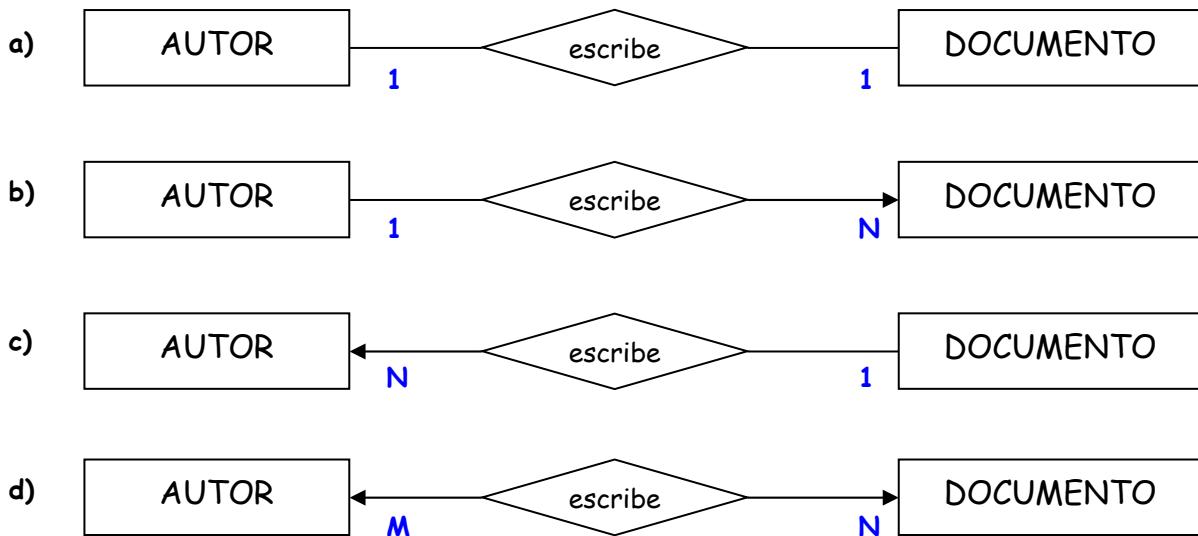
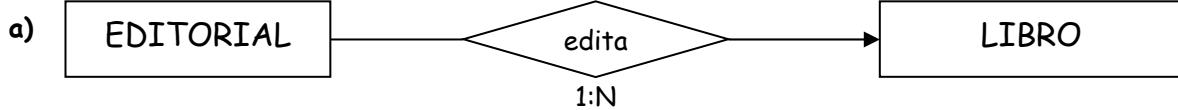


—Ejercicios Unidad 01 —

1.- Interpreta los siguientes diagramas E/R. Indica su tipo de correspondencia.



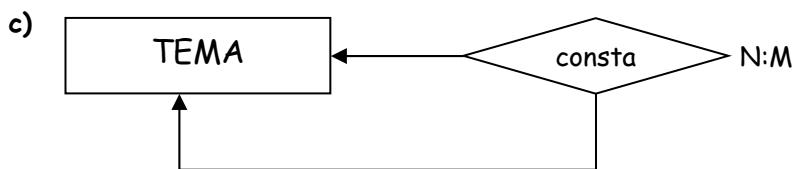
2.- Interpreta los siguientes diagramas E/R.



Editorial publica muchos libros, un libro es editado por una sola editorial



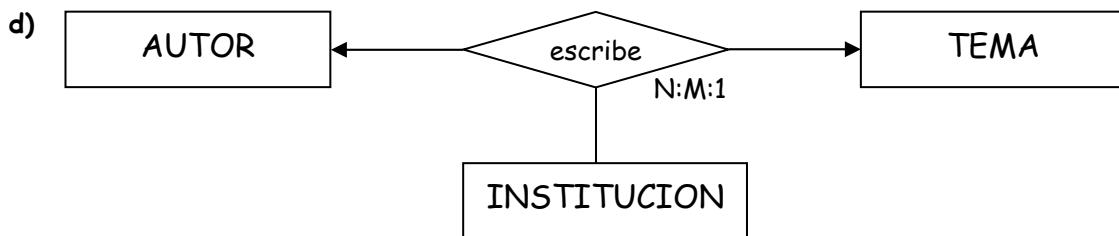
El autor corrige muchos libros, un libro puede ser corregido por muchos autores



Un tema consta de varios subtemas

Un subtema puede formar parte 1 o varios temas

Un libro consta de varios capítulos y un capítulo estar en varios libros.



Una institución puede tener muchos autores y temas.

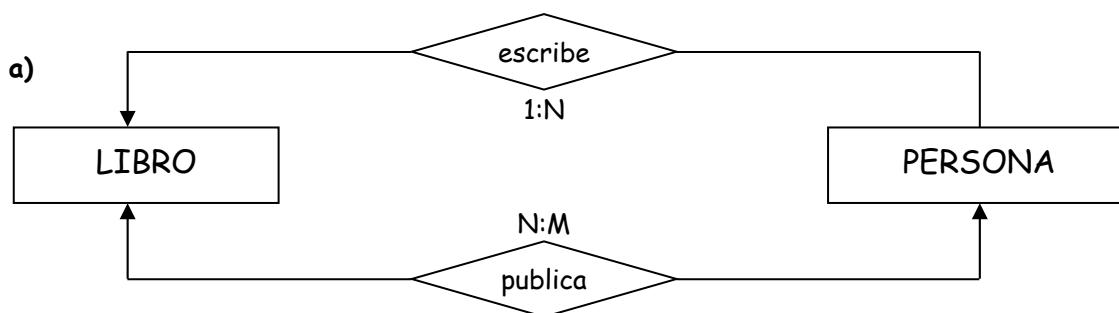
Un autor solo escribe para una única institución.

Un tema pertenece a una única institución

Un autor puede escribir muchos temas.

Un tema puede ser escrito por muchos autores.

3.- Interpreta los siguientes diagramas de modelo E/R.

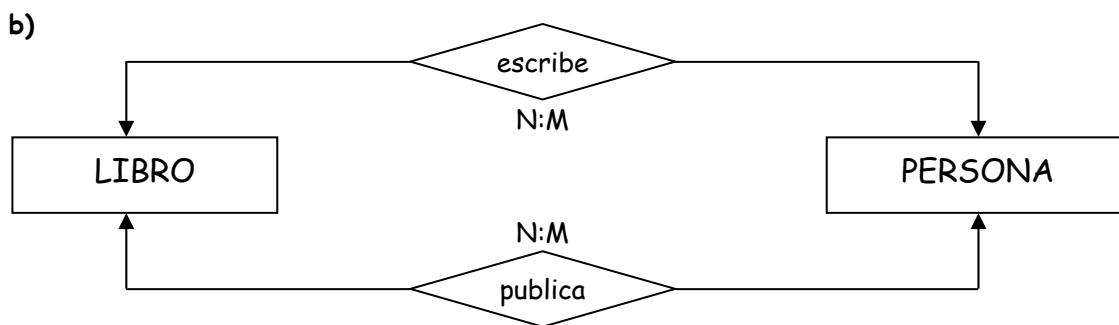


Un libro es escrito por una sola persona

Muchos libros pueden ser escritos por una persona

Un libro puede ser publicado por muchas personas

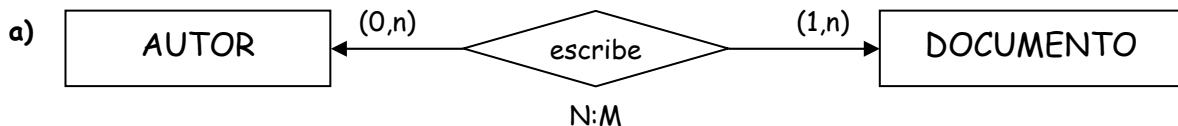
Una persona puede publicar muchos libros



Un libro es publicado por muchas personas y una persona puede publicar muchos libros.

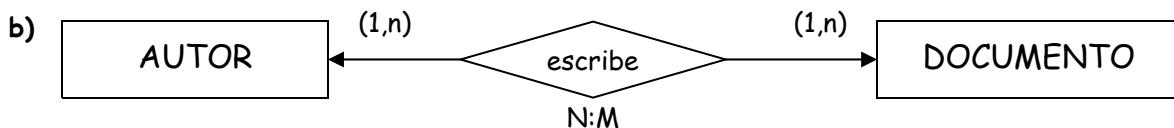
Un libro puede ser escrito por muchas personas y una persona puede escribir muchos libros.

4.- Interpreta las siguientes cardinalidades:

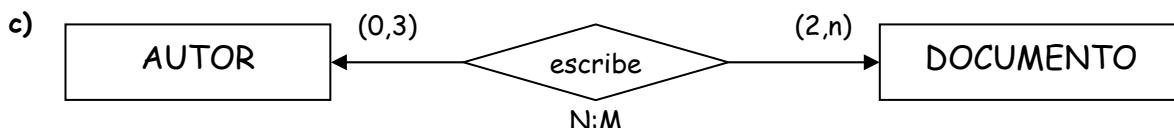


Un autor escribirá 1 o mas documentos

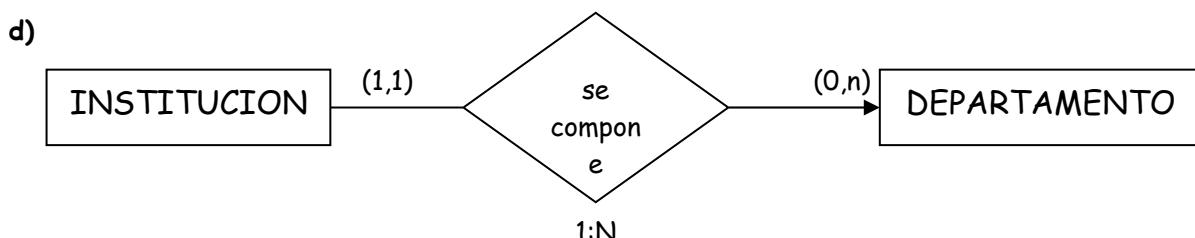
Un documento puede ser escrito por 0 (anónimo) o mas autores



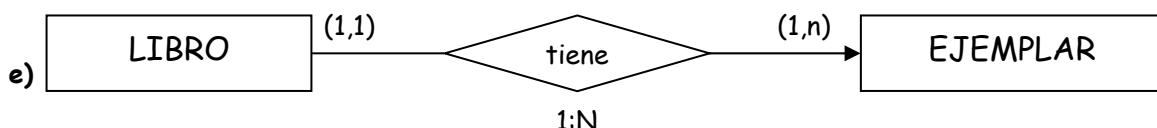
Un autor escribirá 1 o mas documentos
Un documento puede ser escrito por 1 o mas autores



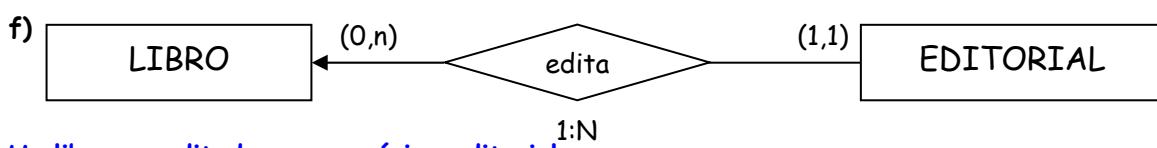
Un autor escribirá 2 o mas documentos
Un documento puede ser escrito por 0 (anónimo) o hasta 3 autores



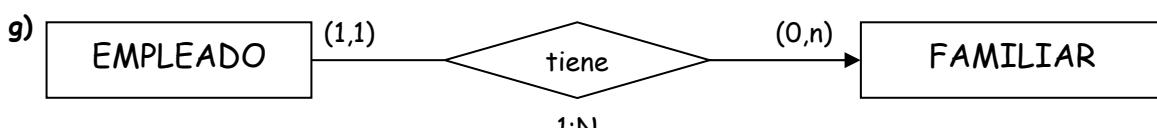
Una Institución se compone de 0 (Ninguno) o mas Departamentos
Un Departamento pertenece a una sola Intitucion



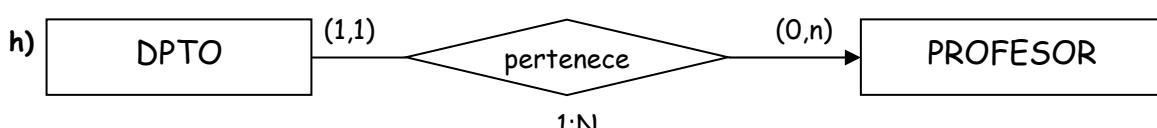
De un Libro hay 1 o mas ejemplares
Los Ejemplares serán del mismo Libro



Un libro es editado por una única editorial
Una editorial puede editar 0 (Ninguno) o varios libros

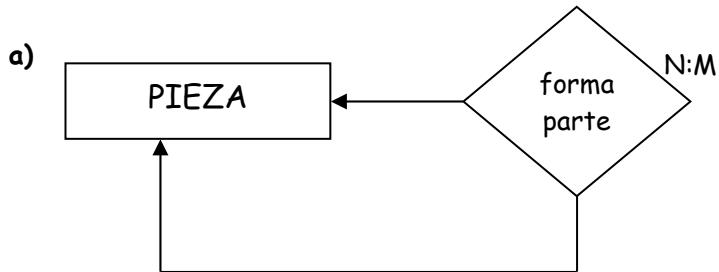


Un empleado puede tener 0 (Ninguno) o mas Familiares
Solo un Familiar puede estar como Empleado



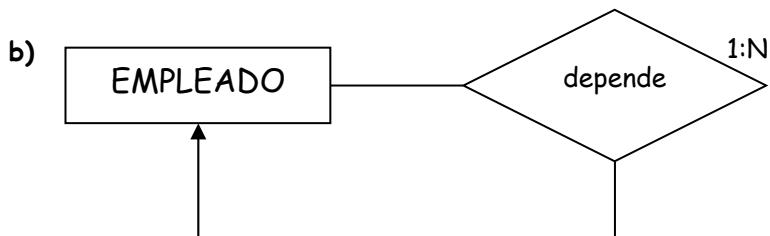
En un Departamento puede haber 0 (Ninguno) o muchos Profesores
Un Profesor solo pertenece a un Departamento

5.- Interpreta las siguientes gráficos del modelo E/R con relaciones reflexivas:



Una pieza puede estar compuesta de muchos elementos (subpiezas)

Un elemento puede formar parte de muchas piezas



De un empleado dependen otros muchos (el seria un jefe)

Pero un empleado depende de solo uno (dependería de 1 solo jefe)

6.- Definida la relación inferior, crea e interpreta distintos tipos de relación y cardinalidades:



1:1 Un profesor imparte a 1 curso/1 curso tiene un profesor

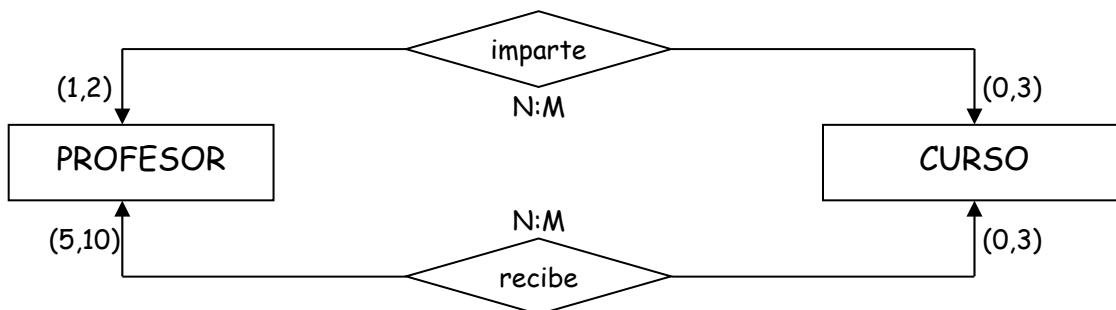
1:N (1,n) Un Profesor puede impartir clases a 1 o mas cursos

N:1 (1,n) En un Curso pueden impartir clases 1 o mas Profesores

M:N 1 o mas profesores dan clase a 1 o mas cursos y viceversa

Y de cada uno las diferentes variantes (0,n/1,n)

7.- Interpreta este modelo:



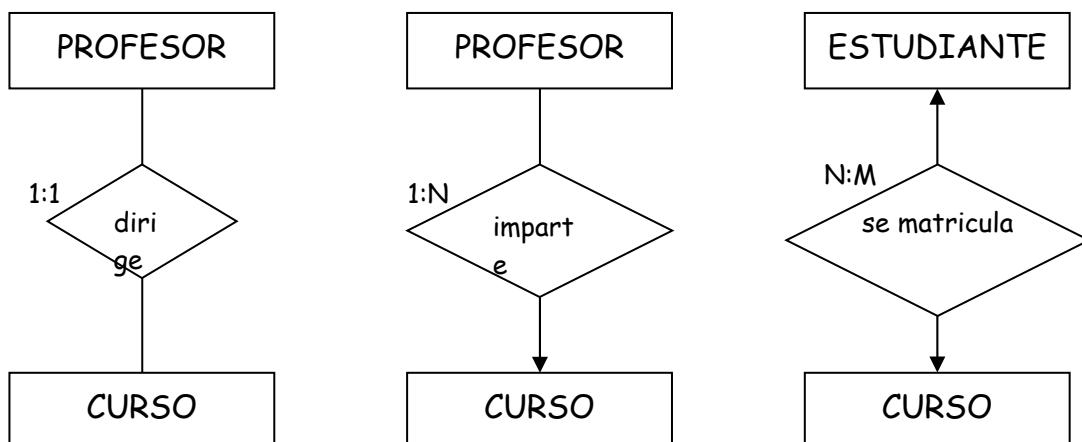
Un Profesor imparte clases a Ninguno o hasta 3 cursos.

A un curso le pueden dar clases 1 o 2 profesores.

A un Curso pueden asistir 5 a 10 profesores.

A un Profesor le pueden dar Ninguno o hasta 3 cursos.

8.- Interpreta los modelos:



Un profesor dirige un curso y un curso es dirigido por un solo profesor

Un profesor imparte 1 o más curso, pero un curso solo lo imparte 1 profesor

Un estudiante se puede matricular en muchos cursos y en un curso puede haber muchos estudiantes.

9.- Haz una interpretación de los gráficos e indica qué atributo(s) sería(n) necesario(s) para la relación:



Hombre se casa con una mujer y un hombre se casa con una mujer

Atributos Hombre: DNI, Nombre, Edad, domicilio, Aficiones

Atributos Mujer: DNI, Nombre, Edad, domicilio, Aficiones

DNI, Nombre, Edad, domicilio

Atributos Matrimonio: Ninguno (Fecha, Hora, Lugar)

Si al matrimonio le ponemos atributos, por ejemplo la Fecha, se puede dar el caso que la boda se realice 2 veces en fechas diferentes y en ese caso la relación seria (N:M).



Un curso puede ser dado por 1 o mas profesores y un profesores puede dar clase a un curso o no.

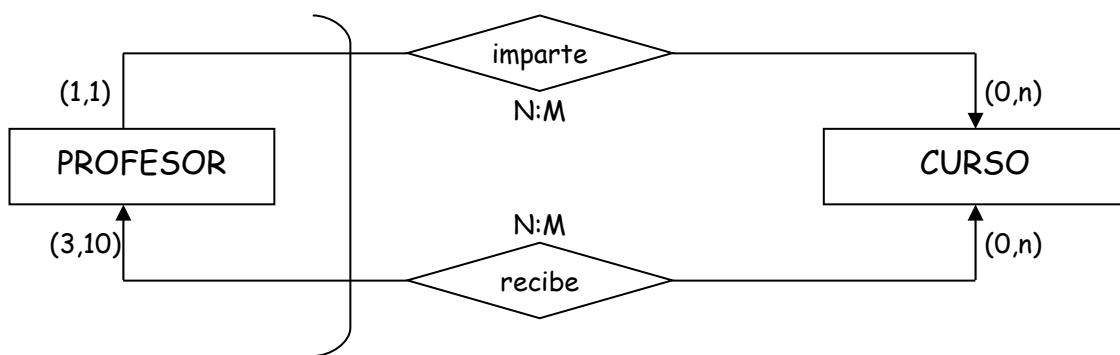
Atributos Profesor: Nombre, DNI, Dirección

Atributos Curso: Código, Tipo, Especialidad, Nombre

Atributos para Imparte: Hora y Dia.

Es N:M porque un profesor puede dar clases a muchos cursos y un curso puede tener muchos profesores.

10.- Interpreta el modelo:



Exclusividad (indica que solo se cumple una de las condiciones)

Un profesor imparte Ninguno o muchos cursos (0,n)

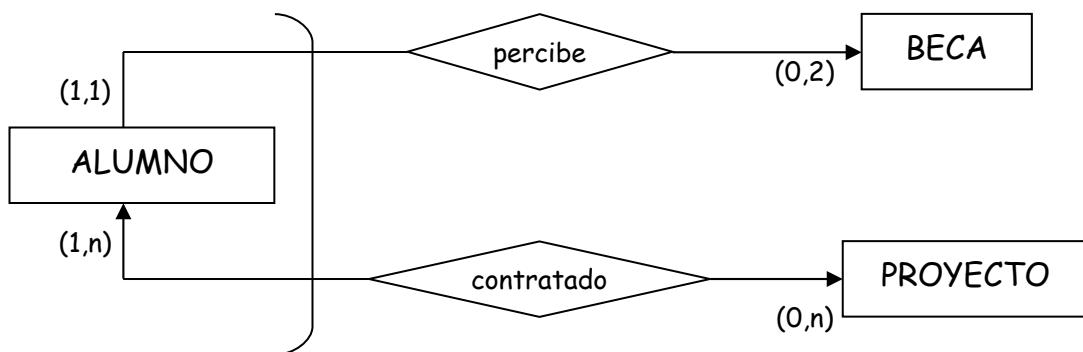
Un curso es impartido por un único profesor (1,1)

Un profesor recibe Ninguno o muchos cursos (0,n)

A un curso pueden asistir de 3 a 10 Profesores (3,10)

Las exclusividad indica que un Profesor o imparte o recibe cursos, no ambas a la vez.

11.- Interpreta el modelo:



Exclusividad

Un Alumno percibe Ninguna o hasta 2 Becas (0,2)

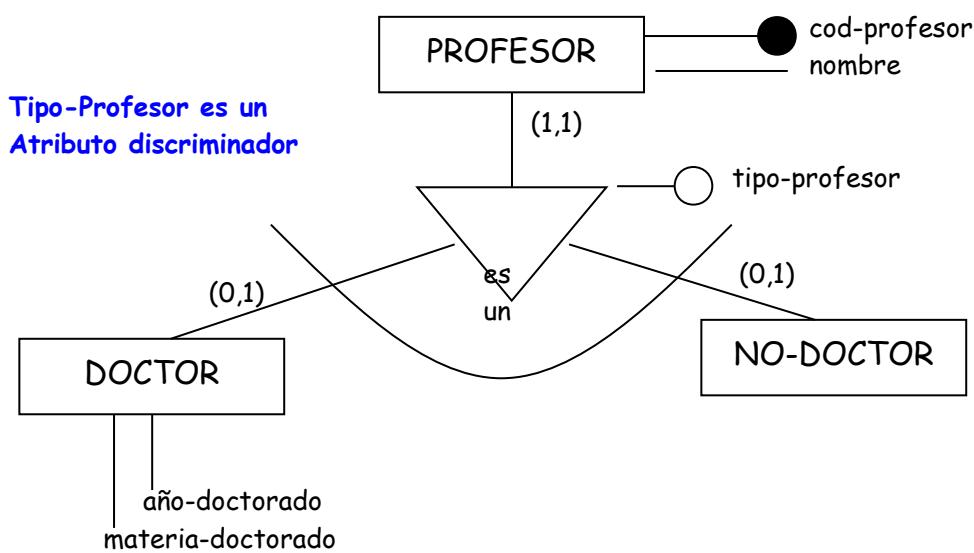
Una Beca es percibida por un solo alumno (1,1)

Un Alumno es contratado para Ninguno o muchos Proyectos (0,n)

En un Proyecto se contratan a 1 o mas Alumnos (1,n)

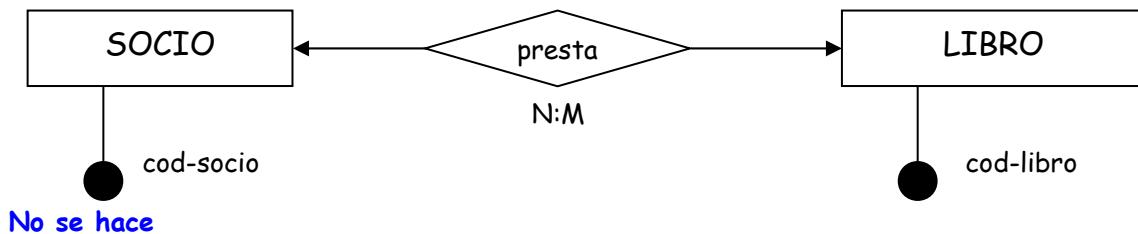
Las exclusividad indica de que un Alumno o percibe una Beca o es contratado para un Proyecto, no ambas a la vez.

12.- Interpreta:

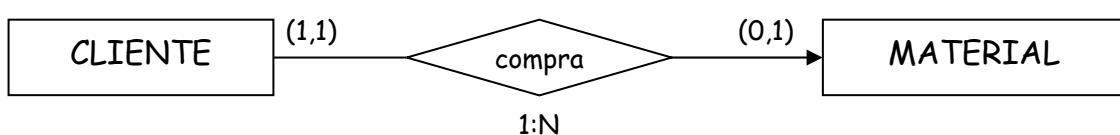


Un profesor o es Doctor o No doctor

13.- ¿Qué atributos asignarías a la relación?

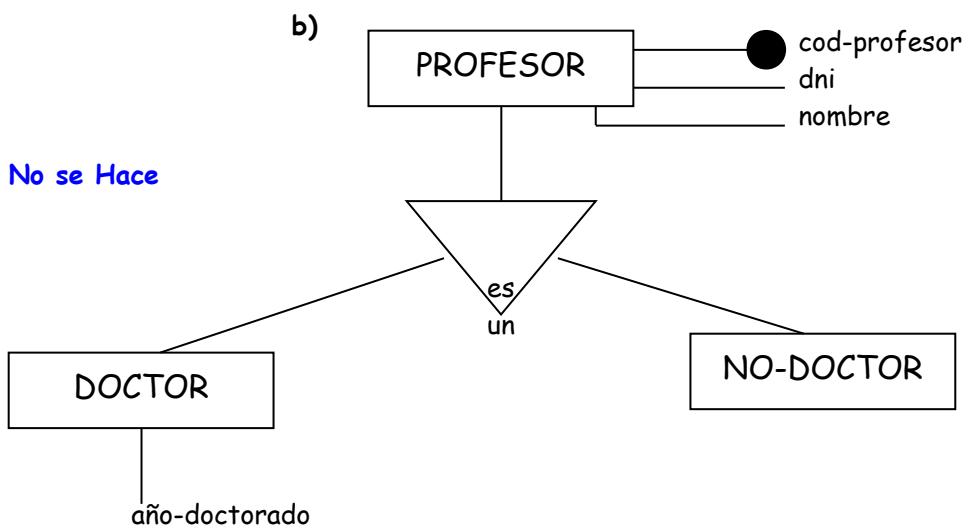
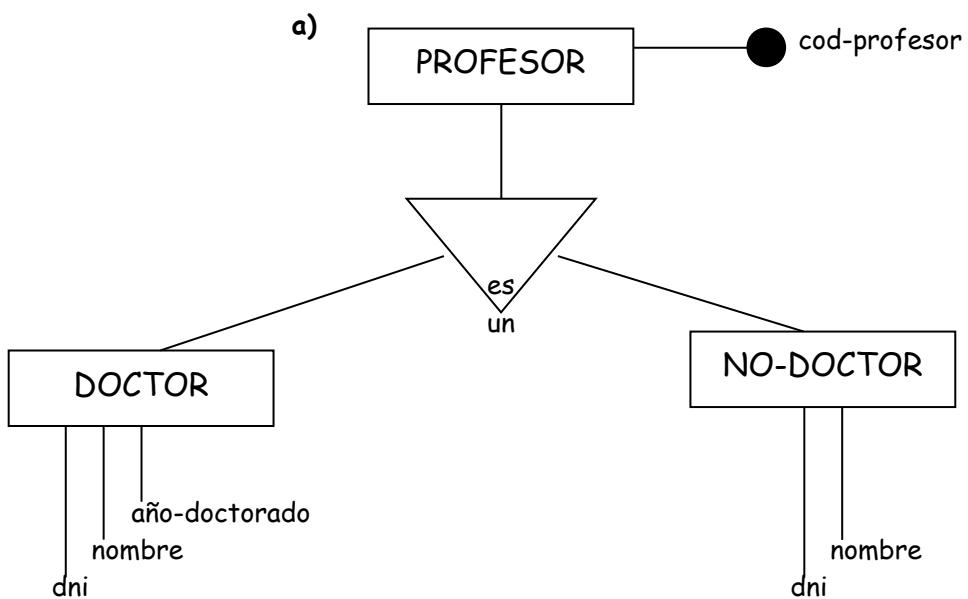


14.- Analiza la idoneidad de las cardinalidades y el tipo de relación:

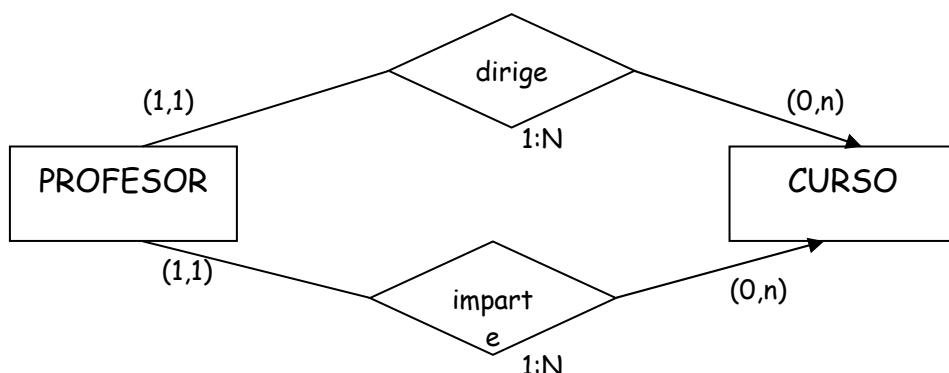


No puede haber una Relación Cliente-Material 1:N y luego la cardinalidad ser (0,1), debiera ser (0,n), o cambiar la relación 1:N por 1:1 y quitar la flecha.

15.- Analiza estas dos estructuras y razona cual es la interpretación más correcta:

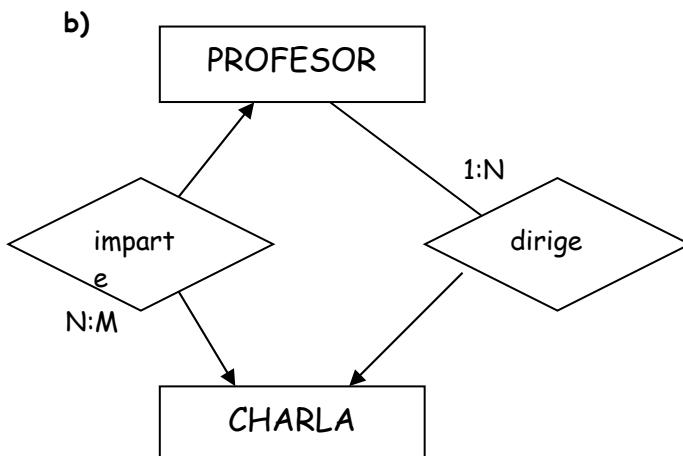
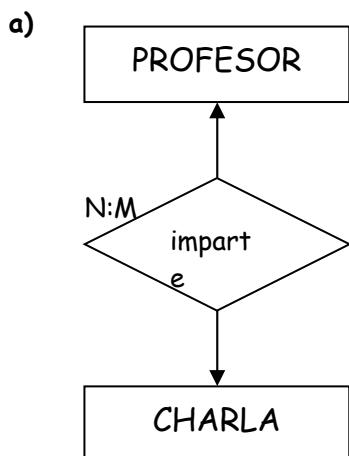


16.- Interpreta el gráfico:



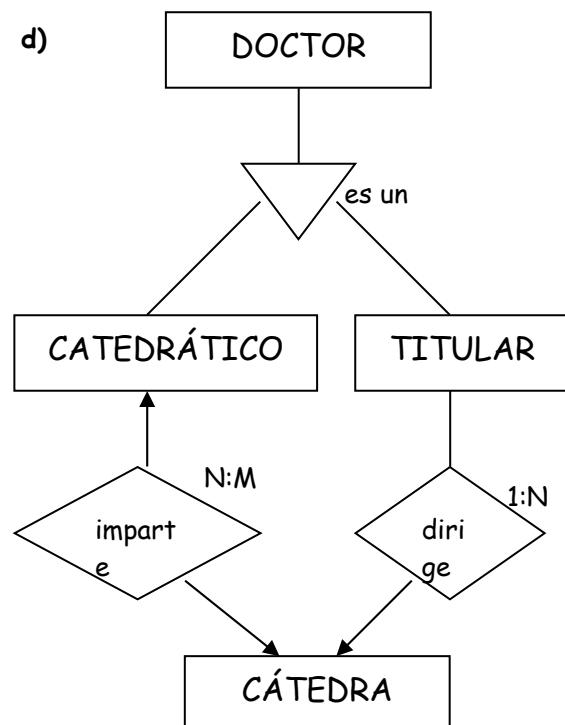
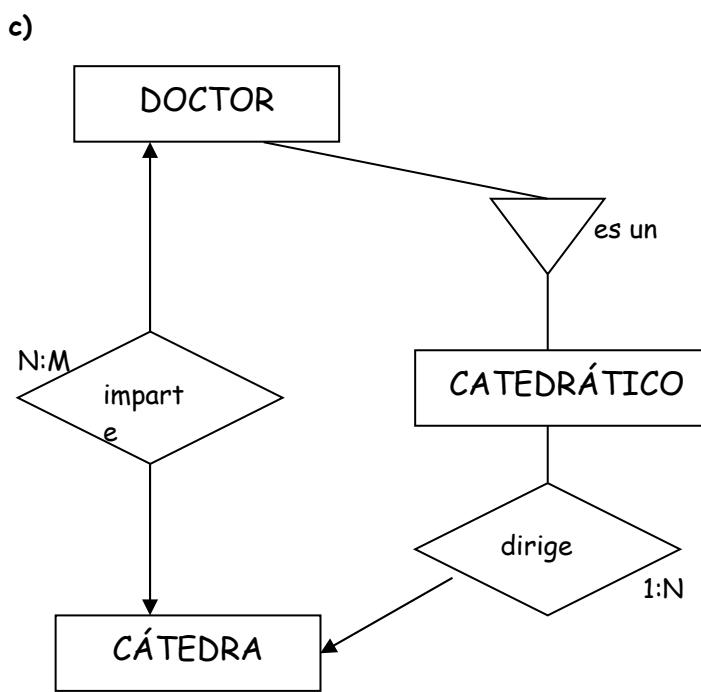
- Un Profesor puede dirigir Ninguno o muchos cursos
- Un Curso solo es dirigido por un único Profesor
- Un Profesor imparte Ninguno o muchos cursos
- Un Curso es impartido por un solo Profesor

17.- Interpreta las relaciones:



a-Un Profesor imparte muchas Charlas y Una Charla puede ser impartida por muchos Profesores.

b-Un Profesor imparte muchas Charlas y Una Charla puede ser impartida por muchos Profesores.
Un Profesor puede dirigir muchas Charlas, pero una Charla solo es dirigida por un Profesor.
Un profesor puede dirigir e impartir una charla.



c- Un Doctor imparte muchas Cátedras, además algunos doctores son Catedráticos y una Cátedra puede ser impartida por muchos Doctores (N.M).

Un Doctor si es Catedrático puede dirigir muchas Cátedras, pero una Cátedra solo es dirigida por un Doctor Catedrático (1:N).

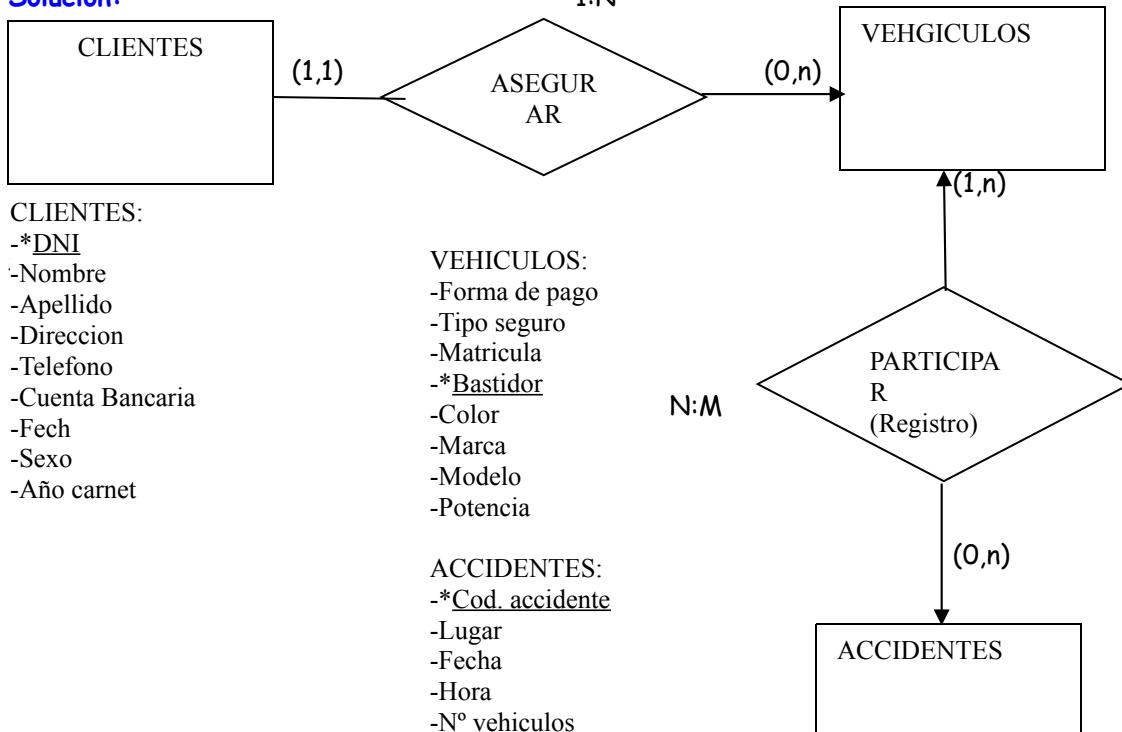
d-Un Doctor puede ser Catedrático, titular o ninguna de las dos e incluso las dos.

Si es Catedrático puede impartir muchas Cátedras y las Cátedras pueden ser impartidas por muchos Catedráticos (N:M).

Si es Titular puede dirigir muchas Cátedras, pero una Cátedra solo puede ser dirigida por un Titular (1:N).

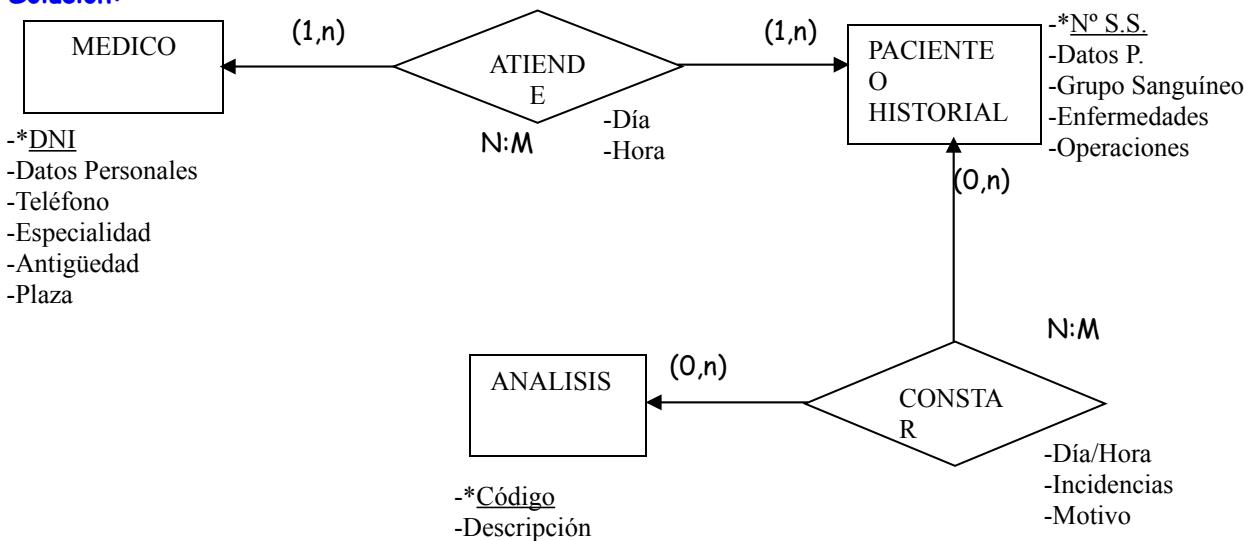
18.- Construye un diagrama E/R para una compañía de seguros automovilísticos que cuenta con un conjunto de **clientes**, cada uno de las cuales posee un cierto número de **vehículos asegurados** en la compañía. Se desea guardar información tanto de los clientes como de los vehículos asegurados. Hay que tener en cuenta que un mismo cliente puede tener varios vehículos asegurados. Se desea disponer de la información sobre todas aquellas personas que sean o hayan sido clientes de la compañía. Cada automóvil está registrado con un número de **accidentes** de los que también se desea recabar información.

Solucion:

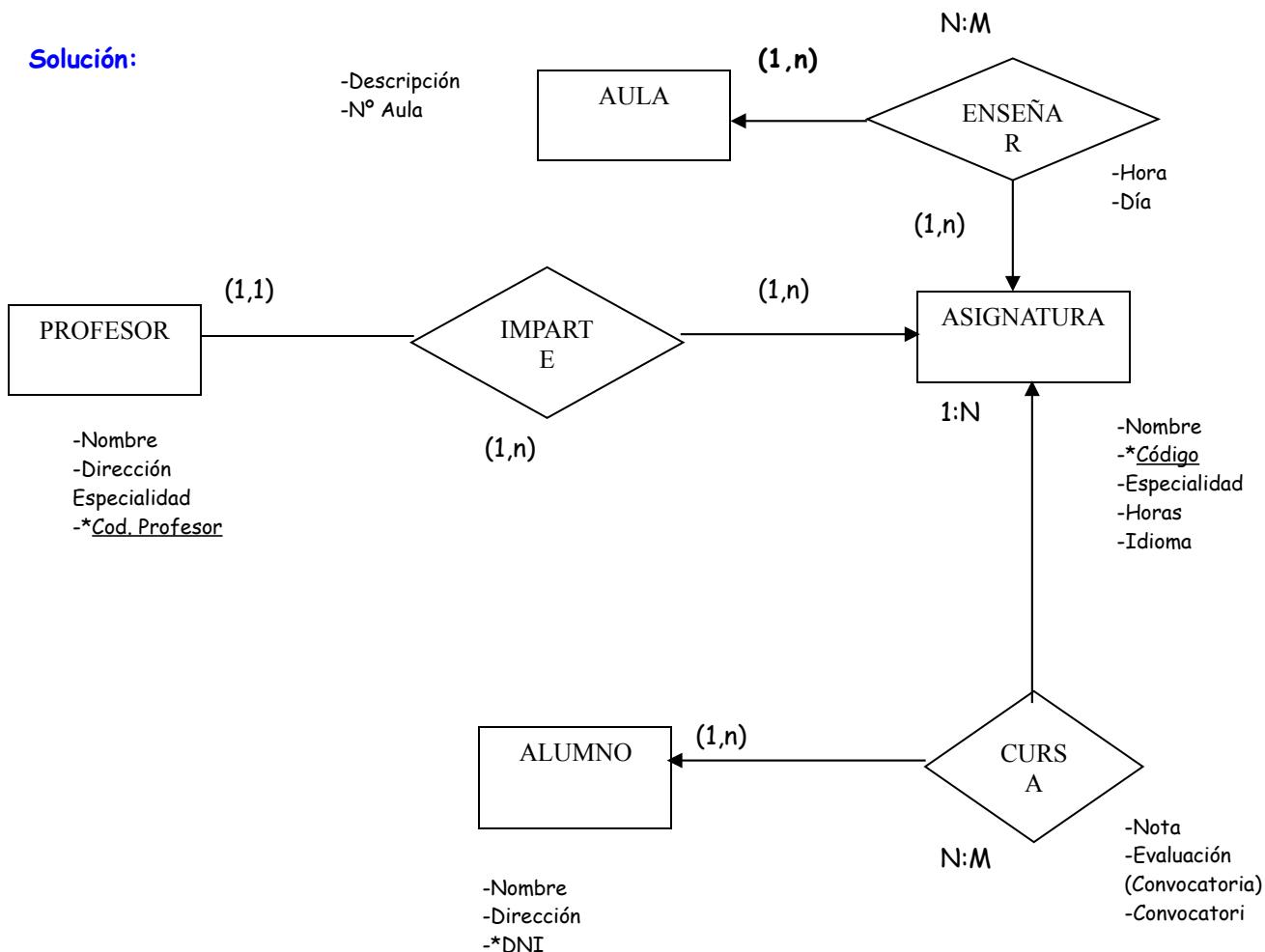


19.- Crea el modelo E/R para la base de datos de un hospital. En la base de datos se desea recopilar información sobre los **médicos** que trabajan en él y los **pacientes** a los que atienden. Cada paciente tendrá asignado un **historial** que queda registrado donde se recogerá información sobre todos los **análisis** que se le hayan hecho en su vida. Un paciente puede ser atendido por más de un médico.

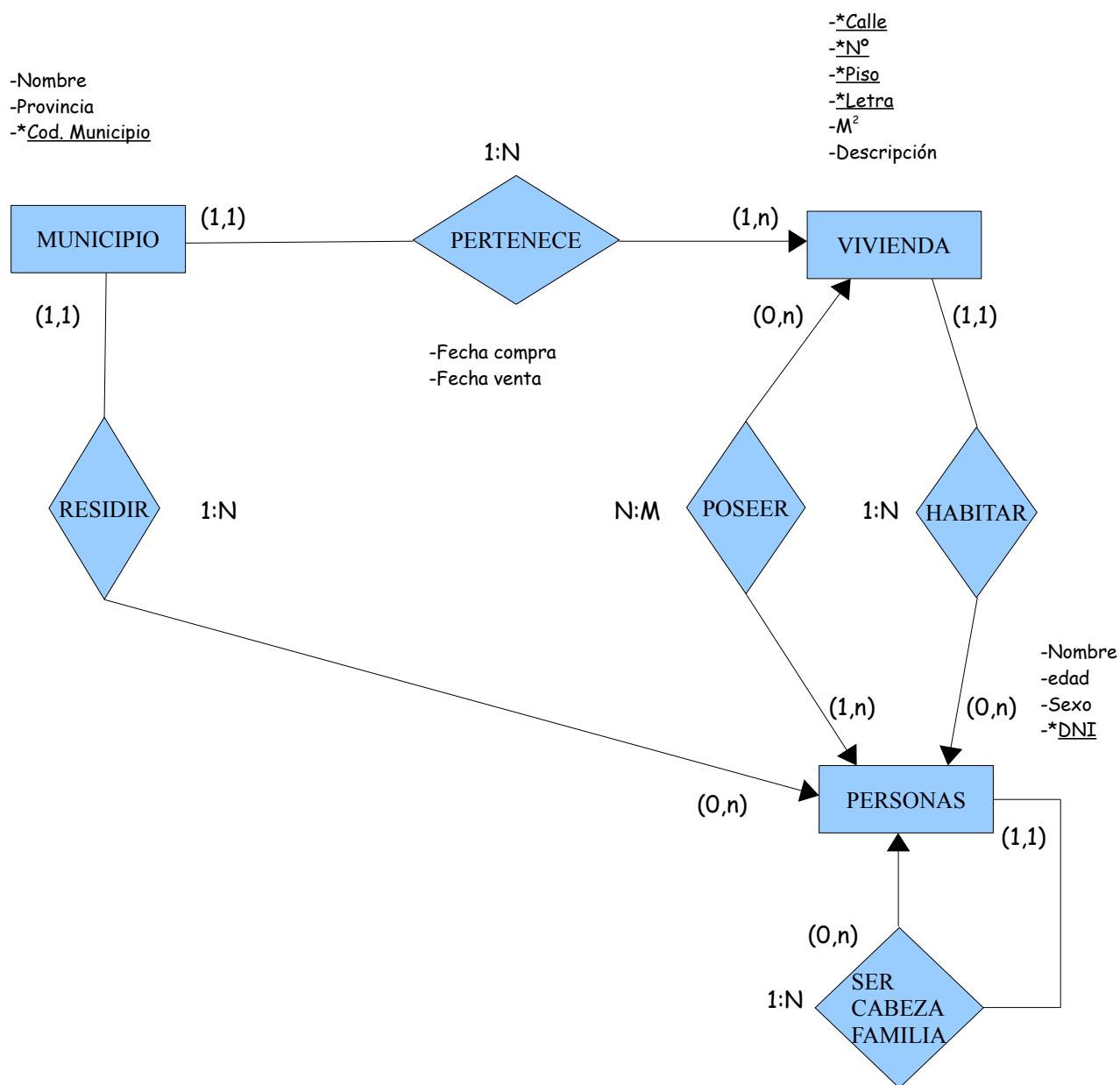
Solucion:



20.- Construye un diagrama E/R para la oficina de una universidad que mantiene información sobre cada una de las **asignaturas** impartidas, los **profesores** que las imparten, el nº de **alumnos** matriculados, y la hora y **aula** en la que se imparte cada asignatura. Por cada pareja estudiante-asignatura se registra una calificación. Además, se supone que cada asignatura la imparte sólo un profesor. Se desea guardar una descripción completa de cada aula.

Solución:

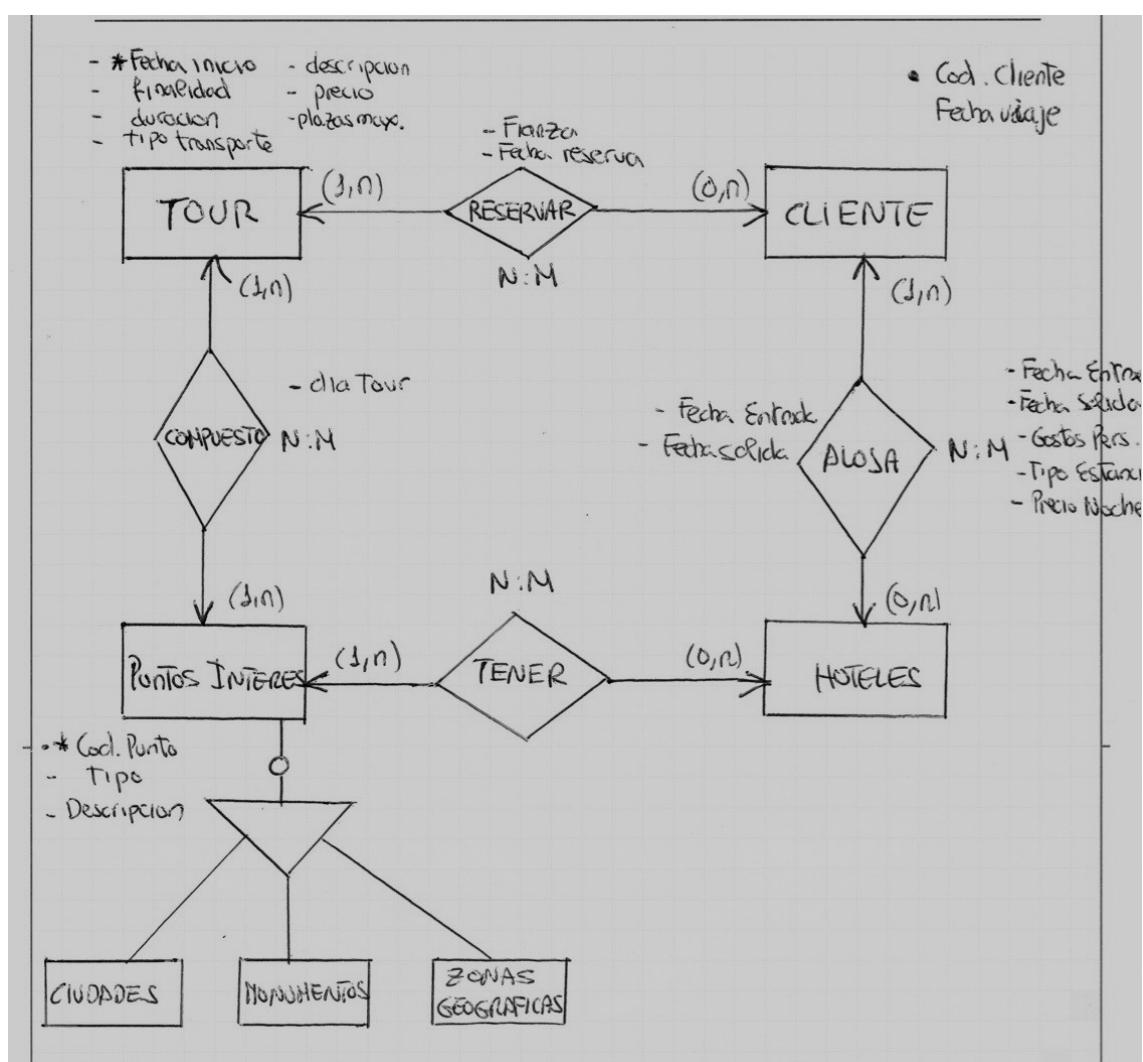
21.- Diseña un esquema E/R que recoja la organización de un sistema de información sobre **municipios**, **viviendas** y **personas**. Cada persona sólo puede habitar una vivienda y residir en un municipio, pero puede ser propietaria de más de una vivienda. Nos interesa también la interrelación de las personas con su cabeza de familia, se asume que no hay municipios sin viviendas, pero que puede haberlos sin habitantes.

Solución:

22.- Representa en un modelo E/R el diseño de una BD para una agencia de viajes que, para ofrecer mejor servicio a sus clientes, considera de interés tener registrada la información referente a los diferentes tours que se pueden ofrecer. Hay que tener en cuenta que:

- Un tour, según su finalidad cultural, histórica, deportiva, etc., tiene unos determinados puntos de ruta de interés.
- El mismo tour se puede repetir varias veces en un año.
- Definimos por viaje un tour concreto a realizar a partir de una fecha determinada.
- Los puntos de ruta de un tour pueden ser ciudades, monumentos o zonas geográficas. Cada punto se puede considerar de visita solamente, o de visita y estancia. En este último caso, el punto de ruta viene asignado a un hotel o a varios.
- Entendemos por cliente de un viaje a la persona que ha decidido hacerlo y ha hecho efectiva una señal.
- Un cliente puede confirmar su participación en más de un viaje, sobreentendiéndose que las fechas serán diferentes.
- Las personas que participan en un mismo viaje pueden estar alojados en el mismo o en diferentes hoteles.

Solución:

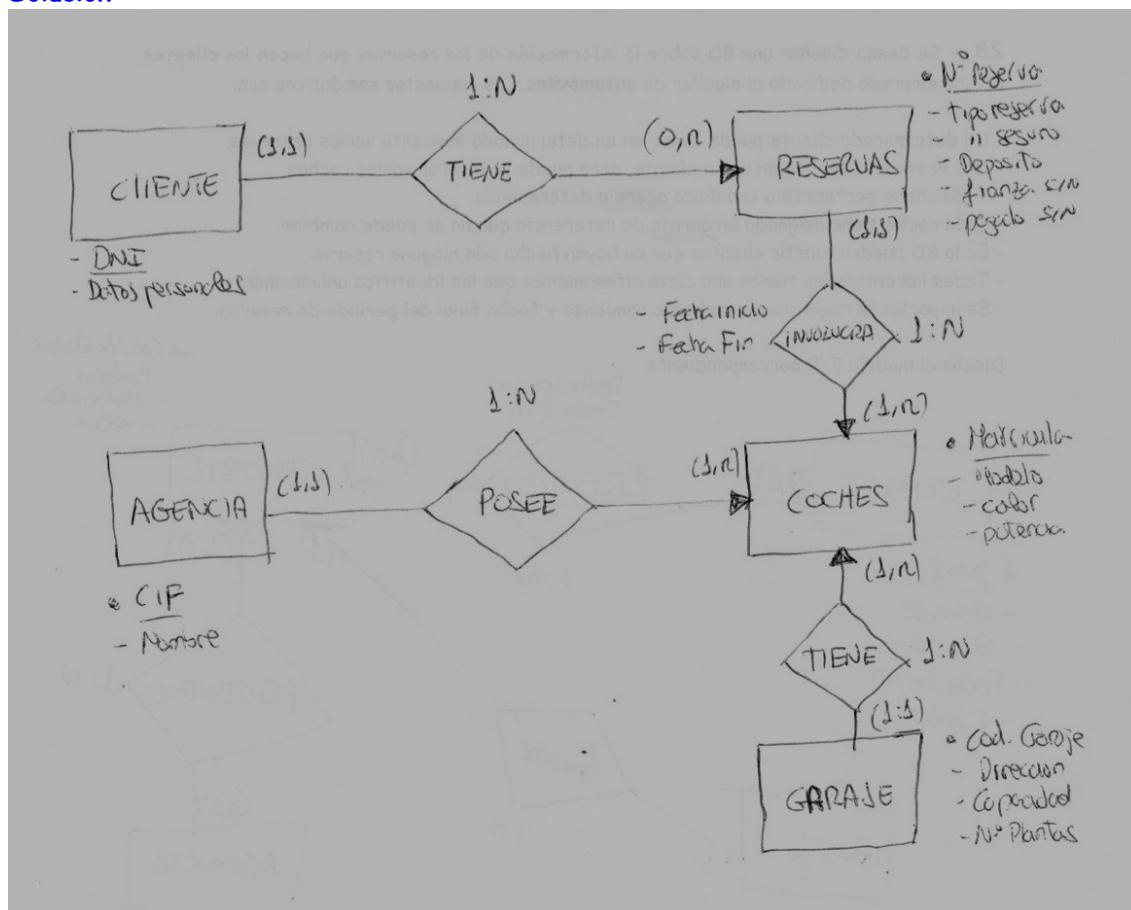


23.- Se desea diseñar una BD sobre la información de las reservas que hacen los **clientes** de una empresa dedicada al alquiler de **automóviles**. Los supuestos semánticos son:

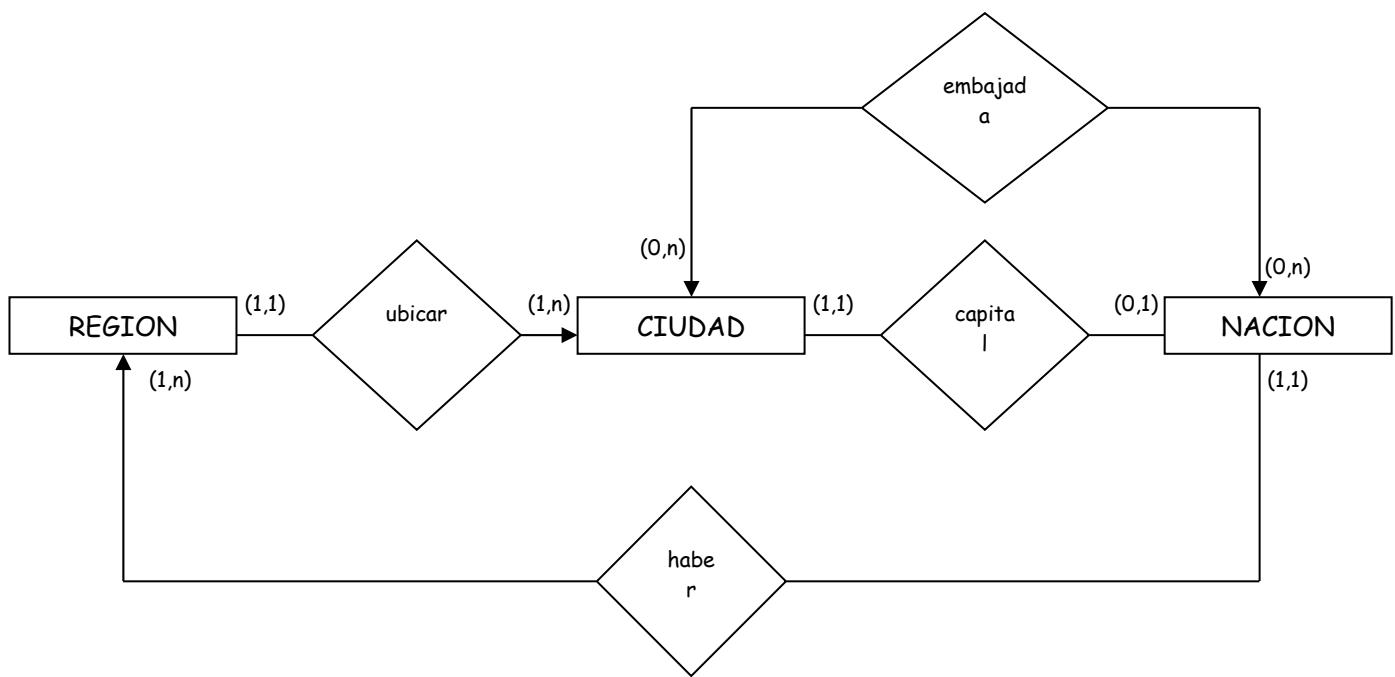
- Un determinado cliente puede tener en un determinado momento varias **reservas**.
- Una reserva la realiza un único cliente, pero puede involucrar varios coches.
- Todo coche pertenece a una única **agencia** determinada.
- Cada coche tiene asignado un **garaje** de referencia que no se puede cambiar.
- En la BD pueden existir clientes que no hayan hecho aún ninguna reserva.
- Todas las entidades tienen una clave alfanumérica que las identifica únicamente.
- Es importante registrar la fecha de comienzo y fecha final del periodo de reserva.

