uc3m | Universidad Carlos III de Madrid

OpenCourseWare

Database

Lourdes Moreno López

Paloma Martínez Fernández

José Luis Martínez Fernández

Rodrigo Alarcón García

Evaluation test 4

Database.
Bachelor in Data Science and Engineering

Final Exam (4 p)

PROBLEM 1: Drug management (1,5 p.)

We want to design a database to control the economic costs derived from the consumption of drugs by patients as well as from the different services (oncology, digestive pediatrics, traumatology, etc.) that make up the hospital. Each patient admitted to the hospital consumes a set of drugs during the period of their hospitalization, the management of which will allow the generation of drug expenditure reports per patient, per service or per diagnosis

It is necessary to store the information related to the admissions of patients with the data of each admission made in specific hospital service, the drug consumption produced by a specific admission, and the general drug consumption generated by the activity of the services itself from the hospital (which are not assigned to a particular patient, e.g. saline, alcohol, bicarbonate, etc.).

Each patient is identified by his medical record number. Also, the name, the social security number (if any), the address, a telephone number and the date of birth must be stored.

A patient may have been admitted to the hospital more than once; each admission is characterized by a sequential number within each medical record number. In addition, the service in which the patient has been admitted, the diagnosis and the date of admission and the discharge date, if it had occurred, must be stored. Besides, the total expense of a patient's admission must be stored.

The consumption of drugs by hospital services and prescribed to patients is measured in number of single-doses (for example, each of the pills in a blister). For these drugs, we want to store a registration number, brand name, clinical name, chemical compound, supplier code, number of single-doses per container and the price per single-dose.

It is necessary to store the doctors who work in the hospital identified by their No. of collegiate. Likewise, the following data is stored: their name, address, a contact telephone number and the hospital service to which they are attached, considering that a doctor can only work in certain hospital service.

The information of which doctor prescribed which medicines to a patient must be stored, considering that during a patient admission, a doctor may have prescribed several drugs, but that a drug is only prescribed to a certain patient by a single doctor. A doctor may prescribe the same drug to a patient admitted multiple times on different dates, and an admitted patient may be prescribed different drugs by doctors.

Lastly, we want to record the information related to the check-up that doctors perform on a certain patient in a certain admission; the date, time and a small report will be saved.

You must:

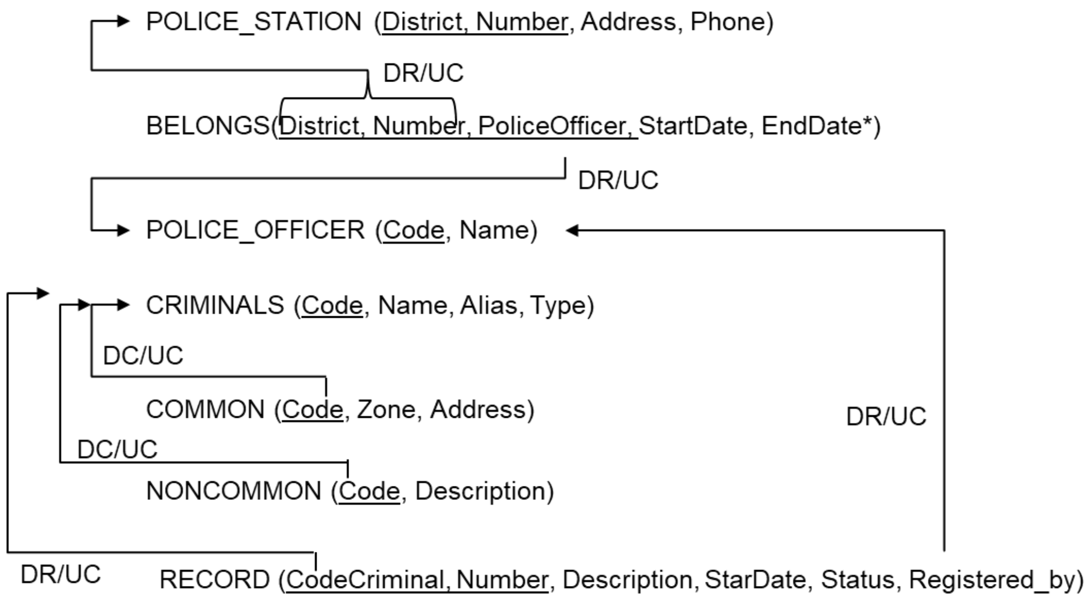
a) Obtain the relational schema/diagram according to requirements with the primary and alternative keys. Indicate the foreign keys with their delete and update options.
b) Write additional semantic assumptions to the statement, if needed
c) Write additional semantic assumptions to the scheme, if needed

PROBLEM 2. SQL QUERIES: POLICE (1,25 p.)

