Statistical Methods for Database Integration
Examination
DATABASES

The exam consists in two parts:

1) Part A: **The exam is closed-book, closed-notes**;

2) Part B: **You are allowed to use lecture and labs notes**.

Each questions is assigned points expressed in cents.

PART A

Ex. 1

(a) **(10 points)** "Every year, millions of people visit museums, from the most famous to the lesser known, representing a significant industry of a country's economy. The table compare the **number of museums and visitors in 2017** in some countries of the European Union, with different historical, cultural and geographical characteristics."

		# Museums			
Nation	**Population**	**Art, Archaeology, History**	**Science and Technology**	**Others**	**Visitors**
France	64.694.497	805	341	78	63.199.181
Italy	60.589.445	2.656	1.184	1.136	50.263.520

Table 1: Cultural Statistics - 2017

Describe by means of the XML language the data. Specify only one country (at your choice), but **do not omit to specify all deduced information**.

A significant information is the average visitors per museum. Do we have to add this information to the XML description?

[*Sol.*:]

```
<statistics year="2017">
   <nation>
      <name> France </name>
      <population> 64694497 </population>
      <museums unit="number">
         <art-arch-hist> 805 </art-arch-hist>
         <science-tech> 341 </science-tech>
         <others> 78 </others>
      </museums>
      <visitors unit="number"> 63199181 </visitors>
   </nation>
</statistics>
```

It is not necessary to add this information because the average visitors per museum is the number of visitors over the total number of museums.

(b) **(10 points)** Assume to converge part of the data in the table of the point (a) into a single "relation", specifically **year**, **nation**, **museum category**, **number of museums**.

Write the SQL statement that creates the relation with the advice to define a 'natural' key (avoid to use ID), and furthermore write SQL statements to enter at least two tuples.

[*Sol.*]

```
CREATE TABLE statMuseums(
    year CHAR(4),
    nation VARCHAR(50),
    category VARCHAR(100),
    number NUMERIC(5,0),
    PRIMARY KEY(year, nation, category)
);

INSERT INTO statMuseums VALUE('2017', 'France', 'science-tech', 341);
INSERT INTO statMuseums VALUE('2017', 'Italy', 'science-tech', 1184);
```

(c) **(Optional: 5 points)** In order to manage the huge quantity of data coming essentially from the Web, traditional DBMS are not adequate considering that this data is typically unstructured. **NoSQL Database Management Systems** is the emerging technology specifically designed to manage unstructured data.

What are the main difference between NoSQL and traditional DBMS?

[*Sol.*: See teaching material]

Es. 2 - Data Modeling

(1) **(35 points)** "Current health emergency entails to different approaches to visit museums and expositions. Most expositions become available virtually, giving the opportunity to visitors to explore by an 'app' main work of art collections".

The aim is to design a database to support the app, the gate to explore virtually the work of arts.

Draw the E/R diagram that capture the requirements stated below. **Use "ID" as key only if strictly necessary**.

(a) Basically an exposition shows different **work of arts** (painting, sculture, new form of art like videos, images, ...), each one has a title and a description, moreover the year it has been realized.

(b) Work of arts have been realized by different **artists**, for which it is essential to know name and surname, date of birth (if known), nationality and the date of death (if it occurred).

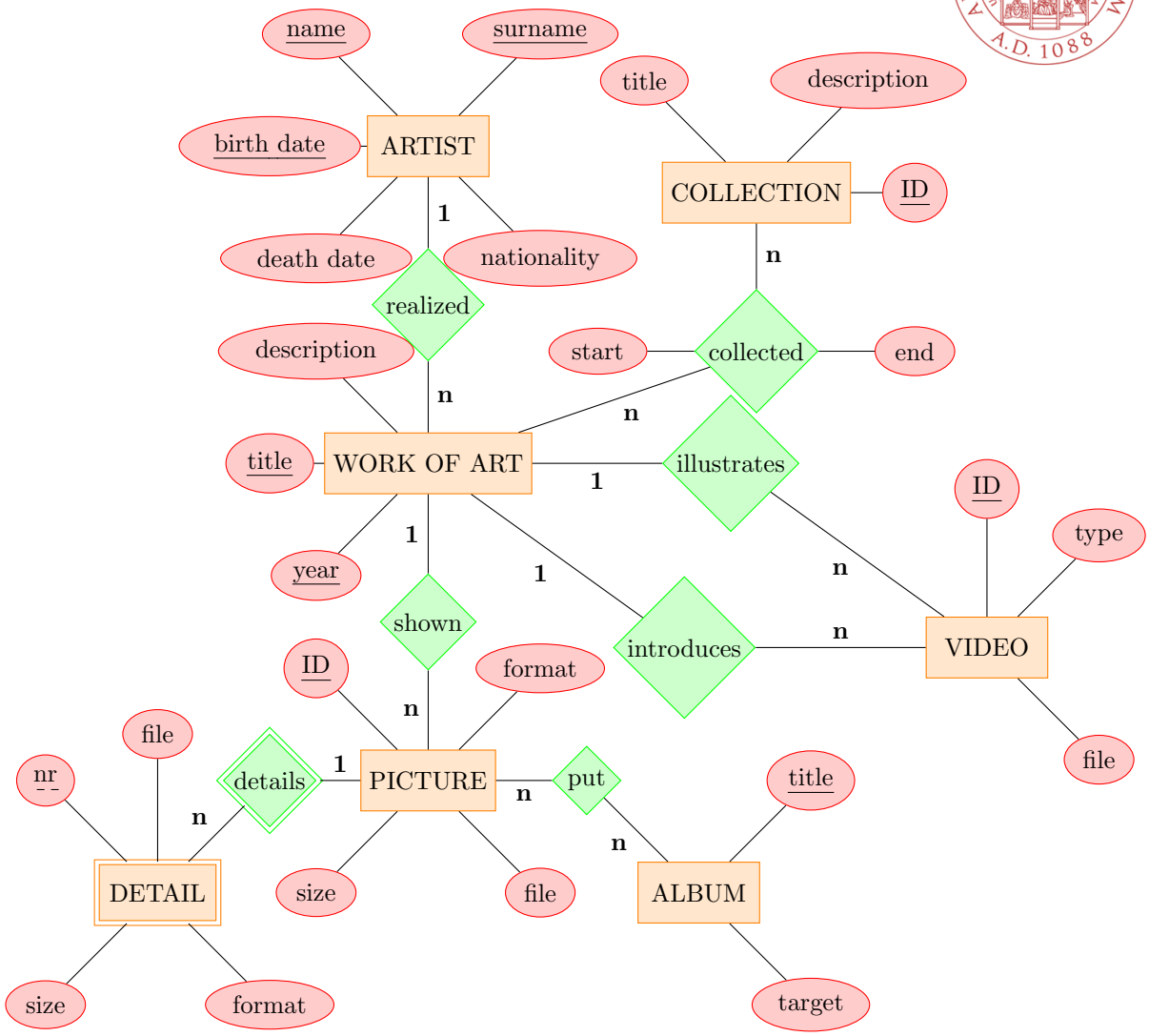
(c) Work of arts are grouped by **theme (or collection)**, which could be the criteria to select work of arts to visit virtually (for instante paintings of '900). For example a collection has a title and general notes. Along time work of arts can be grouped in different collections, so a starting and ending year of inclusion must be tracked.