Diseña el modelo E/R correspondiente.

Solución:



24.- Analiza este diagrama ER y trata de detallar las especificaciones que le corresponden:



Una Nacion puede tener 1 o mas regiones.

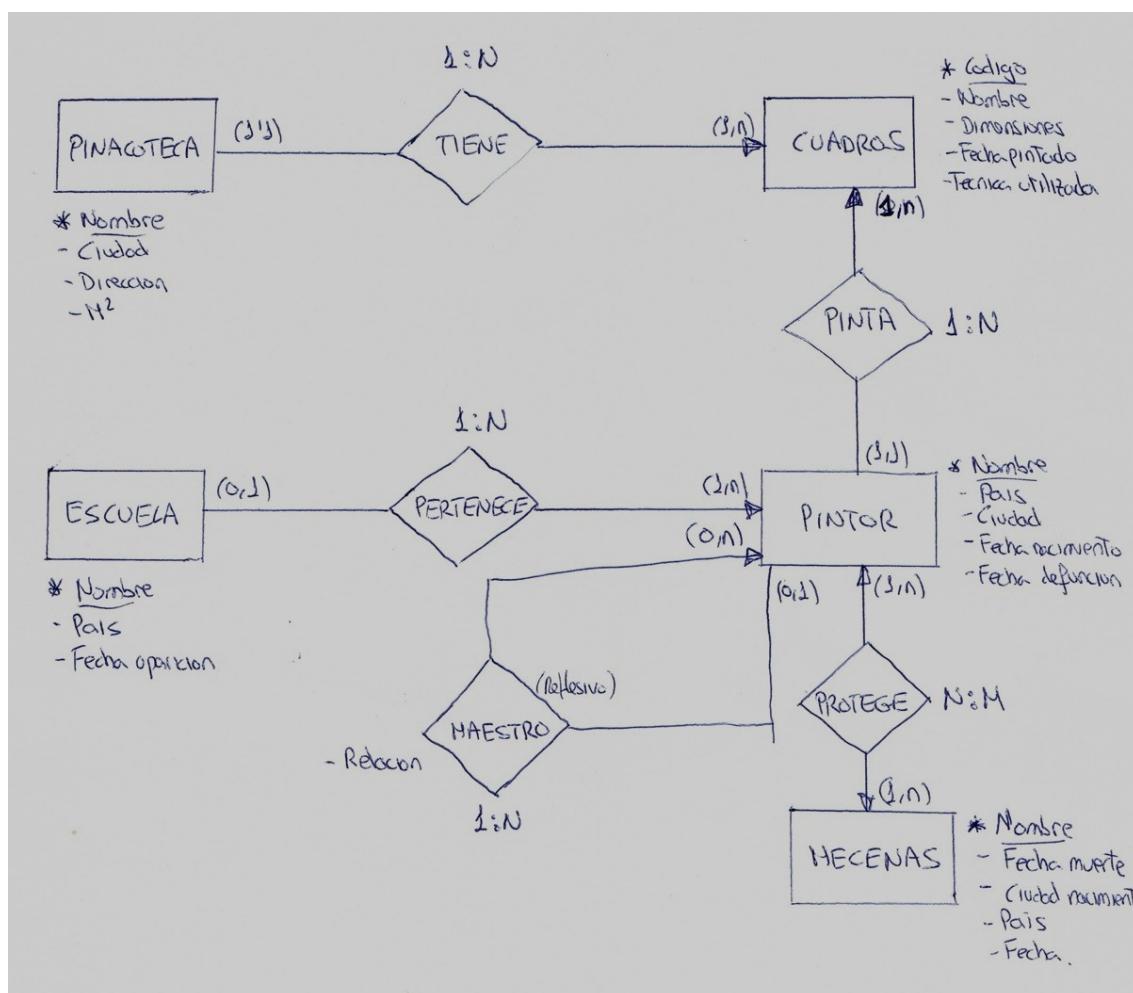
Una Nacion puede o no tener una o varias embajadas en una o varias ciudades.

Una Nacion tiene una ciudad como capital.

Las regiones tienen ciudades y pertenecen a una sola region.

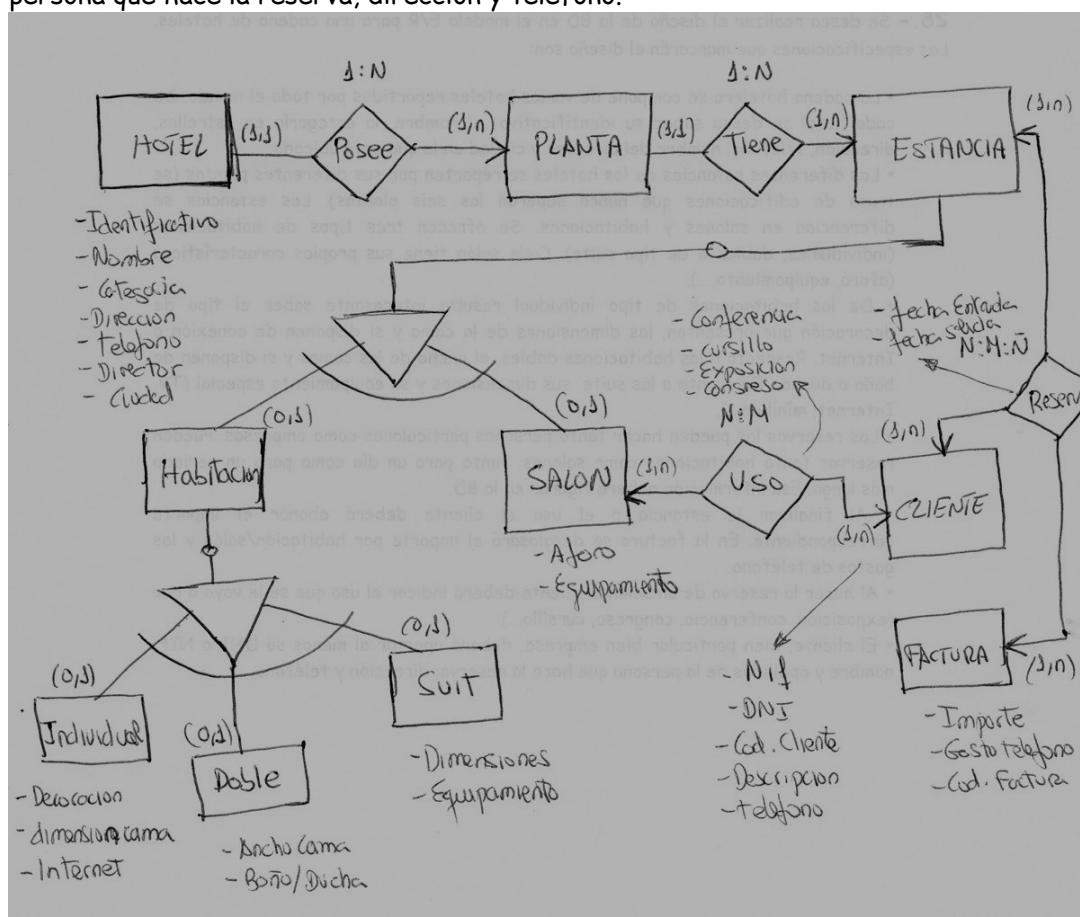
25.- El Ministerio de Educación y Ciencia desea mantener información acerca de todos los cuadros que se encuentran en las pinacotecas españolas y toda la información relacionada con ellos.

- De cada pinacoteca se desea saber el nombre (que se supone único), la ciudad en que se encuentra, la dirección y los metros cuadrados que tiene.
- Cada pinacoteca tiene una serie de cuadros de los que se quiere mantener información acerca de su código, nombre, dimensiones, fecha en que fue pintado y técnica utilizada.
- Cada cuadro es pintado por un determinado pintor (nombre, país, ciudad, fecha de nacimiento y fecha de defunción). Un pintor puede tener a otro como maestro; a su vez, un maestro puede serlo de varios (o de ninguno).
- Los pintores pueden pertenecer o no a una escuela de la que se desea saber su nombre y en qué país y en qué fecha apareció.
- Los pintores pueden tener también uno o varios mecenas que les protegen (nombre, fecha, país y ciudad de nacimiento y fecha de muerte). A su vez, un mismo mecenas puede serlo de varios pintores. Se desea saber cuál es la relación existente entre el pintor y su mecenas.



26.- Se desea realizar el diseño de la BD en el modelo E/R para una cadena de hoteles. Las especificaciones que marcarán el diseño son:

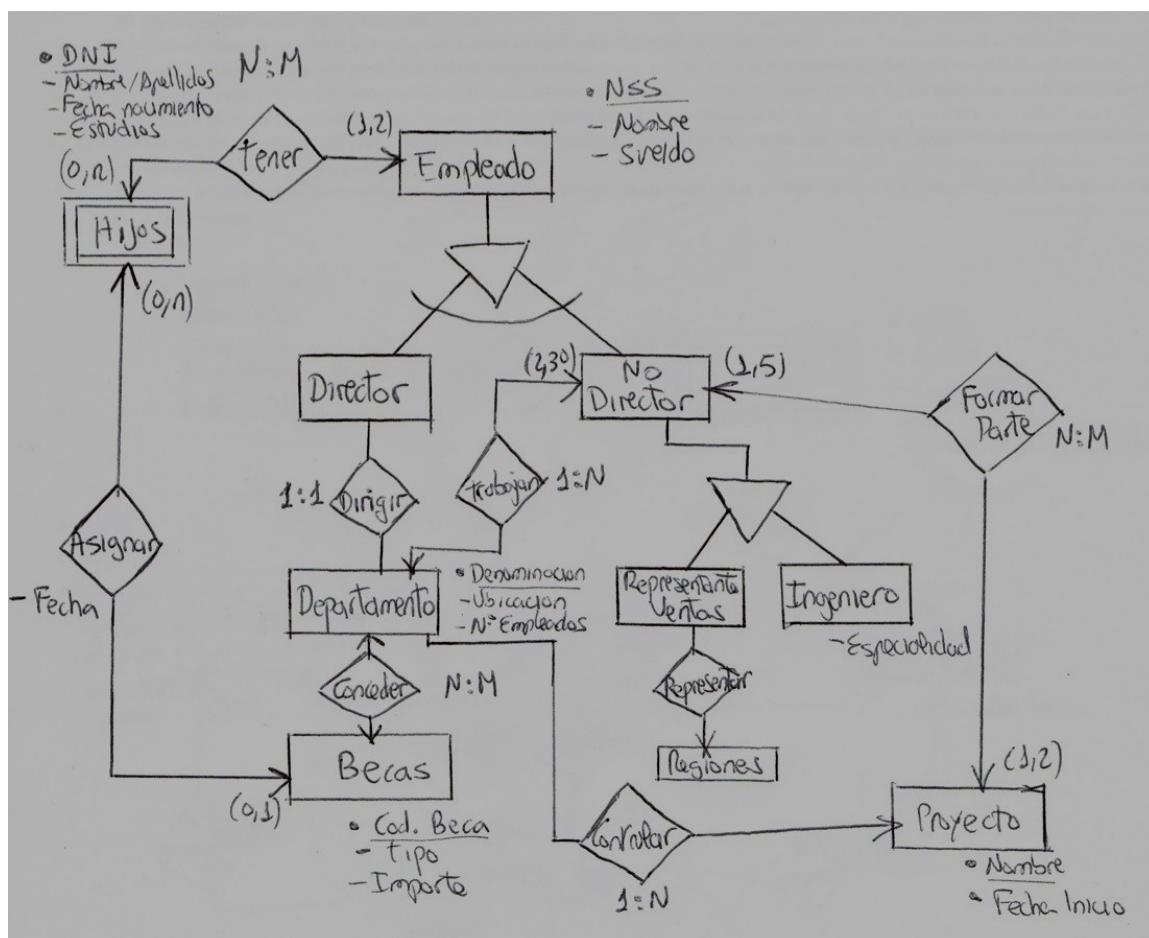
- La cadena hotelera se compone de varios hoteles repartidos por todo el mundo. De cada hotel se desea saber su identificativo, el nombre, la categoría en estrellas, dirección, teléfono, nombre del director y ciudad en la que está ubicado.
- Las diferentes estancias de los hoteles se reparten por sus diferentes plantas (se trata de edificaciones que nunca superan las seis plantas). Las estancias se diferencian en salones y habitaciones. Se ofrecen tres tipos de habitaciones (individuales, dobles o de tipo suite). Cada salón tiene sus propias características (aforo, equipamiento,...).
- De las habitaciones de tipo individual resulta interesante saber el tipo de decoración que presentan, las dimensiones de la cama y si disponen de conexión a Internet.
- Respecto a las habitaciones dobles, el ancho de las camas y si disponen de baño o ducha.
- En cuanto a las suite, sus dimensiones y su equipamiento especial (TV, Internet, minibar,...).
- De las habitaciones de tipo individual resulta interesante saber el tipo de decoración que presentan, las dimensiones de la cama y si disponen de conexión a Internet. Respecto a las habitaciones dobles, el ancho de las camas y si disponen de baño o ducha. En cuanto a las suite, sus dimensiones y su equipamiento especial (TV, Internet, minibar,...).
- Las reservas las pueden hacer tanto personas particulares como empresas. Pueden reservar tanto habitaciones como salones, tanto para un día como para un periodo más largo. Esa información deberá figurar en la BD.
- Al finalizar la estancia o el uso el cliente deberá abonar el importe correspondiente. En la factura se desglosará el importe por habitación/salón y los gastos de teléfono.
- Al hacer la reserva de un salón el cliente deberá indicar el uso que se le vaya a dar (exposición, conferencia, congreso, cursillo...).
- El cliente, bien particular bien empresa, deberá aportar al menos su DNI o NIF, nombre y apellidos de la persona que hace la reserva, dirección y teléfono.



27.- La empresa *ERABIL* desea llevar un control de sus departamentos, empleados y proyectos según las siguientes especificaciones:

- ✓ Se desea conocer el nombre, salario y número de la seguridad social de cada empleado, así como el nombre, fecha de nacimiento y estudios que cursa cada uno de sus hijos. Existen tres tipos de empleados: directores (encargados de un departamento), representantes de ventas (se ocupan de la representación en un número de regiones) e ingenieros (encargados de realizar los proyectos de la empresa). Un director no puede ejercer ninguna otra función; sin embargo, un representante de ventas puede desempeñar también las funciones de un ingeniero y viceversa.
- ✓ Los distintos departamentos, anualmente, conceden becas de estudio a los hijos de los empleados que trabajan en el mismo. Hay becas de diferentes tipos. Se desea conocer la fecha de concesión de cada beca así como la cuantía de ésta, cada hijo solo puede percibir una beca.
- ✓ Un ingeniero puede tener varias especialidades que se desean conocer.
- ✓ De los departamentos se necesita saber el nombre, localización y empleados que trabajan en él, considerando como empleados a todos los trabajadores aparte del director. Un departamento tiene, como mínimo, 2 empleados, y como máximo 30, y está al cargo de un único director. Cada departamento tiene un director distinto.
- ✓ Un departamento puede controlar un número de proyectos, de los que se desea conocer su nombre y fecha de comienzo.
- ✓ En la realización de un proyecto no puede haber involucrados más de 5 personas (sumando ingenieros y representantes). Todo ingeniero/representante debe estar asociado a 1 proyecto como mínimo y a 2 como máximo.

Haz el diagrama correspondiente en el modelo E/R indicando las cardinalidades que estimes oportunas.



28.- EL CATASTRO MUNICIPAL. Se desea considerar la información correspondiente al catastro de viviendas de un determinado municipio. En el municipio existe una serie de zonas urbanas en las cuales se han edificado un conjunto de viviendas, las cuales pueden ser:

- **Viviendas unifamiliares** o casas en las que sólo habita una familia.
- **Bloques de pisos** en los cuales existe un conjunto de viviendas, indeterminado a priori, en cada una de las cuales habita una familia.

Se considera mantener la información correspondiente a las personas que viven en cada una de las viviendas, así como el cabeza de familia de las personas que habitan o son propietarias de las viviendas.

Para cada vivienda, además de la información correspondiente a las características de las mismas, es necesario conocer la persona propietaria de la misma.

Se consideran, además, los siguientes supuestos semánticos en el problema:

1.- Toda persona habita en una y sólo una vivienda, la cual es considerada como su vivienda o residencia principal.

2.- Cada vivienda tiene uno y sólo un propietario.

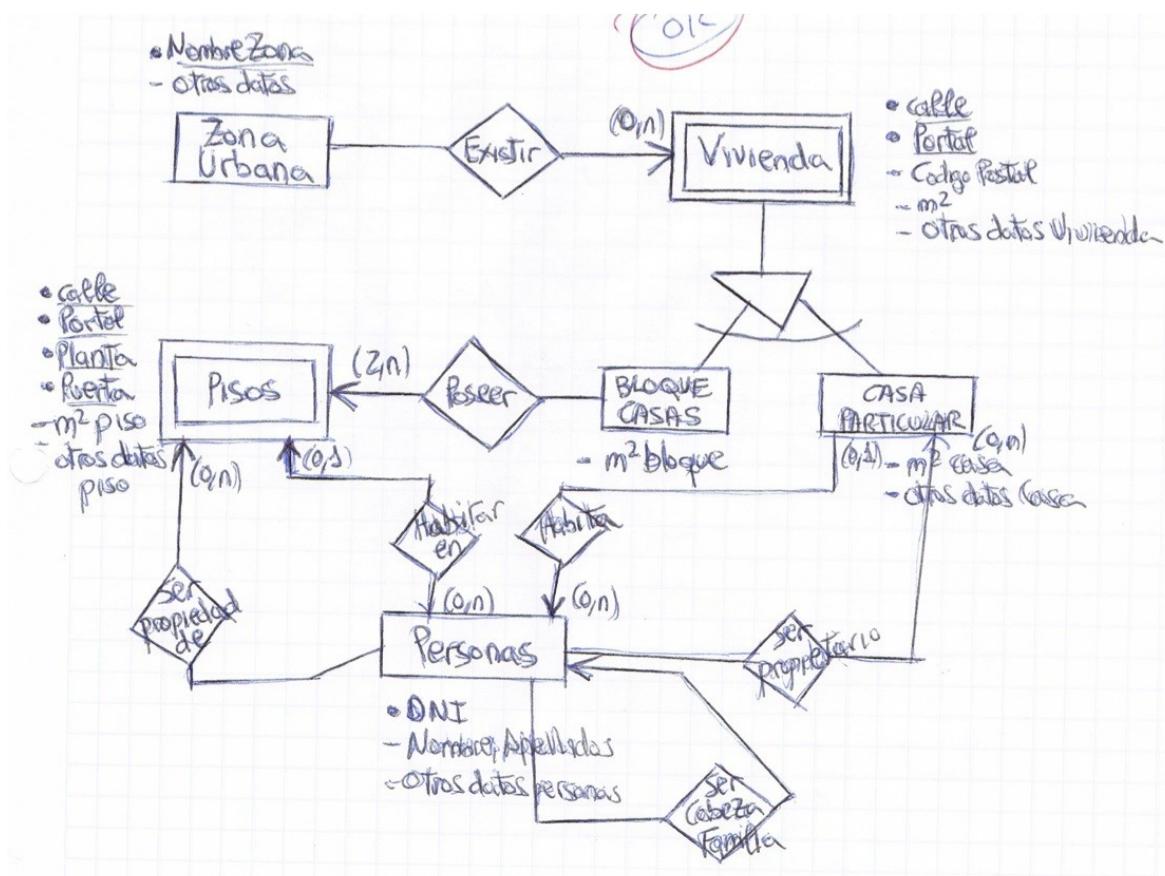
3.- Las viviendas se encuentran en una única zona urbana correspondiente al municipio.

4.- Las zonas urbanas en las que está dividido geográficamente el municipio tienen nombres diferentes.

5.- En cada zona urbana del municipio existen una serie de calles en las que se construyen las viviendas. Los nombres de las calles son únicos para cada municipio con independencia de la zona urbana en la que se encuentren.

6.- En el contexto del problema, una familia es un conjunto de personas que tienen una relación familiar directa y que habita, o no, en una misma vivienda. Este conjunto podrá ser unario.

7.- Como se indica en el enunciado del problema, las viviendas pueden ser casas unifamiliares o bloques en los cuales existen una serie de pisos individuales.



29.- LOS RESIDUOS TÓXICOS. Se desea abordar la problemática ambiental de los residuos tóxicos y peligrosos cuya incorrecta gestión produce daños de gran importancia en el medio ambiente y en la salud del ser humano. La información a contemplar es la que corresponde desde que es producido el residuo por un centro o empresa productora hasta que el residuo se encuentra en lugar seguro, donde los residuos reciben un tratamiento especial como puede ser la incineración, almacenamiento en depósitos de seguridad, etc. En el sistema de información se desea considerar la información de los productores de los residuos, los residuos, las empresas que transportan los residuos hasta los lugares seguros y el traslado de los residuos teniendo en cuenta el tipo de transporte, envase, etc.

Se consideran los siguientes supuestos semánticos en el problema:

- 1.- Una empresa productora produce un número amplio de residuos constituidos por un número variable de constituyentes químicos.
- 2.- Más de una empresa productora puede producir residuos con igual número de constituyentes químicos y con las mismas o distintas cantidades de los mismos.
- 3.- Las empresas productoras asignan un código único a los residuos que producen, lo que les permite diferenciar distintas producciones de los mismos productos. Además, más de una empresa puede asignar el mismo código a los residuos que produce.
- 4.- Los residuos pueden ser trasladados en su totalidad (cantidad total del mismo) o en partes, o no ser trasladados nunca.
- 5.- En cada traslado de residuos la cantidad que se traslada de los mismos es enviada a un único destino.
- 6.- En una misma fecha las empresas productoras pueden ordenar más de un traslado de un mismo o distinto residuo (cantidades parciales del mismo) a un mismo o distinto destino.
- 7.- En cada traslado puede intervenir más de una empresa transportista usando el mismo o distinto transporte, por lo que resulta interesante conocer tanto el medio de transporte utilizado como los kilómetros realizados, así como el coste del trabajo.
- 8.- El residuo se traslada en un tipo de envase determinado por la empresa productora y que no varía a lo largo del recorrido.
- 9.- Es interesante conocer la fecha de llegada a destino y el tratamiento a que se someten los residuos una vez alcanzado el mismo.
- 10.- Por seguridad, se considera que en un traslado sólo puede trasladarse un residuo de una empresa productora.

30.- El departamento de formación de una gran multinacional quiere tener un sistema de información para registrar los cursos que imparte y los asistentes a éstos.

Así, se pretende registrar la información del título, objetivos, programa y duración de cada uno de los cursos que imparte el departamento. También es necesario guardar información sobre los cursos preliminares de un curso, es decir, por ejemplo es necesario haber asistido a *Fundamentos de Programación I* antes de asistir a *Fundamentos de Programación II*.

Un curso se puede impartir varias veces, e incluso en distintos lugares y fechas. Se ha de registrar la oferta de cada curso, esto es: fecha, lugar y horario del curso, y los asistentes a cada uno de ellos, incluyendo la calificación obtenida por el alumno al finalizar el curso.

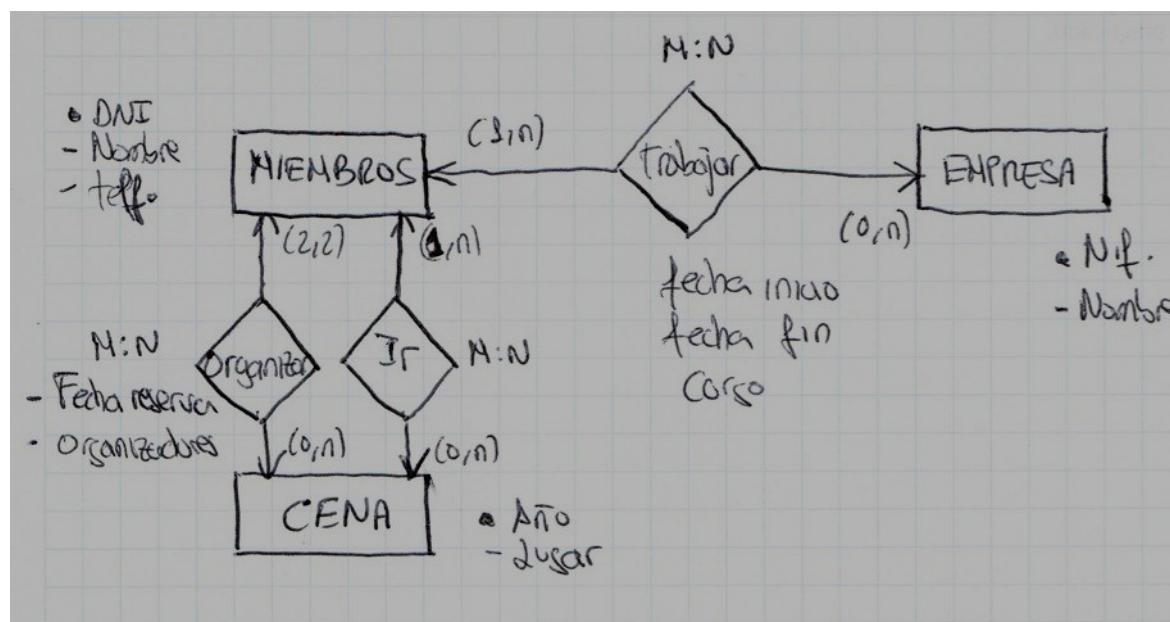
También se registrará el profesor que imparte el curso, que será único. Tanto el profesor como los asistentes son exclusivamente empleados de la compañía y, por tanto, se guardarán datos como el número del empleado, su nombre y dirección, titulación y cargo.

31.- Un grupo de ex-compañeros de estudios decide que sería interesante organizar una cena anual para mantener su amistad. La cena la organizarán dos personas del grupo que irán cambiando cada año.

También se pretende registrar la historia profesional de cada uno de los miembros, o sea, las empresas en las que han trabajado, en qué fechas y el cargo que han ocupado en cada una de ellas.

La aplicación deberá dar respuesta a consultas como:

- 1.- Teléfono de la casa de cada uno de los miembros del grupo.
- 2.-Lista de los que han trabajado en la empresa BURDINOLA.
- 3.-¿Dónde está trabajando actualmente Kattalin Moreno?
- 4.-Lista de los que no estuvieron en la cena del año pasado.
- 5.-¿Dónde tuvo lugar la cena del 2003 y quién la organizó?



32.- BEROTZ se dedica a la fabricación de frigoríficos. Se quiere crear un sistema informático que permita registrar los diseños de los modelos de frigoríficos realizados por la oficina técnica, así como los controles de validación de cada una de las piezas que contiene el frigorífico.

Cuando la oficina técnica diseña un nuevo modelo, le da un número de referencia y una descripción y, a continuación, indica las piezas que lo componen y en qué cantidad. Cada pieza viene descrita por un código de pieza y una descripción.

Las características de validación de cada pieza pueden ser: espesor de chapa, peso, resistencia, intensidad, voltaje, tiempo de arranque, presión,... Por cada una de estas características se indicará el valor máximo y mínimo entre los que debe estar la pieza considerada válida.

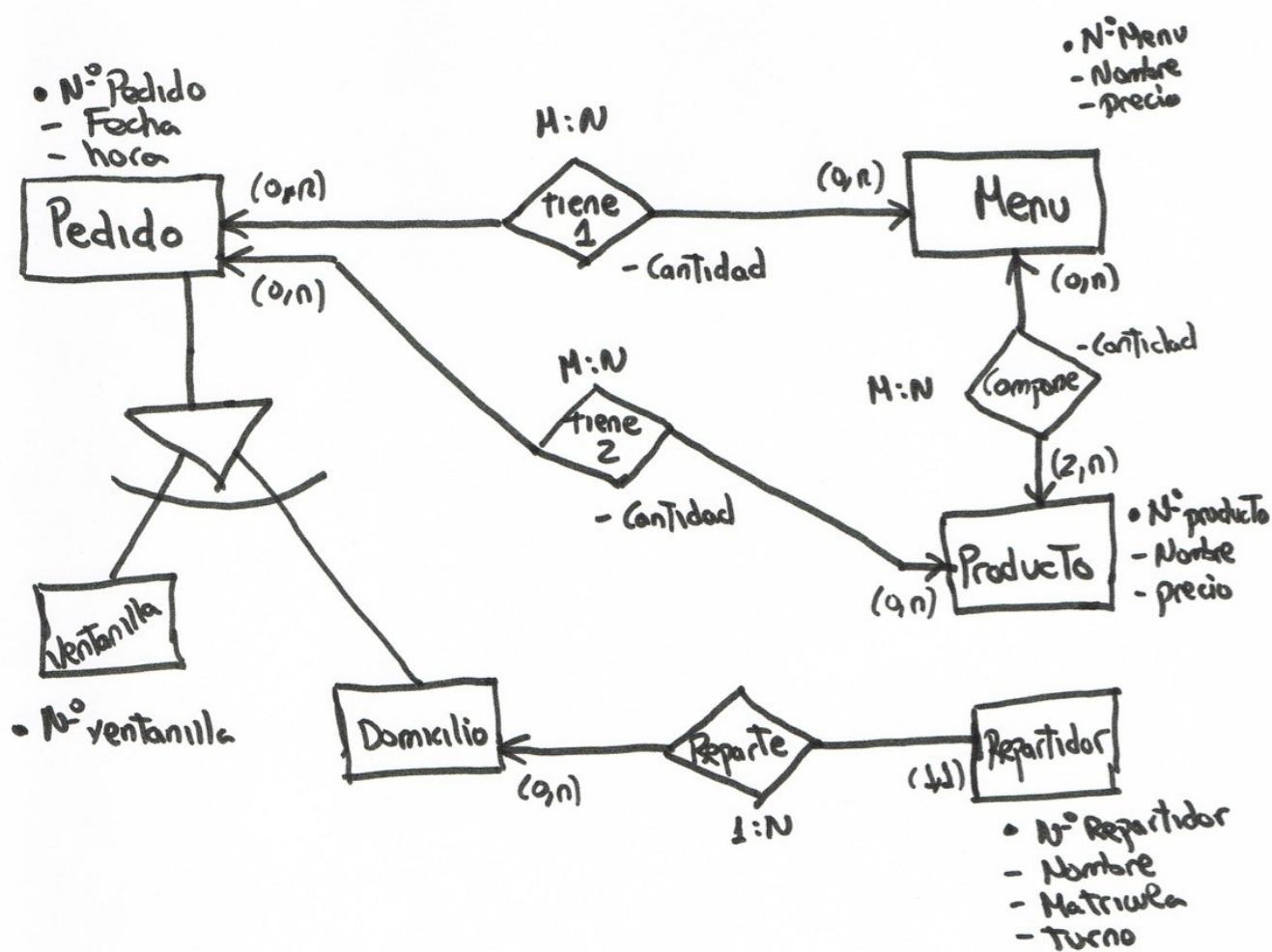
Por otra parte, la CEE obliga a llevar un registro de cada una de las piezas de que se compone el frigorífico; a esto se le llama *trazabilidad*. Ante un fallo de un frigorífico ha de demostrarse que todas las piezas que lo componen dieron valores válidos en la medida de sus características.

Por lo tanto, BEROTZ registrará todos los frigoríficos fabricados con un número de serie, fecha de fabricación, línea de montaje, turno, etc. Registrará también cada una de las piezas que se montaron en el frigorífico y el valor que obtuvieron en los test que se le han practicado.

33.- Un establecimiento de la cadena McDONALDS quiere desarrollar un software para gestionar los pedidos y repartos a domicilio, así como obtener estadísticas de los mismos.

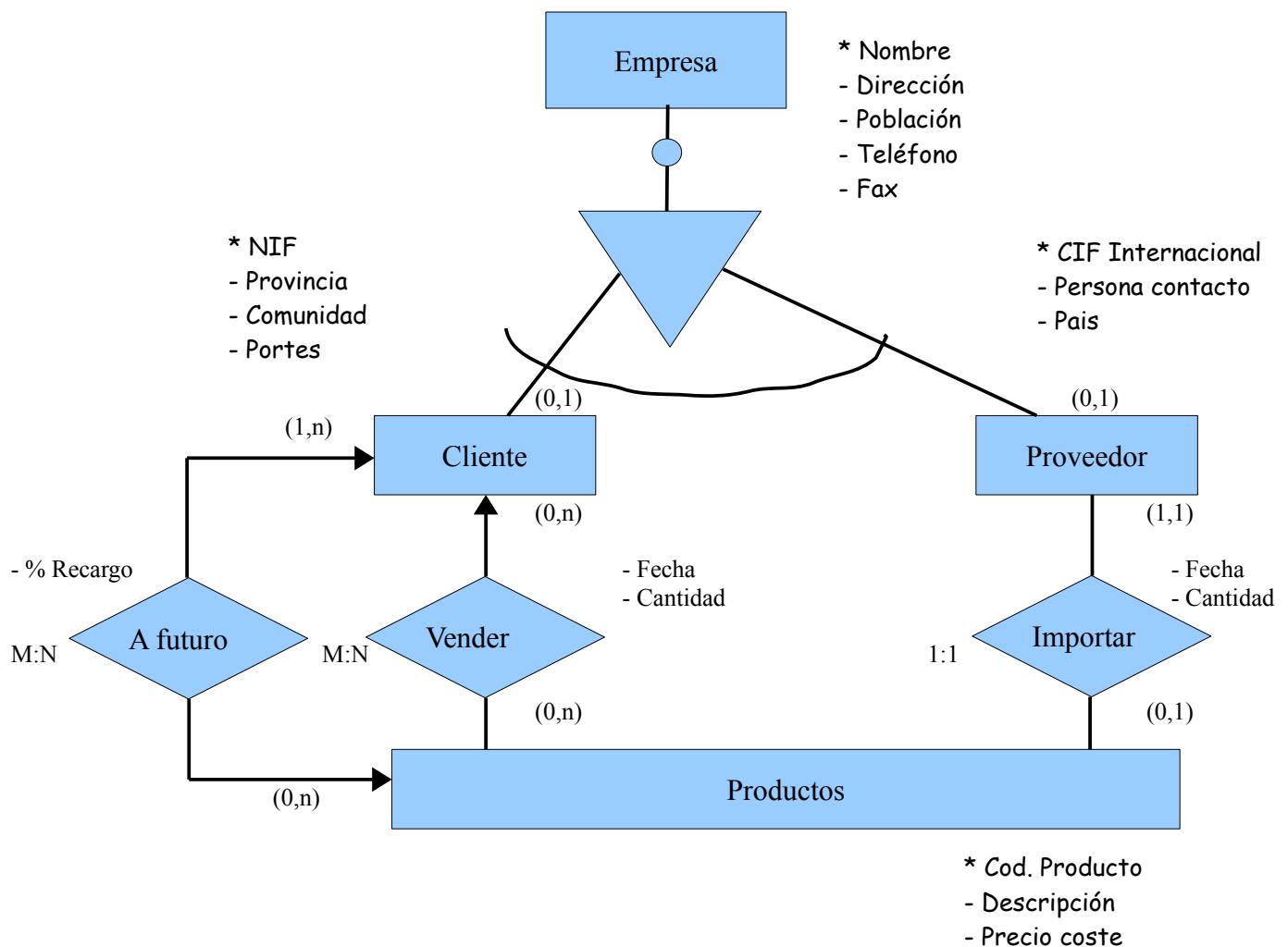
Se quiere registrar la siguiente información:

- Los pedidos para consumir en el establecimiento se recogen en ventanilla y los de entrega a domicilio se recogen por teléfono
- Todos los pedidos van numerados correlativamente. Por cada uno de ellos se recoge la fecha y hora del pedido
- En los pedidos de ventanilla se registra el nº de ventanilla
- En los pedidos a domicilio se registra el teléfono, población y dirección
- Se cuenta con una serie de repartidores identificados por un número, con el nombre completo, una matrícula de moto y un turno (mañana, tarde, noche)
- A cada pedido a domicilio se le asigna un repartidor
- El establecimiento ofrece una serie de productos (hamburguesa simple, doble, alemana, coca-cola pequeña mediana, grande, ensalada...) identificados por un código y con un nombre y precio
- Se ofrecen también menús especiales compuestos por un conjunto de productos. Los menús están numerados, tienen un nombre de menú y un precio
- Un pedido puede constar de varios menús en diferentes cantidades y/o varios productos en diferentes cantidades



34.- Una empresa de importación desea informatizar la gestión de compra/ventas de productos. Compra diversos productos a empresas extranjeras y los vende a empresas nacionales.

- Por cada empresa necesita NIF, dirección, población, teléfono y fax.
- Cada empresa es cliente o proveedora.
- Para las empresas clientes necesita saber además la provincia, comunidad autónoma y portes (debidos o pagados).
- Para las empresas proveedoras necesita saber el país, CIF internacional y nombre de la persona de contacto.
- Por cada producto tenemos código, descripción y precio de coste.
- Cada proveedor provee un único producto, y cada producto es provisto por un único proveedor. En el momento de contactar con un nuevo proveedor se guardará constancia del mismo y el producto que le debemos comprar.
- Tras cerrar el acuerdo con los clientes guardaremos constancia de qué productos nos demandará en el futuro y cual será el porcentaje de recargo a aplicar en cada producto.
- Se desea guardar constancia de cada compra y venta con la fecha y cantidad de productos comprados/vendidos.
- Pueden existir proveedores y/o clientes con los cuales tengamos firmados acuerdos y aún no hayamos realizado ninguna compra y/o venta.



35.- CATERING. La empresa ON EGIN quiere desarrollar un sistema de información para gestionar su funcionamiento.

El futuro sistema deberá registrar todas y cada una de las empresas que contratan los servicios de ON EGIN para actos especiales (comidas de empresa, cenas de cuadrilla, lunch, bodas,...). De cada una de estas empresas se registrará el nombre de la empresa, la dirección y un teléfono de contacto.

-Cada empresa solicitará al menos un presupuesto antes de contratar el servicio para un acto. Se desea registrar la fecha en que se ha solicitado el presupuesto.

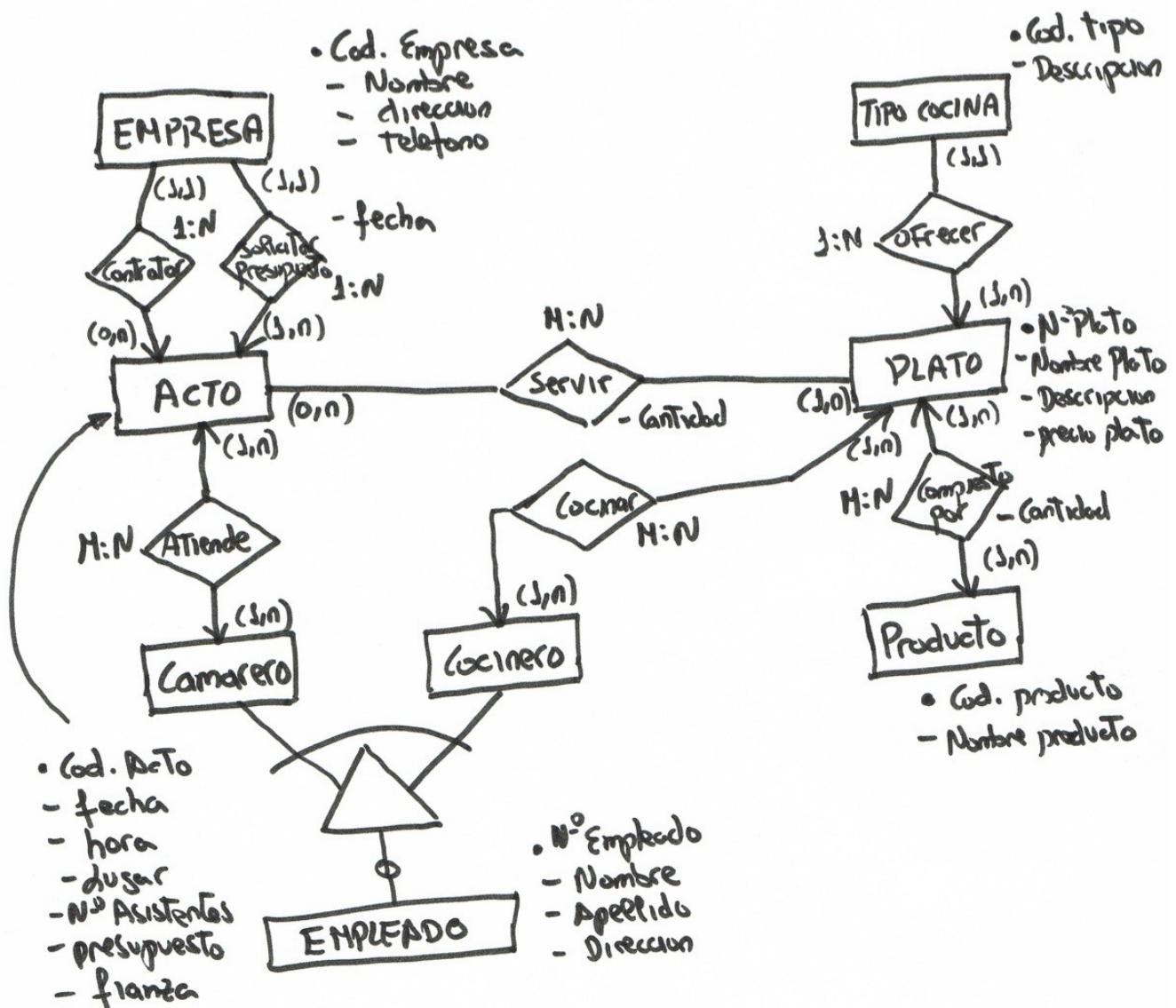
-Los actos especiales se celebran en una fecha, hora y lugar concretos y con un determinado número de asistentes. Además serán atendidos por diversos camareros pertenecientes a la plantilla de ON EGIN.

-Se desea así mismo guardar información de los platos que se sirven en estos actos y en qué cantidad. Cada plato está definido por un nombre, una descripción y un precio. Estos platos pueden ser de diversos tipos, así que se cocinan platos orientales, comida italiana, vegetariana, etc. Un plato determinado podrá ser únicamente de un tipo de cocina.

-Los platos que se cocinan están compuestos por diversos productos y en ciertas cantidades. De cada producto conocemos su nombre.

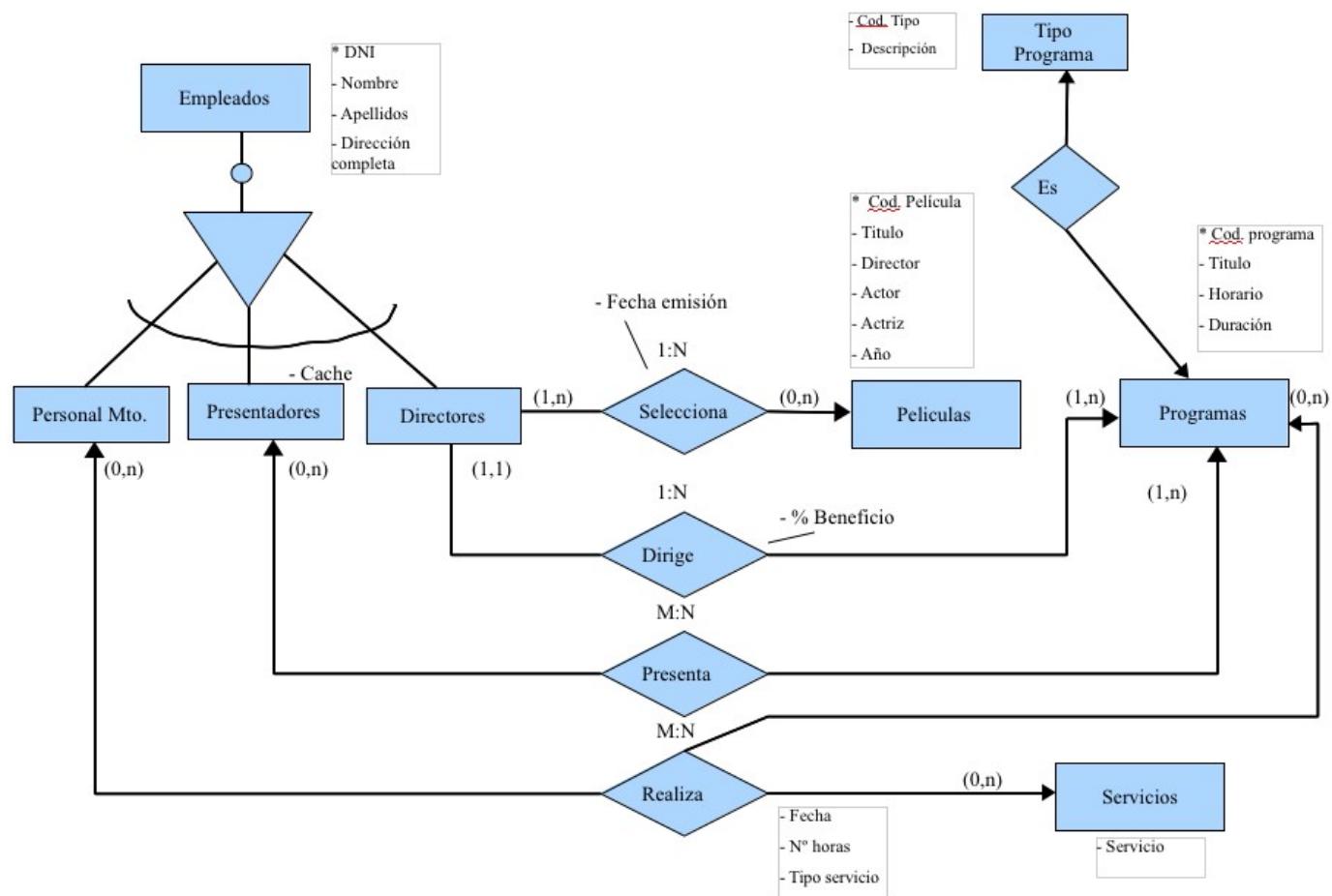
-La empresa ON EGIN dispone de cocineros de diferente grado (chefs, ayudantes de chef, ayudante de cocina, repostero...) que son los encargados de cocinar los diversos platos.

-De todos los empleados de la empresa se desea registrar su nº de empleado, nombre, apellido y dirección.



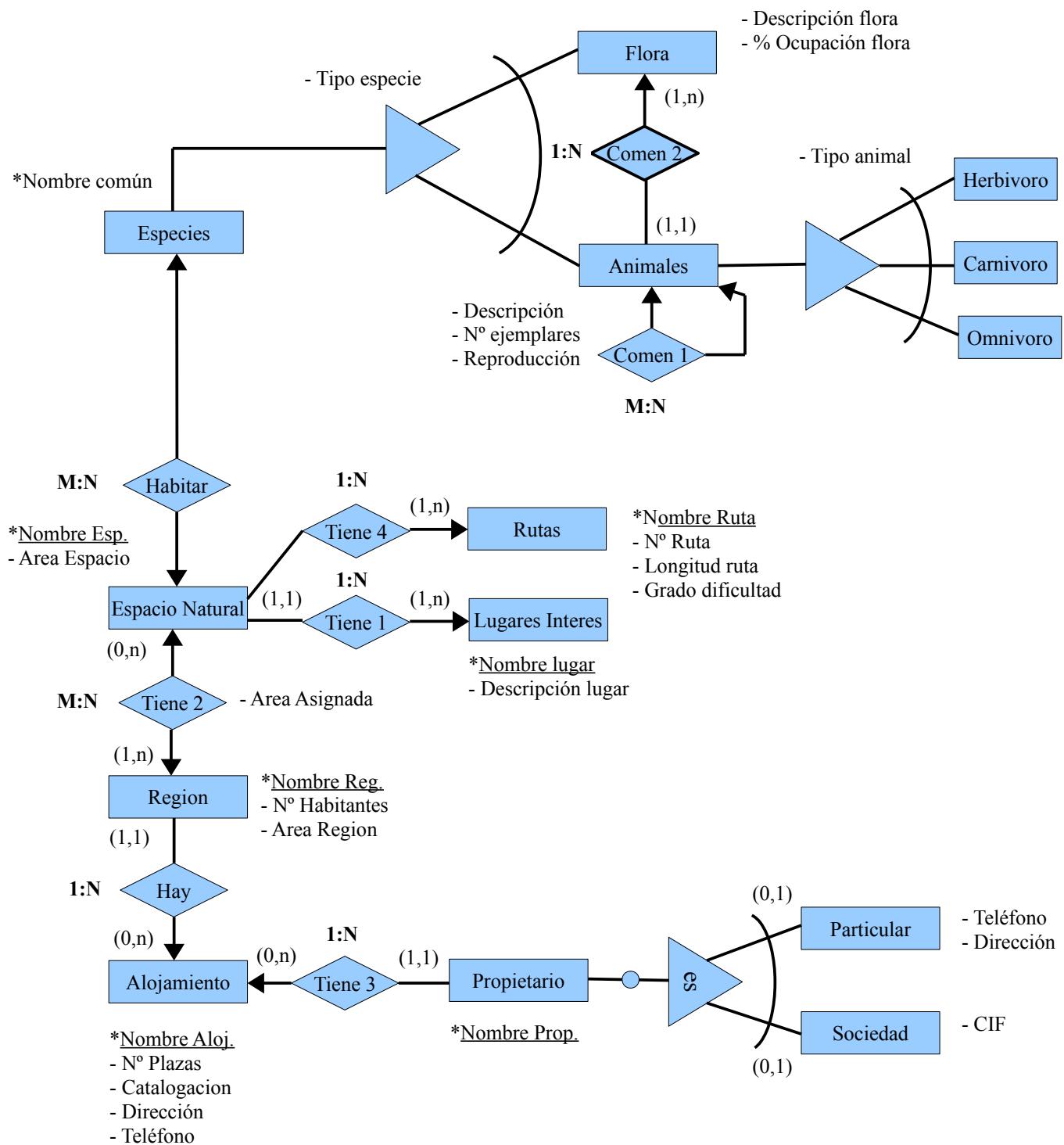
36.- CADENA DE TELEVISIÓN. Una cadena televisiva quiere realizar un sistema informático para registrar información sobre sus empleados y la programación emitida.

1. En esta cadena se emiten tanto programas como películas.
2. La información que se desea registrar de cada película es: título, director, actor, actriz y año.
3. La información que se desea registrar de los programas es: código del programa, título, horario y duración.
4. Existen distintos tipos de programas (documentales, informativos, concurso,...) y de cada uno de ellos conocemos su código y descripción.
5. Las películas que se emiten son seleccionadas por los distintos directores de la cadena. Además los directores dirigen programas, obteniendo por ello un porcentaje de beneficio por programa.
6. Entre los empleados de la empresa, además de directores, existen presentadores y personal de mantenimiento.
7. De cada empleado se desea registrar: DNI, nombre, primer apellido, segundo apellido y dirección completa (calle, nº, piso, CP y población).
8. Los presentadores son los encargados de presentar los distintos programas de la cadena. Existen casos en los que un programa es presentado conjuntamente por más de un presentador, y otros programas que no necesitan presentador.
9. Cada presentador tiene un caché.
10. El personal de mantenimiento realiza servicios sobre los programas, por ejemplo, montaje del decorado, control de sonido, iluminación... Un empleado de mantenimiento puede realizar varios servicios y viceversa.
11. De cada servicio se guarda la fecha en la que se realizó, el nº de horas empleado y el tipo de trabajo realizado.



37.- Debido al auge del turismo rural y ecológico se quiere hacer una aplicación capaz de mantener información relacionada con este tipo de turismo. Aparte, se mantendrá también información científica. A continuación se describe la información que se quiere mantener, en concreto toda la información gira en torno a los espacios naturales existentes en un determinado país:

1. De todo espacio natural interesa saber su nombre y su área.
2. Este país está dividido en regiones semejantes a los términos municipales españoles. Una región es la superficie perteneciente a una población de este país. De una región se quiere saber su nombre, el número de habitantes y el área que ocupa.
3. La distribución de los espacios naturales es independiente de las regiones, de forma que el área de un espacio natural puede estar distribuida en varias regiones, interesando guardar esta información así como el área correspondiente a cada región de cada espacio natural.
4. En cada población pueden existir alojamientos sobre los que se desea guardar información como: el nombre, el número de plazas, la catalogación del alojamiento, su dirección y el teléfono de contacto.
5. Interesa saber el propietario del alojamiento. Un alojamiento o es de un particular (en cuyo caso interesa saber su nombre, teléfono y dirección) o es de una sociedad (de la que interesa saber su nombre y su CIF). Tanto un particular como una sociedad pueden ser propietarios de varios alojamientos, pero un alojamiento es propiedad de una sola persona o sociedad.
6. Todo espacio natural tiene algún lugar de interés que visitar y cuenta con rutas para llegar a éstos. Sólo interesa guardar información sobre las rutas y los lugares ubicados en los espacios naturales.
7. De cada lugar de interés se desea guardar su nombre y una breve descripción del lugar. No existen dos lugares de interés con el mismo nombre. Además un lugar de interés sólo podrá pertenecer a un espacio natural.
8. En cuanto a las rutas hay que decir que se guarda un número, su nombre, su longitud en km. y su grado de dificultad. Tanto el número como el nombre pueden diferenciar una ruta del resto de rutas de un mismo espacio natural. Una ruta sólo pertenece a un espacio natural.
9. En cuanto a la información científica que se desea guardar, ésta consta de la fauna y la flora existentes en cada espacio natural. De cada especie vegetal que conforma la flora de un espacio se quiere mantener información como el nombre común, una descripción y el porcentaje de ocupación del terreno de los espacios naturales donde se encuentre. En cuanto a la fauna, la forman las especies animales existentes en cada espacio, de las que interesa guardar su nombre común, una descripción, información sobre su reproducción y el número de ejemplares existentes en cada espacio en el que se encuentra.
10. Aunque pueden existir especies propias de un espacio natural que no existan en ningún otro, se pueden dar casos de especies que existan en distintos espacios naturales.
11. Interesa guardar información sobre la pirámide alimentaria existente guardando información de la especies vegetales que son consumidas por cada especie animal herbívora existente, y de las especies animales que son consumidas por las especies animales carnívoras existentes.
12. Existen especies omnívoras capaces de alimentarse tanto de especies vegetales como animales. De ellas también se quiere guardar información



38.- Una empresa que regenta varios bares de copas nocturnos en una ciudad desea generar un sistema de información que le permita conocer qué personal trabaja en qué locales, entre otros datos.

El modelo de datos deberá recoger los siguientes puntos:

1. De cada local se desea saber su nombre, dirección, dimensiones, horario, teléfono y descripción.
2. Un local requiere el trabajo de varios empleados todas las noches, pero un empleado puede trabajar diferentes noches en diferentes locales, cubriendo diferentes puestos. Interesa conocer qué puesto ocupó cada empleado cada noche durante los últimos dos meses.
3. De cada empleado se desea conocer su DNI, datos personales, edad, sexo y teléfono de contacto.
4. Cada local siempre tiene un jefe o encargado que es un empleado más.
5. Cada local puede contratar empresas de servicios para que desarrollen determinadas tareas puntuales. De cada empresa de servicios se desea conocer su NIF, razón social, teléfono y persona de contacto. Por supuesto, cada una de estas empresas puede trabajar en más de un local.
6. Los puestos de trabajo que requieren los locales varían de un local a otro. Cada puesto se identifica con un código y una descripción. En realidad, hay cuatro tipos de puestos: porteros, camareros, pinchadiscos y relaciones públicas. De cada uno de ellos se pretenden registrar sus características particulares. En determinados locales lo que se considera un puesto recoge funciones de más de un tipo.
7. El modelo debe recoger los puestos que requiere cada local, dado que no todos los locales necesitan los mismos puestos.
8. Los empleados pueden ser propios, esto es, contratados por la empresa, o externos, es decir, trabajadores de empresas de trabajo temporal contratados eventualmente. En este último caso habrá que saber de cuál. También se recogerá información acerca de las distintas ETT's.

Representa el modelo ER.

