

WUOLAH



Burial

www.wuolah.com/student/Burial



14053

EJERCICIOS.pdf

Todos los Ejercicios de Grupos (Clase) - 2017



2º Bases de Datos



Grado en Ingeniería Informática



**Escuela Politécnica Superior de Córdoba
UCO - Universidad de Córdoba**



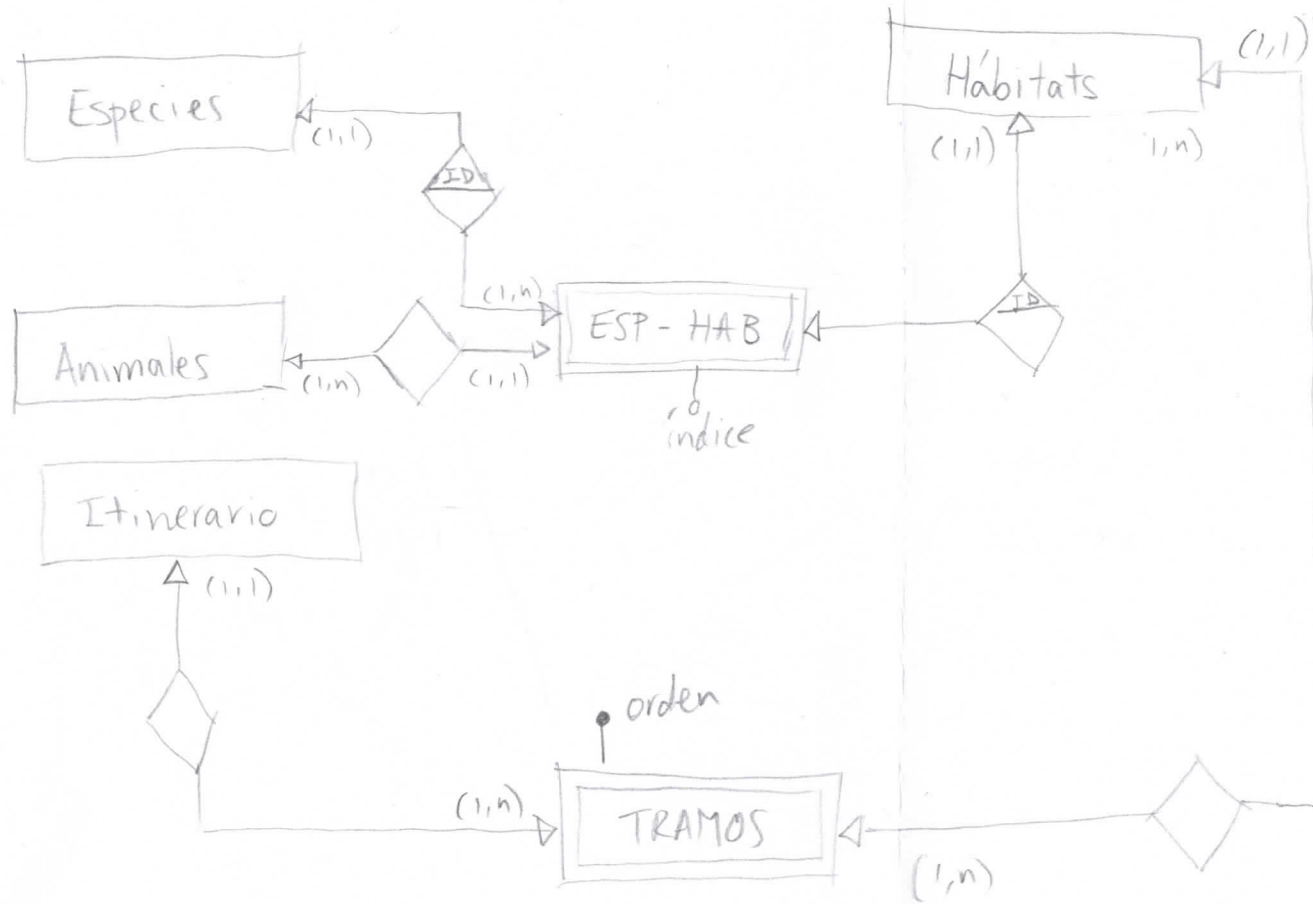
PARQUE ZOOLOGICO

Un parque zoológico quiere construir una BD para organizar las especies que posee y los distintos itinerarios para visitar el parque. La información se estructura de la siguiente forma. De las especies, se desea conocer su nombre común y su nombre científico, así como una descripción general y una fotografía. Cada especie puede vivir en distintos hábitats naturales, definidos por su nombre, clima y vegetación predominante. Cada especie tiene asociado un índice de vulnerabilidad dentro de cada hábitat, que mide el riesgo de extinción de la especie en el dicho hábitat.

Para organizar las visitas, y en función de los hábitats que desee recorrer un visitante, el parque le ofrece una serie de recorridos por los hábitats, que se identifican por su código y se caracterizan por su duración estimada, longitud y número máximo de visitantes permitidos. Un hábitat sólo puede formar parte de un itinerario

Realizar el problema conceptual correspondiente a este problema

SOLUCIÓN MODELO CONCEPTUAL - ZOOLOGICO.





CARTA DEL RESTAURANTE

Se desea construir una base de datos que almacene la carta de un restaurante.