(d) For each work of art we have a wide selections of **pictures**. A picture, has a specific format and size and when it has been taken. Obviously pictures are in the database.

(e) Frequently a picture has many pictures showing **details**, in order to give the opportunity to explore all significant details of the work of art. These pictures have the same characteristics of the main pictures and strictly identified through the main picture (**Tip:** consider weak entity set model).

(f) Furthermore pictures are put together to form an **album**, to browse quickly the most attractive pictures. An album has a title, and the target audience (young people, professionals, ...)

(g) Finally the virtual visit offer a wide selection of **videos**, some of them aiming to tell and show the work of art, other specifically realized to give introductory information. These specificities must be captured by different relationships. Obviously videos are in the database.



(2) **(Optional: 5 points)**. Write the SQL statement to `CREATE` the "relation" that describes all **picture - detail** items. [**Tip**: in order to store media - picture, video - SQL offers the data type `BLOB`]

```
CREATE TABLE detail(
    nr INT,
    format VARCHAR(30),
    size NUMERIC(12,3),
    file BLOB,
    picID CHAR(10),
    FOREIGN KEY (picID) REFERENCES picture(ID),
    PRIMARY KEY(nr, picID)
);
```

PARTE B

Es. 3 - SQL (45 points) Let assume the database "online-market".

(1) Menu(name, description, main)

(2) Food(name, unit, weight, label, price, startDate, endDate, Menu.name)

(3) GiftBasket(name, description)

(4) BasketCombines(GiftBasket.name, Food.name, Food.unit, Food.weight)

Questions

1) You aim to offer a basket "special edition", hence you identify among your food products those related to creams or olives (**Tip:** the menu they belong to has in its name one of the two keywords: **cream** or **olive**).

   Foods should be available for sales, and not yet used to prepare any other basket. Show returned foods from the most expensive to the cheapest one.

   [*Sol.*]

```
SELECT *
  FROM food
 WHERE (menu_name LIKE '%cream%'
    OR menu_name LIKE '%olive%')
   AND endDate IS NULL
   AND NOT EXISTS (SELECT *
                     FROM basketCombines
                    WHERE food_name = food.name
                      AND food_unit = food.unit
                      AND food_weight = food.weight)
ORDER BY price DESC;
```

2) Realize a comparative selection of your food products. The selection is based on **cost/benefit** parameter and on the level of **nutritive** we can benefit, specifically *cost/benefit parameter* equals the ratio price/weight, while *nutritive benefit* is the 30% of the weight. Show those foods which have the *cost/benefit parameter* larger than its average (over all foods) **or** the *nutritive benefit* larger than its average (over all foods).

[*Sol.*]

```
SELECT name, price/weight AS comparative, weight*0.30 AS nutritive
  FROM food
 WHERE price/weight > (SELECT AVG(price/weight)
                         FROM food)
    OR weight*0.30 > (SELECT AVG(weight*0.30)
                        FROM food);
```

3) The goal is to verify if in our food products proposal we can reach different people targets, so we compute two different key numbers. The first one is for each main menu the number of foods having a price smaller than or equal to 5 € (adding the label 'cheaper') while the second one, again for each main menu, the number of foods having a price larger than or equal to 14 € (adding the label 'expensive').

Both results must be returned together.

[*Sol.*]

```
(SELECT 'cheaper' AS type, menu.main, COUNT(*) AS nr
   FROM food, menu
  WHERE food.menu_name = menu.name
    AND price <= 5.0
GROUP BY menu.main)
UNION
(SELECT 'expensive' AS type, menu.main, COUNT(*) AS nr
   FROM food, menu
  WHERE food.menu_name = menu.name
    AND price >= 14.0
GROUP BY menu.main)
```