR,DNUM,DCANTF
FROM FDOC
LEFT JOIN FAG ON FAG.AGTNUM=FDOC.DPAR1
LEFT JOIN FUSERS ON FUSERS.USRSEQ=FDOC.DIUSEQ
WHERE DESFACT=1 AND DITIPMV='T'
```

USRCOD	USRNAME	DPAR1	AGDESCR	DNUM	DCANTF
NULL	NULL	101	NAYELI CHAVEZ ROSAS	T01023254	174.00
NULL	NULL	101	NAYELI CHAVEZ ROSAS	T01023255	234.00
NULL	NULL	1WYF	MIRNA CHACON CUELLAR	T03012184	419.00
C1T1	GUADALUPE RICARDO VERA	1XCW	MARGARITA ROJAS GONZALEZ	T01023268	343.00
C1T1	GUADALUPE RICARDO VERA	1XRN	JOENIA CELORIO GOMEZ	T01023274	259.00
NULL	NULL	1XOI	JULIA SOLANO OJEDA	T03012189	269.00
C1T1	GUADALUPE RICARDO VERA	1024	IVONEE GARCIA SAAVEDRA	T01023280	206.00
NULL	NULL	1XOY	JUANA CRUZ ARRIETA	T01023292	912.00
C1T1	GUADALUPE RICARDO VERA	101	NAYELI CHAVEZ ROSAS	T01023296	219.00
NULL	NULL	1WYF	MIRNA CHACON CUELLAR	T03012196	194.00
C1T1	GUADALUPE RICARDO VERA	1V1R	GUADALUPE PRODUCCIONES AC	T01023306	180.00
C1T1	GUADALUPE RICARDO VERA	1VMF	RAFAEL PEREZ MORALES	T01023314	359.00
C1T1	GUADALUPE RICARDO VERA	1W9Y	YOLANDA MENDOZA LUNA	T01023316	278.00
C1T1	GUADALUPE RICARDO VERA	1XCW	MARGARITA ROJAS GONZALEZ	T01023327	164.00
C1T1	GUADALUPE RICARDO VERA	1V1R	GUADALUPE PRODUCCIONES AC	T01023332	54.00

Agregando un nivel más de conexiones, pediremos obtener la tienda, cajero y vendedor de cada Ticket

```
SELECT CATALM,CATDESCR,USRCOD,USRNAME,DPAR1,AGDESCR,DNUM,DCANTF
FROM FDOC
LEFT JOIN FAG ON FAG.AGTNUM=FDOC.DPAR1
LEFT JOIN FUSERS ON FUSERS.USRSEQ=FDOC.DIUSEQ
LEFT JOIN FALMCAT ON FALMCAT.CATALM=FDOC.DALMACEN
WHERE DESFACT=1 AND DITIPMV='T'
```

CATALM	CATDESCR	USRCOD	USRNAME	DPA...	AGDESCR	DNUM	DCANTF
01	CENTRO	NULL	NULL	101	NAYELI CHAVEZ ROSAS	T01023254	174.00
01	CENTRO	NULL	NULL	101	NAYELI CHAVEZ ROSAS	T01023255	234.00
03	INSURGENTES	NULL	NULL	1WYF	MIRNA CHACON CUELLAR	T03012184	419.00
01	CENTRO	C1T1	GUADALUPE RICARDO VERA	1XCW	MARGARITA ROJAS GONZALEZ	T01023268	343.00
01	CENTRO	C1T1	GUADALUPE RICARDO VERA	1XRN	JOENIA CELORIO GOMEZ	T01023274	259.00
03	INSURGENTES	NULL	NULL	1XOI	JULIA SOLANO OJEDA	T03012189	269.00
01	CENTRO	C1T1	GUADALUPE RICARDO VERA	1024	IVONEE GARCIA SAAVEDRA	T01023280	206.00
01	CENTRO	NULL	NULL	1XOY	JUANA CRUZ ARRIETA	T01023292	912.00
01	CENTRO	C1T1	GUADALUPE RICARDO VERA	101	NAYELI CHAVEZ ROSAS	T01023296	219.00
03	INSURGENTES	NULL	NULL	1WYF	MIRNA CHACON CUELLAR	T03012196	194.00
01	CENTRO	C1T1	GUADALUPE RICARDO VERA	1V1R	GUADALUPE PRODUCCIONES AC	T01023306	180.00
01	CENTRO	C1T1	GUADALUPE RICARDO VERA	1VMF	RAFAEL PEREZ MORALES	T01023314	359.00
01	CENTRO	C1T1	GUADALUPE RICARDO VERA	1W9Y	YOLANDA MENDOZA LUNA	T01023316	278.00
01	CENTRO	C1T1	GUADALUPE RICARDO VERA	1XCW	MARGARITA ROJAS GONZALEZ	T01023327	164.00

Siguiendo el ejemplo del query anterior agruparemos por tienda, cajero y vendedor

```

SELECT CATDESCR,USRNAME,AGDESCR,SUM(1),SUM(DCANTF)
FROM FDOC
LEFT JOIN FAG ON FAG.AGTNUM=FDOC.DPAR1
LEFT JOIN FUSERS ON FUSERS.USRSEQ=FDOC.DIUSEQ
LEFT JOIN FALMCAT ON FALMCAT.CATALM=FDOC.DALMACEN
WHERE DESFACT=1 AND DITIPMV='T'
GROUP BY CATDESCR,USRNAME,AGDESCR

```

CATDESCR	USRNAME	AGDESCR	SUM(1)	SUM(DCANTF)
CENTRO	GUADALUPE RICARDO VERA	HUGO VADAPILLETA ARANDA	60	17391.00
CENTRO	GUADALUPE RICARDO VERA	IMELDA ALVAREZ SAAVEDRA	21	6680.10
CENTRO	GUADALUPE RICARDO VERA	IRAIS SILENCIO ROSAS	4	1084.00
CENTRO	GUADALUPE RICARDO VERA	IRENE JEREZANO BARRIOS	28	7435.00
CENTRO	GUADALUPE RICARDO VERA	ISRAEL HERNANDEZ VIVEROS	26	5930.10
CENTRO	GUADALUPE RICARDO VERA	IVAN RONZON MONTIEL	37	9999.00
CENTRO	GUADALUPE RICARDO VERA	IVONEE GARCIA SAAVEDRA	199	54261.70
CENTRO	GUADALUPE RICARDO VERA	IVONNE MARTINEZ SALAZAR	3	560.00
CENTRO	GUADALUPE RICARDO VERA	JANET LUGO SOSA	6	1127.00
CENTRO	GUADALUPE RICARDO VERA	JANETH LOZANO PANTIGO	2	936.00
CENTRO	GUADALUPE RICARDO VERA	JAVIER CONDE BLANCO	5	2254.00
CENTRO	GUADALUPE RICARDO VERA	JENNY DOMINGUEZ VELASCO	1	258.00
CENTRO	GUADALUPE RICARDO VERA	JOENIA CELORIO GOMEZ	23	4860.00
CENTRO	GUADALUPE RICARDO VERA	JORGE VALLEJO CERVANTES	95	26155.10

Podemos ver claramente que el USRNAME que corresponde al cajero es el mismo, pero AGDESCR que corresponde al vendedor es diferente, además la columna de suma de documentos, nos permite ver quien participo en más ventas y por lo tanto genero el mayor importe.

Hasta este punto, hemos podido ver que las ligas hacia tablas, se pueden hacer de muchas maneras y que todos los comandos se pueden utilizar de forma combinada, enriqueciendo nuestras consultas y análisis de información.

Varias conexiones a una misma tabla

En este capítulo, abordaremos el tema de realizar una conexión o relación hacia una misma tabla, es decir, usaremos 2 datos diferentes para usarlos como llaves secundarias y poder conectar hacia la misma tabla 2 veces.

Veamos el ejemplo que más claramente nos permitirá entender la conexión múltiple a una misma tabla.

Cuando hablamos del catálogo de productos y sus clasificadores normalmente lo podemos visualizar como esta imagen lo muestra:

The screenshot shows a software interface titled 'Catálogo de productos'. It includes a search form with the following fields: 'Código' (AGF10020XF080), 'Descripción' (ZAPATO CERRADO CABALLERO RATA PIEL NOBUCK -MOD 722 0XF 080), and 'Unidad' (AE). Below the form is a section titled 'Selección de parámetros' containing a table with the following data:

SELECCIONADOS			
F.	Fam	Cod.	Descripción
1	PROVEEDORE	AGF	MUHAMMAD ALI
2	DEPARTAMENTO	ZCA	ZAPATOS CABALLERO
3	LINEA	ZCE	ZAPATO CERRADO
4	SUB LINEA	1CU	CERRADO URBANO
5	TEMPORADA	TA	TODO EL AÑO

Podemos ver claramente los clasificadores que describen determinadas características del producto, en la pantalla observamos en columnas, la que indica el número de clasificador, la que describe el criterio o familia, el código del clasificador y la descripción del mismo, este último dato, reside en la tabla FFAM y para poder describir cada uno de los renglones, debe hacer una conexión múltiple a la misma tabla.

Vamos a hacer un query de la tabla FINV (catálogo de productos) mostrando sus clasificadores o familias de una manera simple, es decir sin descripciones de cada familia

