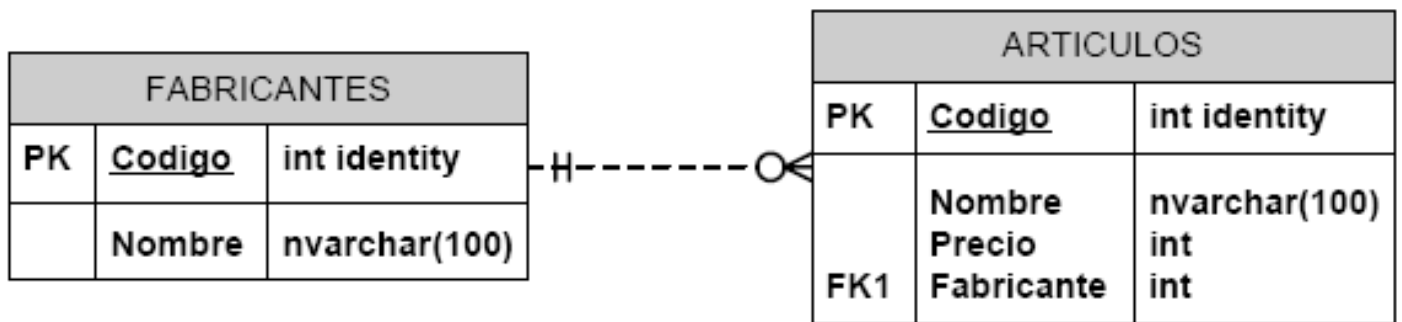
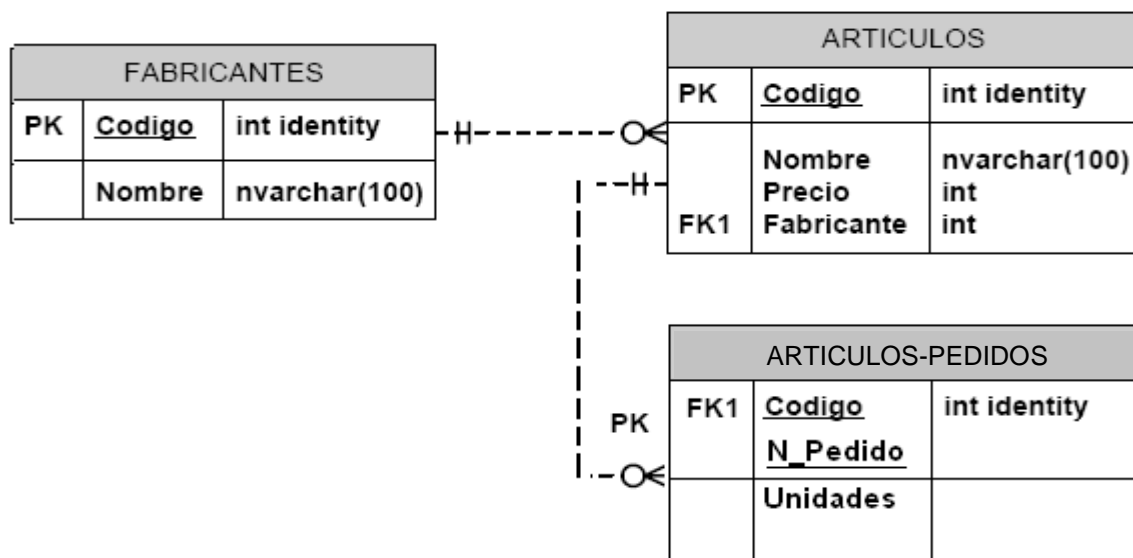


