Introduction to Database
(“Bases de Datos”)

Escuela Universitaria de Informática
Facultad de Informática

Lab exercise nº 2: Representation of reality with the relational data model.

1. Introduction 2
2.General overview of the schemas we will work with 2
3. Questions about the proposed schemas 13

Departamento de Sistemas Informáticos y Computación
year 2007/2008

1. Introduction

The goal of this session is to learn some guidelines for the representation of information systems with relational schemas, as well as to get some practice in the interpretation of simple relational schemas.

An information system based on the real world includes several objects with their attributes and inter-relationships. In order to obtain a relational schema which represents an information system, we must take into account that:

- For each object in the real world about which we want to express some kind of information, we must define a relation whose attributes denote the most significant properties of the object (code, name, ...) in such a way that each tuple in that relation has to be interpreted as a particular instance of that object;
- To represent the associations between objects, we use explicit references through attributes which identify each object. In some cases, new relations representing facts (predicates) in the real world have to be added in order to express these associations.

The relational schema obtained in this way must comply with the following modelling guidelines:

- To satisfy the information system requirements.
- To avoid redundancies
- To adapt to data structures of the data model (in our case the relational model)

2. General Overview of the schemas we will work with

Next we will present some information systems and the relational schemas that represent them:

For the several schemas, we use the following notation:

PK: Primary Key */ CP: Clave Primaria*: the set of attributes with this constraint form the primary key.

UNI: Uniqueness constraint / *UNI: Restricción de unicidad*: the set of attributes with this constraint cannot be repeated.

FK: Foreign Key / *CAj: Clave Ajena*: the set of attributes with this constraint refer to corresponding attributes of the referred relation.

NNV: Not Null Value / *NNV: Valor no nulo*: the set of attributes with this constraint cannot be null.

Information system: ***GEOGRAPHIC INFORMATION***

Information Requirements

Objects: SEA, RIVER, PROVINCE

For each SEA: code, name, details and rivers which flow into it.

For each RIVER: code, name, length, sea it flows into and provinces it crosses, also indicating the kilometres through the province.

For each PROVINCE: code, name, extension, boundaries with other provinces and rivers that cross the province.

Relational Schema

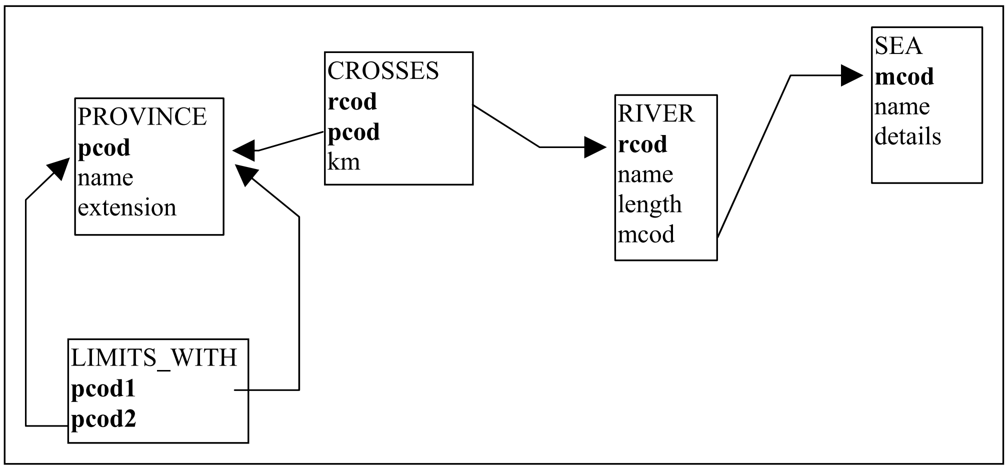
• **RIVER** (rcod: d_rcod, name: d_nom, length: d_long, mcod: d_mcod)
  PK: {rcod}
  FK: {mcod} → SEA

• **SEA** (mcod: d_mcod, name: d_nom, details: d_det)
  PK: {mcod}

• **PROVINCE** (pcod: d_pcod, name: d_nom, extension: d_ext)
  PK: {pcod}

• **CROSSES** (rcod: d_rcod, pcod: d_pcod, km: d_km)
  PK: {pcod,rcod}
  FK: {pcod} → PROVINCE
  FK: {rcod} → RIVER

• **LIMITS_WIH** (pcod1: d_pcod, pcod2: d_pcod)
  PK: {pcod1,pcod2}
  FK: {pcod1} → PROVINCE
  FK: {pcod2} → PROVINCE



Information system: *COMPANY*

Information requirements

Objects: SUPPLIER, PIECE, PROJECT, ORDER

For each SUPPLIER: code, name, city and order for each piece (indicating the project in which it is used).

