

Profesor:	Lourdes Moreno López Francisco Javier Calle Gómez	Grupo	82
Alumno/a:	Gabriel García Martínez	NIA:	100429061
Alumno/a:	Jorge Payno Zarceño	NIA:	100429114
Alumno/a	Cosmin Silviu Carsmar	NIA:	100429096

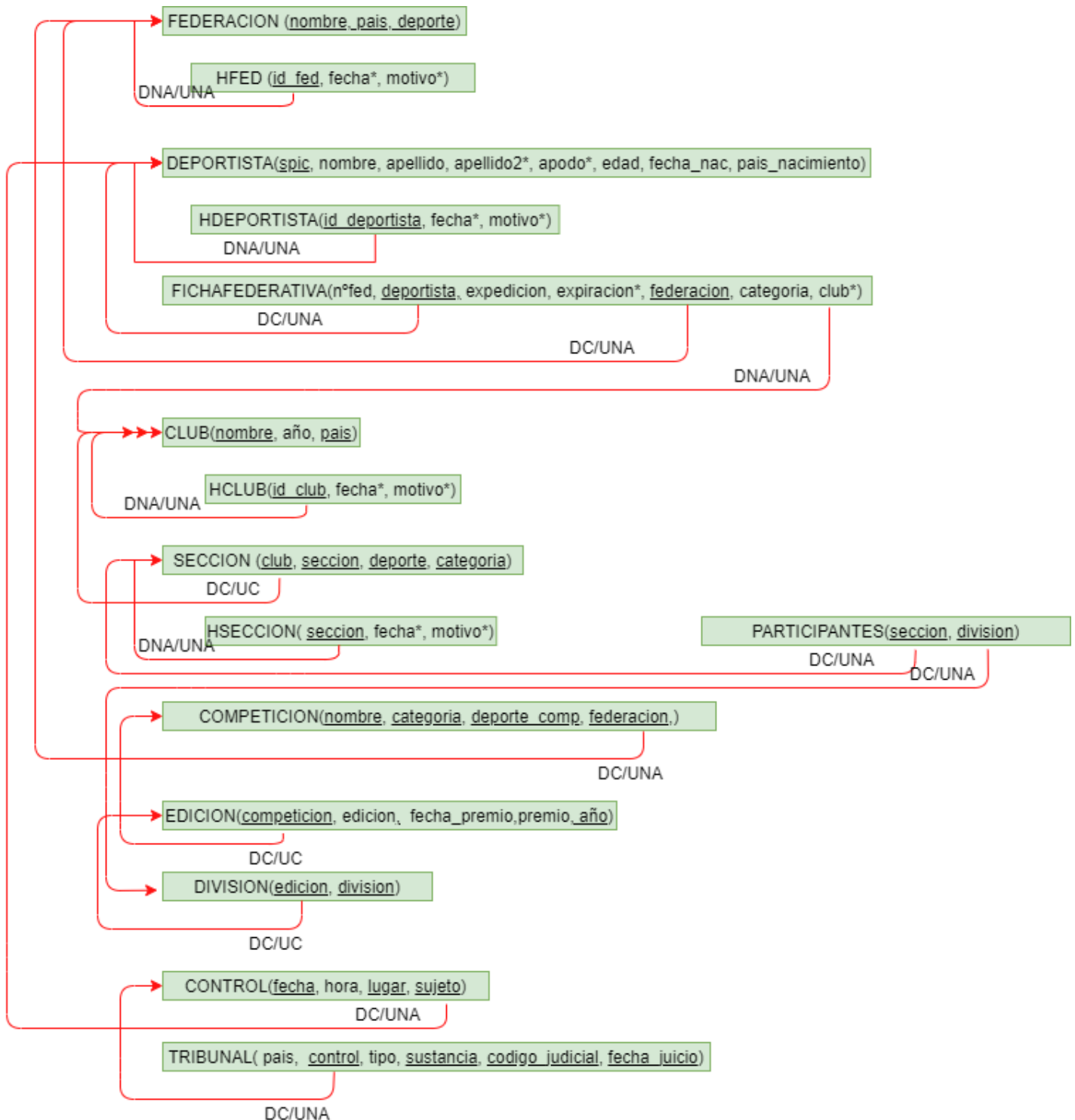
1. Introducción

En esta práctica se nos propone un enunciado referente a un complejo problema sobre el funcionamiento de los eventos deportivos en el mundo. Se nos proporciona una base de datos obsoleta de la que tenemos que realizar un mejor diseño y posteriormente insertar todos los datos de la antigua base de datos en la nueva. Empezaremos explicando el esquema relacional con las semánticas implícitas y explícitas, después indicaremos como hemos implementado las semánticas en sql y para finalizar explicaremos la carga de datos indicando el orden de las tablas y los diferentes problemas encontrados.

