



Tema 4. Diseño inverso de Bases de Datos

Objetivos

- Comprender la necesidad de un proceso de diseño inverso de una base de datos
- Ser capaz de obtener el modelo conceptual (EER) de una base de datos partiendo de su modelo lógico o físico.

Objetivos

- Comprender la necesidad de un proceso de diseño inverso de una base de datos
- **Ser capaz de obtener el modelo conceptual (EER) de una base de datos partiendo de su modelo lógico o físico.**

Del modelo relacional al
EER

Contenido

- Diseño inverso de base de datos: introducción y motivación
- Metodología para el diseño inverso de una base de datos
- Ejemplo: hípica
- Ejercicios

Diseño inverso de Bases de Datos

- ¿Qué es?
 - Es el proceso que nos permite obtener la representación conceptual de un DB a partir de su implementación.

Diseño de Bases de Datos

EER

La empresa tiene "juguetes" en vía de desarrollo y "juguetes" ya finalizados que son los que se comercializan. De todos los juguetes, incluso de los que están en vía de desarrollo, "se conoce" un código que los identifica, una descripción y, en el caso de los ya comercializados, se conocen obligatoriamente tanto el precio de coste como el precio de venta. Para todos los juguetes se debe conocer obligatoriamente su tipo. En la empresa se desarrollan 3 tipos de juguetes: de mesa, de playa y de acción.

Nos interesa, para cada juguete que se está desarrollando, almacenar las versiones que se hacen del mismo. Cada versión se numerará correlativamente, comenzando por 1 para cada juguete que se está desarrollando. De cada versión, además de su número interesa la fecha en la que se comienza esa versión, la fecha en la que se da por finalizada y las observaciones de dicha versión.

La empresa tiene una serie de comerciales que se encargan de la distribución en distintas poblaciones de los juguetes finalizados. Cada población tiene un nombre que la identifica y se conoce su número de habitantes de edad inferior a 18 años. En cuanto a los comerciales, se identifican por su DNI y se conoce también su nombre, teléfono y dirección. De cada juguete nos interesa saber qué comercial se encarga de su venta y en qué localidades lo hace. De cada juguete se encarga exclusivamente un comercial, para todas las localidades en las que se vende.

De entre los empleados de la empresa, de los que se conoce su DNI que los identifica, su nombre, teléfono y dirección, además de los comerciales, están los desarrolladores de juguetes, los administrativos y los vigilantes. Cada empleado se puede dedicar sólo a una de estas categorías.

En cuanto a los desarrolladores de juguetes, existen 3 tipos: los de juguetes de mesa, juguetes de playa y juguetes de acción. Nos interesa saber qué juguetes desarrolla cada uno sabiendo que estos desarrolladores se agrupan en parejas (los 2 componentes del mismo tipo). De manera que cada pareja de desarrolladores sólo pueden estar desarrollando un juguete de su mismo tipo (se encargan de todas las versiones de ese juguete). Del mismo modo un juguete que se está desarrollando sólo se puede estar desarrollando por una pareja de desarrolladores del mismo tipo que el juguete.

Por último, de los vigilantes nos interesa saber sus turnos de trabajo. La empresa está distribuida en zonas, todas con un código que las identifica y una descripción, y nos interesa saber en el mes

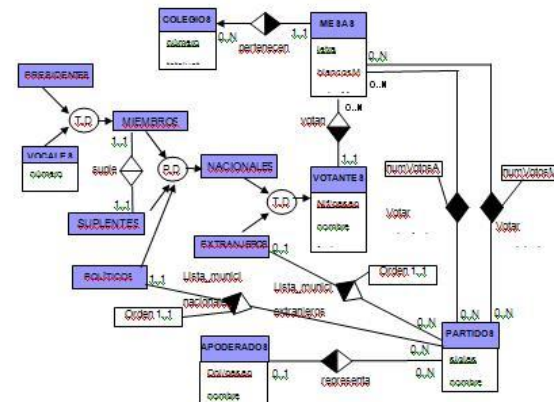
CONCEPTUAL

LÓGICO

FÍSICO

RELACIONAL

- COLEGIOS (num. totalvotantes)
C.P.: num
- MESAS (letra, cole, blq, num. blanco, num. nulo)
C.P.: (letra, cole)
Colegio: cole → COLEGIO
- VOTANTES (dni, nombre, fecha_nac, sexo, letra, colegio)
C.P.: dni
C. Ajen: (letra, colegio) → MESAS
V.N.N.: letra, colegio
- PARTIDOS (siglas, nombre, líder)
C.P.: siglas
- APODERADOS (dni, nombre, partido)
C.P.: dni
C. Ajen: partido → PARTIDOS
- NACIONALES (dni)
C.P.: dni
C. Ajen: dni → VOTANTES
- EXTRANJEROS (dni, go, partido, orden)
C.P.: dni
C. Ajen: dni → VOTANTES
C. Ajen: go, partido → PARTIDOS
- Si tiene valor la columna go, partido, debe tener valor la columna orden.
- No tiene sentido que tenga valor la columna orden y no lo tenga go, partido.
- DE_MESAS (dni titular, dni suplente)
C.P.: dni titular
C. Alternativa: dni suplente
C. Ajen: dni titular → NACIONALES
C. Ajen: dni suplente → NACIONALES
Se debe controlar que no aparezca el mismo dni en las dos columnas.
- POLÍTICOS (dni, go, partido, orden)
C.P.: dni
C. Ajen: dni → NACIONALES
C. Ajen: go, partido → PARTIDOS
V.N.N.: go, partido
V.N.N.: orden
- AUTONÓMICAS (partido, mesa, colegio, votos)
C.P.: (partido, mesa, colegio)
C. Ajen: partido → PARTIDOS
C. Ajen: (mesa, colegio) → MESAS
- MUNICIPALES (partido, mesa, colegio, votos)
C.P.: (partido, mesa, colegio)
C. Ajen: partido → PARTIDOS
C. Ajen: (mesa, colegio) → MESAS
- VOTANTES (dni, nombre, fecha_nac, sexo, letra, colegio)
C.P.: dni
C. Ajen: (letra, colegio) → MESAS
V.N.N.: letra, colegio
- PARTIDOS (siglas, nombre, líder)
C.P.: siglas
- APODERADOS (dni, nombre, partido)
C.P.: dni
C. Ajen: partido → PARTIDOS
- NACIONALES (dni)
C.P.: dni
C. Ajen: dni → VOTANTES
- EXTRANJEROS (dni, go, partido, orden)
C.P.: dni
C. Ajen: dni → VOTANTES
C. Ajen: go, partido → PARTIDOS
- Si tiene valor la columna go, partido, debe tener valor la columna orden.
- No tiene sentido que tenga valor la columna orden y no lo tenga go, partido.
- DE_MESAS (dni titular, dni suplente)
C.P.: dni titular
C. Alternativa: dni suplente
C. Ajen: dni titular → NACIONALES
C. Ajen: dni suplente → NACIONALES
Se debe controlar que no aparezca el mismo dni en las dos columnas.



```
--
-- RECURSO (Table)
--
CREATE TABLE RECURSO
(
  CODIGO          INTEGER,
  DESCRIPCION     VARCHAR2(100 BYTE),
  FALTA           DATE,
  TAMAÑO          NUMBER(8,2),
  TIEMPO_DES      NUMBER(8,2),
  NOMBRE_FORMATO  VARCHAR2(10 BYTE)
)
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;
```

```
ALTER TABLE RECURSO_GRATUITO
DROP PRIMARY KEY CASCADE;

DROP TABLE RECURSO_GRATUITO CASCADE CONSTRAINTS;

--
-- RECURSO_GRATUITO (Table)
--
CREATE TABLE RECURSO_GRATUITO
(
  CODIGO INTEGER
)
```