For each PIECE: code, name, colour, weight, city where it is manufactured and the suppliers who supply some projects.

For each PROJECT: code, name, city and pieces bought to the suppliers of that project.

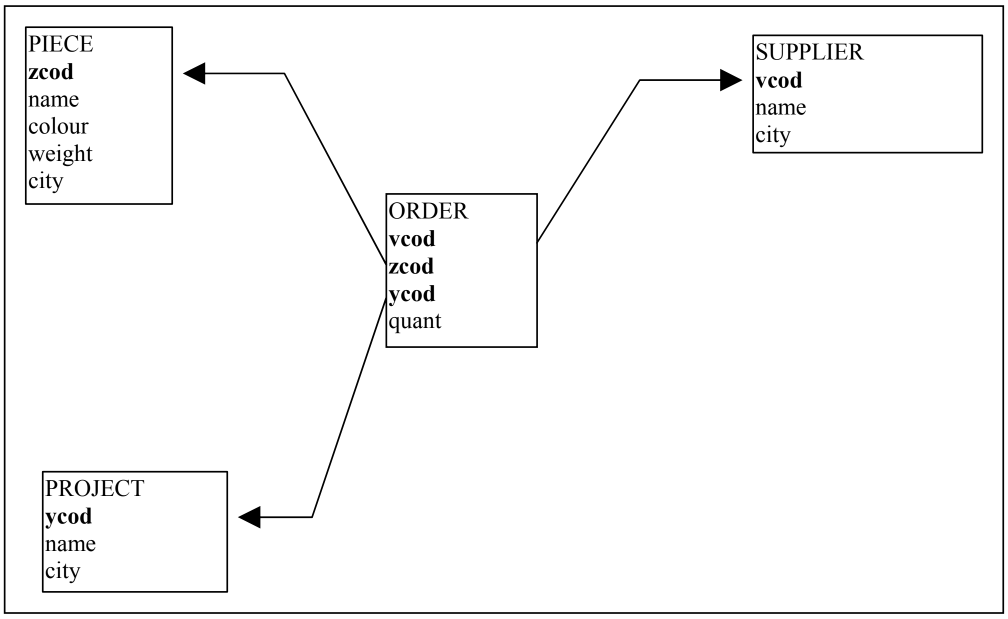
Relational Schema

• **SUPPLIER** (vcod: d_vcod, name: d_nom1, city: d_ciu)
PK:{vcod}

• **PIECE**(zcod: d_zcod, name: d_nom2, colour: d_colour, weight: d_weight, city: d_ciu)
PK:{zcod}

• **PROJECT**(ycod: d_ycod, name: d_nom3, city: d_ciu)
PK: {ycod}

• **ORDER** (vcod: d_vcod, zcod: d_zcod, ycod: d_ycod, quant: d_quant)
PK:{vcod, zcod, ycod}
FK:{vcod} → SUPPLIER
FK:{zcod} → PIECE
FK:{ycod} → PROJECT



Information system: *LIBRARY 1*

Information requirements

Objects: USER, BOOK

For each USER: code, name, address and the books s/he has read.

For each BOOK: code, title, author, topic and the users who have read it.

We also want to establish a hierarchy between different topics.