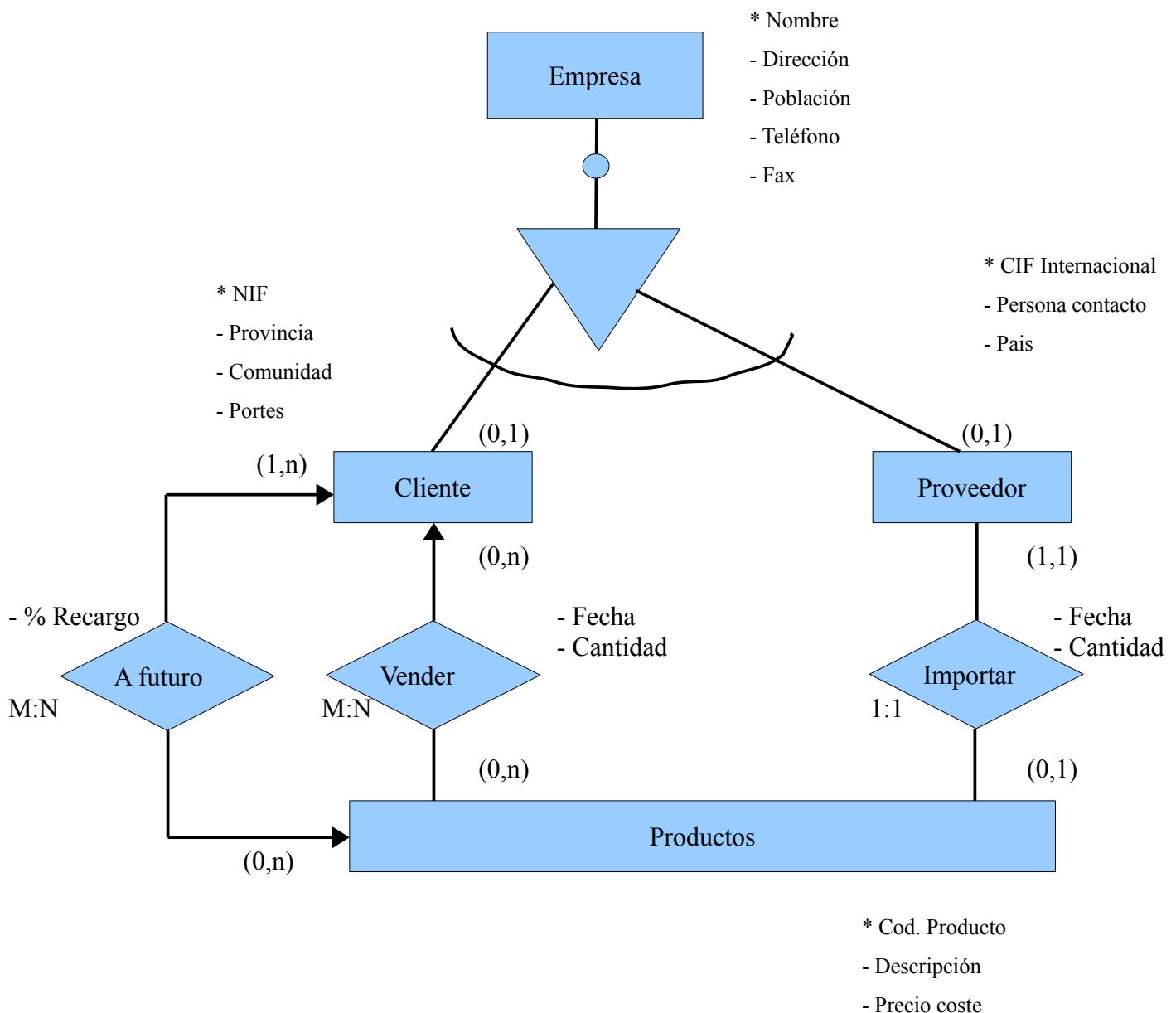
39. – Una agencia de viajes, situada en un lugar de alto interés turístico, desea desarrollar un sistema de información que le permita analizar la naturaleza de las reclamaciones que reciba por parte de sus clientes. El universo del discurso queda resumido en los siguientes puntos:

1. Entendemos por viaje la realización de un determinado tour turístico en una fecha determinada.
2. La agencia realiza viajes por los diferentes destinos de interés de la zona. Un viaje puede ser terrestre o marino, no existen viajes mixtos.
3. Todos los viajes terrestres necesitan un guía turístico. Los guías pueden serlo de diferentes viajes.
4. Todos los viajes terrestres necesitan de uno o varios vehículos diferentes para poder ser realizado. Esto es, es posible el desplazamiento en autobús hasta cierto punto a partir del cual el viaje continúa con motocicleta.
5. Los tipos de vehículos terrestres habilitados para los viajes son los siguientes: motocicletas, vehículos 4x4 y autobuses.
6. Tanto los viajes terrestres como los marinos pueden incluir una comida; en caso de que así sea, se necesita recoger la información relativa al restaurante.
7. Los viajes marinos requieren un transporte marítimo, que será el único utilizado en todo el viaje. Estos transportes sólo pueden pertenecer a uno de estos tres tipos: barcos de época, lanchas motoras o yates.
8. Un viaje lo realiza un conjunto de clientes. Un cliente puede realizar varios viajes.
9. Un cliente puede presentar a la organización varias reclamaciones relativas a un mismo o a distintos viajes que haya realizado. Las reclamaciones se realizan a título personal, no admitiéndose reclamaciones colectivas. Es necesario, para cada reclamación, conocer qué cliente la originó, y cuál fue el viaje que lo provocó.
10. Para algunos clientes, se guarda una ficha personal con sus datos. Las fichas están codificadas por el propio código de cliente.
11. Una vez realizado un viaje, un cliente permanecerá registrado en la base de datos durante dos años, después de los cuales deberá desaparecer automáticamente.
12. Es necesario conocer, para cada reclamación, qué elementos fueron los desencadenantes de la queja. Por supuesto, una reclamación sólo puede estar relacionada con una ocurrencia de cada elemento como máximo, pero un elemento puede ser motivo de varias reclamaciones.
13. No es posible que existan reclamaciones que recojan quejas de elementos incompatibles; esto es, una reclamación no puede serlo a la vez del guía y del equipo de buceo, dado que no existen viajes que requieran los dos elementos.

- Utiliza el modelo ER para diseñar el modelo conceptual correspondiente a las especificaciones anteriores.
- ¿Qué especificaciones NO se pueden recoger en el modelo ER?
- ¿Hay alguna especificación redundante?

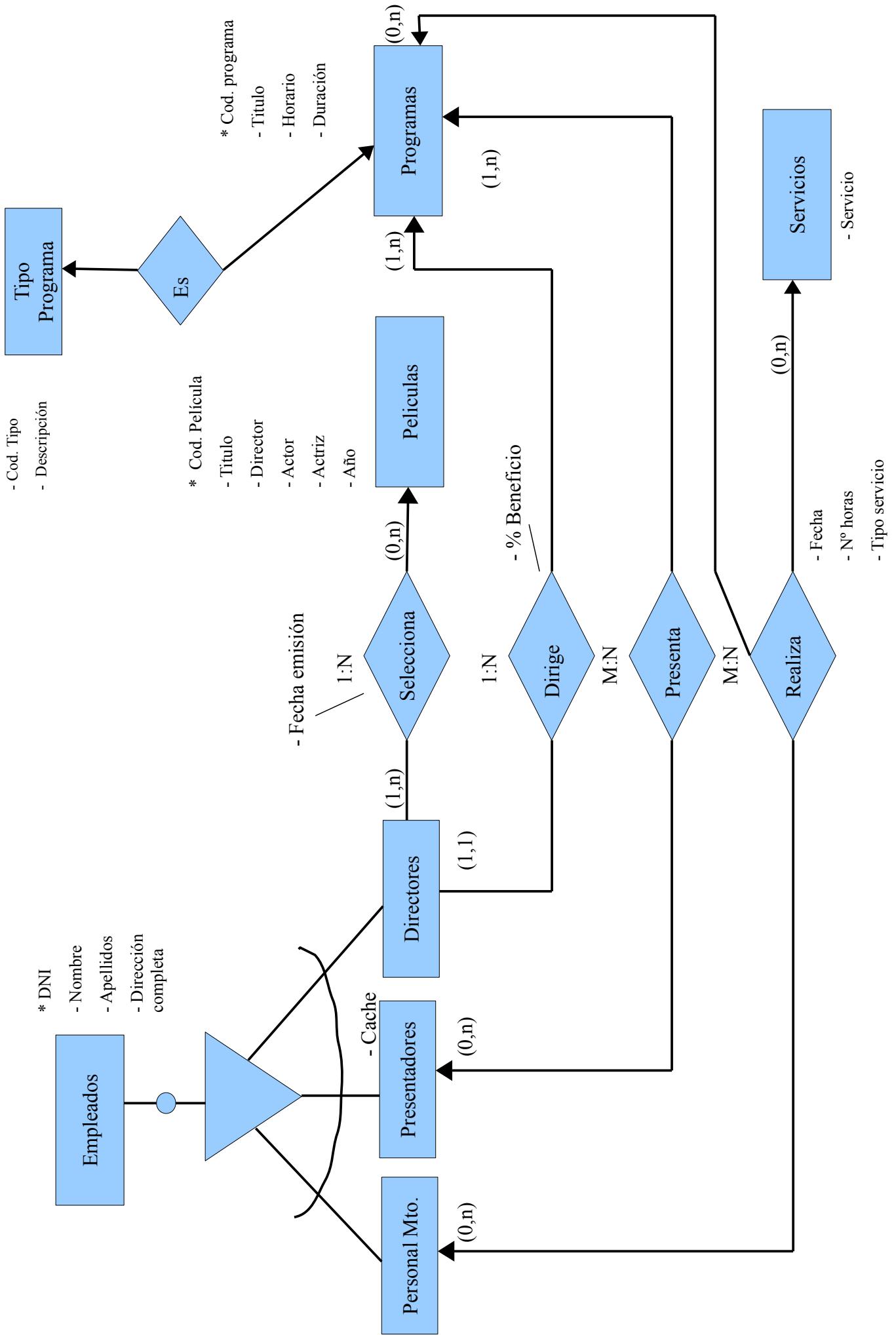
34.- Una empresa de importación desea informatizar la gestión de compra/ventas de productos. Compra diversos productos a empresas extranjeras y los vende a empresas nacionales.

- Por cada empresa necesita NIF, dirección, población, teléfono y fax.
- Cada empresa es cliente o proveedora.
- Para las empresas clientes necesita saber además la provincia, comunidad autónoma y portes (debidos o pagados).
- Para las empresas proveedoras necesita saber el país, CIF internacional y nombre de la persona de contacto.
- Por cada producto tenemos código, descripción y precio de coste.
- Cada proveedor provee un único producto, y cada producto es provisto por un único proveedor. En el momento de contactar con un nuevo proveedor se guardará constancia del mismo y el producto que le debemos comprar.
- Tras cerrar el acuerdo con los clientes guardaremos constancia de qué productos nos demandará en el futuro y cual será el porcentaje de recargo a aplicar en cada producto.
- Se desea guardar constancia de cada compra y venta con la fecha y cantidad de productos comprados/vendidos.
- Pueden existir proveedores y/o clientes con los cuales tengamos firmados acuerdos y aún no hayamos realizado ninguna compra y/o venta.

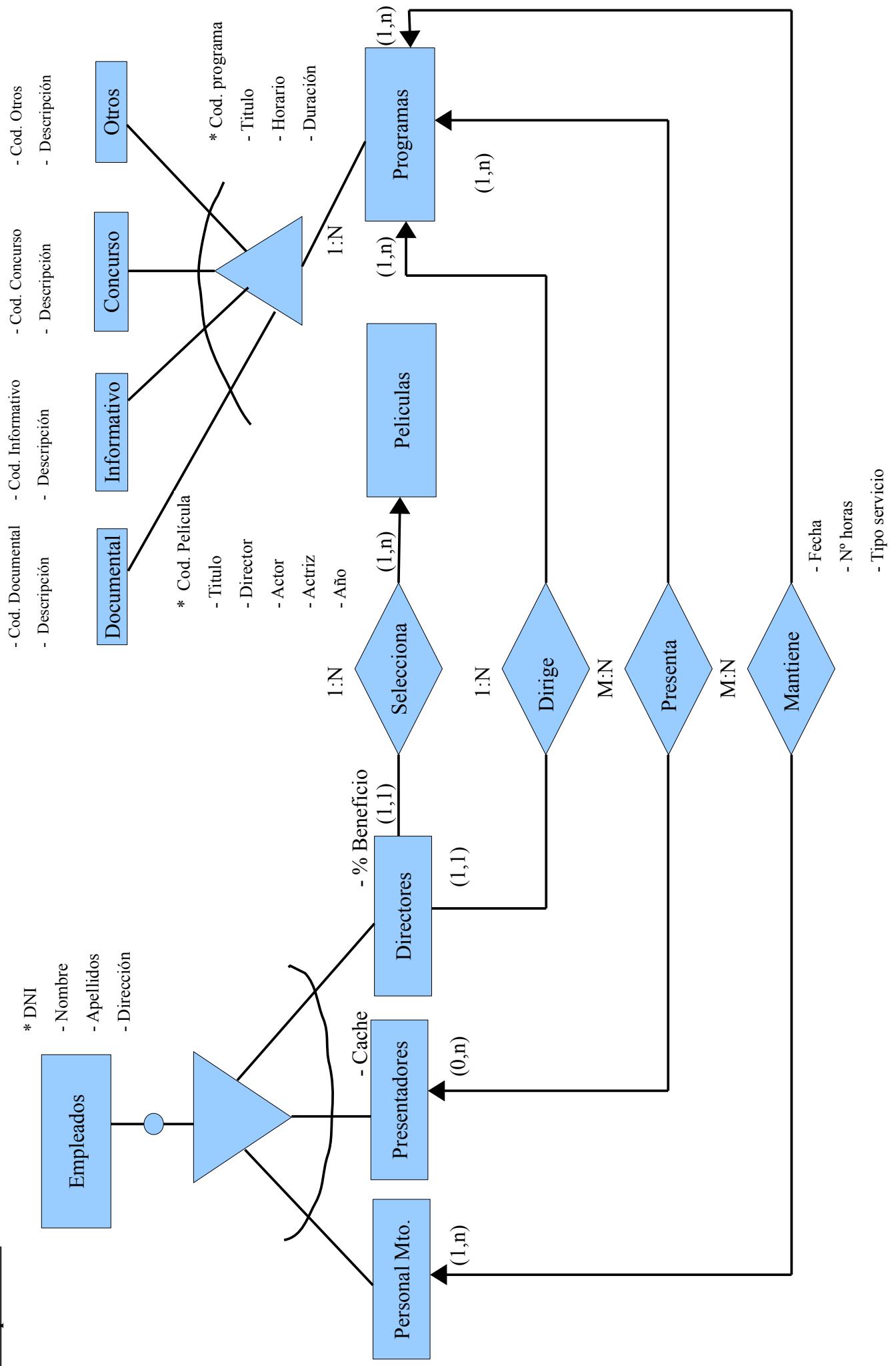


36.- CADENA DE TELEVISIÓN. Una cadena televisiva quiere realizar un sistema informático para registrar información sobre sus empleados y la programación emitida.

1. En esta cadena se emiten tanto programas como películas.
2. La información que se desea registrar de cada película es: título, director, actor, actriz y año.
3. La información que se desea registrar de los programas es: código del programa, título, horario y duración.
4. Existen distintos tipos de programas (documentales, informativos, concurso,...) y de cada uno de ellos conocemos su código y descripción.
5. Las películas que se emiten son seleccionadas por los distintos directores de la cadena. Además los directores dirigen programas, obteniendo por ello un porcentaje de beneficio por programa.
6. Entre los empleados de la empresa, además de directores, existen presentadores y personal de mantenimiento.
7. De cada empleado se desea registrar: DNI, nombre, primer apellido, segundo apellido y dirección completa (calle, nº, piso, CP y población).
8. Los presentadores son los encargados de presentar los distintos programas de la cadena. Existen casos en los que un programa es presentado conjuntamente por más de un presentador, y otros programas que no necesitan presentador.
9. Cada presentador tiene un caché.
10. El personal de mantenimiento realiza servicios sobre los programas, por ejemplo, montaje del decorado, control de sonido, iluminación... Un empleado de mantenimiento puede realizar varios servicios y viceversa.
11. De cada servicio se guarda la fecha en la que se realizó, el nº de horas empleado y el tipo de trabajo realizado.



Hecho por Juan



1.- Dada la siguiente relación:

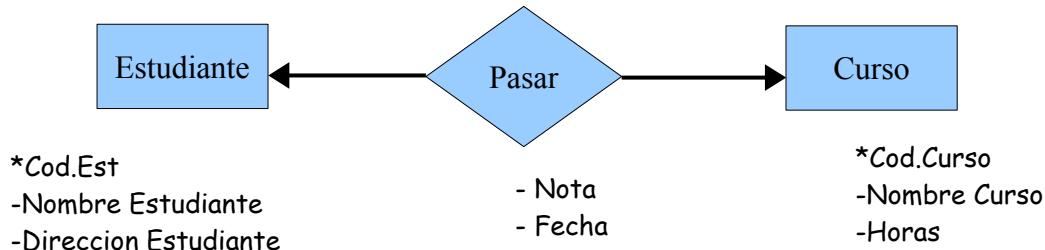
ALUMNO (CodEstudiante, CodCurso, NombreEstudiante, DirecciónEstudiante, NombreCurso, Horas, Fecha, Nota)

Indica las dependencias funcionales.

Estudiante: CodEstudiante, NombreEstudiante, DirecciónEstudiante

Curso: CodCurso, NombreCurso, Horas, Fecha

Nota



2.- Dada la relación:

ALUMNO-BECA (Cod-Estudiante, Nombre Estudiante, Denominación Beca, Cod-Beca, Fecha-Solicitud, Título, Descripción),

y habiéndose definido los campos de esta forma:

- Cod-Estudiante: Identificativo de cada alumno/a
- Nombre y Apellidos del Estudiante
- Denominación Beca
- Nombre de la Beca
- Cod-Beca: Identificativo de cada tipo de beca
- Teléfono del Estudiante
- Fecha Nacimiento
- Fecha-Solicitud: Fecha en la que el alumno solicita cada beca
- Título: Titulación más cualificada del alumno.
- Descripción y características de la Beca

Se debe tener en cuenta que un alumno/a puede solicitar el mismo tipo de beca en diferentes años.

Analiza las dependencias funcionales.



- *Cod. Estudiante
- Nombre Estudiante
- Apellidos Estudiante
- Dirección Estudiante
- Tf. Estudiante
- Fecha Nacimiento
- Titulación

- Fecha solicitud

- *Cod. Beca
- Denominación Beca
- Nombre de la Beca
- Descripción y Características

Alumno (Cod. Estudiante, Nombre Estudiante, Apellidos Estudiante, Dirección Estudiante, Tf. Estudiante, Fecha Nacimiento, Titulación)

Beca (Cod. Beca, Denominación Beca, Nombre de la Beca, Descripción y Características)

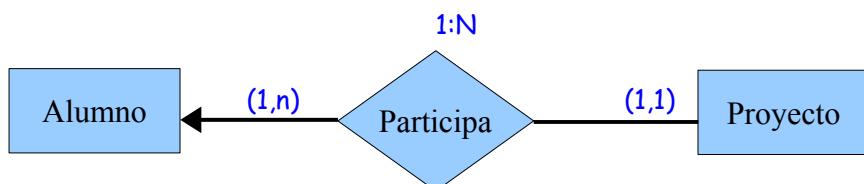
Solicita (Fecha solicitud, Cod. Estudiante (FK), Cod. Beca (FK))

3.- Dada la relación (es continuación del anterior):

ESTUDIANTE (Cod-Estudiante, Cod-Proyecto, Nombre-Proyecto)

y teniendo en cuenta que todo alumno participa en un único proyecto y que en un proyecto pueden participar varios alumnos:

Estudia las dependencias funcionales.



- *Cod. Estudiante
- Nombre Estudiante
- Apellidos Estudiante
- Dirección Estudiante
- Tf. Estudiante
- Fecha Nacimiento
- Titulación

- *Cod. Proyecto
- Nombre Proyecto

Alumno (Cod. Estudiante, Nombre Estudiante, Apellidos Estudiante, Dirección Estudiante, Tf. Estudiante, Fecha Nacimiento, Titulación, Cod. proyecto (FK))

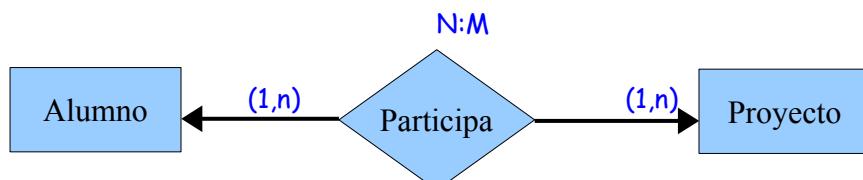
Proyecto (Cod. proyecto, Nombre proyecto)

4.- Teniendo en cuenta esta modificación en la relación del ejercicio anterior:

ESTUDIANTE (Cod-Estudiante, Cod-Proyecto, Nombre-Proyecto, Día-Comienzo)

donde ahora un alumno puede participar en varios proyectos:

Estudia las dependencias funcionales.



*Cod. Estudiante

- Nombre Estudiante
- Apellidos Estudiante
- Dirección Estudiante
- Tf. Estudiante
- Fecha Nacimiento
- Titulación

- Dia comienzo

*Cod. Proyecto

- Nombre Proyecto

Alumno (Cod. Estudiante, Nombre Estudiante, Apellidos Estudiante, Dirección Estudiante, Tf. Estudiante, Fecha Nacimiento, Titulación)

Proyecto (Cod. proyecto, Nombre proyecto)

Participa (Cod. Estudiante (FK), Cod. Proyecto (FK), Dia comienzo)

5.- Transforma el diagrama ER al modelo relacional.



Departamento (Cod. Depto, Denominacion)

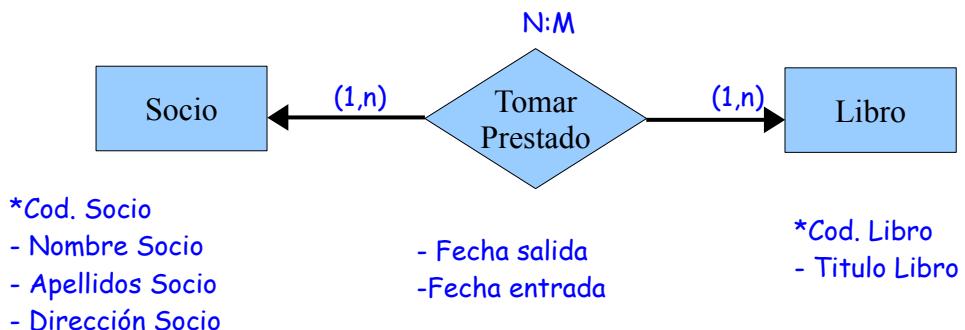
Profesor (DNI, Cod. Depto (FK), Identidad, Especialidad)

Formar part. (DNI (FK) y Cod. Conferencia (FK))

Conferencia (Cod. conferencia, Tema, Fecha, Duracion, Lugar)

6- Representa en un modelo ER la siguiente especificación: un socio toma prestado un libro de la biblioteca durante un periodo de tiempo determinado. Un mismo socio puede tomar prestado el mismo libro en distintos periodos de tiempo.

Una vez preparado el diseño ER, transfórmalo al modelo relacional.

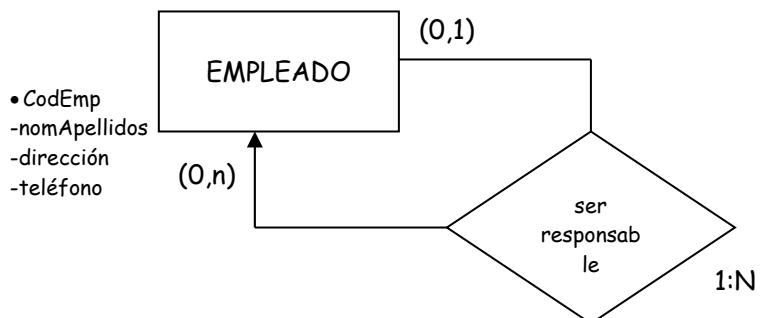


Socio (Cod. Socio, Nombre Socio, Apellidos Socio, Dirección Socio)

Libro (Cod. Libro, Titulo Libro)

Tomar prestado (Cod. socio (FK), Cod. Libro (FK), Fecha salida, Fecha entrada)

7- Dado el diagrama:



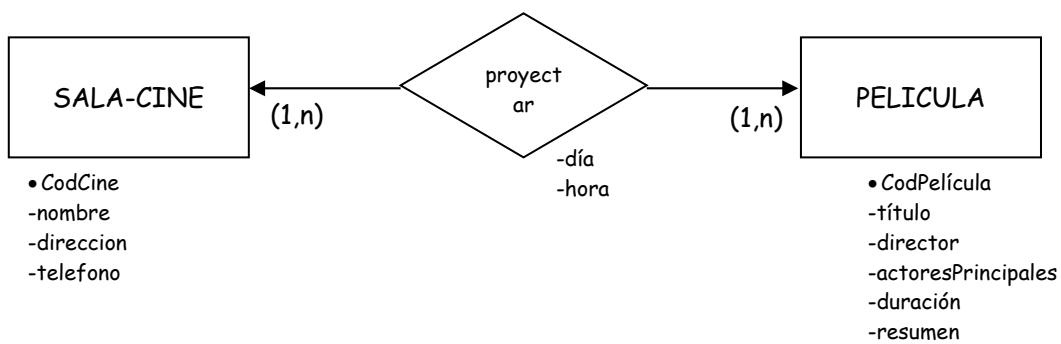
Algunos empleados (jefes, directores, gerentes...) tienen a otros bajo su responsabilidad. Hay también empleados que no son responsables de otros. Todos los empleados tienen a un superior como responsable, excepto el director de la empresa.

¿Cuál de las siguientes opciones sería la más correcta para su correspondiente modelo relacional?

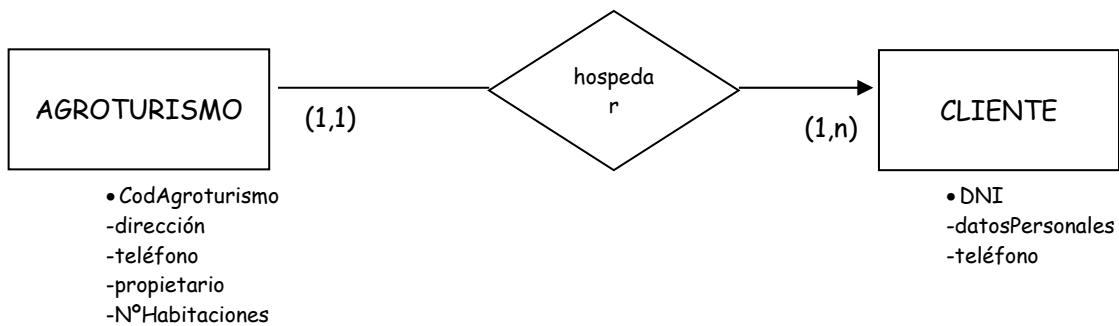
- a) • EMPLEADO (CodEmp, nomApellidos, dirección, teléfono)
- b) • EMPLEADO (CodEmp, nomApellidos, dirección, teléfono, CodEmp)
- c) • EMPLEADO (CodEmp, nomApellidos, dirección, teléfono, CodJefe)
- => d) • EMPLEADO (CodEmp, nomApellidos, dirección, teléfono)**
- RESPONSABLE (CodEmp, CodJefe)

La respuesta c) también podría estar bien suponiendo que el (0,1) fuera (1,1), pero lo mas usual es hacerlo como la d).

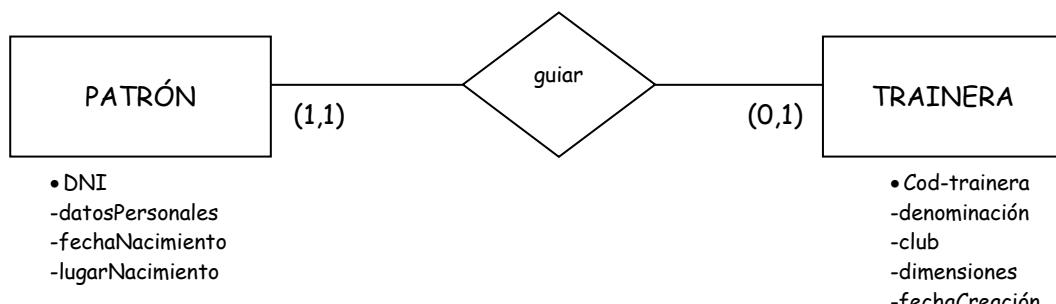
8- Haz la transformación al modelo relacional:

**Sala Cine** (Cod. Cine, Nombre, Dirección, Teléfono)**Película** (Cod. Película, Titulo, director, Actores principales, Duración, Resumen)**Proyectar** (Cod. Cine (FK), Cod. Película (FK), Día, Hora)

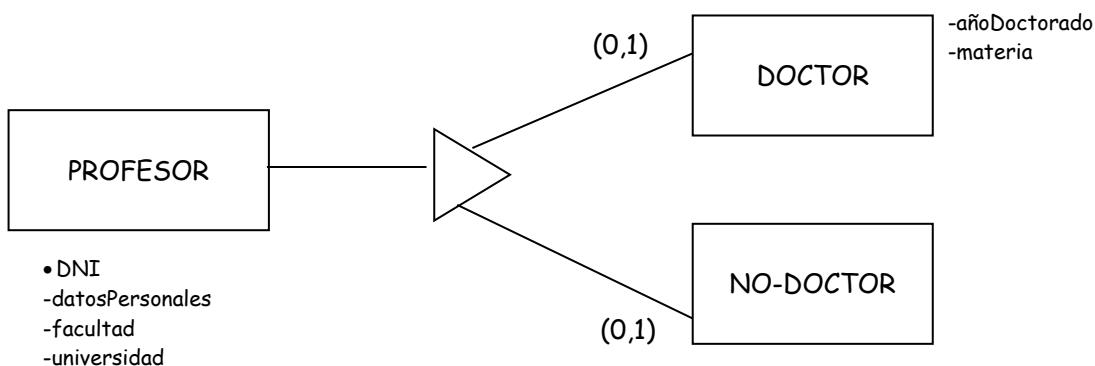
9.- Haz la transformación al modelo relacional:

**Agroturismo** (Cod. Agroturismo, Dirección, Teléfono, Propietario, Nº Habitaciones)**Cliente** (DNI, Cod. Agroturismo (FK), Datos personales, Teléfono)

10.- Transforma el diagrama ER al modelo relacional:

**Patrón** (DNI, Cod., Datos personales, Fecha nacimiento, Lugar Nacimiento)**Trainera** (Cod. Trainera, Denominación, Club, Dimensiones, Fecha creación, DNI(FK))Se propaga del (1,1) a (0,1), porque puede darse el caso de que no haya trainera (0,1)

11.- Analiza tres posibles soluciones para la transformación al modelo relacional de este diagrama ER:



TDOCTOR: (DNI, datospersonales, facultad, universidad, añoDoctorado, materia)

TNoDOCTOR: (DNI, datospersonales, facultad, universidad)

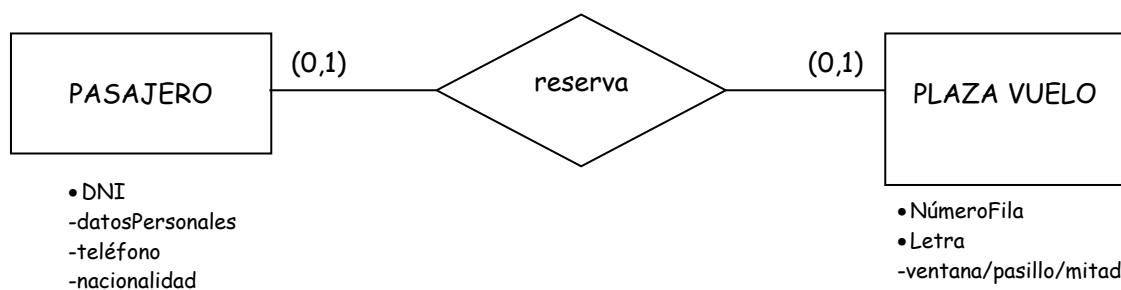
TTPROFESOR: (DNI, datospersonales, facultad, universidad)

TDOCTOR: (DNI(FK), añoDoctorado, materia)

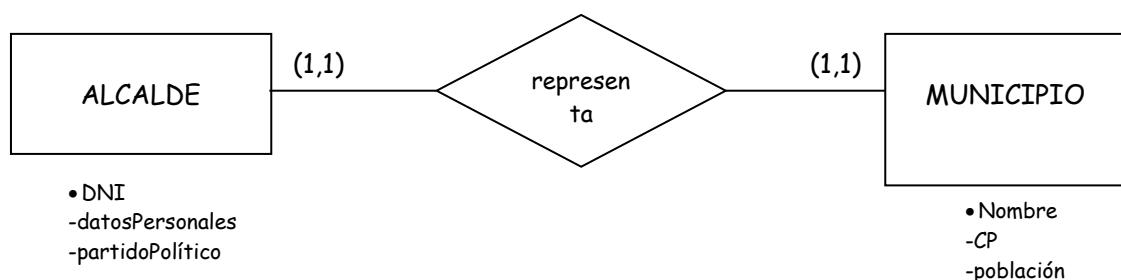
TNoDOCTOR: (DNI (FK))

TTPROFESOR: (DNI, datospersonales, facultad, universidad, añodoctorado, materia)

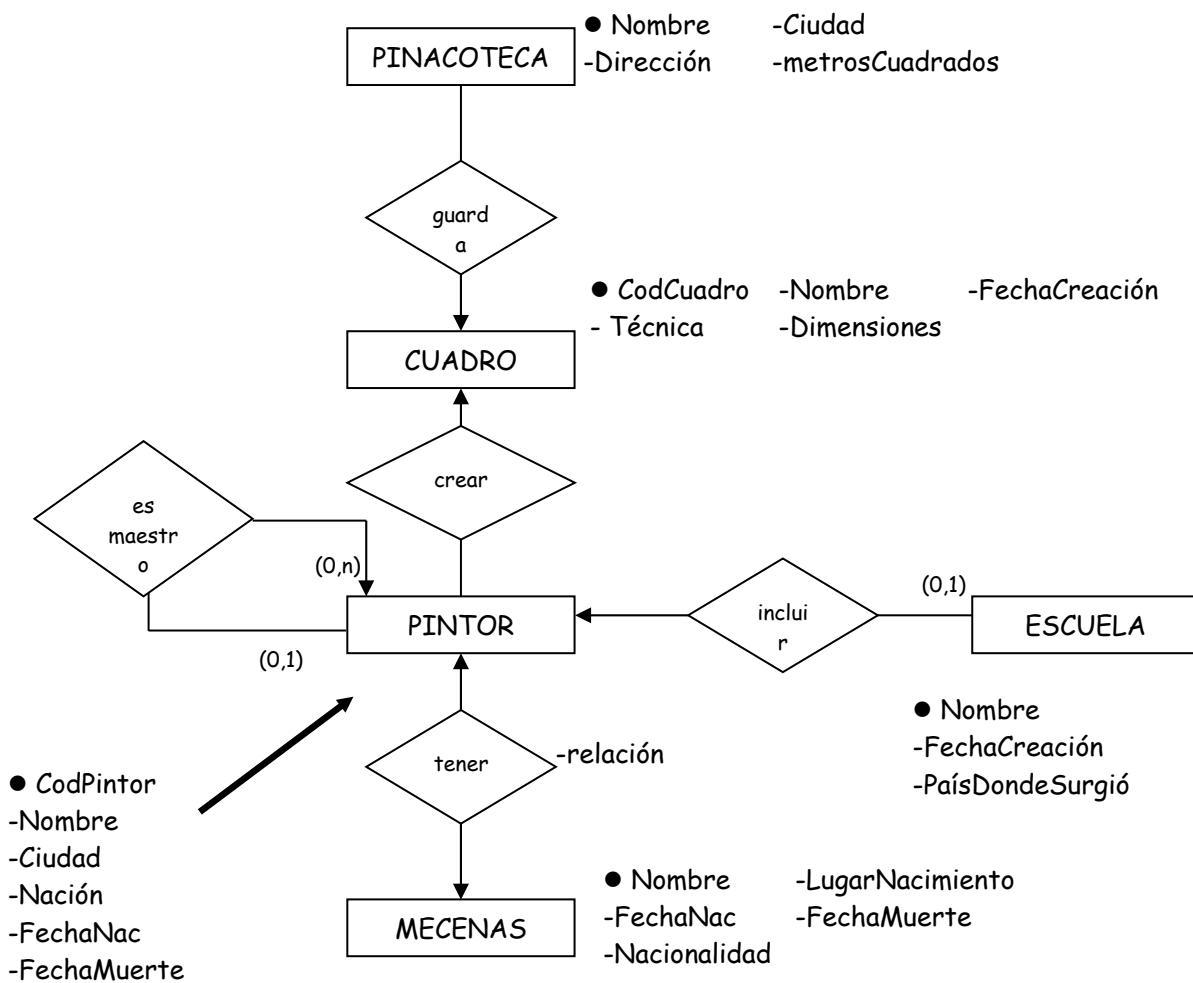
12.- Especifica el modelo relacional que le corresponde al siguiente diagrama ER:



13.- Tranforma el diagrama al modelo relacional:



14.- Transforma este diagrama ER de una Base de Datos a su correspondiente modelo relacional:



TPINACOTECA: (Nombre Pinacoteca, Dirección, Ciudad, MetrosCuadrados)

TCUADRO: (CodCuadro, Técnica, Nombre, Dimensions, Fecha creación, Nombre Pinacoteca(FK), CodPintor(FK))

TPINTOR: (CodPintor, Nombre, Ciudad, Nacion, FechaNac, FechaMuerte, Nombre escuela(FK))

TMECENAS: (Nombre mecenas, FechaNac, Nacionalidad, LugarNacimiento, FechaMuerte)

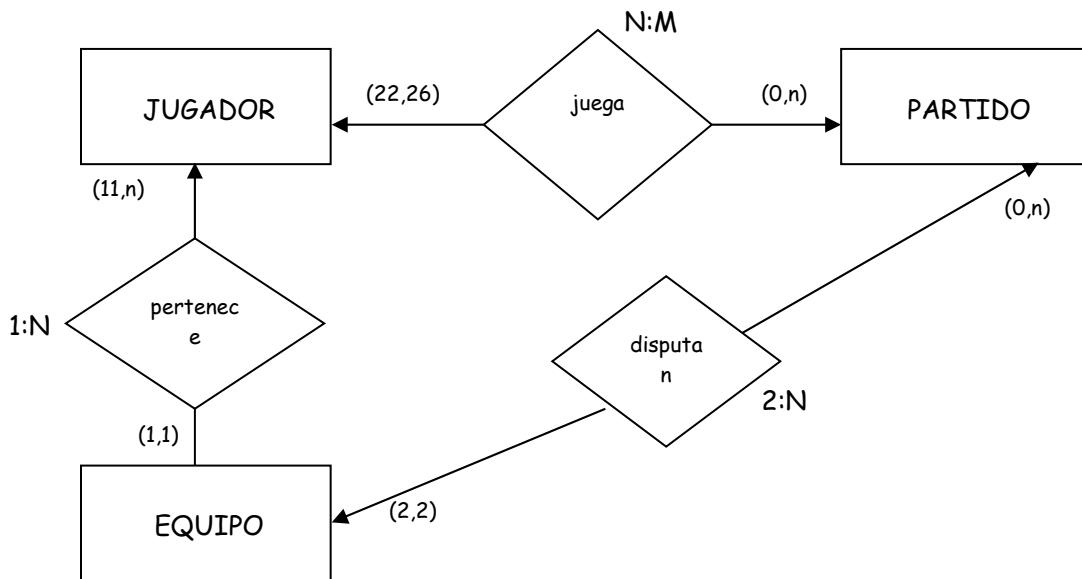
TESCUELA: (Nombre escuela, FechaCreacion, PaisdondeSurgio)

TTener: (CodPintor(FK), Nombre mecenas(FK), relación)

TEsMaestro: (CodPintor(FK), CodMaestro(FK))

15.- Dado el esquema ER de la figura inferior, pásalo al modelo relacionar aplicando las reglas usuales de derivación, teniendo en cuenta que las entidades tienen los siguientes atributos:

- **JUGADOR** (DNI, NombreJugador, FechaNacimiento, Sueldo, Nacionalidad)
- **EQUIPO** (NombreEquipo, Ciudad, Presidente)
- **PARTIDO** (Código, Fecha, Resultado)



TJUGADOR: (DNI, NombreJugador, FechaNacimiento, Sueldo, Nacionalidad, NombreEquipo(FK))

TEQUIPO: (NombreEquipo, Ciudad, Presidente)

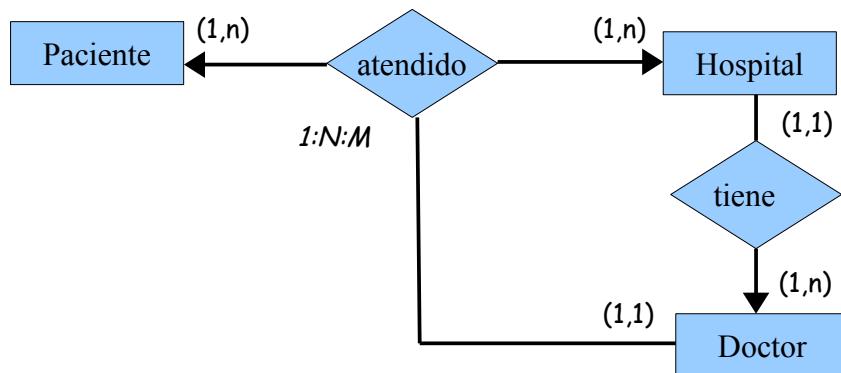
TPARTIDO: (Código, Fecha, Resultado)

TJUEGA: (DNI(FK), Código(FK))

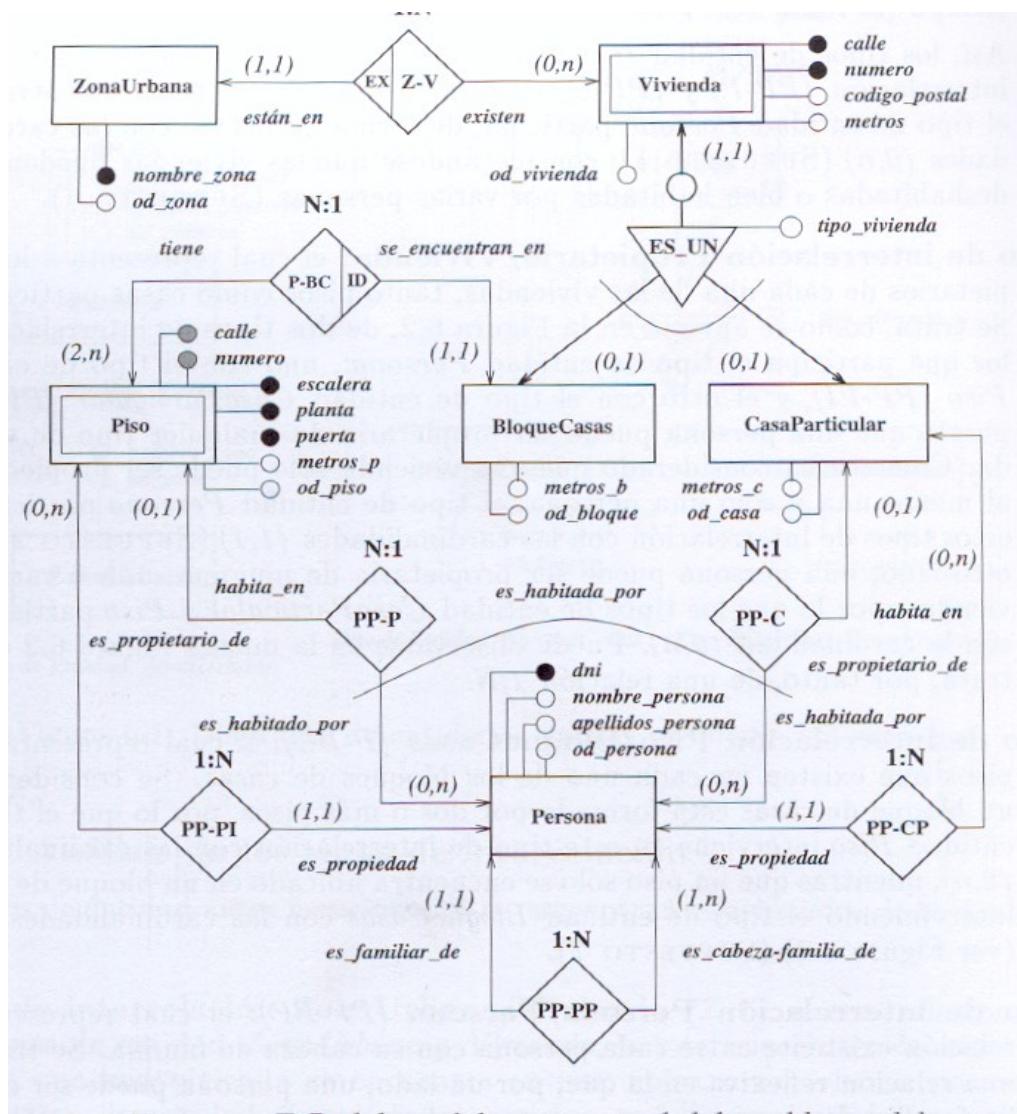
TDISPUTA: (Código(FK), NombreEquipo(FK))

16.- La Seguridad Social desea conocer los pacientes (DNI, Nombre, Dirección) que han sido atendidos en sus hospitales (Cod-Hos) y el doctor que los atiende (Cod-Doc). Suponiendo que un doctor sólo puede atender en un hospital y que, aunque un paciente puede ser atendido en varios hospitales, en cada uno de ellos sólo le atiende un doctor, determinar las dependencias funcionales de la relación:

- **RELACION** (DNI, nombre, dirección, Cod-Hos, Cod-Doc)



17.- Teniendo en cuenta las entidades, relaciones, atributos y cardinalidades de este modelo E/R, haz su transformación al modelo relacional.



TzonaUrbana (Nombre_zona, Od_zona)

TBloqueCasas (Calle, Numero, Código postal, metros, metros_b, Od bloque, Tipo_Vivienda, Nombre de Zona(FK))

TCasaParticular (Calle, Numero, Código postal, metros, metros_c, Tipo_Vivienda, Od_Casa, Nombre de Zona(FK), DNI(FK))

Tpiso (Escalera, Planta, Puerta, Metros_p, Od_piso, Calle(FK), Numero(FK), DNI(FK))

Tpersonas (DNI, DNI Cabeza familia(FK), Nombre_persona, Apellidos_persona, Od_persona, Calle(FK), Numero(FK), escalera(FK), planta(FK), puerta(FK))

La Forma del profesor:

TzonaUrbana (Nombre_zona, Od_zona)

TVivienda (Calle, Numero, Código postal, metros, od_vivienda, Nombre zona(FK))

TBloqueCasas (Calle, Numero, metros_b, Od bloque, Tipo_Vivienda)

TCasaParticular (Calle, Numero, metros_c, Tipo_Vivienda, Od_Casa, DNI(FK))

Tpiso (Calle, Numero, Escalera, Planta, Puerta, Metros_p, Od_piso, , DNI(FK))

Tpersonas (DNI, DNI Cabeza familia(FK), Nombre_persona, Apellidos_persona, Od_persona, Calle(FK), Numero(FK), escalera(FK), planta(FK), puerta(FK))

Desglose Tpersonas para que no queden atributos vacíos:

Tpersonas casa particular (DNI, DNI Cab. familia(FK), Nombre_per. Apellidos_per., Od_persona, Calle(FK), Numero(FK))

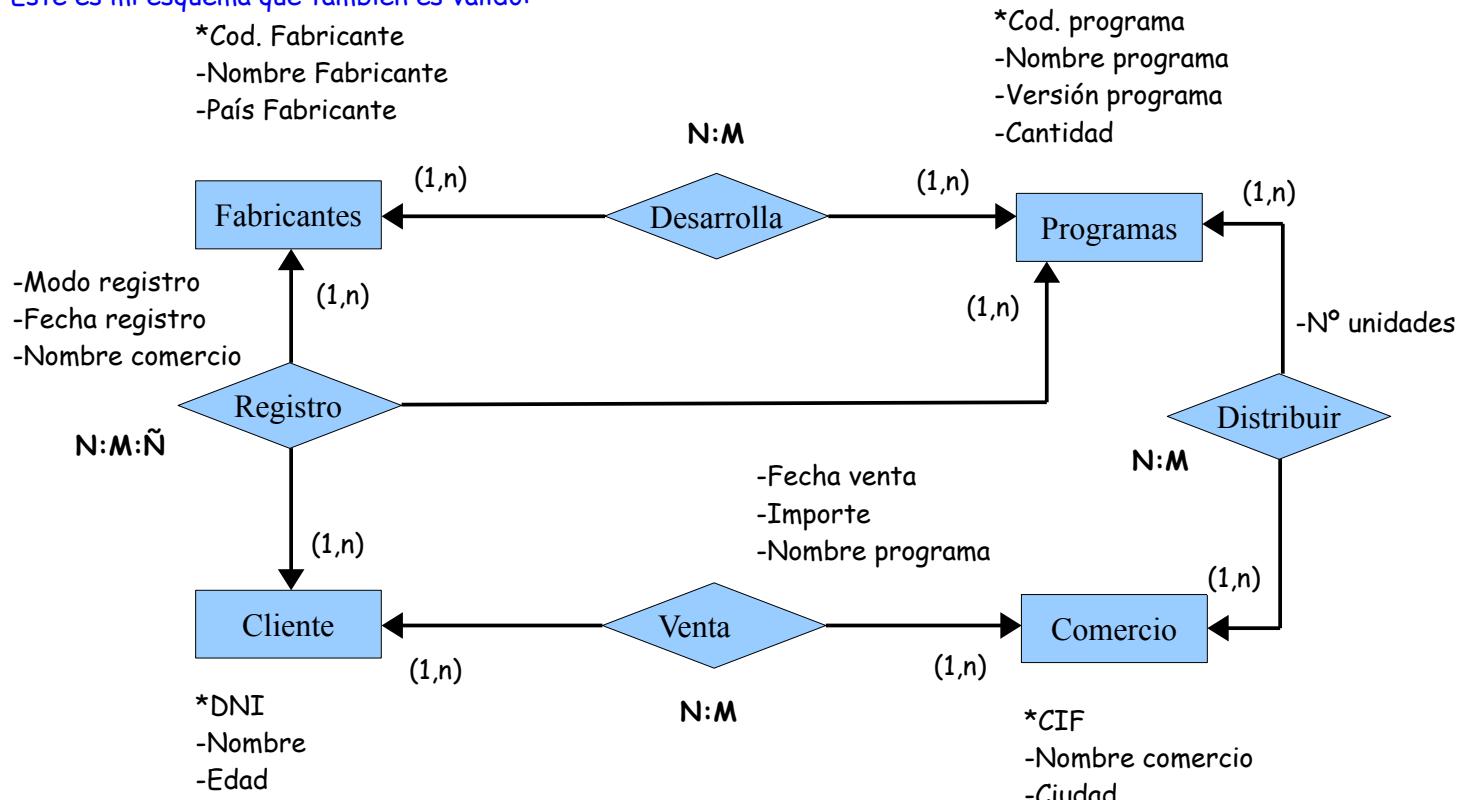
Tpersonas piso (DNI, DNI Cab. familia(FK), Nombre_per., Apellidos_per., Od_per., Calle(FK), Numero(FK), escalera(FK), planta(FK), puerta(FK))

18.- Consideremos una base de datos en la que se relacionen distintos fabricantes de software con los programas que desarrollan, los comercios donde los venden y los clientes que los compran. Las relaciones entre ellos son:

- Cada fabricante se caracteriza por un identificador único, tiene un nombre y pertenece a un país.
- Todos los programas tienen un código que los distinguen de los demás. Además, tienen un nombre y una versión. Por ejemplo, Microsoft Access 2000 tendría código. El código no sirve para distinguir cada CD de Access 2000. Todos los CD de Access 2000 tendrían el mismo código.
- Los comercios tienen un código de identificación fiscal (CIF), único para cada uno, tienen un nombre y se ubican en una ciudad.
- Los clientes se caracterizan por su DNI, nombre y edad.
- Es necesario conocer el número de unidades de cada programa que se distribuye a cada comercio.
- Los clientes, una vez que adquieren el programa, se registran con el fabricante; el registro se puede hacer por diversos medios (Internet, tarjeta postal, teléfono, etc.) y debe quedar constancia del mismo.
- Cada fabricante puede desarrollar muchos programas.
- Los fabricantes se pueden aliar para desarrollar un mismo programa (por tanto, un programa puede desarrollarse por más de un fabricante).
- Un mismo programa se puede distribuir entre muchos comercios y, a su vez, cada comercio puede recibir distribución de muchos programas distintos.
- Un programa que se vende en un comercio puede registrarse a nombre de muchos clientes (por ejemplo, un determinado comercio puede vender muchas unidades de Microsoft Access 2000; cada una se registraría a un cliente distinto).
- Un mismo cliente puede registrar muchos programas distintos adquiridos en el mismo o en distintos comercios.
- Cada cliente que registra un programa, lo hace indicando el comercio donde lo adquirió. Por tanto, por cada cliente y programa sólo se puede indicar un comercio.

Teniendo en cuenta las especificaciones anteriores diseña un modelo conceptual empleando la técnica ER para luego pasarlo al modelo relacional.

Este es mi esquema que también es valido:



TFabricantes (Cod. Fabricante, Nombre Fabricante, País Fabricante)**TPrograma** (Cod. programa, Nombre programa, Versión programa, Cantidad)**TCliente** (DNI, Nombre, Edad)**TComercio** (CIF, Nombre comercio, Ciudad)**TRegistro** (Modo registro, Fecha registro, Nombre comercio, DNI(FK), Cod. Fabricante(FK), Cod. programa(FK))**TVenta** (DNI(FK), CIF(FK), Nombre programa(FK), Fecha venta, Importe)**TDistribuir** (Nº unidades, CIF(FK), Cod. programa(FK))

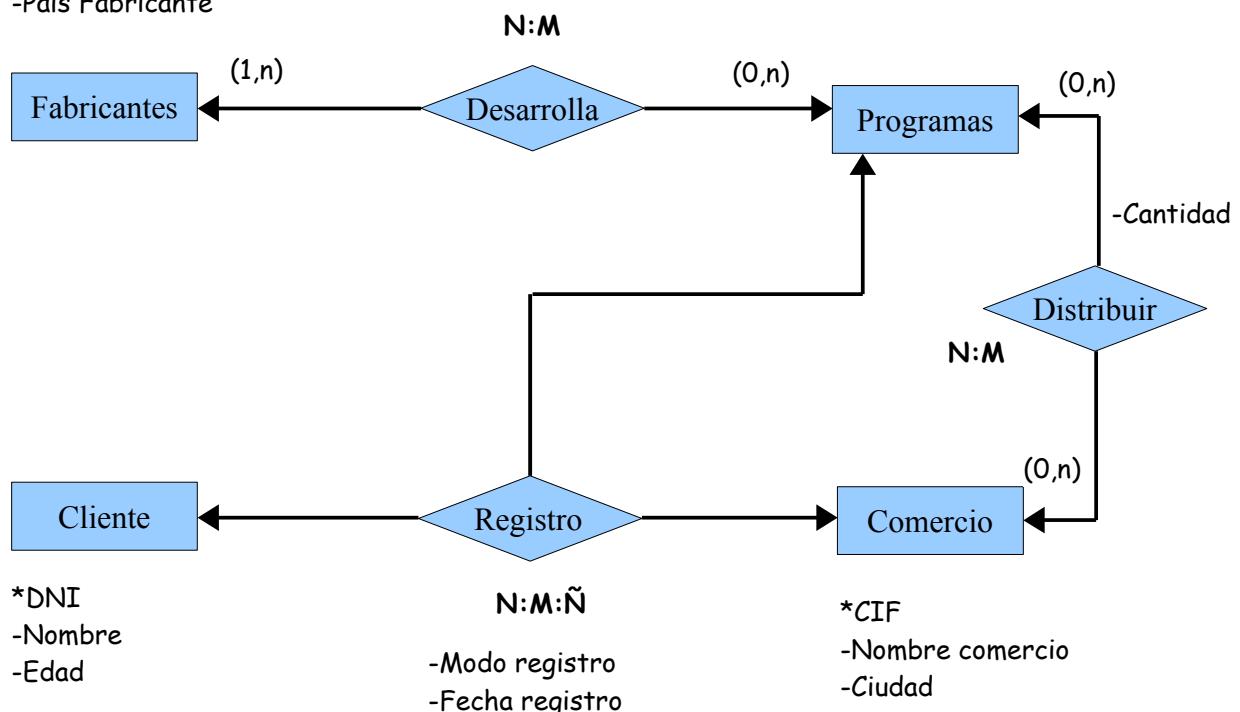
La Tabla Venta relaciona cliente con Comercio y programa así como fecha venta e importe

La Tabla Registro relaciona Fabricante, con cod. programa, con cliente y con el comercio

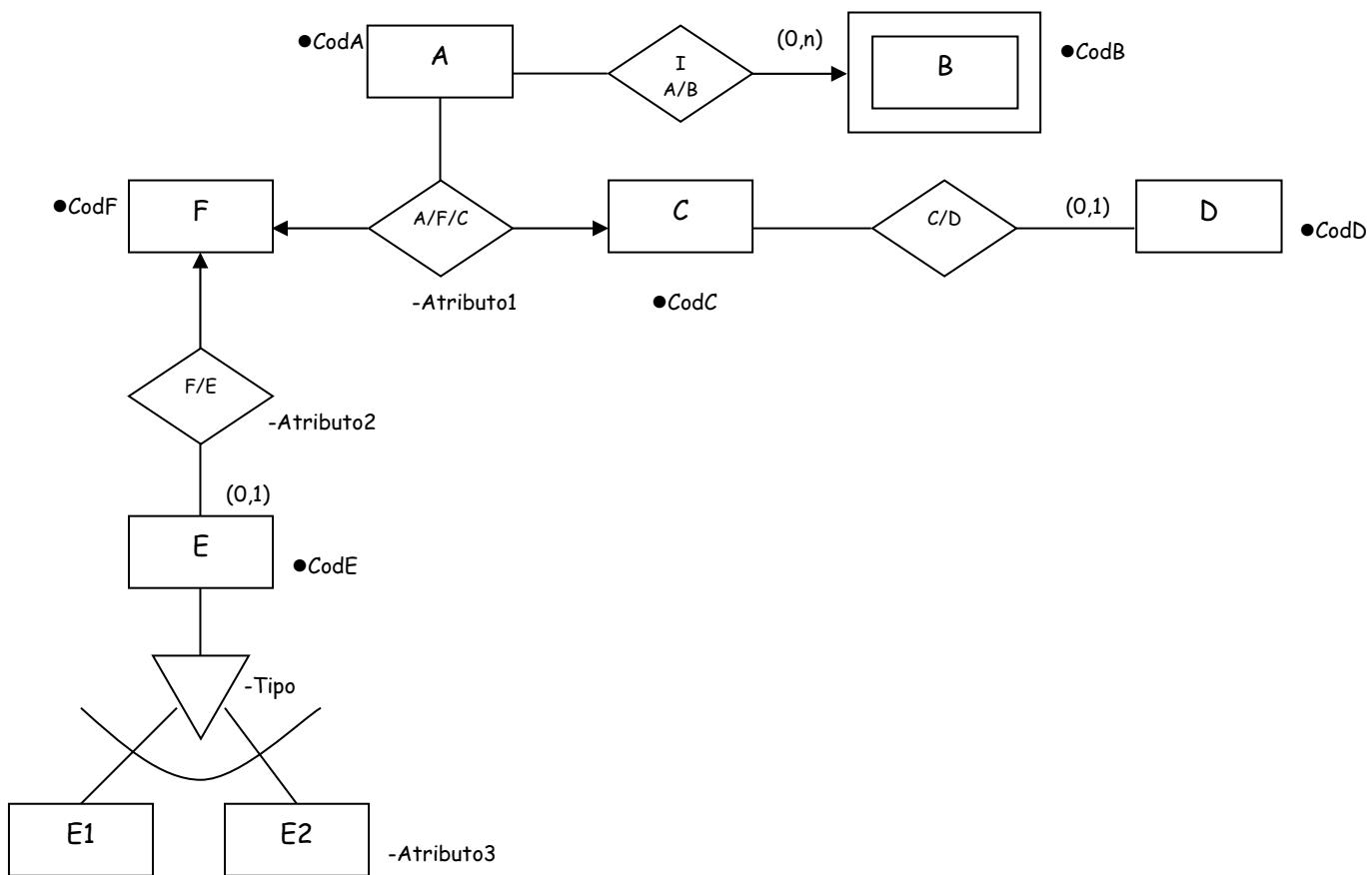
Esquema del profesor:

- *Id. Fabricante
- Nombre Fabricante
- País Fabricante

- *Cod. programa
- Nombre programa
- Versión programa
- Cantidad

**TFabricantes** (Id. Fabricante, Nombre Fabricante, País Fabricante)**TPrograma** (Cod. programa, Nombre programa, Versión programa, Cantidad)**TCliente** (DNI, Nombre, Edad)**TComercio** (CIF, Nombre comercio, Ciudad)**TRegistro** (DNI(FK), CIF(FK), Cod. programa(FK), Modo registro, Fecha registro)**TDistribuir** (Nº unidades, CIF(FK), Cod. programa(FK))

19.- Transforma el siguiente esquema ER extendido a un esquema relacional. Se deberá obtener la transformación que proporcione accesos a la base de datos más eficiente, teniendo en cuenta que los subtipos E1 y E2 se diferencian en muy pocos atributos y que siempre que se accede a la entidad E se querrá toda la información, tanto la del supertipo como la de los subtipos.