ORACLE

Diseño inverso de Bases de Datos

```
--
-- RECURSO (Table)
--
CREATE TABLE RECURSO
(
  CODIGO          INTEGER,
  DESCRIPCION     VARCHAR2(100 BYTE),
  FALTA           DATE,
  TAMAÑO          NUMBER(8,2),
  TIEMPO_DES      NUMBER(8,2),
  NOMBRE_FORMATO  VARCHAR2(10 BYTE)
)
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;

ALTER TABLE RECURSO_GRATUITO
  DROP PRIMARY KEY CASCADE;

DROP TABLE RECURSO_GRATUITO CASCADE CONSTRAINTS;

--
-- RECURSO_GRATUITO (Table)
--
CREATE TABLE RECURSO_GRATUITO
(
  CODIGO INTEGER
)
```

ORACLE

DEL FÍSICO
AL LÓGICO

- COLEGIOS (num, totalvotantes)
C.P.: num
- MESAS (letra, cola, blanco, blanco, num, num)
C.P.: (letra, cola)
C.Ajena: cola → COLEGIO
- VOTANTES (dni, nombre, fecha_nac, direccion, letra, colegio)
C.P.: dni
C.Ajena: (letra, colegio) → MESAS
V.N.N.: letra, colegio
- PARTIDOS (siglas, nombre, lider)
C.P.: siglas
- APODERADOS (dni, nombre, partido)
C.P.: dni
C.Ajena: partido → PARTIDOS
- NACIONALES (dni)
C.P.: dni
C.Ajena: dni → VOTANTES
- EXTRANJEROS (dni, go, partido, orden)
C.P.: dni
C.Ajena: dni → VOTANTES
C.Ajena: go, partido → PARTIDOS
- Si tiene valor la columna go, partido, debe tener valor la columna orden.
- No tiene sentido que tenga valor la columna orden y no lo tenga go, partido.
- DE_MESAS (dni, letra, cola, blanco)
C.P.: dni, letra
C.Ajena: dni, letra → NACIONALES
C.Ajena: dni, letra → NACIONALES
Se debe controlar que no aparezca el mismo dni en las dos columnas.
- POLÍTICOS (dni, go, partido, orden)
C.P.: dni
C.Ajena: dni → NACIONALES
C.Ajena: go, partido → PARTIDOS
V.N.N.: go, partido, V.N.N.: orden
- AUTONOMICAS (partido, mesa, colegio, votos)
C.P.: (partido, mesa, colegio)
C.Ajena: partido → PARTIDOS
C.Ajena: (mesa, colegio) → MESAS
- MUNICIPALES (partido, mesa, colegio, votos)
C.P.: (partido, mesa, colegio)
C.Ajena: partido → PARTIDOS
C.Ajena: (mesa, colegio) → MESAS
- VOCALES (dni, número)
C.P.: dni
C.Ajena: dni → DE_MESAS
- PRESIDENTES (dni)
C.P.: dni
C.Ajena: dni → DE_MESAS
- No se refleja que las generalizaciones sean totales ni tampoco que sean disjuntas. |

RELACIONAL

DEL LÓGICO AL
CONCEPTUAL

¿Cuál sería el
esquema conceptual
correspondiente?

EER

Diseño inverso de Bases de Datos

```
--
-- RECURSO (Table)
--
CREATE TABLE RECURSO
(
  CODIGO           INTEGER,
  DESCRIPCION      VARCHAR2(100 BYTE),
  FALTA            DATE,
  TAMANYO          NUMBER(8,2),
  TIEMPO_DES       NUMBER(8,2),
  NOMBRE_FORMATO   VARCHAR2(10 BYTE)
)
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;
```

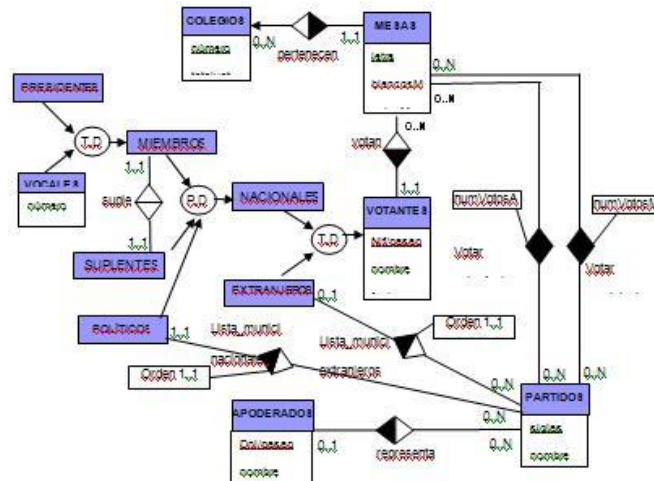
```
ALTER TABLE RECURSO_GRATUITO
    DROP PRIMARY KEY CASCADE;

DROP TABLE RECURSO_GRATUITO CASCADE CONSTRAINTS;
```

```
--
-- RECURSO_GRATUITO (Table)
--
CREATE TABLE RECURSO_GRATUITO
(
  CODIGO INTEGER
)
```

ORACLE

DEL FÍSICO AL LÓGICO



EER

- **COLEGIOS** (num. total, votantes)
C.P.: num
 - **MESAS** (letra, cole, blanq, blanq,
num, num)
C.P.: (letra, cole)
C. Ajena: cole → COLEGIO
 - **VOTANTES** (dni, nombre, fecha,
siglas, cole, letra, colegio)
C.P.: dni
C. Ajena: (letra, colegio) → MESAS
V.N.N.: letra, colegio
 - **PARTIDOS** (siglas, nombre, lider)
C.P.: siglas
 - **APODERADOS** (dni, nombre, partido)
C.P.: dni
C. Ajena: partido → PARTIDOS
 - **NACIONALES** (dni)
C.P.: dni
C. Ajena: dni → VOTANTES
 - **EXTRANJEROS** (dni, partido, orden)
C.P.: dni
C. Ajena: dni → VOTANTES
C. Ajena: partido → PARTIDOS
- Si tiene valor la columna partido, debe tener valor la columna orden.
- No tiene sentido que tenga valor la columna orden y no lo tenga partido.
- **DE_MESAS** (dni, valor, disruptivo)
C.P.: disruptivo
C. Alternativa: disruptivo
C. Ajena: disruptivo → NACIONALES
C. Ajena: disruptivo → NACIONALES
- Se debe controlar que no aparezca el mismo dni en las dos columnas.
- **POLÍTICOS** (dni, partido, orden)
C.P.: dni
C. Ajena: dni → NACIONALES
C. Ajena: partido → PARTIDOS
V.N.N.: partido
V.N.N.: orden
 - **AUTONOMICAS** (partido, mesa, colegio, votos)
C.P.: (partido, mesa, colegio)
C. Ajena: partido → PARTIDOS
C. Ajena: (mesa, colegio) → MESAS
 - **MUNICIPALES** (partido, mesa, colegio, votos)
C.P.: (partido, mesa, colegio)
C. Ajena: partido → PARTIDOS
C. Ajena: (mesa, colegio) → MESAS
 - **VOCALES** (dni, número)
C.P.: dni
C. Ajena: dni → DE_MESAS
 - **PRESIDENTES** (dni)
C.P.: dni
C. Ajena: dni → DE_MESAS
- No se refleja que las generalizaciones sean totales ni tampoco que sean disjuntas. |
- RELACIONES**

RELACIONAL

DEL LÓGICO AL
CONCEPTUAL

Diseño inverso de Bases de Datos: Motivación

Tienes el Grado en Ingeniería Informática y te acaban de contratar en una **hípica** para que te encargues de toda la gestión de su información.

Los encargados de gestionar la base de datos se han marchado. La base de datos ya está creada ... pero no han dejado nada de documentación.

Como eres Ingeniero/a Informático/a, asumen que con un vistazo tendrás claro cómo está relacionada toda la información ... ¿O no? ¿O tienen que volver a contarte todas las especificaciones en detalle?