```
SELECT ICOD, IDESCR, IFAM2, IFAM3  
FROM FINV
```

ICOD	IDESCR	IFAM2	IFAM3
FGF0012	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	3RIN
FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	3RIN
FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	3RIN
FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	3RIN
FGF0012VARXGR	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	3RIN
FJB0307AZU070	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	3RPA
FJB0307AZU090	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	3RPA
FJB0307AZU110	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	3RPA
FJB0307AZU130	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	3RPA
FJB0307AZU150	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	3RPA
FJB0311BLA300	PANTALON CABALLERO ARENA GABARDINA -MOD P...	2RCA	3RPA
FJB0311BLA330	PANTALON CABALLERO ARENA GABARDINA -MOD P...	2RCA	3RPA
FJB0311BLA360	PANTALON CABALLERO ARENA GABARDINA -MOD P...	2RCA	3RPA

Ahora agregaremos la descripción, en este caso a la familia 2 ligando la tabla FFAM

```
SELECT ICOD,IDESCR, IFAM2,FAMDESCR
FROM FINV
LEFT JOIN FFAM ON FFAM.FAMTNUM=FINV.IFAM2
```

ICOD	IDESCR	IFAM2	FAMDESCR
FGF0012	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	ROPA CABALLERO
FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	ROPA CABALLERO
FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	ROPA CABALLERO
FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	ROPA CABALLERO
FGF0012VARXGR	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	ROPA CABALLERO
FJB0307AZU070	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	ROPA DAMA
FJB0307AZU090	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	ROPA DAMA
FJB0307AZU110	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	ROPA DAMA
FJB0307AZU130	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	ROPA DAMA
FJB0307AZU150	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	ROPA DAMA
FJB0311BLA300	PANTALON CABALLERO ARENA GABARDINA -MOD P...	2RCA	ROPA CABALLERO
FJB0311BLA330	PANTALON CABALLERO ARENA GABARDINA -MOD P...	2RCA	ROPA CABALLERO

Hemos utilizado el campo IFAM2 como llave secundaria para conectar a la tabla FFAM y buscar el clasificador correspondiente y así poder traducir el departamento al que corresponde el producto.

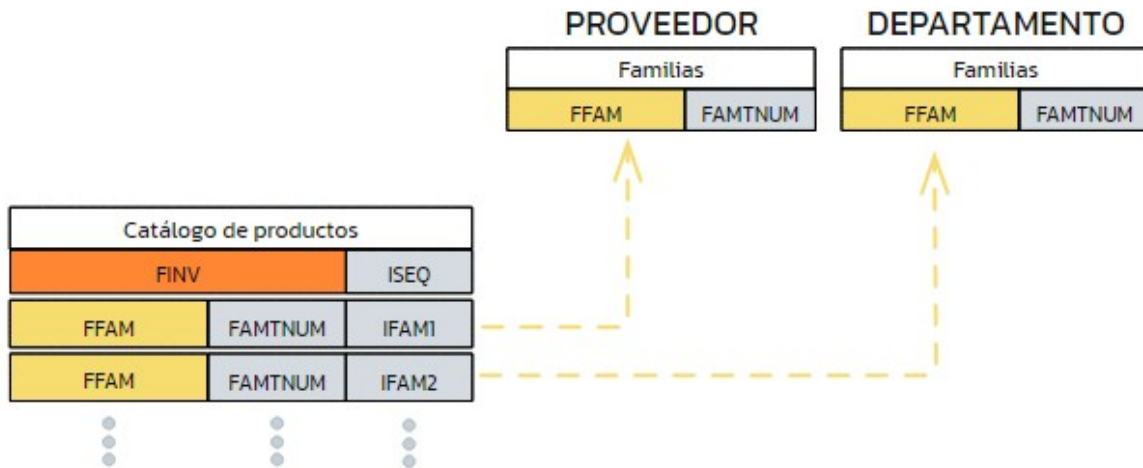
Vamos ahora a usar un alias de tabla en nuestro Query, para ello vamos a darle un nombre a la tabla FFAM, en este caso departamento y cada que usemos un campo de esta tabla, vamos a anteponer el ALIAS seguido de un “.”

```
SELECT ICOD,IDESCR, IFAM2,FAMDESCR
FROM FINV
LEFT JOIN FFAM AS DEPARTAMENTO ON DEPARTAMENTO.FAMTNUM=FINV.IFAM2
```

ICOD	IDESCR	IFAM2	FAMDESCR
FGF0012	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	ROPA CABALLERO
FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	ROPA CABALLERO
FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	ROPA CABALLERO
FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	ROPA CABALLERO
FGF0012VARXGR	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	ROPA CABALLERO
FJB0307AZU070	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	ROPA DAMA
FJB0307AZU090	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	ROPA DAMA
FJB0307AZU110	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	ROPA DAMA
FJB0307AZU130	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	ROPA DAMA
FJB0307AZU150	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	ROPA DAMA
FJB0311BLA300	PANTALON CABALLERO ARENA GABARDINA -MOD P...	2RCA	ROPA CABALLERO
FJB0311BLA330	PANTALON CABALLERO ARENA GABARDINA -MOD P...	2RCA	ROPA CABALLERO
FJB0311BLA360	PANTALON CABALLERO ARENA GABARDINA -MOD P...	2RCA	ROPA CABALLERO
FJB0311BLA400	PANTALON CABALLERO ARENA GABARDINA -MOD P...	2RCA	ROPA CABALLERO

Hasta ahora hemos obtenido por separado en 2 formas diferentes la descripción del departamento, lo mismo podemos hacer para obtener la descripción del clasificador 3 que corresponde a la línea. En el ejercicio anterior donde generamos un alias, pensemos que estamos generando no una relación a una tabla si no que estamos generando una tabla virtual nueva.

Imaginemos que en nuestro diagrama de tablas existe una tabla para cada clasificador quedaría así:



El nombre que se asigna a la tabla mediante el comando “as” virtualiza tantas veces como sea necesaria la tabla original.

Ahora podemos generar un query ligando los campos de IFAM2 e IFAM3 a 2 tablas virtuales. Es un poco lo mismo de los queries anteriores, realicemos el siguiente ejercicio:

```
SELECT ICOD, IDESCR, IFAM2, DEPARTAMENTO.FAMDESCR, IFAM3, LINEA.FAMDESCR
FROM FINV
LEFT JOIN FFAM AS DEPARTAMENTO ON DEPARTAMENTO.FAMTNUM=FINV.IFAM2
LEFT JOIN FFAM AS LINEA ON LINEA.FAMTNUM=FINV.IFAM3
```

ICOD	IDESCR	IFAM2	FAMDESCR	IFAM3	FAMDESCR
FGF0012	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	ROPA CABALLERO	3RIN	INTERIOR
FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	ROPA CABALLERO	3RIN	INTERIOR
FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	ROPA CABALLERO	3RIN	INTERIOR
FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	ROPA CABALLERO	3RIN	INTERIOR
FGF0012VARXGR	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	2RCA	ROPA CABALLERO	3RIN	INTERIOR
FJB0307AZU070	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	ROPA DAMA	3RPA	PANTALON
FJB0307AZU090	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	ROPA DAMA	3RPA	PANTALON
FJB0307AZU110	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	ROPA DAMA	3RPA	PANTALON
FJB0307AZU130	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	ROPA DAMA	3RPA	PANTALON
FJB0307AZU150	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	2RDA	ROPA DAMA	3RPA	PANTALON
FJB0311BLA300	PANTALON CABALLERO ARENA GABARDINA -MOD P...	2RCA	ROPA CABALLERO	3RPA	PANTALON
FJB0311BLA330	PANTALON CABALLERO ARENA GABARDINA -MOD P...	2RCA	ROPA CABALLERO	3RPA	PANTALON

Como podemos observar, cada una de las ligas a la tabla FFAM utilizando un alias, permite conectar a la misma de forma simultánea para cada dato requerido.

De esta forma hemos utilizado un campo como llave secundaria y hemos creado un alias para poder conectar la misma tabla y obtener la descripción de cada clasificador.

Para reafirmar los ejemplos de conexión a tablas con campos que no son llave primaria, pero que sirven para ligar información entre tablas, ahora realicemos un ejemplo de la tabla FENS donde se guardan los ensambles o fórmulas de producción.