TA(CodA)

TB (CodB, CodA(FK))

TC (CodC)

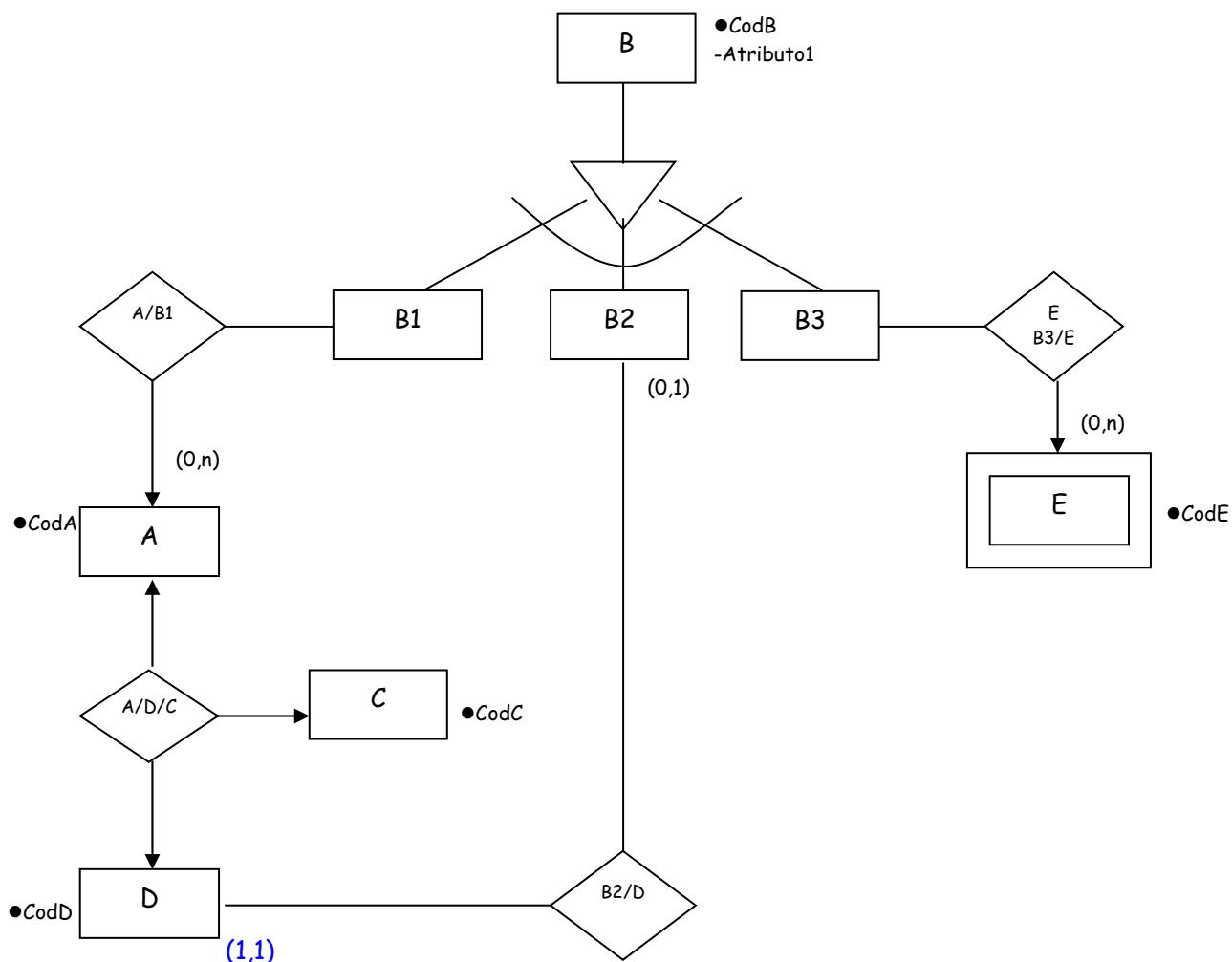
TD (CodD, CodC(FK))

TE (CodE, tipo, Atributo3)

TF (CodF, CodE(FK), atributo2)

TA/F/C (CodA, CodC, CodF, atributo1)

20.- Transforma el siguiente esquema ER a un esquema relacional. Se sabe que los subtipos de la entidad B tienen muchos atributos distintos y que en los accesos a esa entidad se querrán los atributos comunes.



TA(CodA, CodB(FK))

TB1 (CodB, Atributo1, Atributos B1)

TB2 (CodB, Atributo1, Atributos B2, CodD(FK))

TB3 (CodB, Atributo1, Atributos B3)

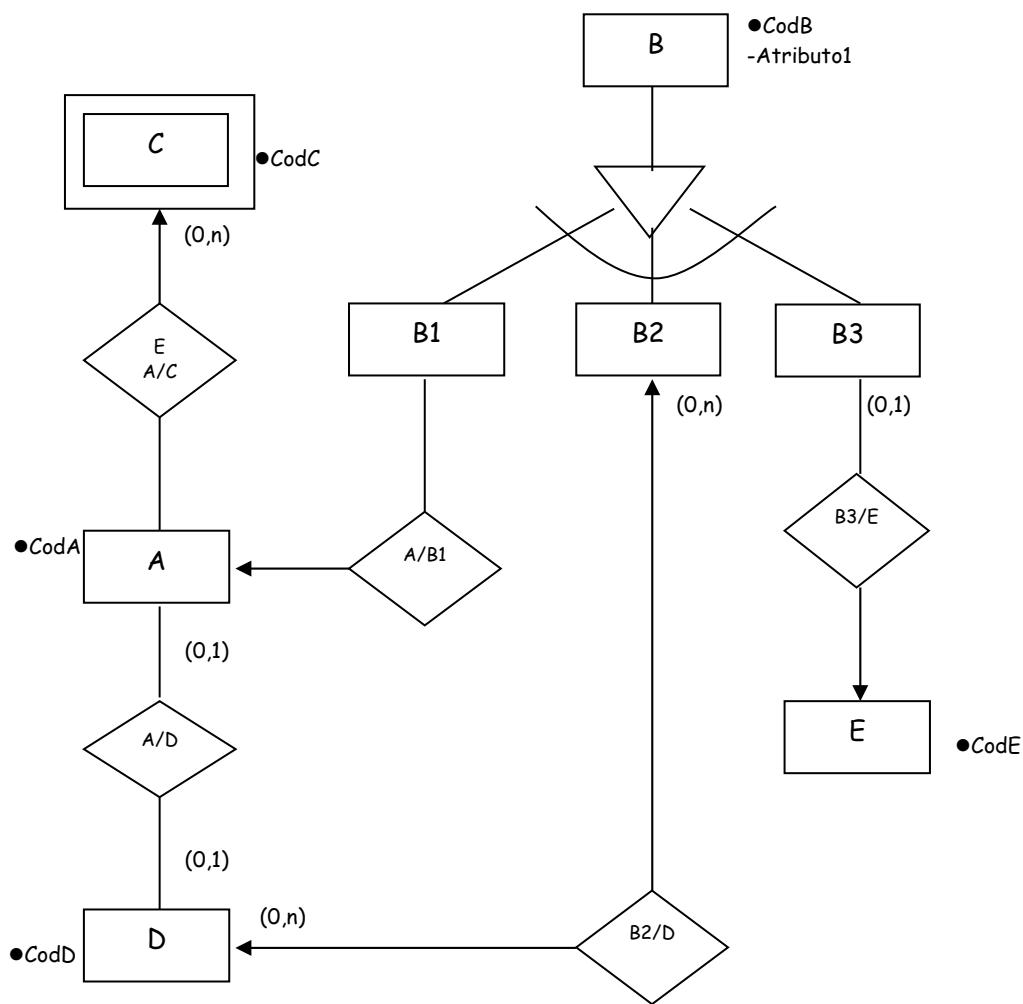
TC (CodC)

TD (CodD)

TE (CodE, CodB(FK))

TA/D/C (CodA(FK), CodC(FK), CodD(FK))

21.- Transforma el siguiente esquema ER a un esquema relacional. Se sabe que los subtipos de la entidad B tienen muchos atributos distintos.



TA (CodA, CodB(FK))

TB1 (CodB, Atributo1, AtributosB1)

TB2 (CodB, Atributo1, AtributosB2)

TB3 (CodB, Atributo1, AtributosB3)

TC (CodC, codA(FK))

TD (CodD)

TE (CodE, CodB(FK))

TB2/D (CodB(FK), CodD(FK))

TB3/E (CodE(FK), CodB(FK))

TA/D (CodA(FK), CodD(FK))

En el caso B3 con E se puede también propagar en el sentido de la flecha (de B3 a E)

En el caso de A y D se puede prescindir de la Tabla A/D y propagar de A a D y de D a A

22. - Transforma los siguientes esquemas relacionales en otros en FNBC. Indica las tablas resultantes:

a) R(a, b, c, d) sabiendo que:

$$\begin{array}{l} a \rightarrow b \\ a \rightarrow e \\ b \rightarrow e \\ b \rightarrow d \\ d \rightarrow e \end{array}$$

b) R(a, b, c, d, f) sabiendo que:

$$\begin{array}{l} a, b \rightarrow d \\ a \rightarrow e \\ c, d \rightarrow f \\ f \rightarrow d \end{array}$$

c) R(a, b, c, d, f) sabiendo que:

$$\begin{array}{l} a, b \rightarrow e \\ a, b \rightarrow d \\ d \rightarrow b \\ c \rightarrow a \\ c \rightarrow f \end{array}$$

23. - Transforma los siguientes esquemas relacionales en otros en FNBC. Indica las tablas resultantes:

a) R(a, b, c, d) sabiendo que:

$$\begin{array}{l} a \rightarrow b \\ a \rightarrow e \\ c \rightarrow d \\ d \rightarrow a \end{array}$$

b) R(a, b, c, d) sabiendo que:

$$\begin{array}{l} a, d \rightarrow b \\ b \rightarrow e \\ c \rightarrow d \end{array}$$

c) R(a, b, c, d) sabiendo que:

$$\begin{array}{l} a, b \rightarrow e \\ a \rightarrow d \\ d \rightarrow b \\ c \rightarrow a \end{array}$$

d) R(a, b, c, d, e, f) sabiendo que:

- a, b, c \rightarrow f
- b, c \rightarrow d
- a \rightarrow e
- f \rightarrow b
- f \rightarrow e

24. - Transforma los siguientes esquemas relacionales en otros en FNBC. Indica las tablas resultantes:

a) R(a, b, c, d, e, f, h) sabiendo que:

- a \rightarrow e
- c \rightarrow d
- d \rightarrow a
- a, b \rightarrow h
- b \rightarrow e
- e \rightarrow f
- f \rightarrow b

b) R(a, b, c, d) sabiendo que:

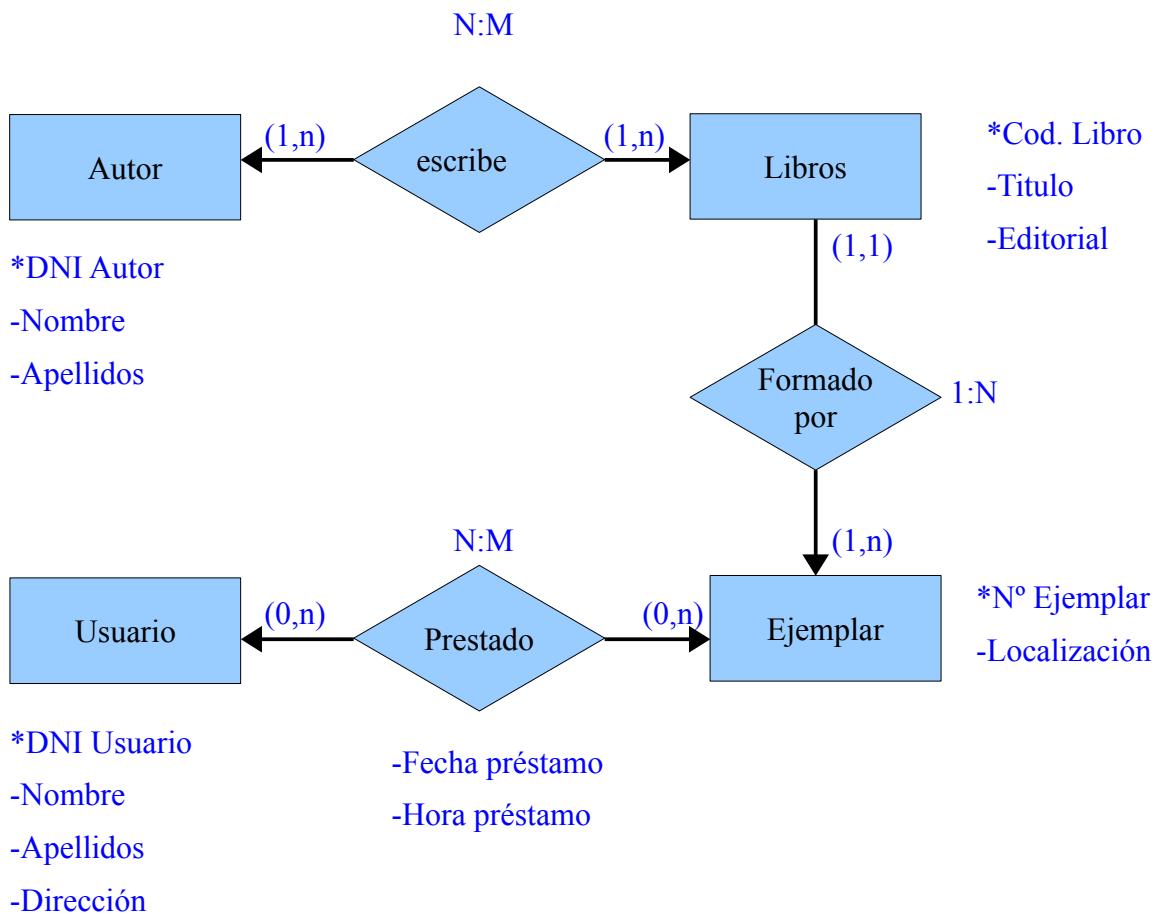
- a, b \rightarrow e
- a, b \rightarrow d
- c \rightarrow a
- d \rightarrow b

Ejercicio 1 pag. 67, del Capítulo 2 del Libro.

Realiza el diagrama E-R que cumpla las especificaciones y pásalo al modelo de datos relacional.

Se desea mecanizar la biblioteca de un centro educativo. En la biblioteca existen fichas de autores y libros. Un autor puede escribir varios libros, y un libro puede ser escrito por varios autores. Un libro está formado por ejemplares que son los que se prestan a los usuarios.

Así un libro tiene muchos ejemplares y un ejemplar pertenece sólo a un libro. De los ejemplares nos interesa saber la localización dentro de la biblioteca. Los ejemplares son prestados a los usuarios, un usuario puede tomar prestados varios ejemplares y un ejemplar puede ser prestado a varios usuarios. Del préstamo nos interesa saber la fecha de préstamo y la de devolución.



Autor (DNI, Nombre, Apellidos)

Escribe (DNI Autor (FK), Código Libro (FK))

Libros (Cod. Libro, Título, Editorial)

Ejemplares (Nº Ejemplar, Localización, Cod. Libro (FK))

Prestado (Nº Ejemplar (FK), DNI Usuario (FK), Fecha préstamo, Hora préstamo)

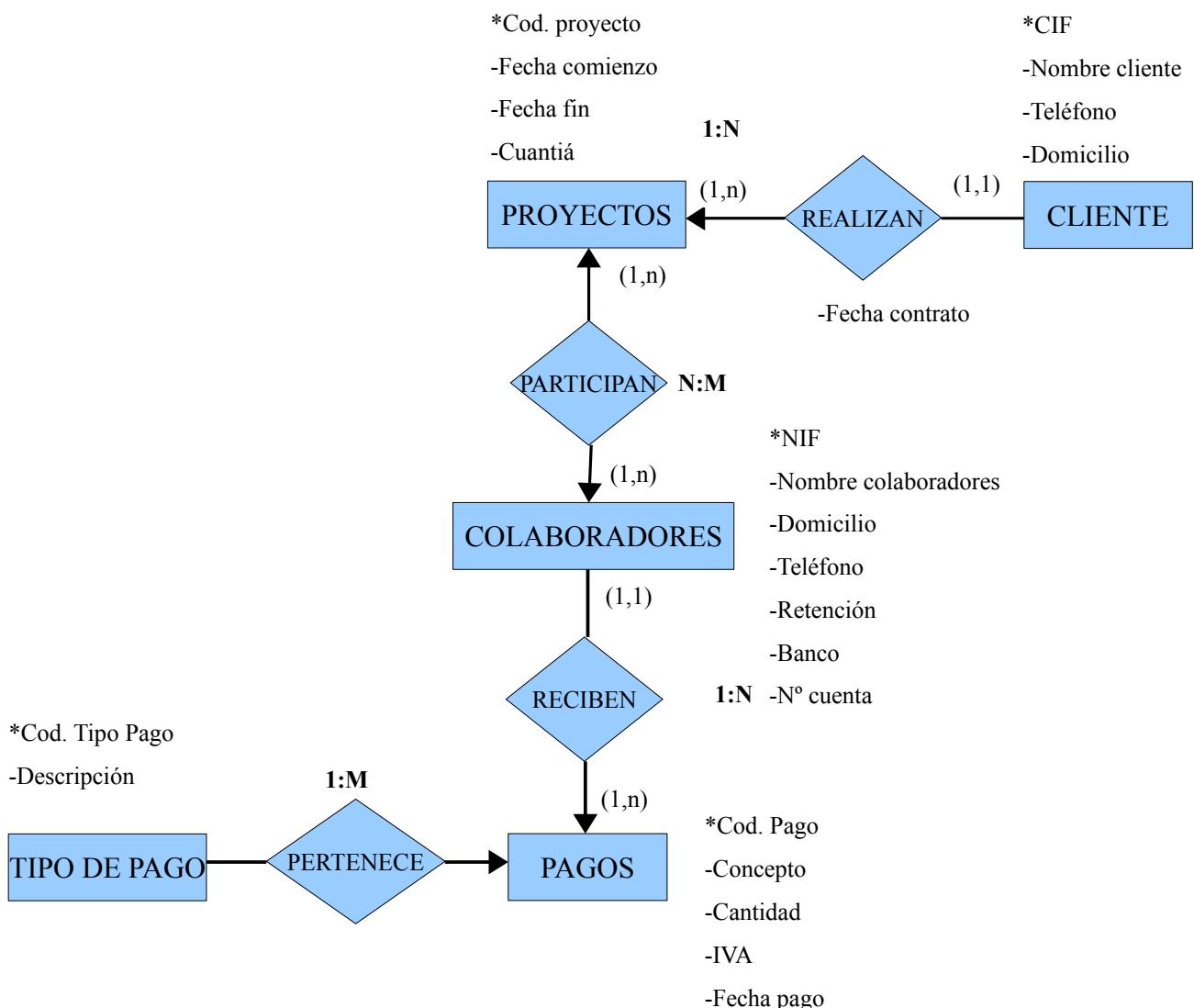
Usuario (DNI Usuario, Nombre, Apellidos, dirección)

Ejercicio 1 pag. 93, del Capítulo 2 del Libro.

Se desea informatizar la gestión de los proyectos del departamento de química de una universidad siguiendo las siguientes especificaciones:

- Al departamento llegan una serie de clientes que quieren realizar proyectos. Generalmente los clientes son empresas que realizan contratos con el grupo de investigación del departamento. Un cliente puede realizar varios proyectos.
- Un proyecto es de un cliente. Cada proyecto tiene asignada una cuantía de dinero que se utilizará para pagar los gastos del proyecto. De esta cuantía se saca el dinero para realizar los pagos a los colaboradores. También nos interesa saber de los proyectos el nombre, la fecha de comienzo, la de fin, entre otros.
- De cada proyecto se realizan muchos pagos para pagar a los colaboradores.
- De los pagos nos interesa saber el concepto, la cantidad, el IVA aplicado y la fecha del pago.
- Existen varios tipos de pagos (por ejemplo nomina, Representación, Material, etc.). Un pago es de un tipo de pago y a un tipo de pago pueden pertenecer muchos pagos.
- Existe una serie de colaboradores que son personas o entidades que van a recibir el dinero de los pagos en concepto de una tarea realizada o la compra de material. Un pago solo puede ser para un colaborador. Este a su vez puede recibir muchos pagos.
- De los colaboradores nos interesa saber: Nombre, NIF, Domicilio, Teléfono, Retención, Banco, Nº Cuenta.

Realiza el diagrama E-R que cumpla las especificaciones y pasalo al modelo de datos relacional.



TCliente (CIF, Nombre cliente, Teléfono, Domicilio)

TProyectos (Cod. proyecto, Fecha comienzo, Fecha fin, Cuantía, CIF(FK))

TParticipar (Cod. proyecto(FK), NIF(FK))

TColaboradores (NIF, Nombre colaborador, Domicilio, Teléfono, Retención, Banco, Nº Cuenta)

TPagos (Cod. pago, Concepto, Cantidad, IVA aplicado, Fecha del pago, Cod. tipo pago(FK), NIF(FK))

TTipoPagos (Cod. tipo pago, Descripción)

Ejercicio 2 pag. 93, del Capítulo 2 del Libro.

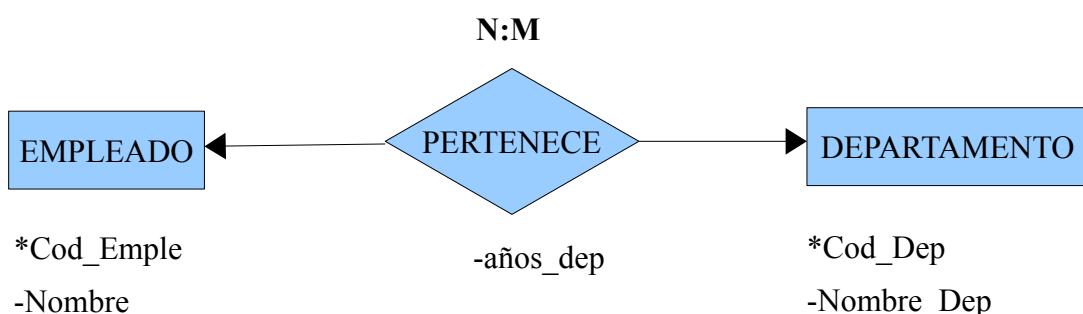
Dada la siguiente Tabla, Transformarla a 3FN

COD_EMPLE	NOMBRE	COD_DEP	NOMBRE_DEP	AÑOS_DEP
1	Juan	6	Contabilidad	6
2	Pedro	3	Sistemas	3
2	Pedro	6	Contabilidad	5
3	Sonia	2	I+D	1
4	Veronica	3	Sistemas	10
4	Veronica	6	Contabilidad	2

Paso a 2FN

COD_EMPLE	NOMBRE	COD_DEP	NOMBRE_DEP	COD_EMPLE	COD_DEP	AÑOS_DEP
1	Juan	6	Contabilidad	1	6	6
2	Pedro	3	Sistemas	2	3	3
3	Sonia	2	I+D	2	6	5
4	Veronica	3	2	1
.....	4	3	10
.....	4	6	2

Paso a 3FN



Ejercicio Normalizacion 1

Tenemos una empresa publica donde los puestos de trabajo, están regulados por el estado, de modo que las condiciones salariales están determinadas por el puesto de trabajo, se ha creado el siguiente esquema relacional: (con el numero de la seguridad social NSS como clave primaria)

Empleados (NSS, Nombre, puesto, salario, emails)

NSS	Nombre	Puesto	Salario	Emails
111	Pepe	Jefe Area	3000	josep@ecn.es jefez@gmail.com
222	Josu	Admtivo	1500	jsanchez@ecn.es
333	Miren	Admtiva	1500	mlopez@ecn.es miren@gmail.com
....				
.....				

Paso a 1FN

NSS	Nombre	Puesto	Salario	Emails
111	Pepe	Jefe Area	3000	josep@ecn.es
111	Pepe	Jefe Area	3000	jefez@gmail.com
222	Josu	Admtivo	1500	jsanchez@ecn.es
333	Miren	Admtiva	1500	mlopez@ecn.es
333	Miren	Admtiva	1500	miren@gmail.com
....				
.....				

Paso a 2FN

NSS	Nombre	Puesto	Salario
111	Pepe	Jefe Area	3000
222	Josu	Admtivo	1500
333	Miren	Admtiva	1500
....
....
....

Emails	NSS (FK)
josep@ecn.es	111
jefez@gmail.com	111
jsanchez@ecn.es	222
mlopez@ecn.es	333
miren@gmail.com	333
.....

Paso a 3FN

NSS	Nombre	Puesto(FK)
111	Pepe	Jefe Area
222	Josu	Admtivo
333	Miren	Admtiva
....
....
....
....

Puesto	Salario
Jefe Area	3000
Admtivo	1500
Admtiva	1500
.....
.....
.....
.....

Emails	NSS (FK)
josep@ecn.es	111
jefez@gmail.com	111
jsanchez@ecn.es	222
mlopez@ecn.es	333
miren@gmail.com	333
.....
.....

TEMPLEADOS (NSS, Nombre, Puesto, Salario, Email) No esta normalizado por haber mas de 1 email por persona.

1FN

TEMPLEADOS (NSS, Nombre, Puesto, Salario, Email)

2FN

TEMPLEADOS (NSS, Nombre, Puesto, Salario)

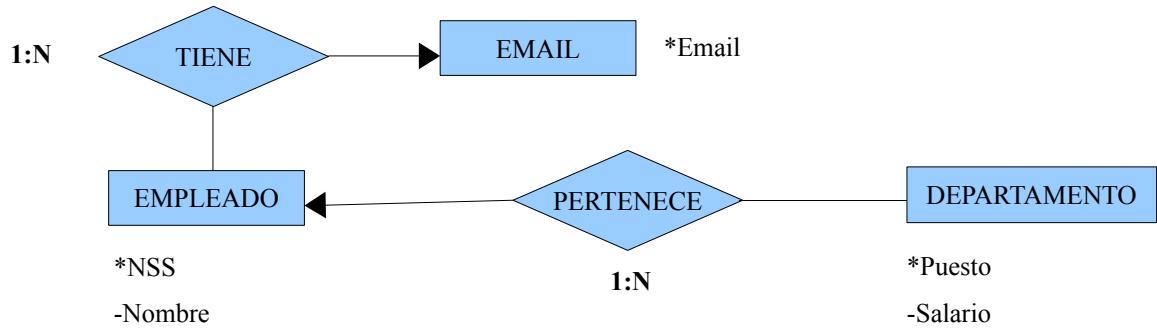
TEMAILS (NSS(FK), Email)

3FN

TEMPLEADOS (NSS, Nombre, Puesto(FK))

TSALARIO (Puesto, Salario)

TEMAILS (NSS(FK), Email)



Ejercicio 3 pag. 93, del Capítulo 2 del Libro.

A partir de las siguientes tablas:

AGENDA (Nombre, Edad, CódigoProvincia, Telef)

PROVINCIAS (Código, NombreProv)

- Escribe la columna o conjunto de columnas que pueden ser claves primarias y ajenas.

TABLA AGENDA

PK	FK	Nombre	Edad	CódigoProvincia	Telef
		Jon alvarez	33	48	944355678

TABLA PROVINCIAS

PK	Código	NombreProv.
	48	VIZCAYA

- Escribe un enunciado para las siguientes expresiones:

a) $\sigma_{EDAD>37} (AGENDA)$

Sacar de la Tabla AGENDA las Tuplas (Filas) que cumplan que la edad es mayor que 37

b) $\Pi_{Nombre, Edad} (AGENDA)$

De la Tabla AGENDA sacar las columnas de Nombre y Edad

c) $(AGENDA * PROVINCIAS) \text{ CódigoProvincia=Código}$

Realizar la combinación o Join de las Tablas AGENDA y PROVINCIAS que cumplan la condición CódigoProvincia=Código

d) $\Pi_{Nombre, CódigoProvincia, Telef, NombreProv} ((AGENDA * PROVINCIAS) \text{ CódigoProvincia=Código})$

Combinación de la Tabla AGENDA y PROVINCIA con la condición CodProvincia=Código y proyectar las columnas Nombre, CódigoProvincia, Telef y NombreProv

e) $\Pi_{Nombre, CódigoProvincia, Telef, NombreProv} (\sigma_{EDAD>37} ((AGENDA * PROVINCIAS) \text{ CódigoProvincia=Código}))$

Combinación de la Tabla AGENDA y PROVINCIA con la condición CodProvincia=Código, de ahí sacamos los que cumplen la condición de Edad>37 y proyectar las columnas Nombre, CódigoProvincia, Telef y NombreProv.

Ejercicio 4 y 5 pag. 86, del Capítulo 2 del Libro

Nº EMPLE	APELLIDO	SALARIO	COMISION	Nº.DEPART	JEFE
7369	SANCHEZ	1040		20	7902
7499	ARROYO	2080	390	30	7698
7521	SALA	1625	650	30	7698
7566	JIMENEZ	3867		20	7839
7654	MARTIN	1625	1820	30	7698
7698	NEGRO	3705		30	7839
7782	CEREZO	3185		10	7839
7788	GIL	3900		20	7566
7839	REY	6500		10	
7876	ALONSO	1430		20	7788

Fig 2.19

Ejercicio 4

A partir de la tabla EMPLEADOS mostrada en la figura 2.19 selecciona aquellas cuyo jefe se corresponda con el numero 7839 y el departamento 30

$\sigma_{JEFE=7839, N^{\circ}DEPART=30} (EMPLEADOS)$

Nº EMPLE	APELLIDO	SALARIO	COMISION	Nº.DEPART	JEFE
7698	NEGRO	3705		30	7839

Ejercicio 5

A partir de la tabla EMPLEADOS mostrada en la fig 2.19, obtén el APELLIDO Y SALARIO de aquellos empleados cuyo JEFE se corresponda con el numero 7839 y el departamento sea el 30.

$\Pi_{APELLIDO, SALARIO} (\sigma_{JEFE=7839, N^{\circ}DEPART=30} (EMPLEADOS))$

APELLIDO	SALARIO
NEGRO	3705

Ejercicio 4 pag. 93, del Capítulo 2 del Libro.

A partir de las siguientes tablas:

ALUMNOS (DNI, Nombre, Dirección, Telef, Curso)
 ASIGNATURAS (Códigoasig, Nombreasig)
 NOTAS (DNINo, Códigoasignatura, Nota)

a) Define las claves primarias y ajenes.

TABLA ALUMNOS

PK

<u>DNI</u> No	Nombre	Dirección	Telef	Curso

TABLA ASIGNATURAS

PK

Códigoasig	Nombreasig

TABLA NOTAS

(FK) (FK)

DNI	Nombre	Nota

b)Obten expresiones relacionales para:

- Los alumnos de primer curso (Curso=1).

$\sigma_{\text{Curso}=1}(\text{ALUMNOS})$

- Los alumnos de primer curso, solo las columnas DNI y Nombre.

$\pi_{\text{DNI}, \text{Nombre}}(\sigma_{\text{Curso}=1}(\text{ALUMNOS}))$

- El nombre de los alumnos, el código de asignatura y la Nota de los alumnos de primer curso.

$\pi_{\text{Nombre}, \text{CódigoAsig}, \text{Nota}}(\sigma_{\text{Curso}=1}(\text{ALUMNOS} \text{*} \text{NOTAS}) \text{DNI}=\text{DNI}_\text{No})$

- Los DNI de los alumnos cuyo nombre de asignatura es INFORMATICA.

$\pi_{\text{DNI}_\text{No}}(\sigma_{\text{NombreAsignatura}=\text{INFORMATICA}}(\text{ASIGNATURA} \text{*} \text{NOTAS}) \text{CódigoAsignatura}=\text{Códigoasig})$

—Ejercicios Unidad 03 —**Ejercicio 1**

a. - A partir de la tabla **EMPLEADOS** haz una selección de las filas cuyo departamento es el 20 mostrando la operación a realizar y representando el resultado en forma de tabla.

EMPLEADOS

N_EMPL	APELLIDO	SALARIO	COMISIÓN	N_DEPAR	JEFE
7369	SÁNCHEZ	1040		20	7902
7499	ARROYO	2080	390	30	7698
7521	SALA	1625	650	30	7698
7566	JIMÉNEZ	3867		20	7839
7654	MARTÍN	1625	1820	30	7698
7698	NEGRO	3705		30	7839
7782	CEREZO	3185		10	7839
7788	GIL	3900		20	7566
7839	REY	6500		10	
7876	ALONSO	1430		20	7788

ON_DEPAR=20 (EMPLEADOS)

EMPLEADOS

N_EMPL	APELLIDO	SALARIO	COMISIÓN	N_DEPAR	JEFE
7369	SÁNCHEZ	1040		20	7902
7566	JIMÉNEZ	3867		20	7839
7788	GIL	3900		20	7566
7876	ALONSO	1430		20	7788

b. - Proyecta la tabla **EMPLEADOS** anterior según las columnas **APELLIDO** y **SALARIO**. Muestra la operación a realizar y representa el resultado en forma de tabla.

ΠAPELLIDO,SALARIO (EMPLEADOS)

APELLIDO	SALARIO
SÁNCHEZ	1040
ARROYO	2080
SALA	1625
JIMÉNEZ	3867
MARTÍN	1625
NEGRO	3705
CEREZO	3185
GIL	3900
REY	6500
ALONSO	1430

c. - Dadas las tablas **EMPLE1** y **EMPLE2** representa en forma de tabla el resultado de las siguientes operaciones algebraicas:

EMPLE1

N_EMPL	NOMBRE
1001	Alain
1005	Inge

EMPLE2

N_EMPL	NOMBRE
2001	Kepa
2010	Nahia
1005	Inge

1) **EMPLE1 U EMPLE2**

N_EMPL	NOMBRE
1001	Alain
1005	Inge
2001	Kepa
2010	Nahia

2) **EMPLE1 ∩ EMPLE2**

N_EMPL	NOMBRE
1005	Inge

3) **EMPLE1 - EMPLE2**

N_EMPL	NOMBRE
1001	Alain

4) **EMPLE2 - EMPLE1**

N_EMPL	NOMBRE
2001	Kepa
2010	Nahia

d.- Dadas las tablas **VENTAS** y **ARTÍCULOS** representa en forma de tabla el resultado de las siguientes operaciones algebraicas:

1) **VENTAS x ARTÍCULOS**

2) **(VENTAS * ARTÍCULOS)** CODI=CÓDIGO

VENTAS**ARTÍCULOS**

CODI	FECHA	CANTIDAD
5100	18/11/03	100
5200	19/11/03	120
5100	19/11/03	45

CÓDIGO	DENOM	EXIST	PVP
5100	Patatas	500	0,78
5200	Cebollas	250	0,90

1) VENTAS x ARTÍCULOS**VENTAS****ARTÍCULOS**

CODI	FECHA	CANTIDAD	CÓDIGO	DENOM	EXIST	PVP
5100	18/11/03	100	5100	Patatas	500	0,78
5100	18/11/03	100	5200	Cebollas	250	0,9
5200	19/11/03	120	5100	Patatas	500	0,78
5200	19/11/03	120	5200	Cebollas	250	0,9
5100	19/11/03	45	5100	Patatas	500	0,78
5100	19/11/03	45	5200	Cebollas	250	0,9

2) (VENTAS * ARTÍCULOS) CODI=CÓDIGO**VENTAS****ARTÍCULOS**

CODI	FECHA	CANTIDAD	CÓDIGO	DENOM	EXIST	PVP
5100	18/11/03	100	5100	Patatas	500	0,78
5200	19/11/03	120	5200	Cebollas	250	0,9
5100	19/11/03	45	5100	Patatas	500	0,78

Ejercicio 2

a.- Inventa varias tablas describiendo sus columnas, dominios, claves primarias y ajenas.

MICROPROCESADORES

C_CPU	NOMBRE	FABRICANTE	VELOCIDAD	SOCKET	FAMILIA
I1224	Tualatin Celeron	INTEL	1GHz	370	Family 6 model 11
I1234	Core 2 Duo T7800	INTEL	2,60GHz	M	Family 6 model 15
I2445	Core i5 650	INTEL	3,2GHz	1156LGA	Family 6 model E
I2466	Core i7 940	INTEL	2,93GHz	1366LGA	Family 6 model F
A1789	Athlon 64 X2 3800	AMD	3,8GHz	AM2	X86-64
A1223	Semprom 3200	AMD	3,2GHz	AM2	X86-32
A1824	Athlon 64 X2 FX-74	AMD	3,6GHz	F	X86-64
I2510	Core 2 Extrem X7900	INTEL	2,8GHz	P	Family 6 model F

b.- A partir de las siguientes tablas:

AGENDA

Nombre	Edad	CodProv	Telef
Ana	35	91	2266778
Rosi	35	925	990087
Nieves	36	925	409876
Raquel	38	925	324567
Verónica	37	926	234563
Paz	40	91	9887622
Pili	37	925	332456
Antonio	44	949	220099
Rafa	39	949	303090

PROVINCIAS

Código	NombreProv
91	Madrid
925	Toledo
949	Guadalajara
926	Ciudad Real

Escribe las tablas resultantes después de realizar las siguientes operaciones:

1) $\sigma_{\text{Edad} > 37}$ (AGENDA)

Nombre	Edad	CodProv	Telef
Raquel	38	925	324567
Rafa	39	949	303090
Paz	40	91	9887622
Antonio	44	949	220099

2) $\Pi_{\text{Nombre}, \text{Edad}}$ (AGENDA)

Nombre	Edad
Ana	35
Rosi	35
Nieves	36
Raquel	38
Verónica	37
Paz	40
Pili	37
Antonio	44
Rafa	39

3) (AGENDA * PROVINCIAS) $\text{CodProv} = \text{Código}$

Nombre	Edad	CodProv	Telef	NombreProv
Ana	35	91	2266778	Madrid
Rosi	35	925	990087	Toledo
Nieves	36	925	409876	Toledo
Raquel	38	925	324567	Toledo
Verónica	37	926	234563	Ciudad Real
Paz	40	91	9887622	Madrid
Pili	37	925	332456	Toledo
Antonio	44	949	220099	Guadalajara
Rafa	39	949	303090	Guadalajara

4) $\Pi_{\text{Nombre}, \text{CodProv}, \text{Telef}, \text{NombreProv}} ((\text{AGENDA} * \text{PROVINCIAS}) \text{ CodProv} = \text{Código})$

Nombre	CodProv	Telef	NombreProv
Ana	91	2266778	Madrid
Rosi	925	990087	Toledo
Nieves	925	409876	Toledo
Raquel	925	324567	Toledo
Verónica	926	234563	Ciudad Real
Paz	91	9887622	Madrid
Pili	925	332456	Toledo
Antonio	949	220099	Guadalajara
Rafa	949	303090	Guadalajara

5) $\Pi_{\text{Nombre}, \text{CodProv}, \text{Telef}, \text{NombreProv}} (\sigma_{\text{Edad} > 37} (\text{AGENDA} * \text{PROVINCIAS}) \text{ CodProv} = \text{Código})$

Nombre	CodProv	Telef	NombreProv
Raquel	925	324567	Toledo
Rafa	949	303090	Guadalajara
Paz	91	9887622	Madrid
Antonio	949	220099	Guadalajara

c. - ¿Qué columnas o conjuntos de columnas pueden ser claves primarias y ajena en las tablas anteriores?

AGENDA: Nombre, Edad, Teléfono

PROVINCIAS: Código

d. - Si un usuario necesita solamente acceder a las columnas Nombre, Edad y NombreProv de las tablas anteriores, ¿qué solución se le puede dar?

$\Pi_{\text{Nombre}, \text{Edad}, \text{NombreProv}} ((\text{AGENDA} * \text{PROVINCIAS}) \text{ CodProv} = \text{Código})$

Ejercicio 3

CLUB
CODCLUB
NOMBRE
DIRECCIÓN
POBLACIÓN
PROVINCIA
CODPOSTAL
TFNO
FAX
AÑOFUNDACIÓN
PRESUPUESTO
COLORES
HIMNO
PRESIDENTE
VICEPRESIDENTE
CODEQUIPACIÓN
SPONSOR
CODESTADIO

JUGADOR
CODJUGADOR
CLUB
LIGA
JUGADOR
DORSAL
CODPAIS
CODDEMAR
FICHA
INICIOCONTRATO
FINCONTRATO

JUGADOR LIGA ACTUAL
CODJUGADOR
CLUB
LIGA

JUGADOR LIGA PASADA
CODJUGADOR
CLUB
LIGA

a.- Dada la relación JUGADOR, las vistas JUGADOR LIGA ACTUAL y JUGADOR LIGA PASADA obtenidas de la relación anterior, y la relación CLUB, crear un ejemplo de cada tipo de operación unitaria o binaria.

- a) Sacar una relación de todos los jugadores que juegan en la liga actual pero no jugaron en la pasada
- b) Selecciona todos los jugadores de la relación jugador que juegan como portero
- c) Selecciona el código del club, el nombre del club y nombre de presidente del Club
- d) Relación de todos los jugadores que jugaron en la liga pasada y la liga actual (tiene que jugar en las dos)
- e) Combina cada registro de la tabla club con todos los registros de la tabla jugador
- f) Obtener una relación que contenga información del jugador y del club al que pertenece
- g) Todos los jugadores de la liga pasada y también los jugadores de la liga actual (no tienen porque haber jugado en las 2)

b.- Hacer un diseño gráfico de las operaciones anteriores.

- a) JUGADOR LIGA ACTUAL - JUGADOR LIGA PASADA
- b) $\sigma_{CODDEMAR=\text{Portero}}(\text{JUGADOR})$
- c) $\Pi_{CODCLUB, \text{NOMBRE}, \text{PRESIDENTE}}(\text{CLUB})$
- d) JUGADOR LIGA ACTUAL \cap JUGADOR LIGA PASADA
- e) CLUB \times JUGADOR
- f) $(\text{CLUB} * \text{JUGADOR}) \text{ CODCLUB} = \text{CLUB}$
- g) JUGADOR LIGA PASADA \cup JUGADOR LIGA ACTUAL

c.- Definir la clave primaria de la relación JUGADOR.

CODJUGADOR

d.- Imponer restricciones a las relaciones JUGADOR y CLUB. Justificar esas restricciones.

e. - Calcular el total de defensas del Valladolid, a partir de la tabla JUGADOR.

f. - Calcular la ficha promedio que tienen los jugadores del Zaragoza.

g. - A partir de la fecha de término de contrato, crear una consulta que contenga el nombre del jugador y el tiempo para el fin de contrato.

h. - Usando el álgebra relacional, a partir de las relaciones JUGADOR y CLUB, crear una nueva relación con el nombre del jugador y la dirección fiscal completa del club.

Ejercicio 4

Partiendo de las siguientes tablas:

ALUMNOS (DNI, NOMBRE, DIRECCION, TELEF, CURSO)

ASIGNATURAS (CODIGOASIG, NOMBREASIG)

NOTAS (DNI_NO, CODIGOASIGNATURA, NOTA)

1) Definir las claves primarias y ajenas.

2) Obtener expresiones relacionales para:

a. - Obtener los alumnos de primer curso.

b. - Obtener los DNI de los alumnos cuyo nombre de asignatura es INFORMATICA.

c. - Obtener los nombres de alumnos que tengan un 7 en INFORMATICA.

1) ALUMNOS (DNI, NOMBRE, DIRECCION, TELEF, CURSO)
ASIGNATURAS (CODIGOASIG, NOMBREASIG)
NOTAS (DNI_NO(FK), CODIGOASIGNATURA(FK), NOTA)

2)

a) $\sigma_{CURSO=1}(\text{ALUMNOS})$

b) $\pi_{DNI_NO}(\sigma_{NOMBREASIG=INFORMATICA}(\text{ASIGNATURA} * \text{NOTAS}))$ CODASIG = CODIGOASIGNATURA

c) $\pi_{NOMBRE}(\sigma_{NOTA=7 \text{ AND } NOMBREASIG=INFORMATICA}(\text{ALUMNOS} * (\text{NOTAS} * \text{ASIGNATURAS})))$ CODASIG = CODIGOASIGNATURA AND DNI=DNI_NO

Ejercicio 5

Si el contenido de las tablas es el siguiente:

LIBRO

CodLib	Titulo	NumPag	CodTipo_L
1111	XX	200	4
2222	AA	250	3
2244	BB	260	4
2266	CC	200	7
2288	DD	150	3
3300	EE	158	3

TIPO_LIBRO

CodTipo	Desc
3	Novela
4	Ensayo
7	Teatro

L_E**ESCRITOR**

DNI	NomApe	Tel	Edad	Pobsa	Prov
22	A.G	123456	18	Orereta	Gipuzkoa
30	L.A.	997654	47	Mendexa	Bizkaia
44	B.A	445566	20	Durango	Bizkaia
48	E.L.	330022	36	Aramaio	Araba

CodLib_LE	DNI_LE
1111	44
2222	22
2244	48
2266	44
2288	30
3300	48

1.- ¿Cuál será el resultado de las siguientes operaciones del álgebra relacional?:

a.- $(L_E * ESCRITOR) \text{ DNI} = \text{DNI_LE}$

DNI	NomApe	Tel	Edad	Pobsa	Prov	CodLib_LE
22	A.G	123456	18	Orereta	Gipuzkoa	2222
30	L.A.	997654	47	Mendexa	Bizkaia	2288
44	B.A	445566	20	Durango	Bizkaia	1111
44	B.A	445566	20	Durango	Bizkaia	2266
48	E.L.	330022	36	Aramaio	Araba	2244
48	E.L.	330022	36	Aramaio	Araba	3300

b.- $((L_E * ESCRITOR) \text{ DNI} = \text{DNI_LE}) * LIBRO \text{ CodLib} = \text{CodLib_LE}$

DNI	NomApe	Tel	Edad	Pobsa	Prov	CodLib_LE	Titulo	NumPag	CodTipo_L
22	A.G	123456	18	Orereta	Gipuzkoa	2222	AA	250	3
30	L.A.	997654	47	Mendexa	Bizkaia	2288	DD	150	3
44	B.A	445566	20	Durango	Bizkaia	1111	XX	200	4
44	B.A	445566	20	Durango	Bizkaia	2266	CC	200	7
48	E.L.	330022	36	Aramaio	Araba	2244	BB	260	4
48	E.L.	330022	36	Aramaio	Araba	3300	EE	158	3

c.- $\pi_{\text{Titulo}, \text{NomApe}} (((L_E * ESCRITOR) \text{ DNI} = \text{DNI_LE}) * LIBRO) \text{ CodLib} = \text{CodLib_LE}$

Titulo	NomApe
AA	A.G
DD	L.A.
XX	B.A
CC	B.A
BB	E.L.
EE	E.L.

2.- Basándote en el modelo relacional obtenido en el ejercicio a), escribe las expresiones oportunas para obtener los siguientes resultados:

a.- El nombre y teléfono de los escritores de 20 años.

$\pi_{\text{NomApe}, \text{Tel}} (\sigma_{\text{Edad}=20} (\text{SCRITOR}))$

b.- El código, título y la descripción del tipo de los libros que tienen más de 200 páginas.

$\Pi_{\text{CodLib}, \text{Titulo}, \text{Desc}} (\sigma_{\text{NumPag}>200} (\text{LIBRO} * \text{TIPO_LIBRO}) \text{CodTipo_L} = \text{CodTipo})$

c.- Los títulos de los libros que han sido escritos por autores de 20 años.

$\Pi_{\text{Titulo}} (\sigma_{\text{Edad}=20} ((\text{LIBRO} * \text{L_E})_{\text{CodLib}=\text{CodLib_LE}} * \text{ESCRITOR}) \text{DNI} = \text{DNI_LE})$

o tambien

$\Pi_{\text{Titulo}} (\sigma_{\text{Edad}=20} (\text{LIBRO} * \text{L_E} * \text{ESCRITOR}) \text{CodLib} = \text{CodLib_LE} \text{ AND } \text{DNI} = \text{DNI_LE})$

d.- Los títulos de los libros de más de 100 páginas que han sido escritos por autores de menos de 20 años.

$\Pi_{\text{Titulo}} (\sigma_{\text{EdadNumpag}>100} (\sigma_{\text{Edad}<20} (\text{LIBRO} * \text{L_E} * \text{ESCRITOR}) \text{CodLib} = \text{CodLib_LE} \text{ AND } \text{DNI} = \text{DNI_LE}))$

e.- Las descripciones de los tipos de libros escritos por autores de Bizkaia.

$\Pi_{\text{Desc}} (\sigma_{\text{Prov}=\text{Bizkaia}} ((\text{LIBRO} * \text{TIPO_LIBRO})_{\text{CodTipo_L} = \text{CodTipo}} * \text{L_E}_{\text{CodLib} = \text{CodLib_LE}}) * \text{ESCRITOR}) \text{DNI} = \text{DNI_LE)))$

Ejercicio 6

Disponemos de las siguientes tablas:

DESARROLLA

ID_Fab	Código
1	1
1	2
1	3
1	4
1	5
2	6
2	7
2	8
2	9
2	10
2	11
2	12
6	13
4	14
5	15
5	16
3	17
3	18
5	19

PROGRAMA

Código	Nombre	Versión
1	Application Server	9i
2	Database	8i
3	Database	9i
4	Database	10g
5	Developer	6i
6	Access	97
7	Access	2000
8	Access	XP
9	Windows	98
10	Windows	XP Professional
11	Windows	XP Home Edition
12	Windows	2003 Server
13	Norton Internet security	2004
14	Freddy Hardest	-
15	Paradox	2
16	C++ Builder	5.5
17	DB/2	2.0
18	OS/2	1.0
19	Jbuilder	X

DISTRIBUYE

CIF	Código	Cantidad
1	1	10
1	2	11
1	6	5
1	7	3
1	10	5
1	13	7
2	1	6
2	2	6
2	6	4
2	7	7
3	10	8
3	13	5
4	14	3
4	20	6
5	15	8
5	16	2
5	17	3
5	19	6
5	8	8

FABRICANTE

ID_Fab	Nombre	País
1	Oracle	Estados Unidos
2	Microsoft	Estados Unidos
3	IBM	Estados Unidos
4	Dinamic	España
5	Borland	Estados Unidos
6	Symantec	Estados Unidos

COMERCIO

CIF	Nombre	Ciudad
1	El Corte Inglés	Sevilla
2	El Corte Inglés	Madrid
3	Jump	Valencia
4	Centro Mail	Sevilla
5	FNAC	Barcelona

REGISTRA

CIF	DNI	Código	Medio
1	1	1	Internet
1	3	4	Tarjeta postal
4	2	10	Teléfono
4	1	10	Tarjeta postal
5	2	12	Internet
2	4	15	Internet

CLIENTE

DNI	Nombre	Edad
1	Pepe Pérez	45
2	Juan González	45
3	María Gómez	33
4	Javier Casado	18
5	Nuria Sánchez	29
6	Antonio Navarro	58

1.- ¿Cuál será el resultado de las siguientes operaciones del álgebra relacional?:

a.- $\sigma_{\text{Nombre} = \text{"Access"}}$ (PROGRAMA)

Código	Nombre	Versión
6	Access	97
7	Access	2000
8	Access	XP

b.- $\sigma_{\text{Nombre} = \text{"Access"} \text{ AND } \text{Código} < 8}$ (PROGRAMA)

6	Access	97
7	Access	2000

c.- π_{Nombre} (COMERCIO)

Nombre
El Corte Inglés
El Corte Inglés
Jump
Centro Mail
FNAC

d.- $\pi_{\text{CIF}, \text{Nombre}}$ (COMERCIO)

CIF	Nombre
1	El Corte Inglés
2	El Corte Inglés
3	Jump
4	Centro Mail
5	FNAC

e.- $\pi_{\text{Nombre}} (\sigma_{\text{CIF} = 3} (\text{COMERCIO}))$

CIF	Nombre	Ciudad
3	Jump	Valencia

f.- PROGRAMA x COMERCIO

Código	Nombre	Versión	CIF	Nombre	Ciudad
1	Application Server	9i	1	El Corte Inglés	Sevilla
1	Application Server	9i	2	El Corte Inglés	Madrid
1	Application Server	9i	3	Jump	Valencia
1	Application Server	9i	4	Centro Mail	Sevilla
1	Application Server	9i	5	FNAC	Barcelona

2.- Utiliza la notación del álgebra relacional para:

a.- seleccionar los datos de los clientes de edad inferior a 30 años

$\sigma_{\text{Edad} < 30} (\text{CLIENTE})$

b.- seleccionar los datos de los clientes de edad inferior a 30 años o superior a 45 años

$\sigma_{\text{Edad} < 30 \text{ AND Edad}>45} (\text{CLIENTE})$

c.- seleccionar los datos de los fabricantes de Estados Unidos

$\sigma_{\text{PAIS}=\text{Estados Unidos}} (\text{FABRICANTE})$

d.- seleccionar los datos de las distintas versiones de Windows

$\sigma_{\text{Nombre}=\text{Windows}} (\text{PROGRAMA})$

e.- determinar el CIF de El Corte Inglés de Sevilla

$\pi_{\text{CIF}} (\sigma_{\text{Nombre}=\text{El Corte Ingles AND Ciudad}=Sevilla} (\text{COMERCIO}))$

f.- seleccionar los nombres de los distintos programas

$\pi_{\text{Nombre}} (\text{PROGRAMA})$

g.- consultar todos los datos guardados sobre los programas

$\sigma (\text{PROGRAMA})$

h.- obtener un listado con los medios usados para registrarse

$\pi_{\text{Medio}} (\text{REGISTRA})$

i.- obtener un listado de las ciudades con algún establecimiento, sin que aparezcan valores duplicados

$\pi_{\text{Ciudad}} (\text{CLIENTE})$, con este comando salen ciudades repetidas y en SQL hay una instrucción para que solo salga uno de cada y se hace con DISTINCT

j.- obtener un listado de los nombres de los clientes de edad inferior a 30 años

$\pi_{\text{Nombre}} (\sigma_{\text{Edad} < 30} (\text{CLIENTE}))$

k.- obtener un listado de los nombres de los clientes de edad inferior a 30 años o superior a 45

$\pi_{\text{Nombre}} (\sigma_{\text{Edad} < 30 \text{ OR Edad}>45} (\text{CLIENTE}))$

I.- obtener un listado del nombre y DNI de los clientes de edad inferior a 30 años o superior a 45

$\Pi_{\text{nOMBRE}, \text{DNI}} (\sigma_{\text{Edad} < 30 \text{ OR } \text{Edad} > 45} (\text{CLIENTE}))$

m.- obtener un listado con todas las combinaciones posibles de clientes y de programas
 $\text{CLIENTES} \times \text{PROGRAMA}$

n.- obtener una relación en la que se muestren los datos de cada programa junto con los de la(s) empresa(s) que lo desarrollan(n)

$((\text{PROGRAMA} * \text{DESARROLLA})_{\text{PROGRAMACodigo} = \text{DESARROYACodigo}}) * \text{FABRICANTE}_{\text{DESARROLLA ID.FAB} = \text{FABRICANTE ID.FAB}}$

ñ.- obtener una relación en la que se muestren los datos de cada programa junto con los del (los) comercio(s) que lo distribuye(n)

$((\text{PROGRAMA} * \text{DISTRIBUYE})_{\text{PROGRAMACodigo} = \text{DISTRIBUYECodigo}}) * \text{COMERCIO}_{\text{DISTRIBUYE NIF} = \text{COMERCIO NIF}}$

o.- obtener una relación en la que se muestren los datos de cada fabricante junto con los de (los) comercio(s) que distribuye(n) sus programas

$\Pi_{\text{FABRICANTE}, \text{Nombre}, \text{PAIS}, \text{COMERCIO}, \text{CIUDAD}} (\text{FABRICANTE} * \text{DESARROLLA})_{\text{FABRICANTE ID.FAB} = \text{DESARROLLA ID.FAB}} * \text{DISTRIBULLE}_{\text{DESARROLLA CODIGO} = \text{DISTRIBULLE CODIGO}} * \text{COMERCIO}_{\text{DISTRIBUYE CIF} = \text{COMERCIO CIF}}$

Ejercicio 7

1.- ¿Cuál es el resultado de unir una tabla consigo misma?

Es ella misma

2.- Tras la fusión de dos entidades bancarias, sería lógico poder acceder desde las oficinas de una a los datos de los clientes de la otra. ¿Qué operación crees que debería usarse para crear una base de datos única a partir de las bases de datos de los dos bancos originales?

Suponiendo que las tablas tengan el mismo numero de columnas y atributos compatibles la Unión (U).

3.- Supón que deseas conocer si tus datos aparecen en una base de datos de un comercio. ¿Qué operación deberías utilizar?

Selección (σ) con la condición de DNI or Nombre, Apellido

4.- Si se aplicase la operación de intersección a las tablas de clientes de las bases de datos de dos bancos, ¿qué se obtendría?

Los clientes que tienen cuenta en ambos bancos

5.- ¿Cuál es el resultado de $A \cap (B \cup A)$? ¿Y de $A \cup (B \cap A)$?

Del primero la Tabla A y del segundo también la Tabla A

6.- Si se aplicase la operación de diferencia a las tablas de clientes de las bases de datos de dos bancos, ¿qué se obtendría?

Los clientes que están solo en uno de los bancos

7.- ¿Es el operador de diferencia, conmutativo? Pon un ejemplo.

