
OpenCourseWare

Database

Lourdes Moreno López
Paloma Martínez Fernández
José Luis Martínez Fernández
Rodrigo Alarcón

Evaluation test 5 solution



Student's first and last name: _____

NIA: _____

PROBLEM 1: Non-governmental organization (NGO) (4 p.)

A non-governmental organization (NGO) is responsible for sending aid materials (medicines and food) and humanitarian aid (medical personnel) to refugee camps.

This organization obtains its income from its partners' fees. We want to know information regarding the partners such as an identifier, name, the bank account where the annual charges are made, the date of last payment and the type of fee. Currently there are three types of fees, which may vary in the future: minimum (10 euros per year), average (20 euros per year) or maximum (30 euros per year).

Each partner belongs to one of the headquarters of the organization, which are all located in different cities. Of course, a headquarter has many associated partners. Concerning each headquarter, we want to know an identifier code, its address, city and who is the manager.

Many volunteers work in the NGO doing different types of jobs. There are two types of volunteers and we want to store information about them: those who perform humanitarian tasks (health personnel) and those who perform administrative tasks (administrative staff). Regarding the former, we want to know apart from the personal data (identifier, name, phone number and address), their profession (doctor, nurse, etc.) and if they are currently available (yes / no). For administrative staff we only keep personal data. For both types of volunteers, the headquarters in which they are registered are also stored in the database.

The organization is responsible for sending shipments that include aid materials to different refugee camps. Each shipment has a destination (name of refugee camp) and an exit date. To identify shipments, they are assigned a unique code. Each shipment contains batches of aid materials. These aid materials are identified by a code, characterized by a description and the type (food or medicines). Batches could include food (number of tons of each food must be known) or medicines (the number of units of each medicine must be known). We want to know the different types of aid material included in each shipment taking into account that each batch of aid material can only be assigned to a shipment. There can be aid material that has not yet been assigned any shipment. We also want to register which headquarters organized each shipment.

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

SOLUTION

PARTNER (Cod, NIF, Personal_data, Bank_account, Date_payment, Fee_type, Cod_headquarter)

DNA/UC

→ HEADQUARTER (Cod, Address, City, Name_director) ←

DNA/UC

→ VOLUNTEER (Cod, name, phone, address, Cod_headquarter, Type)

DC/UC

HEALTH_PERSONNEL (Cod_volunteer, Profession, Is_available)

→ SHIPMENT (Cod, Destination, Exit date)

DNA/UC

ORGANIZE (Cod_shipment, Cod_headquarter)

DNA/UC

MATERIAL_AID (Cod, Description, Type) ←

BATCH (Cod_shipment, Cod_material_aid, Num)

DNA/UC

DNA/UC

ADDITIONAL SEMANTIC ASSUMPTIONS TO THE STATEMENT:

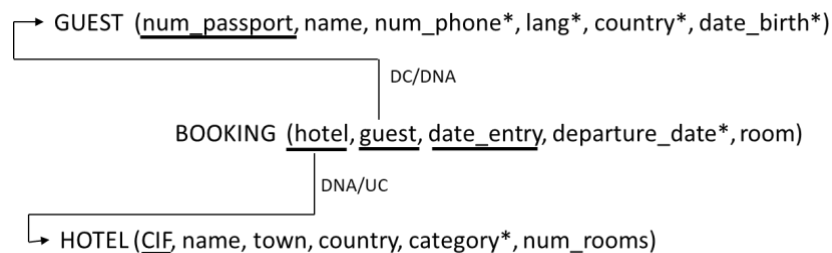
- We have assumed that a PARTNER is identified by a Code, and NIF as alternative identificatory

ADDITIONAL SEMANTIC ASSUMPTIONS TO THE SCHEME:

- PARTNER. Fee_type = {minimum, average, maximum}
- VOLUNTEER. Type = {health personnel, administrative staff}
- HEALTH_PERSONNEL.Profession = {doctor, nurse}
- HEALTH_PERSONNEL.Is_available = {yes,no}
- MATERIAL_AID. Type = {food, medicine}
- A trigger will be needed to check that a shipment contains only products of one type (food or medicines)

PROBLEM 2. SQL QUERIES: (3 p.): Booking a hotel

The schema of the database is as follows



Following this scheme, write the following queries using standard SQL:

