Data Management and Visualization

February 14^{th}, 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 society managing gyms wants to design a data warehouse to efficiently analyze their data.

   Each gym is identified by a unique number. The system records the name of the gym and its address. For each gym, the list of the available services is recorded. A gym can offer one or more services (e.g. "indoor swimming pool", "outdoor swimming pool", "spa", "sauna"). The number of services is limited in size and known a priori. It is also supposed that services offered by the gyms do not change over time.

   For each gym, the system records who subscribed and when, and every time the user entered the gym. Each user can choose one of the following types of subscriptions: daily, monthly, annual, or 2-year long. The same user can have multiple non-overlapping subscriptions to the same gym (or to different gyms) over time, possibly of different type. For each user, his/her date of birth, sex, and the city of birth are recorded.

   The system records the revenues deriving from subscriptions. The full price is payed at the beginning of the subscription. The management wants to analyze the number of entrances in the gym, and the average revenue per subscription. The management is interested in performing the analysis for each:

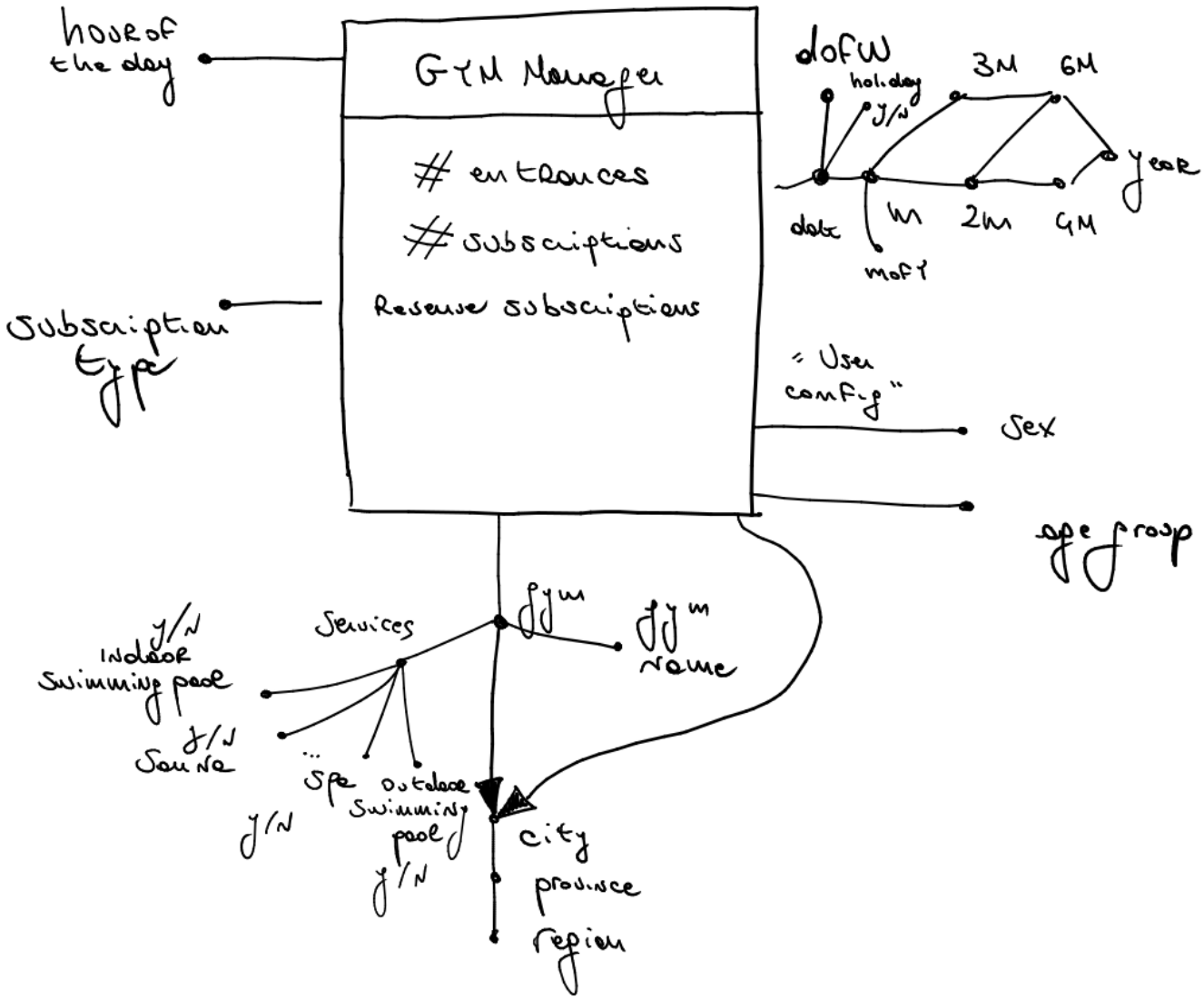
   - Gym, its name, its combination of services, its city, province, and region
   - User age group (in 10-year groups, e.g., 10-19 years, 20-29 years, etc.), user sex, city, province, and region of birth, and subscription type
   - Date, month (e.g., February 2020), month of the year (e.g., February), two-month period, three-month period, four-month period, semester, year, day of the week, holiday, and hour of the day (e.g., 10-11am)

   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 report the logical schema (fact and dimension tables) below.



