



We are ready



Formación

Ejercicios Consultas con DataCycle Reporting

APESOFT

Formación Consultas Básicas con DataCycle Reporting

© APE Software Components
08290 Parc Tecnològic del Vallès
Tel: 902 789 981 www.apesoft.com

Índice

El modelo relacional	4
Componentes del modelo relacional.....	4
Bases de Datos.....	4
Entidades.....	4
Registros.....	4
Campos.....	4
Claves (primaria y foránea).....	5
Relaciones.....	5
Integridad referencial.....	6
Vistas.....	6
Teoría de conjuntos.....	6
Lenguaje SQL.....	6
Traducir las necesidades del usuario.....	7
Ejercicios.....	8
Filtrando los datos.....	11
Consultando múltiples tablas.....	13
Sumarizando y agrupando los datos.....	14
Filtrar datos agrupados.....	15
Union.....	16
Subconsultas.....	17
Prácticas	19

El modelo relacional

Anatomía de una Base de Datos Relacional

Componentes del modelo relacional

Bases de Datos

Una base de datos es un almacén en donde se encuentra la información corporativa, ya sea datos maestros como productos, clientes, proveedores ... como información sobre las operaciones realizadas en su empresa: altas, compras, facturas, ventas, albaranes, órdenes de fabricación ...

DataCycle Reporting definirá la conectividad hacia las Bases de Datos corporativas, que engloba la estructura de campos de sus tablas (Entidades) y las relaciones entre esas tablas (joins).

Entidades

Las entidades contienen la información de su corporación formando un conjunto de registros. Distinguimos entre entidades con datos maestros como productos, clientes o proveedores y entidades con las operaciones de su negocio como compras, ventas o órdenes de trabajo.

En DataCycle, una entidad es equivalente a un fichero de AS/400 o a una tabla en Oracle. DataCycle sólo guarda la estructura de campos, sin importar los datos.

Registros

Las entidades están formadas por registros que recogen la información corporativa de forma individual. En las tablas de facturas tendremos un registro con cada factura, en las de productos cada producto estará en un registro y así sucesivamente.

Campos

Cada entidad tiene definida una estructura de campos que determina la información que se almacena y de que tipo será.

En la entidad de productos habrá campos donde almacenar el nombre del producto, su coste, precio y proveedor.

Claves (primaria y foránea)

Cada registro debe ser direccionable de forma única. Es por ello que en toda entidad habrá una serie de campos que determinen de forma unívoca los registros que lo componen.

El dni para los clientes, un código de producto o el número de factura son tres ejemplos de claves primarias.

La clave foránea es un campo que relaciona los registros de una entidad con los de otra entidad. Por ejemplo en una Base de Datos de una biblioteca tendremos entidades de libros y de autores y las claves foráneas relacionarán unos con otros.

Relaciones

Una relación es una unión entre dos entidades tal y como se ha mostrado en el ejemplo de los libros y autores o entre una factura y sus líneas.

La clave primaria de una entidad debe aparecer como clave foránea en la otra entidad.

Se distinguen las siguientes formas de relacionar entidades:

1:1

Por cada registro de una entidad siempre encontraremos un solo registro en la otra entidad. Este caso nos lo encontramos cuando la información sobre una entidad es muy elevada y se trocea, por ejemplo podemos tener una entidad con información administrativa de los clientes y otra con la información comercial de los clientes. Por tanto por cada registro en una entidad habrá un único registro en la otra.

1:N

Es el caso más habitual. Por cada registro en una entidad encontraremos muchos registros relacionados en otra entidad. Por ejemplo un cliente tiene muchos pedidos, un proveedor muchos productos, una provincia muchas provincias ...

N:M

Esta relación se establece entre aquellas entidades en el que registros de ambas pueden estar relacionados con múltiples registro de la otra entidad. Un ejemplo sería tener la entidad de Alumnos y Asignaturas en donde un alumno puede estar matriculado a varias asignaturas y una asignatura tendrá varios alumnos.

Este caso no es soportado por las Bases de Datos relacionales tradicionales, debiendo ser substituidas por una nueva relación con tres tablas de tipo 1:N M:1.