The police of a city want to create a database to keep a computerized control of the investigations they are conducting. In the city, there are several police stations, each of which is identified by the combination of the district to which it belongs, plus a number. A police officer can work in several city police stations at different time periods, but on a specific date, police can only be working in a single police station.

On the other hand, there is control of the different criminals who have been arrested in the locality. Criminals can be common or uncommon. Criminals have one or more records associated, which may or may not be open and that are registered by a police officer.

The schema of the database is as follows



POLICE_STATION (District, Number, Address, Phone)
DR/UC
BELONGS(District, Number, PoliceOfficer, StartDate, EndDate*)
DR/UC
POLICE_OFFICER (Code, Name)
CRIMINALS (Code, Name, Alias, Type)
DC/UC
COMMON (Code, Zone, Address)
DR/UC
DC/UC
NONCOMMON (Code, Description)
DR/UC
RECORD (CodeCriminal, Number, Description, StarDate, Status, Registered_by)

CRIMINAL.Type={common, noncommon}

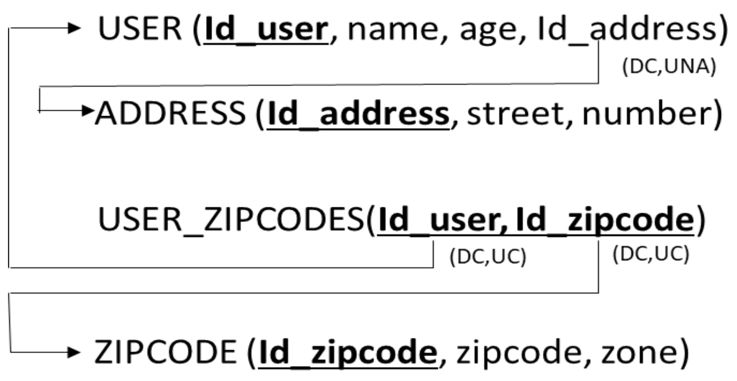
RECORD.Status ={open, close}

Following this scheme, make the following queries:

1. Select criminals (code and name) from open records of non-common criminals (0.5)
2. Select for each police station: district and the number of police officers belonging to that district ordered by the district. (0.25)
3. Select for each criminal: code, name, alias, type and the number of records associated.(0.25)
4. List ID and name of police officers assigned to the “Chamberí” district after 1/11/2019 (0.25)

PROBLEM 3: NoSQL JSON Document Databases (0,75 p.)

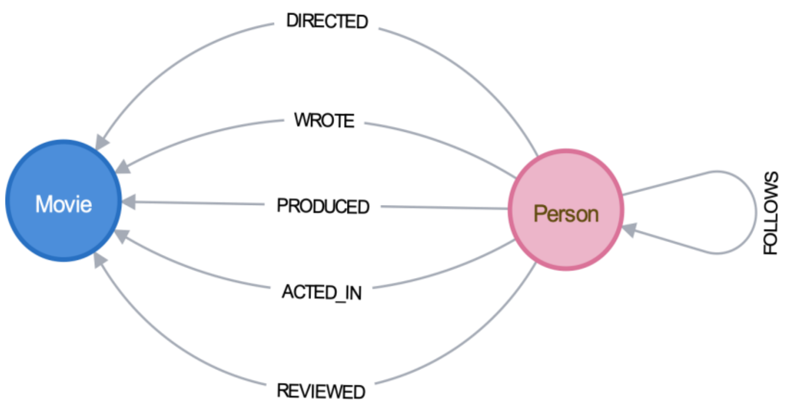
From the following relational schema, write JSON Documents for a document database like MongoDB following modeling guidelines.


