Databases and Information Systems mid-term exam (29/11/21)

An art gallery wants to organize the information of its rooms, the artworks (paintings or sculptures) that are exhibited in each room, and the people in the security team. For this purpose, it has designed a relational database. The logical schema of the database is as follows:

ROOM(*room_code*: char(4), *room_name*: char(20), *benefactor*: char(30))
PK: {room_code} NNV: {room_name}
ARTIST(*art_code*: integer, *name*: char(30), *yearbirth*: integer, *style*: char(15))
PK: {art_code} NNV: {name}
ARTWORK(*work_code*: integer, *year*: integer, *type*: char(15), *art_code*: integer, *room_code* : char(4))
PK: {work_code} NNV: {room_code, art_code}
FK: {art_code} → ARTIST
FK: {room_code} → ROOM
GROUP(*group_code*: char(4), *name*: char(30*), level*: integer)
PK: {group_code} NNV: {name}
GUARD(*dni*: char(9), *name*: char(30), *age*: integer, *group_code*:char(4))
PK: {dni} NNV: {name}
FK: {group_code} → GROUP
WATCH (*room_code*: char(4), *dni*: char(9), *turn*: char(10))
PK: {room_code, dni} NNV: {turn}
FK: {room_code} → ROOM
FK: {dni} → GUARD

Where the relations have the following meaning:

- **Room**
  - *room_code:* Code of the room.
  - *room_name:* Name of the room.
  - *benefactor:* Name of the benefactor who provides economic support of the room.
- **Artist**
  - *art_code:* Code of the artist.
  - *name:* Name del artist.
  - *yearbirth:* Year of the artist's birth.
  - *style: A*rtistic style of the artist.
- **Guard**
  - *dni*: DNI of the guard.
  - *name*: Name and surname of the guard.
  - *age*: Age of the guard.
  - *group_code*: Code of the security group to which the guard belongs.
- **Artwork**
  - *work_code:* Code of the artwork.
  - *year:* Year in which the artwork was created.
  - *type*: Painting or sculpture.
  - *art_code:* Code of the artist who made the artwork.
  - *room_code:* Code of the room where the artwork is exhibited.
- **Group**
  - *group_code:* Security group code.
  - *name:* Security group name.
  - *level:* Level of the security group.
- **Watch**
  - *room_code*: Code of the room.
  - *dni*: dni of the room security guard.
  - *turn*: Morning, Afternoon or Evening.

1) Solve the following queries in SQL:

   a) Obtain the code and the name of the rooms in which all the artworks exhibited are by artists born in the 20th century. *(0,5 points)*

   b) Obtain the name of the youngest guard who does not belong to any security group. *(0.4 points)*

   c) Obtain the room code, the room name and the number of artworks exhibited in the room where the largest number of artworks is exhibited. *(0,5 points)*

   d) Obtain the name and year of birth of the artists who have created the oldest artworks exhibited in the gallery. *(0,5 points)*

   e) For all rooms with more than 5 artworks exhibited, obtain the code of the room, its name and the number of guards assigned to it. *(0,7 points)*

   f) Obtain the list in alphabetical order of the names of the 'Impressionist' style artists who have some artwork exhibited in the gallery of which the year is unknown. *(0.5 points)*

   g) Obtain the code and the name of the security groups such that the average age of their guards is less than 45 years and all of them are in charge of the security of a room. *(0,7 points)*

2) Write the instruction to add a new guard whose name is 'Carmen Mola Más' whose dni is '40634505H', is 40 years old, and we do not know which security group she belongs to. *(0,2 points)*

SOLUTIONS

1 a) Obtain the code and the name of the rooms in which all the artworks exhibited are by artists born in the 20th century.

```
SELECT R.room_code, R.room_name
FROM Room R
WHERE NOT EXISTS (SELECT *
                  FROM Artwork AW, Artist A
                  WHERE AW.room_code=R.room_code
                   AND A.art_code=AW.art_code
                   AND NOT (A.yearbirth BETWEEN 1900 AND 1999))
AND EXISTS (SELECT * FROM Artwork AW
            WHERE AW.room_code=R.room_code)

//Alternative:

SELECT R.room_code, R.room_name
FROM Room R
WHERE NOT EXISTS (SELECT *
                  FROM Artwork AW JOIN Artist A ON AW.art_code = A.art_code
                  WHERE AW.room_code= R.room_code
                   AND (A.yearbirth<1900 OR A.yearbirth>2000))
      AND EXISTS (SELECT *
                  FROM Artwork AW
                  WHERE AW.room_code= R.room_code);

// Alternative:

SELECT R.room_code, R.room_name
FROM Room R
WHERE NOT EXISTS (SELECT *
                  FROM Artwork AW
                  WHERE AW.room_code=R.room_code
                   AND AW.art_code NOT IN (SELECT A.art_code
                                           FROM Artist A
                                           WHERE A.yearbirth BETWEEN 1900 AND 1999))
AND EXISTS (SELECT *
            FROM Artwork AW
            WHERE AW.room_code=R.room_code);
```