GymManager (UserLid, Tid, TSid, Gid, UCid, Sid, numEntrances, numSubscriptions, totRevenues)
Location (Lid, city, province, region)
Time (Tid, date, dayOfWeek, holiday, month, MoY, 2m, 3m, 4m, 6m, year)
Hour (TSid, hourOfTheDay)
Gym (Gid, gymName, Lid, indoor_swimming_pool, outdoor_swimming_pool, spa, ..., sauna)
UserConfig (UCid, ageGroup, sex)
Subscription (Sid, subscriptionType)

(a) Considering only the gyms with outdoor or indoor swimming pools, separately for each year, subscription type, and city of the gym, analyze:

i. the percentage of subscriptions with respect to the total number of subscriptions for all the gyms of the same region

ii. the cumulative yearly revenues

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

```
SELECT year, subscriptionType, LG.city, LG.region,
    100*SUM(numSubscriptions) / SUM(SUM(numSubscriptions)) OVER
            (PARTITION BY year, subscriptionType, LG.region)
            AS percentageSubscriptions,
    SUM(SUM(totRevenues)) OVER
            (PARTITION BY subscriptionType, LG.city
             ORDER BY year ROWS UNBOUNDED PRECEDING))
            AS cumulativeRevenues,
    RANK() OVER (ORDER BY SUM(numSubscriptions) DESC) as RankSubs
FROM GymManager GM, Membership M, Gym G, Location LG
WHERE (joins)
AND (indoor_swimming_pool=True or outdoor_swimming_pool=True)
GROUP BY year, subscriptionType, LG.city, LG.region
```

(b) Separately for each age group and city of birth of the users, analyze:

i. the hourly average number of entrances

ii. the percentage of revenues with respect to the total revenues across all cities of birth

iii. the average revenue per subscription

iv. the average revenue per entrance

```
SELECT ageGroup, UL.city,
    SUM(numEntrances)/COUNT(DISTINCT date, hourOfTheDay),
    100*SUM(totRevenues)/SUM(SUM(totRevenues))
            OVER (PARTITION BY ageGroup),
    SUM(totRevenues)/SUM(numSubscriptions),
    SUM(totRevenues)/SUM(numEntrances)
FROM GymManager, UserInfo, Time, Location UL
WHERE (joins)
GROUP BY ageGroup, UL.city
```

2. Non-relational Database Design

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

   - Gym courses, described by
     - course name (unique),
     - list of course categories, each consisting of a name and an associated colour,
     - course instructors,
     - year,
     - weekly schedule of the lessons, consisting of weekday, start time, and end time for each lesson.
   - Instructors, described by their
     - name and surname,
     - date of birth
     - list of certifications, each consisting of a name and the date of achievement
     - list of the courses.

   Most courses have only one instructor, and no course has a large number of instructors. The number of courses can grow indefinitely, as new courses are created every year. The number of instructors is limited, however, the number of courses an instructor has been teaching can grow indefinitely. The colour of the course category must be saved as a sequence of three numbers in the range 0-255 (RGB, Red Green Blue standard, e.g., white colour is 255, 255, 255).

   Consider the following access pattern to the data.

   - The gym courses of the current year are presented on a website where their name, categories (each coloured accordingly), name and surname of the instructors, and the weekly schedule are to be displayed at every page view.
   - The instructor information is presented on an internal administration software, where the instructor name, surname, date of birth, and list of certifications are presented for each instructor, together with the number of courses taught, separately for each year (e.g., instructor "A": year 2019, 23 courses; year 2018, 45 courses; etc.).

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

```
Course document
{ "name": "Fitness Course ABC",
  "categories": [
    { "Fitness", [255,255,255] },
    { "Health", [0,0,0] },
  ],
  "instructors": [
    { "name": "John", "surname": "Doe" },
    { "name": "Jane", "surname": "Gig" },
   ],
  "year": 2019,
  "schedule": [
    { "weekday": "Monday",
      "from": 10
      "to": 11},
      ...
  ]
}

Instructor document
{ id: "ABC123",
  "name": "John",
  "surname": "Doe",
  "dateofbirth": "1990-01-31",
  "certifications":
    {"name": "Personal trainer level 1",
     "date": 2020-01-31,
    },
  "courses":
    [
     {"year": 2019,
      "count": 23
     },
     {"year": 2018,
      "count": 45
     }
    ]
}
```