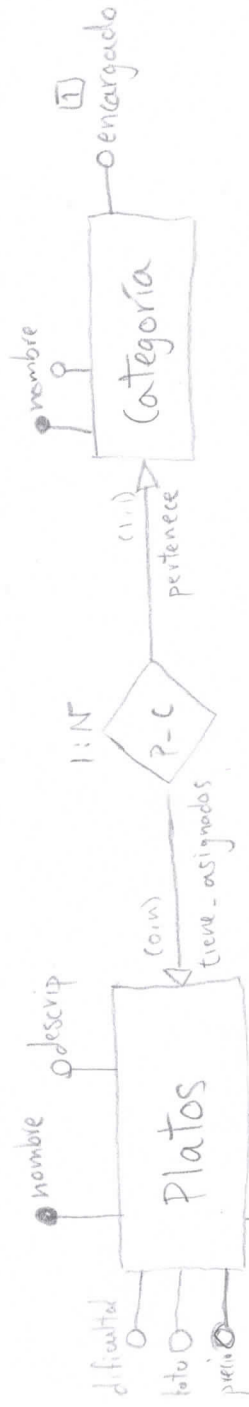
Para cada plato, se desea obtener su nombre, descripción, nivel de dificultad (de elaboración), una foto y el precio final para el cliente. Cada plato pertenece a una categoría.

Las categorías se caracterizan por su nombre, una breve descripción y el nombre del encargado.

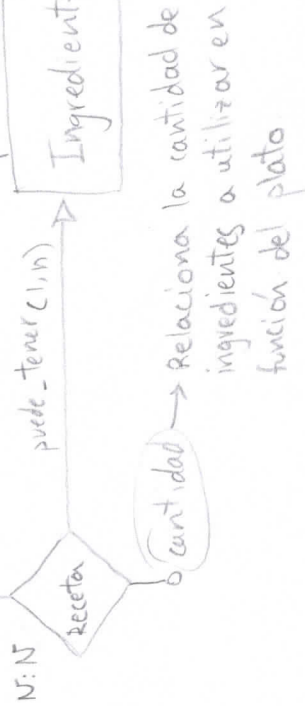
Además de los platos, se desea conocer las recetas para su realización, con la lista de ingredientes necesarios, aportando la cantidad requerida, las unidades de medida (gramos, litros, etc.) y cantidad actual en el almacén.

Realizar el problema conceptual correspondiente a este problema

CARTA DE UN RESTAURANTE

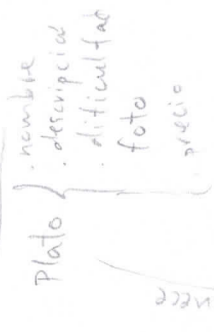


Si no estuviere aquí unidad,
si por ejemplo stock=48,
no sabríamos si es 48 kg, l, vasos...

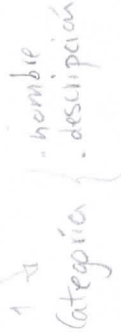


Relaciona la cantidad de
ingredientes a utilizar en
función del plato

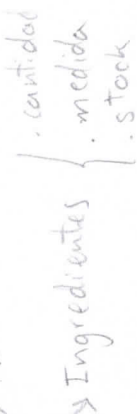
CARTA



pertenece



Receta



1] Encargado NO es una identidad por que nuestro dominio es en torno a la carta, no a los encargados del restaurante como ident.

2] Receta NO es una identidad por que simplemente es una papel asociado al plato y es una DESCRIPCIÓN

3] En el caso de que el atributo stock no estuviere en el dominio del problema, NO EXISTIRÍA la identidad Ingredientes y tan solo sería un atributo de plato.



GESTIÓN DE VUELOS

Una compañía aérea necesita una base de datos para registrar la información de sus vuelos.

Los vuelos están caracterizados por un Id, la fecha y los aeropuertos de origen y destino. Cada vuelo es realizado por un avión. Los aviones tienen una matrícula que los identifica, el fabricante, un modelo e información sobre su capacidad (número máximo de pasajeros) y autonomía de vuelo (en horas). La tripulación asignada al vuelo está formada por el personal de la propia compañía. De cada trabajador se conoce su id, su nombre y su categoría profesional, así como el puesto que ocupa en cada vuelo en particular.

Por último, para cada vuelo, se almacena la lista completa de pasajeros, con su dni, el nombre, el asiento que ocupa y su clase (turista, primera o business).

Realizar el modelo conceptual (Entidad-Interrelación) y generar el correspondiente esquema relacional normalizado a FNBC.



PLANETAS y EXCURSIONES PLANETARIAS

PRIMERA PARTE

Se desea representar y mantener información de los sistemas solares de la Vía Láctea de forma que se conozca la posición (orden de cercanía) de cada planeta con respecto a la estrella a la cual orbita.

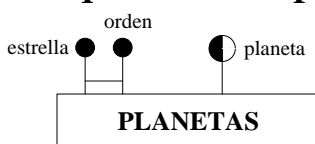
Realizar el diseño conceptual y relacional que sea capaz de representar el problema.

Existe completa libertad para que consideren los atributos o características de los tipos de entidades necesarias de representar en la solución.

Dado que no se solicita ninguna información acerca de la estrella o de los planetas que no sea la de su identificación y orden orbital respecto a la estrella, solo es necesario considerar un tipo de entidad, como se muestra en la figura.

El identificador principal será el agregado de estrella y orden y el identificador secundario será el nombre del planeta (o viceversa). De esta forma cada planeta estará siempre asociado a una estrella y un orden orbital dentro de ella.

Esquema Conceptual



Esquema Relacional

Planetas (estrella, orden, planeta)

El modelo relacional es simple. Una única tabla con los únicos tres atributos y con claves exactamente igual a los identificadores seleccionados en el modelo conceptual.