Relational Schema

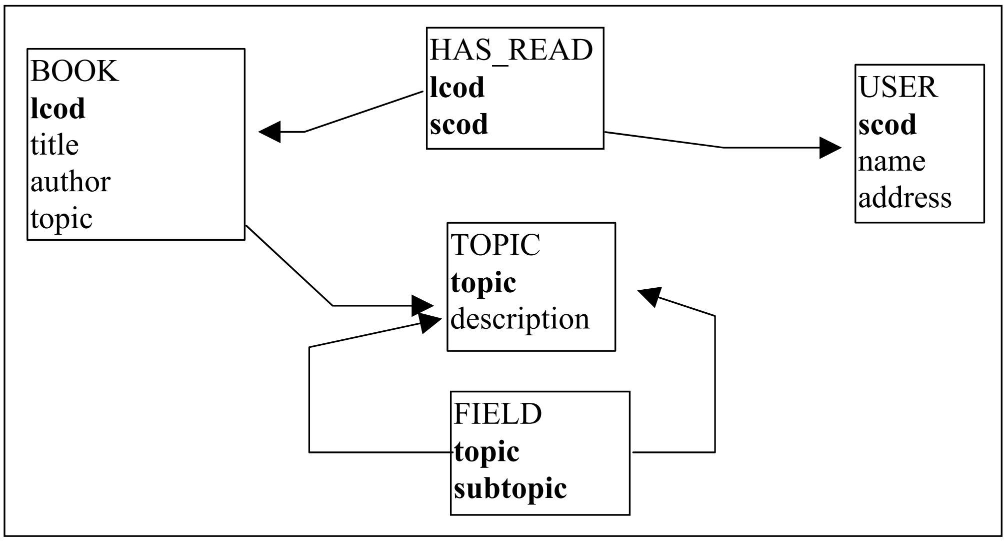
- **USER** (scod: d_scod, name: d_nom, address: d_dir)
  PK: {scod}

- **BOOK** (lcod: d_lcod, title: d_tit, author: d_author, topic: d_topic)
  PK: {lcod}
  FK: {topic} → topic

- **HAS_READ** (scod: d_scod, lcod: d_lcod)
  PK: {scod, lcod}
  FK: {scod} → user
  FK: {lcod} → book

- **TOPIC** (topic: d_topic, description: d_desc)
  PK: {topic}

- **FIELD** (topic: d_topic, subtopic: d_topic)
  PK: {topic, subtopic}
  FK: {topic} → topic
  FK: {subtopic} → topic



Information system: ***RECORD COLLECTION***

Information requirements

Objects: COMPOSER, CONDUCTOR, WORK, DISC

For each COMPOSER: name, birth year, nationality and works s/he has composed.

For each CONDUCTOR: name, birth year, short biography and which works are conduced by him or her in the discs.

For each WORK: code of the work, title, year of its composition and its author, as well as the discs which include it.

For each DISC: reference number, name, publishing year, publishing company and works which contain (includes information about the conductor of the particular interpretation).

Relational Schema

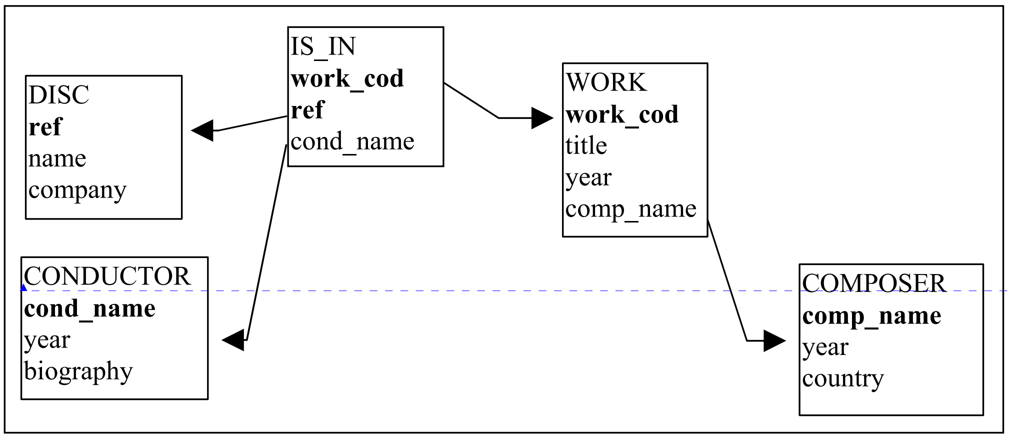
- **COMPOSER** (comp_name: d_name, year: d_year, country: d_country)
  PK: {comp_name}

- **CONDUCTOR** (cond_name: d_name, year: d_year, biography: d_bio)
  PK: {cond_name}

- **WORK** (word_cod: d_work_cod, title: d_title, year: d_year, comp_name: d_name)
  PK: {work_cod}
  FK: {comp_name } → COMPOSER

- **DISC** (ref: d_ref, name: d_name, year: d_year, company: d_comp)
  PK: {ref}

- **IS_IN** (workd_cod: d_work_cod, ref: d_ref, cond_name: d_name)
  PK: {work_cod, ref}
  FK: {work_cod} → WORK
  FK: {ref} → DISC
  FK: {cond_name} → CONDUCTOR



Con formato: Inglés (Reino Unido)

Information system: *TRAVEL AGENCY*

Information requirements

Objects: TRIP, GUIDE, DRIVER, CITY

For each TRIP: code, date, price, assigned guide, drivers and cities which are visited.