ARTÍCULOS



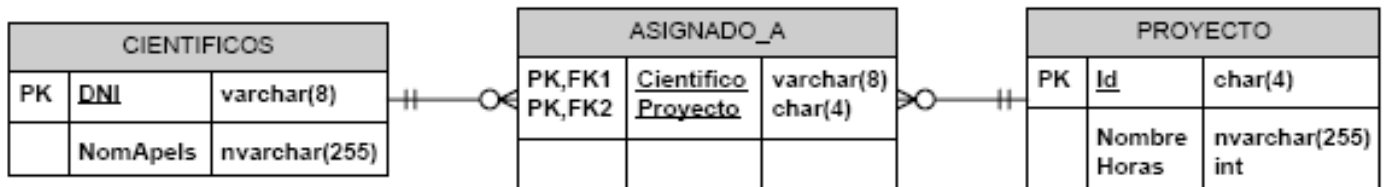
1. Obtener los nombres de los productos de la tienda
2. Obtener los nombres y los precios de los productos de la tienda
3. Obtener el nombre de los productos cuyo precio sea menor o igual a 200 €
4. Obtener todos los datos de los artículos cuyo precio esté entre los 60 € y los 120€ (ambas cantidades incluidas).
5. Obtener el nombre y el precio en pesetas (es decir, el precio en euros multiplicado por 166,386).
6. Seleccionar el precio medio de todos los productos.
7. Obtener el precio medio de los artículos cuyo código de fabricante sea 2
8. Obtener el número de artículos cuyo precio sea mayor o igual a 180€.
9. Obtener el nombre y precio de los artículos cuyo precio sea mayor o igual a 180€ y ordenarlos descendientemente por precio, y luego ascendientemente por nombre.
10. Obtener un listado completo de artículos, incluyendo por cada artículo los datos del artículo y de su fabricante. Si hay algún fabricante sin artículos mostrarlo también.
11. Obtener un listado de artículos, incluyendo el nombre del artículo, su precio, y el nombre de su fabricante.
12. Obtener el precio medio de los productos de cada fabricante, mostrando solo los códigos de fabricante.
13. Obtener el precio medio de los productos de cada fabricante, mostrando el nombre del fabricante.

14. Obtener los nombres de los fabricantes que ofrezcan productos cuyo precio medio sea mayor o igual a 150 €.
15. Obtener el nombre y precio del artículo más barato.
16. Obtener una lista con el nombre y precio de los artículos más caros de cada proveedor (incluyendo el nombre del proveedor)
17. Añadir un nuevo producto: Altavoces de 30 € (del fabricante 2)
18. Cambiar el nombre del producto 8 a 'Impresora Laser'.
19. Aplicar un descuento del 10% (multiplicar el precio por 0,9) a todos los productos.
20. Aplicar un descuento de 10 € a todos los productos cuyo precio sea mayor o igual a 120 €.
21. Mostrar los registros repetidos (nombre, precio y fabricante) de la tabla artículos y el nº de ocurrencias.
22. Mostrar los registros no repetidos (nombre, precio y fabricante) de la tabla artículos.



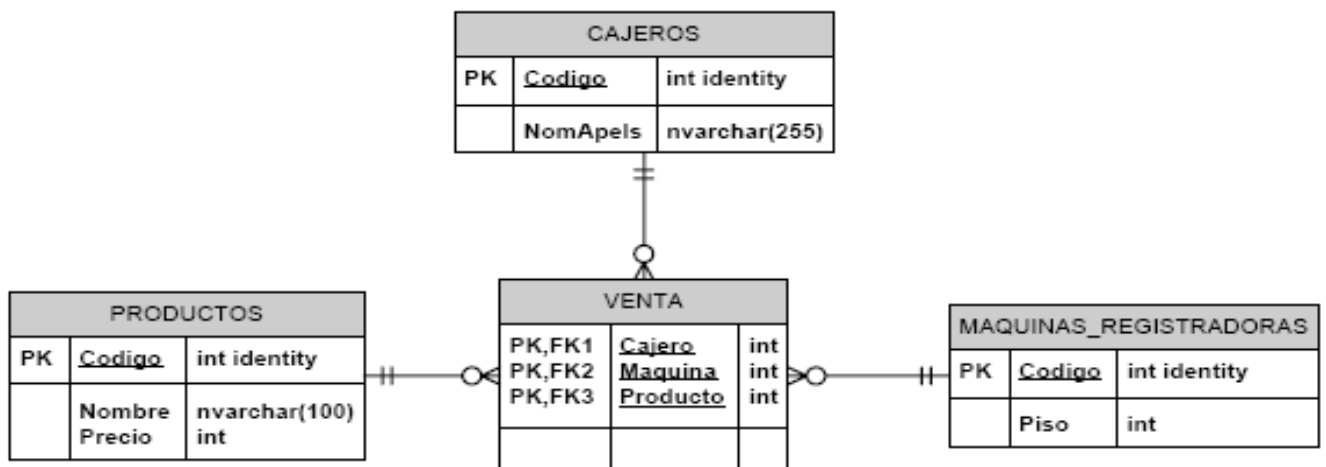
23. Mostrar los registros de la tabla artículos-pedidos (nombre y fabricante) que existan en la tabla artículos.
24. Mostrar los registros de la tabla artículos-pedidos que no existan en artículos.
25. Mostrar los registros de artículos que no existan en artículos pedidos.