La base de datos que te dan

CABALLO(cod, nombre, raza, fechanac)

C.P.:cod

V.N.N.: nombre

CARRERA(código, edición, fecha, ganador)

C.P. (código, edición)

C. Ajena: ganador → CABALLO

ACTUALES(caballo, box, zona)

C.P. caballo

C. Alternativa: box, zona

C. Ajena: caballo → CABALLO

C. Ajena: (box, zona) → BOX

ZONA_BOX(letra)

C.P. : letra

BOX(número, letra, dimensiones)

C.P. : (número, letra)

C. Ajena: letra → ZONA_BOX

HORARIO(fecha, hora)

C.P. : (fecha, hora)

DEL_CENTRO(caballo)

C.P. : caballo

C. Ajena: caballo → ACTUALES

DEL_CLIENTE(caballo, alimentadueño)

C.P. : caballo

C. Ajena: caballo → ACTUALES

con CHECK para alimentadueño (SI o NO)

La base de datos que te dan

PERSONAL(NIF, nombre, dirección, teléfono, fechanac, estudios)

C.P.: NIF

MOZO (NIF, zona)

C.P.: NIF

C. Ajena: NIF → PERSONAL

C. Ajena: zona → ZONA_BOX

VETERINARIO (NIF)

C.P. NIF

C. Ajena: NIF → PERSONAL

CLIENTE(NIF, nombre, dirección, teléfono, fecha_alta)

C.P. : NIF

V.N.N.: fecha_alta

PROFESOR (NIF)

C.P. NIF

C. Ajena: NIF → PERSONAL

CURSILLO (código, finicio, ffin, max_alumnos, profesor)

C.P. : código

C. Ajena: profesor → PROFESOR

NIVEL(número, descripción, precio)

C.P.: número

La base de datos que te dan

CLASE (fecha, hora, profesor, nivel, caballo, cliente)

C.P: (fecha, hora, profesor)

C. Ajena: profesor → PROFESOR

C. Ajena: (fecha, hora) → HORARIO

C. Ajena: nivel → NIVEL

C. Ajena: cliente → CLIENTE

C. Ajena: caballo → ACTUALES

V.N.N.: nivel

V.N.N.: cliente

V.N.N.: caballo

MONTAR(caballo, cliente, cursillo)

C.P: (caballo, cliente)

C. Alternativa: (caballo, cursillo)

C. Alternativa: (cliente, cursillo)

C. Ajena: cursillo → CURSILLO

C. Ajena: cliente → CLIENTE

C. Ajena: caballo → CABALLO

La base de datos que te dan

HEMBRA (codigo, yegua_madre, caballo_padre)

C.P.: código

C. Ajena: código → CABALLO

C. Ajena: yegua_madre → HEMBRA

C. Ajena: caballo_padre → MACHO

MACHO(codigo, yegua_madre, caballo_padre)

C.P.: código

C. Ajena: código → CABALLO

C. Ajena: yegua_madre → HEMBRA

C. Ajena: caballo_padre → MACHO

EN_PRADERA (código, fecha_parto)

C.P. código → YEGUA

La base de datos que te dan

Hay un disparador para controlar que los códigos que son de caballos del centro no son de caballos de clientes y viceversa.

En la aplicación de entrada de datos se observa que, al dar de alta un caballo, se fuerza a elegir obligatoriamente entre CENTRO o CLIENTE.

Hay un disparador para controlar que los códigos que son de hembra no son de macho y viceversa.

En la aplicación de entrada de datos se observa que, al dar de alta un caballo, se fuerza a elegir obligatoriamente entre HEMBRA o MACHO.

Diseño inverso de Bases de Datos: Metodología

Únicamente viendo las tablas ¿te haces una idea?

Lo más recomendable es que consigas el esquema conceptual que sirvió de partida al esquema relacional que te dan.

**ALGUNAS IDEAS A TENER EN
CUENTA EN LA TRANSFORMACIÓN
“INVERSA”**

¿De qué tabla extraemos que existía una nueva entidad en el diseño conceptual?

AULA(cod, superficie, cañón, aulario)

C.P.:cod

C. ajena: aulario → AULARIO

ASIGNATURA(codasig, descripción, credT, credP)

C.P.: codasig

IMPARTIR(asignatura, aula, día, hora)

C.P. (asignatura, aula)

C. ajena: asignatura → ASIGNATURA

C. ajena: aula → AULA

¿De qué tabla extraemos que existe una nueva entidad en el diseño conceptual?

AULA(cod, superficie, cañón, aulario)

C.P.:cod

SÍ

C. ajena: aulario → AULARIO

ASIGNATURA(codasig, descripción, credT, credP)

C.P.: codasig

SÍ

IMPARTIR(asignatura, aula, día, hora)

NO

C.P. (asignatura, aula) **La C.P. con las ajenas indica relación entre entidades ya existentes**

C. ajena: asignatura → ASIGNATURA

C. ajena: aula → AULA

Distinguir restricción de identificador

AULARIO (cod, num_plantas)

C.Primeria: cod

AULA (num, superficie, num_sillas, aulario)

C.Primeria: num

C.ajena: aulario → AULARIO

V.N.N.: aulario

Distinguir restricción de identificador

AULARIO (cod, num_plantas)

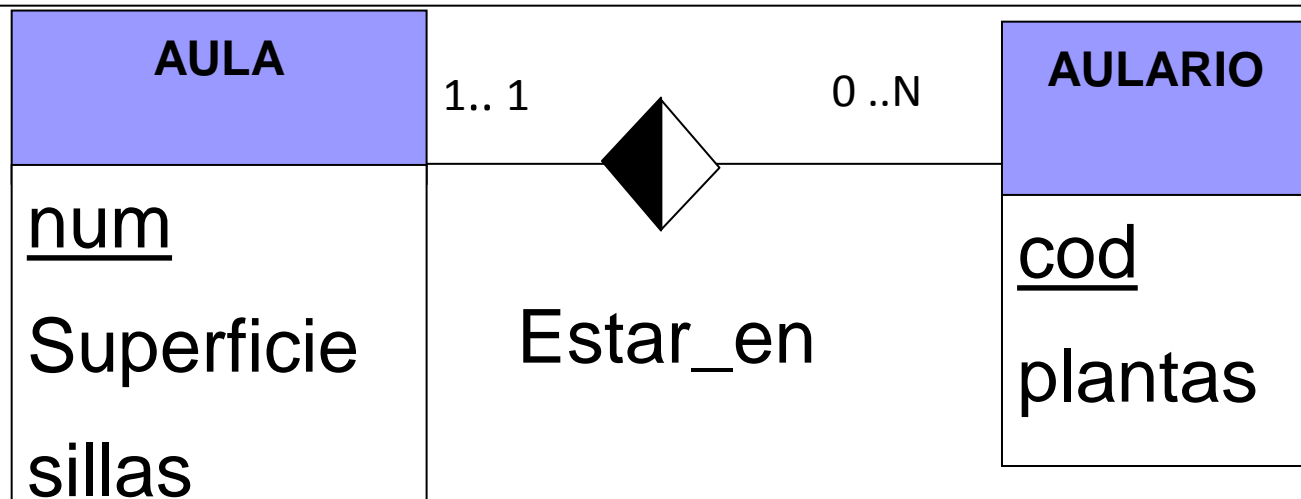
C.Primaria: cod

AULA (num, superficie, num_sillas, aulario)

C.Primaria: num

C.ajena: aulario → AULARIO

V.N.N.: aulario



Distinguir restricción de identificador

AULARIO (cod, num_plantas)

C.Primaria: cod

AULA (num, superficie, num_sillas, aulario)

C.Primaria: (num, aulario)