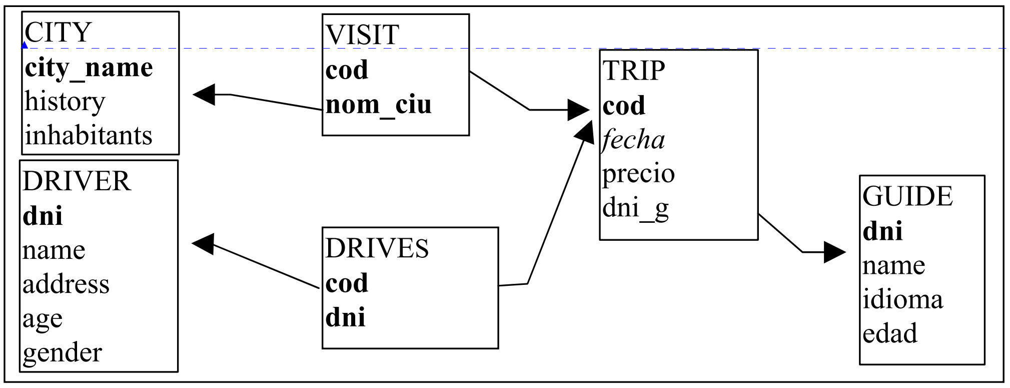
For each GUIDE: dni[1], name, language spoken, trips in which she or he participates.

For each DRIVER: dni, name, age, gender, trips in which s/he participates.

For each CITY: name city, number of inhabitants, brief historical description.

Relational Schema

- **TRIP**(code: d_code, date: d_date, price: d_pri, dni_g: d_dni)
  PK: {code}
  FK: {dni_g} → GUIDE
  NNV: {date}
- **GUIDE**(dni_g: d_dni, name: d_name, language: d_lan, age: d_age)
  PK: {dni_g}
- **DRIVER**(dni: d_dni, name: d_name, address: d_ad, age: d_age, gender: d_gen)
  PK:{dni}
- **CITY**(city_name: d_name1, history: d_his, inhabitants: d_inh)
  PK:{city_name}
- **VISIT**(code: d_code, city_name: d_nom1)
  PK:{code, city_name}
  FK:{code} → TRAVEL
  FK: {city_name} → CITY
- **DRIVES**(dni: d_dni, code: d_code)
  PK: {dni, code}
  FK: {code} → TRAVEL
  FK: {dni} → DRIVER



Con formato: Inglés (Reino Unido)

[1] DNI stands for the number of the "Documento Nacional de Identidad" (*Spanish National Identity Document*).

Information system: *CYCLISM*

Information requirements

Objects: EQUIPO, CICLISTA, PUERTO, MAILLOT, ETAPA

- For each EQUIPO (TEAM): name ("nombre"), the coach ("director") and the racers ("ciclistas") who compose the team.
- For each CICLISTA (RACING CYCLIST): "dorsal" number (cyclist number assigned to the cyclist during the race), name, age, name of the team it belongs to, stages ("etapas") he has won, mountain passes ("puertos") he has gone through on the first position and the maillots ("maillots") he has worn in each stage.
- For each PUERTO (MOUNTAIN PASS): name, maximum height, category ("1ª", "especial",…), slope ("pendiente"), stage where it is located and the cyclist who has passed it on first position.
- For each MAILLOT (MAILLOT): maillot code, prize level of the maillot ("tipo"), colour, prize for wearing it ("premio") and cyclists who have worn it.
- For each ETAPA (STAGE): stage number, length of the stage in kms, departure city ("salida"), arrival city ("llegada") and the cyclist who has won the stage.