CIENTÍFICOS



1. Sacar una relación completa de los científicos asignados a cada proyecto. Mostrar DNI, Nombre del científico, Identificador del proyecto y nombre del proyecto
2. Obtener el numero de proyectos al que está asignado cada científico (mostrar el DNI y el nombre).
3. Obtener el numero de científicos asignados a cada proyecto (mostrar el identificador del proyecto y el nombre del proyecto).
4. Obtener el número de horas de dedicación de cada científico.
5. Obtener el DNI y el nombre de los científicos que se dedican a más de un proyecto y cuya dedicación media a cada proyecto sea superior a 80 horas.

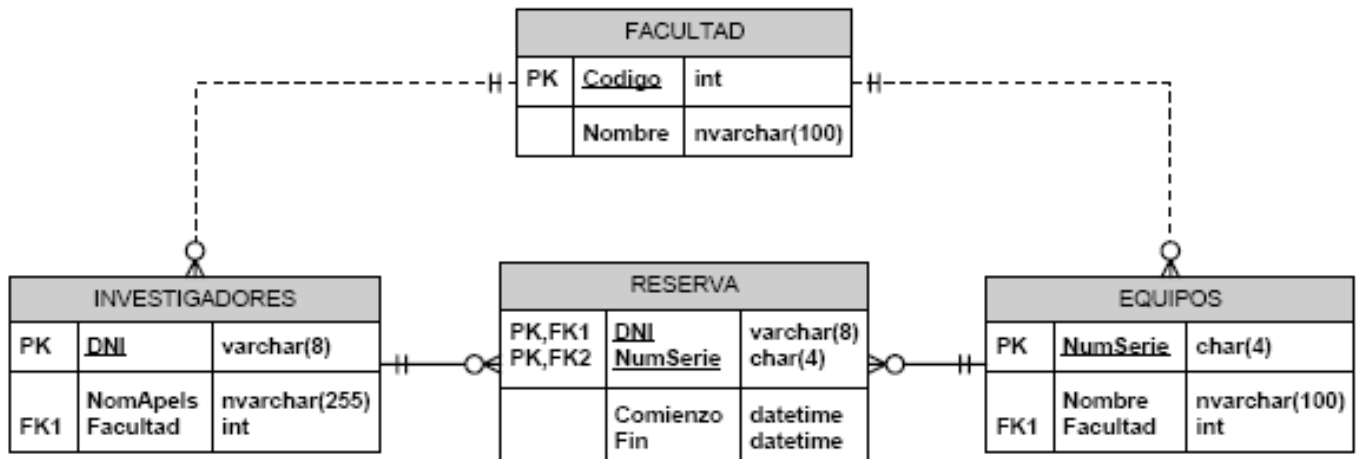
GRANDES ALMACENES



1. Mostrar el número de ventas de cada producto, ordenado de más a menos ventas.
2. Obtener un informe completo de ventas, indicando el nombre del cajero que realizó la venta, nombre y precios de los productos vendidos, y piso en el que se encuentra la maquina registradora donde se realizó la venta.
3. Obtener las ventas totales realizadas en cada piso.