C.ajena: aulario → AULARIO

Distinguir restricción de identificador

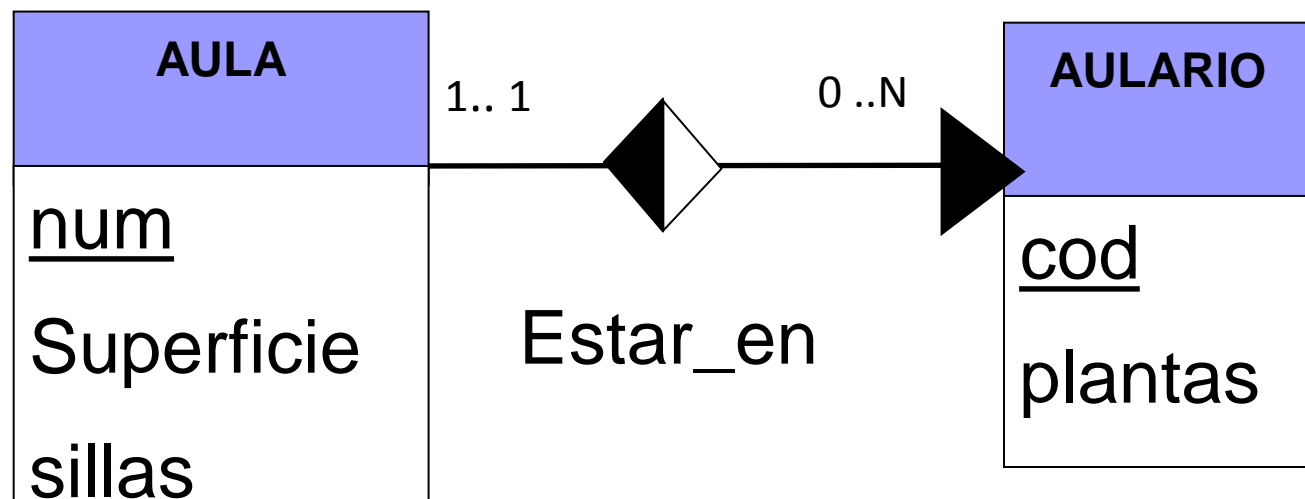
AULARIO (cod, num_plantas)

C.Primaria: cod

AULA (num, superficie, num_sillas, aulario)

C.Primaria:(num, aulario)

C.ajena: aulario → AULARIO



Distinguir si puede ser ternaria

Indica para cada una de las tablas que se muestran en la siguiente diapositiva, **si podrían o no estar representando una ternaria**, razonando tu respuesta en cada caso.

Las claves ajenas de cada tabla se supone que están bien definidas y que referencian a tablas ya creadas, que aunque se llamen igual pueden ser distintas en cada caso.

¿Podrían o no estar representando una ternaria?

~~R (r1, r2, r3, r4)~~
~~C. primaria: r1, r2~~
~~C. alternativa: r3, r4~~
~~C. ajena: r1 → A~~
~~C. ajena: r2 → B~~
~~C. Ajena: r3, r4 → C~~

~~R (r1, r2, r3, r4)~~
~~C. primaria: r1, r2~~
~~C. alternativa: r3~~
~~C. ajena: r1 → A~~
~~C. ajena: r2 → B~~
~~C. Ajena: r3 → C~~

R (r1, r2, r3, r4)
C. primaria: r1, r2
VNN: r3, r4
C. ajena: r1 → A
C. ajena: r2 → B
C. Ajena: r3, r4 → C



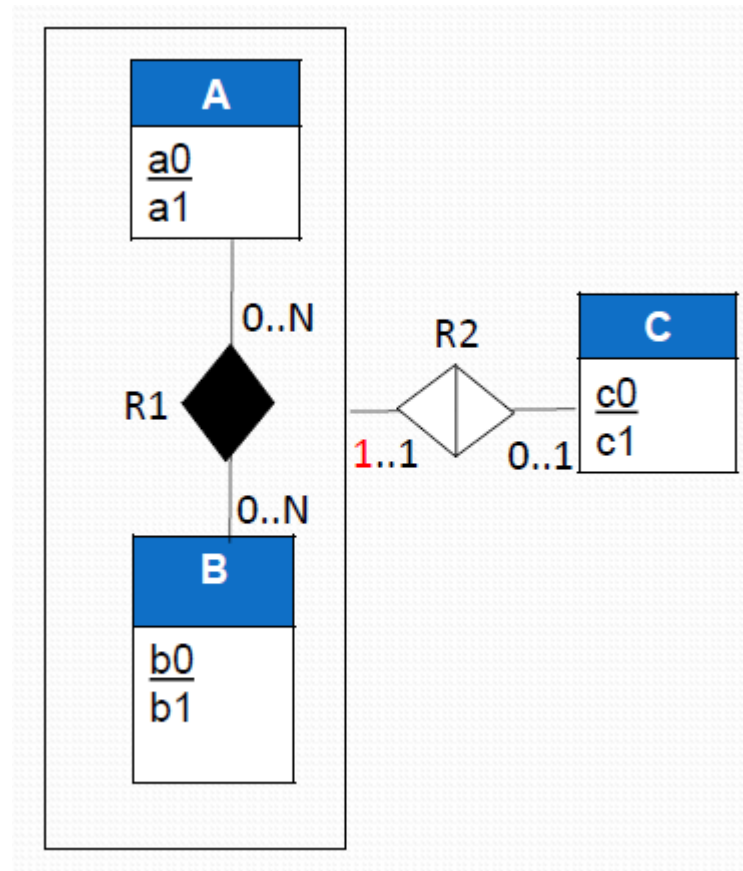
R (r1, r2, r3, r4)
C. primaria: r1, r2
C. alternativa: r1, r2
C. ajena: r1 → A
C. ajena: r2 → B
C. Ajena: r3 → C



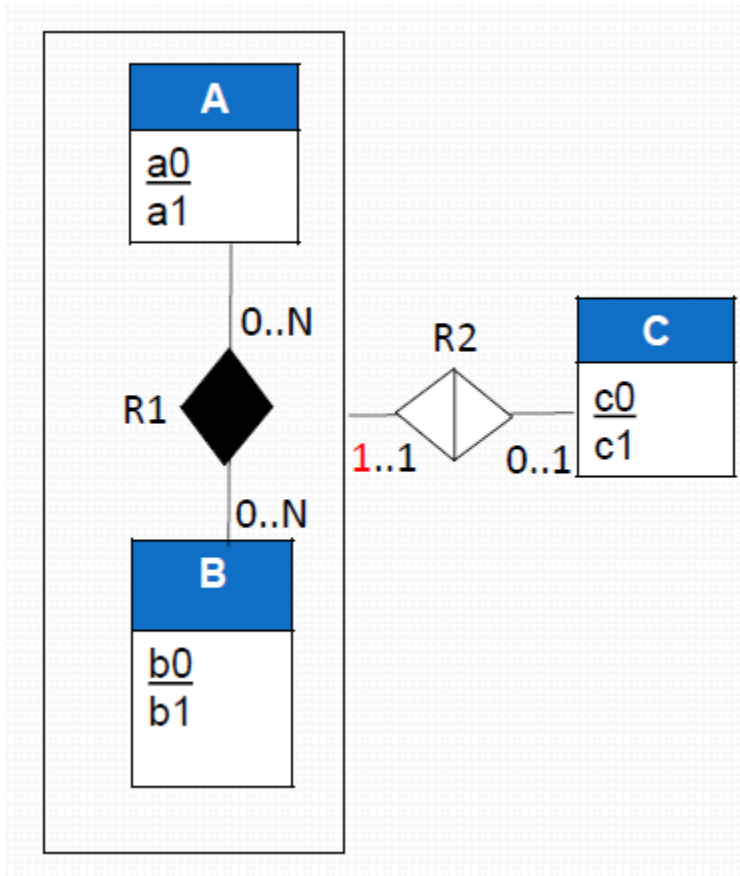
¿Podrían o no estar representando una ternaria?

NO

R (r1, r2, r3, r4)
C. primaria: r1, r2
C. alternativa: r3, r4
C. ajena: r1 → A
C. ajena: r2 → B
C. Ajena: r3, r4 → C



¿Podrían o no estar representando una ternaria?