Relational Schema

EQUIPO(nomeq: d_eq, CONDUCTOR: d_nom)

PK: {nomeq}

CICLISTA(dorsal: d_dor, nombre: d_nom, edad: d_edad, nomeq: d_eq)

PK: {dorsal}

FK: {nomeq}→ EQUIPO

NNV: {nomeq}

NNV: {nombre}

ETAPA(netapa: d_nº, km: d_km, salida: d_ciu, llegada: d_ciu, dorsal: d_dor)

PK: {netapa}

FK: {dorsal}→ CICLISTA

PUERTO(nompuerto: d_nom, altura: d_alt, categoria: d_cat, pendiente: d_pen, netapa: d_nº, dorsal: d_dor)

PK: {nompuerto}

FK: {netapa}→ ETAPA

FK: {dorsal}→ CICLISTA

NNV: {netapa}

MAILLOT(codigo: d_cod, tipo: d_tipo, premio: d_pre, color: d_col)

PK: {codigo}

LLEVAR(dorsal: integer, netapa: d_nº, codigo: d_cod)

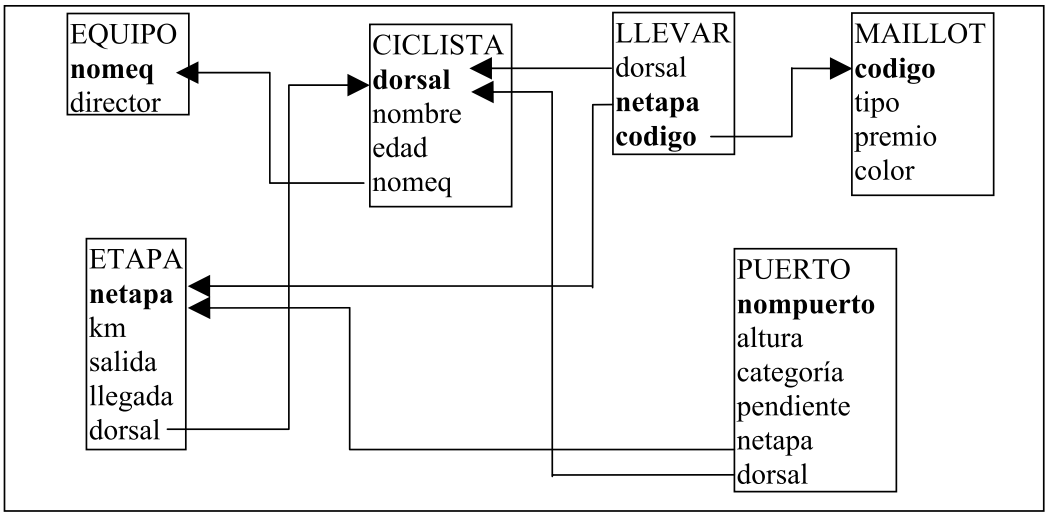
PK: {netapa,codigo}

FK: {netapa}→ ETAPA

FK: {dorsal}→ CICLISTA

FK: {codigo}→ MAILLOT

NNV: {dorsal}



Information system: ***MUSIC***

Information requirements

Objects: COMPAÑÍA, DISCO, GRUPO, ARTISTA, CANCIÓN, CLUB

For each COMPAÑÍA (COMPANY): code, name, address ("dirección"), fax, telephone and discs which have been published by the company.

For each DISCO (DISC): code, name, date, company which publishes the disc, group which has recorded the disc and the songs that it contains.

For each GRUPO (GROUP): code, name, date in which the group was created, country ("país"), artists who compose the group, company that has published it and fans club.

For each ARTISTA (ARTIST): dni, name and group s/he belongs to.

For each CANCIÓN (SONG): code, title, duration and disc where it is found.

For each CLUB (FAN CLUB): code, name, main office ("sede"), number of fans and group they admire.

Relational Schema

CANCION(cod: d_can, título: d_tit, duración: d_dur)

PK: {cod}

NNV: {título}

COMPAÑIA(cod: d_comp, nombre: d_nom, dir: d_dir, fax: d_tel, tfno: d_tel)

PK: {cod}

NNV: {nombre}

DISCO(cod: d_dis, nombre: d_nom, fecha: d_fecha, cod_comp: d_comp, cod_gru: d_gru)

PK: {cod}

FK: {cod_comp}→ COMPAÑIA

NNV: {cod_comp}

FK: {cod_gru}→ GRUPO

NNV: {cod_gru}

ESTA(can: d_can, cod: d_dis)

PK: {can, cod}

FK: {can}→ CANCIÓN

FK: {cod}→ DISC

GRUPO(cod: d_gru, nombre: d_nom, fecha: d_fecha, pais: d_pais)

PK: {cod}

NNV: {nombre}

ARTISTA(dni: d_dni, nombre: d_nom)

PK: {dni}

NNV: {nombre}

CLUB(cod: d_club, nombre: d_nom, sede: d_dir, num: d_num, cod_gru: d_gru)