4. Obtener el código y nombre de cada empleado junto con el importe total de sus ventas.
5. Obtener el código y nombre de aquellos cajeros que hayan realizado ventas en pisos cuyas ventas totales sean inferiores a los 500 ¤.

INVESTIGADORES

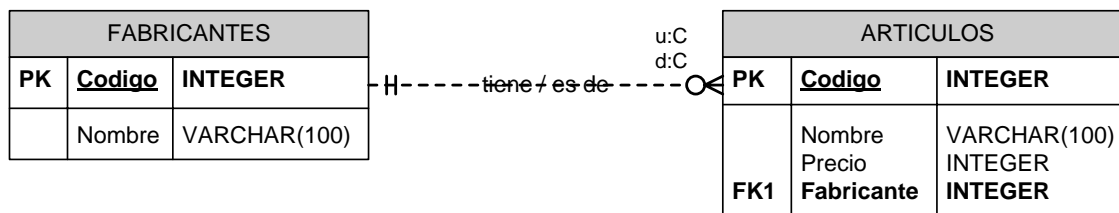


1. Obtener el DNI y nombre de aquellos investigadores que han realizado más de una reserva.
2. Obtener un listado completa de reservas, incluyendo los siguientes datos:
 - a. DNI y nombre del investigador, junto con el nombre de su facultad.
 - b. Numero de serie y nombre del equipo reservado, junto con el nombre de la facultad a la que pertenece.
 - c. Tiempo que ha durado la reserva (desde la Fecha de comienzo hasta la fecha de fin de la reserva)
3. Obtener el DNI y el nombre de los investigadores que han reservado equipos que no son de su facultad; mostrar la letra del DNI en mayúsculas y el Nombre y Apellidos de los investigadores con la primera letra en mayúsculas y el resto en minúsculas.
4. Obtener los nombres de las facultades en las que ningún investigador ha realizado una reserva; llamar a la columna "Facultades con equipos patateros".

5. Obtener los nombres de las facultades con investigadores 'ociosos' (investigadores que no han realizado ninguna reserva).
6. Obtener el número de serie y nombre de los equipos que nunca han sido reservados; dejarle a la columna del N^o de serie 10 espacios y la del Nombre de los equipos 70 espacios.

EJERCICIOS SQL

Suponer una tienda de informática para la que se ha creado una BD con las siguientes tablas:



26. Definir las tablas de la BD.

27. Obtener los nombres de los productos de la tienda

28. Obtener los nombres y los precios de los productos de la tienda

29. Obtener el nombre de los productos cuyo precio sea menor o igual a 200 €

30. Obtener todos los datos de los artículos cuyo precio esté entre los 60 € y los 120€ (ambas cantidades incluidas).

31. Obtener el nombre y el precio en pesetas (es decir, el precio en euros multiplicado por 166,386).

32. Seleccionar el precio medio de todos los productos.

33. Obtener el precio medio de los artículos cuyo código de fabricante sea 2

34. Obtener el número de artículos cuyo precio sea mayor o igual a 180€.

35. Obtener el nombre y precio de los artículos cuyo precio sea mayor o igual a 180€ y ordenarlos descendientemente por precio, y luego ascendientemente por nombre.

36. Obtener un listado completo de artículos, incluyendo por cada artículo los datos del artículo y de su fabricante.

37. Obtener un listado de artículos, incluyendo el nombre del artículo, su precio, y el nombre de su fabricante.

38. Obtener el precio medio de los productos de cada fabricante, mostrando solo los códigos de fabricante.

39. Obtener el precio medio de los productos de cada fabricante, mostrando el nombre del fabricante.

40. Obtener los nombres de los fabricantes que ofrezcan productos cuyo precio medio sea mayor o igual a 150 €.

41. Obtener el nombre y precio del artículo más barato.

42. Obtener una lista con el nombre y precio de los artículos más caros de cada proveedor (incluyendo el nombre del proveedor)

