Data Management and Visualization

January 31^{st}, 2020

First name	
Last name	
Student ID	
Exam version	0

NOTE: The solution to the exercises must be reported in the provided sheets. No additional sheets will be considered for the exam assessment. All the provided stapled sheets must be returned.

1. Data Warehouse Design

   A European society managing various companies operating in the cleaning sector wants to design a data warehouse to efficiently analyze their cleaning service data.

   Each cleaning company has a unique name, and it is characterized by the city of its headquarters and the list of its equipment. A cleaning company can have one or more professional equipment (e.g. "sweepers", "steam cleaners", "floor scrubbers", "wet vacuums", "floor polishers", "escalator cleaners"). The equipment list is short, i.e., very limited in size, and all its values are known.

   The system records how long the service took (in hours), how many employees were involved, the date, the type of cleaning service (e.g., "outdoor cleaning", "gardening", "office cleaning", etc.), and the building in which the cleaning service has been performed. The building is identified by a specific unique number, and it is characterized by a name, a type (e.g., flat complex, office, hotel, etc.), and its full address.

   The management wants to analyze the number of hours of performed services, the number of employees involved, the revenues and the costs for the services offered by all cleaning companies.

   The management is interested in performing the analysis for each:

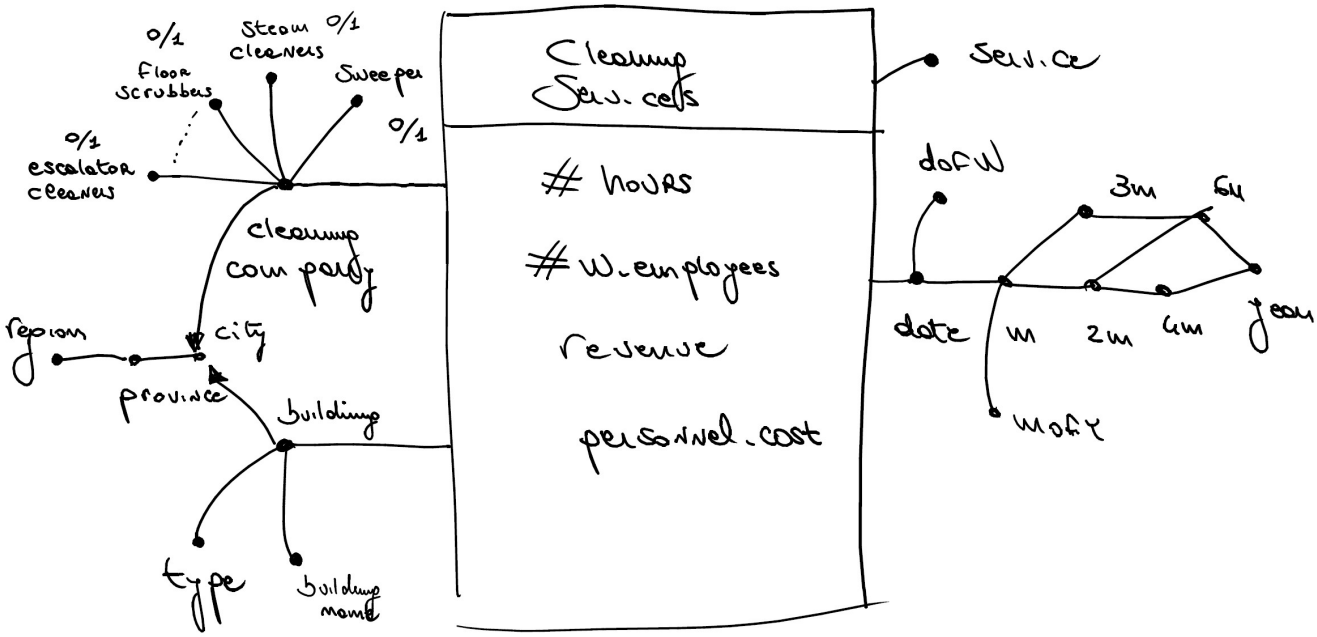
   - cleaning company, which professional equipment the company has, its headquarter city, province, region, and state, and the type of service
   - building, building name, building type, its city, region, province and state
   - date, month, two-month period, three-month period, four-month period, semester, year, day of the week, and month of the year

   Design

   (a) (7 points) Design the data warehouse to address the specifications and to efficiently answer to all the provided frequent queries.