PK: {cod}

FK: {cod_gru}→ GRUPO

NNV: {cod_gru}

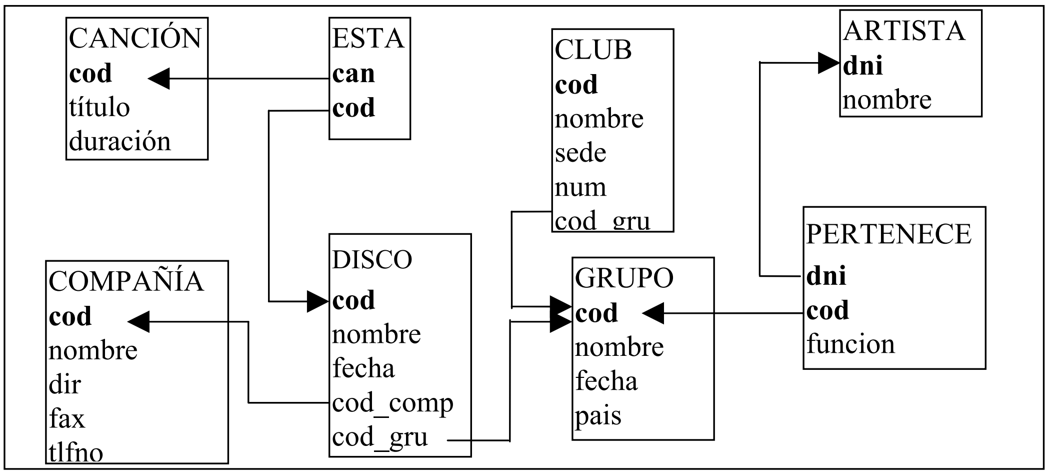
NNV: {nombre}

PERTENECE(dni: d_dni, cod: d_gru, funcion: f_fun)

PK: {dni, cod}

FK: {dni}→ ARTISTA

FK: {cod}→ GRUPO



Information system: *LIBRARY 2*

Information requirements

Objects: AUTOR, LIBRO, TOPIC, WORK, AMIGO

For each AUTOR ("author"): author's identifier, name, nationality and works ("obras") s/he has written.

For each OBRA ("work"): code of the work, title, year, field, author and works in which it is included.

For each TEMA ("topic"): identifier of the topic and a brief description.

For each LIBRO ("book"): identifier of the book, title, year, works it contains and its number, and the friends who have borrowed it.

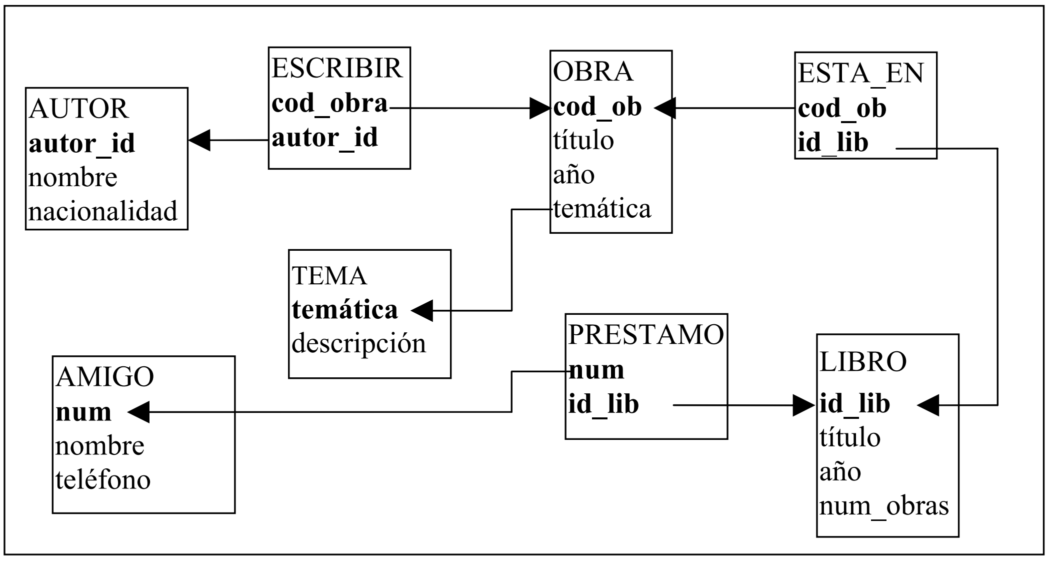
For each AMIGO ("friend"): identifier number, name, telephone and books which s/he has borrowed.