```
SELECT PAPA.ICOD,PAPA.IDESCR,HIJO.ICOD,HIJO.IDESCR,ECANT
FROM FENS
LEFT JOIN FINV AS PAPA ON PAPA.ICOD=FENS.EPRO
LEFT JOIN FINV AS HIJO ON HIJO.ICOD=FENS.EART
```

ICOD	IDESCR	ICOD	IDESCR	ECANT
FFG0179	SWEATER CABALLERO JASPE/OXFORD ALGODON -...	AHILOLA	HILO LANA	15.000000
FFG0179	SWEATER CABALLERO JASPE/OXFORD ALGODON -...	BRESQAA	RESORTE DE LIGA PARA CUELLO	7.000000
FFG0179	SWEATER CABALLERO JASPE/OXFORD ALGODON -...	EETIQ01	ETIQUETA DE LAVADO	1.000000
FFG0179	SWEATER CABALLERO JASPE/OXFORD ALGODON -...	EETIQ02	ETIQUETA DE CARTON COLGABLE	1.000000

Observa que en este ejemplo estamos asignando el alias PAPA al producto que se fabrica e HIJO al producto que es parte del consumo, de esta forma podemos claramente saber que producto se fabrica y cual se consume.

Ahora realicemos otro ejemplo de conexión simultánea a la misma tabla para obtener información que nos diga por un lado el producto PADRE o RAIZ y por otro el producto HIJO o VARIANTE, con el siguiente query podremos obtenerlo:

```
SELECT PAPA.ICOD,PAPA.IDESCR,PAPA.ILISTA1,HIJO.ICOD,HIJO.IDESCR,HIJO.ILISTA1
FROM FINV AS HIJO
LEFT JOIN FINV AS PAPA ON PAPA.ICOD=MID(HIJO.ICOD,1,7)
```

ICOD	IDESCR	ILISTA1	ICOD	IDESCR	ILISTA1
FGF0012	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	24.0000	FGF0012	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	24.0000
FGF0012	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	24.0000	FGF0012VARCHI	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	24.0000
FGF0012	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	24.0000	FGF0012VARMED	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	24.0000
FGF0012	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	24.0000	FGF0012VARGRA	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	24.0000
FGF0012	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	24.0000	FGF0012VARGR	TRUZA CABALLERO SURTIDO ALGODON -MOD 39315...	24.0000
FJB0307	JEANS DAMA STRE-BLANCO MEZCLILL -MOD ATRAC...	419.0000	FJB0307AZU070	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	419.0000
FJB0307	JEANS DAMA STRE-BLANCO MEZCLILL -MOD ATRAC...	419.0000	FJB0307AZU090	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	419.0000
FJB0307	JEANS DAMA STRE-BLANCO MEZCLILL -MOD ATRAC...	419.0000	FJB0307AZU110	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	419.0000
FJB0307	JEANS DAMA STRE-BLANCO MEZCLILL -MOD ATRAC...	419.0000	FJB0307AZU130	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	419.0000
FJB0307	JEANS DAMA STRE-BLANCO MEZCLILL -MOD ATRAC...	419.0000	FJB0307AZU150	JEANS DAMA STRE-BLUE MEZC -MOD ATRACTION	419.0000

Observa con detenimiento este query, como puedes ver, dentro del comando **LEFT JOIN** hemos agregado el uso de un comparador y de la función de extracción de datos, para poder indicar como debe encontrar en la tabla PAPA solo los 7 primeros dígitos del producto HIJO.

Continuando con el ejemplo anterior, podemos usar el mismo query para comparar datos como el precio o cualquier otro entre PAPA e HIJO y encontrar los que sean diferentes, simplemente agregando el comando **WHERE** y agregar la condición que busque solo los diferentes observa:

```
SELECT PAPA.ICOD,PAPA.IDESCR,PAPA.ILISTA1,HIJO.ICOD,HIJO.IDESCR,HIJO.ILISTA1
FROM FINV AS HIJO
LEFT JOIN FINV AS PAPA ON PAPA.ICOD=MID(HIJO.ICOD,1,7)
WHERE PAPA.ILISTA1<>HIJO.ILISTA1
```

ICOD	IDESCR	ILISTA1	ICOD	IDESCR	ILISTA1
GFJ0189	PLAYERA CABALLERO HUESO SPANDEX -MOD PEGA...	10.0000	GFJ0189BEI010	PLAYERA CABALLERO ARENA SPANDEX -MOD PEGA...	20.0000
GIC0015	CONTROL DAMA MORADO SPANDEX -MOD 801	70.0000	GIC0015BEICHI	CONTROL DAMA NUDE SPANDEX -MOD 801	159.0000
GIC0015	CONTROL DAMA MORADO SPANDEX -MOD 801	70.0000	GIC0015BEIMED	CONTROL DAMA NUDE SPANDEX -MOD 801	159.0000
GIC0015	CONTROL DAMA MORADO SPANDEX -MOD 801	70.0000	GIC0015BEIGRA	CONTROL DAMA NUDE SPANDEX -MOD 801	159.0000
GIC0015	CONTROL DAMA MORADO SPANDEX -MOD 801	70.0000	GIC0015BEIXGR	CONTROL DAMA NUDE SPANDEX -MOD 801	159.0000
FJB0616	JEANS DAMA MAX-320 MEZCLILLA-ST -MOD YESS	554.0000	FJB0616MAR050	PANTALON DAMA NEGRO GABARDINA -MOD YESS	439.0000
FJB0616	JEANS DAMA MAX-320 MEZCLILLA-ST -MOD YESS	554.0000	FJB0616MAR090	PANTALON DAMA NEGRO GABARDINA -MOD YESS	439.0000
FJB0616	JEANS DAMA MAX-320 MEZCLILLA-ST -MOD YESS	554.0000	FJB0616MAR130	PANTALON DAMA NEGRO GABARDINA -MOD YESS	439.0000
FJB0616	JEANS DAMA MAX-320 MEZCLILLA-ST -MOD YESS	554.0000	FJB0616NEG050	PANTALON DAMA NEGRO GABARDINA -MOD YESS	439.0000
FJB0616	JEANS DAMA MAX-320 MEZCLILLA-ST -MOD YESS	554.0000	FJB0616NEG090	PANTALON DAMA NEGRO GABARDINA -MOD YESS	439.0000
FJB0616	JEANS DAMA MAX-320 MEZCLILLA-ST -MOD YESS	554.0000	FJB0616NEG130	PANTALON DAMA NEGRO GABARDINA -MOD YESS	439.0000

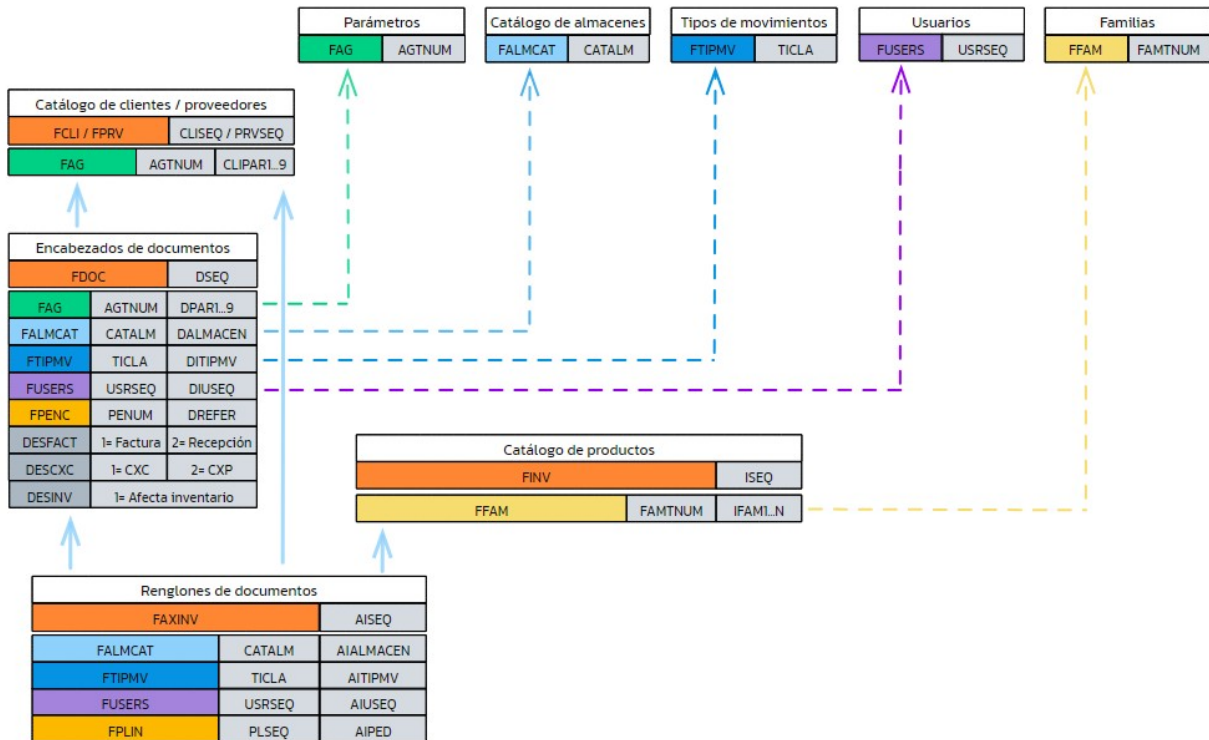
Claramente podemos ver, que en las primeras 3 columnas tenemos la información del PAPA y las siguientes 3 información del HIJO y encontramos las diferencias de precio entre la RAIZ y su VARIANTE.

Como has podido ver al realizar todos estos ejercicios de conexiones entre tablas algunas con campos llave y otras con llaves secundarias la posibilidad de obtener cualquier dato de cualquier tabla es totalmente viable, solo basta con entender el origen de la tabla inicial y hacia que tablas deberás realizar conexiones.

Creando reportes de ventas

En este capítulo, llevaremos a cabo una serie de ejemplos para obtener reportes o informes de ventas, que nos permitirán reafirmar todos los ejercicios anteriormente realizados y hacer uso y combinación de todos los comandos y funciones que hemos utilizado.

Para poder obtener la información de ventas, recordemos el diagrama que nos muestra las tablas de soporte así como las tablas con relación directa involucradas:



Cualquiera de los dos tipos de conexión que se realice, permitirá usar esa misma conexión para definir filtros o condiciones dentro de una consulta.

Recuerda observar que condiciones aplica, que ordenamiento utiliza, que sumatorias considera y que conexiones realiza cada reporte, de esta manera y con lo que has aprendido podrás crear tus propios reportes o consultas fácilmente.

Reporte de ventas ordenado por cliente