NO, 3×5 es igual que 5×3 , pero $3 - 5$ no es igual que $5 - 3$

Ejercicio 8**R1**

A	B	C
a	b	c
d	a	f
c	b	d

S1

A	B	C
b	g	a
d	a	f

R2

A	B	C	D
a	b	c	d
a	b	e	f
b	c	e	f
e	d	c	d
e	d	e	f
a	b	d	e

S2

C	D
c	d
e	f

R3

A	B	C
1	2	3
4	5	6
7	8	9

S3

D	E
3	1
6	2

1.- ¿Cuál será el resultado de las siguientes operaciones del álgebra relacional?:**a.-** $\sigma_{B=b}$ (R1)

A	B	C
a	b	c
c	b	d

b.- $\Pi_{A,C}$ (R1)

A	C
a	c
c	d

c.- R1 U S1

A	B	C
a	b	c
c	b	d
b	g	a
d	a	f

d.- R1 - S1

A	B	C
a	b	c
c	b	d

e.- R1 x S1

A	B	C	A	B	C
a	b	c	b	g	a
a	b	c	d	a	f
d	a	f	b	g	a
d	a	f	d	a	f
c	b	d	b	g	a
c	b	d	d	a	f

f.- $(R3 * S3)_{B < D}$

A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
4	5	6	6	2

g.- R1 ∩ S1

A	B	C
d	a	f

h.- R2 / S2

1º Paso	
A	B
a	b
a	b
b	c
e	d
e	d

2º Paso	
A	B
a	b
e	d

Ejercicio 9

R	
A	B
a	b
c	b
d	e

S	
B	C
b	g
b	d
e	a

Hallar:

a. - $R \cup S$

No se puede hacer

b. - $R - S$

No se puede hacer

c. - $R \times S$

A	R.B	S.B	C
a	b	b	g
a	b	b	d
a	b	e	a
c	b	b	g
c	b	b	d
c	b	e	a
d	e	b	g
d	e	b	d
d	e	e	a

d. - $\Pi_{A, R.B, C} (\sigma_{R.B = S.B} (R \times S))$

e. - $\Pi_{A, S.B, C} ((R * S)_{R.B = S.B})$

A	R.B	C
a	b	g
a	b	d
c	b	g
c	b	d
d	e	a

A	S.B	C
a	b	g
a	b	d
c	b	g
c	b	d
d	e	a

f. - $(\sigma_{A = C} (R \times S))$

A	R.B	S.B	C
a	b	e	a
d	e	b	d

Ejercicio 10

R	A	B	C	D
a	b	a	a	
c	a	c	b	
a	b	b	b	
a	a	c	b	
b	b	a	b	
c	b	b	b	

T	A	B	D	E
b	a	a	a	a
c	a	b	c	
c	c	b	a	
a	c	b	c	
a	a	b	c	
a	c	b	a	
c	b	b	c	
b	a	b	c	

S	A	B
a	b	
c	b	

Calcula $((R/S) \times (T/S)) \times S$

R/S	T/S 1º paso	
C	D	E
b	b	

T/S 2º paso	D	E

$$(R/S) \times (T/S) = (R/S) \times \text{Null} = R/S$$

R/S	C	D
	b	b

$$(R/S) \times S$$

(R/S) \times S	C	D	A	B
	b	b	a	b
	b	b	c	b

Ejercicio 11

R	A	B	C	D
a	b	c	d	
c	d	a	b	
g	c	b	f	
c	d	c	d	
a	b	a	b	
g	c	g	h	
a	b	b	f	

S	C	D
c	d	
a	b	

T	A	B
g	c	

Calcula $((R/S) * (R/T))_{B=c} \cap R$

(R/S)

1º Paso

A	B
a	b
c	d
c	d
a	b

R/S

A	B
a	b
c	d

(R/T)

1º Paso

C	D
b	f
g	h
b	f

R/T

C	D
b	f
g	h

 $((R/S) * (R/T))_{B=c}$

R/S*R/T

A	B	C	D
a	b	b	f

 $((R/S) * (R/T))_{B=c} \cap R$

No se puede hacer la intersección al tener diferente numero de columnas.

(R/S*R/T) ∩ R

A	B	C	D
a	b	b	f

Ejercicio 12

P	A	B	C	D
	1	2	7	8
	3	4	1	2
	7	3	2	6
	1	2	3	4
	1	2	1	2
	3	4	3	4
	7	3	7	8

Q	C	D
	3	4
	1	2

R	A	B
	7	3

Calcula $((P/Q) * (P/R))_{B \neq C} \cap P$

1º Paso

A	B
3	4
1	2
1	2
3	4

P/Q

A	B
3	4
1	2

1º Paso

C	D
2	6
7	8

P/R

C	D
2	6
7	8

$(P/Q) * (P/R))_{B \neq C}$

A	B	C	D
1	2	7	8
3	4	2	6
3	4	7	8

Resultado:

A	B	C	D
1	2	7	8

Ejercicio 13**R**

A	B	C
a	b	c
b	b	c
a	c	d
b	c	d
c	b	c
c	c	b
a	b	b

S

B	C
b	c
c	d

Calcula $((R/S) \times S) / S$:**R/S**

A	
a	si
b	si
a	
b	
c	no
c	
a	

A
a
b

(R/S) × S

A	B	C
a	b	c
a	c	d
b	b	c
b	c	d

((R/S) × S) / S

A
a
a
b
b

A
a

Ejercicio 14

Dadas las siguientes relaciones:

- . SUMINISTRADOR (CodS, Nombre, Ciudad)
Cada fila representa un suministrador o proveedor de artículos.
- . PIEZAS (CodP, Nombre, Peso, Precio)
Cada fila representa una pieza.
- . PROYECTO (CodY, Nombre, Presupuesto)
Cada fila representa un proyecto.
- . SUMINISTROS (CodS, CodY, CodP, Cantidad)
El suministrador CodS ha suministrado para el proyecto CodY la pieza CodP en la cantidad especificada.

Escribe en álgebra relacional las expresiones necesarias para obtener la siguiente información:

- 1.- Piezas suministradas por los proveedores de Donostia.
- 2.- Proyectos que usan, al menos, alguna pieza suministrada por el proveedor "P1".
- 3.- Presupuesto de los proyectos que usan, al menos, alguna pieza suministrada por el proveedor "P1".
- 4.- Proveedores que suministran una misma pieza al menos a todos los proyectos.
- 5.- Proyectos que usan, al menos, todas las piezas suministradas por el proveedor "P3".
- 6.- Proyectos abastecidos por el proveedor "P3" con todas las piezas que éste suministra.
- 7.- Proyectos abastecidos por todos los proveedores que suministran alguna pieza de 3 gramos.

Ejercicio 15

- 1.- El operador de intersección es un operador derivado en el álgebra relacional. Exprésalo en función de la diferencia.
- 2.- El operador de combinación es un operador derivado en el álgebra relacional. Exprésalo en función de los operadores primitivos necesarios.

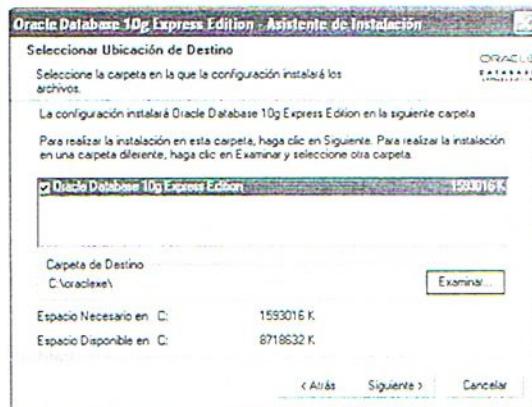
INSTALACION DE ORACLE 10GXE

REQUISITOS HW Y SW:

- Sistema operativo Windows NT Server, Windows 2000 Profesional, Windows XP Profesional , Windows Server 2003.
- Mínimo 256 MB de memoria RAM, aunque se recomienda 512MB.
- Disco duro 2GB + el doble de RAM de memoria virtual.
- Pentium III o superior.

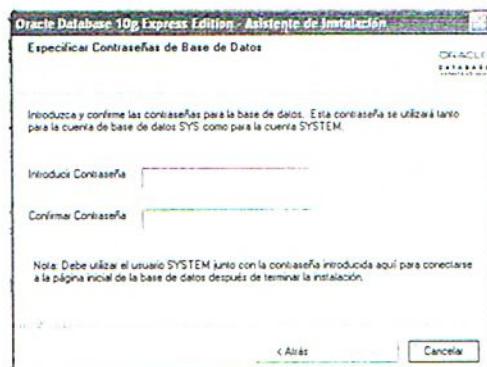
PASO 1:

- Entrar en la carpeta Oracle XE Windows y ejecutar el archivo *OracleXE.exe*.
- En la pantalla de Bienvenida pulsamos *Siguiente*.
- Aceptamos los Términos del Acuerdo de Licencia. Pulsamos *Siguiente*.
- Seleccionamos la carpeta donde se instalará Oracle 10gXE. Dejamos la que está por defecto. *Siguiente*.



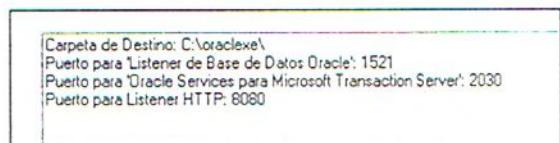
PASO 2:

- A continuación nos pide que introduzcamos las claves de los usuarios Administradores del Sistema: **SYSTEM** y **SYS**
- Como contraseña utilizaremos *manager*.



PASO 3:

Aparece una pantalla resumen para que confirmemos todos los datos. Verificamos que todo es correcto. Si es así pulsamos *Instalar*.

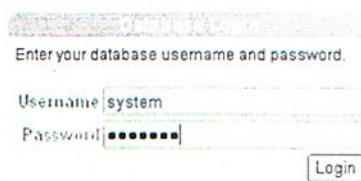
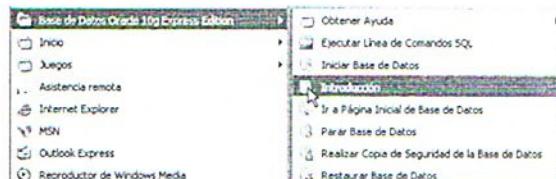


PASO 4:

- Durante la instalación, en la barra de progreso veremos cómo va el proceso.
- Al cabo de unos minutos aparecerá esta pantalla que indica que la instalación se ha realizado correctamente. Pulsamos *Terminar*.

**PASO 5:**

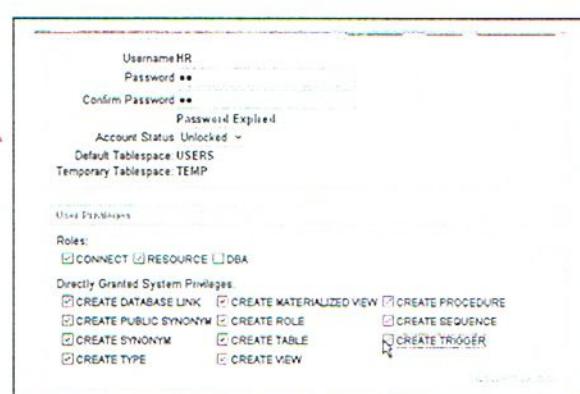
- Vamos al menú inicio de Windows y comprobamos que se han creado las entradas de la pantalla →
- Vamos a la opción de *Ir a Página Inicial de Bases de Datos*.
- Aparece una pantalla donde tenemos que introducir el usuario SYSTEM y la contraseña que hemos elegido al instalar la base de datos. Pulsamos *Login*.

**PASO 6:**

- Pulsamos el ícono *Administration* y luego *Database Users*.
- Aparecerá la siguiente pantalla →
- Pulsamos el ícono *Users*.
- Aparece el usuario HR que el usuario que viene creado por defecto en la base de datos de Oracle XE.

**PASO 7:**

- Tenemos que dar permisos al usuario HR. Como contraseña le daremos HR. Y como Access Status Unlocked. Le daremos todos los roles menos el de Administrador y todos los privilegios.
- Pulsamos *Alter User* para guardar las modificaciones.
- Volvemos a la página inicial de base de datos e intentamos entrar como usuario HR para verificar que lo hemos hecho todo correctamente.



PRIMEROS PASOS EN ORACLE XE

Comprobar que la base de datos funciona correctamente. Siempre hay 2 opciones a partir de la pantalla de MSDOS o de modo Visual.

- a. Inicio - Todos los programas-
Base de datos Oracle 10g Express edition -
Ejecutar Línea de Comandos SQL

CONNECT HR/HR

Tiene que aparecer un mensaje de que está Conectado.

- b. Inicio - Todos los programas-
Base de datos Oracle 10g Express edition -
Ir a Página inicial de Base de datos.

Conectarse como HR/HR

CÓMO COPIAR LAS TABLAS A LA BASE DE DATOS

Los archivos .SQL para poder ejecutarse se tienen que copiar a la carpeta:
C:\ORACLEXE\APP\ORACLE\PRODUCT\10.2.0\SERVER\BIN\

Hay 2 maneras:

- * 1. Ejecutar Línea de Comandos SQL

CONNECT hr/hr

START TABLASUNI3.tablas_u_03

~~seguir~~ tablasuni3.sql

Ejecutamos un par de sentencias para comprobar que todo está bien.

SELECT * FROM EMPLE;

SELECT * FROM DEPART;

- 2. Ir a Página inicial de Base de datos.

CONNECT hr/hr

Ejecutamos un par de sentencias para comprobar que todo está bien.

SELECT * FROM EMPLE;

SELECT * FROM DEPART;

Tablas utilizadas en los ejercicios

EMPLE

EMP_NO Number(4)	APELLIDO Varchar2(10)	OFICIO Varchar2(10)	DIR Number(4)	FECHA_ALT Date	SALARIO Number (10)	COMISION Number(10)	DEPT_NO Number(2)
7369	SANCHEZ	EMPLEADO	7902	17/12/1990	1040		20
7499	ARROYO	VENDEDOR	7698	20/02/1990	1500	39000	30
7521	SALA	VENDEDOR	7698	22/02/1991	1625	65000	30
7566	JIMENEZ	DIRECTOR	7839	02/04/1991	2900		20
7654	MARTIN	VENDEDOR	7698	29/09/1991	1600	182000	30
7698	NEGRO	DIRECTOR	7839	01/05/1981	3005		30
7782	CEREZO	DIRECTOR	7839	09/06/1991	2885		10
7788	GIL	ANALISTA	7566	09/11/1991	3000		20
7839	REY	PRESIDENTE		17/11/1991	4100		10
7844	TOVAR	VENDEDOR	7698	08/09/1991	1350	0	30
7876	ALONSO	EMPLEADO	7788	23/09/1991	1430		20
7900	JIMENO	EMPLEADO	7698	03/12/1991	1335		30
7902	FERNANDEZ	ANALISTA	7566	03/12/1991	3000		20
7934	MUÑOZ	EMPLEADO	7782	23/01/1992	16900		10

DEPART

DEPT_NO Number(2)	DNOMBRE Varchar2(14)	LOC Varchar2(14)
10	CONTABILIDAD	SEVILLA
20	INVESTIGACION	MADRID
30	VENTAS	BARCELONA
40	PRODUCCION	BILBAO

LIBRERÍA

TEMA Char(15)	ESTANTE Char(1)	EJEMPLARES Number(2)
Informática	A	15
Economía	A	10
Deportes	B	8
Filosofía	C	7
Dibujo	C	10
Medicina	C	16
Biología	A	11
Geología	D	7
Sociedad	D	9
Labores	B	20
Jardinería	E	6

ALUMNOS

DNI Varchar2(10)	APENOM Varchar2(30)	DIREC Varchar2(30)	POBLA Varchar2(15)	TELEF Varchar2(10)
12344345	Alcalde García, Elena	C/Las Matas, 24	Madrid	917766545
4448242	Cerrato Vela, Luis	C/ Mina 28 - 3A	Madrid	916566545
56882942	Díaz Fernández, María	C/ Luis Vives 25	Móstoles	915577545

NOTAS**ASIGNATURAS**

COD Number(2)	NOMBRE Varchar2(25)
1	Prog. Leng. Estr.
2	Sist. Informáticos
3	Análisis
4	FOL
5	RET
6	Entornos Gráficos
7	Aplic. Entornos 4ª Gen.
8	Idioma

DNI Varchar2(10)	COD Number(2)	NOTA Number(2)
12344345	1	6
12344345	2	5
12344345	3	6
4448242	4	6
4448242	5	8
4448242	6	4
4448242	7	5
56882942	4	8
56882942	5	7
56882942	6	8
56882942	7	9

ALUMO405

DNI Varchar2(10)	NOMBRE Varchar2(15)	APELLIDOS Varchar2(20)	FECHA_NAC Date	DIRECCION Varchar2(20)	POBLACION Varchar2(20)	PROVINCIA Varchar2(20)	CURSO Number(2)	NIVEL Varchar2(10)	CLASE CHAR(1)	FALTAS1 Number(2)	FALTAS2 Number(2)	FALTAS3 Number(2)
34448974N	Concha	Estruch Vidal	10/11/1985	C/ Medico Felix, 12	Berrocalejo	Caceres	2	DAI	A	0	0	
34448999N	Dora	Martin	10/11/1994	C/ Medico	Berrocalejo	Caceres	1	ESO	C	0	0	0

NOTAS_ALUMNOS

NOMBRE_ALUMNO Varchar2(25)	NOTA1 Number(2)	NOTA2 Number(2)	NOTA3 Number(2)
Alcalde García, M. Luisa	5	5	5
Benito Martín, Luis	7	6	8
Casas Martínez, Manuel	7	5	5
Corregidor Sánchez, Ana	6	9	8
Díaz Sánchez, María			7

Ejercicios SQL (Capítulo 3):

- 1. Obtener la descripción de la tabla DEPART

DESC DEPART;

Object Type **TABLE** Object **DEPART**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPART	DEPT_NO	Number	-	2	0	-	-	-	-
	DNOMBRE	Varchar2	14	-	-	-	✓	-	-
	LOC	Varchar2	14	-	-	-	✓	-	-
1 - 3									

- 2. Seleccionar nombre, localidad y N° Departamento de la tabla DEPART

SELECT * FROM DEPART;

DEPT_NO	DNOMBRE	LOC
10	CONTABILIDAD	SEVILLA
20	INVESTIGACION	MADRID
30	VENTAS	BARCELONA
40	PRODUCCION	BILBAO

- 3. Descripción de la tabla EMPLE.

DESC EMPLE;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLE	EMP_NO	Number	-	4	0	-	-	-	-
	APELLIDO	Varchar2	10	-	-	-	✓	-	-
	OFICIO	Varchar2	10	-	-	-	✓	-	-
	DIR	Number	-	4	0	-	✓	-	-
	FECHA_ALT	Date	7	-	-	-	✓	-	-
	SALARIO	Number	-	7	0	-	✓	-	-
	COMISION	Number	-	7	0	-	✓	-	-
	DEPT_NO	Number	-	2	0	-	-	-	-
1 - 8									

- 4. Seleccionar los empleados del Departamento 30 ordenados por oficio en descendente.

**SELECT * FROM EMPLE
WHERE DEPT_NO=30
ORDER BY OFICIO DESC;**

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7499	ARROYO	VENDEDOR	7698	20/02/90	1500	390	30
7521	SALA	VENDEDOR	7698	22/02/91	1625	650	30
7654	MARTIN	VENDEDOR	7698	29/09/91	1600	1020	30
7844	TOVAR	VENDEDOR	7698	08/09/91	1350	0	30
7900	JIMENO	EMPLEADO	7698	03/12/91	1335	-	30
7698	NEGRO	DIRECTOR	7839	01/05/91	3005	-	30

- 5. Consulta los empleados cuyo oficio sea empleado, clasificado por numero de empleado en ascendente y apellido en descendente.

```
SELECT * FROM EMPLEO
WHERE OFICIO='EMPLEADO'
ORDER BY EMP_NO, APELLIDO DESC;
```

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7369	SANCHEZ	EMPLEADO	7902	17/12/90	1040	-	20
7876	ALONSO	EMPLEADO	7788	23/09/91	1430	-	20
7900	JIMENO	EMPLEADO	7698	03/12/91	1335	-	30
7934	MUÑOZ	EMPLEADO	7782	23/01/92	1690	-	10

- 6. Descripción de la tabla ALUM0405

```
DESC ALUM0405;
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ALUM0405	DNI	Varchar2	10	-	-	-	-	-	-
	NOMBRE	Varchar2	15	-	-	-	-	-	-
	APELLIDOS	Varchar2	20	-	-	-	-	-	-
	FECHA_NAC	Date	7	-	-	-	✓	-	-
	DIRECCION	Varchar2	20	-	-	-	✓	-	-
	POBLACION	Varchar2	20	-	-	-	✓	-	-
	PROVINCIA	Varchar2	20	-	-	-	✓	-	-
	CURSO	Number	-	2	0	-	-	-	-
	NIVEL	Varchar2	10	-	-	-	-	-	-
	CLASE	Char	1	-	-	-	-	-	-
	FALTAS1	Number	-	2	0	-	✓	-	-
	FALTAS2	Number	-	2	0	-	✓	-	-
	FALTAS3	Number	-	2	0	-	✓	-	-

1 - 13

- 7. Sacar los datos de los alumnos que se apellidan Martín o que cursen 2º Curso.

```
SELECT * FROM ALUM0405
WHERE APELLIDOS LIKE 'MARTIN%' OR CURSO=2;
```

DNI	NOMBRE	APELLIDOS	FECHA_NAC	DIRECCION	POBLACION	PROVINCIA	CURSO	NIVEL	CLASE	FALTAS1	FALTAS2	FALTAS3
34448974N	CONCHA	ESTRUCH VIDAL	10/11/85	C/ MEDICO FELIX, 12	BERROCALEJO	CACERES	2	DAI	A	0	0	0
34448999N	DORA	MARTIN BREÑA	10/11/94	C/ MEDICO FELIX, 10	BERROCALEJO	CACERES	1	ESO	C	0	0	0

- 8. Sacar la descripción de la tabla Notas alumno.
DESC NOTAS;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
NOTAS	DNI	Varchar2	10	-	-	-	-	-	-
	COD	Number	-	2	0	-	-	-	-
	NOTA	Number	-	2	0	-	✓	-	-

1 - 3

- 9. Sacar todos los alumnos y sus notas medias de aquellos que tengan una nota media menor que 6 y clarificarlos con alias de la tabla NOTAS_ALUMNOS.

```
SELECT NOMBRE_ALUMNO "Nombre Alumnos", ((NOTA1+NOTA2+NOTA3)/3) "Nota Media"  
FROM NOTAS_ALUMNOS  
WHERE ((NOTA1+NOTA2+NOTA3)/3)<6;
```

- 10. Sacar los alumnos cuya segunda nota sea menor que 6 y su nota media mayor que 5.

```
SELECT NOMBRE_ALUMNO "Nombre Alumnos", ((NOTA1+NOTA2+NOTA3)/3) "Nota Media"  
FROM NOTAS_ALUMNOS  
WHERE ((NOTA1+NOTA2+NOTA3)/3)>5 AND NOTA2<6;
```

- 11. Sacar de la tabla EMPLE, aquellos empleados cuyo apellido empiece por J y termine por O.

```
SELECT * FROM EMPLE  
WHERE APELLIDO LIKE 'J%O';
```

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7900	JIMENO	EMPLEADO	7698	03/12/91	1335	-	30

- 12. Sacar los empleados que no cobran comisión y trabajan en el Departamento 10 o 20.

```
SELECT * FROM EMPLE  
WHERE COMISION IS NULL  
AND (DEPT_NO=10 OR DEPT_NO=20);
```

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7369	SANCHEZ	EMPLEADO	7902	17/12/90	1040	-	20
7566	JIMENEZ	DIRECTOR	7839	02/04/91	2900	-	20
7782	CEREZO	DIRECTOR	7839	09/06/91	2885	-	10
7788	GIL	ANALISTA	7566	09/11/91	3000	-	20
7839	REY	PRESIDENTE	-	17/11/91	4100	-	10
7876	ALONSO	EMPLEADO	7788	23/09/91	1430	-	20
7902	FERNANDEZ	ANALISTA	7566	03/12/91	3000	-	20
7934	MUÑOZ	EMPLEADO	7782	23/01/92	1690	-	10

- 13. Sacar los empleados que no son director y trabajan en el departamento 20.

```
SELECT * FROM EMPLE  
WHERE OFICIO!= 'DIRECTOR'  
AND DEPT NO=20;
```

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7369	SANCHEZ	EMPLEADO	7902	17/12/90	1040	-	20
7788	GIL	ANALISTA	7566	09/11/91	3000	-	20
7876	ALONSO	EMPLEADO	7788	23/09/91	1430	-	20
7902	FERNANDEZ	ANALISTA	7566	03/12/91	3000	-	20

- 14. Sacar los Vendedores cuya comisión es superior a 40.000€.

```
SELECT * FROM EMPLE
WHERE COMISION>40000;
```

no data found

La comisión mas alta es de 1020€: **SELECT * FROM EMPLE;**

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7369	SANCHEZ	EMPLEADO	7902	17/12/90	1040	-	20
7499	ARROYO	VENDEDOR	7698	20/02/90	1500	390	30
7521	SALA	VENDEDOR	7698	22/02/91	1625	650	30
7566	JIMENEZ	DIRECTOR	7839	02/04/91	2900	-	20
7654	MARTIN	VENDEDOR	7698	29/09/91	1600	1020	30
7698	NEGRO	DIRECTOR	7839	01/05/91	3005	-	30
7782	CEREZO	DIRECTOR	7839	09/06/91	2885	-	10
7788	GIL	ANALISTA	7566	09/11/91	3000	-	20
7839	REY	PRESIDENTE	-	17/11/91	4100	-	10
7844	TOVAR	VENDEDOR	7698	08/09/91	1350	0	30
7876	ALONSO	EMPLEADO	7788	23/09/91	1430	-	20
7900	JIMENO	EMPLEADO	7698	03/12/91	1335	-	30
7902	FERNANDEZ	ANALISTA	7566	03/12/91	3000	-	20
7934	MUÑOZ	EMPLEADO	7782	23/01/92	1690	-	10

- 15. De la Tabla EMPLE obtener la Fecha de Alta con el mismo oficio que Fernandez.

```
SELECT FECHA_ALT FROM EMPLE
WHERE OFICIO=(SELECT OFICIO FROM EMPLE WHERE APELLIDO='FERNANDEZ');
```

FECHA_ALT
09/11/91
03/12/91

- 16. Sacar los datos de los empleados con un salario menor que el salario de Gil o que tengan el mismo oficio que Negro.

```
SELECT * FROM EMPLE
WHERE SALARIO<(SELECT SALARIO FROM EMPLE WHERE APELLIDO='GIL')
OR OFICIO=(SELECT OFICIO FROM EMPLE WHERE APELLIDO='NEGRO');
```

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7369	SANCHEZ	EMPLEADO	7902	17/12/90	1040	-	20
7499	ARROYO	VENDEDOR	7698	20/02/90	1500	390	30
7521	SALA	VENDEDOR	7698	22/02/91	1625	650	30
7566	JIMENEZ	DIRECTOR	7839	02/04/91	2900	-	20
7654	MARTIN	VENDEDOR	7698	29/09/91	1600	1020	30
7698	NEGRO	DIRECTOR	7839	01/05/91	3005	-	30
7782	CEREZO	DIRECTOR	7839	09/06/91	2885	-	10
7844	TOVAR	VENDEDOR	7698	08/09/91	1350	0	30
7876	ALONSO	EMPLEADO	7788	23/09/91	1430	-	20
7900	JIMENO	EMPLEADO	7698	03/12/91	1335	-	30
7934	MUÑOZ	EMPLEADO	7782	23/01/92	1690	-	10

- 17. De la tabla empleados sacar el apellido de los empleados del Departamento 20 o 30 cuyo oficio sea vendedor.

```
SELECT APELLIDO FROM EMPLE
WHERE (DEPT_NO=20 OR DEPT_NO=30) AND OFICIO='VENDEDOR';
```

APELLIDO
ARROYO
SALA
MARTIN
TOVAR

- 18. De la tabla empleados sacar el oficio y Apellido de los empleados que trabajen en el Departamento 40 o ganen menos de 2000€.

```
SELECT OFICIO, APELLIDO FROM EMPLE
WHERE DEPT_NO=40 OR SALARIO<2000;
```

OFICIO	APELLIDO
EMPLEADO	SANCHEZ
VENDEDOR	ARROYO
VENDEDOR	SALA
VENDEDOR	MARTIN
VENDEDOR	TOVAR
EMPLEADO	ALONSO
EMPLEADO	JIMENO
EMPLEADO	MUÑOZ

- 19. De la tabla empleados sacar el oficio y Apellido de los empleados que trabajen en el Departamento 40 y ganen menos de 2000€.

```
SELECT OFICIO, APELLIDO FROM EMPLE
WHERE DEPT_NO=40 AND SALARIO<2000;
```

```
SQL> SELECT OFICIO, APELLIDO FROM EMPLE WHERE DEPT_NO=40 AND SALARIO<2000 ;
ninguna fila seleccionada
```

Ejercicios SQL del Libro (Capítulo 3 pagina 125)

- ◆ Visualiza los nombres de los alumnos que tengan una nota entre 7 y 8 en la asignatura de “FOL”.

```
SELECT APENOM FROM ALUMNOS, NOTAS, ASIGNATURAS
WHERE ALUMNOS.DNI=NOTAS.DNI
AND NOTAS.COD=ASIGNATURAS.COD
AND NOTA BETWEEN 7 AND 8
AND NOMBRE='FOL';
```

APENOM
Díaz Fernández, María

- ◆ Visualiza los nombres de asignaturas que no tengan suspensos.

```
SELECT NOMBRE FROM ASIGNATURAS
WHERE COD IN (SELECT COD FROM NOTAS WHERE NOTA>=5)
AND COD !=(SELECT COD FROM NOTAS WHERE NOTA<5);
```

NOMBRE
Prog. Leng. Estr.
Sist. Informáticos
Análisis
FOL
RET
Aplic. Entornos 4ºGen

Ejercicios SQL del Libro (Capítulo 3 pagina 127)

Tablas EMPLE y DEPART

- ◆ 1. Selecciona el apellido, el oficio y la localidad de los departamentos de aquellos empleados cuyo oficio sea "ANALISTA".

Es una consulta de la unión de 2 tablas con una condición

```
SELECT * FROM (SELECT APELLIDO, OFICIO, LOC  
FROM EMPLE, DEPART  
WHERE EMPLE.DEPT_NO=DEPART.DEPT_NO) WHERE OFICIO='ANALISTA';
```

APELLIDO	OFICIO	LOC
FERNANDEZ	ANALISTA	MADRID
GIL	ANALISTA	MADRID

- ◆ 2. Obtén los datos de los empleados cuyo director (columna DIR de la tabla EMPLE) sea "CEREZO". Primero se debe consultar el código de EMP_NO de Cerezo, que es código de Director para otros empleados, para realizar la consulta propuesta.

```
SELECT * FROM EMPLE  
WHERE DIR=(SELECT EMP_NO FROM EMPLE WHERE APELLIDO='CEREZO');
```

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7934	MUÑOZ	EMPLEADO	7782	23/01/92	1690	-	10

- ◆ 3. Obtén los datos de los empleados del departamento de "VENTAS".

```
SELECT DEPT_NO FROM DEPART  
WHERE DNOMBRE='VENTAS';
```

DEPT_NO
30

Primero hay que hacer una consulta para saber el numero de departamento de VENTAS y en base a esa consulta hacer otra para obtener los datos requeridos.

```
SELECT * FROM EMPLE  
WHERE DEPT_NO=(SELECT DEPT_NO FROM DEPART WHERE DNOMBRE='VENTAS');
```

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7499	ARROYO	VENDEDOR	7698	20/02/90	1500	390	30
7521	SALA	VENDEDOR	7698	22/02/91	1625	650	30
7654	MARTIN	VENDEDOR	7698	29/09/91	1600	1020	30
7698	NEGRO	DIRECTOR	7839	01/05/91	3005	-	30
7844	TOVAR	VENDEDOR	7698	08/09/91	1350	0	30
7900	JIMENO	EMPLEADO	7698	03/12/91	1335	-	30

- ◆ 4. Obtén los datos de los departamentos que NO tengan empleados.

**SELECT * FROM DEPART
WHERE DEPT_NO NOT IN (SELECT DISTINCT DEPT_NO FROM EMPLE);**

DEPT_NO	DNOMBRE	LOC
40	PRODUCCION	BILBAO

- ◆ 5. Obtén los datos de los departamentos que tengan empleados.

**SELECT * FROM DEPART
WHERE DEPT_NO IN (SELECT DISTINCT DEPT_NO FROM EMPLE);**

DEPT_NO	DNOMBRE	LOC
20	INVESTIGACION	MADRID
30	VENTAS	BARCELONA
10	CONTABILIDAD	SEVILLA

- ◆ 6. Obtén el apellido y el salario de los empleados que superan todos los salarios de los empleados del departamento 20 (superen el salario máximo del Departamento 20).

**SELECT APELLIDO, SALARIO FROM EMPLE
WHERE SALARIO>(SELECT MAX (SALARIO) FROM EMPLE WHERE DEPT_NO=20);**

Tenemos que sacar el valor máximo de salario de los empleados del departamento 20 con MAX y sobre este resultado hacer la selección que se pide.

APELLIDO	SALARIO
NEGRO	3005
REY	4100

Tabla LIBRERIA

- ◆ 7. Visualiza el tema, estante y ejemplares de las filas de librería con ejemplares comprendidos entre 8 y 15.

**SELECT TEMA, ESTANTE, EJEMPLARES FROM LIBRERIA
WHERE EJEMPLARES BETWEEN 8 AND 15;**

También se puede hacer de la siguiente manera:

**SELECT TEMA, ESTANTE, EJEMPLARES FROM LIBRERIA
WHERE EJEMPLARES>=8 AND EJEMPLARES<=15;**

TEMA	ESTANTE	EJEMPLARES
INFORMATICA	A	15
ECONOMIA	A	10
DEPORTES	B	8
DIBUJO	C	10
BIOLOGIA	A	11
SOCIEDAD	D	9

- ◆ 8. Visualiza las columnas TEMA, ESTANTE y EJEMPLARES de las filas cuyo ESTANTE no este comprendido entre la “B” y la “D”.

**SELECT TEMA, ESTANTE, EJEMPLARES FROM LIBRERIA
WHERE ESTANTE NOT BETWEEN 'B' AND 'D';**

TEMA	ESTANTE	EJEMPLARES
INFORMATICA	A	15
ECONOMIA	A	10
BIOLOGIA	A	11
JARDINERIA	E	6

- ◆ 9. Visualiza con una sola orden SELECT todos los temas de LIBRERIA cuyo numero de ejemplares sea inferior a los que hay en "MEDICINA".

**SELECT TEMA FROM LIBRERIA
WHERE EJEMPLARES<(SELECT EJEMPLARES FROM LIBRERIA WHERE
TEMA='MEDICINA');**

TEMA
INFORMATICA
ECONOMIA
DEPORTES
FILOSOFIA
DIBUJO
BIOLOGIA
GEOLOGIA
SOCIEDAD
JARDINERIA

También valdría con la siguiente sentencia, siempre y cuando no haya mas nombres que empiecen con medicina pero tengan otro nombre detrás como MEDICINA NUCLEAR.

**SELECT TEMA FROM LIBRERIA
WHERE EJEMPLARES<(SELECT EJEMPLARES FROM LIBRERIA WHERE TEMA LIKE
'MEDICINA%');**

- ◆ 10. VISUALIZA los temas de LIBRERIA cuyo numero de ejemplares no este entre 15 y 20, ambos inclusive.

**SELECT TEMA FROM LIBRERIA
WHERE EJEMPLARES NOT BETWEEN 15 AND 20;**

TEMA
ECONOMIA
DEPORTES
FILOSOFIA
DIBUJO
BIOLOGIA
GEOLOGIA
SOCIEDAD
JARDINERIA

Tablas ALUMNOS, ASIGNATURAS y NOTAS

- ◆ 11. Visualiza todas las asignaturas que contengan tres letras “o” en su interior y tengan alumnos matriculados en “Madrid”.

```
SELECT NOMBRE FROM ASIGNATURAS
WHERE NOMBRE LIKE '%oo%oo%oo%'
AND COD IN (SELECT COD FROM ALUMNOS, NOTAS WHERE
ALUMNOS.DNI=NOTAS.DNI AND POBLA='Madrid');
```

NOMBRE
Entornos Gráficos

- ◆ 12. Visualiza los nombres de alumnos de “Madrid” que tengan alguna asignatura suspendida.

```
SELECT APENOM FROM ALUMNOS
WHERE POBLA='Madrid'
AND DNI=(SELECT DNI FROM NOTAS WHERE NOTA<5);
```

APENOM
Cerrato Vela, Luis

- ◆ 13. Muestra los nombres de alumnos que tengan la misma nota que tiene “Díaz Fernández, María” en FOL en alguna asignatura.

```
SELECT DISTINCT APENOM FROM ALUMNOS, NOTAS
WHERE ALUMNOS.DNI=NOTAS.DNI
AND NOTA=(SELECT NOTA FROM ASIGNATURAS, NOTAS, ALUMNOS WHERE
ASIGNATURAS.COD=NOTAS.COD AND NOTAS.DNI=ALUMNOS.DNI AND APENOM='Díaz
Fernández, María' AND NOMBRE='FOL');
```

APENOM
Díaz Fernández, María
Cerrato Vela, Luis

Ahora discriminando el nombre con el que comparamos.

```
SELECT DISTINCT APENOM FROM ALUMNOS, NOTAS
WHERE ALUMNOS.DNI=NOTAS.DNI
AND NOTA=(SELECT NOTA FROM ASIGNATURAS, NOTAS, ALUMNOS WHERE
ASIGNATURAS.COD=NOTAS.COD AND NOTAS.DNI=ALUMNOS.DNI AND APENOM='Díaz
Fernández, María' AND NOMBRE='FOL')
AND APENOM !='Díaz Fernández, María';
```

APENOM
Cerrato Vela, Luis

- ◆ 14. Obtén los datos de las asignaturas que no tengan alumnos.

```
SELECT * FROM ASIGNATURAS WHERE COD NOT IN (SELECT COD FROM NOTAS);
```

COD	NOMBRE
8	Idioma

- ◆ 15. Obtén el nombre y apellido de los alumnos que tengan nota en la asignatura con código 1.

```
SELECT APENOM FROM NOTAS, ALUMNOS  
WHERE NOTAS.DNI=ALUMNOS.DNI  
AND COD=1;
```

APENOM
Alcalde García, Elena

- ◆ 16. Obtén el nombre y apellido de los alumnos que no tengan nota en la asignatura con código 1.

```
SELECT DISTINCT APENOM FROM ALUMNOS, NOTAS  
WHERE ALUMNOS.DNI=NOTAS.DNI  
AND COD!=1  
AND APENOM !=(SELECT APENOM FROM NOTAS, ALUMNOS WHERE  
NOTAS.DNI=ALUMNOS.DNI AND COD=1);
```

APENOM
Díaz Fernández, María
Cerrato Vela, Luis

Tablas utilizadas en los ejercicios**EMPLE**

EMP_NO Number(4)	APELLIDO Varchar2(10)	OFICIO Varchar2(10)	DIR Number(4)	FECHA_ALT Date	SALARIO Number (10)	COMISION Number(10)	DEPT_NO Number(2)
7369	SANCHEZ	EMPLEADO	7902	17/12/1990	1040		20
7499	ARROYO	VENDEDOR	7698	20/02/1990	1500	390	30
7521	SALA	VENDEDOR	7698	22/02/1991	1625	650	30
7566	JIMENEZ	DIRECTOR	7839	02/04/1991	2900		20
7654	MARTIN	VENDEDOR	7698	29/09/1991	1600	1020	30
7698	NEGRO	DIRECTOR	7839	01/05/1991	3005		30
7782	CEREZO	DIRECTOR	7839	09/06/1991	2885		10
7788	GIL	ANALISTA	7566	09/11/1991	3000		20
7839	REY	PRESIDENTE		17/11/1991	4100		10
7844	TOVAR	VENDEDOR	7698	08/09/1991	1350	0	30
7876	ALONSO	EMPLEADO	7788	23/09/1991	1430		20
7900	JIMENO	EMPLEADO	7698	03/12/1991	1335		30
7902	FERNANDEZ	ANALISTA	7566	03/12/1991	3000		20
7934	MUÑOZ	EMPLEADO	7782	23/01/1992	1690		10

DEPART

DEPT_NO Number(2)	DNOMBRE Varchar2(14)	LOC Varchar2(14)
10	CONTABILIDAD	SEVILLA
20	INVESTIGACION	MADRID
30	VENTAS	BARCELONA
40	PRODUCCION	BILBAO

NOMBRES**NOTAS_ALUMNOS**

NOMBRE Varchar2(15)	EDAD Number(2)
PEDRO	17
JUAN	17
MARÍA	16
CLARA	14
	15
	18

NOMBRE_ALUMNO Varchar2(25)	NOTA1 Number(2)	NOTA2 Number(2)	NOTA3 Number(2)
Alcalde García, M. Luisa	5	5	5
Benito Martín, Luis	7	6	8
Casas Martínez, Manuel	7	5	5
Corregidor Sánchez, Ana	6	9	8
Díaz Sánchez, María			7

MISTEXTOS

TITULO Varchar2(32)	AUTOR Varchar2(22)	EDITORIAL Varchar2(15)	PAGINA Number(3)
METODOLOGÍA DE LA PROGRAMACIÓN.	ALCALDE, GARCÍA	MCGRAWHILL	140
"INFORMÁTICA BÁSICA."	GARCÍA GARCERAN	PARANINFO	130
SISTEMAS OPERATIVOS	J.F. GARCÍA	OBSBORNE	300
SISTEMAS DIGITALES.	M.A. RUIZ	PRENTICE HALL	190
"MANUAL DE C."	M.A. RUIZ	MCGRAWHILL	340

LIBRERÍA

TEMA Char(15)	ESTANTE Char(1)	EJEMPLARES Number(2)
Informática	A	15
Economía	A	10
Deportes	B	8
Filosofía	C	7
Dibujo	C	10
Medicina	C	16
Biología	A	11
Geología	D	7
Sociedad	D	9
Labores	B	20
Jardinería	E	6

LIBROS

TITULO Varchar2(32)	AUTOR Varchar2(22)	EDITORIAL Varchar2(15)	PAGINA Number(3)
LA COLMENA	CELA, CAMILO JOSÉ	PLANETA	240
LA HISTORIA DE MI HIJO	GORDIMER, NADINE	TIEM.MODERNOS	327
LA MIRADA DEL OTRO	G. DELGADO, FERNANDO	PLANETA	298
ULTIMAS TARDES CON TERESA	MARSÉ, JUAN	CIRCULO	350
LA NOVELA DE P. ANSUREZ	TORRENTE B., GONZALO	PLANETA	162

NACIMIENTOS

NOMBRE Char(15)	APELLIDO Char(15)	FECHANAC Date	EDAD Number
PEDRO	SÁNCHEZ	12/05/1982	17
JUAN	JIMÉNEZ	23/08/1982	17
MARÍA	LÓPEZ	02/02/1983	16
CLARA	LASECA	20/05/1985	14

Ejercicios Adicionales (Unidad 4).

1.- ¿Cuál sería la salida de ejecutar estas funciones?

$\text{ABS}(146) =$	$\text{ABS}(-30) =$	$\text{POWER}(3,-1) =$	$\text{ROUND}(33.67) =$
$\text{CEIL}(2) =$	$\text{CEIL}(1.3) =$	$\text{ROUND}(-33.67,2) =$	$\text{ROUND}(-33.67,-2) =$
$\text{CEIL}(-2.3) =$	$\text{CEIL}(-2) =$	$\text{ROUND}(-33.27,1) =$	$\text{ROUND}(-33.27,-1) =$
$\text{FLOOR}(-2) =$	$\text{FLOOR}(-2.3) =$	$\text{TRUNC}(67.232) =$	$\text{TRUNC}(67.232,-2) =$
$\text{FLOOR}(2) =$	$\text{FLOOR}(1.3) =$	$\text{TRUNC}(67.232,2) =$	$\text{TRUNC}(67.58,-1) =$
$\text{MOD}(22,23) =$	$\text{MOD}(10,3) =$	$\text{TRUNC}(67.58,1) =$	
$\text{POWER}(10,0) =$	$\text{POWER}(3,2) =$		

SELECT ABS(146) FROM DUAL;

ABS(146)
146

SELECT CEIL(2) FROM DUAL;

CEIL(2)
2

SELECT CEIL(-2.3) FROM DUAL;

CEIL(-2.3)
-2

SELECT FLOOR(-2) FROM DUAL;

FLOOR(-2)
-2

SELECT FLOOR(2) FROM DUAL;

FLOOR(2)
2

SELECT MOD(22,23) FROM DUAL;

MOD(22,23)
22

SELECT POWER(10,0) FROM DUAL;

POWER(10,0)
1

SELECT ABS(-30) FROM DUAL;

ABS(-30)
30

SELECT CEIL(1.3) FROM DUAL;

CEIL(1.3)
2

SELECT CEIL(-2) FROM DUAL;

CEIL(-2)
-2

SELECT FLOOR(-2.3) FROM DUAL;

FLOOR(-2.3)
-3

SELECT FLOOR(1.3) FROM DUAL;

FLOOR(1.3)
1

SELECT MOD(10,3) FROM DUAL;

MOD(10,3)
1

SELECT POWER(3,2) FROM DUAL;

POWER(3,2)
9

SELECT POWER(3,-1) FROM DUAL

POWER(3,-1)

SELECT ROUND(-33.67,2) FROM DUAL;

ROUND(-33.67,2)

```
SELECT ROUND(-33.27,1) FROM DUAL;
```

ROUND(-33.27,1)

```
SELECT TRUNC(67.232) FROM DUAL;
```

TRUNC(67.232)

```
SELECT TRUNC(67.232,2) FROM DUAL;
```

TRUNC(67.232,2)
67,23

```
SELECT TRUNC(67.58,1) FROM DUAL;
```

TRUNC(67.58,1)

```
SELECT ROUND(33.67) FROM DUAL;
```

ROUND(33.67)

```
SELECT ROUND(-33.67,-2) FROM DUAL;
```

ROUND(-33.67,-2)
0

```
SELECT ROUND(-33.27,-1) FROM DUAL;
```

-30

SELECT TRUNC(67.232,-2) FROM DUAL;

TRUNC(67.232,-2)
0

SELECT TRUNC(67.58,-1) FROM DUAL;

TRUNC(67.58,-1)
60

2.- a) A partir de la tabla EMPLE, visualizar cuántos apellidos de los empleados empiezan por la letra 'A'

**SELECT COUNT (APELIDO)
FROM EMPLE
WHERE APELLIDO LIKE 'A%';**

COUNT(APELLIDO)
2

Otra forma de hacerlo

**SELECT COUNT (APELIDO)
FROM EMPLE
WHERE SUBSTR (APELIDO,1,1) = 'A%';**

b) Obtén el apellido o apellidos de empleados que empiecen por la letra 'A' y que tengan máximo salario (de los que empiezan por la letra 'A').

**SELECT APELLIDO
FROM EMPLE
WHERE APELLIDO LIKE 'A%'
AND SALARIO=(SELECT MAX (SALARIO) FROM EMPLE WHERE APELLIDO LIKE 'A%');**

APELLIDO
ARROYO

3.- Contar las filas de LIBRERÍA cuyo tema tenga, por lo menos, una 'a'.

**SELECT COUNT (TEMA)
FROM LIBRERIA
WHERE TEMA LIKE '%a%' OR TEMA LIKE '%a%';**

COUNT(TEMA)
9

Otra forma de hacerlo:

**SELECT COUNT (TEMA)
FROM LIBRERIA
WHERE UPPER (TEMA) LIKE '%A%';**

4.- Visualizar el número de estantes distintos que hay en la tabla LIBRERÍA de aquellos temas que contienen, al menos, una 'e'.

```
SELECT COUNT (DISTINCT ESTANTE) "Distintos Estantes"
FROM LIBRERIA
WHERE TEMA LIKE '%E%';
```

Distintos Estantes
5

5.- Visualizar el número de estantes diferentes que hay en la tabla LIBRERÍA.

```
SELECT COUNT (DISTINCT ESTANTE) "Numero Estantes"
FROM LIBRERIA;
```

Numero Estantes
5

6.- Obtener en una columna el apellido y el oficio de cada uno de los empleados de la tabla EMPLE, de la siguiente manera: APELLIDO es OFICIO. Por ejemplo, 'SANCHEZ es EMPLEADO'.

```
SELECT CONCAT (APELLIDO || ' es ', OFICIO) "Puestos Empleados"
FROM EMPLE;
```

Puestos Empleados
SANCHEZ es EMPLEADO
ARROYO es VENDEDOR
SALA es VENDEDOR
JIMENEZ es DIRECTOR
MARTIN es VENDEDOR
NEGRO es DIRECTOR
CEREZO es DIRECTOR
GIL es ANALISTA
REY es PRESIDENTE
TOVAR es VENDEDOR
ALONSO es EMPLEADO
JIMENO es EMPLEADO
FERNANDEZ es ANALISTA
MUÑOZ es EMPLEADO

7.- Obtener en una columna el apellido y el oficio de cada uno de los empleados de la tabla EMPLE, de la siguiente manera: Apellido es Oficio. Por ejemplo, 'Sanchez es Empleado'.

Todas en minúsculas:

```
SELECT CONCAT (LOWER (APELLIDO) || ' es ', LOWER (OFICIO)) "Puestos Empleados"
FROM EMPLE;
```

La primera en mayúscula y el resto minúsculas:

```
SELECT CONCAT (INITCAP (APELLIDO) || ' es ', INITCAP (OFICIO)) "Puestos Empleados"
FROM EMPLE;
```

Puestos Empleados
sanchez es empleado
arroyo es vendedor
sala es vendedor
jimenez es director
martin es vendedor
negro es director
cerezo es director
gil es analista
rey es presidente
tovar es vendedor
alonso es empleado
jimeno es empleado
fernandez es analista
muñoz es empleado

Puestos Empleados
Sanchez es Empleado
Arroyo es Vendedor
Sala es Vendedor
Jimenez es Director
Martin es Vendedor
Negro es Director
Cerezo es Director
Gil es Analista
Rey es Presidente
Tovar es Vendedor
Alonso es Empleado
Jimeno es Empleado
Fernandez es Analista
Muñoz es Empleado

8.- Obtener en una columna el apellido y el oficio de cada uno de los empleados de la tabla EMPLE, de la siguiente manera: APELLIDO es OFICIO alineado todo a la derecha.

Por ejemplo:

Sanchez	es	Empleado
Arroyo	es	Vendedor
Sala	es	Vendedor

**SELECT LPAD ((CONCAT (INITCAP (APELLIDO) || ' es ', INITCAP (OFICIO))), 25 , ') "Apellidos Empleados"
FROM EMPLE;**

```
SQL> SELECT LPAD ((CONCAT (INITCAP (APELLIDO) || ' es ', INITCAP (OFICIO))), 25  
, ' ) "Apellidos Empleados" FROM EMPLE;  
  
Apellidos Empleados  
-----  
 Sanchez es Empleado  
 Arroyo es Vendedor  
 Sala es Vendedor  
 Jimenez es Director  
 Martin es Vendedor  
 Negro es Director  
 Cerezo es Director  
 Gil es Analista  
 Rey es Presidente  
 Tovar es Vendedor  
 Alonso es Empleado  
  
Apellidos Empleados  
-----  
 Jimeno es Empleado  
 Fernandez es Analista  
 Muñoz es Empleado
```

9.- Utilizar la función LPAD para obtener las siguientes salidas.

Ejem1	Ejem2	Ejem3	Ejem4
****X	*.*.X	*.*.XHOLA

SELECT LPAD ('X', 5, '*') "Ejem1" FROM DUAL;

Ejem1
****X

SELECT LPAD ('X', 6, '*.') "Ejem1" FROM DUAL;

Ejem1
..X

SELECT LPAD ('X', 5, '*.') "Ejem1" FROM DUAL;

Ejem1
**.X

SELECT LPAD ('HOLA', 9, '.') "Ejem1" FROM DUAL;

Ejem1
.....HOLA

10.- Mostrar el apellido y primera letra del apellido de la tabla empleados.

**SELECT APELLIDO "Apellido", SUBSTR (APELLIDO, 1, 1) "Inicial Apellido"
FROM EMPLE;**

Apellido	Inicial Apellido
SANCHEZ	S
ARROYO	A
SALA	S
JIMENEZ	J
MARTIN	M
NEGRO	N
CEREZO	C
GIL	G
REY	R
TOVAR	T
ALONSO	A
JIMENO	J
FERNANDEZ	F
MUÑOZ	M

11.- Apellido y primera letra del apellido seguido de ocho asteriscos.

**SELECT APELLIDO "Apellido", RPAD ((SUBSTR (APELLIDO, 1, 1)),8,'*') "Inicial Apellido"
FROM EMPLE;**

Apellido	Inicial Apellido
SANCHEZ	S*****
ARROYO	A*****
SALA	S*****
JIMENEZ	J*****
MARTIN	M*****
NEGRO	N*****
CEREZO	C*****
GIL	G*****
REY	R*****
TOVAR	T*****
ALONSO	A*****
JIMENO	J*****
FERNANDEZ	F*****
MUÑOZ	M*****

12.- Mostrar apellido de todos los empleados sustituyendo EZ por O.

SELECT REPLACE (APELIDO, 'EZ' , 'O') "Apellido empleados"

FROM EMPLE;

Apellido Empleados
SANCHO
ARROYO
SALA
JIMENO
MARTIN
NEGRO
CEROO
GIL
REY
TOVAR
ALONSO
JIMENO
FERNANDO
MUÑOZ

13.- Mostrar apellido con la primera letra en mayúscula.

SELECT INITCAP (APELIDO) "Apellido empleados"

FROM EMPLE;

Apellido Empleados
Sanchez
Arroyo
Sala
Jimenez
Martin
Negro
Cerezo
Gil
Rey
Tovar
Alonso
Jimeno
Fernandez
Muñoz

14.- De la tabla Nacimientos, mostrar el nombre y apellido con el siguiente formato: Apellido, Nombre. Las primeras letras han de ir en mayúsculas.

SELECT INITCAP (APELIDO) "Apellidos" ,INITCAP (NOMBRE) "Nombre"

FROM NACIMIENTOS;

Apellidos	Nombre
Sánchez	Pedro
Jiménez	Juan
López	María
Laseca	Clara

15.- De la tabla Nacimientos, mostrar el nombre y apellido con el siguiente formato: NOMBRE inicial APELLIDO punto, seguido de la fecha de nacimiento y sustituyendo en ésta las barras por guiones. Es decir PEDRO S., 12-05-1982

SELECT UPPER (NOMBRE) "Nombre", RPAD ((SUBSTR (APELIDO,1,1)),3,'.') "Apellido", REPLACE (FECHANAC, '/', '-') "Fecha Nacimiento" FROM NACIMIENTOS;

Nombre	Apellido	Fecha Nacimiento
PEDRO	S.,	12-05-82
JUAN	J.,	23-08-82
MARÍA	L.,	02-02-83
CLARA	L.,	20-05-85

16.- Buscar el empleado con el apellido más largo.

**SELECT APELLIDO "Apellido"
FROM EMPLE
WHERE LENGTH (APELIDO)=(SELECT MAX (LENGTH (APELIDO)) FROM EMPLE);**

Apellido
FERNANDEZ

17.- ¿Cuál es el resultado de éstas sentencias SELECT?

**SELECT TRANSLATE ('OGRO', 'O', 'AS'); →
SELECT TRANSLATE ('OGRO', 'O', 'AS') FROM DUAL;**

TRANSLATE('OGRO','O','AS')
AGRA

**SELECT REPLACE ('OGRO', 'O', 'AS'); →
SELECT REPLACE ('OGRO', 'O', 'AS') FROM DUAL;**

REPLACE('OGRO','O','AS')
ASGRAS

**SELECT TRANSLATE ('OGRON', 'ON', 'AS'); →
SELECT TRANSLATE ('OGRON', 'ON', 'AS') FROM DUAL;**

TRANSLATE('OGRON','ON','AS')
AGRAS

**SELECT REPLACE ('OGRON', 'ON', 'AS'); →
SELECT REPLACE ('OGRON', 'ON', 'AS') FROM DUAL;**

REPLACE('OGRON','ON','AS')
OGRAS

18.- ¿Cuál es el resultado de éstas sentencias SELECT?

**SELECT INSTR ('abracadabra', 'bra', 2, 2); →
SELECT INSTR ('abracadabra', 'bra', 2, 2) FROM DUAL;**

INSTR('ABRACADABRA','BRA',2,2)
9

**SELECT INSTR ('abracadaBRA', 'bra', 4, 2); →
SELECT INSTR ('abracadaBRA', 'bra', 4, 2) FROM DUAL;**

INSTR('ABRACADABRA','BRA',4,2)
0

**SELECT INSTR ('abracadabra', 'BRA', 2, 2); →
SELECT INSTR ('abracadabra', 'BRA', 2, 2) FROM DUAL;**

INSTR('ABRACADABRA','BRA',2,2)
0

19.- ¿Cuál es el resultado de éstas sentencias SELECT?

**SELECT INSTR ('II VUELTA CICLISTA A TALAVERA', 'TA', 3, 2); →
SELECT INSTR ('II VUELTA CICLISTA A TALAVERA', 'TA', 3, 2) FROM DUAL;**

INSTR('IIVUELTACICLISTATALAVERA','TA',3,2)
17

**SELECT INSTR ('II VUELTA CICLISTA A TALAVERA', 'A', -1); →
SELECT INSTR ('II VUELTA CICLISTA A TALAVERA', 'A', -1) FROM DUAL;**

INSTR('IIVUELTACICLISTATALAVERA','A',-1)
29

**SELECT INSTR ('II VUELTA CICLISTA A TALAVERA', 'A', -3); →
SELECT INSTR ('II VUELTA CICLISTA A TALAVERA', 'A', -3) FROM DUAL;**

INSTR('IIVUELTACICLISTATALAVERA','A',-3)
25

20.- Encontrar la primera ocurrencia de la letra 'A' en la columna AUTOR de la tabla MISTEXTOS

**SELECT AUTOR, INSTR (AUTOR, 'A') "Posición 1º A"
FROM MISTEXTOS;**

AUTOR	Posición 1º A
ALCALDE GARCÍA	1
GARCÍA GARCERÁN	2
GARCÍA STRUCH	2
RUÍZ LOPEZ	0
RUÍZ LOPEZ	0

21.- Encontrar el número de caracteres de las columnas TITULO y AUTOR para todas las filas de la tabla MISTEXTOS.

**SELECT TITULO, LENGTH (TITULO) "Longitud Titulo", AUTOR, LENGTH (AUTOR) "Longitud Autor"
FROM MISTEXTOS;**

TITULO	Longitud Titulo	AUTOR	Longitud Autor
METODOLOGÍA DE LA PROGRAMACIÓN.	31	ALCALDE GARCÍA	14
"INFORMÁTICA BÁSICA."	21	GARCÍA GARCERÁN	15
SISTEMAS OPERATIVOS	19	GARCÍA STRUCH	13
SISTEMAS DIGITALES.	19	RUÍZ LOPEZ	10
"MANUAL DE C."	14	RUÍZ LOPEZ	10

22.- Calcular el número de caracteres de la columnas TEMA para todas las filas de la tabla LIBRERIA.

SELECT TEMA, INSTR(TEMA, ' ') - 1 FROM LIBRERIA;

SELECT TEMA, LENGTH (RTRIM (TEMA)) FROM LIBRERIA;

TEMA	LENGTH(RTRIM(TEMA))
INFORMATICA	11
ECONOMIA	8
DEPORTES	8
FILOSOFIA	9
DIBUJO	6
MEDICINA	8
BIOLOGIA	8
GEOLOGIA	8
SOCIEDAD	8
LABORES	7
JARDINERIA	10

23.- Calcular el número de días que tiene febrero del año que viene.

SELECT TO_CHAR (LAST_DAY('12/02/2013'), 'dd') FROM DUAL ;

TO_CHAR(LAST_DAY('12/02/2013'),'DD')
28

24.- Calcular la edad de cada uno utilizando la función MONTHS_BETWEEN de la tabla NACIMIENTOS.

SELECT TRUNC (MONTHS_BETWEEN (SYSDATE, FECHANAC)/12) "Edad" FROM NACIMIENTOS;

Edad
29
29
28
26

25.- A partir de la tabla EMPLE, obtener la fecha de alta formateada, de manera que aparezca el nombre del mes con todas sus letras en minúscula, el número de día del mes y el año. Por ejemplo, diciembre 17, 1990

SELECT FECHA_ALT, TO_CHAR (FECHA_ALT, 'month dd yyyy') "Fecha Formateada" FROM EMPLE;

FECHA_ALT	Fecha Formateada
17/12/90	diciembre 17 1990
20/02/90	febrero 20 1990
22/02/91	febrero 22 1991

26.- A partir de la tabla EMPLE, obtener la fecha de alta formateada, de manera que aparezca el nombre del mes con la primera letra en MAYÚSCULA, el número de día del mes y el año.

SELECT TO_CHAR (FECHA_ALT, 'Month "," dd yyyy') "Fecha Formateada" FROM EMPLE;

Fecha Formateada
Diciembre , 17 1990
Febrero , 20 1990
Febrero , 22 1991
Abril , 02 1991
Septiembre , 29 1991
Mayo , 01 1991
Junio , 09 1991
Noviembre , 09 1991
Noviembre , 17 1991
Septiembre , 08 1991
More than 10 rows available. Increase rows selector to view more rows.

27.- A partir de la tabla EMPLE, obtener la fecha de alta formateada, de manera que aparezca el nombre del mes con todas sus letras en MAYÚSCULA, el número de día del mes y el año.

SELECT TO_CHAR (FECHA_ALT, 'MONTH "," dd yyyy') "Fecha Formateada" FROM EMPLE;

Fecha Formateada
DICIEMBRE , 17 1990
FEBRERO , 20 1990
FEBRERO , 22 1991
ABRIL , 02 1991
SEPTIEMBRE , 29 1991
MAYO , 01 1991
JUNIO , 09 1991
NOVIEMBRE , 09 1991
NOVIEMBRE , 17 1991
SEPTIEMBRE , 08 1991
More than 10 rows available. Increase rows selector to view more rows.

28.- A partir de la tabla EMPLE, obtener la fecha de alta formateada, de manera que aparezca el nombre del mes con tres letras, el número de día del año y los tres últimos dígitos del año. Ejemplo, dic 352 990

SELECT TO_CHAR (FECHA_ALT, 'mon "," ddd yy') "Fecha Formateada" FROM EMPLE;

Fecha Formateada
dic , 351 990
feb , 051 990
feb , 053 991
abr , 092 991
sep , 272 991
may , 121 991
jun , 160 991
nov , 313 991
nov , 321 991
sep , 251 991
More than 10 rows available. Increase rows selector to view more rows.

29.- Obtener la fecha de hoy formateada de la siguiente manera "Hoy es martes, 1 de noviembre de 2010".
SELECT SYSDATE, TO_CHAR (SYSDATE, '"Hoy es " day dd " de " month " de " yyyy') "Fecha Formateada"
FROM DUAL;

SYSDATE	Fecha Formateada
19/01/12	Hoy es jueves 19 de enero de 2012

30.- Visualizar los temas con menor número de ejemplares de la tabla librería que contengan una "g".

SELECT TEMA
FROM LIBRERIA
WHERE UPPER (TEMA) LIKE '%G%'
AND EJEMPLARES = (SELECT MIN (EJEMPLARES) FROM LIBRERIA WHERE TEMA LIKE '%G%');
Con UPPER obligamos a poner todo en mayúsculas a la hora de hacer al comprobación con G

TEMA
GEOLOGIA

31.- Mirad si hay a algún alumno le sale nota media negativa en la tabla notas_alumnos

SELECT NOMBRE_ALUMNO
FROM NOTAS_ALUMNOS
WHERE (NOTA1+NOTA2+NOTA3)/3<0;

no data found

Ejercicios Del Libro (pagina 155)

1.- Dada la tabla EMPLE, obtén el sueldo medio, el numero de comisiones no nulas, el máximo sueldo y el mínimo sueldo de los empleados del departamento 30. Emplea el formato adecuado para la salida para las cantidades numéricas.

