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)** "Eurostat offers different types of data related to social aspects of life. We aim to investigate on intentional homicide victims - distinct by male and female - trends in the European countries and in the main cities. Firstly we compare the number of victims in Germany and Italy in the year 2018. Data is arranged in the table".

2018	Total	Male (%)	Female (%)
Germany	788	53.4%	46.6%
Italy	345	61.4%	38.6%

Table 1: Intentional homicide victims

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**.

We need male and female total units! Is it necessary to describe them?

[*Sol.*:]

```
<homicide>
   <country>
      <name> Germany </name>
      <year> 2018 </year>
      <numbers>
         <total> 788 </total>
         <male unit="%"> 53.4 </male>
         <female unit="%"> 46.6 </female>
      </numbers>
   </country>
</homicide>
```

It is not necessary to describe male and female as total units because these numbers can be easily computed.

(b) **(10 points)** Assume to transform the table in the point (a) into a single "relation", that could be seen as **a data set**. Write **the relation** specifying for each identified attribute the corresponding **data type** (refer to the SQL data types).

Male and *Female* header shall be assumed to be values not variables of the data set, and the corresponding values should be expressed both as total units and as percentage.

[*Sol.*]

```
Homicide(country: VARCHAR(50),
            year: INT,
          gender: VARCHAR(10),  [gender = {Male, Female}]
           units: NUMERIC(5,0),
      percentage: NUMERIC(4,2)
     )
```

(c) **(Optional: 5 points)** Currently the Web is one of the most important data source, but most of data is classified as semi-structured or unstructured data. A DBMS can support the management for this type of data only partly therefore a new and different technology should be considered. *Describe shortly the new technology for databases.*

[*Sol.*: 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** are specifically designed to manage non-relational data.]

Es. 2 - Data Modeling

(1) **(35 points)** "Current health emergency entails to different approaches to promote products. Wine sales particularly suffer the emergency, therefore wineries rely on digital solutions".

The aim is to design a database to support winery societies to promote wine productions, in an innovative way by means of apps for smartphone.

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

(a) A list of **wines** to promote is registered: name, land (area), type of grape, type of production (DOC, bio, ...).

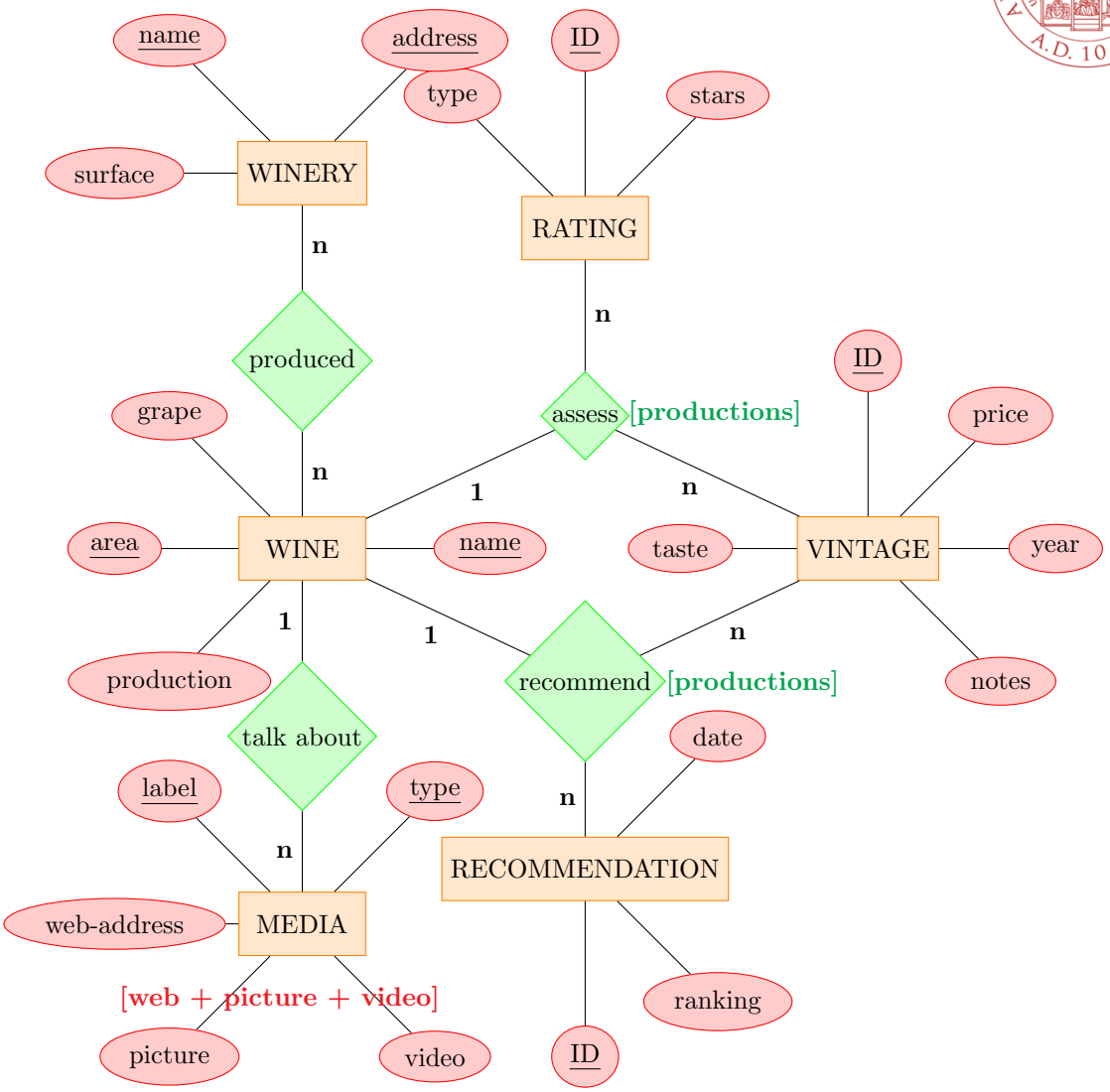
(b) For each **winery** - associated to the society -, which aims to promote its wines, is tracked the name, the address, and the surface yearly harvested.

