



# 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)** “Currently the Web shows a variety of data in effective way”. Marketing needs to compare one night stay offers in luxury hotels. Consider the following data of a room availability and describe it by means of the XML language. Data are arranged in a table, hence you could describe it by row or by column (describe a very **small** part of the data).

Prices shown in EUR for 1-night stay						
< July 2022						
Su	Mo	Tu	We	Th	Fr	Sa
					1 €350	2 €320
3 €320	4 €320	5 €320	6 €320	7 €320	8 €320	9 €320
10 €320	11 €400	12 €320	13 €320	14 €320	15 €320	16 €320
17 €320	18 €320	19 €320	20 €320	21 €320	22 €320	23 €320
24 €320	25 €320	26 €320	27 €320	28 €320	29 €320	30 €320
31 €320						

Figure 1: source: <https://grandhotelmajestic.duetorrihotels.com/>

[Sol.:]



```

<availability>
  <ref currency="euro"> 1-night stay </ref>
  <time>
    <year> 2022 </year>
    <month> July </month>
  </time>
  <table>
    <column>
      <day> Monday </day>
      <case>
        <num> 4 </num>
        <price> 320 </price>
      </case>
      <case>
        <num> 11 </num>
        <price> 400 </price>
      </case>
      <case>
        <num> 18 </num>
        <price> 320 </price>
      </case>
      <case>
        <num> 25 </num>
        <price> 320 </price>
      </case>
    </column>
  </table>
</availability>

```

Figure 2: XML file

- (b) **(10 points)** A table `calendar` should be created in order to show data on the web (room, price, and availability). We have to register mandatory: **Room nr.**, **year**, **month**, **dayOfWeek**, **day**, **workingDay**, **price**, and **availability**. Write the SQL statement to create the table, paying attention to data type and to the primary key.

Then write one **INSERT** statement.

[Sol.: ]

```

CREATE TABLE Calendar(
  room_nr INT,
  year CHAR(4),
  month VARCHAR(15),
  dayOfWeek VARCHAR(10),
  day INT,
  workingDay BOOLEAN,
  price NUMERIC(8,2),
  availability BOOLEAN,
  primary key(room_nr, year, month)
);

```

```

INSERT INTO Calendar VALUE(301,'2022','July','Monday',11,TRUE,320.00,FALSE);

```

- (c) **(Optional: 5 points)** Data integration really represents a challenge in data management. ODBC technology can be easily used to integrate some applications and relational databases. What does ODBC technology enable? Describe the experience of data integration we have set up in the lab.

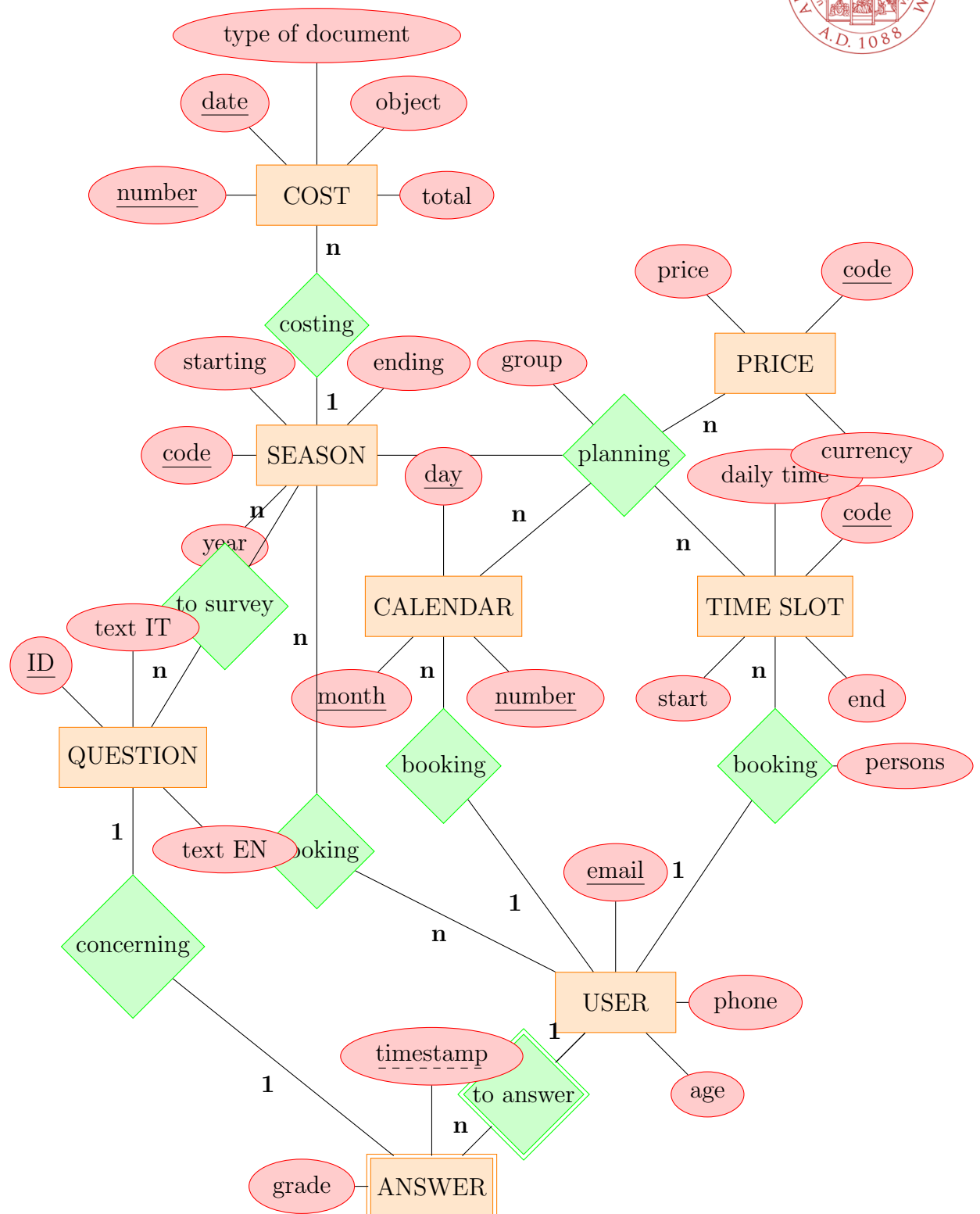
[Sol.: See teaching material]