```

SELECT CLICOD,CLINOM,DNUM,DFECHA,DCANTF
FROM FDOC
LEFT JOIN FCLI ON FCLI.CLISEQ=FDOC.CLISEQ
WHERE DESFACT=1 AND DFECHA>='2018-01-01' AND DFECHA<='2019-01-31'
ORDER BY CLICOD

```

CLICOD	CLINOM	DNUM	DFECHA	DCANTF
00121	EMMA GABRIELA RODRIGUEZ COLORADO	T03017718	2018-04-01	120.00
00121	EMMA GABRIELA RODRIGUEZ COLORADO	T03013910	2018-02-01	27.00
00121	EMMA GABRIELA RODRIGUEZ COLORADO	T03031142	2018-09-23	2669.69
00121	EMMA GABRIELA RODRIGUEZ COLORADO	T03038032	2018-12-30	383.00
00121	EMMA GABRIELA RODRIGUEZ COLORADO	T03033597	2018-10-27	-193.00
00121	EMMA GABRIELA RODRIGUEZ COLORADO	T03034630	2018-11-10	270.75
00121	EMMA GABRIELA RODRIGUEZ COLORADO	T03014409	2018-02-12	416.00
00121	EMMA GABRIELA RODRIGUEZ COLORADO	T03035698	2018-11-27	151.00
00121	EMMA GABRIELA RODRIGUEZ COLORADO	T03025542	2018-06-18	208.00
00121	EMMA GABRIELA RODRIGUEZ COLORADO	T03032191	2018-10-07	301.50
00121	EMMA GABRIELA RODRIGUEZ COLORADO	T03033834	2018-11-01	986.00
00121	EMMA GABRIELA RODRIGUEZ COLORADO	T03031961	2018-10-04	340.00
00121	EMMA GABRIELA RODRIGUEZ COLORADO	T03027400	2018-07-21	29.00
00121	EMMA GABRIELA RODRIGUEZ COLORADO	T03034081	2018-11-03	426.80

Ventas agrupadas por cliente y ordenadas de mayor a menor

```

SELECT CLICOD,CLINOM,DNUM,DFECHA,SUM(DCANTF) AS SUMA
FROM FDOC
LEFT JOIN FCLI ON FCLI.CLISEQ=FDOC.CLISEQ
WHERE DESFACT=1 AND DFECHA>='2018-01-01' AND DFECHA<='2019-01-31'
GROUP BY CLICOD
ORDER BY SUMA DESC

```

CLICOD	CLINOM	DNUM	DFECHA	SUMA
32247	JULIA GALLEGOS CAMPOS	T03012419	2018-01-05	2642.00
16317	NORMA MARTINEZ LOPEZ	T03014960	2018-02-23	3653.00
30926	LAURA GONZALEZ OSORIO	T01024581	2018-01-10	4636.00
19740	MERCEDES CASTRO CASTRO	T01023327	2018-01-01	5160.70
01631	YAZMIN AGUILAR AVILA	T01023710	2018-01-04	5821.00
37660	ISABEL ORTIZ FALCON	T03013114	2018-01-17	6128.30
25840	MACRINA VICENTE LUJAN	T01026408	2018-01-26	6943.90
55010	CRISTINA GOMEZ TORRES	T01037066	2018-04-21	9105.00
00339	MARIA CAROLINA SANCHEZ GERARDO	T01023364	2018-01-01	9302.00
23061	MARIA RICARDEZ ARIAS	T03013627	2018-01-26	9500.00
21619	MARIO MARTINEZ SEGURA	T01024254	2018-01-08	9717.30
22328	MARIBEL PALMEQUE VILLAGRAN	T03012260	2018-01-02	9818.00
37167	ISIDRO CAMPUZANO ANGELES	T05004649	2018-01-05	9875.57
30727	LAURA PAEZ ALVAREZ	T01024598	2018-01-11	9886.80
35834	JANET VAZQUEZ LUNA	T05005445	2018-02-02	10142.00
11512	RICARDO CONDE ANDRADE	T01024106	2018-01-06	10611.40

Ventas agrupadas por cliente en orden descendente

En este ejercicio aunque parece muy similar al anterior, estamos condicionando que la longitud de caracteres en el código de clientes sea superior a 3.

```
SELECT CLICOD,CLINOM,SUM(DCANTF) AS SUMA
FROM FDOC
LEFT JOIN FCLI ON FCLI.CLISEQ=FDOC.CLISEQ
WHERE DESFACT=1 AND DFECHA>='2018-01-01' AND DFECHA<='2019-01-31' AND
LENGTH(CLICOD)>3
GROUP BY CLICOD
ORDER BY SUMA DESC
```

CLICOD	CLINOM	SUMA
35877	JANET MARTINEZ HUERTA	40184.00
13938	PATRICIA SANCHEZ CASTILLO	31322.80
16421	NORMA BEATRIZ FERRERA	29786.30
16505	NOI MATUS CARRASCO	28863.80
29963	LETICIA GOMEZ SANCHEZ	24205.00
11121	ROBERTO GARCIA MORALES	23132.90
52672	DOMINGO RAMON BUSTOS	20540.92
06817	SILVIA LOPEZ MONTEROLA	18917.00
07868	SANDY VAZQUEZ ELIZALDE	16965.00
15040	OYUKI BENITEZ ALVARADO	16044.90
40646	GUILLERMINA HERNANDEZ GARCIA	15605.00
35533	JAVIER ALVAREZ CRUZ	14815.70
16060	ODETH MOSCAIRO TORRES	14516.00
28704	LINDA ZUÑIGA OLIVARES	14448.20
14077	PATRICIA PEÑA LOPEZ	13841.00

Ventas por cliente '19740'

Seleccionamos un cliente, para hacer algunos análisis específicos, para ello se coloca el cliente en los filtros. Es importante considerar que para comparar campos alfanuméricos requiero colocarlo entre " (apostrofes) por ejemplo '19740'

```
SELECT CLICOD,CLINOM,DNUM,DFECHA,SUM(DCANTF) AS SUMA
FROM FDOC
LEFT JOIN FCLI ON FCLI.CLISEQ=FDOC.CLISEQ
WHERE DESFACT=1 AND DFECHA>='2018-01-01' AND DFECHA<='2019-01-31' AND
CLICOD='19740'
GROUP BY CLICOD
ORDER BY SUMA DESC
```

CLICOD	CLINOM	DNUM	DFECHA	SUMA
19740	MERCEDES CASTRO CASTRO	T01023327	2018-01-01	5160.70

Ventas desglosadas por cliente '19740'

Con solo cambiar de tabla base, en este caso usaremos FAXINV, agregamos campos, tablas, se hacen las relaciones correspondientes, y recuerda que podemos relacionar FDOC sin tener ningún campo de esa tabla, en el select, solo lo utilizamos para poder filtrar datos de dicha tabla

```
SELECT CLICOD,CLINOM,ICOD,IDESCR,SUM(AICANTF) AS SUMA
FROM FAXINV
LEFT JOIN FCLI ON FCLI.CLISEQ=faxinv.CLISEQ
LEFT JOIN FDOC ON FDOC.DSEQ=FAXINV.DSEQ
LEFT JOIN FINV ON FINV.ISEQ=FAXINV.ISEQ
WHERE DESFACT=1 AND DITIPMV='T' AND DFECHA>='2018-01-01' AND DFECHA<='2019-01-31' AND CLICOD='19740'
GROUP BY ICOD
ORDER BY SUMA DESC
```

CLICOD	CLINOM	ICOD	IDESCR	SUMA
19740	MERCEDES CASTRO CASTRO	FCJ0132NEG070	PANTALON DAMA NEGRO GABARDINA-STRE -MOD 0...	1.000
19740	MERCEDES CASTRO CASTRO	HBB0022NEGEMED	SACO CABALLERO NEGRO GABARDINA -MOD 44210...	1.000
19740	MERCEDES CASTRO CASTRO	GFA0096LIL130	JEANS DAMA INDIGO MEZCLILL -MOD 2395	1.000
19740	MERCEDES CASTRO CASTRO	HFB0018AZUMED	CAMISA CABALLERO AZUL-RAYAS ALGODON -MOD 1...	1.000
19740	MERCEDES CASTRO CASTRO	GCB3213NAR110	PANTALON DAMA 2014172 VINO GABARDI -MOD BELLA	1.000
19740	MERCEDES CASTRO CASTRO	HAG1264ROJMED	CHALECO DAMA ROJO POLIESTER -MOD IM1212LV	1.000
19740	MERCEDES CASTRO CASTRO	FEJ01008EI070	PANTALON DAMA BEIGE GABARDINA-STRE -MOD GR...	1.000
19740	MERCEDES CASTRO CASTRO	FJB1642GRI090	JEANS DAMA RING-STONE MEZCLILLA -MOD MILAH	1.000
19740	MERCEDES CASTRO CASTRO	HJ0038NEGGRA	SACO DAMA NEGRO SPANDEX -MOD JR2229	1.000
19740	MERCEDES CASTRO CASTRO	HCD0290NAR010	PLAYERA DAMA NEGRO SPANDEX -MOD 674	1.000
19740	MERCEDES CASTRO CASTRO	FFG01610XFMED	SWEATER CABALLERO NEGRO/OXFORD ACRILIC -M...	1.000
19740	MERCEDES CASTRO CASTRO	FGH0706ROSGRA	BLUSA DAMA PALO-DE-ROSA RAYON -MOD 7439	1.000
19740	MERCEDES CASTRO CASTRO	FJB1525MAR050	JEANS DAMA RING-STONE MEZCLILLA -MOD DEMI M...	1.000
19740	MERCEDES CASTRO CASTRO	FJC0103AMA010	SWEATER DAMA ORO ACRILICO -MOD 4-0075	1.000
19740	MERCEDES CASTRO CASTRO	GCH0489CAFMED	CHAMARRA DAMA TABACO VINI-PIEL -MOD 19167	1.000

Hemos obtenido el desglose de ventas que muestra los productos y cantidades que el cliente ha adquirido en un rango de fechas determinado.

Ventas por clientes '19740' resumizado por producto PAPA

Hacemos un pequeño cambio y tenemos las ventas a nivel papá (raíz), esto es agregando MID(ICOD,1,7)

