"BASES DE DATOS Y SISTEMAS DE INFORMACIÓN" EXAM 28/11/2016

UNITS 2 AND 3

Consider the following relational schema about marathon races around the Word.

CITY (city_code: char(15), name: char(50), country: char(20), history: char(200))
PK: {city_code}
NNV: {name, country}

MARATHON (mcode: int, name: char(20), mdate: date, city_code: char(15), edition: int)
PK: {mcode}
NNV: {mdate, edition, city_code}
FK: {city_code} →CITY
UNI: {city_code, edition}

RUNNER (ssn:int, name:char(40), age:int, country:char(20), gender:char(1))
PK: {ssn}
NNV: {name, age, country, gender}

HAS_RUN(mcode: int, ssn: int, rtime: time)
PK: {mcode, ssn}
FK: {mcode}→MARATHON
FK: {ssn} →RUNNER
NNV: {rtime}

INCIDENCE(num: int, mcode: int, description: char(100), km: int)
PK: {mcode, num}
FK: {mcode}→MARATHON
NNV: {km, description}

Where the meaning of the relations is the following::

- **City**:
  - *city_code*: Code of the city
  - *name*: Name of the city
  - *country*: Country where the city is
  - *history*: Short history of the city
- **Marathon**:
  - *mcode*: Code of the marathon
  - *name*: Name of the marathon race
  - *mdate*: Date of the race
  - *city_code*: Code of the city where the marathon is run
  - *edition*: Number of the edition of that marathon in the city
- **Runner**:
  - *ssn:* SSN of the runner
  - *name*: Name of the runer
  - *age*: Age of the runner
  - *country*: Country where the runner was born
  - *gender*: {M,F}
- **Incidence**:
  - *mcode*: Code of the marathon
  - *num*: Number of the incidence
  - *description*: What has happened
  - *km*: Km where the incidence has happened
- **Has_run**: The runner with SSN ***ssn*** has run the marathon coded ***mcode*** in a total time of ***rtime*.**

1) Define the following concepts (0.6 points):
   a) conceptual schema
   b) logical schema,
   c) internal schema.

2) Write the following queries in SQL:

   a) List, for the marathons with 2 or more incidences before the km #25, the code and name of the marathon. List also the name and the country of the city where the marathon has taken place. (0.6 points)

   b) List, for the marathons where have run at least one woman, the code and name of the marathon, indicating also the ssn and name of the female runner who has run that marathon in the lowest time. (0.6 points)

   c) List the code and name of the marathon with the most (highest number of) runners (0.6 points)

   d) List the ssn and name of the runners who have run in less than 02:50:00 all the marathons of his/her country (if there is at least one marathon in his/her country). (0.8 points)

   e) List the code and name of all the cities in the database which are in a country with 10 or more runners. List also how many marathons have been run in each of those cities before January 1st, 2000. (0.8 points)

ANSWERS

1)

- **Conceptual schema:** Description of the information system from the organizational point of view, independently of the used DBMS and of the use or not of database techniques.
- **Logical schema:** Database definition in terms of the data model used in the DBMS, without including any detail about the physical representation of the database.
- **Internal (physical) schema:** Database description in terms of its (physical) representation in secondary memory.

2.a)

```
SELECT M.mcode, M.name, C.name, C.country
FROM Marathon M, City, C
WHERE M.city_code = C.city_code AND
      (SELECT COUNT(*)
       FROM Incidence I
       WHERE I.mcode = M.mcode AND km < 25) > 1;

-- alternative

SELECT M.mcode, M.name, C.name, C.country
FROM Marathon M, City, C, Incidence I
WHERE M.city_code = C.city_code AND I.mcode = M.mcode AND km < 25
GROUP BY M.mcode, M.name, C.name, C.country
HAVING COUNT(*)> 1;
```

2.b)

```
SELECT M.mcode, M.name, R.ssn, R.name
FROM Marathon M, Has_run H, Runner R
WHERE M.mcode = H.mcode AND H.ssn = R.ssn AND R.gender = 'F' AND
      H.rtime = (SELECT MIN(H.rtime)
                 FROM Has_run H, Runner R
                 WHERE M.mcode = H.mcode AND H.ssn = R.ssn AND R.gender = 'F');
```