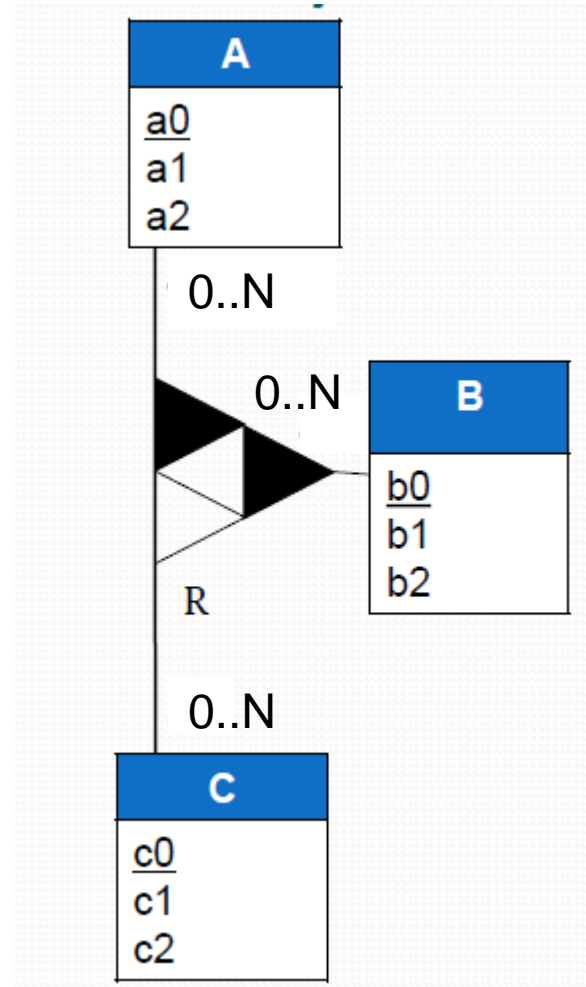
NO

R (r1, r2, r3, r4)
C. primaria: r1, r2
C. alternativa: r3
C. ajena: r1 → A
C. ajena: r2 → B
C. Ajena: r3 → C

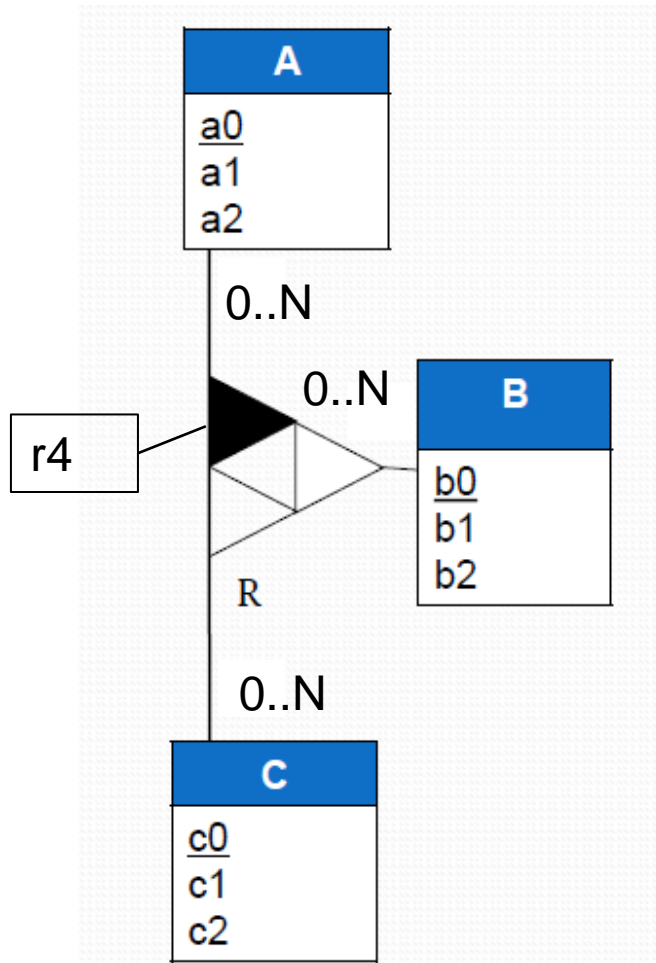
¿Podrían o no estar representando una ternaria?

SÍ

R (r1, r2, r3, r4)
C. primaria: r1, r2
VNN: r3, r4
C. ajena: r1 → A
C. ajena: r2 → B
C. Ajena: r3, r4 → C



¿Podrían o no estar representando una ternaria?



SÍ

$R(r1, r2, r3, r4)$
C. primaria: $r1, r2$
C. alternativa: $r1, r3$
C. ajena: $r1 \rightarrow A$
C. ajena: $r2 \rightarrow B$
C. Ajena: $r3 \rightarrow C$

Teniendo en cuenta estas aclaraciones y todo lo que sabes de diseño lógico, volvamos a analizar el esquema relacional de la hípica para obtener su esquema conceptual de partida.

ENCONTREMOS EL ESQUEMA CONCEPTUAL (EER) DE PARTIDA

CABALLO(cod, nombre, raza, fechanac)

C.P.:cod

V.N.N.: nombre

ACTUALES(caballo, box, zona)

C.P. caballo

C. Alternativa: box, zona

C. Ajena: caballo → CABALLO

C. Ajena: (box, zona) → BOX

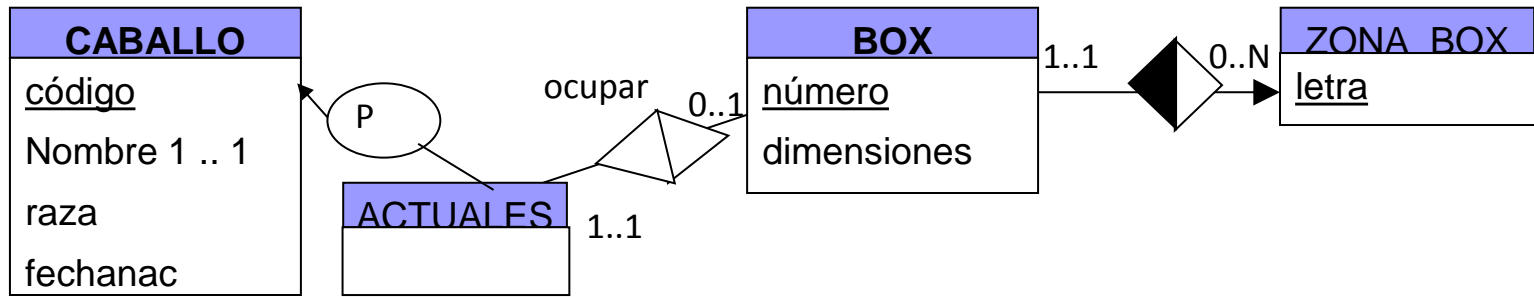
ZONA_BOX(letra)

C.P. : letra

BOX(número, letra, dimensiones)

C.P. : (número, letra)

C. Ajena: letra → ZONA_BOX



DEL_CENTRO(caballo)

C.P. : caballo

C. Ajena: caballo → ACTUALES

DEL_CLIENTE(caballo, alimentadueño)