Relational Schema

AUTOR(autor_id: string(4), name: string(35), nacionalidad: string(20))
PK: {autor_id}
NNV: {name}

LIBRO(id_lib: string(10), titulo: string(80), año: integer, num_obras: integer)
PK: {id_lib}

TEMA(tematica: string(20), descripcion: string(50))
PK: {tematica}
OBRA(cod_ob: integer, titulo: string(80), año: d_cat, tematica: string(20))
PK: {cod_ob}
FK: {tematica}→ TEMA
NNV: {titulo}
AMIGO(num: integer, name: string(60), telefono: string(10))
PK: {num}
NNV: {name}
PRESTAMO(num: integer, id_lib:string(10))
PK: {num,id_lib}
FK: {num} → AMIGO
FK: {id_lib} → LIBRO
ESTA_EN(cod_ob: integer, id_lib:string(10))
PK: {cod_ob,id_lib}
FK: {cod_ob} → OBRA
FK: {id_lib} → LIBRO
ESCRIBIR(cod_ob: integer, autor_id:string(4))
PK: {cod_ob,autor_id}
FK: {cod_ob}→ OBRA
FK: {autor_id}→ AUTOR


AUTOR
autor_id
nombre
nacionalidad
ESCRIBIR
cod_obra
autor_id
OBRA
cod_ob
título
año
temática
ESTA_EN
cod_ob
id_lib
TEMA
temática
descripción
PRESTAMO
num
id_lib
AMIGO
num
nombre
teléfono
LIBRO
id_lib
título
año
num_obras

3. Questions about the proposed schemas

Once you have studied each of the proposed schemas, answer the following questions which are presented next. Every answer must be reasoned in terms of the Relational Schema described above (in no way it is acceptable to use a rationale based on our background knowledge of the topic). After that, you can then compare this interpretation with your background knowledge and analyse whether it is an adequate model of reality. In case of detecting some limitation or malfunctioning of the schema for the part of reality it wants to model, propose the modifications of the schema you think can solve the limitations.

Schema: GEOGRAPHIC INFORMATION

1. Can a river flow into two seas?
2. Can a river cross two provinces?
3. Can a river cross the same province twice?
4. Can a province limit with itself?
5. How many seas, as a maximum, can a river flow into? And the minimum?

Schema: COMPANY

6. Can two pieces have the same "zcod" code?
7. Can a piece have two colours?
8. Can a supplier provide two orders with the same piece and project?
9. We could add the attribute "date" to the relation "order", and it can also be included in the primary key. What would this modification allow?

Schema: LIBRARY 1

10. Can a user read more than one book?
11. Can a user read the same book more than once?
12. Can a book have more than one author?
13. Can a book have more than one topic?
14. Can a topic be a subtopic of itself?
15. Can a book be read by two different users?

Schema: RECORD COLLECTION

16. Can a disc contain more than one work?
17. Can a work have more than one composer?
18. Can the same work appear in the same disk twice, conducted by different conductors?, and by the same conductor?
19. Can works with the same composer appear in different discs?

Schema: TRAVEL AGENCY

20. Can the same trip visit the same city twice?
21. Can a guide speak two languages?
22. Can a driver be in two trips at the same time (date)?
23. Can a driver be also a guide?
24. How many drivers are there, as a minimum, in each trip?
25. Can a guide participate in more than one trip?

Schema: CYCLISM

26. Can a cyclist belong to more than one team?
27. Can a cyclist wear more than one maillot in the same stage?
28. And more than one maillot during the whole race ("tour")?
29. Can a mountain pass appear in more than one stage?
30. Can a cyclist win more than one stage?

Schema: MUSIC

31. Can a disc be published by more than one company? And by none?
32. Can a club admire more than one group? And can a group have more than a fans club?
33. Can a song appear in more than one disc?
34. How many groups can an artist belong to as a maximum? And as a minimum?
35. Can a disc have no songs?

Schema: LIBRARY 2

36. Can a friend have more than one borrowed book at the same time? And borrow the same book twice?
37. Can a work have more than one topic?
38. Can a book contain a work more than once? And can a work be included in more than one book?
39. How many works can an author write, as a minimum? And as a maximum?
40. Can a book be borrowed by different friends?