43. Añadir un nuevo producto: Altavoces de 30 € (del fabricante 2)

44. Cambiar el nombre del producto 8 a 'Impresora Laser'.

45. Aplicar un descuento del 10% (multiplicar el precio por 0,9) a todos los productos.

46. Aplicar un descuento de 10 € a todos los productos cuyo precio sea mayor o igual a 120 €.

47. Mostrar los registros repetidos (nombre, precio y fabricante) de la tabla artículos y el nº de ocurrencias.

48. Mostrar los registros no repetidos (nombre, precio y fabricante) de la tabla artículos.

49. Mostrar los registros de la tabla artículos-pedidos (nombre y fabricante) que existan en la tabla artículos.

50. Mostrar los registros de la tabla artículos-pedidos que no existan en artículos.

51. Mostrar los registros de artículos que no existan en artículos pedidos.

1. Búsqueda de registros existentes y no existentes entre tablas en T-SQL

En este artículo (que es bastante sencillo pero demasiado útil) voy a exponer de una forma fácil y práctica la forma de obtener registros que se encuentren y no se encuentren en otra tabla según nuestros criterios de búsqueda.

Esto resulta demasiado útil cuando intentamos detectar registros duplicados dentro de una tabla o dentro de varias tablas, o cuando necesitamos ver si en una tabla faltan datos de que se encuentren en otra, etc.

Supongamos lo siguiente: Tenemos una tabla con información de Compañeros, esta cuenta con un registro repetido dos veces:

--Registros dentro de Compañeros

CompaneroId ApellidoPat ApellidoMat Nombres

```
-----  
1 Lopez Garcia Pedro  
2 Perez Rodriguez Juan  
3 Lopez Garcia Roque  
4 Perez Garcia Jesus  
5 Gonzalez Abudes Puyul  
6 Perez Rodriguez Juan
```

(6 row(s) affected)

Como pueden ver el registro repetido es:

--Registros repetidos dentro de Compañeros y el total de las ocurrencias

ApellidoPat ApellidoMat Nombres Total

```
-----  
Perez Rodriguez Juan 2
```

(1 row(s) affected)

Y los registros que se mantienen sin redundancia son:

--Registros no repetidos dentro de Compañeros

ApellidoPat ApellidoMat Nombres

```
-----  
Gonzalez Abudes Puyul  
Lopez Garcia Pedro  
Lopez Garcia Roque  
Perez Garcia Jesus
```

(4 row(s) affected)

Esto en veces es demasiado útil y práctico y es algo que se nos presenta muy seguido (por eso lo publico hasta a mi mismo se me olvida jejeje). Aquí les dejo los queries correspondientes:

- print '--Registros dentro de Compañeros'
`select * from Compañeros`

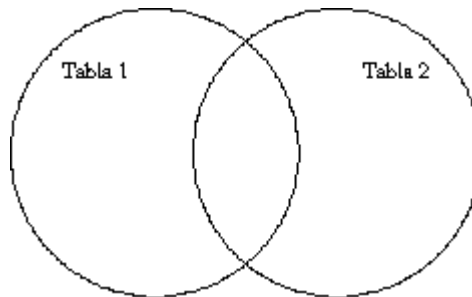
- print '--Registros repetidos dentro de Compañeros y el total de las ocurrencias'

```
SELECT ApellidoPat,ApellidoMat,Nombres, count(*) Total  
from Compañeros  
group by ApellidoPat,ApellidoMat, Nombres  
having count(*) > 1
```

- `print '--Registros no repetidos dentro de Companeros'`
`SELECT ApellidoPat,ApellidoMat,Nombres`
`from Companeros`
`group by ApellidoPat,ApellidoMat, Nombres`
`having count(*) = 1`

Ahora, muchas de las ocasiones necesitamos determinar ocurrencias o redundancia entre datos almacenados en diversas tablas. Esto lo vamos a comenzar explicándolo de forma gráfica:

Supongamos el siguiente conjunto de datos:



Entre estos dos existe información en común, lo que requerimos es determinar esa intersección de información para determinar las ocurrencias entre los dos conjuntos de datos. Esto sería sencillo hacerlo con un simple INNER JOIN, pero hay ocasiones en las que cuando el número de tablas es mayor y estas se relacionan con más tablas resulta ser algo tedioso.

