

Unit 1: Relational Databases

1.1. Fundamentals

1.2. The Relational Data Model

1.3. Interpretation of a Relational Database

UD 1.3 Uso de Bases de Datos Relacionales

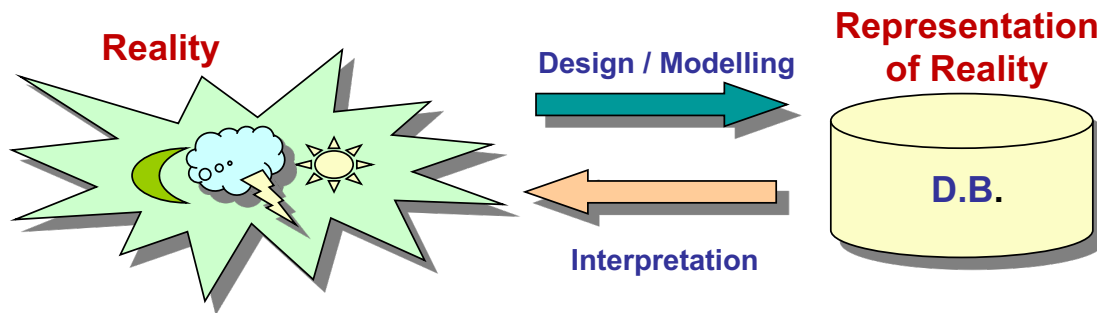
1 Introduction: Representation of reality

2 The “Music Library” Database

3 Interpreting database schemas

4 Examples

1 Introduction



- For each object in reality about which we want to have information we define a **relation** whose **attributes** denote the most significant properties of these objects (code, name, ...) in such a way that each **tuple** which is present in this relation must be interpreted as a particular **instance** of an object.
- In order to represent **associations** between objects we use explicit **references** through **attributes** which identify each object.
- Associations between objects where the cardinality is **many-to-many** require an extra **relation**

3

UD1.3 Uso de Bases de Datos Relacionales

1 Introduction: Representation of reality

2 The “Music Library” Database

3 Interpreting database schemas

4 Examples

4

Music Library

Canción (song)

cod: song code (id).

título: Song title.

duración: Length of the song.

Está (is_in)

It stores what songs are included in each record, where “*can*” is the code of a song appearing in the record “*cod*”.

Compañía (company)

cod: record company (record label) code.

nombre: company name.

dir: Address of the company.

fax: Fax number of the company.

tfno.: Phone number of the company.

Disco (record)

cod: record code (id).

nombre: record name.

fecha: Publishing date.

cod_comp: Code of the record company which has published this record.

cod_gru: Code of the music group (band) which has recorded this record.

5

Music Library

Grupo (group/band)

cod: Group (band) code.

nombre: Name of the group.

fecha: Date of the group foundation.

país: Country where the group was created.

Artista (artist)

dni: artist id.

nombre: name of the artist.

Club (fan club)

cod: fan club code (id).

nombre: name of the club.

sede: Address of the main office.

num: number of members of the club.

cod_gru: code of the group which the club is fan of.

Pertenece (belongs_to)

It contains the group members information: The artist “*dni*” is member of the group “*cod*” performing the function “*funcion*” (e.g. plays the guitar, sings,...).

6

Música

Canción (*cod*: entero, *título*: char(30), *duración*: real)

CP:{cod}

VNN:{título}

Compañía (*cod*: char(3), *nombre*: char(30), *dir*: char(30), *fax*: char(15), *tfno*: char(15))

CP:{cod}

VNN:{nombre}

Disco (*cod*: char(3), *nombre*: char(30), *fecha*: date, *cod_comp*: char(3), *cod_gru*: char(3))

CP:{cod}

CAj:{cod_comp} → Compañía

VNN:{cod_comp}

CAj:{cod_gru} → Grupo

VNN:{cod_gru}

Esta (*can*: entero, *cod*: char(3))

