

Databases / Bases de Datos

EXERCISES

Each exercise on a separate sheet. Exercises and subsections in order. Fold all the sheets and deliver together.
30% minimum

FIRST (40%, exercises part)

Develop an Entity-Relationship diagram (capturing the most semantics), using extended features of the E-R model if needed, and then **reduce it to a set of relations (tables) in the relational model**. The universe of discourse is a web site for an online database about videos (films, tv episodes...).

- The database stores videos. Each video has a unique identifier, a title, a release date, and the production company.
- There are 3 types of videos: films, TV episodes, and collections.
- Each episode is a different video, and belongs to one collection (or more), and has an episode number.
- Each collection has a descriptive name, and groups the episodes. A collection contains one or more seasons of episodes that should be recorded. For example, the collection “CSI: the Grissom seasons” contains seasons 1,2,3,4,5,6,7,8, and 9.
- A collection can include up to 50 episodes.
- Films have no numbering, and are always independent from each other (“Karate Kid”, “Karate Kid II”, and “Karate Kid III” are different videos). The box office (revenue) of the films has to be known.
- Each video belongs to one or more genres (“Comedy”, “Drama”, “Action”, etc.). Each of these genres has a specific “percentage of importance” for the video. For example “Karate Kid” is 70% action, 20% drama, and 10% comedy.
- Each genre has a relative popularity within the database.
- Users rate videos with a 0-5 scale. More precisely, each of the genres for a video is rated. For example, a user can rate the action part of “Karate Kid” with 5, the drama part with 2, and then the comedy part with 3. All the genres for a video must be rated. Each user can rate the same film several times (in different dates).
- The information about the users is the user id, the user name, and the country.
- Users can be “friends” of another users. That conforms the beginnings of a social network for the site. The date when a user accepted the friend request from another user should be recorded.
- Each video has an average score, which is computed using the individual ratings of the users.

List the CONSTRAINTS NOT CAPTURED IN THE DIAGRAM

Note: Only the fundamental attributes needed to understand the schema should be drawn on the diagram. Represent in the broadest way (the most reasonable way) the parts that are not completely defined by the previous wording. Explain the representation chosen in each case.

SECOND (40%, exercises part)

Write SQL queries (reduce to ANSI SQL86 (SQL1) if a later version is used), using the database in the annex...

- a) SQL – Delete all the information stored about the “Gortel” investigation (*investigación*).
- b) SQL *and Relational Algebra* – Couples of DNI and name of judge and prosecutor (*fiscal*) that live in the same city, and have participated together in at least two corruption trials (*juicios de corrupción*). (Fi: 1 Paco and 2 Carmen as they have participated together as judge and prosecutor in two trials, and both trials are for corruption).
- c) SQL – DNI and name of the citizen (or citizens) that live in Oviedo and have the most number of complaints (*imputaciones*)
- d) SQL – DNI and name of the citizens that have got “not guilty” sentences in all the trials held by judges that live in their same city. (Fi: 7 Luis, as Luis lives in Oviedo and was accused in 5 trials held by judges that live in Oviedo, and he got “not guilty” sentences in all 5 trials.

THIRD (20%, exercises part)

Find a lossless join (LJ), dependency preserving (DP) decomposition, normalized as best as possible in 3NF or BCNF (minimal number of schemas and redundancy), using the schema $R = (A, B, C, D, E, F, G, H, I, J, K)$, and the set of functional dependencies F :

$$F = \{ \begin{array}{l} B \rightarrow FEG \\ FE \rightarrow HD \\ I \rightarrow JK \\ CDI \rightarrow AB \end{array} \}$$

Mark the normal form of each relation. Show that the decomposition is in fact LJ (Lossless Join) and DP (Dependency Preserving).

ANEXO

Base de datos relacional que almacena información acerca de investigaciones y juicios.

Funcionario(f_DNI, f_nombre, f_ciudad)

// Datos de los funcionarios (tanto jueces como fiscales): DNI, nombre y ciudad de residencia

Ej: (1, "Paco", "Oviedo") – El funcionario 1 se llama Paco y vive en Oviedo

Ej: (2, "Carmen", "Gijón")

Fiscal(f_DNI, f_ámbito) [*f_DNI referencia Funcionario*]

// Datos de los fiscales: DNI y ámbito (civil, penal, social)

Ej: (1, "civil") – El funcionario 1 (Paco) es un fiscal de civil

Juez(f_DNI, j_número_juzgado, j_ámbito) [*f_DNI referencia Funcionario*]

// Datos de los jueces: DNI, número del juzgado del que es titular y ámbito del mismo (civil, penal, social)

Ej: (2, "5", "penal") – El juez 2 (Carmen) es titular del juzgado 5 de lo penal

Ciudadano(c_DNI, c_nombre, c_ciudad)

// Datos de los ciudadanos: DNI, nombre y ciudad de residencia

Ej: (3, "Alberto", "Oviedo") – El ciudadano 2 se llama Alberto y vive en Oviedo

Juicio(j_código, j_fecha, f_DNI-juez, f_DNI-fiscal, j_tipo) [*f_DNI-juez referencia Juez, f_DNI-fiscal referencia Fiscal*]

// Datos de los juicios: código, fecha, juez que juzga y fiscal que acusa en el juicio, tipo del juicio (corrupción, terrorismo, otros)

Ej: ("J-1", "12/05/2015", 2, 1) – El juicio J-1 se celebrará el 12/05/2015. El juez será el 2 (Carmen) y el fiscal el 1 (Paco)

Investigación(i_código, i_nombre_clave, i_código-principal) [*i_nombre_clave clave candidato*] [*i_código-principal referencia Investigación*]

// Datos de las investigaciones: código, nombre clave, investigación de la que es subinvestigación

Ej: ("I-2", "Pokemon", "I-1") – La investigación I-2 tiene nombre en clave Pokemon y es una subinvestigación de la investigación I-1 (Gortel).

Ej: ("I-1", "Gortel", -)

Imputado(i_código, c_DNI)

// Ciudadanos que están imputados en investigaciones

Ej: ("I-1", 3) – El ciudadano 3 (Alberto) está imputado en la investigación I-1 (Gortel)

Es_acusado(i_código, c_DNI, j_código, sentencia, multa)

// Ciudadanos que son acusados en un juicio derivado de una investigación, indicando la sentencia recibida (culpable, no culpable) y la multa aparejada.

// Ej: ("I-1", 3, "J-1", "culpable", 10M) – El ciudadano 3 (Alberto) es acusado en el juicio J-1 y salió culpable con 10 millones de euros de multa.

Vista(j_código, v_fecha, v_contenido, v_número) [*j_código+v_número clave candidato*]

// Vistas que se celebran de cada juicio, indicando el código del juicio, la fecha de la vista, el número que hace y su contenido)

Ej: ("J-1", "12/05/2015", "vista preliminar", 1) – El 12/05/2015 se celebró la vista preliminar del juicio J-1. Esta vista hace la primera de ese juicio.

Nota 1 : DEBEN usarse las abreviaturas
número : n // nombre : nom_ // código: cód

Nota 2 : Supónganse los dominios acostumbrados para
cada atributo

Nota 3 : Nombres alternativos para las tablas:

Funcionario fu | Juez jz | Fiscal fi | Ciudadano ci | Juicio
jc | Investigación iv | Imputado ip | Es_acusado ac | Vista vi