   (b) (8 points) Write the following frequent queries using the extended SQL language.

Draw the conceptual schema of the data warehouse and the logical schema (fact and dimension tables) below.



CleaningServices (<u>CID, TId, BId, SId</u>, numHours, revenue, numEmployee, personnel_cost)
Location (<u>LId</u>, city, province, region)
Time (<u>TId</u>, date, dayOfWeek, month, monthOfYear, 2m, 3m, 4m, 6m, year)
CleaningCompany (<u>CId</u>, cName, C_LId, steam_cleaners, floor_scrubbers, wet_vacuums, floor_ polishers, ..., escalator _cleaners,)
Building (<u>BId</u>, bName, bType, B_LId)
Services(<u>SID</u>, SName)

(a) Separately for each cleaning company, type of service and month, analyze:

i. the average daily revenue

ii. the cumulative revenue since the beginning of the year

iii. the percentage of hours of each service type with respect to the the total number of hours across all service types

```
SELECT serviceType, cName, year, month
    SUM(revenue)/COUNT(DISTINCT date),
    SUM(SUM(revenue)) OVER (PARTITION BY serviceType, cName, year
                            ORDER BY month
                            ROWS UNBOUNDED PRECEDING)
    100*SUM(numHours)/SUM(SUM(numHours)) OVER (PARTITION BY month, cName)

FROM CleaningServices, Time, Service
WHERE ...
GROUP BY serviceType, year, month, cName
```

(b) Considering the cleaning services of 2019, separately for each building and cleaning company, analyze:

i. the profits (i.e., the difference between revenues and costs)

ii. the percentage of revenues with respect to the total revenues of the company for all the buildings in the same city

iii. assign a rank according to the total number of employees (rank 1st the highest)

```
SELECT cName, BID,
    SUM(revenue)-SUM(cost) AS ,
    100*SUM(revenue) / SUM(SUM(revenue)) OVER (PARTITION BY cName, LB.city),
    RANK() OVER ( ORDER BY SUM(#numEmployees) DESC ) as RankState

FROM CleaningServices, Time, Service, Location LB
WHERE ...
AND year=2019
GROUP BY cName, BID, LB.city
```

2. Non-relational Database Design

   Design a document-based NoSQL database for storing the following data of a company.

   - Employees, described by their
     - name,
     - surname,
     - list of education certificates (e.g., "Master Degree", "PhD Degree", etc.),
     - list of addresses, each one consisting of city, postal code, street name, and street number
     - identification number (a unique integer identifying each employee).
   - Projects, described by their
     - name (two different projects cannot have the same name),
     - time period, consisting of start date, end date, duration in days, duration in months,
     - list of topics (e.g., "mobile app", "web service", "machine learning", etc.).

   The number of projects is virtually unbounded, as every day new projects are created. Employees are assigned to many projects, and potentially, the list of historical projects an employee was assigned to can grow indefinitely over time. The company has a limited number of employees, and each project has a limited number of employees assigned.

   Consider the following access pattern to the data.

   - The employee data is presented on a dashboard where the employee identification number, name, surname, list of education certificates, and list of addresses are always displayed.
   - The project data is presented on a dashboard where the project name, start date, end date, duration in days, duration in months, list of topics, and list of assigned employees, with their identification number, name, and surname, are always displayed.

   (10 points) Provide below a relevant sample document for each collection you design to address the described context, e.g., a sample employee document and a sample project document with sample values for all the attributes you expect to store inside each document.

```
Employee document
{ "id": 123456,
  "name": "John",
  "surname": "Doe",
  "education": ["Master Degree", "PhD Degree", ...],
  "addresses":
   [
    {"city": "Torino",
     "postalcode": 10100,
     "streetname": "Via Torino",
     "streetnumber": 10
    },
    {"city": "Milano",
     "postalcode": 20200,
     "streetname": "Via Milano",
     "streetnumber": 20
    },
    ...
   ]
}

Project document
{ "name": "Project007",
  "timeperiod":
    {"startdate": 2020-01-30,
     "enddate":   2020-01-31,
     "days": 2,
     "months": 0
    },
  "topics": ["mobile app", "machine learning", ...],
  "employees":
    [
     {"id": 123456,
      "name": "John",
      "surname": "Doe"},
     {"id": 234567,
      "name": "Anothername",
      "surname": "Anothersurname"
     }
    ]
}
```