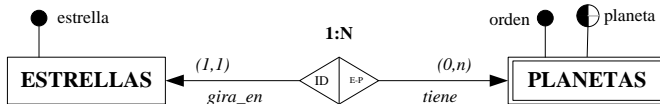
El problema se presentaría si existieran estrellas que no tuvieran planetas y se deseara su representación, puesto que ni el atributo orden, ni el atributo planeta pueden ser nulos, este modelo conceptual no sería apropiado.

Un mecanismo de diseño sería el considerado en la siguiente figura. Aunque planetas pueda identificarse por el atributo de su nombre, se puede definir el tipo de entidad *Planetas* como débil por identificación respecto al tipo de entidad *Estrellas* y utilizar el atributo *planeta* como identificador alternativo.



PLANETAS y EXCURSIONES PLANETARIAS

Esquema Conceptual



Esquema Relacional

Estrella (estrella)

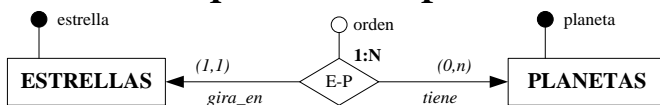
Planetas (estrella, orden, planeta)

Planetas.estrella -> Estrellas.estrella

El modelo relacional estaría formado por una tabla para el tipo de entidad *Estrellas* y otra para el tipo de entidad *Planetas*, con clave foránea a la tabla *Estrellas*.

Si no se utiliza este mecanismo de diseño se debería incluir el atributo *orden* bien en el tipo de entidad *Planetas* o bien en la relación entre los tipos de entidad *Estrellas* y *Planetas*, la cual ya no representaría ninguna debilidad tal y como muestra la siguiente figura.

Esquema Conceptual



Esquema Relacional

Estrella (estrella)

Planetas (planeta, estrella, orden)

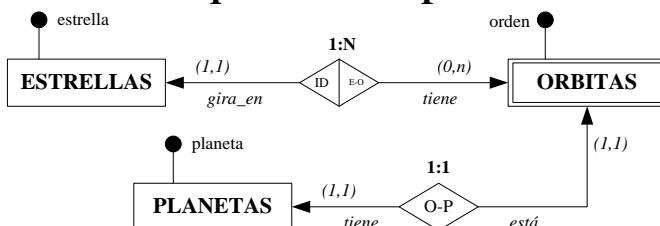
Planetas.estrella -> Estrellas.estrella

En este caso es necesario especificar sintácticamente una restricción de dominio de forma que el atributo *orden* no se pueda repetir para el conjunto relación *E-P* para un mismo valor del atributo *estrella*.

El modelo relacional resultante es sensiblemente diferente al caso anterior, puesto que en este caso la clave principal sería el atributo *planeta* y la clave secundaria sería el agregado de los atributos *estrella* y *orden*.

La descomposición del tipo de interrelación E-P en un tipo de entidad débil por identificación respecto de estrella como se muestra en la figura siguiente no es apropiada por diferentes razones.

Esquema Conceptual



Esquema Relacional

Estrella (estrella)

Orbitas (estrella, orden, planeta)

Planetas (planeta, estrella, orden)

Orbitas.estrella -> Estrellas.estrella

Planetas.(estrella, orden) ->

Orbitas.(estrella, orden)

Orbitas.planeta -> Planetas.planeta



PLANETAS y EXCURSIONES PLANETARIAS

El tipo de interrelación E-P es 1:N y su descomposición genera dos tipos de interrelaciones 1:N y 1:1. El tipo de entidad Planetas con un interrelación 1:1 puede ser considerado como un atributo del tipo de entidad Orbitas (al no tener además más atributos) por lo que no aporta nada al modelo.

El tipo de entidad Orbitas no tiene propiedades que lo caractericen, además del orden, por lo que no es necesaria en el modelo.

Tanto el modelo conceptual, como el relacional resultantes son mucho más complejos y no se elimina ninguna restricción de inserción, borrado o modificación al mantenerse las mismas claves en la tabla Planetas.

Al existir claves foráneas traslapadas entre las tablas Orbitas y Planetas es necesario generar procedimientos con transacciones que garanticen la integridad en los procesos de inserción/borrado/modificación, dado que no se podrá realizar estas operaciones in deshabilitar las restricciones impuestas por estas claves foráneas



PLANETAS y EXCURSIONES PLANETARIAS

SEGUNDA PARTE

La agencia de viajes “Virgin2120” organiza excursiones a planetas de la Vía Láctea. En cada excursión de su catálogo se visitan 1 o varios planetas de uno o varios sistemas solares. En cada excursión los planetas se visitan en viajes de ida/vuelta desde el planeta Tierra y se pueden visitar uno o varios sistemas solares y uno o varios planetas de cada sistema solar.

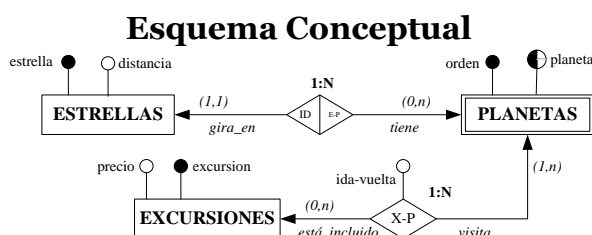
Los itinerarios de cada una de las excursiones siempre se realizan en base a la cercanía o lejanía del planeta a su estrella y de la estrella del sistema solar a la Tierra, de forma que en el viaje de ida el orden de visita es desde más cercano a más lejano y en el de vuelta es desde el más lejano al más cercano.