**Es. 2 - Data Modeling**

- (1) **(35 points)** “Apps are valid and friendly tools to manage services: reservations, costs and user satisfaction”. For the coming summer - and for the next three years - we have in charge to manage a beach volley field, and we evaluate a good solution to realize an **App** that should accept reservations, register costs, incomes and collect user’ satisfaction. We aim to plan different time-slots, each one dedicated to different group of ages (14 - 18: junior, 18 - 60, 60 and over: senior).

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

- (a) A **beach volley season** is identified by a code, the year, starting and ending dates.
- (b) The App should be used to register every sort of **costs** (costs for badges, t-shirts, drinks, local taxes, ...), specifically the type of document, the date, the object, and total paid.
- (c) For each group of ages is planned a **calendar**, the day (Monday, ....., Sunday), the month, and the number identifying the day (1, ....., 31).
- (d) Daily **time slots** are defined, by a code, start and end hours, and daily time (morning, afternoon, night).
- (e) For a season is planned a calendar (dates) and time slots *reserved for each group of ages*.
- (f) Further for this planning (day + time slot) are fixed **prices**, for example for juniors in the morning time slot the price is cheapest! Prices are registered with a code, the price itself and the currency.
- (g) Through the App the **user** books a time slot. A user should provide an email and a telephone number, the name and age.
- (h) After that user selects an available time-slot in the calendar, specifying the number of persons. The payment is made directly at the box office.
- (i) The day after the user is invited - by a message - to fill a survey. For this reason **questions** are registered, with an ID, the text of the question and in two different version Italian and English. Questions could change for different years (season).
- (j) **Answers** consist in a grade: number between 1 and 10, therefore for each user expressed answer we register a time stamp (date + time), and the grade. [*Consider answer as a weak entity set*].



Matriculation:

Last and First Name:



- (2) (Optional: 5 points). Write the SQL statement to CREATE the “relation” that describes the **cost**.

```
CREATE TABLE cost(  
    date DATE,  
    number INT,  
    document VARCHAR(20),  
    object VARCHAR(100),  
    total NUMERIC(6,2),  
    seasonCode CHAR(5),  
    PRIMARY KEY (date, number),  
    FOREIGN KEY (seasonCode) REFERENCES season(code)  
);
```



## PARTE B

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

- (1) Region(name, description)
- (2) Sheet(ID, description, Region.name)
- (3) Producer(name, description)
- (4) Produced(Producer.name, Sheet.ID)
- (5) Ingredient(name, description)
- (6) Made(Ingredient.name, Sheet.ID)
- (7) Menu(name, description, main)
- (8) Food(name, unit, weight, label, price, startDate, endDate, Menu.main\_name, Sheet.ID)
- (9) GiftBasket(name, description)
- (10) BasketCombines(GiftBasket.name, Food.name, Food.unit, Food.weight)
- (11) User(ID, date, time, network\_info)
- (12) Consulted(User.ID, Food.name, Food.unit, Food.weight, time)
- (13) Selected(User.ID, Food.name, Food.unit, Food.weight, time, quantity)

### Questions

- 1) The marketing would like to reconsider number and value of food products to offer as speciality of a region. The request is to extract the region name - excluding the whole **Italy** - for which the online market offers the largest number of products, moreover for that region the cheapest and the expensive product offered. **Use mandatory the explicit JOIN if it is necessary to join tables.**

[Sol.]

```
SELECT region_name, MIN(food.price), MAX(food.price)
  FROM sheet JOIN food ON food.sheet_ID = sheet.ID
 WHERE NOT(region_name = 'Italy')
GROUP BY region_name
HAVING COUNT(ID) >= ALL(SELECT COUNT(ID)
                        FROM sheet
                        WHERE NOT(region_name = 'Italy')
                        GROUP BY region_name);
```



- 2) Marketing needs to know for which food products there not exist descriptive sheets and for those released (published) before the 1<sup>st</sup> June 2020. Return the food product identifier and further compute for how many years this product is in the virtual shop [Tip: to compute years use a suitable expression.]

[Sol.]

```
SELECT name, unit, weight, startDate, year(now()) - year(startDate) AS years
FROM food
WHERE startDate <= '2020-06-01'
AND NOT EXISTS (SELECT *
                 FROM sheet
                 WHERE ID = food.sheet_ID)
```

- 3) Report how many products have been selected (**not the quantity of items!!!**) for a purchase, distinct for **main** menu of the food. Show the main menu name, and the number of selections. For the report consider only currently available food, and order from the highest to lowest returned values. **Use mandatory the explicit JOIN if it is necessary to join tables.**

[Sol.]

```
SELECT menu.main, COUNT(food_name) AS nr
FROM food JOIN selected ON (food.name = selected.food_name
                           AND food.unit = selected.food_unit
                           AND food.weight = selected.food_weight)
      JOIN menu ON (food.menu_name = menu.name)
WHERE food.endDate IS NULL
GROUP BY menu.main
ORDER BY nr DESC;
```