```
SELECT CLICOD,CLINOM,MID(ICOD,1,7) AS PAPA,IDESCR,SUM(AICANTF) AS SUMA
FROM FAXINV
LEFT JOIN FCLI ON FCLI.CLISEQ=faxinv.CLISEQ
LEFT JOIN FDOC ON FDOC.DSEQ=FAXINV.DSEQ
LEFT JOIN FINV ON FINV.ISEQ=FAXINV.ISEQ
WHERE DESFACT=1 AND DITIPMV='T' AND DFECHA>='2018-01-01' AND DFECHA<='2019-01-31' AND CLICOD='19740'
GROUP BY PAPA
ORDER BY SUMA DESC
```

CLICOD	CLINOM	PAPA	IDESCR	SUMA
19740	MERCEDES CASTRO CASTRO	FCJ0132	PANTALON DAMA NEGRO GABARDINA-STRE -MOD 0...	1.000
19740	MERCEDES CASTRO CASTRO	GFA0096	JEANS DAMA INDIGO MEZCLILL -MOD 2395	1.000
19740	MERCEDES CASTRO CASTRO	FEJ0100	PANTALON DAMA BEIGE GABARDINA-STRE -MOD GR...	1.000
19740	MERCEDES CASTRO CASTRO	HCJ0038	SACO DAMA NEGRO SPANDEX -MOD JR2229	1.000
19740	MERCEDES CASTRO CASTRO	HCD0290	PLAYERA DAMA NEGRO SPANDEX -MOD 674	1.000
19740	MERCEDES CASTRO CASTRO	FGH0706	BLUSA DAMA PALO-DE-ROSA RAYON -MOD 7439	1.000
19740	MERCEDES CASTRO CASTRO	FJB1525	JEANS DAMA RING-STONE MEZCLILLA -MOD DEMI M...	1.000
19740	MERCEDES CASTRO CASTRO	GCH0489	CHAMARRA DAMA TABACO VINI-PIEL -MOD 19167	1.000
19740	MERCEDES CASTRO CASTRO	HII0638	CHALECO CABALLERO MARINO POLIESTER -MOD 31...	1.000
19740	MERCEDES CASTRO CASTRO	HFC0004	CHAMARRA CABALLERO AZUL POLIESTER -MOD PN...	1.000
19740	MERCEDES CASTRO CASTRO	FCG0157	PLAYERA CABALLERO AMARILLO/CAFE ALGODO -MO...	1.000
19740	MERCEDES CASTRO CASTRO	HBB0022	SACO CABALLERO NEGRO GABARDINA -MOD 44210-...	1.000
19740	MERCEDES CASTRO CASTRO	HFB0018	CAMISA CABALLERO AZUL-RAYAS ALGODON -MOD 1...	1.000
19740	MERCEDES CASTRO CASTRO	GCB3213	PANTALON DAMA 2014172 VINO GABARDI -MOD BELLA	1.000

Ventas por cliente '19740' desglosadas a nivel color

Ahora generamos la consulta a nivel modelo color, observa como cambiamos la longitud de los caracteres a obtener a través de la función mid()

```
SELECT CLICOD,CLINOM,MID(ICOD,1,10) AS PAPA,IDESCR,SUM(AICANTF) AS SUMA
FROM FAXINV
LEFT JOIN FCLI ON FCLI.CLISEQ=faxinv.CLISEQ
LEFT JOIN FDOC ON FDOC.DSEQ=FAXINV.DSEQ
LEFT JOIN FINV ON FINV.ISEQ=FAXINV.ISEQ
WHERE DESFACT=1 AND DITIPMV='T' AND DFECHA>='2018-01-01' AND DFECHA<='2019-01-31' AND CLICOD='19740'
GROUP BY PAPA
ORDER BY SUMA DESC
```

CLICOD	CLINOM	PAPA	IDESCR	SUMA
19740	MERCEDES CASTRO CASTRO	FCJ0132NEG	PANTALON DAMA NEGRO GABARDINA-STRE -MOD 0...	1.000
19740	MERCEDES CASTRO CASTRO	GFA0096LIL	JEANS DAMA INDIGO MEZCLILL -MOD 2395	1.000
19740	MERCEDES CASTRO CASTRO	FEJ0100BEI	PANTALON DAMA BEIGE GABARDINA-STRE -MOD GR...	1.000
19740	MERCEDES CASTRO CASTRO	HCJ0038NEG	SACO DAMA NEGRO SPANDEX -MOD JR2229	1.000
19740	MERCEDES CASTRO CASTRO	HCD0290NAR	PLAYERA DAMA NEGRO SPANDEX -MOD 674	1.000
19740	MERCEDES CASTRO CASTRO	FGH0706ROS	BLUSA DAMA PALO-DE-ROSA RAYON -MOD 7439	1.000
19740	MERCEDES CASTRO CASTRO	FJB1525MAR	JEANS DAMA RING-STONE MEZCLILLA -MOD DEMI M...	1.000
19740	MERCEDES CASTRO CASTRO	GCH0489CAF	CHAMARRA DAMA TABACO VINI-PIEL -MOD 19167	1.000
19740	MERCEDES CASTRO CASTRO	HII0638MAR	CHALECO CABALLERO MARINO POLIESTER -MOD 31...	1.000
19740	MERCEDES CASTRO CASTRO	HFC0004AZU	CHAMARRA CABALLERO AZUL POLIESTER -MOD PN...	1.000
19740	MERCEDES CASTRO CASTRO	FCG0157AMA	PLAYERA CABALLERO AMARILLO/CAFE ALGODO -MO...	1.000
19740	MERCEDES CASTRO CASTRO	HBB0022NEG	SACO CABALLERO NEGRO GABARDINA -MOD 44210-...	1.000
19740	MERCEDES CASTRO CASTRO	HF80018AZU	CAMISA CABALLERO AZUL-RAYAS ALGODON -MOD 1...	1.000
19740	MERCEDES CASTRO CASTRO	GCB3213NAR	PANTALON DAMA 2014172 VINO GABARDI -MOD BELLA	1.000

Ventas totales por mes y cliente

En este ejercicio vamos a incorporar la función: DATE_FORMAT, la cual nos permite obtener el mes, el día o año de un campo de fecha.

Para traducir una fecha a solo un mes, debemos usar esta función de la siguiente manera: DATE_FORMAT(DFECHA,'%m') estamos pidiendo que al campo DFECHA, le aplique un formato visual de fecha y el símbolo '%m' significa mes, así presentará solo el mes.

```
SELECT CLICOD,CLINOM,DATE_FORMAT(DFECHA,'%m') AS MES,SUM(AICANTF) AS SUMA
FROM FAXINV
LEFT JOIN FCLI ON FCLI.CLISEQ=faxinv.CLISEQ
LEFT JOIN FDOC ON FDOC.DSEQ=FAXINV.DSEQ
WHERE DESFACT=1 AND DITIPMV='T' AND DFECHA>='2018-01-01' AND DFECHA<='2019-12-31' AND CLICOD='19740'
GROUP BY MES
ORDER BY SUMA DESC
```

CLICOD	CLINOM	MES	SUMA
19740	MERCEDES CASTRO CASTRO	05	5.000
19740	MERCEDES CASTRO CASTRO	04	4.000
19740	MERCEDES CASTRO CASTRO	08	4.000
19740	MERCEDES CASTRO CASTRO	03	4.000
19740	MERCEDES CASTRO CASTRO	02	1.000

Observa que la columna mes además de asignarle un alias presenta el mes del año en que fue realizada la venta.

Ventas por cliente '19740' modelo y color incluyendo departamento

Vamos a obtener la venta por producto a nivel modelo y color y agregaremos el departamento al que pertenece cada producto.