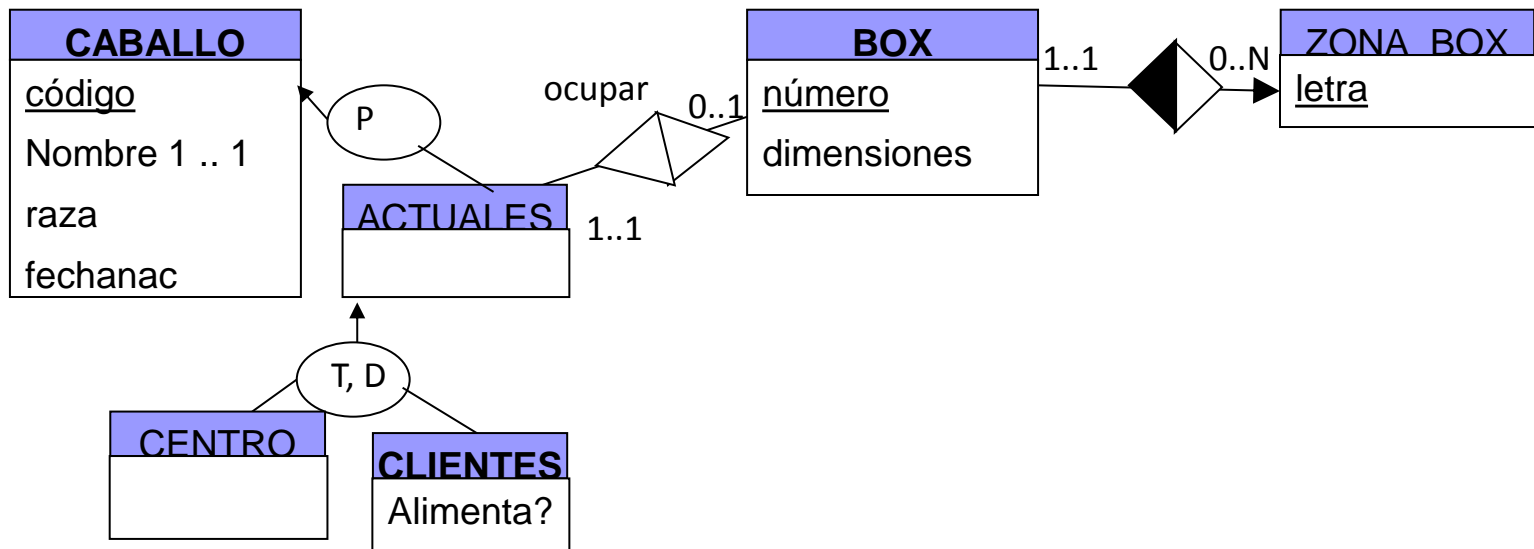
C.P. : caballo

C. Ajena: caballo → ACTUALES

con CHECK para alimentadueño (SI o NO)

Hay un disparador para controlar que los códigos que son de caballos del centro no son de caballos de clientes y viceversa.

En la aplicación de entrada de datos se observa que, al dar de alta un caballo, se fuerza a elegir obligatoriamente entre CENTRO o CLIENTE.



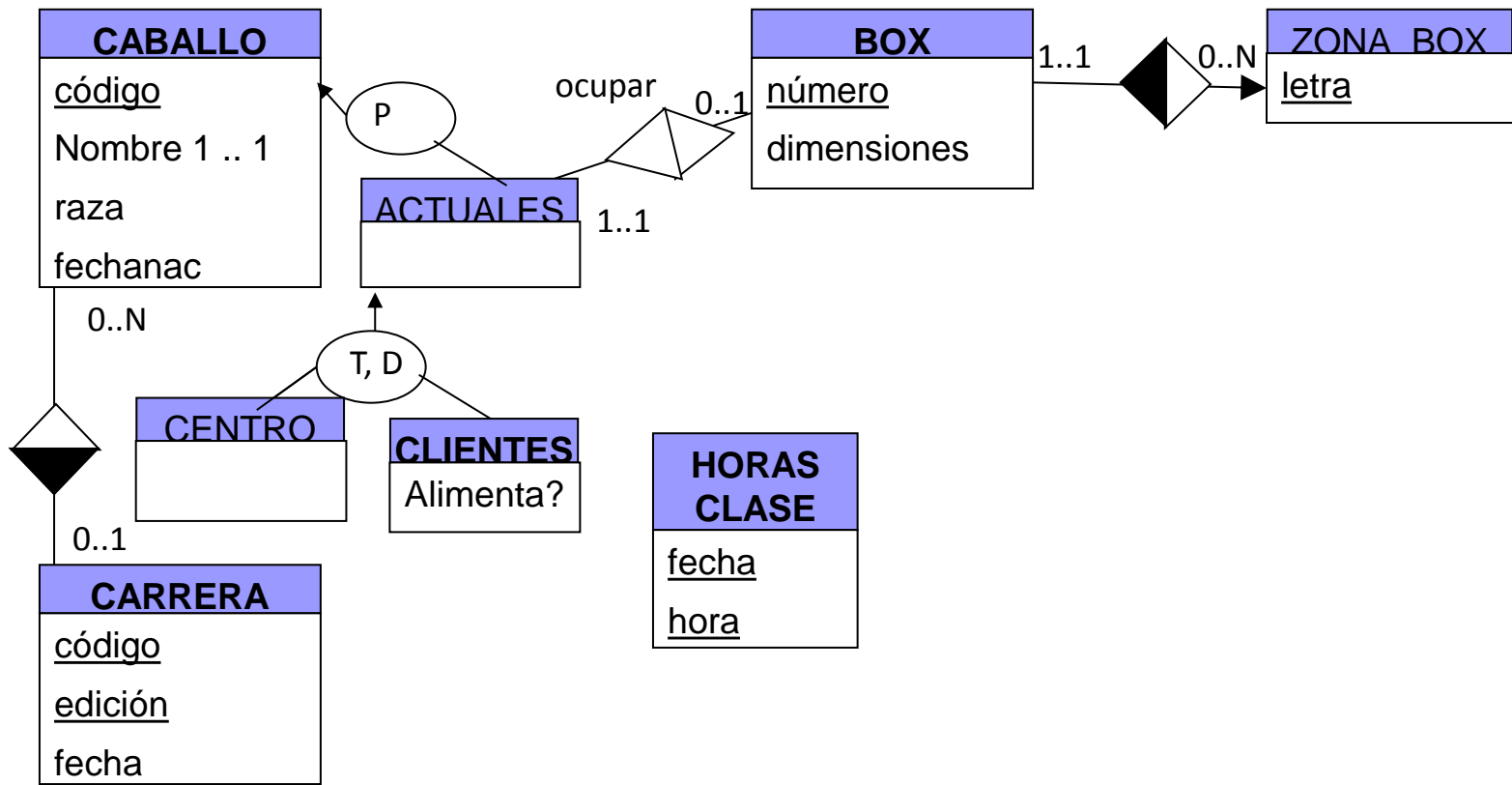
CARRERA(código, edición, fecha, ganador)

C.P. (código, edición)

C. Ajena: ganador → CABALLO

HORARIO(fecha,
hora)

C.P. : (fecha, hora)



PERSONAL(NIF, nombre, dirección,
teléfono, fechanac, estudios)

C.P.: NIF

VETERINARIO (NIF)

C.P. NIF

C. Ajena: NIF → PERSONAL

PROFESOR (NIF)