Un caso particular sería el 0:M por ejemplo cuando un vendedor aún no ha hecho ventas o un alumno se ha dado de alta pero aún no se ha matriculado en ningún curso.

Integridad referencial

La coherencia de la Base de Datos se consigue si todas las relaciones entre las entidades están correctamente informadas. Por ejemplo no puede existir un pedido a un cliente que aún no haya sido introducido en el sistema.

Para asegurar la Integridad Referencial se debe adoptar determinadas políticas al realizar operaciones de modificación y borrado de campos clave y registros. Las posibilidades son las siguientes:

Borrado en cascada: En caso de borrar un registro deben borrarse todos los registros relacionados. Por ejemplo si se elimina una asignatura se deben eliminar la información sobre la matriculación a la misma.

Modificación en cascada: En caso de modificar la clave primaria se deberán modificar la clave foránea relacionada en todos aquellos registros en los que aparezca.

Se impide el borrado y modificación: mientras exista registros relacionados no se permite el borrado o eliminación de los registros principales.

Vistas

Una Base de Datos y sus entidades pueden contener información sensible y privilegiada. Una vista es una forma de enmascarar la información, mostrando tan solo un subconjunto de ella a las personas que desee.

Teoría de conjuntos

Todas las operaciones de consulta realizadas sobre las entidades se basan en la teoría de conjuntos.

En toda consulta se determina los campos que componen la respuesta. En conjunto de campos devueltos corresponderá a la respuesta buscada o podrá ser manipulado mediante uniones, intersecciones o diferencias para la obtención de nuevos resultados.

Lenguaje SQL

El lenguaje SQL es el que permite interrogar las Bases de Datos y resolver las consultas.

Aunque SQL es un lenguaje con vocación universal y estándar, han aparecido numerosas versiones y cada Base de Datos dispone de la suya propia. Las diferencias entre ellas son mínimas pero deben ser tenidas en cuenta al diseñar sus consultas. Posteriormente se muestra algunas de las diferencias existentes entre ellas.

Los ejemplos que se muestran a continuación están escritos en SQL de MS Access.

Traducir las necesidades del usuario

Una de las partes más complejas consiste en comprender el lenguaje en el que se expresa las necesidades de consulta de las Bases de Datos. Normalmente son expresadas por personal que no conoce las características informáticas ni la forma de interna de guardar los datos y que normalmente solo expresan el enunciado del problema. Por ejemplo:

Me han pedido que haga un estudio sobre los productos deficitarios

Que categorías son más rentables para la empresa

En que delegación se prevé problemas en el suministro de los pedidos

Sobre este aspecto poca ayuda podemos dar. Solo la experiencia y el conocimiento particular de usuarios y de la empresa pueden servir de ayuda.

Ejercicios

Consultas simples sobre una sola tabla

Las siguientes consultas trabajan seleccionando campos de una sola tabla de la Base de Datos Neptuno.

Ejemplo 01-01

En que ciudad viven mis clientes

```
SELECT Ciudad
FROM [Clientes]
```

Nota 1: También es válido el siguiente formato:

```
SELECT Clientes.Ciudad
FROM [Clientes]
```

Nota 2: Si se desea tener todos los campos disponibles puede utilizar la siguiente consulta aunque posteriormente no podrá seleccionar un campo concreto para su informe:

```
SELECT *
FROM [Clientes]
```

Ejemplo 01-02

Quiero conocer el nombre y dirección de todos mis empleados

```
SELECT Empleados.Nombre, Empleados.Apellidos, Empleados.Dirección, Empleados.Ciudad,
Empleados.Región, Empleados.CódPostal, Empleados.País
FROM [Empleados]
```

Ejemplo 01-03

Que categorías ofrecemos actualmente

```
SELECT Categorías.Descripción
FROM [Categorías]
```

Ejemplo 01-04

Cuáles son los productos y precios y a que categorías pertenecen

```
SELECT Productos.NombreProducto, Productos.PrecioUnidad, Productos.IdCategoría
FROM [Productos]
```

Nota: tan solo disponemos del identificador de categoría y no de su nombre.

Eliminar duplicados

Ahora veremos como evitar la aparición de información repetida.