CP:{can,cod}

CAj:{can} → Cancion

CAj:{cod} → Disco

7

Música

Grupo (*cod*: char(3), *nombre*: char(30), *fecha*: date, *país*: char(10))

CP:{cod}

VNN:{nombre}

Único:{nombre}

Artista (*dni*: char(10), *nombre*: char(30))

CP:{dni}

VNN:{nombre}

Club (*cod*: char(3), *nombre*: char(30), *sede*: char(30), *num*: entero, *cod_gru*: char(3))

CP:{cod}

CAj:{cod_gru} → Grupo

VNN:{cod_gru}

Único:{cod_gru}

VNN:{nombre}

Pertenece (*dni*: char(10), *cod*: char(3), *función*: char(15))

CP:{dni,cod}

CAj:{dni} → Artista

CAj:{cod} → Grupo

8

UD1.3 Uso de Bases de Datos Relacionales

1 Introduction: Representation of reality

2 The “Music Library” Database

3 Interpreting database schemas

4 Examples

9

3 Interpreting database schemas

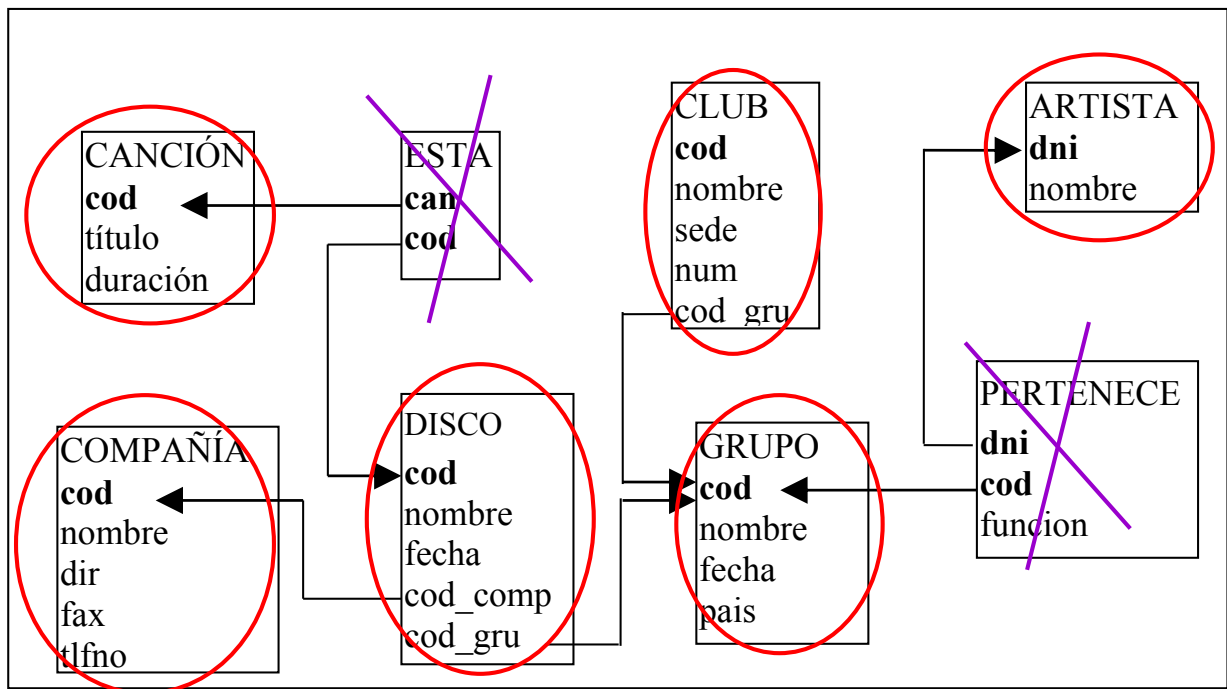
Tables:

Objects are represented by tables for which none of the components of their PK refer to other tables (they have existence on their own).