```
SELECT TO_CHAR (AVG (SALARIO), '99G999D99') "Salario medio", COUNT (COMISION) "Con NO nula",
TO_CHAR (MAX (SALARIO), '99G999D99') "Salario maximo", TO_CHAR (MIN (SALARIO), '99G999D99')
"Salario minimo" FROM EMPLE WHERE DEPT_NO=30;
```

Salario Medio	Con NO Nula	Salario Maximo	Salario Minimo
1.735,83	4	3.005,00	1.335,00

2.- Visualiza los temas con mayor numero de ejemplares de la tabla librería y que tengan al menos, una 'E' (pueden ser un tema o varios).

```
SELECT TEMA, EJEMPLARES
FROM LIBRERIA
WHERE TEMA LIKE '%E%'
AND EJEMPLARES=(SELECT MAX (EJEMPLARES) FROM LIBRERIA);
```

TEMA	EJEMPLARES
LABORES	20

3.- Dada la tabla MISTEXTOS ¿que sentencia SELECT se debe ejecutar para tener este resultado?

Resultado

METODOLOGIA DE LA PROGRAMACION-^-^-^-^-
INFORMATICA BASICA-^-^-^-^-^-^-^-^-^-
SISTEMAS OPERATIVOS-^-^-^-^-^-^-^-^-^-
SISTEMAS DIGITALES-^-^-^-^-^-^-^-^-^-
MANUAL DE C-^-^-^-^-^-^-^-^-^-^-^-^-^-^-
en total 40 caracteres.

```
SELECT RPAD (RTRIM (LTRIM (TITULO,'.'),'.'),40,'-') "Resultado" FROM MISTEXTOS;
```

Resultado
METODOLOGÍA DE LA PROGRAMACIÓN-^-^-^-^-
INFORMÁTICA BÁSICA-^-^-^-^-^-^-^-^-^-
SISTEMAS OPERATIVOS-^-^-^-^-^-^-^-^-^-
SISTEMAS DIGITALES-^-^-^-^-^-^-^-^-^-
MANUAL DE C-^-^-^-^-^-^-^-^-^-^-^-^-^-^-

4.- Visualiza los títulos de la tabla MISTEXTOS sin los caracteres punto y comillas, y en minúscula, de dos formas conocidas.

```
SELECT LOWER (RTRIM (LTRIM (TITULO,'.'),'.')) "Resultado" FROM MISTEXTOS;
```

Resultado
metodología de la programación
informática básica
sistemas operativos
sistemas digitales
manual de c

5.- Dada la tabla LIBROS, escribe la sentencia SELECT que visualice dos columnas, una con el AUTOR y otra con el apellido del autor.

Busco la coma del apellido y sabiendo la posición de la coma, se hasta donde llega el apellido y cojo esa parte de la cadena.

SELECT INSTR(AUTOR,(',')) "Apellido" FROM LIBROS; devuelve la posición de la ','

SELECT AUTOR, SUBSTR (AUTOR, 1, INSTR (AUTOR,',') -1) "Apellido" FROM LIBROS; Devuelve el contenido de autor hasta la posición indicada por la sentencia anterior y restando 1 para que no aparezca la ","
CELA, CAMILO JOSE queremos que de: CELA

AUTOR	Apellido
CELA, CAMILO JOSÉ	CELA
GORDIMER, NADINE	GORDIMER
G.DELGADO, FERNANDO	G.DELGADO
MARSE, JUAN	MARSE
TORRENTE B., GONZALO	TORRENTE B.

6.- Escribe la sentencia SELECT que visualice las columnas de AUTOR y otra columna con el nombre del autor (sin el apellido) de la tabla LIBROS.

SELECT AUTOR, SUBSTR (AUTOR, INSTR (AUTOR,',') +2) "Nombre" FROM LIBROS;

CELA, CAMILO JOSE queremos que de: CAMILO JOSE

AUTOR	Nombre
CELA, CAMILO JOSÉ	CAMILO JOSÉ
GORDIMER, NADINE	NADINE
G.DELGADO, FERNANDO	FERNANDO
MARSE, JUAN	JUAN
TORRENTE B., GONZALO	GONZALO

7.- A partir de la tabla LIBROS, realiza una sentencia SELECT que visualice en una columna, primero el nombre del autor y luego, su apellido.

SELECT SUBSTR (AUTOR, INSTR (AUTOR,',') +2) "Nombre", SUBSTR (AUTOR, 1, INSTR (AUTOR,',') -1) "Apellido" FROM LIBROS;

Nombre	Apellido
CAMILO JOSÉ	CELA
NADINE	GORDIMER
FERNANDO	G.DELGADO
JUAN	MARSE
GONZALO	TORRENTE B.

8.- A partir de la tabla LIBROS, realiza una sentencia SELECT para que aparezcan los títulos ordenados por su numero de caracteres.

SELECT TITULO, LENGTH (TITULO) "Longitud"

FROM LIBROS

ORDER BY LENGTH (TITULO);

TITULO	Longitud
LA COLMENA	10
LA MIRADA DEL OTRO	18
LA HISTORIA DE MI HIJO	22
LA NOVELA DE P. ANSUREZ	23
ÚLTIMAS TARDES CON TERESA	25

9.- Dada la tabla NACIMIENTOS, realiza una sentencia SELECT que obtenga la siguiente salida: NOMBRE, FECHANAC, FECHA_FORMATADA, donde FECHA_FORMATADA tiene el siguiente formato.

"Nació el 12 de mayo de 1982".

SELECT NOMBRE, FECHANAC, TO_CHAR (FECHANAC, '"Nacio el " dd " de " month " de " yyyy') "Fecha Formatada" FROM NACIMIENTOS;

NOMBRE	FECHANAC	Fecha Formatada
PEDRO	12/05/82	Nacio el 12 de mayo de 1982
JUAN	23/08/82	Nacio el 23 de agosto de 1982
MARÍA	02/02/83	Nacio el 02 de febrero de 1983
CLARA	20/05/85	Nacio el 20 de mayo de 1985

11.- A partir de la tabla NACIMIENTOS, visualiza en una columna el NOMBRE seguido de su fecha de nacimiento formateada (quita blancos del nombre).

SELECT CONCAT (RTRIM (NOMBRE), TO_CHAR (FECHANAC, '" Nacio el " dd " de " month " de " yyyy')) "Nombre y Fecha nacimiento" FROM NACIMIENTOS;

Nombre Y Fecha Nacimiento
PEDRO Nacio el 12 de mayo de 1982
JUAN Nacio el 23 de agosto de 1982
MARÍA Nacio el 02 de febrero de 1983
CLARA Nacio el 20 de mayo de 1985

12.- Convierte la cadena '010712' a fecha y visualiza su nombre de mes en mayúsculas.

SELECT TO_CHAR (TO_DATE ('010712', 'ddmmyy'), 'MONTH') FROM DUAL

TO_CHAR(TO_DATE('010712','DDMMYY'),'MONTH')
JULIO

13,- Visualiza aquellos temas de la tabla LIBRERIA cuyos ejemplares sean 7 con el nombre de tema de "SEVEN"; el resto de temas que no tengan 7 ejemplares se visualizaran como están.

SELECT TEMA, EJEMPLARES, DECODE (EJEMPLARES, 7, 'SEVEN', EJEMPLARES) "Código" FROM LIBRERIA ;

TEMA	EJEMPLARES	CODIGO
Informática	15	Informática
Economía	10	Economía
Deportes	8	Deportes
Filosofía	7	SEVEN
Dibujo	10	Dibujo
Medicina	16	Medicina
Biología	11	Biología
Geología	7	SEVEN
Sociedad	9	Sociedad
Labores	20	Labores
Jardinería	6	Jardinería

11 filas seleccionadas.

14.- A partir de la tabla EMPLE, obtén el apellido de los empleados que lleven mas de 15 años trabajando.

SELECT APELLIDO

FROM EMPLE

WHERE TRUNC (MONTHS_BETWEEN (SYSDATE, FECHA_ALT)/12)>15;

APELLIDO
SANCHEZ
ARROYO
SALA
JIMENEZ
MARTIN
NEGRO
CEREZO
GIL
REY
TOVAR
ALONSO
JIMENO
FERNANDEZ
MUÑOZ

15.- Selecciona el apellido de los empleados de la tabla EMPLE que lleven mas de 16 años trabajando en el departamento "VENTAS".

```
SELECT APELLIDO
FROM EMPLE
WHERE TRUNC(MONTHS_BETWEEN(SYSDATE, FECHA_ALT)/12)>16
AND DEPT_NO=(SELECT DEPT_NO FROM DEPART WHERE DNOMBRE='VENTAS');
```

APELLIDO
ARROYO
SALA
MARTIN
NEGRO
TOVAR
JIMENO

16.- Visualiza el apellido, el salario, y el numero de departamento de aquellos empleados de la tabla EMPLE cuyo salario sea el mayor de su departamento.

```
SELECT APELLIDO, SALARIO, DEPT_NO
FROM EMPLE A
WHERE SALARIO=(SELECT MAX(SALARIO) FROM EMPLE B WHERE A.DEPT_NO=B.DEPT_NO);
```

APELLIDO	SALARIO	DEPT_NO
NEGRO	3005	30
GIL	3000	20
REY	4100	10
FERNANDEZ	3000	20

17.- Visualiza el apellido, el salario y el numero de departamento de aquellos empleados de la tabla EMPLE cuyo salario supere a la media en su departamento.

```
SELECT APELLIDO, SALARIO, DEPT_NO
FROM EMPLE A
WHERE SALARIO>(SELECT AVG(SALARIO) FROM EMPLE B WHERE A.DEPT_NO=B.DEPT_NO);
```

APELLIDO	SALARIO	DEPT_NO
JIMENEZ	2900	20
NEGRO	3005	30
GIL	3000	20
REY	4100	10
FERNANDEZ	3000	20

Actividades Complementarias (Unidad 4).

1.- Obtén en una columna el apellido y el oficio de cada uno de los empleados de la tabla EMPLE, de la siguiente manera: APELLIDO es OFICIO, por ejemplo, SANCHEZ es EMPLEADO (hay que anidar dos funciones CONCAT).
SELECT CONCAT (CONCAT (APELIDO, ' es '), OFICIO) FROM EMPLE;

CONCAT(CONCAT(APELLIDO,'ES'),OFICIO)
SANCHEZ es EMPLEADO
ARROYO es VENDEDOR
SALA es VENDEDOR
JIMENEZ es DIRECTOR
MARTIN es VENDEDOR
NEGRO es DIRECTOR
CEREZO es DIRECTOR
GIL es ANALISTA
REY es PRESIDENTE
TOVAR es VENDEDOR
More than 10 rows available. Increase rows selector to view more rows.

2.- ¿Qué salida obtiene esta SELECT?:

SELECT RPAD ('X', 5, '*.') "Der", LPAD ('X', 5, '*.') "Izq" FROM DUAL;

Der	Izq
X**.	**.X

3.- Visualiza la columna TITULO de la tabla MISTEXTOS sin las comillas de la derecha y de la izquierda ; y el punto de la derecha.

SELECT LTRIM (RTRIM (TITULO, '.'), '') "Titulo sin comillas y pto" FROM MISTEXTOS;

Titulo Sin Comillas Y Pto
METODOLOGÍA DE LA PROGRAMACIÓN
INFORMÁTICA BÁSICA
SISTEMAS OPERATIVOS
SISTEMAS DIGITALES
MANUAL DE C

4.- Visualiza el apellido del empleado y la primera letra del apellido en minúscula.

SELECT APELLIDO, SUBSTR(LOWER (APELIDO),1,1)"Inicial" FROM EMPLE;

APELLIDO	Inicial
SANCHEZ	s
ARROYO	a
SALA	s
JIMENEZ	j
MARTIN	m
NEGRO	n
CEREZO	c
GIL	g
REY	r
TOVAR	t
ALONSO	a
JIMENO	j
FERNANDEZ	f
MUÑOZ	m

5.- A partir de la tabla MISTEXTOS visualiza la columna TITULO sin los caracteres punto y comillas dobles (.").
SELECT LTRIM(RTRIM(TITULO,'."'),'")"Titulo sin caracteres" FROM MISTEXTOS;

Titulo Sin Caracteres
METODOLOGÍA DE LA PROGRAMACIÓN
INFORMÁTICA BÁSICA
SISTEMAS OPERATIVOS
SISTEMAS DIGITALES
MANUAL DE C

6.- Calcula el numero de caracteres de la columna TEMA para todas las filas de la tabla librería. Comenta el resultado obtenido.

SELECT LENGTH (RTRIM(TEMA,' ')) FROM LIBRERIA;

LENGTH(RTRIM(TEMA,''))
11
8
8
9
6
8
8
8
7
10

SELECT LENGTH (RTRIM(TEMA)) FROM LIBRERIA;

Daría el mismo resultado (usado para quitar espacios).

7.- Resta 3 años a la fecha de alta de los empleados de EMPLE.

SELECT ADD_MONTHS(FECHA_ALT,-36) FROM EMPLE;

ADD_MONTHS(FECHA_ALT,-36)
17/12/87
20/02/87
22/02/88
02/04/88
29/09/88
01/05/88
09/06/88
09/11/88
17/11/88
08/09/88
23/09/88
03/12/88
03/12/88
23/01/89

8.- ¿Cuál es la fecha del ultimo día del mes de Febrero del año 2008? ¿Y del año 2009?.

SELECT LAST_DAY ('01022008') FROM DUAL;

LAST_DAY('01022008')
29/02/08

9.- Obtén la fecha de hoy con el siguiente formato: "Hoy es nombre_día, día_mes de nombre_mes de año".

SELECT TO_CHAR (SYSDATE, '"Hoy es " day ", " dd " de " month " de " yyyy') FROM DUAL;

TO_CHAR(SYSDATE,'"HOYES"DAY","DD"DE"MONTH"DE"YYYY')
--

Hoy es miércoles , 25 de enero de 2012
--

10.- Visualiza la suma de salarios de la tabla EMPLE de manera formateada, de tal manera que aparezca el símbolo de la moneda local, el punto para los miles y la coma para los decimales.

SELECT TO_CHAR (SUM (SALARIO), '99G999L')"Suma Salario formateada" FROM EMPLE;

Suma Salario Formateada

30.460€

11.- ¿En qué día de la semana naciste?

SELECT TO_CHAR (TO_DATE ('02071964','ddmmyy'), 'day')"Dia que naci" FROM DUAL;

Dia Que Naci

jueves

12.- Dada la tabla librería, visualiza todas sus filas sustituyendo el tema 'DIBUJO' por 'DISEÑO', y 'LABORES' por 'HOGAR'. En cualquier otro caso, deja el tema como está.

SELECT REPLACE (REPLACE (TEMA, 'DIBUJO', 'DISEÑO'), 'LABORES', 'HOGAR') "Dibujo=diseño y Labores=Hogar" FROM LIBRERIA;

Dibujo=diseño Y Labores=Hogar

INFORMATICA

ECONOMIA

DEPORTES

FILOSOFIA

DISEÑO

MEDICINA

BIOLOGIA

GEOLOGIA

SOCIEDAD

HOGAR

JARDINERIA

13.- Representa en formato carácter los caracteres 1 al 4 del APELLIDO 'SALA' de la tabla EMPLE.

Tablas utilizadas en los ejercicios (Unidad 5)**EMPLE**

EMP_NO Number(4)	APELLIDO Varchar2(10)	OFICIO Varchar2(10)	DIR Number(4)	FECHA_ALT Date	SALARIO Number (10)	COMISION Number(10)	DEPT_NO Number(2)
7369	SANCHEZ	EMPLEADO	7902	17/12/1980	104000		20
7499	ARROYO	VENDEDOR	7698	20/02/1980	208000	39000	30
7521	SALA	VENDEDOR	7698	22/02/1981	162500	65000	30
7566	JIMENEZ	DIRECTOR	7839	02/04/1981	386750		20
7654	MARTIN	VENDEDOR	7698	29/09/1981	162500	182000	30
7698	NEGRO	DIRECTOR	7839	01/05/1981	370500		30
7782	CEREZO	DIRECTOR	7839	09/06/1981	318500		10
7788	GIL	ANALISTA	7566	09/11/1981	390000		20
7839	REY	PRESIDENTE		17/11/1981	650000		10
7844	TOVAR	VENDEDOR	7698	08/09/1981	195000	0	30
7876	ALONSO	EMPLEADO	7788	23/09/1981	143000		20
7900	JIMENO	EMPLEADO	7698	03/12/1981	123500		30
7902	FERNANDEZ	ANALISTA	7566	03/12/1981	390000		20
7934	MUÑOZ	EMPLEADO	7782	23/01/1982	169000		10

DEPART

DEPT_NO Number(2)	DNOMBRE Varchar2(14)	LOC Varchar2(14)
10	CONTABILIDAD	SEVILLA
20	INVESTIGACION	MADRID
30	VENTAS	BARCELONA
40	PRODUCCION	BILBAO

PARALEER

COD_LIBRO Number(15)	NOMBRE_LIBRO Varchar2(40)
100	Cien Años de Soledad
200	Los Mitos Griegos
300	El Camino

ANTIGUOS

NOMBRE Varchar2(20)	EDAD Number(2)	LOCALIDAD Varchar2(15)
MARÍA	20	MADRID
ERNESTO	21	MADRID
ANDRÉS	26	LAS ROZAS
IRENE	24	LAS ROZAS

LEIDOS

COD_LIBRO Number(3)	FECHA Date
300	20/02/1999
200	11/04/1999

NUEVOS

NOMBRE Varchar2(20)	EDAD Number(2)	LOCALIDAD Varchar2(15)
JUAN	18	COSLADA
MAITE	15	ALCALA
SOFÍA	14	ALCALA
ANA	17	ALCALA
ERNESTO	21	MADRID

ALUM

NOMBRE Varchar2(20)	EDAD Number(2)	LOCALIDAD Varchar2(15)
JUAN	18	COSLADA
PEDRO	19	COSLADA
ANA	17	ALCALA
LUISA	18	TORREJÓN
MARÍA	20	MADRID
ERNESTO	21	MADRID
RAQUEL	19	TOLEDO

PERSONAL

COD_CENTRO Number(4)	DNI Number(10)	APELLIDOS Varchar2(30)	FUNCION Varchar2(15)	SALARIO Number(10)
10	1112345	Martínez Salas, Fernando	PROFESOR	220000
10	4123005	Bueno Zarco, Elisa	PROFESOR	220000
10	4122025	Montes García, M.Pilar	PROFESOR	220000
15	1112345	Rivera Silvestre, Ana	PROFESOR	205000
15	9800990	Ramos Ruiz, Luis	PROFESOR	205000
15	8660990	De Lucas Fdez, M.Angel	PROFESOR	205000
22	7650000	Ruiz Lafuente, Manuel	PROFESOR	220000
45	43526789	Serrano Laguía, María	PROFESOR	205000
10	4480099	Ruano Cerezo, Manuel	ADMINISTRATIVO	180000
15	1002345	Albarrán Serrano, Alicia	ADMINISTRATIVO	180000
15	7002660	Muñoz Rey, Felicia	ADMINISTRATIVO	180000
22	5502678	Marín Marín, Pedro	ADMINISTRATIVO	180000
22	6600980	Peinado Gil, Elena	CONSERJE	175000
45	4163222	Sarro Molina, Carmen	CONSERJE	175000

PROFESORES

COD_CENTRO Number(4)	DNI Number(10)	APELLIDOS Varchar2(30)	ESPECIALIDAD Varchar2(16)
10	1112345	Martínez Salas, Fernando	INFORMÁTICA
10	4123005	Bueno Zarco, Elisa	MATEMÁTICAS
10	4122025	Montes García, M.Pilar	MATEMÁTICAS
15	9800990	Ramos Ruiz, Luis	LENGUA
15	1112345	Rivera Silvestre, Ana	DIBUJO
15	8660990	De Lucas Fdez, M.Angel	LENGUA
22	7650000	Ruiz Lafuente, Manuel	MATEMÁTICAS
45	43526789	Serrano Laguía, María	INFORMÁTICA

CENTROS

COD_CENTRO Number(4)	TIPO_CENTRO Char(1)	NOMBRE Varchar2(30)	DIRECCION Varchar2(26)	TELEFONO Varchar2(10)	NUM_PLAZAS Number(4)
10	S	IES El Quijote	Avda. Los Molinos 25	965-887654	538
15	P	CP Los Danzantes	c/Las Musas s/n	985-112322	250
22	S	IES Planeta Tierra	C/Mina 45	925-443400	300
45	P	CP Manuel Hidalgo	C/Granada 5	926-202310	220
50	S	IES Antoñete	C/ Los Toreros 21	989-406090	310

TEMA Char(15)	ESTANTE Char(1)	EJEMPLARES Number(2)
Informática	A	15
Economía	A	10
Deportes	B	8
Filosofía	C	7
Dibujo	C	10
Medicina	C	16
Biología	A	11
Geología	D	7
Sociedad	D	9
Labores	B	20
Jardinería	E	6

LIBRERÍA

TABLAS BANCOS CAP 05**BANCOS**

COD_BANCO	NF_BANCO	NOMBRE_BANC	DOM_FISCAL	POBLACION
1111	122322223	BANCO DE ESPAÑA	GRAN VIA	MADRID
1112	222322223	BANCO DE GUADALAJARA	MAYOR	GUADALAJARA
1113	322322223	BANCO POPULAR	LA PLAZA	TOLEDO
1114	422322223	CAJA CM	RIO DULCE	ALBACETE

SUCURSALES

COD_BANCO	COD_SUCUR	NOMBRE_SUC	DIREC_SUC	LOC_SUC	PROV_SUC
1111	5000	* SUCURSAL 1*	ANCHA 24	TOLEDO	TOLEDO
1111	5001	* SUCURSAL 2*	PILON 33	PASTRANA	GUADALAJARA
1112	6000	* SUCURSAL 6000*	MAYOR 55	ALCALA	ALCALA
1113	4000	* SUCURSAL 4000*	ERAS 77	ARANJUEZ	MADRID

CUENTAS

COD_BANCO	COD_SUCUR	NUM_CTA	FECHA_ALTA	NOMBRE_CTA	DIREC_CTA	POBLA_CTA	SALDO_DEBE	SALDO_HABER
1111	5000	123456789	01/02/12	JUAN	TOLEDO	TOLEDO	200	400
1111	5000	123456788	02/02/12	PEDRO	TOLEDO	TOLEDO	0	200
1111	5001	123456787	31/10/11	ANA	GUADALAJARA	GUADALAJARA	198	-
1111	5001	123456786	02/02/12	MANUEL	GUADALAJARA	GUADALAJARA	0	998
1111	5001	123456785	23/06/11	ANDRES	ALCALA	ALCALA	0	-
1112	6000	33334444	08/02/12	ISABEL	MADRID	MADRID	5100	3000
1112	6000	33334440	21/09/11	MARIA	MADRID	MADRID	200	200

MOVIMIENTOS

COD_BANCO	COD_SUCUR	NUM_CTA	FECHA_MOV	TIPO_MOV	IMPORTE
1112	6000	33334444	08/02/12	I	1000
1112	6000	33334444	07/02/12	I	2000
1112	6000	33334444	06/02/12	R	100
1112	6000	33334444	08/02/12	R	5000
1112	6000	33334440	08/02/12	R	200
1112	6000	33334440	06/02/12	I	200
1111	5000	123456789	01/02/12	I	100
1111	5000	123456789	03/02/12	R	200
1111	5000	123456789	04/02/12	I	300
1111	5000	123456788	08/02/12	I	200
1111	5001	123456787	08/02/12	R	198
1111	5001	123456786	08/02/12	I	998

Ejercicios Propuestos y Adicionales (Unidad 5).

1.- Visualizar los departamentos en los que el salario medio es mayor o igual que la media de todos los salarios.

```
SELECT DEPT_NO, AVG(SALARIO)
FROM EMPLE
GROUP BY DEPT_NO
HAVING AVG(SALARIO)>=(SELECT AVG(SALARIO) FROM EMPLE);
```

2.- a1) Obtén los nombres de departamentos que tengan más de 4 personas trabajando.

SELECT DNOMBRE "Nombre Departamentos"

FROM DEPART

```
WHERE DEPT_NO IN (SELECT DEPT_NO FROM EMPLE GROUP BY DEPT_NO HAVING COUNT(*)>4);
```

Nombre Departamentos

VENTAS
INVESTIGACION

a2) Obtén los nombres de departamentos que tengan más de 4 personas trabajando y numero de empleados.

#Esta sentencia no cuenta el numero de empleados:

SELECT DEPT_NO, DNOMBRE

FROM DEPART

WHERE DEPT_NO

IN (SELECT DEPT_NO FROM EMPLE GROUP BY DEPT_NO HAVING COUNT(*)>4);

DEPT_NO	DNOMBRE
30	VENTAS
20	INVESTIGACION

Esta sentencia si da el resultado que se pide:

SELECT E.DEPARTMENT, COUNT(*)

FROM EMPLOYEE, DEPARTMENT

WHERE E.DEPT_NO=D.DEPT_NO

GROUP BY E.DEPT_NO, DNOMBRE

HAVING COUNT (*)>4;

DEPT_NO	DNOMBRE	COUNT(*)
30	VENTAS	6
20	INVESTIGACION	5

Lo mismo de la forma clásica sin renombrar (da el mismo resultado):

```
SELECT EMPLE.DEPT_NO, DNOMBRE, COUNT (*)
FROM EMPLE, DEPART
WHERE EMPLE.DEPT_NO=DEPART.DEPT_NO
GROUP BY EMPLE.DEPT_NO, DNOMBRE
HAVING COUNT (*)>4;
```

b) Visualiza el número de departamento, el nombre de departamento y el número de empleados del departamento con más empleados.

```
SELECT EMPLE.DEPT_NO, DNOMBRE, COUNT (*)
FROM EMPLE, DEPART
WHERE EMPLE.DEPT_NO=DEPART.DEPT_NO
GROUP BY EMPLE.DEPT_NO, DNOMBRE
HAVING COUNT (*)= (SELECT MAX (COUNT (*)) FROM EMPLE GROUP BY DEPT_NO);
```

DEPT_NO	DNOMBRE	COUNT(*)
30	VENTAS	6

3.- Sentencia Ejemplo de Combinación externa (OUTER JOIN):

```
SELECT D.DEPT_NO, DNOMBRE, COUNT (E.EMP_NO)
FROM EMPLE E, DEPART D
WHERE E.DEPT_NO (+) = D.DEPT_NO
GROUP BY D.DEPT_NO, DNOMBRE;
```

DEPT_NO	DNOMBRE	COUNT(E.EMP_NO)
30	VENTAS	6
10	CONTABILIDAD	3
40	PRODUCCION	0
20	INVESTIGACION	5

a) Analiza lo que ocurre si en lugar de COUNT(E.EMP_NO) ponemos COUNT(*) en la sentencia SELECT anterior.

```
SELECT D.DEPT_NO, DNOMBRE, COUNT (*)
FROM EMPLE E, DEPART D
WHERE E.DEPT_NO (+) = D.DEPT_NO
GROUP BY D.DEPT_NO, DNOMBRE;
```

DEPT_NO	DNOMBRE	COUNT(*)
30	VENTAS	6
10	CONTABILIDAD	3
40	PRODUCCION	1
20	INVESTIGACION	5

Da un 1 en la fila del Dept. 40 porque interpretamos que como aparece una vez la fila hay 1 empleado, esto es debido a usar COUNT (*).

b) Analiza también lo que ocurre si a la derecha de SELECT ponemos E.DEPT_NO en lugar de D.DEPT_NO.

```
SELECT E.DEPT_NO, DNOMBRE, COUNT(E.EMP_NO)
FROM EMPLE E, DEPART D
WHERE E.DEPT_NO (+) = D.DEPT_NO
GROUP BY D.DEPT_NO, DNOMBRE;
```



ORA-00979: no es una expresión GROUP BY

Se corrige la parte del GROUP BY con E.DEPT_NO también y da:

```
SELECT E.DEPT_NO, DNOMBRE, COUNT(E.EMP_NO)
FROM EMPLE E, DEPART D
WHERE E.DEPT_NO (+) = D.DEPT_NO
GROUP BY E.DEPT_NO, DNOMBRE;
```

DEPT_NO	DNOMBRE	COUNT(E.EMP_NO)
-	PRODUCCION	0
30	VENTAS	6
10	CONTABILIDAD	3
20	INVESTIGACION	5

4.- Esta consulta también se puede hacer usando el operador IN. Escribe la consulta anterior utilizando el operador IN.

5.- a) Visualizar los nombres de los alumnos de la tabla ALUM que aparezcan en alguna de estas tablas: NUEVOS y ANTIGUOS. o ALUM intersección (NUEVOS unión ANTIGUOS) o (ALUM intersección ANTIGUOS) unión (ALUM intersección ANTIGUOS)

OR = unión = UNION AND = intersección = INTERSEC = IN

b) Escribir las distintas formas en que se puede poner la consulta anterior llegando al mismo resultado.

1º Forma de hacerlo:

```
SELECT NOMBRE
FROM ALUM
WHERE NOMBRE
IN (SELECT NOMBRE FROM NUEVOS UNION SELECT NOMBRE FROM ANTIGUOS);
```

NOMBRE
JUAN
ANA
MARÍA
ERNESTO

2^a Forma de hacerlo:

```
SELECT NOMBRE
FROM ALUM
WHERE NOMBRE
IN (SELECT NOMBRE FROM NUEVOS)
OR NOMBRE
IN (SELECT NOMBRE FROM ANTIGUOS);
```

NOMBRE
JUAN
ANA
MARÍA
ERNESTO

3^a Forma de hacerlo:

```
SELECT NOMBRE
FROM ALUM
INTERSECT (SELECT NOMBRE FROM NUEVOS UNION SELECT NOMBRE FROM
ANTIGUOS);
```

NOMBRE
ANA
ERNESTO
JUAN
MARÍA

c) Visualizar los nombres de los alumnos de la tabla ALUM que aparezcan en las dos tablas: NUEVOS y ANTIGUOS. O : ALUM intersección (NUEVOS intersección ANTIGUOS)

1^a Forma de hacerlo:

```
SELECT NOMBRE
FROM ALUM
WHERE NOMBRE IN (SELECT NOMBRE FROM NUEVOS INTERSECT SELECT NOMBRE
FROM ANTIGUOS);
```

NOMBRE
ERNESTO

2^a Forma de hacerlo:

```
SELECT NOMBRE
FROM ALUM
INTERSECT (SELECT NOMBRE FROM NUEVOS INTERSECT SELECT NOMBRE FROM
ANTIGUOS);
```

3^a Forma de hacerlo:

SELECT NOMBRE

FROM ALUM

INTERSECT (SELECT NOMBRE FROM NUEVOS WHERE NOMBRE IN (SELECT NOMBRE FROM ANTIGUOS));

d) Visualizar los nombres de los alumnos de la tabla ALUM que no aparezcan en las tablas: NUEVOS y ANTIGUOS. O lo que es lo mismo: ALUM - (NUEVOS **unión** ANTIGUOS)

- = **NOT IN = MINUS**

1^a Forma de hacerlo:

SELECT NOMBRE

FROM ALUM

WHERE NOMBRE

NOT IN (SELECT NOMBRE FROM NUEVOS UNION SELECT NOMBRE FROM ANTIGUOS);

NOMBRE
PEDRO
LUISA
RAQUEL

2^a Forma de hacerlo:

SELECT NOMBRE

FROM ALUM

MINUS (SELECT NOMBRE FROM NUEVOS UNION SELECT NOMBRE FROM ANTIGUOS);

NOMBRE
LUISA
PEDRO
RAQUEL

6.- A partir de la tabla EMPLE, visualizar el número de vendedores del departamento 'VENTAS'.

SELECT DEPT_NO "Dpto. VENTAS", COUNT(*) "Empleados"

FROM EMPLE

WHERE DEPT_NO=(SELECT DEPT_NO FROM DEPART WHERE DNOMBRE='VENTAS')

GROUP BY DEPT_NO ;

Dpto. VENTAS	Empleados
30	6

7.- Dada la tabla LIBRERIA, visualizar por cada estante la suma de los ejemplares.

SELECT ESTANTE, SUM (EJEMPLARES) "Suma Ejemplares"

FROM LIBRERIA

GROUP BY ESTANTE;

ESTANTE	Suma Ejemplares
D	16
A	36
B	28
C	33
E	6

8.- Visualizar el estante con más ejemplares de la tabla LIBRERIA.

```
SELECT ESTANTE, SUM (EJEMPLARES) "Suma Ejemplares"
FROM LIBRERIA
GROUP BY ESTANTE
HAVING SUM (EJEMPLARES) = (SELECT MAX(SUM(EJEMPLARES))
                             FROM LIBRERIA
                             GROUP BY ESTANTE);
```

ESTANTE	Suma Ejemplares
A	36

9.- En la tabla PERSONAL, obtener por cada función el número de trabajadores.

```
SELECT FUNCION, COUNT(*) "Numero Trabajadores"
FROM PERSONAL
GROUP BY FUNCION;
```

FUNCION	Numero Trabajadores
CONSERJE	2
PROFESOR	8
ADMINISTRATIVO	4

10.- Visualizar los diferentes estantes de la tabla LIBRERIA ordenados descendenteamente por estante.

```
SELECT ESTANTE
FROM LIBRERIA
GROUP BY ESTANTE
ORDER BY ESTANTE DESC;
```

ESTANTE
E
D
C
B
A

Otra forma de hacerlo que da el mismo resultado:

```
SELECT DISTINCT ESTANTE
FROM LIBRERIA
ORDER BY ESTANTE DESC;
```

11.- Averiguar cuántos temas tiene cada estante de la tabla LIBRERÍA.

```
SELECT ESTANTE, COUNT(*) "Numero de Temas"
FROM LIBRERIA
GROUP BY ESTANTE;
```

ESTANTE	Numero De Temas
D	2
A	3
B	2
C	3
E	1

12.- Visualizar los estantes que tengan tres temas en la tabla LIBRERÍA.

```
SELECT ESTANTE, COUNT(*) "Numero de Temas"
FROM LIBRERIA
GROUP BY ESTANTE
HAVING COUNT(*)=3;
```

ESTANTE	Numero De Temas
A	3
C	3

Actividades complementarias Cap. 5 del Libro (pag. 167)

1.- Partiendo de la tabla EMPLE, visualiza por cada oficio de los empleados del departamento 'VENTAS' la suma de salarios.

```
SELECT DEPT_NO, OFICIO, SUM (SALARIO)"Suma salarios"
```

```
FROM EMPLE
```

```
WHERE DEPT_NO= (SELECT DEPT_NO FROM DEPART WHERE DNOMBRE='VENTAS')
GROUP BY DEPT_NO, OFICIO ORDER BY SUM (SALARIO);
```

DEPT_NO	OFICIO	Suma Salarios
30	EMPLEADO	1335
30	DIRECTOR	3005
30	VENDEDOR	6075

2.- Selecciona aquellos apellidos de la tabla EMPLE cuyo salario sea igual a la media del salario en su departamento.

```
SELECT APELLIDO
```

```
FROM EMPLE E1
```

```
WHERE SALARIO=(SELECT AVG(SALARIO) FROM EMPLE E2 WHERE E2.DEPT_NO =
E1.DEPT_NO GROUP BY DEPT_NO);
```

no data found

3.- A partir de la tabla EMPLE, visualiza el numero de empleados de cada departamento cuyo oficio sea 'EMPLEADO'.

```
SELECT DEPT_NO "Departamento", COUNT (*) "Empleados con Oficio EMPLEADO"
```

```
FROM EMPLE
```

```
WHERE OFICIO='EMPLEADO'
```

```
GROUP BY DEPT_NO;
```

Departamento	Empleados Con Oficio EMPLEADO
30	1
20	2
10	1

4.- Desde la tabla EMPLE, visualiza el departamento que tenga mas empleados cuyo oficio sea 'EMPLEADO'.

```
SELECT DEPT_NO "Departamento", COUNT (*) "Empleados con Oficio EMPLEADO"
```

```
FROM EMPLE
```

```
WHERE OFICIO='EMPLEADO'
```

```
GROUP BY DEPT_NO
```

```
HAVING COUNT(*)=(SELECT MAX (COUNT(*)) FROM EMPLE WHERE
OFICIO='EMPLEADO' GROUP BY DEPT_NO );
```

Departamento	Empleados Con Oficio EMPLEADO
20	2

5.- A partir de las tablas EMPLE y DEPART, visualiza el numero de departamento y el nombre de departamento que tenga mas empleados cuyo oficio sea 'EMPLEADO'.

```
SELECT DNOMBRE "Nombre Departamento", E.DEPT_NO "Departamento", COUNT (*)
"Empleados con Oficio EMPLEADO"
FROM EMPLE E, DEPART D
WHERE E.DEPT_NO=D.DEPT_NO AND OFICIO='EMPLEADO'
GROUP BY E.DEPT_NO, DNOMBRE
HAVING COUNT(*)=(SELECT MAX (COUNT(*)) FROM EMPLE WHERE
OFICIO='EMPLEADO' GROUP BY DEPT_NO );
```

Nombre Departamento	Departamento	Empleados Con Oficio EMPLEADO
INVESTIGACION	20	2

6.- Busca los departamento que tienen mas de dos personas trabajando en la misma profesión.

```
SELECT DNOMBRE
FROM DEPART
WHERE DEPT_NO=(SELECT DISTINCT (DEPT_NO) FROM EMPLE GROUP BY
EMPLE.DEPT_NO, OFICIO HAVING COUNT(*)>2);
```

DNOMBRE
VENTAS

```
SELECT DEPT_NO, COUNT(*), OFICIO
FROM EMPLE
GROUP BY DEPT_NO, OFICIO HAVING COUNT(*)>2;
```

DEPT_NO	COUNT(*)	OFICIO
30	4	VENDEDOR

10.- Realiza una consulta en la que aparezca por cada centro y en cada especialidad el numero de profesores. Si el centro no tiene profesores, debe aparecer un 0 en la columna de numero de profesores. Las columnas a visualizar son: nombre de centro, especialidad y numero de profesores.

```
SELECT NOMBRE "Centro", ESPECIALIDAD "Especialidad", COUNT (DNI)"Profesores"
FROM CENTROS C, PROFESORES P
WHERE C.COD_CENTRO =P.COD_CENTRO (+)
GROUP BY NOMBRE, ESPECIALIDAD ORDER BY COUNT (DNI) DESC;
```

Centro	Especialidad	Profesores
IES El Quijote	MATEMÁTICAS	2
CP Los Danzantes	LENGUA	2
IES El Quijote	INFORMÁTICA	1
IES Planeta Tierra	MATEMÁTICAS	1
CP Manuel Hidalgo	INFORMÁTICA	1
CP Los Danzantes	DIBUJO	1
IES Antoñete	-	0

11.- Obtén por cada centro el numero de empleados. Si el centro carece de empleados, ha de aparecer un 0 como numero de empleados.

```
SELECT NOMBRE "Centro", COUNT (DNI)"Profesores"
FROM CENTROS C, PERSONAL P
WHERE C.COD_CENTRO =P.COD_CENTRO (+)
GROUP BY NOMBRE ORDER BY COUNT (DNI) DESC;
```

Centro	Profesores
CP Los Danzantes	5
IES El Quijote	4
IES Planeta Tierra	3
CP Manuel Hidalgo	2
IES Antoñete	0

12.- Obtener la especialidad con menos profesores.

```
SELECT ESPECIALIDAD
FROM PROFESORES
GROUP BY ESPECIALIDAD
HAVING COUNT (*) = (SELECT MIN (COUNT (*)) FROM PROFESORES GROUP BY
ESPECIALIDAD);
```

ESPECIALIDAD
DIBUJO

13.- Obten el banco con mas sucursales. Los datos a obtener son:

Nombre de Banco NºSucursales

XXXXXX XXX

```
SELECT NOMBRE_BANC "BANCO", COUNT (*) "Sucursales"
FROM BANCOS B, SUCURSALES S
WHERE B.COD_BANCO=S.COD_BANCO
GROUP BY NOMBRE_BANC
HAVING COUNT (*) = (SELECT MAX (COUNT(*)) FROM BANCOS B, SUCURSALES S
WHERE B.COD_BANCO=S.COD_BANCO GROUP BY NOMBRE_BANC);
```

BANCO	Sucursales
BANCO DE ESPAÑA	2

```
SELECT NOMBRE_BANC "BANCO", COUNT (COD_SUCUR) "Sucursales"
FROM BANCOS B, SUCURSALES S
WHERE B.COD_BANCO=S.COD_BANCO
GROUP BY NOMBRE_BANC
HAVING COUNT(COD_SUCUR)=(SELECT MAX (COUNT(COD_SUCUR)) FROM
SUCURSALES GROUP BY COD_BANCO);
```

BANCO	Sucursales
BANCO DE ESPAÑA	2

14.- El saldo actual de los bancos de 'GUADALAJARA', 1 fila por cada banco.

<u>Nombre de Banco</u>	<u>Saldo Debe</u>	<u>Saldo Haber</u>
xxxxx	xx.xx	xx,xx

```
SELECT B.NOMBRE_BANC "Nombre Banco", SUM (SALDO_DEBE) "Saldo Debe", SUM (SALDO_HABER) "Saldo Haber"
FROM BANCOS B, CUENTAS C
WHERE B.COD_BANCO = C.COD_BANCO
AND POBLACION = 'GUADALAJARA'
GROUP BY B.NOMBRE_BANC;
```

Nombre Banco	Saldo Debe	Saldo Haber
BANCO DE GUADALAJARA	5300	3200

15.- Datos de la cuenta o cuentas con mas movimientos.

<u>Nombre Cta</u>	<u>Nºmovimientos</u>
-------------------	----------------------

xxxxx	xx
-------	----

```
SELECT M.NUM_CTA "Numero Cta", NOMBRE_CTA "NOMBRE Cta", COUNT(*) "N Movimientos"
FROM MOVIMIENTOS M, CUENTAS C
WHERE M.NUM_CTA=C.NUM_CTA
AND C.COD_SUCUR=M.COD_SUCUR
AND C.COD_BANCO=M.COD_BANCO
GROUP BY M.NUM_CTA, NOMBRE_CTA
HAVING COUNT(*)=(SELECT MAX (COUNT (*)) FROM MOVIMIENTOS GROUP BY NUM_CTA);
```

Numero Cta	NOMBRE Cta	N Movimientos
33334444	ISABEL	4

16.- El nombre de la sucursal que haya tenido mas suma de reintegros.

<u>Nombre Sucursal</u>	<u>Suma Reintegros</u>
------------------------	------------------------

xxxxx	xx,xx
-------	-------

```
SELECT S.NOMBRE_SUC, SUM (IMPORTE) "Suma Reintegros"
FROM SUCURSALES S, MOVIMIENTOS M
WHERE S.COD_BANCO = M. COD_BANCO AND S.COD_SUCUR = M. COD_SUCUR and M.TIPO_MOV = 'R'
GROUP BY NOMBRE_SUC
HAVING SUM(IMPORTE) = (SELECT MAX( SUM (IMPORTE)) FROM MOVIMIENTOS M
WHERE M.TIPO_MOV = 'R'
GROUP BY COD_BANCO, COD_SUCUR);
```

NOMBRE_SUC	Suma Reintegros
* SUCURSAL 6000*	5300

Tablas utilizadas en los ejercicios**EMPLE**

EMP_NO Number(4)	APELLIDO Varchar2(10)	OFICIO Varchar2(10)	DIR Number(4)	FECHA_ALT Date	SALARIO Number (10)	COMISION Number(10)	DEPT_NO Number(2)
7369	SANCHEZ	EMPLEADO	7902	17/12/1980	104000		20
7499	ARROYO	VENDEDOR	7698	20/02/1980	208000	39000	30
7521	SALA	VENDEDOR	7698	22/02/1981	162500	65000	30
7566	JIMENEZ	DIRECTOR	7839	02/04/1981	386750		20
7654	MARTIN	VENDEDOR	7698	29/09/1981	162500	182000	30
7698	NEGRO	DIRECTOR	7839	01/05/1981	370500		30
7782	CEREZO	DIRECTOR	7839	09/06/1981	318500		10
7788	GIL	ANALISTA	7566	09/11/1981	390000		20
7839	REY	PRESIDENTE		17/11/1981	650000		10
7844	TOVAR	VENDEDOR	7698	08/09/1981	195000	0	30
7876	ALONSO	EMPLEADO	7788	23/09/1981	143000		20
7900	JIMENO	EMPLEADO	7698	03/12/1981	123500		30
7902	FERNANDEZ	ANALISTA	7566	03/12/1981	390000		20
7934	MUÑOZ	EMPLEADO	7782	23/01/1982	169000		10

DEPART

DEPT_NO Number(2)	DNOMBRE Varchar2(14)	LOC Varchar2(14)
10	CONTABILIDAD	SEVILLA
20	INVESTIGACION	MADRID
30	VENTAS	BARCELONA
40	PRODUCCION	BILBAO

ANTIGUOS

NOMBRE Varchar2(20)	EDAD Number(2)	LOCALIDAD Varchar2(15)
MARÍA	20	MADRID
ERNESTO	21	MADRID
ANDRÉS	26	LAS ROZAS
IRENE	24	LAS ROZAS

NUEVOS

NOMBRE Varchar2(20)	EDAD Number(2)	LOCALIDAD Varchar2(15)
JUAN	18	COSLADA
MAITE	15	ALCALA
SOFÍA	14	ALCALA
ANA	17	ALCALA
ERNESTO	21	MADRID

ALUM

NOMBRE Varchar2(20)	EDAD Number(2)	LOCALIDAD Varchar2(15)
JUAN	18	COSLADA
PEDRO	19	COSLADA
ANA	17	ALCALA
LUISA	18	TORREJÓN
MARÍA	20	MADRID
ERNESTO	21	MADRID
RAQUEL	19	TOLEDO

PERSONAL

COD_CENTRO Number(4)	DNI Number(10)	APELLIDOS Varchar2(30)	FUNCION Varchar2(15)	SALARIO Number(10)
10	1112345	Martínez Salas, Fernando	PROFESOR	220000
10	4123005	Bueno Zarco, Elisa	PROFESOR	220000
10	4122025	Montes García, M.Pilar	PROFESOR	220000
15	1112345	Rivera Silvestre, Ana	PROFESOR	205000
15	9800990	Ramos Ruiz, Luis	PROFESOR	205000
15	8660990	De Lucas Fdez, M.Angel	PROFESOR	205000
22	7650000	Ruiz Lafuente, Manuel	PROFESOR	220000
45	43526789	Serrano Laguía, María	PROFESOR	205000
10	4480099	Ruano Cerezo, Manuel	ADMINISTRATIVO	180000
15	1002345	Albarrán Serrano, Alicia	ADMINISTRATIVO	180000
15	7002660	Muñoz Rey, Felicia	ADMINISTRATIVO	180000
22	5502678	Marín Marín, Pedro	ADMINISTRATIVO	180000
22	6600980	Peinado Gil, Elena	CONSERJE	175000
45	4163222	Sarro Molina, Carmen	CONSERJE	175000

PROFESORES

COD_CENTRO Number(4)	DNI Number(10)	APELLIDOS Varchar2(30)	ESPECIALIDAD Varchar2(16)
10	1112345	Martínez Salas, Fernando	INFORMÁTICA
10	4123005	Bueno Zarco, Elisa	MATEMÁTICAS
10	4122025	Montes García, M.Pilar	MATEMÁTICAS
15	9800990	Ramos Ruiz, Luis	LENGUA
15	1112345	Rivera Silvestre, Ana	DIBUJO
15	8660990	De Lucas Fdez, M.Angel	LENGUA
22	7650000	Ruiz Lafuente, Manuel	MATEMÁTICAS
45	43526789	Serrano Laguía, María	INFORMÁTICA

CENTROS

COD_CENTRO Number(4)	TIPO_CENTRO Char(1)	NOMBRE Varchar2(30)	DIRECCION Varchar2(26)	TELEFONO Varchar2(10)	NUM_PLAZAS Number(4)
10	S	IES El Quijote	Avda. Los Molinos 25	965-887654	538
15	P	CP Los Danzantes	c/Las Musas s/n	985-112322	250
22	S	IES Planeta Tierra	C/Mina 45	925-443400	300
45	P	CP Manuel Hidalgo	C/Granada 5	926-202310	220
50	S	IES Antoñete	C/ Los Toreros 21	989-406090	310

ARTICULOS

ARTICULO Varchar2(20)	COD_FABRICANTE Number(3)	PESO Number(3)	CATEGORIA Varchar2(10)	PRECIO_VENTA Number(4)	PRECIO_COSTO Number(4)	EXISTENCIAS Number(5)
Macarrones	20	1	Primera	100	98	120
Tallarines	20	2	Primera	120	100	100
Tallarines	20	1	Segunda	99	50	100
Macarrones	20	1	Tercera	80	50	100
Atún	10	3	Primera	200	150	220
Atún	10	3	Segunda	150	100	220
Atún	10	3	Tercera	100	50	220
Sardinillas	10	1	Primera	250	200	200
Sardinillas	10	1	Segunda	200	160	200
Sardinillas	10	1	Tercera	100	150	220
Mejillones	10	1	Tercera	90	50	200
Mejillones	10	1	Primera	200	150	300
Macarrones	25	1	Primera	90	68	150
Tallarines	25	1	Primera	100	90	100
Fideos	25	1	Segunda	75	50	100
Fideos	25	1	Primera	100	80	100
Galletas Cuadradas	15	1	Primera	100	80	100
Galletas Cuadradas	15	1	Segunda	70	50	100
Galletas Cuadradas	15	1	Tercera	50	40	100
Barquillos	15	1	Primera	100	80	100
Barquillos	15	1	Segunda	100	80	100
Canutillos	15	2	Primera	170	150	110
Canutillos	15	2	Segunda	120	150	110
Leche entera	30	1	Primera	110	100	300
Leche desnat.	30	1	Primera	120	100	300
Leche semi.	30	1	Primera	130	110	300
Leche entera	30	2	Primera	210	200	300
Leche desnat.	30	2	Primera	220	200	300
Leche semi.	30	2	Primera	230	210	300
Mantequilla	30	1	Primera	510	400	200
Mantequilla	30	1	Segunda	450	340	200

FABRICANTES

COD_FABRICANTE Number(3)	NOMBRE Varchar2(15)	PAIS Varchar2(15)
10	CALVO	ESPAÑA
15	LU	BELGICA
20	BARILLA	ITALIA
25	GALLO	ESPAÑA
30	PRESIDENT	FRANCIA

TIENDAS

NIF Varchar2(10)	NOMBRE Varchar2(20)	DIRECCIÓN Varchar2(20)	POBLACIÓN Varchar2(20)	PROVINCIA Varchar2(20)	CODPOSTAL Number(5)
1111-A	Almacenes Pérez	C/Toledo, 20	Siguenza	GUADALAJARA	19104
5555-B	La Gacela	C/Santander Rios, 45	Azuqueca	GUADALAJARA	19209
2222-A	Comestibles Rodolfo	C/ del Val s/n	Alcalá de Henares	MADRID	28804
4444-A	La Pasta Gansa	C/Alcalá, 2	Ajalvir	MADRID	28765
3333-N	Ultramarinos Montse	Avda. Pio 10	Toledo	TOLEDO	45100
4141-B	Todo Toledo	C/Avila 24	Talavera	TOLEDO	45199

PEDIDOS

NIF Varchar2(10)	ARTICULO Varchar2(20)	COD_ FABRICANTE Number(3)	PESO Number(3)	CATEGORIA Varchar2(10)	FECHA_PEDIDO Date	UNIDADES_ PEDIDAS Number(4)
5555-B	Macarrones	20	1	Primera	18/02/1999	30
5555-B	Atún	10	3	Primera	21/02/1999	10
5555-B	Atún	10	3	Segunda	11/03/1999	4
5555-B	Sardinillas	10	1	Segunda	11/03/1999	10
5555-B	Macarrones	25	1	Primera	14/04/1999	12
5555-B	Fideos	25	1	Segunda	18/05/1999	24
5555-B	Fideos	25	1	Segunda	19/05/1999	20
5555-B	Galletas Cuadradas	15	1	Segunda	20/06/1999	15
1111-A	Barquillos	15	1	Primera	20/02/1999	10
1111-A	Canutillos	15	2	Segunda	10/04/1999	12
1111-A	Leche Semi.	30	1	Primera	24/06/1999	5
1111-A	Leche Semi.	30	2	Primera	04/07/1999	11
1111-A	Mantequilla	30	1	Primera	10/07/1999	10
4141-B	Macarrones	20	1	Primera	16/04/1999	30
4141-B	Atún	10	3	Primera	21/06/1999	10
4141-B	Atún	10	3	Segunda	12/08/1999	9
2222-A	Sardinillas	10	1	Segunda	12/08/1999	20
2222-A	Sardinillas	10	1	Tercera	12/08/1999	22
2222-A	Mejillones	10	1	Primera	sysdate	1000
3333-A	Macarrones	25	1	Primera	10/11/1999	8
3333-A	Tallarines	25	1	Primera	12/11/1999	9
3333-A	Fideos	25	1	Primera	15/11/1999	11
3333-A	Galletas Cuadradas	15	1	Primera	20/11/1999	6
3333-A	Barquillos	15	1	Segunda	20/11/1999	40
3333-A	Canutillos	15	2	Segunda	20/11/1999	10

VENTAS

NIF Varchar2(10)	ARTICULO Varchar2(20)	COD_ FABRICANTE Number(3)	PESO Number(3)	CATEGORIA Varchar2(10)	FECHA_VENTA Date	UNIDADES_ VENDIDAS Number(4)
5555-B	Macarrones	20	1	Primera	19/02/2006	5
5555-B	Atún	10	3	Primera	19/02/2006	6
5555-B	Atún	10	3	Segunda	20/03/2006	15
5555-B	Sardinillas	10	1	Segunda	20/03/2006	8
5555-B	Macarrones	25	1	Primera	17/04/2006	2
5555-B	Fideos	25	1	Segunda	18/05/2006	14
5555-B	Leche semi.	30	1	Primera	24/06/2006	5
2222-A	Galletas Cuadradas	15	1	Segunda	20/06/2006	5
2222-A	Barquillos	15	1	Primera	20/02/2006	6
2222-A	Canutillos	15	2	Segunda	10/06/2006	2
2222-A	Sardinillas	10	1	Segunda	15/08/2006	5
2222-A	Sardinillas	10	1	Tercera	15/08/2006	5
3333-A	Leche semi.	30	2	Primera	06/07/2006	11
3333-A	Mantequilla	30	1	Primera	16/07/2006	10
3333-A	Macarrones	20	1	Primera	26/04/2006	30
3333-A	Atún	10	3	Primera	26/04/2006	10
3333-A	Barquillos	15	1	Segunda	25/11/2006	4
3333-A	Canutillos	15	2	Segunda	25/11/2006	4
4141-B	Atún	10	3	Segunda	12/08/2006	2
4141-B	Macarrones	25	1	Primera	10/11/2006	2
4141-B	Tallarines	25	1	Primera	11/11/2006	3
4141-B	Canutillos	15	2	Segunda	11/11/2006	8

NOMBRES

NOMBRE Varchar2(15)	EDAD Number(2)
PEDRO	17
JUAN	17
MARÍA	16
CLARA	14
	15
	18
JESÚS	

Actividades complementarias 1 (Unidad 6)

Tablas ALUM, NUEVO y ANTIQUOS

1.- Dadas las tablas ALUM y NUEVOS, insertar en la tabla ALUM los nuevos alumnos.

INSERT INTO ALUM

SELECT * FROM NUEVOS MINUS SELECT * FROM ALUM;

2.- Borrar de la tabla ALUM los ANTIQUOS alumnos.

DELETE FROM ALUM

WHERE (NOMBRE, EDAD, LOCALIDAD)

IN (SELECT * FROM ANTIQUOS);

2 row(s) deleted.

Tablas EMPLE y DEPART

3.- Insertar a un empleado de apellido 'SAAVEDRA' con número 2000. La fecha de alta será la actual, el SALARIO será el mismo salario de 'SALA' más el 20 por 100 ($\text{SALARIO} * 1.2$) y el resto de datos serán los mismos que los datos de 'SALA'.

INSERT INTO EMPLE

SELECT 2000, 'SAAVEDRA', OFICIO, DIR, SYSDATE, SALARIO*1.2, COMISION, DEPT_NO
FROM EMPLE

WHERE APELLIDO='SALA';

4.- Modificar el número de departamento de 'SAAVEDRA'. El nuevo departamento será el departamento donde hay más empleados cuyo oficio sea 'EMPLEADO'.

UPDATE EMPLE SET DEPT_NO =

(SELECT DEPT_NO FROM EMPLE WHERE OFICIO = 'EMPLEADO'
GROUP BY DEPT_NO HAVING COUNT(*) = (SELECT MAX(COUNT(*)) FROM
EMPLE WHERE OFICIO = 'EMPLEADO' GROUP BY DEPT_NO))
WHERE APELLIDO = 'SAAVEDRA';

5.- Borrar todos los departamentos de la tabla DEPART para los cuales no existan empleados en EMPLE.

DELETE FROM DEPART

WHERE DEPT_NO =

(SELECT DEPT_NO FROM DEPART WHERE DEPT_NO NOT IN
(SELECT DEPT_NO FROM EMPLE));

1 row(s) deleted.

Otra forma seria:

DELETE FROM DEPART

WHERE DEPT_NO IN

(SELECT DEPT_NO FROM DEPART MINUS SELECT DEPT_NO FROM EMPLE);

Otra forma seria:

```
DELETE FROM DEPART  
WHERE DEPT_NO NOT IN (SELECT DEPT_NO FROM EMPLE);
```

Tablas PERSONAL, PROFESORES y CENTROS

6.- Modificar el número de plazas con un valor igual a la mitad en aquellos centros con menos de dos profesores.