Ejemplo 01-05

En que ciudades tengo algún cliente

```
SELECT DISTINCT Clientes.Ciudad  
FROM [Clientes]
```

Nota: En este caso, y para evitar confusiones entre ciudades diferentes pero con el mismo nombre, tiene más sentido la siguiente consulta:

```
SELECT DISTINCT Clientes.País, Clientes.Región, Clientes.Ciudad, Clientes.CódPostal  
FROM [Clientes]
```

Ordenando la información

Veremos ahora como solicitar información comercial ordenada por diferentes campos.

Ejemplo 01-06

Listado de nombres de nuestros empleados, incluyendo su número de teléfono e identificación, y mostrados ordenados por apellido y nombre

```
SELECT Empleados.Apellidos, Empleados.Nombre, Empleados.TelDomicilio, Empleados.IdEmpleado  
FROM [Empleados]  
Order By Empleados.Apellidos, Empleados.Nombre
```

Ejemplo 01-07

Listado de precios de nuestros productos mostrados de más caro a más barato

```
SELECT Productos.NombreProducto, Productos.PrecioUnidad  
FROM [Productos]  
Order By Productos.PrecioUnidad desc
```

Nota: para ver los 4 productos más caros:

```
SELECT top 4 Productos.NombreProducto, Productos.PrecioUnidad  
FROM [Productos]  
Order By Productos.PrecioUnidad desc
```

Selecciones con operaciones simples

Ahora vamos a pedir información que implique la realización de operaciones con los campos existentes.

Ejemplo 01-08

Cual es el valor de inventario de cada producto

```
SELECT Productos.IdProducto, Productos.NombreProducto,  
       Productos.PrecioUnidad * Productos.UnidadesEnExistencia as ValorDeInventario  
FROM [Productos]
```

Ejemplo 01-09

Cuales son los pedidos con peor tiempo de entrega

```
SELECT Pedidos.IdPedido, Pedidos.FechaPedido, Pedidos.FechaEntrega,  
       Pedidos.FechaEntrega - Pedidos.FechaPedido as DiasParaEntrega  
FROM [Pedidos]  
order by Pedidos.FechaEntrega - Pedidos.FechaPedido desc
```

Utilización de funciones

Ejemplo 01-10

Inicial del nombre y apellido de mis empleados

```
SELECT Left(Empleados.Nombre,1) & ' ' & Empleados.Apellidos  
FROM [Empleados]
```

Filtrando los datos

Vamos a introducir el concepto de criterios en la solicitud de información y utilización de parámetros.

Ejemplo 02-01

Quiero ver la fecha de pedido y entrega de los pedidos del cliente ANTON

```
SELECT Pedidos.IdPedido, Pedidos.FechaPedido, Pedidos.FechaEntrega
FROM [Pedidos]
WHERE Pedidos.IdCliente='ANTON'
Order By Pedidos.FechaPedido
```

Nota1: Para aprovechar la consulta en otros informes, se recomienda seleccionar todos los campos:

```
SELECT Pedidos.IdPedido, Pedidos.IdCliente, Pedidos.IdEmpleado, Pedidos.FechaPedido,
Pedidos.FechaEntrega, Pedidos.FechaEnvío, Pedidos.FormaEnvío, Pedidos.Cargo,
Pedidos.Destinatario, Pedidos.DirecciónDestinatario, Pedidos.CiudadDestinatario,
Pedidos.RegiónDestinatario, Pedidos.CódPostalDestinatario, Pedidos.PaísDestinatario
FROM [Pedidos]
WHERE Pedidos.IdCliente='ANTON'
Order By Pedidos.FechaPedido
```

Nota2: Para hacer más portable la consulta, se recomienda la utilización de parámetros (explicados mas adelante):

```
SELECT
Pedidos.IdPedido, Pedidos.IdCliente, Pedidos.IdEmpleado, Pedidos.FechaPedido,
Pedidos.FechaEntrega, Pedidos.FechaEnvío, Pedidos.FormaEnvío, Pedidos.Cargo,
Pedidos.Destinatario, Pedidos.DirecciónDestinatario, Pedidos.CiudadDestinatario,
Pedidos.RegiónDestinatario, Pedidos.CódPostalDestinatario, Pedidos.PaísDestinatario
FROM [Pedidos]
WHERE Pedidos.IdCliente='@PAR(Cliente)'
Order By Pedidos.FechaPedido
```