Realizar el diseño conceptual y relacional que sea capaz de representar el problema.

Existe completa libertad para que consideren los atributos o características de los tipos de entidades necesarias de representar en la solución.

Dado que no se solicita ninguna información acerca de las paradas que se realizan en cada planeta y que el orden siempre indica la distancia a la estrella (en cualquier caso en lugar del atributo *orden* se podía considerar un atributo *distancia*), no es necesario considera un orden de visita pues está intrínsecamente establecido en la posición del planeta a su estrella.

Para considerar este orden de visita en función de la distancia de la estrella a la tierra simplemente es necesario considerar un atributo *distancia* en el tipo de entidad *Estrellas*.



Esquema Relacional

Estrella (estrella, orden)
Planetas (estrella, orden, planeta)
Excursiones (excursion)
Trayectos (excursión, estrella, orden, ida-vuelta)

Planetas.estrella -> Estrellas.estrella
Trayectos.excursion -> Excursiones.excursion
Trayectos.(estrella, orden) -> Planetas.(estrella, orden)

El orden de recorrido se obtiene mediante la cláusula ORDER BY de una operación SELECT sobre la tabla *Excursiones*. *El atributo ida-vuelta permite caracterizar los planetas que se visitan en uno u otro recorrido, permitiendo que un planeta se pueda visitar en ambas direcciones.*



GESTIÓN DE VUELOS

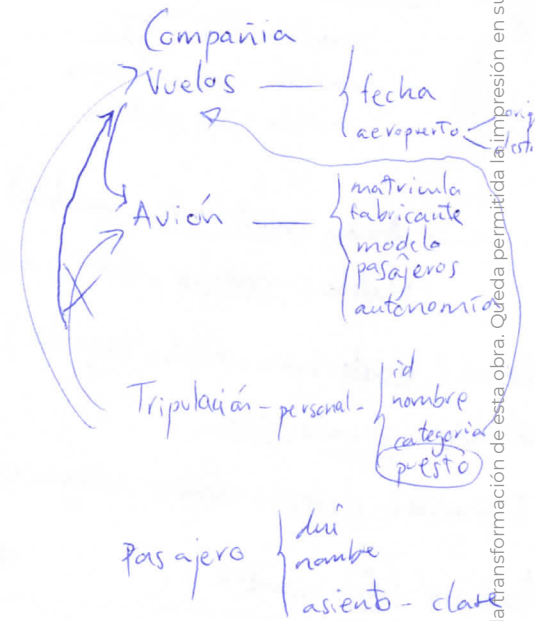
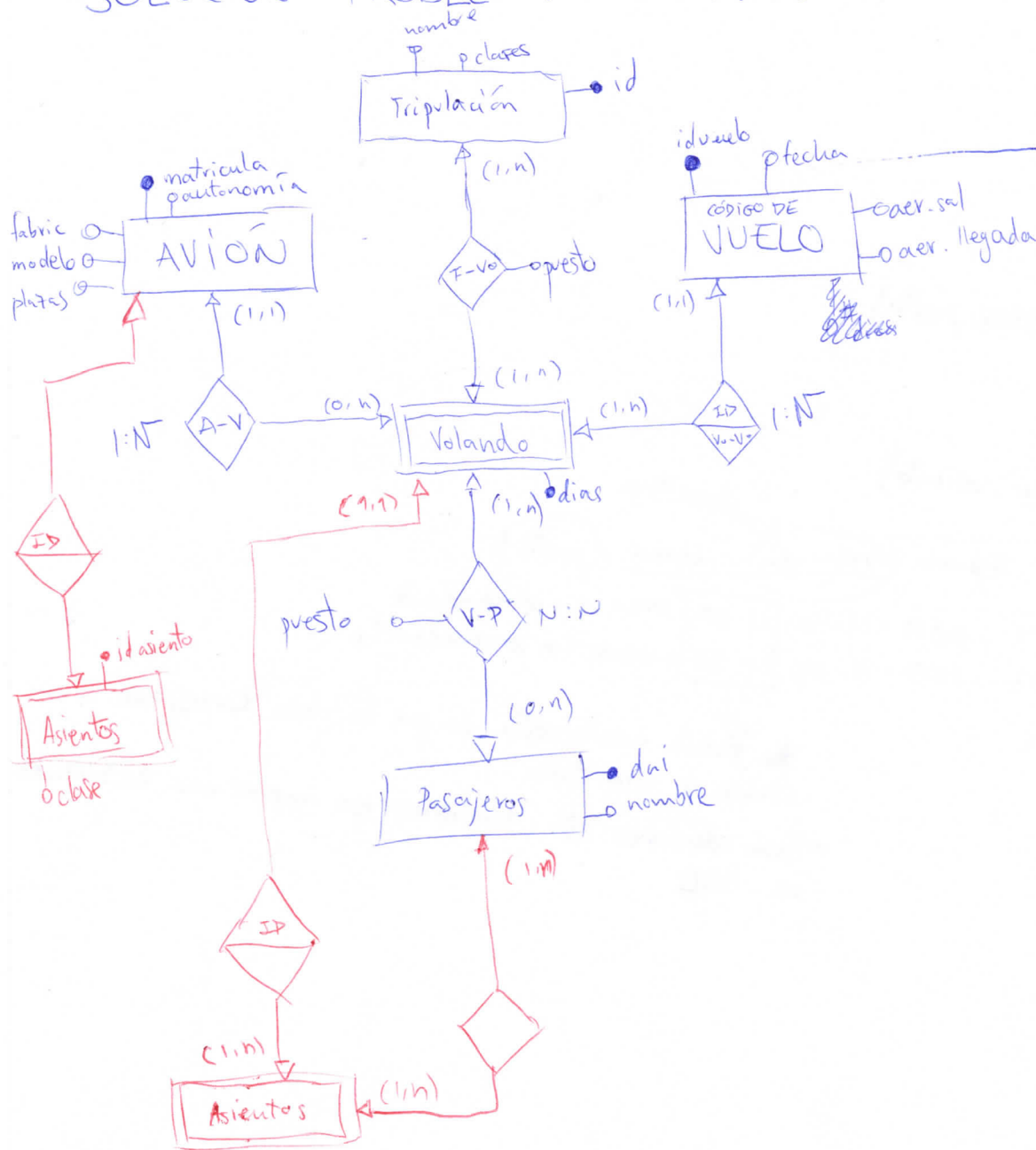
Una compañía aérea necesita una base de datos para registrar la información de sus vuelos.