```
SELECT CLICOD,CLINOM,MID(ICOD,1,10) AS MODELO,IDESCR,FAMDESCR,SUM(AICANTF) AS SUMA
FROM FAXINV
LEFT JOIN FCLI ON FCLI.CLISEQ=faxinv.CLISEQ
LEFT JOIN FDOC ON FDOC.DSEQ=FAXINV.DSEQ
LEFT JOIN FINV on FINV.ISEQ=FAXINV.ISEQ
LEFT JOIN FFAM on FFAM.FAMTNUM=FINV.IFAM2
WHERE DESFACT=1 AND DITIPMV='T' AND DFECHA>='2018-01-01' AND DFECHA<='2019-12-31' AND
CLICOD='19740'
GROUP BY MODELO
```

CLICOD	CLINOM	MODELO	IDESCR	FAMDESCR	SUMA
19740	MERCEDES CASTRO CASTRO	FBJ0355VER	CAMISA CABALLERO MENTA/MARINO-CUADROS -MO...	ROPA CABALLERO	-1.000
19740	MERCEDES CASTRO CASTRO	FCG0157AMA	PLAYERA CABALLERO AMARILLO/CAFE ALGODO -MO...	ROPA CABALLERO	1.000
19740	MERCEDES CASTRO CASTRO	FCJ0107GRI	JEANS DAMA STONE MEZCLILLA-STRE -MOD 007-SALV	ROPA DAMA	-1.000
19740	MERCEDES CASTRO CASTRO	FCJ0132NEG	PANTALON DAMA NEGRO GABARDINA-STRE -MOD 0...	ROPA DAMA	1.000
19740	MERCEDES CASTRO CASTRO	FCJ0176LIL	JEANS DAMA INDIGO MEZCLILLA-STR -MOD 039-SALV	ROPA DAMA	0.000
19740	MERCEDES CASTRO CASTRO	FEJ0100BEI	PANTALON DAMA BEIGE GABARDINA-STRE -MOD GR...	ROPA DAMA	1.000
19740	MERCEDES CASTRO CASTRO	FFG01610XF	SWEATER CABALLERO NEGRO/OXFORD ACRILIC -M...	ROPA CABALLERO	1.000
19740	MERCEDES CASTRO CASTRO	FFG0054GRI	PANTALON CABALLERO LADRILLO GABARDINA-S -M...	ROPA CABALLERO	0.000
19740	MERCEDES CASTRO CASTRO	FGH0706ROS	BLUSA DAMA PALO-DE-ROSA RAYON -MOD 7439	ROPA DAMA	1.000
19740	MERCEDES CASTRO CASTRO	FJB1525MAR	JEANS DAMA RING-STONE MEZCLILLA -MOD DEMI M...	ROPA DAMA	1.000
19740	MERCEDES CASTRO CASTRO	FJB1642GRI	JEANS DAMA RING-STONE MEZCLILLA -MOD MILAH	ROPA DAMA	1.000

Es importante recordar que cuando realizamos conexiones, debemos considerar que alguna de la información necesaria para realizar la liga puede depender de otra tabla u otra conexión, así que el orden en la definición del comando **LEFT JOIN** va de acuerdo a la jerarquía entre tablas. En este ejercicio ligamos primero FINV para poder obtener el dato que está en el campo IFAM2 que a su vez servirá para acceder a FFAM y obtener la descripción.

Si se invierte el orden de conexión, aparecerá un error ya que no será posible encontrar el dato a buscar y obtener la descripción.

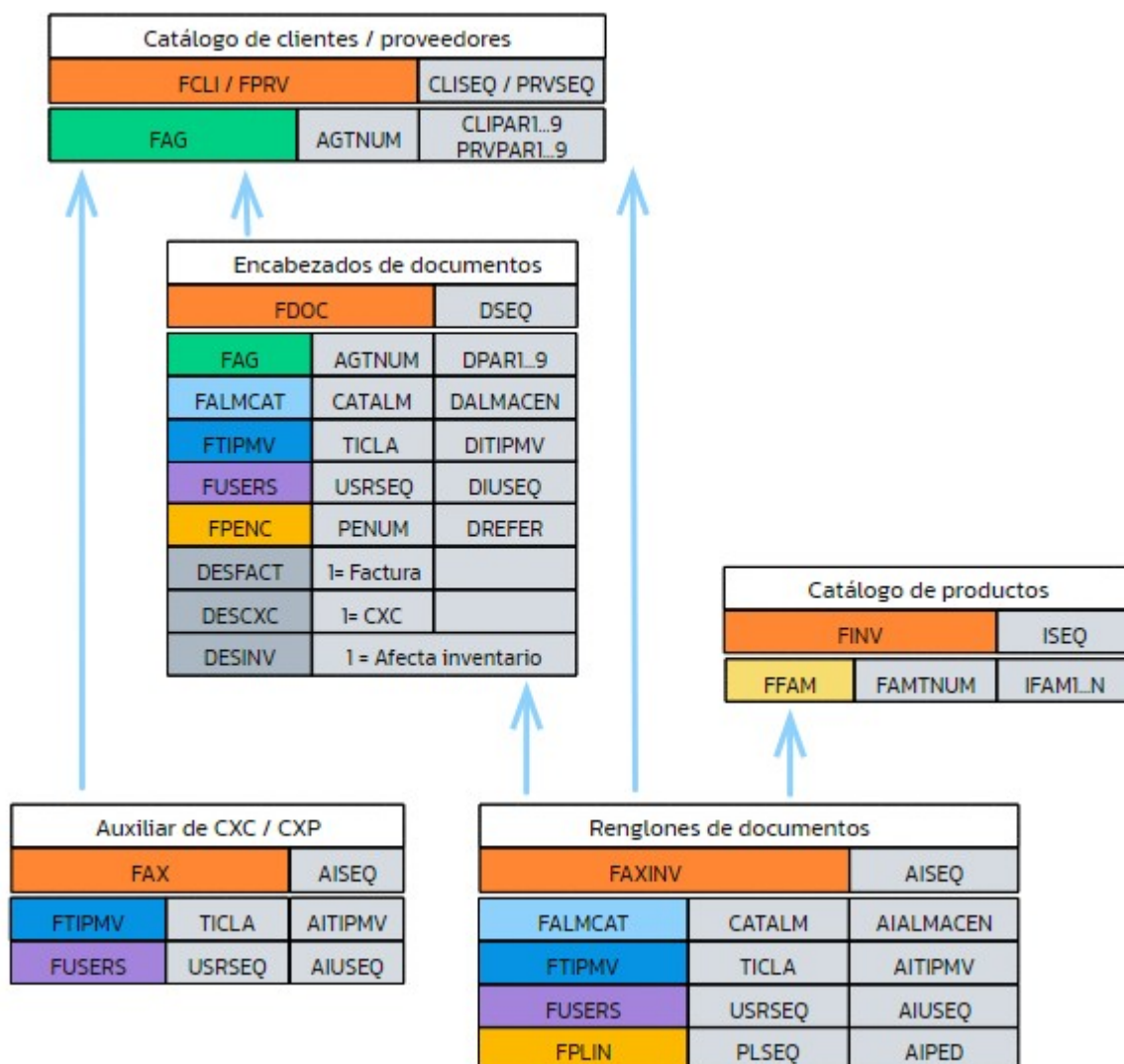
Ventas por mes y departamento

Para obtener las ventas globales por departamento que incluya todos los clientes.