This is not true for hierarchical relationships (specialization).

The rest of tables represent **relationships** (non-entity tables).

10

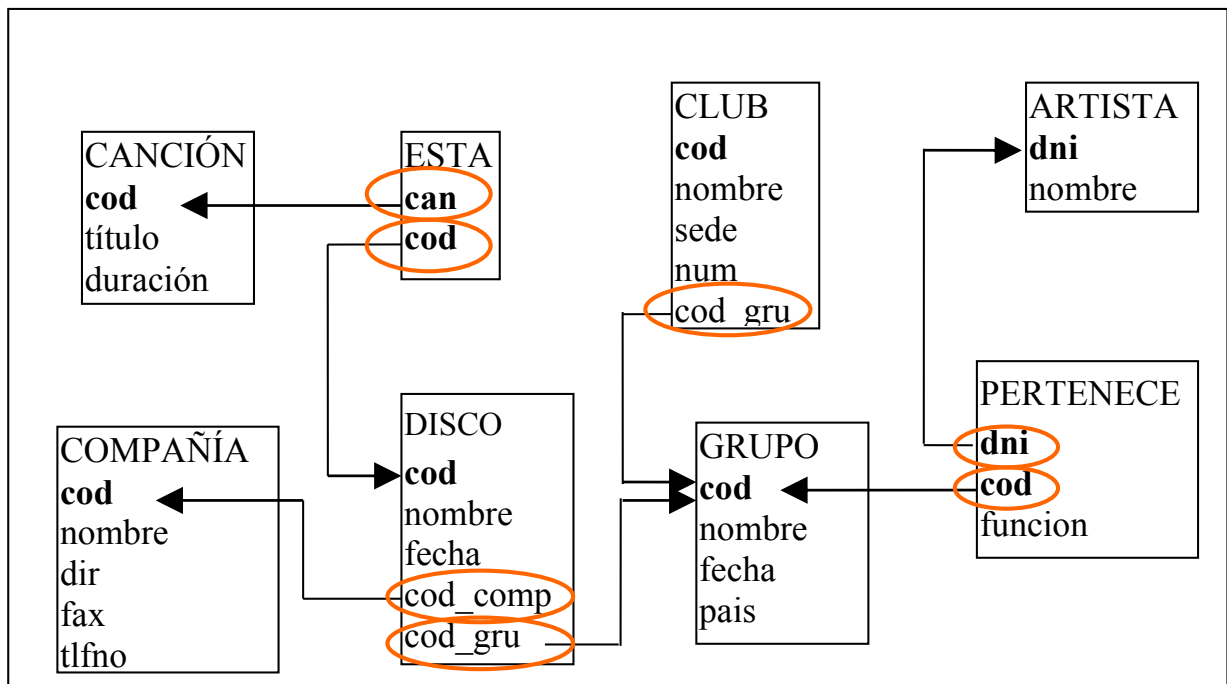


11

Attributes

- Represent **properties** of objects (if they are not FK)
- If they are FK, they represent **relationships** between objects