Los vuelos están caracterizados por un Id, la fecha y los aeropuertos de origen y destino. Cada vuelo es realizado por un avión. Los aviones tienen una matrícula que los identifica, el fabricante, un modelo e información sobre su capacidad (número máximo de pasajeros) y autonomía de vuelo (en horas). La tripulación asignada al vuelo está formada por el personal de la propia compañía. De cada trabajador se conoce su id, su nombre y su categoría profesional, así como el puesto que ocupa en cada vuelo en particular.

Por último, para cada vuelo, se almacena la lista completa de pasajeros, con su dni, el nombre, el asiento que ocupa y su clase (turista, primera o business).

Realizar el modelo conceptual (Entidad-Interrelación) y generar el correspondiente esquema relacional normalizado a FNBC.

SOLUCIÓN PROBLEMA: AEROLÍNEA



Idvuelo: un identificador puede ser usado en más de un vuelo. ~~Responde a la pregunta principal de la tarea.~~

| | FECHA | HORA |
|---------|------------|-------|
| IB 2024 | 2017-11-09 | 11:30 |
| IB 2024 | 2017-11-09 | 12:30 |

*Condición impuesta por el profesor.

*Corrección 16/11/17

- En la solución anterior, ~~sea~~ el cliente no podía saber el número de asientos disponibles antes de tomar el avión.

Relacional

Avión (matricula, autonomía, ~~fabric~~, plazas, modelo)

modelo $\xrightarrow{\text{determina}}$ fabric
 modelo \rightarrow autonomía
 modelo \rightarrow plazas

Avión (matricula, modelo)

Modelo (modelo, fabric, autonomía, plazas)

Vuelos (idvuelo, hora, salida, llegada)

Volando (idvuelo, día)

Tripulación (idcrew, nombre, categoría)

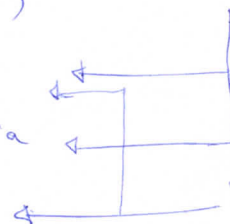
Pasajero (dni, nombre)

Avión
 Modelo
 Vuelos
 Volando

Tripulación

Pasajero

Trip-vuelo (vuelo, día, idcrew, puesto)



Vol- pasajeros (idvuelo, dni, asiento)

Trip-vuelo hay un problema

Si presieremos tuplas:

IB 2024, 2017-11-10, 11:30, crew-1, comandante
 IB 4043, 2017-11-10, 11:30, crew-1, comandante

↓
 Se podría suceder esto, ya que la hora corresponde a un atributo de otra tabla.

*Para solucionar esto, tendremos que imponer una **RESTRICIÓN** en **SAB**



RUTAS DE TRANSPORTE

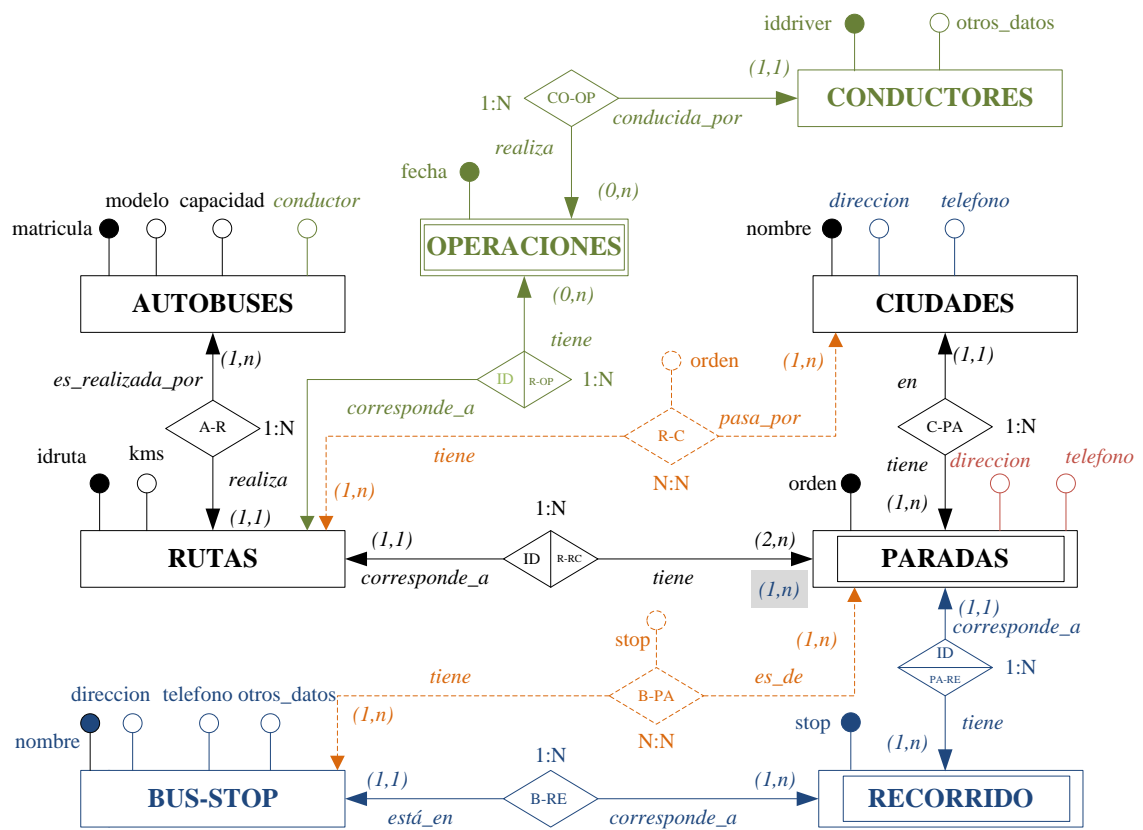
Una empresa de transporte desea crear una base de datos para almacenar información sobre sus rutas.