USER (Id_user, name, age, Id_address)
(DC,UNA)
ADDRESS (Id_address, street, number)
USER_ZIPCODES(Id_user, Id_zipcode)
(DC,UC)
(DC,UC)
ZIPCODE (Id_zipcode, zipcode, zone)

PROBLEM 4: Graph Database (Neo4j) Test (0,5 p.)

Given the following schema of a graph database (Neo4j) and part of its creation script, answer the following quiz questions.

(There is only only one correct option. (Correct option: +0.1 p; Incorrect option: -0.05 p))



```
CREATE (TheMatrix:Movie {title:'The Matrix', released:1999, tagline:'Welcome
to the Real World'})

CREATE (TomH:Person {name:'Tom Hanks', born:1956})

CREATE
(TomH)-[:ACTED_IN {roles:['Hero Boy', 'Father', 'Conductor', 'Hobo',
'Scrooge', 'Santa Claus']}]->(ThePolarExpress),
(RobertZ)-[:DIRECTED]->(ThePolarExpress)

CREATE
(Keanu)-[:ACTED_IN {roles:['Neo']}]->(TheMatrix),

(LillyW)-[:DIRECTED]->(TheMatrix),
(LanaW)-[:DIRECTED]->(TheMatrix),
(JoelS)-[:PRODUCED]->(TheMatrix)


CREATE
(JamesThompson)-[:FOLLOWS]->(JessicaThompson),
(AngelaScope)-[:FOLLOWS]->(JessicaThompson),
(PaulBlythe)-[:FOLLOWS]->(AngelaScope)

CREATE
(JessicaThompson)-[:REVIEWED {summary:'An amazing journey', rating:95}]-
>(CloudAtlas),
(JessicaThompson)-[:REVIEWED {summary:'Silly, but fun', rating:65}]-
>(TheReplacements),
```

QUIZ

1. Select the query to display the schema from the database. Mark the right option.
   a) CALL db.schema().visualization
   b) MATCH (n) RETURN n
   c) MATCH (p:Person) RETURN p
   d) MATCH (p:Movie) RETURN p

2. Select the query to list all Tom Hanks movies. Mark the right option.
   a) MATCH (tom {name: "Tom Hanks"}) RETURN tom
   b) MATCH (tom:Person {name: "Tom Hanks"})-[:ACTED_IN]->(tomHanksMovies) RETURN tom,tomHanksMovies
   c) MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]->(m)<-[:ACTED_IN]- (coActors) RETURN coActors.name
   d) MATCH (a:Person)-[:ACTED_IN]->(:Movie) WHERE a.name STARTS WITH 'Tom Hanks' RETURN a.name

3. Given the following query. What data will it return? Mark the right option

```
MATCH (:Person)-[r:REVIEWED]->(m:Movie)
RETURN m.title AS movie, r.rating AS rating
ORDER BY r.rating DESC LIMIT 5
```

   a) Top 5 ratings and their associated movies returning the movie title and the rating are retrieved.
   b) Titles of the films whose rating equal to 5 are retrieved
   c) People who are reviewers and have assigned a rating equal to 5 are recovered
   d) The names of directors of films with the top 5 ratings are returned.

4. Select the query to display all people who have produced a movie, but have not directed a movie, returning their names and the movie titles. Mark the right option.

   a) MATCH (a:Person)-[:PRODUCED]->(m:Movie) WHERE NOT ((a)-[:DIRECTED]->(:Movie)) RETURN a.name, m.title
   b) MATCH (a:Person)-[:DIRECTED]->(m:Movie) WHERE NOT ((a)-[:PRODUCED]->(:Movie)) RETURN a.name, m.title
   c) MATCH (a:Person)-[:PRODUCED]->(m:Movie) WHERE NOT ((a)-[:DIRECTED]->(:Movie)) RETURN a. title, m. tagline
   d) MATCH (a:Person)-[:PRODUCED]->(m:Movie) WHERE NOT ((a)-[:DIRECTED]->(:Movie))

5. Given the following query. What data will it return? Mark the right option

```
MATCH (a:Person) WHERE a.born >= 1970 AND a.born < 1980
```

```
RETURN a.name as Name, a.born as `Year Born`
```

a) All people that were born in the 80’s returning their names and year born are retrieved.
b) All people that were born between 1970 and 1980 returning their names and year born are retrieved.
c) All people that were born in the 70’s returning their names are retrieved.
d) All the movies of people born in the 70's are retrieved.