12



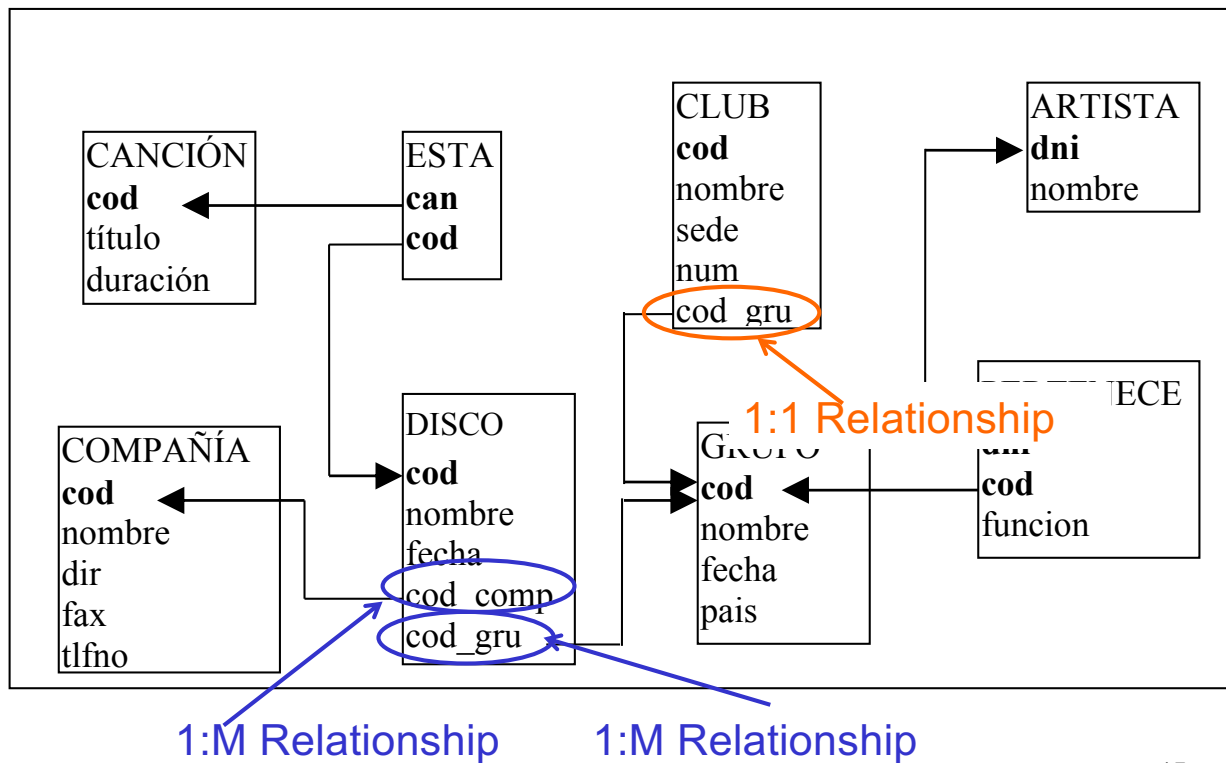
13

Relationship types

1. The FK is in a table representing an object
 - FK has UNI constraint
1:1 relationship (one-to-one), or 0:1
 - FK doesn't have UNI constraint:
1:M relationship (one-to-many)
2. The FK is in a non-entity (object) table. The PK is composed of two FK
M:M relationship (many-to-many)