C.P. NIF

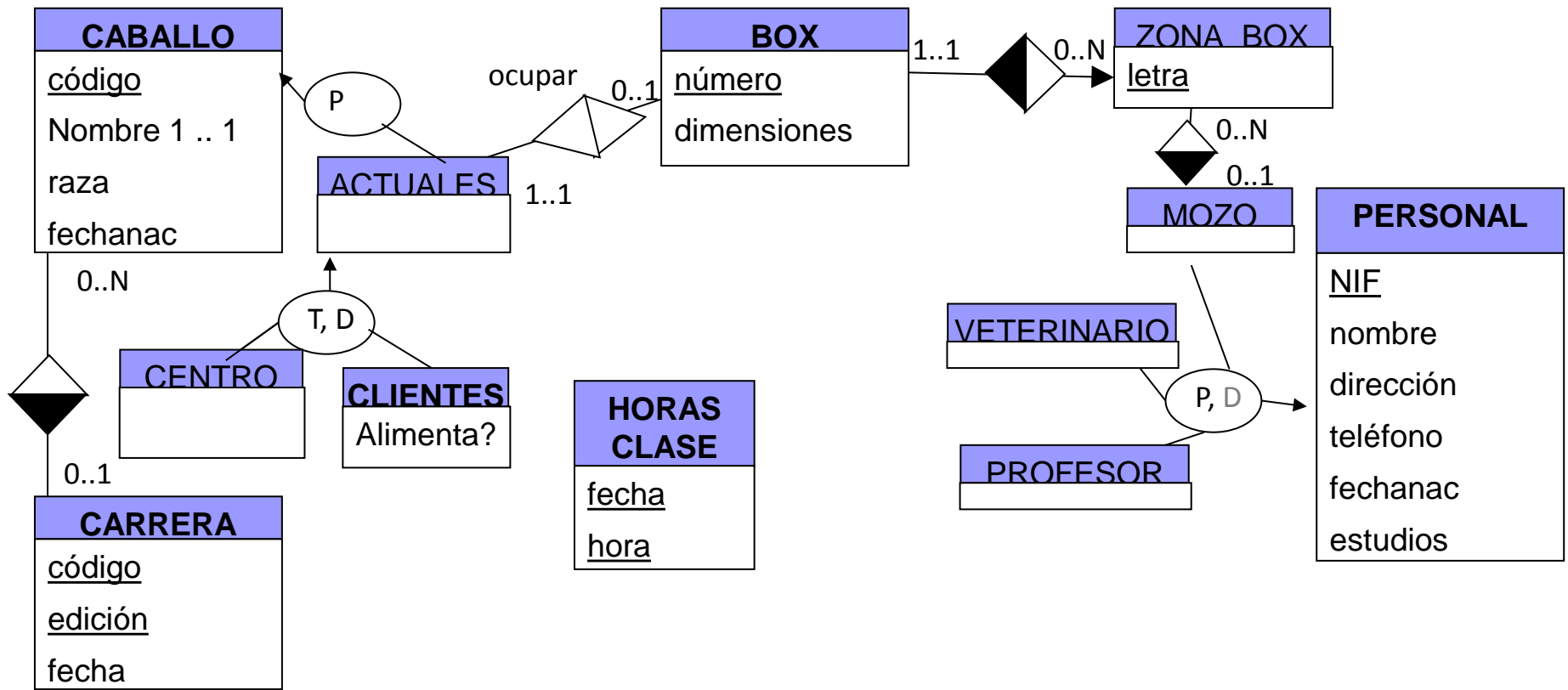
C. Ajena: NIF → PERSONAL

MOZO (NIF, zona)

C.P.: NIF

C. Ajena: NIF → PERSONAL

C. Ajena: zona → ZONA_BOX



CLIENTE(NIF, nombre, dirección, teléfono, fecha_alta)

C.P. : NIF

V.N.N.: fecha_alta

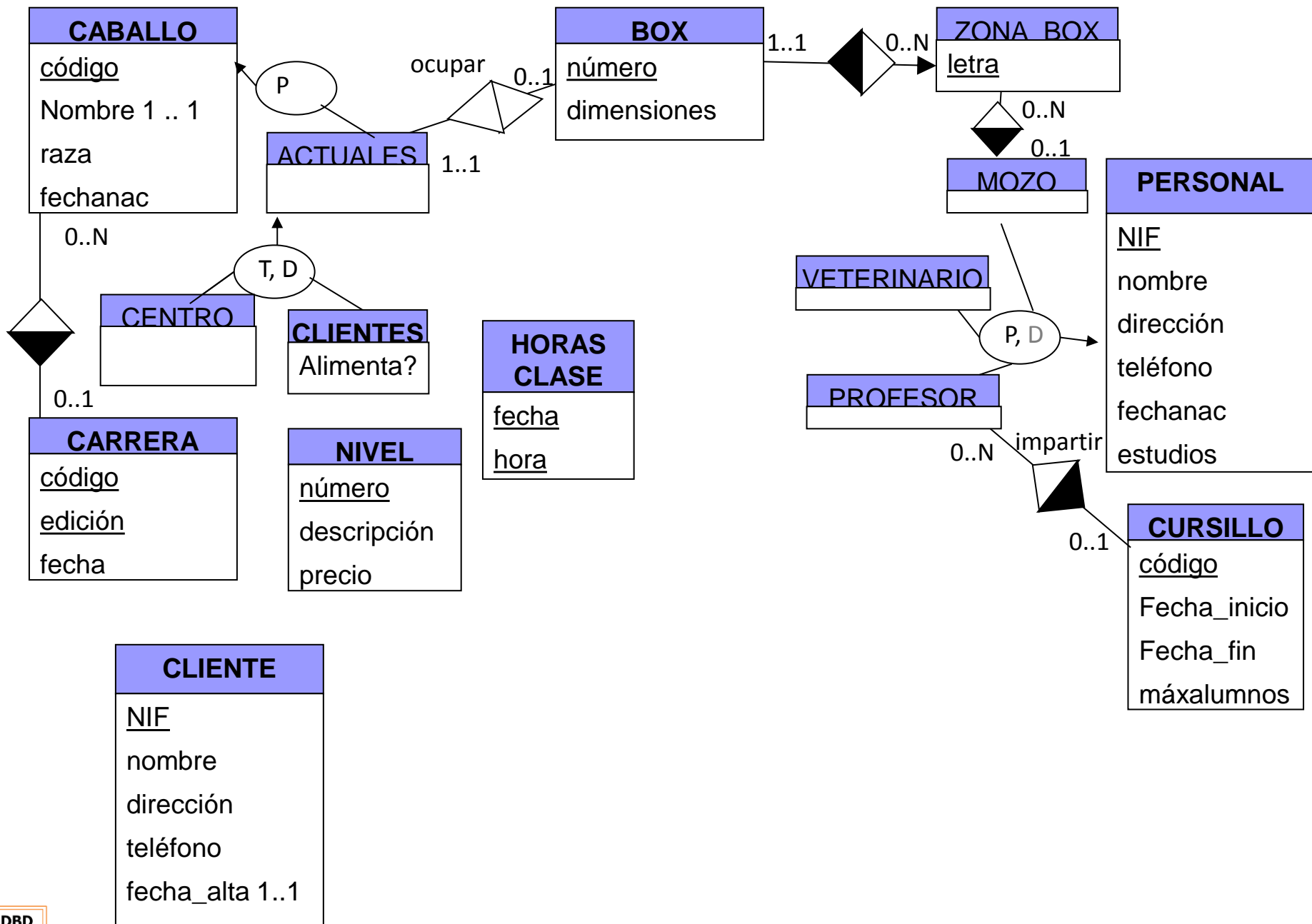
CURSILLO (código, finicio, ffin,
max_alumnos, profesor)

C.P. : código

C. Ajena: profesor → PROFESOR

NIVEL(número, descripción, precio)

C.P.: número



MONTAR(caballo, cliente, cursillo)

C.P: (caballo, cliente)

C. Alternativa: (caballo, cursillo)

C. Alternativa: (cliente, cursillo)

C. Ajena: cursillo → CURSILLO

C. Ajena: cliente → CLIENTE

C. Ajena: caballo → CABALLO

CLASE (fecha, hora, profesor, nivel, caballo, cliente)

C.P: (fecha, hora, profesor)