print '--Registros dentro de Amigos'
`select * from Amigos`

```
AmigoId ApellidoPat ApellidoMat Nombres
-----
1 Villaneda Avila Sergio José
2 Villaneda Avila Andrea
3 Villaneda Avila Sarahi

(3 row(s) affected)
```

print '--Registros dentro de Conocidos'
`select * from Conocidos`

```
ConocidoId ApellidoPat ApellidoMat Nombres
-----
1 Villaneda Avila Sergio José
2 Sanchez Gamboa Mario Alberto
3 Gozalez Cancalas Luis

(3 row(s) affected)
```

print '--Registros dentro de Amigos que existan dentro de Conocidos (INNER JOIN)'
`SELECT dbo.Amigos.ApellidoPat, dbo.Amigos.ApellidoMat, dbo.Amigos.Nombres`
`FROM dbo.Amigos`
`INNER JOIN dbo.Conocidos`
`ON dbo.Amigos.ApellidoPat = dbo.Conocidos.ApellidoPat`
`AND dbo.Amigos.ApellidoMat = dbo.Conocidos.ApellidoMat`
`AND dbo.Amigos.Nombres = dbo.Conocidos.Nombres`

```
ApellidoPat ApellidoMat Nombres
-----
Villaneda Avila Sergio José
```


(1 row(s) affected)

Hay otra forma de hacerlo y es con la cláusula EXISTS y sería de la siguiente forma:

print '--Registros dentro de Amigos que existan dentro de Conocidos (EXISTS)'

```
select t1.ApellidoPat, t1.ApellidoMat, t1.Nombres
from Amigos as t1
where exists
(select t2.ApellidoPat, t2.ApellidoMat, t2.Nombres
from Conocidos as t2
where t1.ApellidoPat=t2.ApellidoPat
and t1.ApellidoMat=t2.ApellidoMat
and t1.Nombres=t2.Nombres)
```

ApellidoPat ApellidoMat Nombres

Villaneda Avila Sergio José

(1 row(s) affected)

Como ven arroja el mismo resultado?. Ahora, si queremos seleccionar los registros de la forma inversa, es decir, que estén dentro de la tabla Amigos pero no estén dentro de Conocidos, sería algo por el estilo: (también dejo la forma inversa, entre Conocidos y Amigos)

print '--Registros dentro de Amigos que no existan dentro de Conocidos'

```
select t1.ApellidoPat, t1.ApellidoMat, t1.Nombres
from Amigos as t1
where not exists
(select t2.ApellidoPat, t2.ApellidoMat, t2.Nombres
from Conocidos as t2
where t1.ApellidoPat=t2.ApellidoPat
and t1.ApellidoMat=t2.ApellidoMat
and t1.Nombres=t2.Nombres)
```

ApellidoPat ApellidoMat Nombres

Villaneda Avila Andrea
Villaneda Avila Sarahi

(2 row(s) affected)

print '--Registros dentro de Conocidos que existan dentro de Amigos'

```
select t2.ApellidoPat, t2.ApellidoMat, t2.Nombres
from Conocidos as t2
where exists
(select t1.ApellidoPat, t1.ApellidoMat, t1.Nombres
from Amigos as t1
where t2.ApellidoPat=t1.ApellidoPat
and t2.ApellidoMat=t1.ApellidoMat
and t2.Nombres=t1.Nombres)
```

ApellidoPat ApellidoMat Nombres

Villaneda Avila Sergio José

(1 row(s) affected)

print '--Registros dentro de Conocidos que no existan dentro de Amigos'

```
select t2.ApellidoPat, t2.ApellidoMat, t2.Nombres
```

```
from Conocidos as t2
where not exists
(select t1.ApellidoPat, t1.ApellidoMat, t1.Nombres
from Amigos as t1
where t2.ApellidoPat=t1.ApellidoPat
and t2.ApellidoMat=t1.ApellidoMat
and t2.Nombres=t1.Nombres)
```

ApellidoPat	ApellidoMat	Nombres
-------------	-------------	---------

Sanchez	Gamboa	Mario Alberto
Gonzalez	Cancalas	Luis

(2 row(s) affected)