2. Diseño Relacional

Esta sección se subdivide en tres apartados:

- Esquema relacional:



- **Semántica implícita:** supuestos semánticos que, por referirse a información ausente en la descripción explícita (es decir, no se encuentran en el enunciado), es necesario añadir para completar el diseño.

Sup_id	Mecanismo	Descripción
I ₁		Asumimos que Federación también se identifica por el país y el deporte
I ₂		Asumimos que todos los clubes están federados ya que solo existe una federación de un deporte por país, por lo tanto a cada club le pertenece esa federación.
I ₃		Las secciones de los clubes participantes de una competición son aquellas que sus categorías y deportes coinciden con los de la edición de la competición.
I ₄		El país de nacimiento de un deportista coincide con el país que está federado.
I ₅		Los controles se identifican también por el lugar.

Tabla 1: Semántica implícita

- **Semántica explícita no contemplada en el diseño:** supuestos semánticos indicados en el enunciado que no han podido representarse en el esquema relacional. Para cada uno de los supuestos, crea una fila en la tabla presentada a continuación.

Sup_id	Descripción
S ₁	El atributo categoría puede ser: Profesional (7), Senior (6), Junior (5), Cadet (4), Old Pioneer (3), Young Pioneer (2) y Benjamin(1) o 'unique'
S ₂	El atributo división puede ser: (1ª, 2ª, 3ª, ...) o 'unique'
S ₃	Cuando hay uno o más años en las competiciones se coge el mayor.
S ₄	Asumimos que solo se pueden meter deportistas en distintas federaciones pero solo si tienen el mismo país.
S ₅	Asumimos que un deportista no puede estar en clubes diferentes que tengan la misma categoría y deporte.
S ₆	Asumimos que no puede haber equipos subsidiarios de un mismo club compitiendo en la misma categoría.
S ₇	Asumimos que solo se puede hacer un solo control en el mismo día y en el mismo lugar a un deportista.

Tabla 2: Semántica explícita no contemplada

3. Implementación de la Estática Relacional en SQL (LDD)

Esta sección complementa al fichero con el script de creación de la base de datos (NEWcreation.sql). Añadiendo los siguientes apartados:

Semántica explícita re-incorporada: Incluir aquellos supuestos de la Tabla 2 que se han podido contemplar con las sentencias de definición de SQL.

Sup id	Descripción de la solución
S ₃	MAX(año)
S ₄	spic varchar2(12)
S ₅	CONSTRAINT CK TRIBUNAL CHECK (código judicial>0)
S ₆	TO_DATE(fechas, 'yyyy-mm-dd')

Tabla 3: Semántica explícita re-incorporada

Semántica implícita: (continúa la numeración donde terminó en la tabla 1)

Sup id	Mecanismo	Descripción
I ₆	Check (rest. sem. simple)	No existe valor de edad menor que 1 y mayor que 99.
I ₇	spic varchar2(12)	El spic es una cadena de 12 caracteres.
I ₈	Check (código > 0)	El código judicial es un número mayor que 0.
I ₉	TO_DATE(fecha)	El formato de todos los atributos de fechas es 'yyyy-mm-dd'
I ₁₀	insert deportista(spic) from fsbd.inscriptions where spic != hdeportista.spic	Para evitar utilizar un SPIC de un deportista borrado se debe introducir el nuevo deportista con su spic en las tablas deportista y comprobar que no esté en hdeportista.
I ₁₁	Realizar dos insert iguales para cada tabla y su histórico	En todas las tablas históricos (hfederación, hdeportista, hclub, hsección) se deben introducir a la vez los mismos datos que en sus tablas normales correspondientes (federación, deportista, club, sección).
I ₁₂	fk_ficha_fed on delete cascade fk_comp_fed on delete cascade	Si se borran las federaciones se borran las fichas y competiciones asociadas a las mismas

Tabla 1(cont.): Semántica implícita

Semántica excluida: Al crear la base de datos en SQL específico del SGBD Oracle puede que no se hayan podido contemplar algunas restricciones semánticas explícitas (tabla 2 – tabla 3), o implícitas que no han podido incorporarse (tabla 1).