Nota3: Para hacer aún más portable la consulta, se recomienda no poner criterios de negocio en la consulta y en su lugar añadir un filtro en el proceso. El resultado es:

```
SELECT
Pedidos.IdPedido, Pedidos.IdCliente, Pedidos.IdEmpleado, Pedidos.FechaPedido,
Pedidos.FechaEntrega, Pedidos.FechaEnvío, Pedidos.FormaEnvío, Pedidos.Cargo,
Pedidos.Destinatario, Pedidos.DirecciónDestinatario, Pedidos.CiudadDestinatario,
Pedidos.RegiónDestinatario, Pedidos.CódPostalDestinatario, Pedidos.PaísDestinatario
FROM [Pedidos]
Order By Pedidos.FechaPedido
```

Y añadir el siguiente filtro:

```
Pedidos.IdCliente='@PAR(Cliente)'
```

Ejemplo 02-02

Quiero ver la lista en orden alfabético de nuestros productos que contengan 'pan'

```
SELECT Productos.NombreProducto
FROM [Productos]
WHERE Productos.NombreProducto like '%pan%'
Order by Productos.NombreProducto
```

Nota: Recuerde la utilización de parámetros:

```
SELECT Productos.NombreProducto
FROM [Productos]
WHERE Productos.NombreProducto like '%@PAR(Producto)%'
Order by Productos.NombreProducto
```

Ejemplo 02-03

Lista de los proveedores de Italia o España

```
SELECT Proveedores.NombreCompañía, Proveedores.País, Proveedores.Ciudad
FROM [Proveedores]
WHERE Proveedores.País in ('Italia', 'España')
```

Nota: si quisiéramos la lista de los proveedores que no son de Italia o España:

```
SELECT Proveedores.NombreCompañía, Proveedores.País, Proveedores.Ciudad
FROM [Proveedores]
WHERE Proveedores.País not in ('Italia', 'España')
```

Ejemplo 02-04

Quiero saber que productos tienen un precio entre 80 y 300 dólares

```
SELECT Productos.IdProducto, Productos.NombreProducto, Productos.PrecioUnidad
FROM [Productos]
WHERE Productos.PrecioUnidad between 80 and 300
```

Ejemplo 02-05

Pedidos cuyo plazo de entrega superan las 2 semanas

```
SELECT Pedidos.IdPedido, Pedidos.FechaPedido, Pedidos.FechaEntrega
FROM [Pedidos]
WHERE Pedidos.FechaEntrega - Pedidos.FechaPedido >= 14
```

Ejemplo 02-06

Productos de la categoría 2, 4 o 8 o cuyo precio sea igual o superior a 100\$

```
SELECT Productos.NombreProducto, Productos.IdCategoría, Productos.PrecioUnidad
FROM [Productos]
WHERE Productos.IdCategoría in (2, 4, 8) OR Productos.PrecioUnidad >= 100
```

Nota: Compliquemos un poco la consulta: Necesito los productos de la categoría 2, 4 o 8 con precio superior a 40 o los productos con precio superior a 100 independientemente de su categoría.

```
SELECT Productos.NombreProducto, Productos.IdCategoría, Productos.PrecioUnidad
FROM [Productos]
WHERE (Productos.IdCategoría in (2, 4, 8) AND Productos.PrecioUnidad >= 40) OR Productos.PrecioUnidad >= 100
```

Consultando múltiples tablas

Inner Join

Permite unir tablas mediante los campos que forman la clave principal en una tabla y foránea en la otra. El Inner Join solo devuelve resultados en caso de que un registro de una tabla sea referenciado desde la otra.

Por ejemplo, si tenemos una Base de Datos universitaria con alumnos y asignaturas, pero alguno de los alumnos está matriculado pero aún no está apuntado a ninguna asignatura, o bien la asignatura existe pero nadie se ha apuntado a ella, no aparecerían dichos registros en caso de consultas en las que se relacionasen alumnos y asignaturas.