La empresa dispone de una flota de autobuses que distribuye en una serie de rutas. En cada ruta, el autobús pasa por un conjunto de ciudades en las que tiene parada.

Una ruta se identifica por un código y se caracteriza por los km. totales de recorrido, el origen y el destino final. De cada autobús, se almacena su matrícula, el modelo, su capacidad (plazas) y el nombre del conductor. Se asume que un autobús sólo puede estar realizando una ruta.

Las rutas tienen paradas en distintas ciudades. De cada ciudad, almacenaremos el nombre, junto con la dirección y el teléfono del lugar de parada. Para organizar las rutas, cada parada tiene un número de orden, que puede variar entre distintas rutas (una misma ciudad puede pertenecer a varias rutas).

Realizar el modelo conceptual (Entidad-Interrelación) y generar el correspondiente esquema relacional normalizado a FNBC.





ALQUILER DE VEHÍCULOS

Una empresa de alquiler de vehículos desea conocer en todo momento el estado de su flota. La empresa tiene diversas oficinas repartidas por todo el territorio español. Cada oficina se identifica por un código único y se caracteriza por la ciudad en la que se encuentra y su dirección completa (calle, número y código postal) y teléfono.

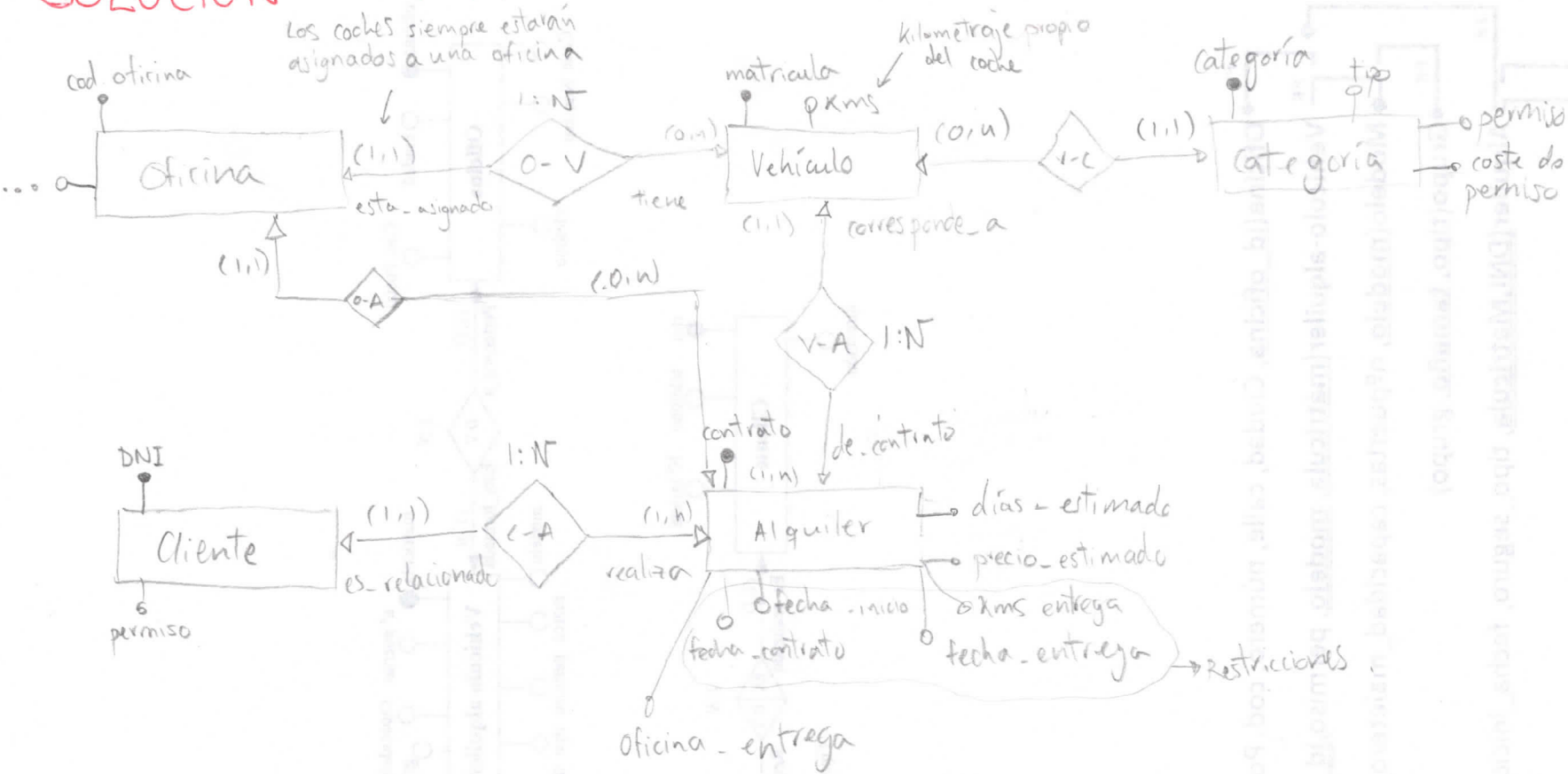
En cada oficina hay disponible un conjunto de vehículos, de los cuales se conoce su matrícula, el grupo al que pertenece: A, B, C, D, E, F o G (depende del tipo y tamaño del vehículo), la marca, el modelo, el número de puertas, el número de plazas, la capacidad del maletero y el permiso de circulación exigido para el alquiler.