2.c)

```
SELECT M.mcode, M.name
FROM Marathon M, Has_run H
WHERE M.mcode = H.mcode
GROUP BY M.mcode, M.name
HAVING COUNT(*) = (SELECT MAX(COUNT(*))
                   FROM Has_run H
                   GROUP BY H.mcode);
```

2.d)

```
SELECT R.ssn, R.name
FROM Runner R
WHERE NOT EXISTS (SELECT * FROM Marathon M, City CI
                  WHERE M.city_code = CI.city_code AND CI.country = R.country AND
                        NOT EXISTS (SELECT *
                                    FROM Has_run H
                                    WHERE H.ssn = R.ssn AND M.mcode = H.mcode AND
                                          H.rtime < '02:50:00'))
      AND EXISTS (SELECT * FROM Marathon M, City CI
                  WHERE M.city_code = CI.city_code AND CI.country = R.country);
```

```
-- alternative

SELECT R.ssn, R.name
FROM Runner R
WHERE (SELECT COUNT(*) FROM Marathon M, City CI
       WHERE M.city_code = CI.city_code AND CI.country = R.country)
             =
      (SELECT COUNT(*) FROM City CI, Marathon M, Has_run H
       WHERE CI.country = R.country AND CI.city_code = M.city_code AND H.ssn = R.ssn
       AND  H.mcode = M.mcode AND H.rtime < '02:50:00')
 AND
      (SELECT COUNT(*) FROM Marathon M, City CI
       WHERE M.city_code = CI.city_code AND CI.country = R.country) > 0 ;

-- alternative

SELECT R.ssn, R.name
FROM Runner R, City CI, Marathon M, Has_run H
WHERE R.country = CI.country AND CI.city_code = M.city_code AND M.mcode = H.mcode AND
      H.ssn = R.ssn AND H.rtime < '02:50:00'
GROUP BY R.ssn, R.name
HAVING COUNT(*) = (SELECT COUNT(*) FROM Marathon M, City CI
                   WHERE M.city_code = CI.city_code AND CI.country = R.country);

2.e)
SELECT C.city_code, C.name, COUNT(M.mcode)
FROM City C LEFT JOIN  Marathon M ON C.city_code=M.city_code AND M.mdate<'01-01-2000'
WHERE C.country IN
      (SELECT X.country
       FROM Runner X
       GROUP BY X.country
       HAVING COUNT(*) >=10)
GROUP BY C.city_code, C.name ;

-- alternative

SELECT C.city_code, C.name, COUNT(M.mcode)
FROM City C LEFT JOIN  Marathon M ON C.city_code=M.city_code AND M.mdate<'01-01-2000'
GROUP BY C.city_code, C.name, C.country
HAVING C.country IN
       (SELECT X.country
        FROM Runner X
        GROUP BY X.country
        HAVING COUNT(*) >=10) ;

-- alternative

SELECT C.city_code, C.name, COUNT(M.mcode)
FROM City C, Marathon M
WHERE C.city_code=M.city_code AND M.mdate<'01-01-2000'
GROUP BY C.city_code, C.name, C.country
HAVING (SELECT COUNT(*)
        FROM Runner X
        WHERE C.country = X.country)  >=10
UNION
SELECT C.city_code, C.name, 0
FROM City C
WHERE (SELECT COUNT(*)
       FROM Runner X
       WHERE C.country = X.country) >=10 AND
             C.city_code NOT IN  (SELECT M.city_code
                                  FROM Marathon M
                                  WHERE M.mdate<'01-01-2000') ;
```

```
-- A different method to solve this query

SELECT C.city_code, C.name,   (SELECT COUNT(M.mcode)
                               FROM Marathon M
                               WHERE C.city_code=M.city_code
                                 AND M.mdate<'01-01-2000')
FROM City C
WHERE C.country IN (SELECT X.country
                    FROM Runner X
                    GROUP BY X.country
                    HAVING COUNT(*)>=10) ;
```