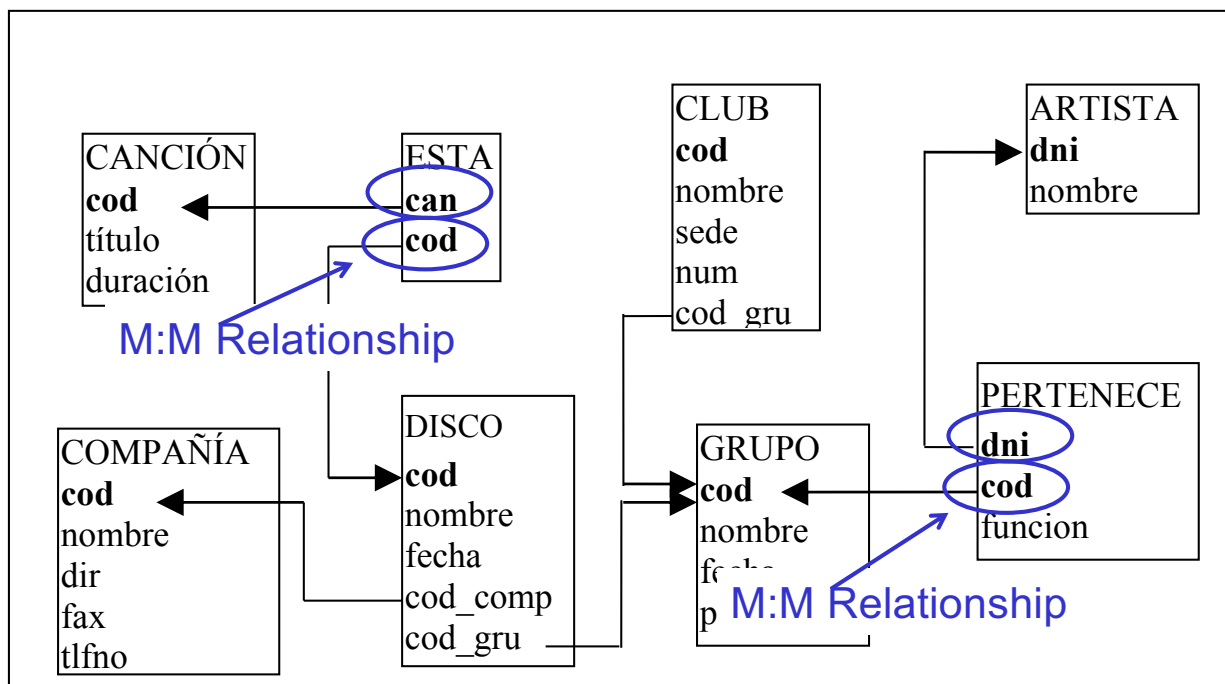
14

Relationship types



15

Relationship types



16

Constraints

Non null value

- If a FK from R to S has a NNV constraint, then **every** object in R is associated with **one** object in S.
- If a FK from R to S does not have a NNV constraint, then every object in R is **not necessarily** associated with any object in S

Uniqueness

- If a FK from R to S has a uniqueness constraint, then **every** object in S can **at most** be associated with **one** object in R.

17

UD1.3 Uso de Bases de Datos Relacionales

- 1 Introduction: Representation of reality
- 2 The “Music Library” Database
- 3 Interpreting database schemas
- 4 Examples

18

Geographical Information

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

BORDERS_ON (pcod1: d_pcod, pcod2: d_pcod)