Para llevar el control del estado de cada vehículo, la empresa mantiene un registro de todos los alquileres que ha realizado, indicando para cada uno de ellos el nombre del conductor, su DNI, su dirección, un teléfono de contacto y un número de tarjeta de crédito sobre la que realizar los cargos correspondientes. Además de esta información de los clientes, para cada alquiler se almacena su duración (en días), el tipo de seguro contratado, los kilómetros realizados y el precio total.

Realizar el modelo conceptual y relacional correspondiente a este problema

SOLUCIÓN - COMPAÑÍA DE ALQUILER DE COCHES.

Los coches siempre estarán asignados a una oficina



PROBLEMAS CONCEPTUALES

- 1.- Un coche no siempre pertenece a una oficina
- 2.- Coste del seguro

* Existe un problema entre permiso de una persona y la categoría del vehículo (podría un cliente alquilar un camión con el permiso B)

→ CREAMOS ENTIDAD "CATEGORÍA"

* Restricciones

Fecha_inicio < Fecha_termino

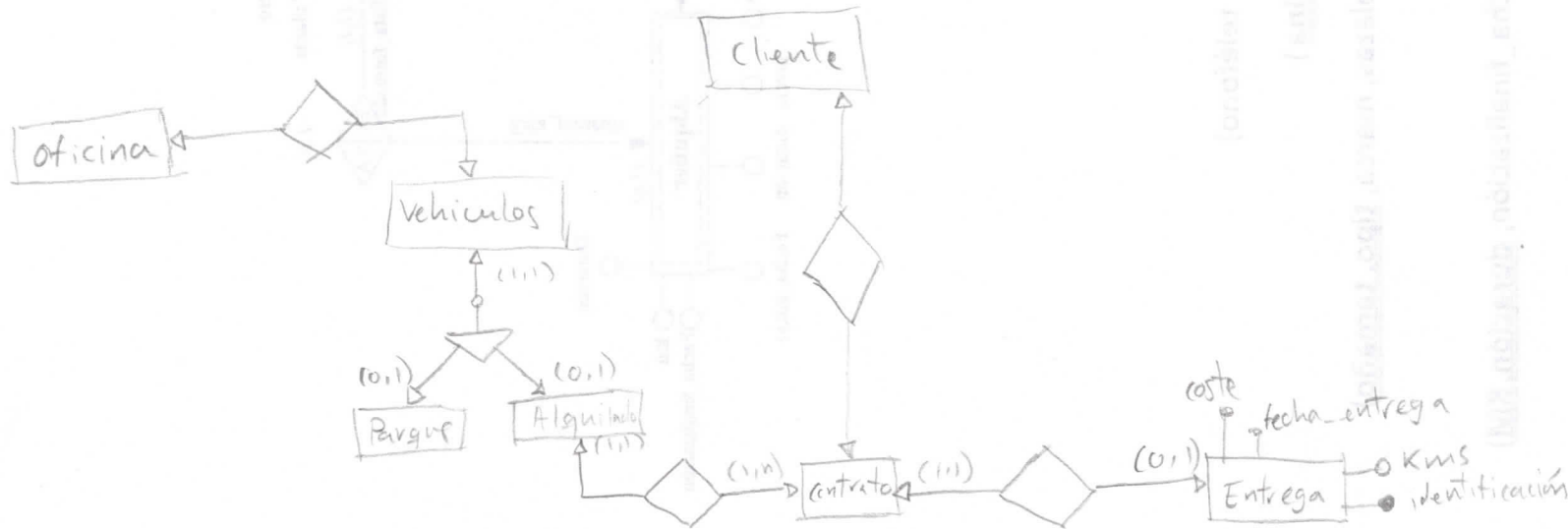
* Alquiler hay que relacionarlo con oficina (entregamos en una determinada oficina)

→ PROBLEMA

Podemos reestructurar nuestro conceptual de forma completa

→ Existe un bucle peligroso entre oficina-vehículo-alquiler

que se soluciona con el atrib. "Oficina-entrega"