```
UPDATE CENTROS
```

```
SET NUM_PLAZAS=NUM_PLAZAS/2  
WHERE COD_CENTRO IN (SELECT COD_CENTRO FROM PROFESORES GROUP BY  
COD_CENTRO HAVING COUNT(*)<2);
```

```
0 row(s) updated.
```

no hay centros con menos de 2 profesores.

7.- Eliminar los centros que no tengan personal.

```
DELETE FROM CENTROS
```

```
WHERE COD_CENTRO
```

```
IN (SELECT COD_CENTRO FROM CENTROS MINUS SELECT COD_CENTRO FROM  
PERSONAL);
```

```
1 row(s) deleted.
```

Ha borrado el centro con código 50

Otra forma:

```
DELETE FROM CENTROS
```

```
WHERE COD_CENTRO
```

```
NOT IN (SELECT DISTINCT COD_CENTRO FROM PERSONAL);
```

8.- Añadir un nuevo profesor en el centro o en los centros cuyo número de administrativos sea 1 en la especialidad de 'IDIOMA', con DNI 8790055 y de nombre 'Clara Salas'.

```
INSERT INTO PROFESORES
```

```
SELECT DISTINCT COD_CENTRO, 879055, 'Clara Salas', 'IDIOMA'
```

```
FROM PERSONAL
```

```
WHERE COD_CENTRO
```

```
IN (SELECT COD_CENTRO FROM PERSONAL WHERE FUNCION='ADMINISTRATIVO'  
GROUP BY COD_CENTRO HAVING COUNT(*)=1);
```

```
2 row(s) inserted.
```

COD_CENTRO	DNI	APELLIDOS	ESPECIALIDAD
10	879055	Clara Salas	IDIOMA
22	879055	Clara Salas	IDIOMA

9.- Borrar al personal que esté en centros de menos de 300 plazas y con menos de dos profesores.

```
DELETE FROM PERSONAL  
WHERE COD_CENTRO IN  
(SELECT COD_CENTRO FROM CENTROS WHERE NUM_PLAZAS<300)  
AND COD_CENTRO IN (SELECT COD_CENTRO FROM PERSONAL GROUP BY  
COD_CENTRO HAVING COUNT(*)<2);
```

0 row(s) deleted.

No hay centros con menos de 2 profesores.

10.- Borrar a los profesores que estén en la tabla PROFESORES y que no estén en la tabla PERSONAL.

```
DELETE FROM PROFESORES  
WHERE DNI IN  
(SELECT DNI FROM PROFESORES MINUS SELECT DNI FROM PERSONAL);
```

3 row(s) deleted.

Ejercicio 1 del libro pag. 171

```
INSERT INTO PROFESORES  
VALUES (22, 23444800, 'Gonzalez Sevilla, Miguel A.', 'HISTORIA');
```

Escribe la sentencia INSERT anterior de otra manera:

```
INSERT INTO PROFESORES (COD_CENTRO, DNI, APELLIDOS, ESPECIALIDAD)  
VALUES (22, 23444800, 'Gonzalez Sevilla, Miguel A.', 'HISTORIA');
```

1 row(s) inserted.

inserta un profesor cuya especialidad supere los 16 caracteres de longitud.

```
INSERT INTO PROFESORES (COD_CENTRO, DNI, APELLIDOS, ESPECIALIDAD)
```

```
VALUES (22, 23444806, 'Gonzalez Perez, Angel', 'HISTORIA del arte contemporaneo de  
hoy');
```

ORA-12899: el valor es demasiado grande para la columna "HR"."PROFESORES"."ESPECIALIDAD" (real: 38, máximo: 16)

Ejercicio 3 del libro pag. 174

aumenta en 100€ el salario y en 10 € la comisión a todos los empleados del departamento 10, de la tabla EMPLE.

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7782	CEREZO	DIRECTOR	7839	09/06/91	2885	-	10
7839	REY	PRESIDENTE	-	17/11/91	4100	-	10
7934	MUÑOZ	EMPLEADO	7782	23/01/92	1690	-	10

```
UPDATE EMPLE
```

```
SET SALARIO=SALARIO+100, COMISION=COMISION+10 WHERE DEPT_NO=10;
```

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7782	CEREZO	DIRECTOR	7839	09/06/91	2985	-	10
7839	REY	PRESIDENTE	-	17/11/91	4200	-	10
7934	MUÑOZ	EMPLEADO	7782	23/01/92	1790	-	10

Actividades complementarias 2 (Unidad 6)

Un almacén de distribución de artículos desea mantener información sobre las ventas hechas por las tiendas que compran al almacén. Dispone de las siguientes tablas para mantener esta información:

Tablas ARTICULOS, FABRICANTES, TIENDAS, PEDIDOS y VENTAS

Artículos: Almacena cada uno de los artículos que el almacén puede abastecer a las tiendas. Cada artículo viene determinado por las columnas : ARTICULO, COD_FABRICANTE, PESO y CATEGORIA. La categoría puede ser 'Primera', 'Segunda' o 'Tercera'.

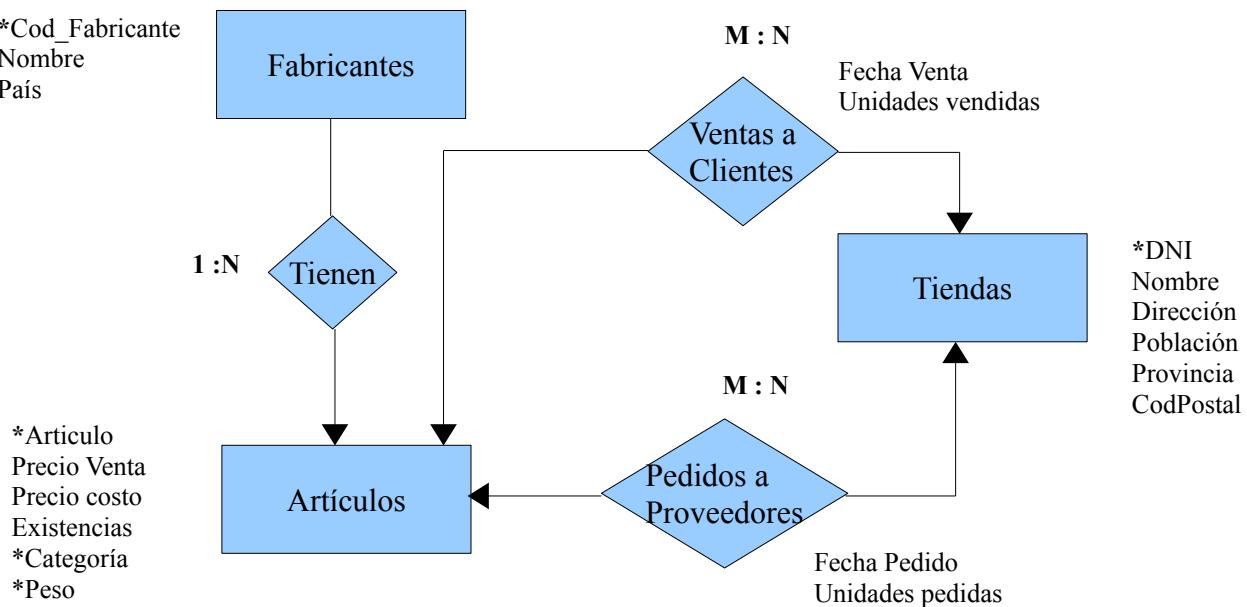
Fabricantes: Contiene los países de origen de los fabricantes de artículos. Cada COD_FABRICANTE tiene su país.

Tiendas: Almacena los datos de las tiendas que venden artículos. Cada tienda se identifica por su NIF.

Pedidos: Son los pedidos que realizan las tiendas al almacén. Cada pedido se identifica por: NIF, ARTICULO, COD_FABRICANTE, PESO, CATEGORIA y FECHA_PEDIDO. Cada fila de la tabla representa un pedido.

Ventas: Almacena las ventas de artículos que hace cada una de las tiendas. Cada venta se identifica por: NIF, ARTICULO, COD_FABRICANTE, PESO, CATEGORIA y FECHA_VENTA. Cada fila de la tabla representa una venta.

Diagrama E/R:



11.- Dar de alta un nuevo artículo de 'Primera' categoría para los fabricantes de 'FRANCIA' y abastecer con cinco unidades de ese artículo a todas las tiendas y en la fecha de hoy.

INSERT INTO ARTICULOS

```
SELECT 'Tocino', COD_FABRICANTE, 2, 'Primera', 50, 30, 100  
FROM FABRICANTES  
WHERE PAIS='FRANCIA';
```

Esta sentencia inserta 1 fila

INSERT INTO PEDIDOS

```
SELECT NIF, ARTICULO, COD_FABRICANTE, PESO, 'CATEGORIA', SYSDATE, 5  
FROM TIENDAS T, FABRICANTES F, ARTICULOS A  
WHERE F.PAIS='FRANCIA' AND A.ARTICULO='Tocino';
```

Esta sentencia inserta 6 filas

12.- Realizar una venta para todas las tiendas de 'TOLEDO' de 10 unidades en los artículos de 'Primera' categoría.

INSERT INTO VENTAS

```
SELECT NIF, ARTICULO, COD_FABRICANTE, PESO, CATEGORIA, SYSDATE, 10  
FROM ARTICULOS, TIENDAS  
WHERE CATEGORIA='Primera'  
AND PROVINCIA='TOLEDO';
```

Esta sentencia inserta 38 filas (31 + 1 +6 , esta ultimas son la insertadas en el ejercicio 11)

13.- Cambiar todos los artículos de 'Primera' categoría a 'Segunda' categoría del país

UPDATE ARTICULOS

```
SET CATEGORIA='Segunda'  
WHERE CATEGORIA='Primera' AND COD_FABRICANTE IN (SELECT COD_FABRICANTE FROM  
FABRICANTES WHERE PAIS='ITALIA');
```

Esta sentencia actualiza 2 filas

14.- Eliminar aquellas tiendas que no han realizado ventas.

DELETE FROM TIENDAS

```
WHERE NIF NOT IN (SELECT NIF FROM VENTAS);
```

Esta sentencia borra 2 filas

15.- Restar uno a las unidades de los últimos pedidos de la tienda con NIF '5555-B'.

UPDATE PEDIDOS

```
SET UNIDADES_PEDIDAS=UNIDADES_PEDIDAS-1  
WHERE NIF='5555-B'  
AND FECHA_PEDIDO=(SELECT MAX(FECHA_PEDIDO) FROM PEDIDOS WHERE NIF='5555-B');
```

16.- Para aquellos artículos de los que se hayan vendido más de 30 unidades, realizar un pedido de 10 unidades para la tienda con NIF '5555-B' con la fecha actual.

INSERT INTO PEDIDOS

```
SELECT '5555-B', ARTICULO, COD_FABRICANTE, PESO, CATEGORIA, SYSDATE, 10  
FROM VENTAS  
GROUP BY ARTICULO, COD_FABRICANTE, PESO, CATEGORIA  
HAVING SUM(UNIDADES_VENDIDAS)>30;
```

17.- Dar de alta dos tiendas en la provincia de 'SEVILLA' y abastecerlas con 30 unidades de artículos de la marca de fabricante 'GALLO'.

INSERT INTO TIENDAS

```
VALUES ('5456-C', 'Tienda Sevilla 1', 'C/Sevilla 10', 'Ajete', 'SEVILLA','56789');
```

INSERT INTO TIENDAS

```
VALUES ('5457-J', 'Tienda Sevilla 2', 'C/Sevilla 12', 'Ajete', 'SEVILLA','56789');
```

INSERT INTO PEDIDOS

```
SELECT '5456-C', ARTICULO, COD_FABRICANTE, PESO, CATEGORIA, SYSDATE, 30  
FROM ARTICULOS  
WHERE COD_FABRICANTE = (SELECT COD_FABRICANTE FROM FABRICANTES WHERE  
NOMBRE='GALLO');
```

INSERT INTO PEDIDOS

```
SELECT '5457-J', ARTICULO, COD_FABRICANTE, PESO, CATEGORIA, SYSDATE, 30  
FROM ARTICULOS  
WHERE COD_FABRICANTE = (SELECT COD_FABRICANTE FROM FABRICANTES WHERE  
NOMBRE='GALLO');
```

Como solo hay de Sevilla las dos que hemos metido también se puede hacer así:

INSERT INTO PEDIDOS

```
SELECT NIF, ARTICULO, COD_FABRICANTE, PESO, CATEGORIA, SYSDATE, 30  
FROM TIENDAS, ARTICULOS A, FABRICANTES F  
WHERE PROVINCIA='SEVILLA', AND F.NOMBRE='GALLO' AND A.COD_FABRICANTE =  
F.COD_FABRICANTE;
```

18.- Borrar los pedidos de 'Primera' categoría cuyo país de procedencia sea 'BÉLGICA'.

DELETE FROM PEDIDOS

WHERE CATEGORIA='Primera'

```
AND COD_FABRICANTE=(SELECT COD_FABRICANTE FROM FABRICANTES WHERE  
PAIS='BELGICA');
```

19.- Borrar los pedidos que no tengan tienda.

DELETE FROM PEDIDOS

```
WHERE NIF NOT IN (SELECT NIF FROM TIENDAS);
```

12.- Insertar un pedido de 20 unidades en la tienda '1111-A' con el artículo que mayor número de ventas haya realizado.

Esta sentencia actualiza 1 filas

13.- Dar de alta una tienda en la provincia de 'MADRID' y abastecerla con 20 unidades de cada uno de los artículos existentes.

17.- Cambiar los datos de la tienda con NIF '1111-A' igualándolos a los de la tienda con NIF '2222-A'.

19.- Modificar aquellos pedidos en los que la cantidad pedida sea superior a las existencias del artículo, asignando el 20 por 100 de las existencias a la cantidad que se ha pedido.

21.- Eliminar los artículos que no hayan tenido ni compras ni ventas.

Capítulo 7 - Ejercicios Propuestos.

1.- Crear las siguientes tablas de acuerdo con las restricciones que se mencionan:

Tabla ARTICULOS, TIENDAS, FABRICANTES, PEDIDOS y VENTAS

CREATE TABLE ARTICULOS2

```
(  
ARTICULO VARCHAR2 (20),  
COD_FABRICANTE NUMBER (3),  
PESO NUMBER (3),  
CATEGORIA VARCHAR2 (10),  
PRECIO_VENTA NUMBER (6,2),  
PRECIO_COSTO NUMBER (6,2),  
EXISTENCIAS NUMBER (5),  
CONSTRAINT FK_CODFAB FOREIGN KEY (COD_FABRICANTE) REFERENCES FABRICANTES2 ON DELETE  
CASCADE,  
CONSTRAINT PK_CLAVEP PRIMARY KEY (ARTICULO, COD_FABRICANTE, PESO, CATEGORIA),  
CONSTRAINT MAYORQUEO CHECK (PRECIO_VENTA>0 AND PRECIO_COSTO>0 AND PESO>0),  
CONSTRAINT CATEGORI CHECK (CATEGORIA IN ('PRIMERA', 'SEGUNDA', 'TERCERA'))  
);
```

Object Type TABLE Object ARTICULOS2

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ARTICULOS2	ARTICULO	Varchar2	20	-	-	1	-	-	-
	COD_FABRICANTE	Number	-	3	0	2	-	-	-
	PESO	Number	-	3	0	3	-	-	-
	CATEGORIA	Varchar2	10	-	-	4	-	-	-
	PRECIO_VENTA	Number	-	6	2	-	✓	-	-
	PRECIO_COSTO	Number	-	6	2	-	✓	-	-
	EXISTENCIAS	Number	-	5	0	-	✓	-	-
1 - 7									

CREATE TABLE FABRICANTES2

```
(  
COD_FABRICANTE NUMBER (3),  
NOMBRE VARCHAR2 (15),  
PAIS VARCHAR (15),  
CONSTRAINT CODFAB_PK PRIMARY KEY (COD_FABRICANTE),  
CONSTRAINT NOMBRE_MAYUSCULA CHECK (NOMBRE=UPPER(NOMBRE)),  
CONSTRAINT PAIS_MAYUSCULAS CHECK (PAIS=UPPER (PAIS))  
);
```

Object Type TABLE Object FABRICANTES2

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FABRICANTES2	COD_FABRICANTE	Number	-	3	0	1	-	-	-
	NOMBRE	Varchar2	15	-	-	-	✓	-	-
	PAIS	Varchar2	15	-	-	-	✓	-	-
1 - 3									

```
CREATE TABLE TIENDAS2
(
NIF VARCHAR2 (10),
NOMBRE VARCHAR2 (30) NOT NULL,
DIRECCION VARCHAR2 (20),
POBLACION VARCHAR2 (20),
PROVINCIA VARCHAR2 (20),
CODPOSTAL NUMBER (5),
CONSTRAINT PK_ELNIF PRIMARY KEY (NIF),
CONSTRAINT MAYUSCU CHECK (PROVINCIA = UPPER (PROVINCIA))
);
```

Object Type TABLE Object TIENDAS2

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TIENDAS2	NIF	Varchar2	10	-	-	1	-	-	-
	NOMBRE	Varchar2	30	-	-	-	-	-	-
	DIRECCION	Varchar2	20	-	-	-	✓	-	-
	POBLACION	Varchar2	20	-	-	-	✓	-	-
	PROVINCIA	Varchar2	20	-	-	-	✓	-	-
	CODPOSTAL	Number	-	5	0	-	✓	-	-

1 - 6

```
CREATE TABLE PEDIDOS2
```

```
(

NIF VARCHAR2 (10),
ARTICULO VARCHAR2 (20),
COD_FABRICANTE NUMBER (3),
PESO NUMBER (3),
CATEGORIA VARCHAR2 (10),
FECHA_PEDIDO DATE,
UNIDADES_PEDIDAS NUMBER (4),
EXISTENCIAS NUMBER (5),
CONSTRAINT FK_CODFABP FOREIGN KEY (COD_FABRICANTE) REFERENCES FABRICANTES2 ON DELETE CASCADE,
CONSTRAINT PK_CLAVEPP PRIMARY KEY (NIF, ARTICULO, COD_FABRICANTE, PESO, CATEGORIA, FECHA_PEDIDO),
CONSTRAINT FK_CLAVENIFP FOREIGN KEY (NIF) REFERENCES TIENDAS2 ON DELETE CASCADE,
CONSTRAINT FK_CODFRABP FOREIGN KEY (COD_FABRICANTE) REFERENCES FABRICANTES2 ON DELETE CASCADE,
CONSTRAINT CATEGORIP CHECK (CATEGORIA IN ('PRIMERA', 'SEGUNDA', 'TERCERA'))
);
```

Object Type TABLE Object PEDIDOS2

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PEDIDOS2	NIF	Varchar2	10	-	-	1	-	-	-
	ARTICULO	Varchar2	20	-	-	2	-	-	-
	COD_FABRICANTE	Number	-	3	0	3	-	-	-
	PESO	Number	-	3	0	4	-	-	-
	CATEGORIA	Varchar2	10	-	-	5	-	-	-
	FECHA_PEDIDO	Date	7	-	-	6	-	-	-
	UNIDADES_PEDIDAS	Number	-	4	0	-	✓	-	-
	EXISTENCIAS	Number	-	5	0	-	✓	-	-
1 - 8									

CREATE TABLE VENTAS2

```

(
NIF VARCHAR2 (10),
ARTICULO VARCHAR2 (20),
COD_FABRICANTE NUMBER (3),
PESO NUMBER (3),
CATEGORIA VARCHAR2 (10),
FECHA_VENTA DATE,
UNIDADES_VENDIDAS NUMBER (4),
CONSTRAINT PK_CLAVEPV PRIMARY KEY (NIF, ARTICULO, COD_FABRICANTE, PESO, CATEGORIA,
FECHA_VENTA),
CONSTRAINT FK_CODFABV FOREIGN KEY (COD_FABRICANTE) REFERENCES FABRICANTES2 ON
DELETE CASCADE,
CONSTRAINT VENDIDAS_MAYO CHECK (UNIDADES_VENDIDAS >0),
CONSTRAINT CATEGORIV CHECK (CATEGORIA IN ('PRIMERA', 'SEGUNDA', 'TERCERA')),
CONSTRAINT FK_CLAVEAJEV FOREIGN KEY (ARTICULO, COD_FABRICANTE, PESO, CATEGORIA)
REFERENCES ARTICULOS2 ON DELETE CASCADE,
CONSTRAINT FK_NIFV FOREIGN KEY (NIF) REFERENCES TIENDAS2 ON DELETE CASCADE
);

```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
VENTAS2	NIF	Varchar2	10	-	-	1	-	-	-
	ARTICULO	Varchar2	20	-	-	2	-	-	-
	COD_FABRICANTE	Number	-	3	0	3	-	-	-
	PESO	Number	-	3	0	4	-	-	-
	CATEGORIA	Varchar2	10	-	-	5	-	-	-
	FECHA_VENTA	Date	7	-	-	6	-	-	-
	UNIDADES_VENDIDAS	Number	-	4	0	-	✓	-	-
1 - 7									

2.- Añadir una restricción a la tabla TIENDAS para que el NOMBRE de la tienda sea de tipo título (InitCap).

ALTER TABLE TIENDAS2

ADD CONSTRAINT NOMBRETIENDAMAY CHECK (NOMBRE = INITCAP (NOMBRE));

INSERT INTO TIENDAS2

VALUES (16789654, 'romero', 'Valderejo 5', 'VIZCAYA', 'EREMUA', 56342);

```
INSERT INTO TIENDAS2
```

```
VALUES (16789654, 'romero', 'Valderejo 5', 'VIZCAYA', 'EREMUA', 56342);
```

Results Explain Describe Saved SQL History

ORA-02290: restricción de control (HR.NOMBRETIENDAMAY) violada

3.- Visualizar las constraints definidas para las tablas anteriores.

SELECT CONSTRAINT_NAME, COLUMN_NAME FROM USER_CONS_COLUMNS WHERE TABLE_NAME= 'TIENDAS2';

4.- Modificar las columnas de las tablas PEDIDOS y VENTAS para que las UNIDADES_VENDIDAS y las UNIDADES_PEDIDAS puedan almacenar cantidades numéricas de 6 dígitos.

ALTER TABLE PEDIDOS2

MODIFY (UNIDADES_PEDIDAS NUMBER (6));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PEDIDOS2	NIF	Varchar2	10	-	-	1	-	-	-
	ARTICULO	Varchar2	20	-	-	2	-	-	-
	COD_FABRICANTE	Number	-	3	0	3	-	-	-
	PESO	Number	-	3	0	4	-	-	-
	CATEGORIA	Varchar2	10	-	-	5	-	-	-
	FECHA_PEDIDO	Date	7	-	-	6	-	-	-
	UNIDADES_PEDIDAS	Number	-	6	0	-	✓	-	-
	EXISTENCIAS	Number	-	5	0	-	✓	-	-
1 - 8									

ALTER TABLE VENTAS2

MODIFY (UNIDADES_VENDIDAS NUMBER (6));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
VENTASS2	NIF	Varchar2	10	-	-	1	-	-	-
	ARTICULO	Varchar2	20	-	-	2	-	-	-
	COD_FABRICANTE	Number	-	3	0	3	-	-	-
	PESO	Number	-	3	0	4	-	-	-
	CATEGORIA	Varchar2	10	-	-	5	-	-	-
	FECHA_VENTA	Date	7	-	-	6	-	-	-
	UNIDADES_VENDIDAS	Number	-	6	0	-	✓	-	-
1 - 7									

5.- Impedir que se den de alta más tiendas en la provincia de 'TOLEDO'.

ALTER TABLE TIENDAS2

ADD CONSTRAINT PROVNOTOLEDO CHECK (PROVINCIA != 'TOLEDO')

;

6.- Añadir a las tablas PEDIDOS y VENTAS una nueva columna para que almacenen el PVP del artículo.

ALTER TABLE PEDIDOS2

ADD (PVP NUMBER (9));

ALTER TABLE VENTAS2

ADD (PVP NUMBER (9));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
VENTAS2	NIF	Varchar2	10	-	-	1	-	-	-
	ARTICULO	Varchar2	20	-	-	2	-	-	-
	COD_FABRICANTE	Number	-	3	0	3	-	-	-
	PESO	Number	-	3	0	4	-	-	-
	CATEGORIA	Varchar2	10	-	-	5	-	-	-
	FECHA_VENTA	Date	7	-	-	6	-	-	-
	UNIDADES_VENDIDAS	Number	-	6	0	-	✓	-	-
	PVP	Number	-	9	0	-	✓	-	-
1 - 8									

Tablas PERSONAL, PROFESORES y CENTROS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PERSONAL	COD_CENTRO	Number	-	4	0	-	-	-	-
	DNI	Number	-	10	0	-	✓	-	-
	APELLIDOS	Varchar2	30	-	-	-	✓	-	-
	FUNCION	Varchar2	15	-	-	-	✓	-	-
	SALARIO	Number	-	10	0	-	✓	-	-
1 - 5									

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PROFESORES	COD_CENTRO	Number	-	4	0	-	-	-	-
	DNI	Number	-	10	0	-	✓	-	-
	APELLIDOS	Varchar2	30	-	-	-	✓	-	-
	ESPECIALIDAD	Varchar2	16	-	-	-	✓	-	-
1 - 4									

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CENTROS	COD_CENTRO	Number	-	4	0	-	-	-	-
	TIPO_CENTRO	Char	1	-	-	-	✓	-	-
	NOMBRE	Varchar2	30	-	-	-	✓	-	-
	DIRECCION	Varchar2	26	-	-	-	✓	-	-
	TELEFONO	Varchar2	10	-	-	-	✓	-	-
	NUM_PLAZAS	Number	-	4	0	-	✓	-	-
1 - 6									

7.- Crear una vista que se llame CONSERJES que contenga el nombre del centro y el nombre de sus conserjes.

8.- Crear un sinónimo llamado CONSER asociado a la vista creada antes.

9.- Añadir a la tabla PROFESORES una columna llamada COD_ASIG con dos posiciones numéricas.

ALTER TABLE PROFESORES

ADD (COD_ASIG NUMBER (2));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PROFESORES	COD_CENTRO	Number	-	4	0	-	-	-	-
	DNI	Number	-	10	0	-	✓	-	-
	APELLIDOS	Varchar2	30	-	-	-	✓	-	-
	ESPECIALIDAD	Varchar2	16	-	-	-	✓	-	-
	COD_ASIG	Number	-	2	0	-	✓	-	-
1 - 5									

10.- Crear la tabla TASIG con las siguientes columnas: COD_ASIG numérico, 2 posiciones y NOM_ASIG cadena de 20 caracteres.

CREATE TABLE TASIG

(
NOM_ASIG VARCHAR2 (20),
COD_ASIG NUMBER (2)
);

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TASIG	NOM_ASIG	Varchar2	20	-	-	-	✓	-	-
	COD_ASIG	Number	-	2	0	-	✓	-	-
1 - 2									

11.- Añadir la restricción de clave primaria a la columna COD_ASIG de la tabla TASIG.

ALTER TABLE TASIG

ADD CONSTRAINT PK_CODASIG PRIMARY KEY (COD_ASIG);

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TASIG	NOM_ASIG	Varchar2	20	-	-	-	✓	-	-
	COD_ASIG	Number	-	2	0	1	-	-	-
1 - 2									

12.- Añadir la restricción de clave ajena a la columna COD_ASIG de la tabla PROFESORES.

ALTER TABLE PROFESORES

ADD CONSTRAINT FK_CODASIG FOREIGN KEY (COD_ASIG)

REFERENCES TASIG ON DELETE CASCADE;

Se pone **ON DELETE CASCADE** si queremos que se borre en las dos al actualizar.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>PROFESORES</u>	<u>COD_CENTRO</u>	Number	-	4	0	-	-	-	-
	<u>DNI</u>	Number	-	10	0	-	✓	-	-
	<u>APELLIDOS</u>	Varchar2	30	-	-	-	✓	-	-
	<u>ESPECIALIDAD</u>	Varchar2	16	-	-	-	✓	-	-
	<u>COD_ASIG</u>	Number	-	2	0	-	✓	-	-

1 - 5

13.- Visualizar los nombres de constraints y las columnas afectadas para las tablas TASIG y PROFESORES.

14.- Cambiar de nombre la tabla PROFESORES y llamarla PROFES.

15.- Borrar la tabla TASIG.

16.- Devolver la tabla PROFESORES a su situación inicial.

EJERCICIOS DEL LIBRO:

Ejercicio 4 pag.197

- 4 Crea la tabla EJEMPLO2 con las columnas DNI, NOMBRE y USUARIO; y asigna por defecto a la columna NOMBRE el literal 'No definido' y a la columna USUARIO, el número identificativo del usuario (pseudocolumna UID). Inserta una fila en la tabla dando valor solo al DNI y visualiza el contenido.

CREATE TABLE EJEMPLO2

```
(  
DNI VARCHAR2 (10) NOT NULL,  
NOMBRE VARCHAR (30) DEFAULT 'No Definido',  
USUARIO NUMBER DEFAULT UID  
);
```

Ejercicio 6 pag.199

- 6 Crea las siguientes tablas con las restricciones definidas.

Tabla FABRICANTES		Tabla ARTICULOS	
COD_FABRICANTE	NUMBER(3)	ARTICULO	VARCHAR2(20)
NOMBRE	VARCHAR2(15)	COD_FABRICANTE	NUMBER(3)
PAIS	VARCHAR2(15)	PESO	NUMBER(3)
		CATEGORIA	VARCHAR2(10)
		PRECIO_VENTA	NUMBER(6,2)
		PRECIO_COSTO	NUMBER(6,2)
		EXISTENCIAS	NUMBER(5)

Restricciones para la tabla FABRICANTES:

- La clave primaria es COD_FABRICANTE.
- Las columnas NOMBRE y PAIS han de almacenarse en mayúscula.

Restricciones para la tabla ARTICULOS:

- La clave primaria está formada por las columnas: ARTICULO, COD_FABRICANTE, PESO y CATEGORIA.
- COD_FABRICANTE es clave ajena que referencia a la tabla FABRICANTES.
- PRECIO_VENTA, PRECIO_COSTO y PESO han de ser > 0.
- CATEGORIA ha de ser 'Primera', 'Segunda' o 'Tercera'.

CREATE TABLE ARTICULOS2

```
(  
ARTICULO VARCHAR2 (20),  
COD_FABRICANTE NUMBER (3),
```

PESO NUMBER (3),
 CATEGORIA VARCHAR2 (10),
 PRECIO_VENTA NUMBER (6,2),
 PRECIO_COSTO NUMBER (6,2),
 EXISTENCIAS NUMBER (5),
CONSTRAINT FK_CODFAB FOREIGN KEY (COD_FABRICANTE) REFERENCES FABRICANTES2 ON DELETE CASCADE,
CONSTRAINT PK_CLAVEP PRIMARY KEY (ARTICULO, COD_FABRICANTE, PESO, CATEGORIA),
CONSTRAINT MAYORQUEO CHECK (PRECIO_VENTA>0 AND PRECIO_COSTO>0 AND PESO>0),
CONSTRAINT CATEGORI CHECK (CATEGORIA IN ('PRIMERA', 'SEGUNDA', 'TERCERA'))
`);`

CREATE TABLE FABRICANTES2

```
(  

  COD_FABRICANTE NUMBER (3) ,  

  NOMBRE VARCHAR2 (15) ,  

  PAIS VARCHAR (15) ,  

CONSTRAINT CODFAB_PK PRIMARY KEY (COD_FABRICANTE),  

CONSTRAINT NOMBRE_MAYUSCULA CHECK (NOMBRE=UPPER(NOMBRE)),  

CONSTRAINT PAIS_MAYUSCULAS CHECK (PAIS=UPPER (PAIS))  

);
```

Ejercicio 7 pag.207

7 Crea la tabla TIENDAS sin restricciones; la descripción es la siguiente:

NIF	VARCHAR2 (10)	Población	VARCHAR2 (20)
NOMBRE	VARCHAR2 (20)	Provincia	VARCHAR2 (20)
DIRECCION	VARCHAR2 (20)	CodPostal	NUMBER (5)

Después añade la siguientes restricciones:

- La clave primaria es NIF.
- PROVINCIA ha de almacenarse en mayúscula.
- Cambia la longitud de NOMBRE a 30 caracteres y no nulo.

CREATE TABLE TIENDAS2

```
(  

  NIF VARCHAR2 (10),  

  NOMBRE VARCHAR2 (20),  

  DIRECCION VARCHAR2 (20),  

  POBLACION VARCHAR2 (20),
```

PROVINCIA VARCHAR2 (20),
CODPOSTAL NUMBER (5)
);

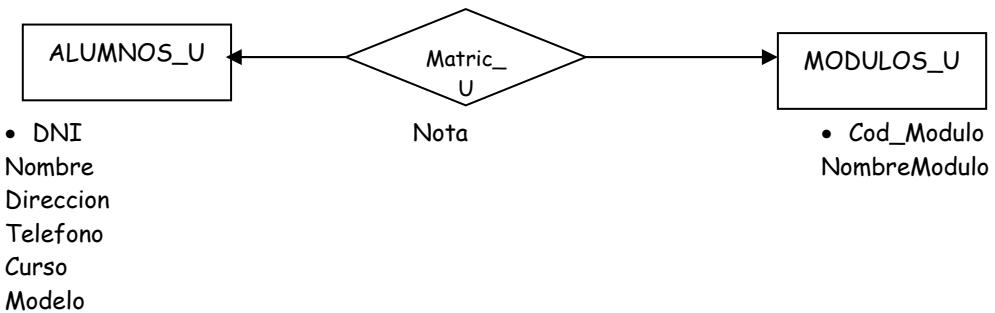
ALTER TABLE TIENDAS2
ADD CONSTRAINT PK_ELNIF PRIMARY KEY (NIF);

ALTER TABLE TIENDAS2
ADD CONSTRAINT MAYUSCU CHECK (PROVINCIA = UPPER (PROVINCIA));

ALTER TABLE TIENDAS2
MODIFY (NOMBRE VARCHAR2 (30) NOT NULL);

Ejercicio1. Matrículas

Supongamos el siguiente diagrama E/R:



1.- Crea las tablas teniendo en cuenta las características que se señalan:

a) tabla ALUMNOS_U.

- Clave Primaria (PK): DNI
- CURSO: valores 1 o 2.
- MODELO: valores A (Castellano) o D (Euskara).

ALUMNOS_U	Null?	Tipo
DNI	Not Null	VARCHAR2(9)
NOMBRE	Not Null	VARCHAR2(20)
LOCALIDAD		VARCHAR2(20)
TEL		VARCHAR2(9)
CURSO		NUMBER
MODELO		CHAR(1)

(Euskara).

b) tabla MODULOS_U.

MODULOS_U	Null?	Tipo
COD_MODULO	Not Null	NUMBER(2)
NOMBRE_MODULO		VARCHAR2(50)

- Clave Primaria (PK): Cod_Modulo

c) tabla MATRIC_U.

MATRIC_U	Null?	Tipo
DNI	Not Null	VARCHAR2(9)
COD_MODULO	Not Null	NUMBER(2)
NOTA		NUMBER(2)

- Clave Primaria (PK): DNI, Cod_Modulo
- Clave ajena: DNI => tabla ALUMNOS_U.
Cod_Modulo => tabla MODULOS_U.
- NOTA: valor 0 por defecto. Valores entre 0 y 10.

Comandos para crear las tablas:

```
CREATE TABLE ALUMNOS_U
(
DNI VARCHAR2(9) NOT NULL,
NOMBRE VARCHAR2(20) NOT NULL,
LOCALIDAD VARCHAR2(20),
TEL VARCHAR2(9),
CURSO NUMBER,
MODELO CHAR(1),
CONSTRAINT PK_DNI_A PRIMARY KEY (DNI),
CONSTRAINT CUR_A CHECK (CURSO IN (1,2)),
CONSTRAINT MOD_A CHECK (MODELO IN ('A','D'))
);
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ALUMNOS_U	DNI	Varchar2	9	-	-	1	-	-	-
	NOMBRE	Varchar2	20	-	-	-	-	-	-
	LOCALIDAD	Varchar2	20	-	-	-	✓	-	-
	TEL	Varchar2	9	-	-	-	✓	-	-
	CURSO	Number	-	-	-	-	✓	-	-
	MODELO	Char	1	-	-	-	✓	-	-

1 - 6

CREATE TABLE MODULOS_U

```
(

COD_MODULO NUMBER(2) NOT NULL,
NOMBRE_MODULO VARCHAR2(50),
CONSTRAINT PK_COD_M PRIMARY KEY (COD_MODULO)
);
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MODULOS_U	COD_MODULO	Number	-	2	0	1	-	-	-
	NOMBRE_MODULO	Varchar2	50	-	-	-	✓	-	-

1 - 2

CREATE TABLE MATRIC_U

```
(

DNI VARCHAR2(9) NOT NULL,
COD_MODULO NUMBER(2) NOT NULL,
NOTA NUMBER(2) DEFAULT 0,
CONSTRAINT PK_DNI_M PRIMARY KEY (DNI, COD_MODULO),
CONSTRAINT FK_DNI_M FOREIGN KEY (DNI) REFERENCES ALUMNOS_U ON DELETE CASCADE,
CONSTRAINT FK_COD_M FOREIGN KEY (COD_MODULO) REFERENCES MODULOS_U ON DELETE CASCADE,
CONSTRAINT NOTA_M CHECK (NOTA BETWEEN 0 AND 10)
);
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MATRIC_U	DNI	Varchar2	9	-	-	1	-	-	-
	COD_MODULO	Number	-	2	0	2	-	-	-
	NOTA	Number	-	2	0	-	✓	-	-

1 - 3

2.- Crea la tabla MODULOS_U con estos datos:

COD_MODULO	NOMBRE_MODULO
20	
25	Poner las asignaturas del ciclo
10	
15	
50	
55	
40	
45	
60	
65	

Datos tabla MODULOS_U:

01	IMPLANTACION DE SISTEMAS OPERATIVOS	cod 20
02	PLANIFICACION Y ADMINISTRACION DE REDES	cod 25
03	FUNDAMENTOS DE HARDWARE	cod 10
04	GESTION DE BASES DE DATOS	cod 15
05	LENGUAJE DE MARCAS	cod 50
06	ADMINISTRACION DE SISTEMAS OPERATIVOS	cod 55
07	SERVICIOS DE RED E INTERNET	cod 40
08	INGLES	cod 45
09	ADMINISTRACION DE SGBD	cod 60
10	SEGURIDAD	cod 65

Comandos para introducir los datos:

```
INSERT INTO MODULOS_U
VALUES (20, 'IMPLANTACION DE SISTEMAS OPERATIVOS');
```

```
INSERT INTO MODULOS_U
VALUES (25, 'PLANIFICACION Y ADMINISTRACION DE REDES');
```

```
INSERT INTO MODULOS_U
VALUES (10, 'FUNDAMENTOS DE HARDWARE');
```

```
INSERT INTO MODULOS_U
VALUES (15, 'GESTION DE BASES DE DATOS');
```

```
INSERT INTO MODULOS_U
VALUES (50, 'LENGUAJE DE MARCAS');
```

```
INSERT INTO MODULOS_U
VALUES (55, 'ADMINISTRACION DE SISTEMAS OPERATIVOS');
```

```
INSERT INTO MODULOS_U
VALUES (40, 'SERVICIOS DE RED E INTERNET');
```

```
INSERT INTO MODULOS_U
VALUES (45, 'INGLES');
```

```
INSERT INTO MODULOS_U
VALUES (60, 'ADMINISTRACION DE SGBD');
```

```
INSERT INTO MODULOS_U
VALUES (65, 'SEGURIDAD');
```

Vista de los datos introducidos:

```
SELECT * FROM MODULOS_U;
```

COD_MODULO	NOMBRE_MODULO
10	FUNDAMENTOS DE HARDWARE
20	IMPLANTACION DE SISTEMAS OPERATIVOS
25	PLANIFICACION Y ADMINISTRACION DE REDES
15	GESTION DE BASES DE DATOS
50	LENGUAJE DE MARCAS
55	ADMINISTRACION DE SISTEMAS OPERATIVOS
40	SERVICIOS DE RED E INTERNET
45	INGLES
60	ADMINISTRACION DE SGBD
65	SEGURIDAD

- Los datos de la tabla ALUMNOS_U se crearán ejecutando el fichero ALUMNOSCICLO.SQL.

Estos son los datos:

DNI	NOMBRE	LOCALIDAD	TEL	CURSO	MODELO
12345678A	Juan Ramirez	Bilbao	654345678	2	D
12342278A	Jose Valenciano	San Ignacio	675894567	2	D
12342278B	Unai Rico	Algorta	666453212	1	D
12342278C	Juan Carlos Perrez	Getxo	685342312	2	A
12342278D	Maialen Saenz	Astrabudua	645342354	2	A
12342278E	Jon Ander Lopez	Plencia	634562398	1	D
12342278F	Diego Freijo	Berango	647234512	2	D
12342278G				1	A
12342278H				2	D
12342278I				1	A
12342278J				2	A
12342278K				1	D
12342278L				1	D
12342278M				1	A
12342278N				2	D
12342278P				2	A
12342278Q				2	A
12342278R				2	A
12342278S				1	D
12342278T				1	D
12342278U				2	D
12342278V				2	D
12342278X				1	D
12342278Y				1	A
12342278Z				2	A
12342271A				2	A
12342272A				1	D
12342273A				1	A
12342274A				1	A
12342275A				1	A
12342276A				1	A

Instrucciones para insertar los datos:

INSERT INTO ALUMNOS_U

VALUES ('12345678A', 'Juan Ramirez', 'Bilbao', 654345678, 2, 'D');

INSERT INTO ALUMNOS_U

VALUES ('12342278A', 'Jose Valenciano', 'San Ignacio', 675894567, 2, 'D');

INSERT INTO ALUMNOS_U

VALUES ('12342278B', 'Unai Rico', 'Algorta', 666453212, 1, 'D');

INSERT INTO ALUMNOS_U

VALUES ('12342278C', 'Juan Carlos Perrez', 'Getxo', 685342312, 2, 'A');

INSERT INTO ALUMNOS_U

VALUES ('12342278D', 'Maialen Saenz', 'Astrabudua', 645342354, 2, 'A');

INSERT INTO ALUMNOS_U

VALUES ('12342278E', 'Jon Ander Lopez', 'Plencia', 634562398, 1, 'D');

INSERT INTO ALUMNOS_U

VALUES ('12342278F', 'Diego Freijo', 'Berango', 647234512, 2, 'D');

INSERT INTO ALUMNOS_U

VALUES ('12342273F', 'Juan Navarro', 'Leioa', 647234592, 1, 'D');

INSERT INTO ALUMNOS_U

VALUES ('12942273F', 'Pedro Aznar', 'Bilbao', 657232592, 1, 'A');

Vista de los datos introducidos:

SELECT * FROM ALUMNOS_U ;

DNI	HOMBRE	LOCALIDAD	TEL	CURSO	MODELO
12345678A	Juan Ramirez	Bilbao	654345678	2	D
12342278A	Jose Valenciano	San Ignacio	675894567	2	D
12342278B	Unai Rico	Algorta	666453212	1	D
12342278C	Juan Carlos Perrez	Getxo	685342312	2	A
12342278D	Maialen Saenz	Astrabudua	645342354	2	A
12342278E	Jon Ander Lopez	Plencia	634562398	1	D
12342278F	Diego Freijo	Berango	647234512	2	D
12342273F	Juan Navarro	Leioa	647234592	1	D
12942273F	Pedro Aznar	Bilbao	657232592	1	A

- Invéntate algunos datos para la tabla MATRIC_U. Incluye algunos suspensos.

Instrucciones para insertar los datos:

INSERT INTO MATRIC_U

VALUES ('12342273F', 15, 7);

INSERT INTO MATRIC_U

VALUES ('12342273F', 25, 6);

```
INSERT INTO MATRIC_U
VALUES ('12342273F', 55, 9);
```

```
INSERT INTO MATRIC_U
VALUES ('12345678A', 10, 5);
```

```
INSERT INTO MATRIC_U
VALUES ('12345678A', 40, 9);
```

```
INSERT INTO MATRIC_U
VALUES ('12342278A', 20, 4);
```

```
INSERT INTO MATRIC_U
VALUES ('12342278A', 50, 4);
```

```
INSERT INTO MATRIC_U
VALUES ('12342278B', 25, 8);
```

```
INSERT INTO MATRIC_U
VALUES ('12342278B', 15, 5);
```

```
INSERT INTO MATRIC_U
VALUES ('12342278C', 15, 3);
```

```
INSERT INTO MATRIC_U
VALUES ('12342278C', 10, 5);
```

```
INSERT INTO MATRIC_U
VALUES ('12342278C', 50, 5);
```

```
INSERT INTO MATRIC_U
VALUES ('12342278D', 50, 5);
```

```
INSERT INTO MATRIC_U
VALUES ('12342278E', 55, 5);
```

```
INSERT INTO MATRIC_U
VALUES ('12342278F', 40, 6);
```

```
INSERT INTO MATRIC_U
VALUES ('12342278F', 55, 9);
```

```
INSERT INTO MATRIC_U
VALUES ('12342278F', 10, 8);
```

```
INSERT INTO MATRIC_U
VALUES ('12345678A', 45, 9);
```

```
INSERT INTO MATRIC_U
VALUES ('12342278D', 60, 4);
```

```
INSERT INTO MATRIC_U
VALUES ('12342278D', 55, 7);
```

```
INSERT INTO MATRIC_U
VALUES ('12942273F', 50, 4);
```

Vista de los datos introducidos:

SELECT * FROM MATRIC_U ;

DNI	COD_MODULO	NOTA
12342273F	15	7
12342273F	25	6
12342273F	55	9
12345678A	10	5
12345678A	40	9
12342278A	20	4
12342278A	50	4
12342278B	25	8
12342278B	15	5
12342278C	15	3
12342278C	10	5
12342278C	50	5
12342278D	50	5
12342278E	55	5
12342278F	40	6
12342278F	55	9
12342278F	10	8
12345678A	45	9
12342278D	60	4
12342278D	55	7
12942273F	50	4

3.- Crea las sentencias SQL para obtener la información de cada ítem:

a) La lista de todos los alumnos, ordenados por curso y modelo.

**SELECT * FROM ALUMNOS_U
ORDER BY CURSO, MODELO ;**

DNI	NOMBRE	LOCALIDAD	TEL	CURSO	MODELO
12942273F	Pedro Aznar	Bilbao	657232592	1	A
12342278E	Jon Ander Lopez	Plencia	634562398	1	D
12342278B	Unai Rico	Algorta	666453212	1	D
12342273F	Juan Navarro	Leioa	647234592	1	D
12342278C	Juan Carlos Perrez	Getxo	685342312	2	A
12342278D	Maialen Saenz	Astrabudua	645342354	2	A
12342278F	Diego Freijo	Berango	647234512	2	D
12345678A	Juan Ramirez	Bilbao	654345678	2	D
12342278A	Jose Valenciano	San Ignacio	675894567	2	D

b) El número de alumnos de cada grupo (1A, 1D, 2A y 2D).

**SELECT CURSO, MODELO, COUNT(*) "Nº ALUMNOS"
FROM ALUMNOS_U
GROUP BY CURSO, MODELO ;**

CURSO	MODELO	Nº ALUMNOS
2	D	3
1	A	1
2	A	2
1	D	3

c) ¿Cuál es el grupo con el mayor número de alumnos?

```
SELECT CURSO, MODELO, COUNT(*) "Nº ALUMNOS"
```

```
FROM ALUMNOS_U
```

```
GROUP BY CURSO, MODELO
```

```
HAVING COUNT(*) = (SELECT MAX(COUNT(*)) FROM ALUMNOS_U GROUP BY CURSO, MODELO);
```

CURSO	MODELO	Nº ALUMNOS
2	D	3
1	D	3

d) Este se hace al final.

e) Calcula el número de alumnos matriculado en cada uno de los módulos.

```
SELECT NOMBRE_MODULO "MODULO", COUNT(*)
```

```
FROM MATRIC_U M , MODULOS_U N
```

```
WHERE M.COD_MODULO=N.COD_MODULO
```

```
GROUP BY NOMBRE_MODULO ;
```

MODULO	COUNT(*)
PLANIFICACION Y ADMINISTRACION DE REDES	2
SERVICIOS DE RED E INTERNET	2
GESTION DE BASES DE DATOS	3
ADMINISTRACION DE SISTEMAS OPERATIVOS	4
ADMINISTRACION DE SGBD	1
INGLES	1
IMPLANTACION DE SISTEMAS OPERATIVOS	1
LENGUAJE DE MARCAS	4
FUNDAMENTOS DE HARDWARE	3

f) Calcula la nota media de todos los alumnos matriculados en 1º.

```
SELECT NOMBRE_MODULO "MODULO", M.COD_MODULO, AVG(NOTA) "NOTA MEDIA"
```

```
FROM MATRIC_U M , MODULOS_U N , ALUMNOS_U A
```

```
WHERE M.COD_MODULO=N.COD_MODULO AND M.DNI=A.DNI AND CURSO=1
```

```
GROUP BY NOMBRE_MODULO, M.COD_MODULO ;
```

MODULO	COD_MODULO	NOTA MEDIA
ADMINISTRACION DE SISTEMAS OPERATIVOS	55	7
GESTION DE BASES DE DATOS	15	6
LENGUAJE DE MARCAS	50	4
PLANIFICACION Y ADMINISTRACION DE REDES	25	7

g) Calcula el número de aprobados en cada uno de los módulos.

```
SELECT NOMBRE_MODULO, COUNT(*)
FROM MATRIC_U M, MODULOS_U N
WHERE M.COD_MODULO=N.COD_MODULO
AND NOTA>=5
GROUP BY NOMBRE_MODULO ;
```

NOMBRE_MODULO	COUNT(*)
PLANIFICACION Y ADMINISTRACION DE REDES	2
SERVICIOS DE RED E INTERNET	2
GESTION DE BASES DE DATOS	2
ADMINISTRACION DE SISTEMAS OPERATIVOS	4
INGLES	1
FUNDAMENTOS DE HARDWARE	3
LENGUAJE DE MARCAS	2

h) Calcula el porcentaje de aprobados en cada uno de los módulos.

```
CREATE VIEW "APROBADOS"
```

```
AS SELECT *
FROM MATRIC_U
WHERE NOTA>=5 ;
```

```
SELECT MA.COD_MODULO, (TO_CHAR (COUNT(A.NOTA)/COUNT(MA.NOTA)*100,'990D99') || '%')
"PORCENTAJE APROBADOS"
FROM MATRIC_U MA, APROBADOS A
WHERE A.COD_MODULO(+) = MA.COD_MODULO
AND A.DNI(+) = MA.DNI
GROUP BY MA.COD_MODULO ;
```

COD_MODULO	PORCENTAJE APROBADOS
25	100,00%
20	0,00%
55	100,00%
40	100,00%
50	50,00%
45	100,00%
15	66,67%
10	100,00%
60	0,00%

d) Ponles un 7 a todos los alumnos matriculados en 'Base de Datos' y 'Redes'.

Alumnos que cumplen alguna de las condiciones: **Hacerlo al final después del h)**

```
SELECT * FROM MATRIC_U WHERE COD_MODULO=15 OR COD_MODULO=25;
```

DNI	COD_MODULO	NOTA
12342273F	15	7
12342273F	25	6
12342278B	25	8
12342278B	15	5
12342278C	15	3

Como se puede observar hay tres alumnos que cumplen la condición indicada, DNI: 12342273F, DNI: 12342278B y DNI: 12342278C.

Necesitamos los códigos para Bases de Datos y Redes:

`SELECT COD_MODULO FROM MODULOS_U WHERE NOMBRE_MODULO = 'GESTION DE BASES DE DATOS';`

`SELECT COD_MODULO FROM MODULOS_U WHERE NOMBRE_MODULO = 'PLANIFICACION Y ADMINISTRACION DE REDES';`

COD_MODULO
15

COD_MODULO
25

`SELECT COD_MODULO FROM MODULOS_U
WHERE NOMBRE_MODULO= 'GESTION DE BASES DE DATOS' OR NOMBRE_MODULO= 'PLANIFICACION Y ADMINISTRACION DE REDES';`

COD_MODULO
25
15

La instrucción que responde a la pregunta es:

`UPDATE MATRIC_U`

`SET NOTA=7`

`WHERE COD_MODULO IN (SELECT COD_MODULO FROM MODULOS_U WHERE NOMBRE_MODULO= 'GESTION DE BASES DE DATOS' OR NOMBRE_MODULO= 'PLANIFICACION Y ADMINISTRACION DE REDES') ;`

5 row(s) updated.

`SELECT * FROM MATRIC_U WHERE COD_MODULO=15 OR COD_MODULO=25;`

DNI	COD_MODULO	NOTA
12342273F	15	7
12342273F	25	7
12342278B	25	7
12342278B	15	7
12342278C	15	7

Otros ejercicios. Ejercicio2. Gimnasio

Tenemos una BD en la que se refleja la situación de un gimnasio donde los usuarios pagan las distintas actividades que realizan por medio de un recibo mensual por el banco. Además se contemplan también las posibles relaciones de parentesco o amistad entre los socios. La BD cuenta con las siguientes tablas:

- La tabla de BANCOS contiene una fila por cada uno de los bancos:

```
CREATE TABLE BANCOS
( ENT_SUC          NUMBER(8) NOT NULL,
  NOMBRE           VARCHAR2(50),
  DIRECCION        VARCHAR2(50),
  LOCALIDAD        VARCHAR2(30),
  TELEFONOS        VARCHAR2(30),
  CONSTRAINT CLAVE_PRIMARIA_BANCOS PRIMARY KEY(ENT_SUC));
```

- La tabla de USUARIOS contiene una fila por cada usuario:

```
CREATE TABLE USUARIOS
( NUM_SOCIO        VARCHAR2(7) NOT NULL,
  DNI              VARCHAR2(9) NOT NULL,
  NOMBRE           VARCHAR2(20),
  APELLIDOS        VARCHAR2(30),
  FOTOGRAFIA       LONG RAW,
  DOMICILIO        VARCHAR2(40),
  LOCALIDAD        VARCHAR2(50),
  CP               VARCHAR2(5),
  FECHA_NACIMIENTO DATE,
  TELEFONO         VARCHAR2(20),
  TAQUILLA         VARCHAR2(15),
  HORARIO          VARCHAR2(15),
  FECHA_ALTA       DATE,
  FECHA_BAJA       DATE,
  CUOTA_SOCIO      NUMBER(7),
  CUOTA_FAMILIAR   NUMBER(7),
  PAGA_BANCO        CHAR NOT NULL
  CONSTRAINT CHEQUEO1 CHECK (PAGA_BANCO IN ('S','N')),
  CODIGO_BANCO      NUMBER(8),
  CUENTA            NUMBER(10),
  DIGITO_CONTROL    NUMBER(2),
  OBSERVACIONES     VARCHAR2(500),
  CONSTRAINT CLAVE_PRIMARIA_USUARIOS PRIMARY KEY (NUM_SOCIO),
  CONSTRAINT CLAVE_ALTERNATIVA_USUARIOS UNIQUE (DNI),
  CONSTRAINT CLAVE_AJENA_BANCOS FOREIGN KEY (CODIGO_BANCO)
  REFERENCES BANCOS(ENT_SUC));
```

- La tabla de ACTIVIDADES contiene una fila por cada actividad:

```
CREATE TABLE ACTIVIDADES
( CODIGO_ACTIVIDAD      VARCHAR2(7) NOT NULL,
  DESCRIPCION           VARCHAR2(50),
  CUOTA                 NUMBER(7),
  CONSTRAINT CLAVE_PRIMARIA_ACTIVIDADES PRIMARY KEY (CODIGO_ACTIVIDAD));
```

- La tabla de ACTIVIDADES_USUARIOS contiene una fila por cada actividad que realiza un usuario:

```
CREATE TABLE ACTIVIDADES_USUARIOS
( CODIGO_ACTIVIDAD      VARCHAR2(7) NOT NULL,
  CODIGO_USUARIO         VARCHAR2(7) NOT NULL,
  FECHA_ALTA             DATE,
  FECHA_BAJA              DATE,
  CONSTRAINT CLAVE_PRIMARIA_ACT_USU
    PRIMARY KEY(CODIGO_ACTIVIDAD,CODIGO_USUARIO),
  CONSTRAINT CLAVE_AJENA_ACT FOREIGN KEY (CODIGO_ACTIVIDAD)
    REFERENCES ACTIVIDADES(CODIGO_ACTIVIDAD),
  CONSTRAINT CLAVE_AJENA_USU FOREIGN KEY (CODIGO_USUARIO)
    REFERENCES USUARIOS(NUM_SOCIO) );
```

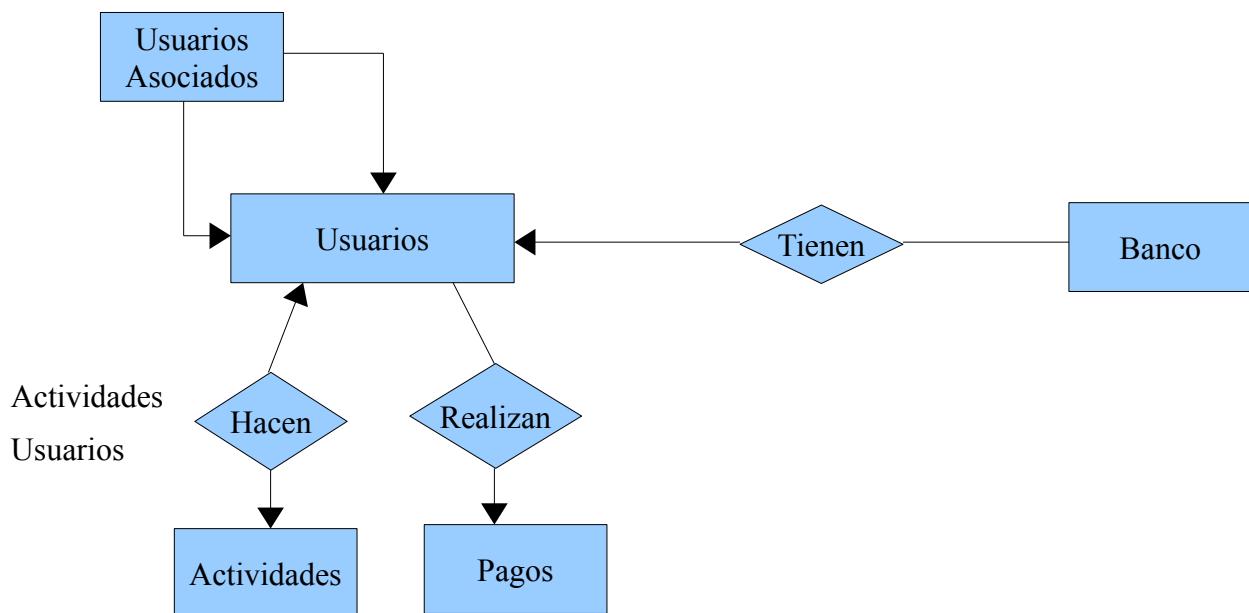
- La tabla de PAGOS contiene una fila por cada pago mensual de cada usuario:

```
CREATE TABLE PAGOS
( CODIGO_USUARIO  VARCHAR2(7) NOT NULL,
  NUMERO_MES       NUMBER(2) NOT NULL,
  CUOTA            NUMBER(7),
  OBSERVACIONES   VARCHAR2(500),
  CONSTRAINT CLAVE_PRIMARIA_PAGOS PRIMARY KEY(CODIGO_USUARIO, NUMERO_MES),
  CONSTRAINT CLAVE_AJENA_PAG_USU FOREIGN KEY(CODIGO_USUARIO)
    REFERENCES USUARIOS(NUM_SOCIO) ON DELETE CASCADE);
```

- La tabla de USUARIOS_ASOCIADOS contiene una fila por usuario que tiene relación con otro:

```
CREATE TABLE USUARIOS_ASOCIADOS
( CODIGO_USUARIO      VARCHAR2(7) NOT NULL,
  USUARIO_ASOCIADO    VARCHAR2(7) NOT NULL,
  CONSTRAINT CLAVE_PRIMARIA_USUARIOS_ASOC
    PRIMARY KEY(CODIGO_USUARIO, USUARIO_ASOCIADO),
  CONSTRAINT CLAVE_AJENA_USU_USU FOREIGN KEY(CODIGO_USUARIO)
    REFERENCES USUARIOS(NUM_SOCIO),
  CONSTRAINT CLAVE_AJENA_USU_ASOC FOREIGN KEY(USUARIO_ASOCIADO)
    REFERENCES USUARIOS(NUM_SOCIO));
```

1.- Crea el modelo E/R que les corresponde a las especificaciones.