```
SELECT DATE_FORMAT(DFECHA,'%m') as MES,FAMDESCR,SUM(AICANTF) AS SUMA
FROM FAXINV
LEFT JOIN FDOC ON FDOC.DSEQ=FAXINV.DSEQ
LEFT JOIN FINV ON FINV.ISEQ=FAXINV.ISEQ
LEFT JOIN FFAM ON FFAM.FAMTNUM=FINV.IFAM2
WHERE DESFACT=1 AND DITIPMV='T' AND DFECHA>='2018-01-01'
GROUP BY MES,IFAM2
ORDER BY SUMA DESC
```

mes	FAMDESCR	SUMA
05	ROPA DAMA	2035.000
05	ROPA CABALLERO	1699.000
03	ROPA DAMA	1339.000
04	ROPA DAMA	1019.000
02	ROPA DAMA	916.000
09	ROPA DAMA	850.000
08	ROPA DAMA	820.000
06	ROPA DAMA	814.000
03	ROPA CABALLERO	806.000
12	ROPA DAMA	794.000
10	ROPA DAMA	759.000
04	ROPA CABALLERO	710.000
07	ROPA DAMA	661.000
11	ROPA DAMA	650.000
06	ROPA CABALLERO	534.000
01	ROPA DAMA	523.000
11	ROPA CABALLERO	514.000
02	ROPA CABALLERO	482.000

Hasta este punto, todos los ejercicios que hemos realizado nos han permitido practicar y reforzar el desarrollo de consultas, todos los ejercicios han sido basados en obtener información de ventas y vinculada hacia clientes. Sin embargo como podemos ver en el diagrama siguiente, la tabla FPRV se encuentra al mismo nivel que FCLI y comparte las mismas ligas hacia las tablas inferiores:



Y a su vez recordando también el grupo de banderas que existen principalmente dentro de la tabla FDOC:

DNUM	DESFACT	DESCXC	DESINV	DCANCELADA	DSTATUSCFD	DITIPMV	CLISEQ	PRVSEQ
F00000485	1	1	1	0	3	F	3357	0
R01478	0	2	1	0	-3	R	0	4715

Que nos ayudan a determinar el tipo de operación dentro del sistema, con lo cual podemos entender que la misma cantidad de consultas se podrían obtener con el solo hecho de cambiar las llaves de conexión directa involucradas y las condiciones definidas modificando las variables que sirven como condicionante como DESFACT=2 y la información que obtendremos nos permitirá enfocar el análisis hacia el proceso de compras y cuentas por pagar.

CAPÍTULO 5

HERRAMIENTAS ADICIONALES

CUBO Proscai

Ahora que conoces la estructura de la Base de datos, sus tablas, campos y relaciones, así como la forma de formular sentencias para obtener información, puedes enriquecer tu análisis, haciendo uso de CUBO **Proscai**.

CUBO **Proscai**, es una poderosa herramienta de análisis multidimensional para la exploración de tu información, de forma gráfica, sencilla, rápida y confiable.

Podrás extraer información prácticamente de cualquier área (módulo, campo) de la base de datos a través del manejo de sentencias SQL y lo mejor de todo, es la velocidad en la que obtienes dicha información.

Nuestro objetivo es dotar a las empresas de una herramienta que ayude a conformar una estrategia Business Intelligence capaz de rentabilizar la información disponible. Para ello, nos basamos en el desarrollo de soluciones de Inteligencia de Negocio.

CUBO **Proscai** SQL, como herramienta para el análisis multidimensional, constituye un elemento clave para el área directiva y gerencial al permitir un análisis dinámico e interactivo, y ofrece resultados ágiles y precisos sobre grandes volúmenes de información en tiempos muy cortos.

Este análisis multidimensional se basa en la obtención de diferentes vistas y resultados a partir de un conjunto de datos.

Puedes diseñar las vistas en función de un objetivo. Todos los datos disponibles se pueden “cruzar”, “ordenar”, “filtrar”, “resumir”, “detallar”, “combinar”, “agrupar”.

		OTOÑO-INVERNO 2015			OTOÑO-INVERNO 2016			OTOÑO-INVERNO 2017		
ALMACEN	LINEA	CANTIDAD	PRECIO	COSTO	CANTIDAD	PRECIO	COSTO	CANTIDAD	PRECIO	CC
02	MERCANCIA DE LINEA	1.00	128.45	60.00				2.00	360.34	
	MERCANCIA DE REBAJA	15.00	1,669.83	1,014.00				3.00	4.30	
	Total	16.00	1,798.28	1,074.00				5.00	364.65	
03	MERCANCIA DE LINEA									
	MERCANCIA DE REBAJA	20.00	2,312.07	1,404.00						
	Total	20.00	2,312.07	1,404.00						
04	MERCANCIA DE LINEA	2.00	256.90	120.00				1.00	25.00	
	MERCANCIA DE REBAJA	6.00	770.69	468.00				2.00	969.83	
	Total	8.00	1,027.59	588.00				3.00	994.83	
06	MERCANCIA DE LINEA	3.00	385.34	180.00						
	MERCANCIA DE REBAJA									
	Total	3.00	385.34	180.00						
08	MERCANCIA DE LINEA	3.00	385.34	180.00				4.00	1,401.72	
	MERCANCIA DE REBAJA	19.00	2,312.07	1,404.00						
	Total	22.00	2,697.41	1,584.00				4.00	1,401.72	
09	MERCANCIA DE LINEA									
	MERCANCIA DE REBAJA	3.00	385.34	234.00	4.00	688.21	435.84	17.00	3,885.69	
	Total	3.00	385.34	234.00	4.00	688.21	435.84	17.00	3,885.69	

Esta herramienta permite que solamente el personal autorizado tenga acceso.

ALMACEN	CODIGO	DESCRIPCION	TIPO	GENERO	TEMPORADA
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
02	T1111383GRM05	ABRIGO DM ALASKA IN10 NGO 11	I	I	I
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
03	T1111389AZY07	ABRIGO DM ALONDRA IN10 AZX 05	ABRIGO	MERCANCIA DE DAMA	OTOÑO-INVIerno 2007
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
04	T1111389NGO05	ABRIGO DM ALONDRA IN10 RJX 07	ACCESORIOS	MERCANCIA DE NIÑA	OTOÑO-INVIerno 2008
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
05	T1111394NGO09	ABRIGO DM ALONDRA IN10 RJX 11	BERMUDAS	MONEDERO ELECTRONICO	OTOÑO-INVIerno 2009
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
08	T1111394NGO11	ABRIGO DM AMBAR IN09 GRM 05	BLUSA		OTOÑO-INVIerno 2010
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
09	T1111394PJA05	ABRIGO DM ANITA IN10 CFO 05	BLUSONES		PRIMAVERA-VERANO 2008