Sup_id	Descripción semántica	Motivo	Explícita/ Implícita
E ₁	Se pueden meter varias federaciones con el mismo nombre para diferentes países y deportes	Necesitamos que deporte y país sean claves primarias para poder hacer las relaciones y el resto de la práctica	Implícita
E ₂	Los deportistas pueden estar en federaciones de países diferentes	La tabla fichafederativa tiene una relación con federación que permite que puede haber federaciones de países diferentes en cada ficha	Explícita
E ₃	Cabe la posibilidad de que un deportista pueda estar en diferentes clubs para una misma sección, categoría y deporte .	Al poder estar los clubs vacíos sin ningún integrante y que un deportista puede estar en diferentes federaciones y clubs cabe la posibilidad que para el mismo deporte haya dos club en el que puedas meter al mismo deportista	Explícita
E ₄	Cabe la posibilidad que en una misma competición haya equipos subsidiarias de un mismo club	Al tener la tabla participantes una relación tanto a sección como a división y al hacer que puedan haber equipos subsidiarios se podría meter en participantes dos equipos del mismo club con la misma categoría y deporte.	Explícita
E ₅	Cabe la posibilidad que una persona realice más de un control el mismo día pero en lugares diferentes.	Necesitamos que lugar sea clave primaria para poder hacer las relaciones y el resto de la práctica	Explícita

Tabla 4: Semántica excluida en la creación de tablas

4. Carga de datos (LMD)

Esta sección describirá la carga de datos realizada desde las tablas desnormalizadas entregadas junto con la entrega del fichero de carga (NEWload.sql). A tal efecto, se analizará el problema de la carga y se describirá la solución, haciendo hincapié en:

- El orden de tablas que se adopta para volcar en ellas los datos (justificado).
- Los problemas que surgen (campos obligatorios sin valor, defectos en los datos originales, conversiones de datos, etc) y las soluciones que se adoptan para superarlos

Para cargar los datos en nuestras nuevas tablas hay que seguir un orden ya que no se puede crear una tabla hija que hereda de su padre antes que la tabla padre.

Orden de creación de las tablas:

Primero creamos las tablas FEDERACIÓN y DEPORTISTA, ya que la mayoría del resto de tablas heredan de aquí. Luego podemos seguir con la creación de CLUB y los históricos correspondientes. Las siguientes tablas a crear son FICHA FEDERATIVA, COMPETICIÓN, EDICIÓN, DIVISIÓN, SECCIÓN y PARTICIPANTES (todas creadas en el orden escrito). Por último, se crean las tablas correspondientes para el control de infracciones y los históricos para las federaciones, deportistas, clubes y secciones. El orden de creación a seguir es CONTROL, TRIBUNAL, HFEDERACION, HDEPORTISTA, HCLUB, HSECCION.

Orden de inserción masiva de datos:

La inserción masiva de datos se deben realizar en el mismo orden en el que hemos creado las tablas.

Campos obligatorios sin valor:

En la tabla PARTICIPANTES: los campos tienen que estar obligatoriamente vacíos ya que por la información que nos proporciona la tabla obsoleta no sabemos qué clubes participan en que competiciones.

En las tablas de Histórico: los campos de fecha y motivo inicialmente estarán vacíos ya que las federaciones siguen existiendo, pero al borrar algún dato de las tablas FEDERACIÓN, CLUB, DEPORTISTA Y SECCIÓN, estos campos se deben rellenar con la información correspondiente

En la tabla FICHA FEDERATIVA: el campo club, el cual asocia un club a un deportista, no podemos asignarle ningún valor ya que desconocemos el club al que pertenece cada deportista. Se deja vacío y en un futuro se podrá rellenar.

Defectos en los datos originales:

Fecha 29/2/2017: en la fecha de uno de los controles era 2017-02-29 que al ser un año no bisiesto da error al transformar el varchar2 al formato fecha. Para solucionarlo e introducir los datos hemos puesto un where CTRL_DATE != '2017-02-29'.

Muchas de las infracciones estaban eran nulas por lo que daba error al introducir los datos en TRIBUNAL al tener una relación con la tabla CONTROL. Para solucionarlo metimos un where infracción is not null.

No estaba en la base de datos la edad de los deportistas pero sí que te la pide el enunciado por lo que teniendo la fecha de nacimiento podríamos saber su edad con este comando TRUNC((sysdate-(TO_DATE(BIRTHDATE,'YYYY-MM-DD')))/365.2422,0).

Tuvimos un problema con un error de clave primaria violada porque hay competiciones que tienen varios años y varios premios y algunos de ellos nulos por lo que como indica el enunciado cogemos el que tenga mayor año y un solo premio con MAX(año) y MAX(premio) al insertar los datos.

En la base de datos fsdb.controls venían spics de deportistas que no estaban en la base de datos fsdb.inscriptions. Para solucionar esto y poder realizar los controles a los deportistas pusimos en el select que cogiera las dos tablas y que seleccionara los spics que coincidieran en las dos tablas. Esto último lo realizamos con un WHERE fsdb.controls.SPIC = fsdb.inscriptions.SPIC dentro del select que utilizamos para insertar los datos en control.

Todas las fechas las hemos transformado con TO_DATE de varchar2 a formato fecha.