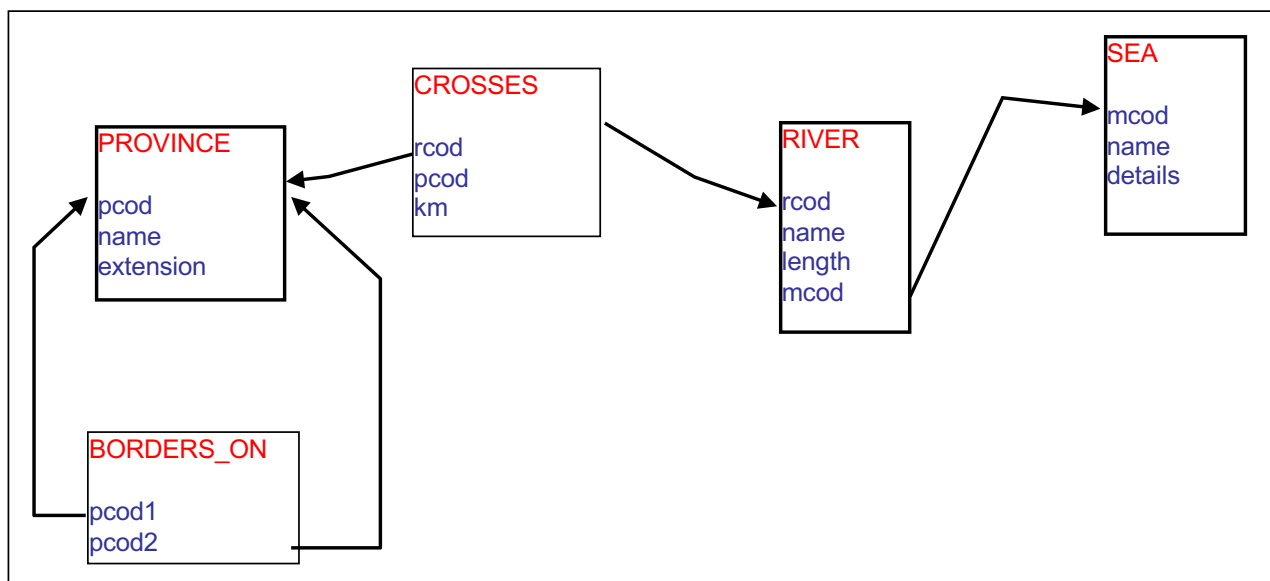
PK: {pcod1,pcod2}

FK: {pcod1} -> PROVINCE

FK: {pcod2} -> PROVINCE

19

Geographical Information



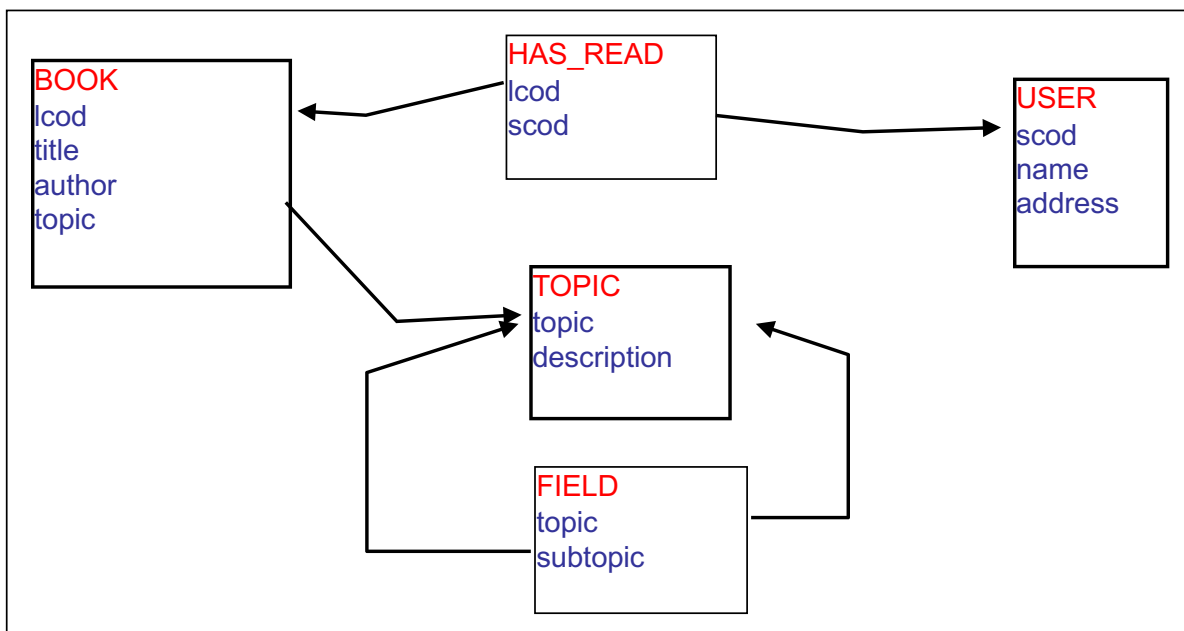
20

Schema: Geographical 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 border on itself?
5. How many seas, as a maximum, can a river flow into? And the minimum?