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	T1111394PJA09	ABRIGO DM ANITA IN10 ROJ 07	BOLSOS		PRIMAVERA-VERANO 2009
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	T1111394PJA11	ABRIGO DM BERLIN IN10 MIRO 05	BUFANDA		PRIMAVERA-VERANO 2010
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	T1111396AZX05	ABRIGO DM BERLIN IN10 ROJ 07	CHALECOS		PRIMAVERA-VERANO 2011
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	T1111396RJX07	ABRIGO DM DUBLIN IN10 FIU 05	CHAMARRAS		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	T1111396RJX11	ABRIGO DM DUBLIN IN10 GRO 07	DEPTO. DE SUPERVISION		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	T1111397NGO11	ABRIGO DM DUBLIN IN10 MNO 11	FALDAS		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
19	T1111398PJA07	ABRIGO DM ELOISAIN10 CFO 09	GABARDINAS		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
24	T1111399CFO07	ABRIGO DM ELOISAIN10 NGO 07	MALLON		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
25	T1111399NGO05	ABRIGO DM ESPAÑA IN10 PJA 07	PALCIO DE HIERRO		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
27	T1111400CFO09	ABRIGO DM ISRAEL IN10 NGO 07	PANTALONES		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
28	T1111400NGO07	ABRIGO DM ISRAEL IN10 ROJ 05	PESQUEROS		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
90	T1111403CFO05	ABRIGO DM ISRAEL IN10 ROJ 09	PLAYERAS		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
91	T1111403ROJ07	ABRIGO DM ISRAEL IN10 ROJ 11	SACOS		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	T1111404MRO05	ABRIGO DM MATI IN10 MRO 05	SWEATER		

El nivel de profundidad en el análisis de los datos no tiene límites, dependerá únicamente de los conocimientos en SQL, ya que Proscái permite acceder a cualquier tabla.

Podrás revisar de manera sencilla, información de interés de tu base de datos.

Cada empresa podrá generar sus propios cubos de acuerdo a las necesidades de cada área, sin depender de terceros.

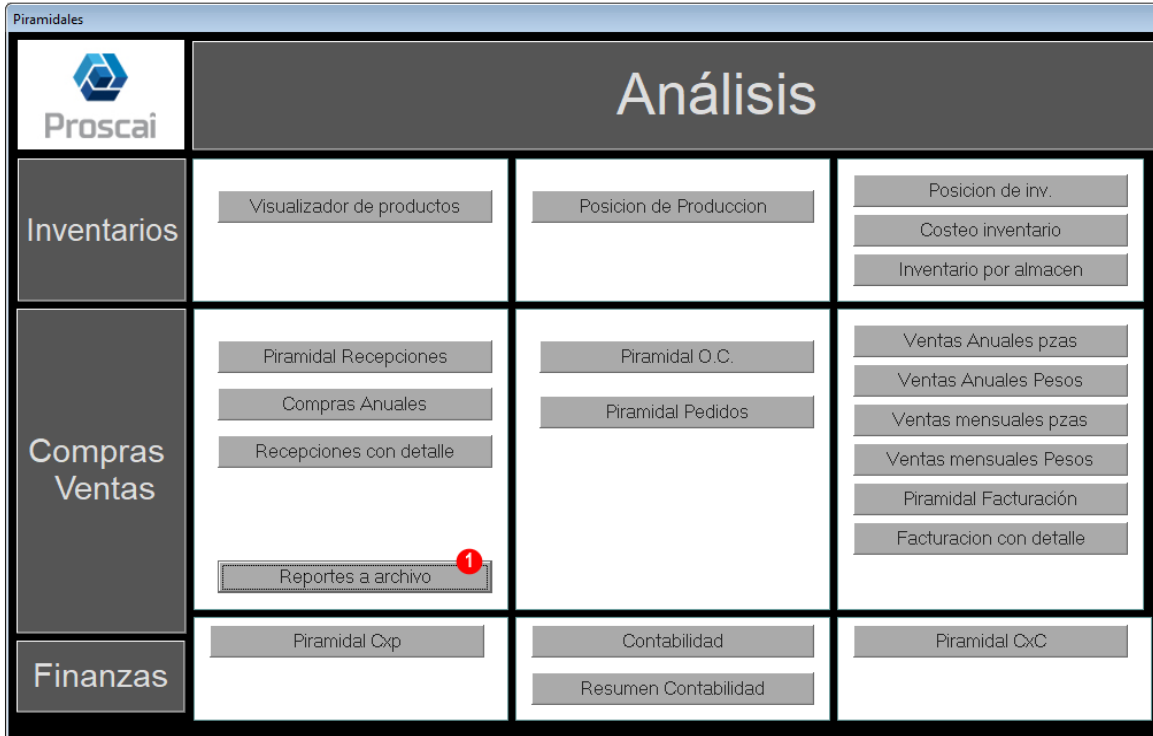
Es con el Cubo donde realmente valoraremos la importancia de los atributos correspondientes a cada tabla, tal como las familias en los productos, los clasificadores en los clientes y proveedores, entre otros, ya que son los principales campos que nos permitirán filtrar y explotar mejor la información para un buen análisis.

Y es así como puede considerarse el flujo de Análisis de información,

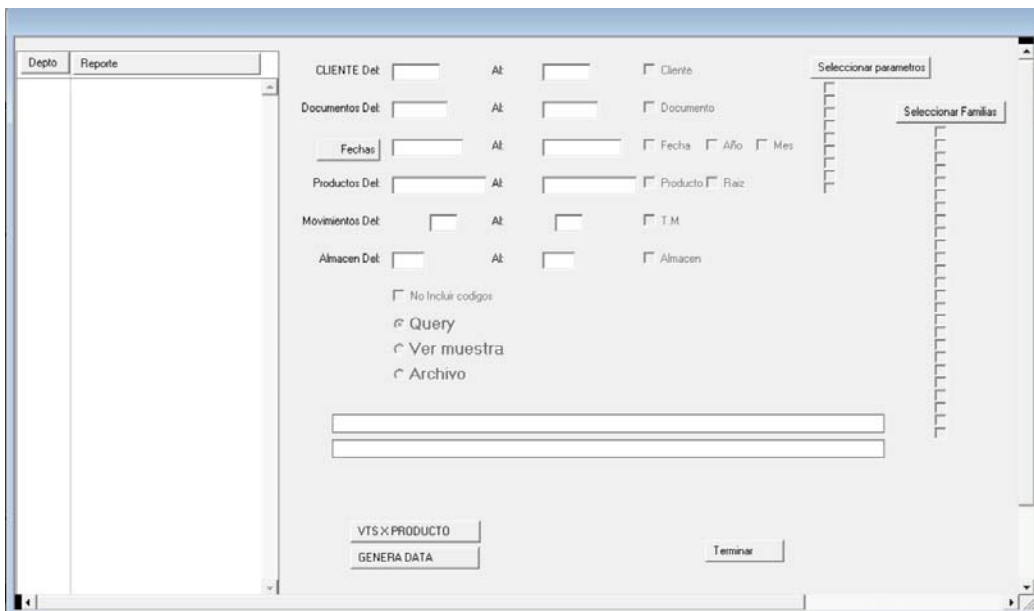
- Se capturan datos en Proscái
- Se consultan y/o extraen mediante consultas (QUERYS)
- Dichos Querys se utilizan para armar cubos
- Con los cubos se tiene mejor análisis y explotación de la información
- Podemos visualizar gráficamente

Generar reportes a archivo y queries desde Proscai

Proscai busca facilitarte el análisis de información y hemos desarrollado una nueva mecánica que permite crear queries con facilidad para analizarlos en MySQL Query Browser, pero también exportarlos a otras aplicaciones, como el bloc de notas o Excel, de forma muy rápida y segura.



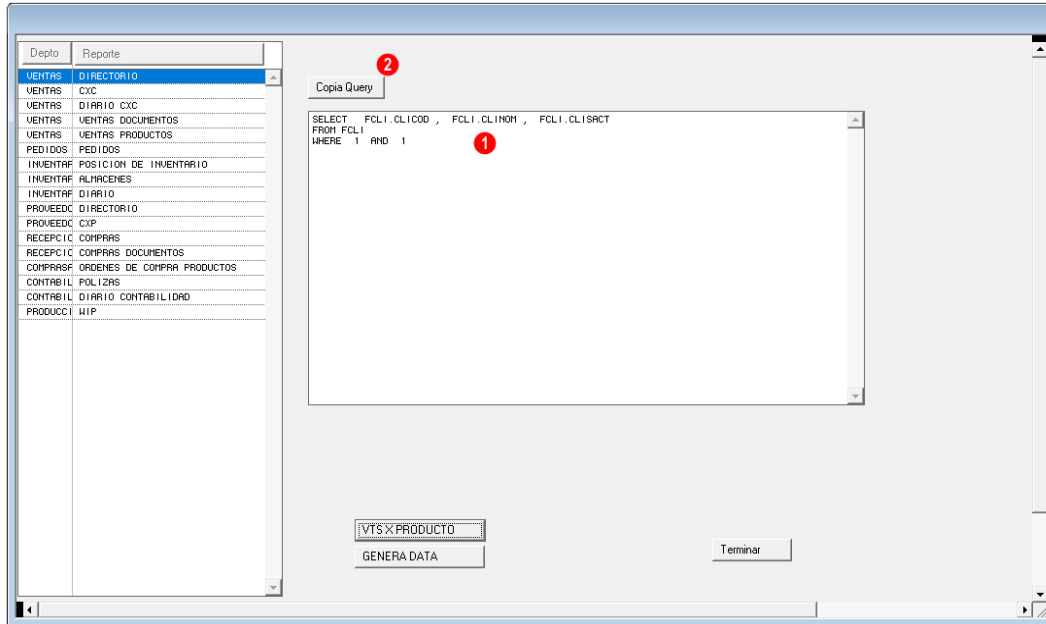
Dentro de Proscai en la sección de consultas SIG, existe el botón que permite acceder a la creación de queries y reportes.



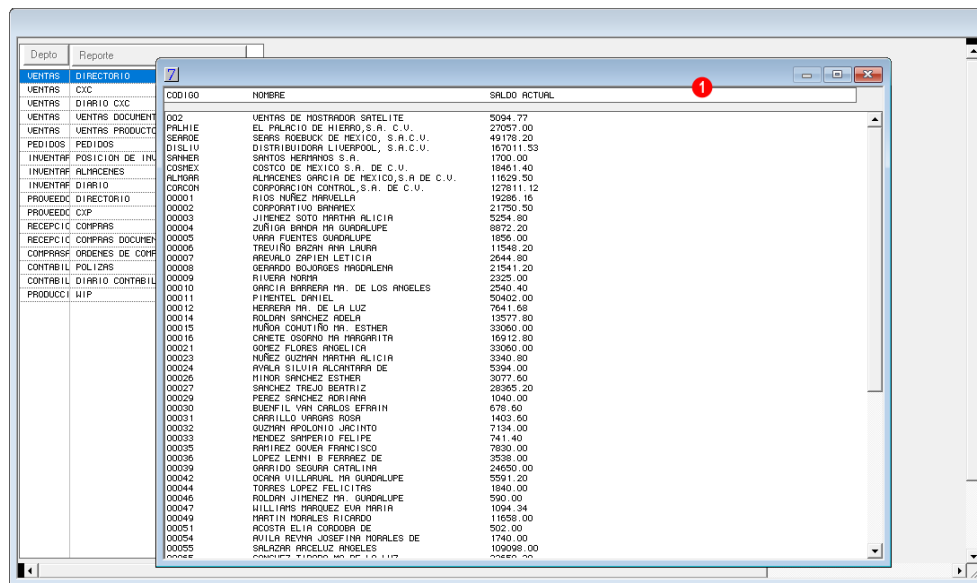
Mediante esta opción podrás crear fácilmente consultas que te generaran la sintaxis necesaria para usarlos dentro del Query Browser o directamente crear archivos planos que podrás abrir en Excel o cualquier otra herramienta.

El objetivo de esta herramienta es ayudar a los usuarios que no estén tan familiarizados con los comandos y sintaxis general de SQL y mediante un ambiente grafico simplemente seleccionar los 'elementos' o campos que se desean obtener y el sistema genere la consulta lista para usarla y obtener la información necesaria.

Creando querys sin preocuparte por la sintaxis de escritura y totalmente funcional.



Usando la opción ver muestra podrás pre-visualizar el archivo que obtendrás y guardarlo como texto



Con la opción archivo, directamente lo guardas como archivo y podrás abrirlo en Excel sin necesidad de copiar y pegar información.

	A	B	C	D
1	CODIGO	NOMBRE	SALDO ACTUAL	
2		2 VENTAS DE MOSTRADOR SATELITE	5094.77	
3	PALHIE	EL PALACIO DE HIERRO, S.A. C.V.	27057	
4	SEAROE	SEARS ROEBUCK DE MEXICO, S.A.C.V.	49178.2	
5	DISLIV	DISTRIBUIDORA LIVERPOOL, S.A.C.V.	167011.53	
6	SANHER	SANTOS HERMANOS S.A.	1700	
7	COSMEX	COSTCO DE MEXICO S.A. DE C.V.	18461.4	
8	ALMGAR	ALMACENES GARCIA DE MEXICO, S.A DE C.V.	11629.5	
9	CORCON	CORPORACION CONTROL, S.A. DE C.V.	127811.12	
10		1 RIOS NUÑEZ MARVELLA	19286.16	
11		2 CORPORATIVO BANAMEX	21750.5	
12		3 JIMENEZ SOTO MARTHA ALICIA	5254.8	
13		4 ZUÑIGA BANDA MA GUADALUPE	8872.2	
14		5 VARA FUENTES GUADALUPE	1856	
15		6 TREVIÑO BAZAN ANA LAURA	11548.2	
16		7 AREVALO ZAPIEN LETICIA	2644.8	
17		8 GERARDO BOJORGES MAGDALENA	21541.2	
18		9 RIVERA NORMA	2325	
19		10 GARCIA BARRERA MA. DE LOS ANGELES	2540.4	
20		11 PIMENTEL DANIEL	50402	

Con estas herramientas estamos seguros que podemos facilitarte el análisis de información que tu empresa necesita.

Agradezco que hayas completado la lectura de este trabajo que pretende incentivar la facilidad con que tú mismo puedes revisar y construir métodos de consulta y análisis y encontrar en tu información aquellos puntos que te ayuden a obtener los mejores resultados.