1b) Obtain the name of the youngest guard who does not belong to any security group.

```
SELECT G.name
FROM Guard G
WHERE G.group_code IS NULL
 AND G.age = (SELECT min(age)
              FROM Guard G
              WHERE G.group_code IS NULL)
```

1c) Obtain the room code, the room name and the number of artworks exhibited in the room where the largest number of artworks is exhibited.

```
SELECT R.room_code, R.room_name, COUNT(AW.work_code)
FROM Room R, Artwork AW
WHERE AW.room_code=R.room_code
GROUP BY R.room_code, R.room_name
HAVING COUNT(AW.work_code) = (SELECT MAX (COUNT(AW1.work_code)
                             FROM Artwork AW1
                             GROUP BY AW1.room_code)
```

1d) Obtain the name and year of birth of the artists who have created the oldest artworks exhibited in the gallery.

```
SELECT distinct A.name, A.yearbirth
FROM Artist A JOIN Artwork AW ON A.art_code=AW.art_code
WHERE AW.year = (SELECT MIN(year) FROM Artwork)

// Alternative:

SELECT distinct A.name, A.yearbirth
FROM Artist A JOIN Artwork AW ON A.art_code=AW.art_code
              JOIN Room R ON AW.room_code = R.room_code
WHERE AW.year = (SELECT MIN(year) FROM Artwork)

// Alternative:

SELECT A.name, A.yearbirth
FROM Artist A
WHERE A.art_code IN (SELECT AW.art_code
                     FROM Artwork AW
                     WHERE AW.year = (SELECT MIN(year) FROM Artwork));
```

1e) For all rooms with more than 5 artworks exhibited, obtain the code of the room, its name and the number of guards assigned to it.

```
// Using LEFT JOIN:

SELECT R.room_code, R.room_name, COUNT(W.dni) AS GUARDS
FROM Room R LEFT JOIN Watch W ON R.room_code=W.room_code
WHERE (SELECT COUNT(*)
       FROM Artwork AW
       WHERE AW.room_code=R.room_code) > 5
GROUP BY R.room_code, R.room_name;

// Alternative:

SELECT R.room_code, R.room_name,
     (SELECT COUNT(*) FROM Watch W
      WHERE R.room_code=W.room_code) AS GUARDS
FROM Room R
WHERE (SELECT COUNT(*)
       FROM Artwork AW
       WHERE AW.room_code=R.room_code) > 5
```

1f) Obtain the list in alphabetical order of the names of the 'Impressionist' style artists who have some artwork exhibited in the gallery of which the year is unknown.

```
SELECT A.name
FROM Artist A
WHERE A.style = 'Impressionist'
 AND EXISTS (SELECT *
             FROM Artwork AW
              WHERE AW.art_code=A.art_code AND AW.year IS NULL)
ORDER BY A.name;

//Alternative:

SELECT distinct A.name
FROM Artist A, Artwork AW
WHERE A.style = 'Impressionist'
  AND AW.art_code=A.art_code
  AND AW.year IS NULL
ORDER BY A.name;
```

1g) Obtain the code and the name of the security groups such that the average age of their guards is less than 45 years and all of them are in charge of the security of a room.

```
SELECT GR.group_code, GR.name
FROM Group GR
WHERE NOT EXISTS (SELECT * FROM Guard G
                  WHERE G.group_code=GR.group_code
                    AND NOT EXISTS (SELECT * FROM Watch W
                                    WHERE G.dni=W.dni))
     AND (SELECT AVG(G2.age)
          FROM Guard G2
          WHERE G2.group_code=GR.group_code)<45
```

2 Write the instruction to add a new guard whose name is 'Carmen Mola Más' whose dni is '40634505H', is 40 years old, and we do not know which security group she belongs to.

```
INSERT INTO Guard(name, dni, age)
VALUES ('Carmen Mola Más', '40634505H', 40);

//Alternative

INSERT INTO Guard(name, dni, age, group_code)
VALUES ('Carmen Mola Más', '40634505H', 40, null);

//Alternative

INSERT INTO Guard
VALUES ('40634505H', 'Carmen Mola Más', 40, null);
```