Ejemplo 03-01

Quiero ver los productos y el nombre de su categoría

```
SELECT Productos.NombreProducto, Categorías.NombreCategoría
FROM [Categorías], [Productos]
WHERE Productos.IdCategoría = Categorías.IdCategoría
Order By Categorías.NombreCategoría
```

Nota: al haber dos tablas es necesario indicar a que tabla pertenecen los campos en caso de ambigüedad

Nota:

Ejemplo 03-02

Necesito conocer todos los pedidos y sus líneas con el nombre del cliente y el nombre de los productos

```
SELECT Pedidos.IdPedido, Clientes.NombreCompañía, Pedidos.FechaPedido,
Pedidos.FechaEntrega, Productos.NombreProducto,
[Detalles de pedidos].PrecioUnidad, [Detalles de pedidos].Cantidad,
[Detalles de pedidos].Cantidad * [Detalles de pedidos].PrecioUnidad
FROM [Pedidos], [Detalles de pedidos], [Productos], [Clientes]
WHERE [Detalles de pedidos].IdProducto = Productos.IdProducto AND
Pedidos.IdPedido = [Detalles de pedidos].IdPedido AND
Pedidos.IdCliente = Clientes.IdCliente
Order By [Detalles de pedidos].IdPedido, [Detalles de pedidos].IdProducto
```

Outer Join

De la misma forma que el Inner Join, relaciona tablas mediante sus claves principal en una tabla y foránea en la otra. Como diferencia con el Inner Join, en este caso, no se discriminan aquellos registros cuya clave de join aún no haya sido informada o para los que no existan registros relacionados en la otra tabla.

Ejemplo 03-04

Que productos nunca han sido pedidos

```
SELECT Productos.NombreProducto
FROM [Productos] left join [Detalles de pedidos]
    on Productos.IdProducto = [Detalles de pedidos].IdProducto
WHERE isnull([Detalles de pedidos].IdPedido)
```

Ejemplo 03-05

Vendedores que aún no han hecho honor a su cargo

```
SELECT Empleados.Nombre, Empleados.Apellidos
FROM [Empleados] left join [Pedidos]
    on Empleados.IdEmpleado = Pedidos.IdEmpleado
WHERE isnull(idpedido)
```

Sumarizando y agrupando los datos

Ejemplo 04-01

Quiero saber cuantos empleados tiene mi empresa

```
SELECT count(*) as TotalEmpleados
FROM [Empleados]
```

Nota: Si quiero saber cuantos de ellos son del Reino Unido:

```
SELECT count(*) as TotalEmpleados
FROM [Empleados]
WHERE [Empleados].País = 'Reino Unido'
```

Ejemplo 04-02

Cual es el mayor monto de una línea de pedido

```
SELECT max([Detalles de pedidos].Cantidad*[Detalles de pedidos].PrecioUnidad) as MayorLineaPedido
FROM [Detalles de pedidos]
```

Ejemplo 04-03

Cuanto valen los pedidos realizados de cada producto

```
SELECT [Detalles de pedidos].IdProducto as IdProducto,
    sum([Detalles de pedidos].Cantidad*[Detalles de pedidos].PrecioUnidad) as ValorPedidos
FROM [Detalles de pedidos]
GROUP BY [Detalles de pedidos].IdProducto
```

Ejemplo 04-04

Para valorar las comisiones que debemos abonar a nuestros vendedores por las ventas del mes actual tenemos la siguiente consulta. La comisión representa el 15% de sus ventas

```
SELECT Empleados.Tratamiento, Empleados.Nombre, Empleados.Apellidos,
       sum([Detalles de pedidos].Cantidad*[Detalles de pedidos].PrecioUnidad),
       sum([Detalles de pedidos].Cantidad*[Detalles de pedidos].PrecioUnidad) * 0.15
FROM [Empleados], [Pedidos], [Detalles de pedidos]
WHERE Pedidos.IdEmpleado = Empleados.IdEmpleado AND
      Pedidos.IdPedido = [Detalles de pedidos].IdPedido and
      month(Pedidos.FechaPedido)=month(now()) and
      year(pedidos.fechapedido)=year(now())
Group By Empleados.Tratamiento, Empleados.Nombre, Empleados.Apellidos
Order By Empleados.Apellidos
```

Nota1: porque no aparece ningún valor?