21

Library



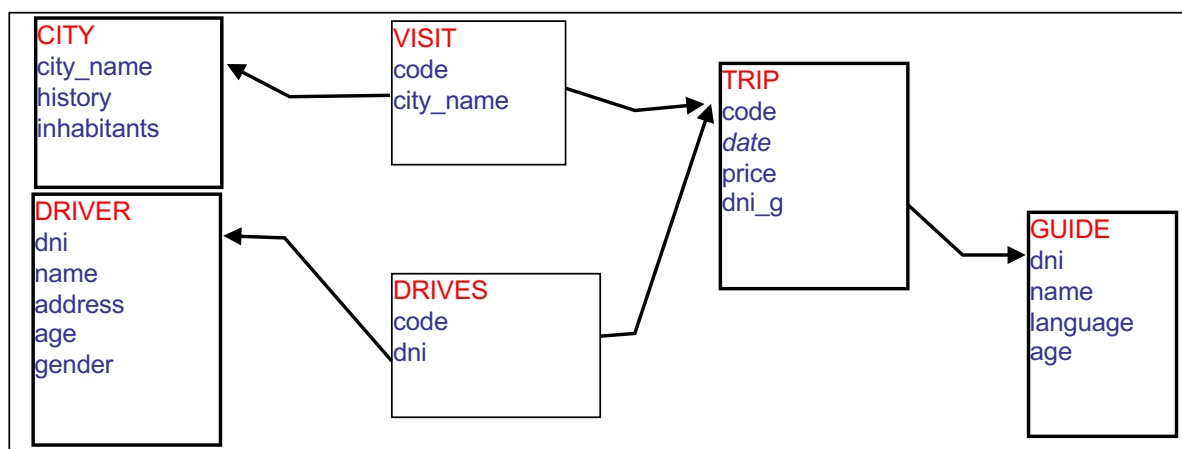
22

Library

1. Can a user read more than one book?
2. Can a user read the same book more than once?
3. Can a book have more than one author?
4. Can a book have more than one topic?
5. Can a topic be a subtopic of itself?
6. Can a book be read by two different users?

23

Travel Agency



24

Travel Agency

1. Can the same trip visit the same city twice?
2. Can a guide speak two languages?
3. Can a driver be in two trips at the same time (date)?
4. Can a driver be also a guide?
5. How many drivers are there, as a minimum, in each trip?
6. Can a guide participate in more than one trip?

25

Cycle racing

Equipo (team)

nomeq: name of the team.

director: name of the team coach.

Ciclista (racing cyclist)

dorsal: cyclist number assigned to the cyclist during the race.

nombre: cyclist name.

edad: age of the cyclist.

nomeq: name of the cyclist team.

Etapas (stage)

netapa: stage number (in the race).

km: How many kilometers the stage has.

salida: name of the city where the stage starts (departure).

llegada: name of the city where the stage finish (arrival).

dorsal: number of the cyclist who has won the stage.

26

Cycle racing

Puerto (mountain pass)

nompuerto: name of the mountain pass.

altura: maximum height in the pass.

categoria: category of the pass: 1ª/primera (first), especial (special),

pendiente: mean slope of the pass (in %).

netapa: stage number where the mountain pass is climbed.

dorsal: number of the cyclist who has won the mountain pass ("puerto")

Maillot (Jersey)

codigo: code of the maillot.

tipo: indicates the prize level of the maillot.

color: color of the prize.

premio: how much money the cyclist wins if he finishes wearing this jersey.

Llevar (Wear)

The cyclist with number '*dorsal*' has worn the maillot identified by '*codigo*' at the stage with number '*netapa*'.

27

Ciclismo

EQUIPO (*nomeq*: d_eq, *director*: 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*: d_dor, *netapa*: d_nº, *codigo*: d_cod)

PK: {netapa,codigo}

FK: {netapa}→ ETAPA FK: {dorsal}→ CICLISTA

FK: {codigo}→ MAILLOT NNV: {dorsal}

28

Schema: Cycle racing

1. Can a cyclist belong to more than one team?
2. Can a cyclist wear more than one maillot in the same stage?
3. And more than one maillot during the whole race ("tour")?
4. Can a mountain pass appear in more than one stage?
5. Can a cyclist win more than one stage?