1. Names and telephone numbers (num_phone) of guests that have booked Solymar Hotel since the 1st of March of 2021

```
SELECT name, num_phone FROM GUEST G, BOOKING B, HOTEL H
WHERE G.num_passport = B.guest AND B.hotel = H.CIF AND
HOTEL.name = 'Solymar Hotel' AND
B.date_entry > '1/03/2021'
```

2. Number of bookings per hotel starting after the 1st of July of 2021

```
SELECT COUNT(*) FROM BOOKING GROUP BY B.hotel HAVING date_entry > '1/07/2021'
```

3. Hotels (CIFs and names) that have not been booked by visitors from United Kingdom

```
SELECT CIF, name FROM HOTEL WHERE CIF NOT IN (SELECT hotel FROM BOOKING, GUEST  
WHERE BOOKING.guest = GUEST.num_passport AND GUEST.country = 'United Kingdom')
```

PROBLEM 3: NoSQL JSON Document Databases (1 p.)

From the relational schema of problem 2, write JSON Documents for a document database like MongoDB modeling guidelines.

SOLUTION

```
/* 1 */  
{  
  "_id" : "Guest1",  
  "num_passport" : "X123456",  
  "name" : "John Doe",  
  "num_phone" : "25",  
  "lang" : "en",  
  "country" : "UK",  
  "date_birth" : "5/5/2000",  
  "booking" : [  
    {  
      "hotel1",  
      "date_entry" : "5/5/2021",  
      "departure_date" : "11/5/2021",  
      "room" : "325"  
    },  
    {  
      "hotel2",  
      "date_entry" : "7/1/2021",  
      "departure_date" : "12/1/2021",  
      "room" : "456"  
    }  
  ]  
}  
  
/* 2 */  
{  
  "_id" : "hotel1",  
  "CIF" : "A231456",  
  "name" : "Hotel Solymar",  
  "town" : "Hamburgo",  
  "country" : "de",  
  "category" : "3",  
  "num_rooms" : "240"  
}  
  
/* 3 */  
{  
  "_id" : "hotel2",
```

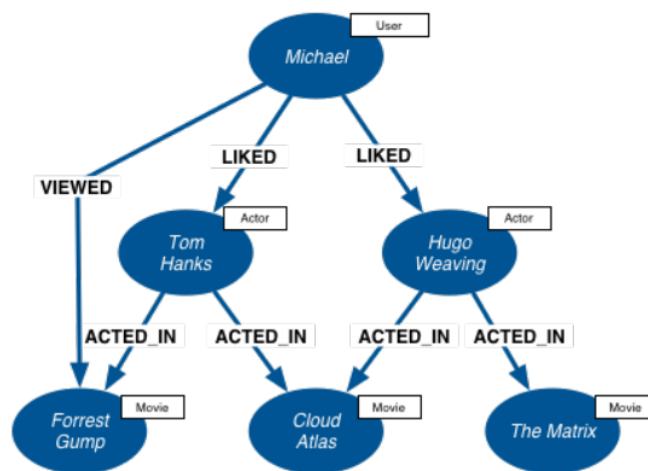
```

"CIF" : "B876543",
"name" : "Hotel LaPlaya",
"town" : "Málaga",
"country" : "es",
"category" : "4",
"num_rooms" : "520"
}

```

PROBLEM 4: Graph Database (Neo4j) (2 p.)

Assume a Movies Database with the following data model:



```

(:Movie {title, released, ... })
(:Person {name, born, ... })
(:Person)-[:ACTED_IN|:DIRECTED|:PRODUCED]->(:Movie)

```

Provide queries for the following questions:

1. Find users who like Tom Hanks and Adam Sandler.

```

MATCH (tom:Person {name: "Tom Hanks"}, Adam:Person {name: "Adam Sandler"})<-[:LIKED] -
(hanksSandlerFans) RETURN hanksSandlerFans.name

```

2. Find actors who also have directed a movie.
MATCH (actor:Person) - [:ACTED_IN] -> (:Movie)
WHERE actor - [:DIRECTED] -> (:Movie)
RETURN actor.name

3. Find actors who also have directed a movie they were acting in.

```

MATCH (actor:Person) - [:ACTED_IN] -> (movie) <- [:DIRECTED] - (actor) RETURN actor.name

```