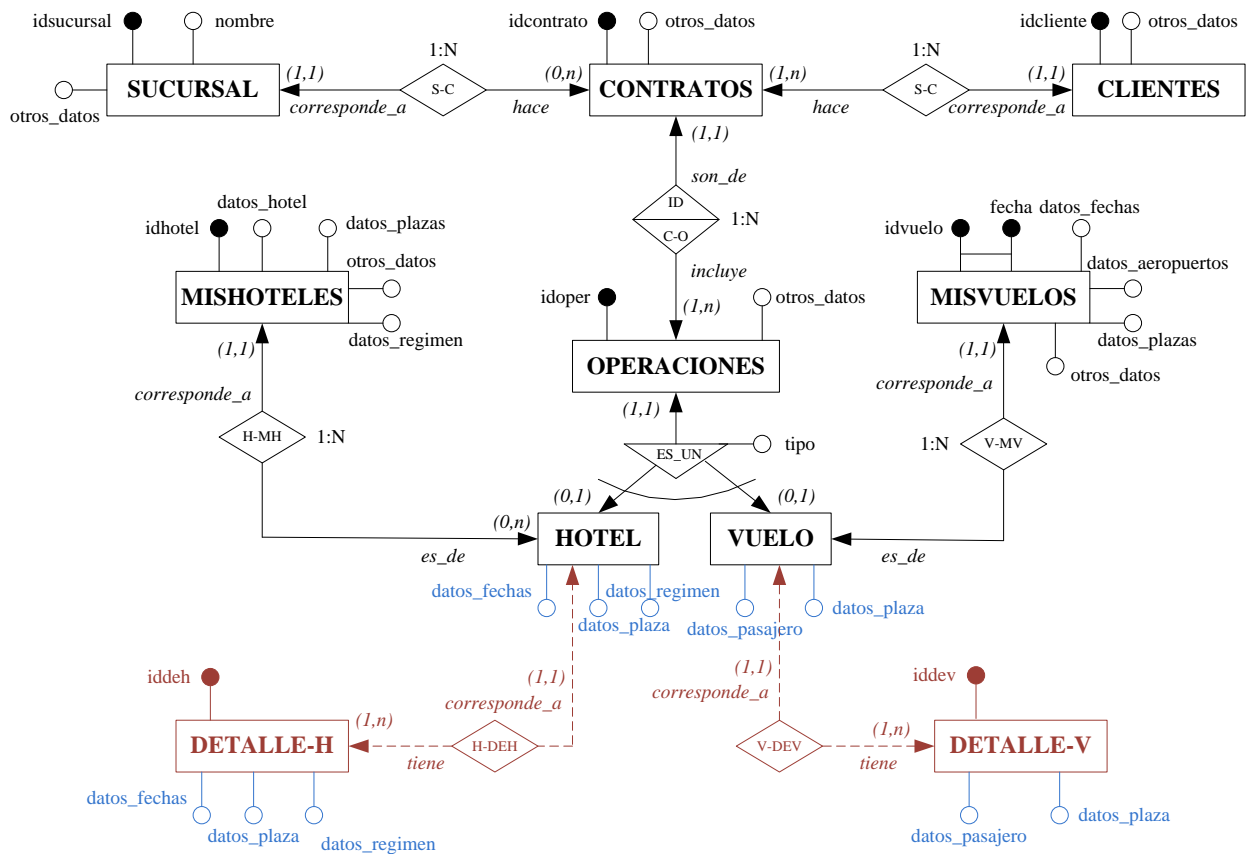




BASES DE DATOS

Una agencia de viajes desea disponer de una Base de Datos que contemple información relativa al hospedaje y vuelos de los clientes que contratan dichos servicios con ellos. La cadena cuenta con una serie de sucursales en todo su territorio nacional cuyos datos y dirección son utilizados en los medios de difusión de la agencia. Por su parte la agencia dispone de hoteles y vuelos en exclusiva que ofrece a sus clientes. Cada hotel estará definido por el código de hotel, nombre, dirección, ciudad, teléfono y número de plazas/habitaciones disponibles y tipo de servicios (alojamiento AD, MP, PC). De igual forma, la cadena tiene contratados una serie de vuelos regulares de forma exclusiva. Cada vuelo viene definido por el número de vuelo, fecha y hora, origen y destino, plazas business y turista de las que dispone. Cuando las sucursales realizan una venta de sus servicios se requiere conocer los datos del cliente que los contrata, los servicios contratados y, en el caso de los vuelos, las personas involucradas.

- Realizar el modelo conceptual y realizar las explicaciones necesarias
- Realizar el modelo relacional y realizar las explicaciones necesarias





BASES DE DATOS

Sucursal (idsucursal, nombre, otros_datos)

Cientes (idcliente, otros_datos)

Contratos (idcontrato, otros_datos, idcliente, idsucursal)

idcliente - FK (Cientes.idcliente)

idsucursal - FK (Sucursal.idsucursal)

idcliente NOT NULL

idsucursal NOT NULL

Operaciones (idcontrato, idoper, otros_datos, tipo)

idcontrato - FK (Contratos.idcontrato)

MisHoteles (idhotel, otros_datos)

MisVuelos (idvuelo, fecha, otros_datos)

Hotel (idoper, idhotel, otros_datos)

idoper - FK (Operaciones.idoper)

idhotel - FK (MisHoteles.idhotel)

idhotel NOT NULL

Vuelo (idoper, idvuelo, fecha, otros_datos)

idoper - FK (Operaciones.idoper)

(idvuelo, fecha) - FK (Misvuelos.(idvuelo, fecha))

idvuelo NOT NULL

fecha NOT NULL

DetalleH (iddeh, idoper, otros_datos)

idoper - FK (Hotel.idoper)

idoper, otros_datos NOT NULL

DetalleV (iddev, idoper, otros_datos)

idoper - FK (Vuelo.idoper)

idoper, otros_datos NOT NULL