(c) For this promotional campaign digital items can support it. Specifically items could be web sites, reviewing wines, or could be nice pictures or even nice videos. **For maintenance reasons we design a single entity to register all these items**.

(d) Promotions should be designed for each wine-**vintage** (harvest year), for these reasons additional data are needed: harvest year, additional notes (dry summer, excellent harvest,...), comments about the taste, and the price. The design should take in consideration that for each wine there could be many vintages along time, but that vintage description is specific for just one wine.

(e) Each "navigator" of the app is requested to assign voluntarily a **rating**, declaring for each characteristic (the color, the taste, the price...) a number of stars (from one to five). The rating can be expressed only for the couple *wine - vintage*.

(f) This app is designed to be innovative, therefore a *recommender system* is set up. The systems outputs **recommendations** based on the "navigator" selections and rating expressed, and consist in wine rankings. Recommendations are registered as a rank and the date it has been processed. Recommendations are processed for the couple *wine - vintage*.



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

```
CREATE TABLE media(
   type VARCHAR(30),
   label VARCHAR(30),
   webAddress VARCHAR(500),
   picture BLOB,
   video BLOB,
   wineName VARCHAR(100),
   wineArea VARCHAR(100),
   PRIMARY KEY (type, label),
   FOREIGN KEY (wineName, wineArea) REFERENCES wine(name, area)
);
```

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) In order to make food management easy we create, for each product a sort of "label" following this format:

   `Traditional Bolognese Ragù - [La Dispensa di Amerigo:  180.00 g]`

   which includes name, label, weight, unit. Other than the label the query must return **NULL** if the food does not belong to any gift basket, the gift basket name (capitalized) otherwise. [**Tip:** the explicit join could be helpful]

   [*Sol.*]

```
SELECT CONCAT(name, ' - [',label, ': ', weight, ' ', unit, ']') AS label,
       UCASE(basket_name) AS basket
  FROM food LEFT JOIN basketCombines
    ON (food.name = basketCombines.food_name
        AND food.unit = basketCombines.food_unit
        AND food.weight = basketCombines.food_weight)
ORDER by name DESC;
```

2) Based on yearly food sales, food products are classified assigning a sort of mark for appreciation, that is a number of stars (from one to five).

   Write a Query returning for food, **yet available on the online market**, produced by **La Nicchia, Scyavuru, Olis Geraci** - resulting significantly appreciated - the food name, unit, weight, and label, additional data is the sequence '*****', representing **five** stars and the price, and considering this commercial interest this is increased by the 20%. The Query should return also the same data of the remaining food products, yet available on the online market, which are assigned just '**' two stars, price is unchanged.

[*Sol.*]

```
(SELECT name, unit, weight, label, '*****' stars, ROUND(price*1.2, 1) AS price
   FROM food
  WHERE label IN ('La Nicchia', 'Scyavuru', 'Olis Geraci')
    AND endDATE IS NULL)
UNION
(SELECT name, unit, weight, label, '**' stars, price
   FROM food
  WHERE label NOT IN ('La Nicchia', 'Scyavuru', 'Olis Geraci')
    AND endDATE IS NULL)
ORDER BY name;
```

3) We aim to detect main menus that suffer of a poor offer of food products. Return **main menu** which offer less than **seven** food products, additional data is the actual number of food products in this menu and their commercial value, that is the total price of the food products.

[*Sol.*]

```
SELECT main, COUNT(*) AS nr, SUM(price) AS value
  FROM food, menu
 WHERE food.menu_name = menu.name
GROUP BY main
HAVING nr < 7;
```