Nota2: vamos a preguntar por las ventas de agosto de 2005:

```
Sustituimos:
month(Pedidos.FechaPedido)=month(now()) and year(pedidos.fechapedido)=year(now())
Por:
month(Pedidos.FechaPedido)=8 and year(pedidos.fechapedido)=2004
```

Filtrar datos agrupados

Hasta ahora habíamos aplicado un criterio a los registros de forma individual. Ahora vamos a aplicar un criterio al resultado de la agrupación de varios registros.

Ejemplo 04-06

Conocer cuáles son los clientes que compran un importe bajo de productos de una categoría:

```
SELECT
    Categorías.NombreCategoría,
    Clientes.NombreCompañía,
    sum([Detalles de pedidos].Cantidad*[Detalles de pedidos].PrecioUnidad ) as VentasTotales
FROM
    Categorías, Clientes, Pedidos, Productos, [Detalles de pedidos]
WHERE
    [Detalles de pedidos].IdProducto = Productos.IdProducto AND
    Pedidos.IdCliente = Clientes.IdCliente AND
    Pedidos.IdPedido = [Detalles de pedidos].IdPedido AND
    Productos.IdCategoría = Categorías.IdCategoría
GROUP BY
    Categorías.NombreCategoría, Clientes.NombreCompañía
having
    sum([Detalles de pedidos].Cantidad*[Detalles de pedidos].PrecioUnidad ) < 3000
```

Union

Los conjuntos que forman parte de la Union deben contener el mismo número de campos y del mismo tipo.

Ejemplo 04-07

Nombre y dirección de todos los clientes y empleados ordenados por código postal

```
SELECT DISTINCT Clientes.NombreCompañía, Clientes.País, Clientes.Ciudad, clientes.códpostal
FROM [Clientes]
union
SELECT DISTINCT Empleados.Nombre & " " & empleados.apellidos, empleados.País, empleados.Ciudad,
empleados.códpostal
FROM [Empleados]
order by códpostal
```

Subconsultas

Ejemplos 05-01

Vendedores que han hecho alguna venta a clientes de Italia

```
SELECT DISTINCT
  Empleados.Apellidos as Apellidos
FROM
  empleados , pedidos
WHERE
  pedidos.idempleado = empleados.idempleado and pedidos.idcliente in
  (SELECT DISTINCT
    Pedidos.IdCliente
  FROM
    Pedidos, Clientes
  WHERE
    Pedidos.IdCliente = Clientes.IdCliente and Clientes.País = "Italia" )
```

Ejemplo 05-02

Proveedores de los 20 productos más vendidos durante este mes.

Igual que en los ejemplos anteriores, sustituimos el mes y año actual por octubre de 2005 para obtener datos válidos.

```
SELECT DISTINCT proveedores.nombrecompañía
FROM (
  SELECT TOP 20
    productos.idproducto, Productos.NombreProducto as prod, productos.idproveedor, sum([Detalles
de pedidos].Cantidad*[Detalles de pedidos].PrecioUnidad ) as imp
  FROM
    [Productos], [Detalles de pedidos], [Pedidos]
  WHERE
    [Detalles de pedidos].IdProducto = Productos.IdProducto AND Pedidos.IdPedido = [Detalles de
pedidos].IdPedido AND year(Pedidos.FechaPedido) = 2005 and month(pedidos.fechapedido)=10
  GROUP BY
    productos.idproducto,Productos.NombreProducto, productos.idproveedor
  ORDER BY
    sum([Detalles de pedidos].Cantidad*[Detalles de pedidos].PrecioUnidad ) desc ) as
  ProductosMasVendidos, Proveedores
WHERE ProductosMasVendidos.IdProveedor = Proveedores.IdProveedor
```

Ejemplo 05-03

Productos que son más caros que cualquiera de la categoría de Mariscos

```
SELECT Productos.NombreProducto, productos.precioUnidad
FROM [Productos]
WHERE Productos.PrecioUnidad > all
  (select productos.precioUnidad
  from [productos],[categorías]
  where Productos.IdCategoría = Categorías.IdCategoría and
  [Categorías].NombreCategoría like "%marisc%")
```

Nota: si se quiere saber aquellos productos que son más caros que algún marisco se sustituye ALL por ANY