2.- Ejecuta el fichero GYM.SQL para crear las tablas en tu ordenador.

CONNECT HR/HR
START GYM.SQL

3.- Ejecuta las sentencias SQL para obtener los siguientes datos:

a.- Selecciona las actividades cuya cuota es superior al 15% de la media de las cuotas de los usuarios que no pagan mediante banco.

SELECT CODIGO_ACTIVIDAD "Codigo",DESCRIPCION "Actividad"

FROM ACTIVIDADES

WHERE CUOTA>(SELECT AVG (CUOTA_SOCIO)*0.15 FROM USUARIOS WHERE PAGA_BANCO='N');

Codigo	Actividad
AM00001	JUDO
AM00002	KARATE
G000003	MUSCULACION

b.- Multiplica por 3 las cuotas de todas las actividades.

CODIGO_ACTIVIDAD	DESCRIPCION	CUOTA
G000001	GIMNASIA DE MANTENIMIENTO	1000
G000002	GIMNASIA RITMICA	800
AM00001	JUDO	1200
AM00002	KARATE	1100
N000001	NATACION 1	900
N000002	NATACION 2	1000
G000003	MUSCULACION	1300

UPDATE ACTIVIDADES
SET CUOTA = CUOTA*3 ;

7 row(s) updated.

CODIGO_ACTIVIDAD	DESCRIPCION	CUOTA
G000001	GIMNASIA DE MANTENIMIENTO	3000
G000002	GIMNASIA RITMICA	2400
AM00001	JUDO	3600
AM00002	KARATE	3300
N000001	NATACION 1	2700
N000002	NATACION 2	3000
G000003	MUSCULACION	3900

c.- Selecciona las actividades cuya cuota es igual a alguna de las cuotas de los usuarios.

**SELECT CODIGO_ACTIVIDAD "CODIGO", DESCRIPCION "ACTIVIDAD"
FROM ACTIVIDADES
WHERE CUOTA IN (SELECT CUOTA_SOCIO FROM USUARIOS) :**

CODIGO	ACTIVIDAD
G000001	GIMNASIA DE MANTENIMIENTO
N000002	NATACION 2

d.- Selecciona las actividades cuya cuota es inferior a la cuota de socio mínima de entre todos los usuarios.

**SELECT CODIGO_ACTIVIDAD "CODIGO", DESCRIPCION "ACTIVIDAD"
FROM ACTIVIDADES
WHERE CUOTA < (SELECT MIN (CUOTA_SOCIO) FROM USUARIOS) ;**

CODIGO	ACTIVIDAD
G000002	GIMNASIA RITMICA
N000001	NATACION 1

e.- Selecciona los nombres y el número de socio de los usuarios cuyas cuotas de socio sean inferiores al total de pagos realizados.

**SELECT NOMBRE "Nombre", NUM_SOCIO "Nº Socio"
FROM USUARIOS
WHERE CUOTA_SOCIO < (SELECT SUM (CUOTA) FROM PAGOS WHERE USUARIOS.NUM_SOCIO =
PAGOS.CODIGO_USUARIO) ;**

Nombre	Nº Socio
JUAN LUIS	A1111
ANA	J1111
INES	A2222
MARIA	J3333
LAURA	J2222
MARTA	I2222
MARIA	I1111

f.- Crea una vista que muestre el nombre de cada usuario, la descripción de las actividades en que participa y la fecha de alta en esa actividad.

```
CREATE VIEW NOMBRE_USUARIO
AS SELECT U.NOMBRE, A.DESCRIPCION, AU.FECHA_ALTA FROM USUARIOS U,
ACTIVIDADES_USUARIOS AU, ACTIVIDADES A WHERE U.NUM_SOCIO = AU.CODIGO_USUARIO
AND AU.CODIGO_USUARIO = A.CODIGO_ACTIVIDAD ;
```

View created.

g.- Crea una vista con los socios que no tengan cuota nula.

```
CREATE VIEW USUCUOTANONULA
AS SELECT NOMBRE, APELLIDOS
FROM USUARIOS
WHERE CUOTA_SOCIO IS NOT NULL ;
```

h.- Crea una vista con las actividades cuya cuota sea superior a 3500.

```
CREATE VIEW ACTIVIDADCUOTA3500
AS SELECT CODIGO_ACTIVIDAD, DESCRIPCION
FROM ACTIVIDADES
WHERE CUOTA > 3500 ;
```

i.- Crea una vista que muestre los códigos de los bancos y la suma de las cuotas para cada banco.

```
CREATE VIEW CODIBANK_CUOTASUM
AS SELECT CODIGO_BANCO, SUM(CUOTA_SOCIO)
FROM USUARIOS
GROUP BY CODIGO_BANCO ;
```

j.- Actualiza en un 10% la cuota de socio de aquellos usuarios que estén inscritos en la actividad de 'Gimnasia de Mantenimiento'.

```
UPDATE USUARIOS
SET CUOTA_SOCIO = (CUOTA_SOCIO*0.10)+CUOTA_SOCIO
WHERE NUM_SOCIO IN (SELECT CODIGO_USUARIO
FROM ACTIVIDADES_USUARIOS AU, ACTIVIDADES A
WHERE AU.CODIGO_ACTIVIDAD = A.CODIGO_ACTIVIDAD
AND DESCRIPCION = 'GIMNASIA DE MANTENIMIENTO') ;
1 row(s) updated.
```

k.- Elimina los usuarios que practican 'Natacion 2'.

```
DELETE FROM USUARIOS
WHERE NUM_SOCIO IN (SELECT CODIGO_USUARIO
FROM ACTIVIDADES_USUARIOS AU, ACTIVIDADES A
WHERE AU.CODIGO_ACTIVIDAD = A.CODIGO_ACTIVIDAD
AND DESCRIPCION = 'NATACION 2') ;
```

ORA-02292: restriccin de integridad (HR.CLAVE_AJENA_USU) violada - registro secundario encontrado

Nota: Reflexiva en Entidad usuario (realiza actividades)

Datos de las tablas:**BANCOS**

ENT_SUC Number(8)	NOMBRE Varchar2(50)	DIRECCION Varchar2(50)	LOCALIDAD Varchar2(30)	TELEFONOS Varchar2(30)
30700018	BANESTO	MANUEL LLANEZA, 33	MIERES	
20480070	CAJA DE ASTURIAS	MANUEL LLANEZA, 17	MIERES	
43000250	HERRERO	MANUEL LLANEZA, 22	MIERES	
85002222	SANTANDER	LA CAMARA, 13	AVILES	
22223333	BBV	LA RIBERA, 17	LUANCO	
33334444	ATLANTICO	GIJON, 56	LUANCO	

ACTIVIDADES

CODIGO_ ACTIVIDAD Varchar2(7)	DESCRIPCION Varchar2(50)	CUOTA Number(7)
G000001	GIMNASIA DE MANTENIMIENTO	1000
G000002	GIMNASIA RITMICA	800
AM00001	JUDO	1200
AM00002	KARATE	1100
N000001	NATACION 1	900
N000002	NATACION 2	1000
G000003	MUSCULACION	1300

ACTIVIDADES_USUARIOS

CODIGO_ ACTIVIDAD Varchar2(7)	CODIGO_USUARIO Varchar2(7)	FECHA_ ALTA Date	FECHA_ BAJA Date
G000001	A1111	30/03/1999	
AM00002	A1111	30/03/1999	
AM00001	A2222	30/03/1999	
N000001	A2222	30/03/1999	
N000002	I2222	01/03/1999	
AM00002	I2222	11/02/19999	

PAGOS

CODIGO_USUARIO Varchar2(7)	NUMERO_MES Number(2)	CUOTA Number(7)	OBSERVACIONES Varchar2(500)
A1111	1	5000	
A1111	2	5000	
A1111	3	5000	
A1111	4	5500	
A1111	5	5500	
A1111	6	5000	
A2222	1	7800	
A2222	2	7800	
A2222	5	7800	
A2222	6	8000	
A2222	3	7800	
A2222	4	9000	
I1111	5	4500	
I1111	7	5000	
I1111	1	3456	
A2222	7	8000	
A3333	2	4500	
I2222	1	5000	
I2222	2	5000	
I2222	3	5500	
I2222	4	5500	
I2222	6	6000	
J1111	1	1000	
J1111	2	10000	
J1111	4	10000	
J1111	5	10000	
J1111	12	10000	
J2222	12	8000	
J2222	1	1000	
J2222	3	10000	
J2222	11	10000	
J3333	1	1000	
J3333	2	2000	
J3333	3	3000	
J3333	4	4000	
J3333	5	500	
J3333	11	800	
J3333	12	1500	

USUARIOS

NUM_SOCIO Varchar(7)	DNI Varchar2 (9)	NOMBRE Varchar2(20)	APELLIDOS Varchar2(30)	DOMICILIO Varchar2 (40)	LOCALIDAD Varchar2 (50)	CP Varchar2 (5)	FECHA_ ALTA Date	CUOTA_ SOCIO N(7)	PAGA_ BANCO Char	CODIGO_ BANCO Number(8)	CUENTA Number (10)	DC Number (2)
A1111	01111111	JUAN LUIS	ARIAS ALVAREZ	C/LA VEGA, 18	MIERES	33600	10/01/1996	5500	S	30700018	1111	10
A2222	02222222	INES	PEREZ DIAZ	C/DR. FLEMING, 14	MIERES	33600	10/01/1996	6800	S	20480070	2222	47
A3333	03333333	JOSE	RUTZ PEÑA	C/AVTLES, 18	LUANCO	33400	21/05/1996	6000	N			
I1111	00111111	MARIA	EGIA SANTAMARINA	C/GERNIKA, 3	MIERES	33600	10/11/1998	7000	S	43000250	123456	21
I22222	77896542	MARTA	ARIAS SANTOLAYA	C/ASTURIAS, 23 4º	MIERES	33600	12/12/1997	4500	S	20480070	2242	41
J1111	11111111	ANA	GUTIERREZ ALONSO	C/ASTURIAS, 51 4º	LUANCO	33440	02/03/1998	5000	S	85002222	2342	61
J22222	22222222	LAURA	FERNANDEZ ALONSO	C/TEVERGA, 17 2º	LUANCO	33440	11/12/1997	5000	S	85002222	234255	34
J33333	33333333	MARIA	ALONSO GUTIERREZ	C/OVIEDO, 8 1º	LUANCO	33440	10/11/1998	6000	S	22223333	425566	43
J44444	44444444	MANUEL	ALONSO OVIES	C/ OVIEDO, 18 4º	LUANCO	33440	05/06/1996	6000	S	22223333	425566	43
J55555	55555555	RAMON	ARBOLEYA GARCIA	C/LANGREO, 9 1º	AVILES	33400	03/04/1995	4000	S	85002222	4343	12
J66666	66666666	DOLORES	MORENO RODRIGUEZ	C/OVIEDO, 23 6º	AVILES	33400	03/03/1998	5	S	22223333	6675	21
J77777	77777777	PABLO	RODRIGUEZ ARIAS	C/LA FLORIDA, 3 6º	AVILES	33400	06/09/1997	4000	S	33334444	6775	12
J88888	88888888	MARTA	ARRIEN GONZALEZ	PLAZA SAN JUAN, 9	MIERES	33600	06/09/1997	7000	S	33334444	9975	11
J99999	99999999	LUIS	BULNES BALBIN	C/LA VEGA, 49	MIERES	33600	14/11/1996	7000	S	33334444	39975	22
J1010	10101010	JOSE	ALVAREZ CASTRO	C/ONON, 23 2º	MIERES	33600	01/04/1995	6000	S	33334444	933375	42
J0011	11011011	PELAYO	ESLA CASARIEGO	C/LA PISTA, 14 1º	MIERES	33600	01/05/1997	8000	N			
J0012	12121212	VICTOR	ALBA PRIETO	C/LA LILA, 49 2º	AVILES	33400	01/05/1998	8000	S	85002222	54679	32
J003	13131313	LUZ	CUETO ARROYO	C/MAYOR, 91 5º	AVILES	33400	01/06/1995	7000	S	22223333	66785	32
J004	14141414	MARIO	FERNANDEZ VEGA	C/PEZ, 19 2º	AVILES	33400	01/04/1998	3000	S	33334444	33554679	53

3. ARIKETA -SQL-

En este ejercicio trabajaremos con 5 tablas: PRODUCTOS, OFICINAS, CLIENTES, TRABAJADORES, PEDIDOS.

La información que tenemos en cada tabla es la siguiente:

a) Tabla OFICINAS.

Nombre Columna	Null?	Tipo
OFICINA	NOT NULL	NUMBER(2)
CIUDAD		VARCHAR2(15)
REGION		VARCHAR2(10)
DIR		NUMBER(3)
OBJETIVO		NUMBER(10)
VENTAS		NUMBER(10)

CLAVE PRIMARIA: OFICINA

b) Tabla CLIENTES.

Nombre Columna	Null?	Tipo
NUMCLIE	NOT NULL	NUMBER(4)
NOMBRE		VARCHAR2(20)
NUMEMP		NUMBER(3)
LIMITECREDITO		NUMBER(10)

CLAVE PRIMARIA: NUMCLIE

CLAVE FORÁNEA: NUMEMP → TRABAJADORES

c) Tabla PEDIDOS.

Nombre Columna	Null?	Tipo
CODIGO	NOT NULL	NUMBER(3)
NUMPEDIDO	NOT NULL	NUMBER(9)
FECHAPEDIDO		DATE
NUMCLIE	NOT NULL	NUMBER(4)
NUMEMP	NOT NULL	NUMBER(3)
IDFAB	NOT NULL	VARCHAR2(10)
IDPRODUCTO	NOT NULL	VARCHAR2(15)
CANT		NUMBER(4)

CLAVE PRIMARIA: CODIGO

CLAVE FORÁNEA: NUMCLIE → Tabla CLIENTES

NUMEMP → Tabla TRABAJADORES

IDFAB + IDPRODUCTO → Tabla PRODUCTOS

d) Tabla TRABAJADORES - EMPLEADOS

Nombre Columna	Null?	Tipo
NUMEMP	NOT NULL	NUMBER(3)
NOMBRE		VARCHAR2(20)
EDAD		NUMBER(2)
OFICINA		NUMBER(2)
TITULO		VARCHAR2(15)
CONTRATO		DATE
JEFE		NUMBER(3)
CUOTA		NUMBER(10)
VENTAS		NUMBER(10)

CLAVE PRIMARIA: NUMEMP

CLAVE FORÁNEA: OFICINA → OFICINAS

18 <= EDAD <=70

e) Tabla PRODUCTOS.

Nombre Columna	Null?	Tipo
IDFAB	NOT NULL	VARCHAR2(10)
IDPRODUCTO	NOT NULL	VARCHAR2(15)
DESCRIPCION		VARCHAR2(20)
PRECIO		NUMBER(10)
EXISTENCIAS		NUMBER(5)

CLAVE PRIMARIA: IDFAB + IDPRODUCTO

1. Con toda la información necesaria para crear estas 5 tablas, crea un fichero SQL, con el nombre CREA2.SQL.

CREATE TABLE PRODUCTOS

```
(  
IDFAB VARCHAR2(10) NOT NULL,  
IDPRODUCTO VARCHAR2(15) NOT NULL,  
DESCRIPCION VARCHAR2(20),  
PRECIO NUMBER(10),  
EXISTENCIAS NUMBER(5),  
CONSTRAINT PK_IDFAB_IDPRODUCTO PRIMARY KEY (IDFAB, IDPRODUCTO)  
);
```

```
CREATE TABLE OFICINAS
(
OFICINA NUMBER(2) NOT NULL,
CIUDAD VARCHAR2(15),
REGION VARCHAR2(10),
DIR NUMBER(3),
OBJETIVO NUMBER(10),
VENTAS NUMBER(10),
CONSTRAINT PK_OFICINA PRIMARY KEY (OFICINA)
);
```

```
CREATE TABLE EMPLEADOS
(
NUMEMP NUMBER(3) NOT NULL,
NOMBRE VARCHAR2(20),
EDAD NUMBER(2),
OFICINA NUMBER(2),
TITULO VARCHAR2(15),
CONTRATO DATE,
JEFE NUMBER(3),
CUOTA NUMBER(10),
VENTAS NUMBER(10),
CONSTRAINT PK_NUMEMP PRIMARY KEY (NUMEMP),
CONSTRAINT FK_OFICINA FOREIGN KEY (OFICINA) REFERENCES OFICINAS ON
    DELETE CASCADE,
CONSTRAINT MARGENEDAD CHECK (EDAD BETWEEN 18 AND 70)
);
```

```
CREATE TABLE CLIENTES
(
NUMCLIE NUMBER(4) NOT NULL,
NOMBRE VARCHAR2(20),
NUMEMP NUMBER(3),
LIMITECREDITO NUMBER(10),
CONSTRAINT PK_NUMCLIE PRIMARY KEY (NUMCLIE),
CONSTRAINT FK_NUMEMP1 FOREIGN KEY (NUMEMP) REFERENCES EMPLEADOS ON
    DELETE CASCADE
);
```

```

CREATE TABLE PEDIDOS
(
  CODIGO NUMBER(3) NOT NULL,
  NUMPEDIDO NUMBER(9) NOT NULL,
  FECHAPEDIDO DATE,
  NUMCLIE NUMBER(4) NOT NULL,
  NUMEMP NUMBER(3) NOT NULL,
  IDFAB VARCHAR2(10) NOT NULL,
  IDPRODUCTO VARCHAR2(15) NOT NULL,
  CANT NUMBER(4),
  CONSTRAINT PK_CODIGO PRIMARY KEY (CODIGO),
  CONSTRAINT FK_NUMCLIE FOREIGN KEY (NUMCLIE) REFERENCES CLIENTES ON
    DELETE CASCADE,
  CONSTRAINT FK_NUMEMP2 FOREIGN KEY (NUMEMP) REFERENCES EMPLEADOS ON
    DELETE CASCADE,
  CONSTRAINT FK_IDFAB_IDPRODUCTO FOREIGN KEY (IDFAB, IDPRODUCTO)
    REFERENCES PRODUCTOS ON DELETE CASCADE
);

```

2. En el disquete tenemos un fichero llamado DATOS2.SQL. Ejecuta este fichero para meter filas en las tablas anteriores.

Se tiene el fichero tofice7.sql, con las instrucciones de introducción de datos, se le agrega la información anterior de creación de tablas y se ejecuta:

```

CONNECT HR/HR
START tofice7.sql

```

3. Comprueba si todas las restricciones se han creado correctamente.

```

INSERT INTO EMPLEADOS
VALUES (101,'Antonio Viguer',17,12,'representante','20/10/86',104,300000,305000);

```

ORA-02290: restricción de control (HR.MARGENEDAD) violada

```

INSERT INTO EMPLEADOS
VALUES (101,'Antonio Viguer',71,12,'representante','20/10/86',104,300000,305000);

```

ORA-02290: restricción de control (HR.MARGENEDAD) violada

4. Mira la información que tiene cada una de las tablas.

SELECT * FROM OFICINAS ;

OFICINA	CIUDAD	REGION	DIR	OBJETIVO	VENTAS
11	Valencia	este	106	575000	693000
12	Alicante	este	104	800000	693000
13	Castellón	este	105	350000	693000
21	Badajoz	oeste	108	725000	693000
22	A Coruña	oeste	108	300000	693000
23	Madrid	centro	108	-	693000
24	Madrid	centro	-	250000	693000
26	Pamplona	norte	-	-	693000
28	Valencia	este	-	900000	693000

SELECT * FROM CLIENTES ;

NUMCLIE	NOMBRE	NUMEMP	LIMITECREDITO
2101	Luis García Antón	106	65000
2102	Alvaro Rodríguez	101	65000
2103	Jaime Llorens	105	50000
2105	Antonio Canales	101	45000
2106	Juan Suárez	102	65000
2107	Julián López	110	35000
2108	Julia Antequera	109	55000
2109	Alberto Juanes	103	25000
2111	Cristóbal García	103	50000
2112	Maria Silva	108	50000
2113	Luisa Maron	104	20000
2114	Cristina Bulini	102	20000
2115	Vicente Martínez	101	20000
2117	Carlos Tena	106	35000
2118	Junípero Alvarez	108	60000
2119	Salomon Bueno	109	25000
2120	Juan Malo	102	50000
2121	Vicente Ríos	103	45000
2122	José Marchante	105	30000
2123	José Libros	102	40000
2124	Juan Bolto	107	40000

SELECT * FROM PEDIDOS ;

CODIGO	NUMPEDIDO	FECHAPEDIDO	NUMCLIE	NUMEMP	IDFAB	IDPRODUCTO	CANT
1	110036	02/01/97	2107	110	aci	4100z	9
2	110036	02/01/97	2117	106	rei	2a44l	7
3	112963	10/05/97	2103	105	aci	41004	28
4	112968	11/01/90	2102	101	aci	41004	34
5	112975	11/02/97	2111	103	rei	2a44g	6
6	112979	12/10/89	2114	108	aci	4100z	6
7	112983	10/05/97	2103	105	aci	41004	6
8	112987	01/01/97	2103	105	aci	4100y	11
9	112989	10/12/97	2101	106	fea	114	6
10	112992	15/04/90	2118	108	aci	41002	10
11	112993	10/03/97	2106	102	rei	2a45c	24
12	112997	04/04/97	2124	107	bic	41003	1
13	113003	05/02/97	2108	109	imm	779c	3
14	113007	01/01/97	2112	108	imm	773c	3
15	113012	05/05/97	2111	105	aci	41003	25
16	113013	06/08/97	2118	108	bic	41003	1
17	113024	04/07/97	2114	108	qsa	xk47	20
18	113027	05/02/97	2103	105	aci	41002	54
19	113034	05/11/97	2107	110	rei	2a45c	8
20	113042	01/01/97	2113	101	rei	2a44r	5
21	113045	02/07/97	2112	108	rei	2a44r	10
22	113048	02/02/97	2120	102	imm	779c	2
23	113049	04/04/97	2118	108	qsa	xk47	2
24	113051	06/07/97	2118	108	qsa	xk47	4
25	113055	01/04/97	2108	101	aci	4100x	6
26	113057	01/11/97	2111	103	aci	4100x	24
27	113058	04/07/89	2107	109	fea	112	10
28	113062	04/07/97	2124	107	bic	41003	10
29	113065	03/06/97	2106	102	qsa	xk47	6
30	113069	01/08/97	2109	107	imm	773c	22

SELECT * FROM EMPLEADOS ;

NUMEMP	NOMBRE	EDAD	OFICINA	TITULO	CONTRATO	JEFE	CUOTA	VENTAS
101	Antonio Viguer	45	12	representante	20/10/86	104	300000	305000
102	Alvaro Jaumes	48	21	representante	10/12/86	108	350000	474000
103	Juan Rovira	29	12	representante	01/03/87	104	275000	286000
104	José González	33	12	dir ventas	19/05/87	106	200000	143000
105	Vicente Pantalla	37	13	representante	12/02/88	104	350000	368000
106	Luis Antonio	52	11	dir general	14/06/88	-	275000	299000
107	Jorge Gutiérrez	49	22	representante	14/11/88	108	300000	186000
108	Ana Bustamante	62	21	dir ventas	12/10/89	106	350000	361000
109	María Sunta	31	11	representante	12/10/99	106	300000	392000
110	Juan Víctor	41	13	representante	13/10/90	104	-	76000

SELECT * FROM PRODUCTOS ;

IDFAB	IDPRODUCTO	DESCRIPCION	PRECIO	EXISTENCIAS
aci	41001	arandela	58	277
aci	41002	bisagra	80	167
aci	41003	art t3	112	207
aci	41004	art t4	123	139
aci	4100x	junta	26	37
aci	4100y	extractor	2888	25
aci	4100z	mont	2625	28
bic	41003	manivela	652	3
bic	41089	rodamiento	225	78
bic	41672	plato	180	0
fea	112	cubo	148	115
fea	114	cubo	243	15
imm	773c	reostato	975	28
imm	775c	reostato 2	1425	5
imm	779c	reostato 3	1875	0
imm	887h	caja clavos	54	223
imm	887p	perno	25	24
imm	887x	manivela	475	32
qsa	xk47	red	355	38
qsa	xk48	red	134	203
qsa	xk48a	red	117	37
rei	2a44g	pas	350	14
rei	2a44l	bomba l	4500	12
rei	2a44r	bomba r	4500	12
rei	2a45c	junta	79	210

5. Obtener una lista de todos los productos indicando para cada uno su idfab, idproducto, descripción, precio y precio con I.V.A. incluido.

```
SELECT IDFAB, IDPRODUCTO, DESCRIPCION, PRECIO, PRECIO*0.18+PRECIO
"PRECIO CON IVA"
FROM PRODUCTOS ;
```

IDFAB	IDPRODUCTO	DESCRIPCION	PRECIO	PRECIO CON IVA
aci	41001	arandela	58	68,44
aci	41002	bisagra	80	94,4
aci	41003	art t3	112	132,16
aci	41004	art t4	123	145,14
aci	4100x	junta	26	30,68
aci	4100y	extractor	2888	3407,84
aci	4100z	mont	2625	3097,5
bic	41003	manivela	652	769,36
bic	41089	rodamiento	225	265,5
bic	41672	plato	180	212,4
fea	112	cubo	148	174,64
fea	114	cubo	243	286,74
imm	773c	reostato	975	1150,5
imm	775c	reostato 2	1425	1681,5
imm	779c	reostato 3	1875	2212,5
imm	887h	caja clavos	54	63,72
imm	887p	perno	25	29,5
imm	887x	manivela	475	560,5
qsa	xk47	red	355	418,9
qsa	xk48	red	134	158,12
qsa	xk48a	red	117	138,06
rei	2a44g	pas	350	413
rei	2a44l	bomba l	4500	5310
rei	2a44r	bomba r	4500	5310
rei	2a45c	junta	79	93,22

6. De cada pedido queremos saber su número de pedido, idfab, idproducto, cantidad, precio unitario e importe.

```
SELECT NUMPEDIDO,P.IDFAB, P.IDPRODUCTO, CANT, PRECIO, CANT*PRECIO
"IMPORTE"
FROM PEDIDOS P, PRODUCTOS PD
WHERE P.IDPRODUCTO=PD.IDPRODUCTO
AND P.IDFAB=PD.IDFAB ;
```

7. De cada empleado, listar su nombre y los años que lleva trabajando.

```
SELECT NOMBRE, CONTRATO, TRUNC((SYSDATE-CONTRATO)/365) "Años Trabajados"  
FROM EMPLEADOS ;
```

```
SELECT NOMBRE, CONTRATO, TRUNC (MONTHS_BETWEEN (SYSDATE,  
CONTRATO)/12) "AÑOS TRABAJADOS"  
FROM EMPLEADOS;
```

NOMBRE	CONTRATO	Años Trabajados
Antonio Viguer	20/10/86	25
Alvaro Jaumes	10/12/86	25
Juan Rovira	01/03/87	25
José González	19/05/87	24
Vicente Pantalla	12/02/88	24
Luis Antonio	14/06/88	23
Jorge Gutiérrez	14/11/88	23
Ana Bustamante	12/10/89	22
María Sunta	12/10/99	12
Juan Víctor	13/10/90	21

8. Obtener la lista de los clientes ordenados por código de empleado asignado. Visualizar todas las columnas de la tabla.

```
SELECT * FROM CLIENTES ORDER BY NUMEMP ;
```

NUMCLIE	NOMBRE	NUMEMP	LIMITECREDITO
2115	Vicente Martínez	101	20000
2102	Alvaro Rodríguez	101	65000
2105	Antonio Canales	101	45000
2123	José Libros	102	40000
2120	Juan Malo	102	50000
2114	Cristina Bulini	102	20000
2106	Juan Suárez	102	65000
2111	Cristóbal García	103	50000
2121	Vicente Ríos	103	45000
2109	Alberto Juanes	103	25000
2113	Luisa Maron	104	20000
2103	Jaime Llorens	105	50000
2122	José Marchante	105	30000
2101	Luis García Antón	106	65000
2117	Carlos Tena	106	35000
2124	Juan Bolto	107	40000
2112	María Silva	108	50000
2118	Junípero Alvarez	108	60000
2119	Salomon Bueno	109	25000
2108	Julia Antequera	109	55000
2107	Julián López	110	35000

9. Obtener las oficinas ordenadas por orden alfabético de región y dentro de cada región por ciudad.

```
SELECT OFICINA, REGION, CIUDAD
FROM OFICINAS
ORDER BY REGION, CIUDAD ;
```

OFICINA	REGION	CIUDAD
23	centro	Madrid
24	centro	Madrid
12	este	Alicante
13	este	Castellón
11	este	Valencia
28	este	Valencia
26	norte	Pamplona
22	oeste	A Coruña
21	oeste	Badajoz

10. Obtener los pedidos ordenados por fecha de pedido.

```
SELECT NUMPEDIDO, FECHAPEDIDO
FROM PEDIDOS
ORDER BY FECHAPEDIDO ;
```

NUMPEDIDO	FECHAPEDIDO
113058	04/07/89
112979	12/10/89
112968	11/01/90
112992	15/04/90
112987	01/01/97

11. Listar toda la información de los pedidos de marzo.

Dos maneras de hacerlo:

1º **SELECT * FROM PEDIDOS
WHERE TO_CHAR (FECHAPEDIDO, 'month') LIKE 'marzo%' ;**
2º **SELECT * FROM PEDIDOS
WHERE TO_CHAR (FECHAPEDIDO, 'mm') = 3 ;**

CODIGO	NUMPEDIDO	FECHAPEDIDO	NUMCLIE	NUMEMP	IDFAB	IDPRODUCTO	CANT
11	112993	10/03/97	2106	102	rei	2a45c	24

12. Listar los datos de las oficinas de las regiones del norte y del este (tienen que aparecer primero las del norte y después las del este).

```
SELECT * FROM OFICINAS
WHERE REGION LIKE 'norte'
OR REGION LIKE 'este'
ORDER BY REGION DESC ;
```

Otra forma de hacerlo:

```
SELECT * FROM OFICINAS  
WHERE REGION IN ('norte' , 'este')  
ORDER BY REGION DESC ;
```

OFICINA	CIUDAD	REGION	DIR	OBJETIVO	VENTAS
26	Pamplona	norte	-	-	693000
12	Alicante	este	104	800000	693000
28	Valencia	este	-	900000	693000
11	Valencia	este	106	575000	693000
13	Castellón	este	105	350000	693000

13. Listar los empleados de nombre Alvaro.

```
SELECT * FROM EMPLEADOS WHERE NOMBRE LIKE 'Alvaro %' ;
```

NUMEMP	NOMBRE	EDAD	OFICINA	TITULO	CONTRATO	JEFE	CUOTA	VENTAS
102	Alvaro Jaumes	48	21	representante	10/12/86	108	350000	474000

14. Listar los productos cuyo idproducto acabe en x..

```
SELECT * FROM PRODUCTOS  
WHERE IDPRODUCTO  
LIKE '%x' ;
```

IDFAB	IDPRODUCTO	DESCRIPCION	PRECIO	EXISTENCIAS
aci	4100x	junta	26	37
imm	887x	manivela	475	32

15. Listar las oficinas del este indicando para cada una de ellas su número, ciudad, números y nombre de sus trabajadores. Hacer una versión en la que aparezcan sólo las que tienen trabajadores, y hacer otra en las que aparezcan las oficinas del este que no tienen trabajadores.

```
SELECT E.OFICINA, O.CIUDAD, E.NUMEMP, E.NOMBRE  
FROM OFICINAS O, EMPLEADOS E  
WHERE O.OFICINA=E.OFICINA  
AND REGION='este'  
ORDER BY E.OFICINA ;
```

OFICINA	CIUDAD	NUMEMP	NOMBRE
11	Valencia	109	María Sunta
11	Valencia	106	Luis Antonio
12	Alicante	103	Juan Rovira
12	Alicante	104	José González
12	Alicante	101	Antonio Viguer
13	Castellón	110	Juan Víctor
13	Castellón	105	Vicente Pantalla

si se pone en el WHERE E.OFICINA (+) lista también las que no tienen empleados.

**SELECT OFICINA FROM OFICINAS
WHERE OFICINA NOT IN (SELECT OFICINA FROM EMPLEADOS)
AND REGION='este';**

OFICINA
28

16. Listar los pedidos mostrando su número, importe, nombre del cliente y el límite de crédito del cliente correspondiente.

**SELECT NUMPEDIDO, SUM (PR.PRECIO*P.CANT)"Importe", C.NOMBRE,
C.LIMITECREDITO
FROM PEDIDOS P, CLIENTES C, PRODUCTOS PR
WHERE P.NUMCLIE=C.NUMCLIE AND P.IDPRODUCTO=PR.IDPRODUCTO
GROUP BY P.NUMPEDIDO, C.NOMBRE, C.LIMITECREDITO
ORDER BY NUMPEDIDO, C.NOMBRE ;**

NUMPEDIDO	Importe	NOMBRE	LIMITECREDITO
110036	31500	Carlos Tena	35000
110036	23625	Julián López	35000
112963	3444	Jaime Llorens	50000
112968	4182	Alvaro Rodríguez	65000
112975	2100	Cristóbal García	50000
112979	15750	Cristina Bulini	20000
112983	738	Jaime Llorens	50000
112987	31768	Jaime Llorens	50000
112989	1458	Luis García Antón	65000
112992	800	Junípero Alvarez	60000
112993	1896	Juan Suárez	65000
112997	764	Juan Bolto	40000
113003	5625	Julia Antequera	55000
113007	2925	Maria Silva	50000
113012	19100	Cristóbal García	50000
113013	764	Junípero Alvarez	60000
113024	7100	Cristina Bulini	20000
113027	4320	Jaime Llorens	50000
113034	632	Julián López	35000
113042	22500	Luisa Maron	20000
113045	45000	Maria Silva	50000
113048	3750	Juan Malo	50000
113049	710	Junípero Alvarez	60000
113051	1420	Junípero Alvarez	60000
113055	156	Julia Antequera	55000
113057	624	Cristóbal García	50000
113058	1480	Julián López	35000
113062	7640	Juan Bolto	40000
113065	2130	Juan Suárez	65000
113069	21450	Alberto Juanes	25000

Esta otra sentencia da el mismo resultado:

```
SELECT P.NUMPEDIDO, (P.CANT*PR.PRECIO) "IMPORTE", C.NOMBRE "CLIENTE",
C.LIMITECREDITO
FROM PEDIDOS P, PRODUCTOS PR, CLIENTES C
WHERE C.NUMCLIE=P.NUMCLIE
AND P.IDPRODUCTO=PR.IDPRODUCTO
AND P.IDFAB=PR.IDFAB
ORDER BY P.NUMPEDIDO, C.NOMBRE ;
```

17. Listar el número, nombre, ciudad y región de cada trabajador.

```
SELECT NUMEMP "Nº Empleado", NOMBRE, O.CIUDAD, O.REGION
FROM EMPLEADOS E, OFICINAS O
WHERE E.OFICINA=O.OFICINA ;
```

Nº Empleado	NOMBRE	CIUDAD	REGION
101	Antonio Viguer	Alicante	este
102	Alvaro Jaumes	Badajoz	oeste
103	Juan Rovira	Alicante	este
104	José González	Alicante	este
105	Vicente Pantalla	Castellón	este
106	Luis Antonio	Valencia	este
107	Jorge Gutiérrez	A Coruña	oeste
108	Ana Bustamante	Badajoz	oeste
109	Maria Sunta	Valencia	este
110	Juan Víctor	Castellón	este

18. Listar el número de las oficinas con objetivo superior a 600.000 pts, indicando para cada una de ellas el nombre de su director.

```
SELECT O.OFICINA, OBJETIVO, NOMBRE "Director"
FROM EMPLEADOS E, OFICINAS O
WHERE O.DIR=E.NUMEMP(+)
AND OBJETIVO>600000
ORDER BY OFICINA ;
```

OFICINA	OBJETIVO	Director
12	800000	José González
21	725000	Ana Bustamante
28	900000	-

19. Listar los pedidos superiores a 25.000 pts, incluyendo el nombre del empleado que tomó el pedido y el nombre del cliente que lo solicitó.

NUMPEDIDO	Importe	Cliente	Empleado
110036	31500	Carlos Tena	Luis Antonio
112987	31768	Jaime Llorens	Vicente Pantalla
113045	45000	María Silva	Ana Bustamante

Varias formas de hacerlo:

```
SELECT P.NUMPEDIDO, (P.CANT*PR.PRECIO) "Importe", C.NOMBRE "Cliente",
E.NOMBRE "Empleado"
FROM PEDIDOS P, EMPLEADOS E, CLIENTES C, PRODUCTOS PR
WHERE P.NUMCLIE=C.NUMCLIE
AND P.NUMEMP=E.NUMEMP
AND P.IDPRODUCTO=PR.IDPRODUCTO
AND P.IDFAB=PR.IDFAB
AND (P.CANT*PR.PRECIO)>25000 ;
```

```
SELECT NUMPEDIDO,SUM (PR.PRECIO*P.CANT)"Importe", C.NOMBRE "Cliente",
E.NOMBRE "Empleado"
FROM PEDIDOS P, CLIENTES C, PRODUCTOS PR, EMPLEADOS E
WHERE P.NUMCLIE=C.NUMCLIE
AND P.NUMEMP=E.NUMEMP
AND P.IDPRODUCTO=PR.IDPRODUCTO
GROUP BY P.NUMPEDIDO, C.NOMBRE, E.NOMBRE
HAVING SUM (PR.PRECIO*P.CANT)>25000
ORDER BY NUMPEDIDO, C.NOMBRE ;
```

20. Hallar los trabajadores que realizaron su primer pedido el mismo día en que fueron contratados.

```
SELECT NOMBRE, CONTRATO "Fecha Contrato", FECHAPEDIDO "Fecha Pedido"
FROM EMPLEADOS E, PEDIDOS P
WHERE E.CONTRATO=P.FECHAPEDIDO
AND E.NUMEMP=P.NUMEMP ;
```

NOMBRE	Fecha Contrato	Fecha Pedido
Ana Bustamante	12/10/89	12/10/89

21. Listar los empleados con una cuota superior a la de su jefe; para cada empleado sacar sus datos y el número, nombre y cuota de su jefe.

Se crea una vista con los empleados y sus cuotas:

NUMEMP	NOMBRE	CUOTA
101	Antonio Viguer	300000
102	Alvaro Jaumes	350000
103	Juan Rovira	275000
104	José González	200000
105	Vicente Pantalla	350000
106	Luis Antonio	275000
107	Jorge Gutiérrez	300000
108	Ana Bustamante	350000
109	Maria Sunta	300000
110	Juan Víctor	-

```

CREATE VIEW EMPLECUOTA
AS SELECT NUMEMP, NOMBRE, CUOTA
FROM EMPLEADOS ;

```

```

SELECT E.NUMEMP, E.NOMBRE, E.CUOTA, C.NOMBRE "Jefe", C.NUMEMP, C.CUOTA
"Cuota Jefe"
FROM EMPLEADOS E, EMPLECUOTA C
WHERE E.CUOTA>C.CUOTA
AND E.JEFE=C.NUMEMP ;

```

NUMEMP	NOMBRE	CUOTA	Jefe	NUMEMP	Cuota Jefe
101	Antonio Viguer	300000	José González	104	200000
103	Juan Rovira	275000	José González	104	200000
105	Vicente Pantalla	350000	José González	104	200000
108	Ana Bustamante	350000	Luis Antonio	106	275000
109	María Sunta	300000	Luis Antonio	106	275000

22. Listar los códigos de los trabajadores que tengan una cuota inferior a 10.000 pts.

```

SELECT NUMEMP "Codigo Empleado"
FROM EMPLEADOS
WHERE CUOTA<10000 ;

```

no data found

23. ¿Cuál es la cuota media y las ventas medias de todos los empleados?

```

SELECT TO_CHAR(AVG(CUOTA), '999G990') "Cuota media",
TO_CHAR(AVG(VENTAS), '999G990') "Ventas medias"
FROM EMPLEADOS ;

```

Cuota Media	Ventas Medias
300.000	289.000

24. Hallar el importe medio de pedidos, el importe total de pedidos y el precio medio de venta (el precio de venta es el precio unitario en cada pedido).

```

SELECT TRUNC(AVG(PR.PRECIO*P.CANT)) "Importe medio", SUM(PR.PRECIO*P.CANT)
"Importe total", TRUNC (AVG (PR.PRECIO)) "Importe medio venta"
FROM PEDIDOS P, PRODUCTOS PR
WHERE P.IDPRODUCTO=PR.IDPRODUCTO
AND P.IDFAB=PR.IDFAB ;

```

Importe Medio	Importe Total	Importe Medio Venta
8256	247707	1076

25. Hallar el pedido medio de los productos del fabricante ACI.

```

SELECT TRUNC(AVG (PR.PRECIO*P.CANT)) "Importe"
FROM PEDIDOS P, PRODUCTOS PR
WHERE P.IDPRODUCTO=PR.IDPRODUCTO
AND P.IDFAB LIKE 'aci' ;

```

26. Hallar en qué fecha se realizó el primer pedido.

```
SELECT * FROM PEDIDOS  
WHERE FECHAPEDIDO= (SELECT MIN(FECHAPEDIDO) FROM PEDIDOS) ;
```

CODIGO	NUMPEDIDO	FECHAPEDIDO	NUMCLIE	NUMEMP	IDFAB	IDPRODUCTO	CANT
27	113058	04/07/89	2107	109	fea	112	10

o mas escueto:

```
SELECT MIN(FECHAPEDIDO) "Fecha primer Pedido"  
FROM PEDIDOS ;
```

27. Hallar cuántos pedidos hay de más de 25.000 pts..

Varias maneras de hacerlo:

```
SELECT COUNT(*) "Nº Pedidos" FROM PEDIDOS P, PRODUCTOS PR  
WHERE P.IDPRODUCTO=PR.IDPRODUCTO  
AND P.IDFAB=PR.IDFAB  
AND (PR.PRECIO*P.CANT)>25000 ;
```

```
SELECT COUNT (SUM (PRECIO*CANT)) "Nº Pedidos"  
FROM PEDIDOS P, PRODUCTOS PR  
WHERE P.IDPRODUCTO=PR.IDPRODUCTO  
AND P.IDFAB=PR.IDFAB  
GROUP BY NUMPEDIDO  
HAVING SUM (PRECIO*CANT)>25000 ;
```

Nº Pedidos
3

28. Listar el número de la oficina y el total del importe vendido por cada oficina.

```
SELECT O.OFICINA, SUM(P.CANT*PR.PRECIO)  
FROM OFICINAS O, PEDIDOS P, PRODUCTOS PR, EMPLEADOS E  
WHERE P.IDPRODUCTO=PR.IDPRODUCTO  
AND P.IDFAB=PR.IDFAB  
AND O.OFICINA=E.OFICINA  
AND E.NUMEMP=P.NUMEMP  
GROUP BY O.OFICINA  
ORDER BY O.OFICINA ;
```

OFICINA	SUM(P.CANT*PR.PRECIO)
11	40063
12	29562
13	67327
21	82133
22	28622

29.Para cada empleado cuyos pedidos suman más de 30.000 pts, hallar su importe medio de pedidos. En el resultado indicar el número de empleado y su importe medio de pedidos.

```
SELECT P.NUMEMP, ROUND(AVG(P.CANT*PR.PRECIO), 2) "Importe medio"  
FROM PEDIDOS P, PRODUCTOS PR  
WHERE P.IDPRODUCTO=PR.IDPRODUCTO  
AND P.IDFAB=PR.IDFAB  
GROUP BY P.NUMEMP  
HAVING SUM(P.CANT*PR.PRECIO)>30000  
ORDER BY P.NUMEMP ;
```

NUMEMP	Importe Medio
105	8614
106	16479
108	9294,63

30.Listar los nombres de los clientes que tienen asignado el representante Alvaro Jaumes.

Varias formas de hacerlo:

```
SELECT C.NOMBRE "Cliente", E.NOMBRE "Representante"  
FROM CLIENTES C, EMPLEADOS E  
WHERE C.NUMEMP=E.NUMEMP  
AND C.NUMEMP=(SELECT NUMEMP FROM EMPLEADOS WHERE NOMBRE LIKE 'Alvaro  
Jaumes') ;
```

```
SELECT C.NOMBRE "Cliente", E.NOMBRE "Representante"  
FROM CLIENTES C, EMPLEADOS E  
WHERE E.NUMEMP=C.NUMEMP  
AND UPPER(E.NOMBRE) LIKE 'ALVARO JAUMES' ;
```

Cliente	Representante
Juan Suárez	Alvaro Jaumes
Cristina Bulini	Alvaro Jaumes
Juan Malo	Alvaro Jaumes
José Libros	Alvaro Jaumes

31.Listar los empleados (numemp, nombre y nº de oficina) que trabajan en oficinas "buenas" (las que tienen ventas superiores a su objetivo).

Se crea una vista con los datos de objetivo y ventas realizadas por las oficinas:

```
CREATE VIEW VENTASOFICINAS  
AS SELECT O.OFICINA, O.OBJETIVO, SUM(P.CANT*PR.PRECIO) "VENTAS"  
FROM OFICINAS O, PEDIDOS P, PRODUCTOS PR, EMPLEADOS E  
WHERE P.IDPRODUCTO=PR.IDPRODUCTO  
AND P.IDFAB=PR.IDFAB  
AND O.OFICINA=E.OFICINA  
AND E.NUMEMP=P.NUMEMP
```

```
GROUP BY O.OFICINA, O.OBJETIVO  
ORDER BY O.OFICINA ;
```

Se hace la consulta tomando como referencia la vista creada:

```
SELECT E.NUMEMP, E.NOMBRE, E.OFICINA  
FROM EMPLEADOS E, OFICINAS O  
WHERE E.OFICINA=O.OFICINA  
AND O.OFICINA IN (SELECT OFICINA FROM VENTASOFICINAS  
WHERE VENTAS>OBJETIVO) ;
```

no data found

32. Listar los empleados que no trabajan en oficinas dirigidas por el empleado 108.

```
SELECT NOMBRE, OFICINA, JEFE  
FROM EMPLEADOS  
WHERE OFICINA NOT IN (SELECT OFICINA FROM OFICINAS WHERE DIR=108)  
ORDER BY OFICINA ;
```

NOMBRE	OFICINA	JEFE
Maria Sunta	11	106
Luis Antonio	11	-
Juan Rovira	12	104
José González	12	106
Antonio Viguer	12	104
Juan Víctor	13	104
Vicente Pantalla	13	104

33. Listar los productos (idfab, idproducto y descripción) para los cuales no se ha recibido ningún pedido de 25000 ó más.

IDFAB	IDPRODUCTO	DESCRIPCION
qsa	xk47	red
fea	112	cubo
fea	114	cubo
rei	2a44g	pas
rei	2a44r	bomba r
rei	2a45c	junta
aci	4100x	junta
aci	4100z	mont
aci	41002	bisagra
aci	41003	manivela
bic	41003	manivela
aci	41003	artt3
bic	41003	artt3
aci	41004	artt4
imm	773c	reostato
imm	779c	reostato 3

```

SELECT P.IDFAB, P.IDPRODUCTO, PR.DESCRIPCION
FROM PEDIDOS P, PRODUCTOS PR
WHERE P.IDPRODUCTO=PR.IDPRODUCTO
AND (P.CANT*PR.PRECIO)<25000
GROUP BY P.IDPRODUCTO, P.IDFAB, PR.DESCRIPCION
ORDER BY IDPRODUCTO;

```

34. Listar los clientes asignados a Ana Bustamante que no han remitido un pedido superior a 3.000 pts.

Varias formas de hacerlo:

```

SELECT P.NUMPEDIDO, P.IDFAB, P.IDPRODUCTO, PR.DESCRIPCION, P.NUMCLIE
FROM PEDIDOS P, PRODUCTOS PR
WHERE P.IDPRODUCTO=PR.IDPRODUCTO
AND (P.CANT*PR.PRECIO)<3000
GROUP BY P.NUMPEDIDO, P.IDPRODUCTO, P.IDFAB, PR.DESCRIPCION, P.NUMCLIE
HAVING P.NUMCLIE IN (SELECT NUMCLIE FROM CLIENTES WHERE
NUMEMP=(SELECT NUMEMP FROM EMPLEADOS WHERE NOMBRE LIKE 'Ana
Bustamante'))
ORDER BY P.NUMPEDIDO ;

```

```

SELECT P.NUMPEDIDO, P.IDFAB, P.IDPRODUCTO, PR.DESCRIPCION, C.NUMCLIE,
C.NOMBRE
FROM CLIENTES C, PEDIDOS P, PRODUCTOS PR
WHERE C.NUMCLIE=P.NUMCLIE
AND P.IDPRODUCTO=PR.IDPRODUCTO
AND C.NUMEMP=(SELECT NUMEMP FROM EMPLEADOS WHERE UPPER(NOMBRE)
LIKE 'ANA BUSTAMANTE')
AND (P.CANT*PR.PRECIO)<3000
ORDER BY P.NUMPEDIDO ;

```

NUMPEDIDO	IDFAB	IDPRODUCTO	DESCRIPCION	NUMCLIE	NOMBRE
112992	aci	41002	bisagra	2118	Junípero Alvarez
113007	imm	773c	reostato	2112	María Silva
113013	bic	41003	manivela	2118	Junípero Alvarez
113013	bic	41003	art t3	2118	Junípero Alvarez
113049	qsa	xk47	red	2118	Junípero Alvarez
113051	qsa	xk47	red	2118	Junípero Alvarez

35.Listar las oficinas en donde haya un vendedor cuyas ventas representen más del 55% del objetivo de su oficina

Esto es calculando las ventas por empleado de la tabla PEDIDOS:

```
SELECT O.OFICINA,E.NUMEMP, O.OBJETIVO, SUM(P.CANT*PR.PRECIO) "VENTAS"
FROM OFICINAS O, PEDIDOS P, PRODUCTOS PR, EMPLEADOS E
WHERE P.IDPRODUCTO=PR.IDPRODUCTO
AND P.IDFAB=PR.IDFAB
AND O.OFICINA=E.OFICINA
AND E.NUMEMP=P.NUMEMP
GROUP BY O.OFICINA,E.NUMEMP, O.OBJETIVO
HAVING SUM(P.CANT*PR.PRECIO)> O.OBJETIVO*0.55
ORDER BY O.OFICINA ;
```

no data found

Si nos fijamos en la columna VENTAS de la tabla EMPLEADOS seria:

```
SELECT O.OFICINA, E.NUMEMP, O.OBJETIVO "Objetivo Oficina", E.VENTAS
"VENTAS Empleado"
FROM OFICINAS O, EMPLEADOS E
WHERE O.OFICINA=E.OFICINA
GROUP BY O.OFICINA,E.NUMEMP, O.OBJETIVO, E.VENTAS
HAVING E.VENTAS> O.OBJETIVO*0.55
ORDER BY O.OFICINA ;
```

otra manera:

```
SELECT O.OFICINA, E.NUMEMP, O.OBJETIVO "Objetivo oficina", E.VENTAS "Ventas
empleado"
FROM OFICINAS O, EMPLEADOS E
WHERE E.VENTAS>O.OBJETIVO*0.55
AND O.OFICINA=E.OFICINA
ORDER BY O.OFICINA ;
```

OFICINA	NUMEMP	Objetivo Oficina	Ventas Empleado
11	109	575000	392000
13	105	350000	368000
21	102	725000	474000
22	107	300000	186000

36.Listar las oficinas en donde todos los vendedores tienen ventas que superan el 50% del objetivo de la oficina.

Se crea una vista con los empleados, ventas, oficinas y objetivos.

```
CREATE VIEW VENTASOBJETIVO
AS SELECT E.NUMEMP, E.VENTAS, O.OFICINA, O.OBJETIVO
FROM EMPLEADOS E, OFICINAS O
WHERE E.OFICINA=O.OFICINA
ORDER BY O.OFICINA ;
```

NUMEMP	VENTAS	OFICINA	OBJETIVO
106	299000	11	575000
109	392000	11	575000
103	286000	12	800000
104	143000	12	800000
101	305000	12	800000
110	76000	13	350000
105	368000	13	350000
108	361000	21	725000
102	474000	21	725000
107	186000	22	300000

Con esta sentencia listo las oficinas y el numero de empleados que cumplen la condición:

```
SELECT OFICINA, COUNT(*) FROM VENTASOBJETIVO V
WHERE VENTAS > (SELECT OBJETIVO * .50 FROM VENTASOBJETIVO WHERE
V.NUMEMP=NUMEMP)
GROUP BY OFICINA ;
```

OFICINA	COUNT(*)
22	1
13	1
11	2
21	1

Con esta sentencia listo las oficinas y el numero de empleados que tienen:

```
SELECT OFICINA, COUNT(*) FROM VENTASOBJETIVO
GROUP BY OFICINA ;
```

OFICINA	COUNT(*)
22	1
13	2
11	2
21	2
12	3

Con esta sentencia saco las oficinas que todos sus empleados cumplen la condición:

```
SELECT OFICINA, COUNT(*) FROM VENTASOBJETIVO V
WHERE VENTAS > (SELECT OBJETIVO * .50 FROM VENTASOBJETIVO WHERE
V.NUMEMP=NUMEMP)
GROUP BY OFICINA
INTERSECT
SELECT OFICINA, COUNT(*)
FROM VENTASOBJETIVO
GROUP BY OFICINA ;
```

OFICINA	COUNT(*)
11	2
22	1

37. Listar las oficinas que tengan un objetivo mayor que la suma de las cuotas de sus trabajadores.

```
SELECT O.OFICINA, O.OBJETIVO, SUM(CUOTA) "Suma cuotas empleados"
FROM OFICINAS O, EMPLEADOS E
WHERE O.OFICINA=E.OFICINA
GROUP BY O.OFICINA, O.OBJETIVO
HAVING OBJETIVO>SUM(CUOTA);
```

OFICINA	OBJETIVO	Suma Cuotas Empleados
12	800000	775000
21	725000	700000

38. Crear una tabla (llamarla nuevatrabajadores) que contenga las filas de la tabla trabajadores.

```
CREATE VIEW NUEVATRABAJADORES
AS SELECT * FROM EMPLEADOS;
```

se crea la tabla:

```
CREATE TABLE NUEVATRABAJADORES
AS SELECT * FROM EMPLEADOS;
```

39. Crear una tabla (llamarla nuevaoficinas) que contenga las filas de la tabla oficinas.

```
CREATE VIEW NUEVAOFICINAS
AS SELECT * FROM OFICINAS;
```

se crea la tabla:

```
CREATE TABLE NUEVAOFICINAS
AS SELECT * FROM OFICINAS;
```

40. Crear una tabla (llamarla nuevaproductos) que contenga las filas de la tabla productos.

```
CREATE VIEW NUEVAPRODUCTOS
AS SELECT * FROM PRODUCTOS;
```

se crea la tabla:

```
CREATE TABLE NUEVAPRODUCTOS
AS SELECT * FROM PRODUCTOS;
```

41. Crear una tabla (llamarla nuevapedidos) que contenga las filas de la tabla pedidos.

```
CREATE VIEW NUEVAPEDIDOS
AS SELECT * FROM PEDIDOS;
```

se crea la tabla:

**CREATE TABLE NUEVAPEDIDOS
AS SELECT * FROM PEDIDOS ;**

42. Subir un 5% el precio de todos los productos del fabricante ACI.

**UPDATE PRODUCTOS
SET PRECIO=PRECIO+PRECIO*0.05
WHERE IDFAB='aci' ;
7 row(s) updated.**

IDFAB	IDPRODUCTO	DESCRIPCION	PRECIO	EXISTENCIAS
aci	41001	arandela	61	277
aci	41002	bisagra	84	167
aci	41003	art t3	118	207
aci	41004	art t4	129	139
aci	4100x	junta	27	37
aci	4100y	extractor	3032	25
aci	4100z	mont	2756	28

43. Añadir una nueva oficina para la ciudad de Madrid, con el número de oficina 30, con un objetivo de 100.000 y región Centro.

1º Forma

**INSERT INTO OFICINAS
SELECT DISTINCT 30, CIUDAD, REGION, '', 100000,
FROM OFICINAS
WHERE CIUDAD='Madrid' ;**

2º Forma

**INSERT INTO OFICINAS (OFICINA, CIUDAD, REGION, DIR, OBJETIVO, VENTAS)
VALUES (30, 'Madrid', 'centro', '', 100000, '') ;**

OFICINA	CIUDAD	REGION	DIR	OBJETIVO	VENTAS
30	Madrid	centro	-	100000	-
40	Madrid	centro	-	100000	-
11	Valencia	este	106	575000	693000
12	Alicante	este	104	800000	693000
13	Castellón	este	105	350000	693000
21	Badajoz	oeste	108	725000	693000
22	A Coruña	oeste	108	300000	693000
23	Madrid	centro	108	-	693000
24	Madrid	centro	-	250000	693000
26	Pamplona	norte	-	-	693000
28	Valencia	este	-	900000	693000

44.Cambiar los empleados de la oficina 21 a la oficina 31.

```
UPDATE EMPLEADOS  
SET OFICINA=31  
WHERE OFICINA=21 ;
```

ORA-02291: restriccin de integridad (HR.FK_OFICINA) violada - clave principal no encontrada

Da error porque la oficina no existe, habria que insertarlas en la tabla OFICINAS

45. Eliminar los pedidos del empleado 105.

```
DELETE FROM PEDIDOS  
WHERE NUMEMP=105 ;  
5 row(s) deleted.
```

46.Eliminar las oficinas que no tengan empleados.

```
DELETE FROM OFICINAS  
WHERE OFICINA NOT IN (SELECT OFICINA FROM EMPLEADOS);  
7 row(s) deleted.
```

47.Recuperar los precios originales de los productos a partir de la tabla nuevaproductos.

No se puede hacer porque anteriormente se ha creado una vista.

Tras crear la tabla esta seria la sentencia para recuperar los precios:

```
UPDATE PRODUCTOS  
SET PRECIO=(SELECT PRECIO FROM NUEVAPRODUCTOS WHERE IDFAB='aci')  
WHERE IDFAB='aci' ;
```

48.Recuperar los pedidos borrados en el ejercicio 45, a partir de la tabla nuevapedidos.

No se puede hacer porque anteriormente se ha creado una vista.

Tras crear la tabla esta seria la sentencia para recuperar los pedidos:

```
INSERT INTO PEDIDOS  
SELECT * FROM NUEVAPEDIDOS WHERE NUMEMP=105 ;
```