C. Ajena: profesor → PROFESOR

C. Ajena: (fecha, hora) → HORARIO

C. Ajena: nivel → NIVEL

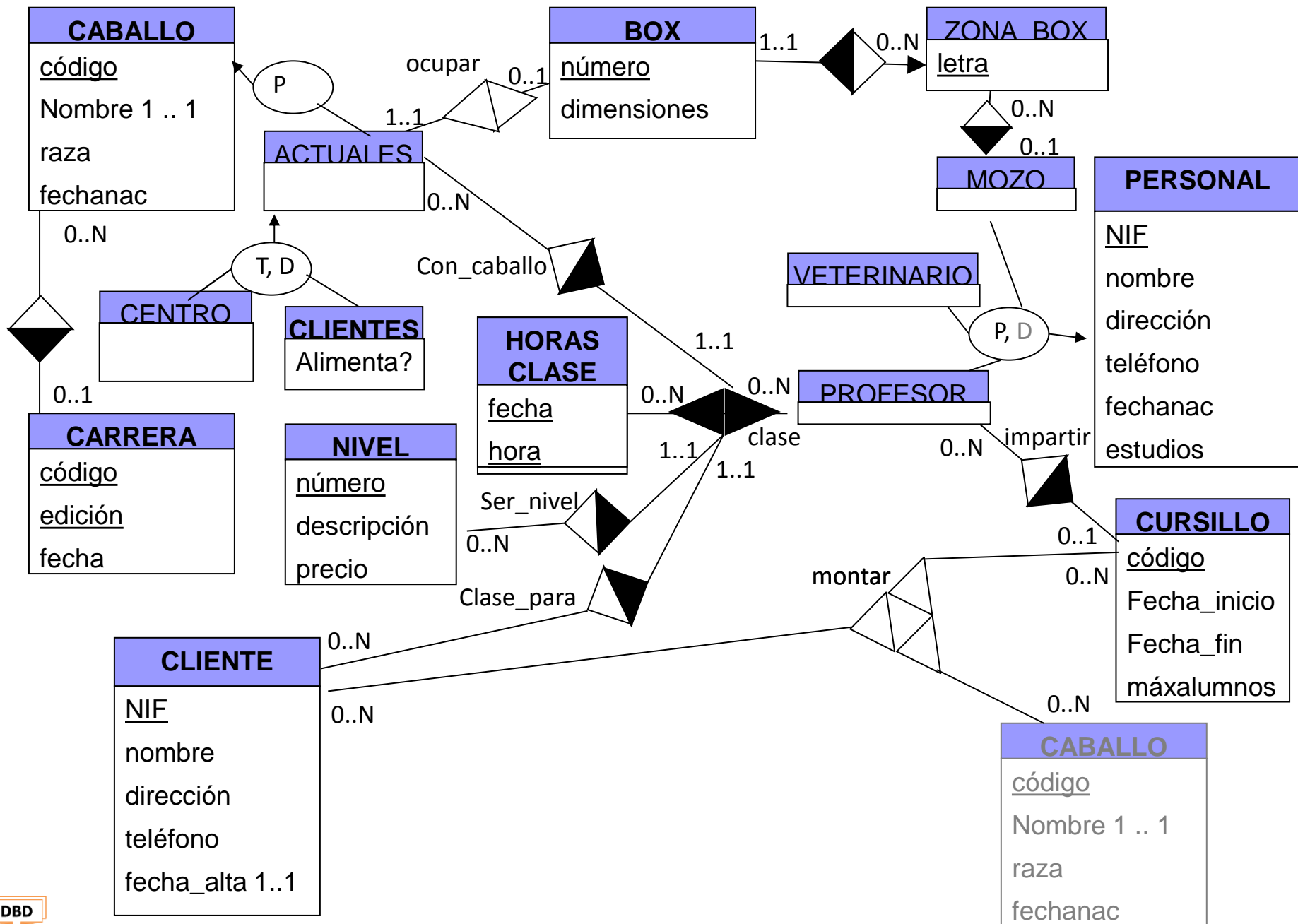
C. Ajena: cliente → CLIENTE

C. Ajena: caballo → ACTUALES

V.N.N.: nivel

V.N.N.: cliente

V.N.N.: caballo



HEMBRA (codigo, yegua_madre, caballo_padre)

C.P.: código

C. Ajena: código → CABALLO

C. Ajena: yegua_madre → HEMBRA

C. Ajena: caballo_padre → MACHO

MACHO(codigo, yegua_madre, caballo_padre)

C.P.: código

C. Ajena: código → CABALLO

C. Ajena: yegua_madre → HEMBRA

C. Ajena: caballo_padre → MACHO

EN_PRADERA (código, fecha_parto)

C.P. código

C.Aj. código → HEMBRA

Descripción de los requerimientos

Los caballos de este centro tienen un código que los identifica y, además obligatoriamente un nombre, pudiendo conocerse también su raza y su fecha de nacimiento. De los caballos (el término se usa tanto para yeguas como caballos ...) nos interesan los que están actualmente, y de éstos se distinguen los que son propiedad del centro y los que son propiedad de algunos clientes..

En el centro de equitación hay boxes que se numeran comenzando por 1 dentro de cada zona de boxes y se conoce de cada uno sus dimensiones. Cada zona de boxes se identifica por una letra.

Descripción de los requerimientos

Nos interesa conocer qué caballo ocupa actualmente cada box (un box como máximo lo ocupa un caballo). En cuanto a la ocupación de boxes, sólo se mantendrá la actual, pero los datos (código, nombre ...) de los caballos que han pasado por el centro se mantendrá, independientemente de que ya no estén. En el caso del alojamiento (pupilage), puede ser que los propietarios de los caballos decidan ser ellos los que van a dar la comida. En el centro se debe conocer para los caballos alojados actualmente, los que van a ser alimentados por sus dueños y los que no.

En cuanto a las carreras de caballos, todas están identificadas por un código y una edición. De cada carrera se conoce su fecha de comienzo. En el caso de que ya se haya realizado la carrera y haya ganado algún caballo relacionado con el centro, se sabrá cuál ha sido el ganador.

.

Descripción de los requerimientos

En lo referente a la enseñanza, se distinguen clases particulares y cursillos. Las clases particulares se contratan indicando siempre un nivel, según la experiencia requerida a los alumnos. Cada nivel se identifica por un número y se conoce una descripción de lo que implica ese nivel. Los profesores de equitación se dedican exclusivamente a las clases, a ninguna otra tarea. Se conocerán los precios por clase para cada nivel, pudiendo variar de un nivel a otro. Para optimizar los huecos, se han establecido unos horarios de comienzo de las clases, cada hora de cada día, desde las 9:00 hasta las 20:00. En la BD se debe controlar que un profesor no esté a la vez asignado a más de una clase particular. Al dar de alta una clase se conoce obligatoriamente cuando se impartirá, el alumno que la recibirá, el profesor y el caballo con el que se realizará la clase (ya se encargarán de que el caballo esté desocupado).

Descripción de los requerimientos

Por otro lado están los cursillos. Se identifican por un código y se conoce la fecha de inicio, la de finalización, una descripción, y el número máximo de alumnos que admite. Cada cursillo lo imparte un único profesor, y en un cursillo un alumno únicamente monta un caballo, al igual que un caballo en un cursillo sólo lo montará un alumno (ya se encargarán de que tanto profesor como caballo no estén ocupados en ese horario). Mientras que de las clases particulares se mantiene sólo la información actual sobre clases previstas, de los cursillos se mantiene la información relativa a todos los cursillos realizados ya que se pretende que un alumno no se acostumbre a un caballo, por lo que un alumno no montará el mismo caballo en dos cursillos distintos.

Descripción de los requerimientos

En cuanto al cruce de caballos nos va a interesar conocer las relaciones familiares de los caballos que hayan estado en algún momento dados de alta en el centro, independientemente de que sean de su propiedad o no. Es decir, dado un caballo nos interesará saber si han estado en el centro su padre/madre o hijo/hija.

El centro cuenta con una pequeña pradera donde viven gran parte del día las yeguas preñadas, en una situación de semilibertad. Nos interesa saber cuáles son las yeguas que se encuentran actualmente ocupando la pradera (además mantienen su box), y la fecha prevista del parto.

Descripción de los requerimientos

Del personal del centro conocemos su NIF, nombre, dirección, teléfono, fecha de nacimiento y estudios realizados. Aparte del profesorado, hay: mozos, veterinarios, administrativos, ...

Los mozos se encargan de la comida y limpieza de los caballos. Cada mozo está asignado a una zona de boxes, pudiendo estar varios asignados a la misma zona. En cuanto a los veterinarios, aunque revisan a todos los caballos de forma rutinaria, interesa que la base de datos refleje si algún veterinario está haciendo actualmente un seguimiento especial de algún caballo, en ese caso se deberá conocer obligatoriamente los motivos que causan este seguimiento.

De los clientes del centro (alumnos, propietarios de caballos) se conoce su NIF, nombre, domicilio, teléfono de contacto, y obligatoriamente, la fecha en la que se dieron de alta como clientes.

Diseño inverso de bases de datos

- Ejercicios sobre diseño lógico y diseño inverso



Tema 4. Diseño inverso de Bases de Datos