Ejemplo 05-04

Cientes que han comprado alguna repostería y algún queso

```
SELECT Clientes.NombreCompañía, Clientes.Dirección, Clientes.Ciudad, Clientes.Región,  
       Clientes.CódPostal, Clientes.País, Clientes.Teléfono  
FROM [Clientes]  
WHERE exists  
      (SELECT *  
       FROM [Pedidos], [Detalles de pedidos], [Productos], [Categorías]  
       WHERE [Detalles de pedidos].IdProducto = Productos.IdProducto AND  
             Pedidos.IdPedido = [Detalles de pedidos].IdPedido AND  
             Productos.IdCategoría = Categorías.IdCategoría and  
             categorías.nombrecategoría like '%repostería%')  
and exists  
      (SELECT *  
       FROM [Pedidos], [Detalles de pedidos], [Productos]  
       WHERE [Detalles de pedidos].IdProducto = Productos.IdProducto AND  
             Pedidos.IdPedido = [Detalles de pedidos].IdPedido AND  
             productos.nombreproducto like '%queso%')
```

Ejemplo 05-05

Cientes que no han comprado nada durante el año 2005

```
SELECT Clientes.NombreCompañía  
FROM [Clientes]  
WHERE not exists  
      (select *  
       from [Pedidos]  
       where Pedidos.IdCliente = Clientes.IdCliente AND year(pedidos.fechapedido) = 2005)
```

Ejemplo 05-06

Lista de productos y la fecha del último pedido en el que aparece

```
SELECT Productos.NombreProducto,  
       (select max (pedidos.fechapedido)  
        from pedidos, [detalles de pedidos]  
        where [Detalles de pedidos].IdProducto = Productos.IdProducto AND  
              Pedidos.IdPedido = [Detalles de pedidos].IdPedido)  
From productos
```

Prácticas

Consultas simples

Quiero conocer el nombre de todos mis proveedores.

De que países son mis clientes.

Necesito toda la información disponible sobre las compañías de envío.

Deme la fecha de entrega de todos los pedidos y su destino ordenado por país y fecha más lejana a más cercana.

Quiero las ciudades por orden alfabético de mis proveedores con el nombre de todos ellos, también por orden alfabético.

Cuantos trienios tiene cada uno de mis empleados.

Filtrando los datos

Cuanto supone un descuento de un 12% en el pedido 10251.

Que proveedores no tienen Web.

Necesito conocer los pedidos del mes de agosto de 2005.

Que pedidos han sido entregados en 14 días.

Sumarizando y agrupando los datos

De cuantos países son mis clientes.

Cuál es el valor total de los pedidos mensualmente.

Lista de los empleados con la fecha del último pedido realizado. Esta consulta la suele realizar el departamento de RRHH.

Cuáles son nuestros 10 clientes más antiguos.

Cuantos productos se pidieron en el pedido 10252 y cuál es el importe total del mismo.

Que vendedor ha conseguido ventas superiores a 1000\$ durante el último mes.

Que categorías están un 20% por encima del resto en cuanto a importe facturado.

Consultando múltiples tablas

Quiero saber cuáles son los clientes que realizan pedidos de marisco de diferente tipo.

Vendedores que nunca han vendido bebidas y fecha de contratación del mismo.

Dime los clientes que han realizado pedidos de un mínimo de tres categorías diferentes durante el último mes.

Dime los clientes que han tenido trato con un mínimo de tres vendedores.

Vamos a recuperar a los clientes que no nos han hecho ninguna compra durante este año.

Quiero saber que compañías de envíos utilizo más en los pedidos al extranjero.

Quiero saber con los productos de que proveedor he factura más, he hecho más pedidos y he vendido más unidades.

Lista de los clientes que han comprado marisco o bien pedidos con destino a Bélgica.

Lista de mis empleados con los clientes que viven en la misma ciudad.

Subconsultas

Pedidos en el que todas las líneas sean de más de 10\$.

Cientes cuyo importe total de pedidos está por encima de la media en un 50%.

Productos que no han sido pedidos durante el último mes.

Vendedores que han hecho alguna venta a clientes de Italia.

Deseo ver los pedidos de los artículos que tengan pedidos por importe superior a.....

Categorías para cuyos productos ha habido más de 5 pedidos mensuales.

Vendedores que han vendido todos los productos alguna vez.