University of Ottawa
School of Electrical Engineering and Computer Science
CSI4142 Introduction to Data Science

Assignment 1 2018

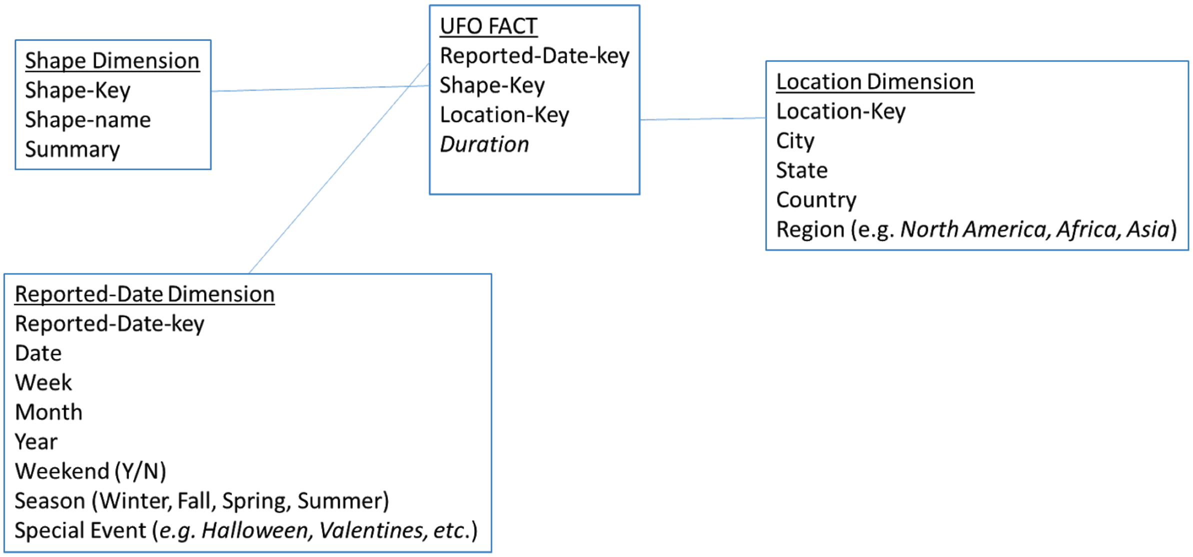
Total marks: 50

This is an individual assignment. Submit your answer using the Virtual Campus.

The National UFO reporting center (http://www.nuforc.org/index.html) maintains information about unidentified flying objects (UFOs) as reported since 1400. The database was first created in 1998 and may be accessed at this link: http://www.nuforc.org/webreports.html

The data include information such as the reported shape(s), the date of observation and reporting, the duration, the location and a short textual description of the UFOs, amongst others. The CEO of the National UFO reporting center asks you to build a data analytics application in order to explore this data.

You are given the task to create a data mart for the National UFO reporting data, in order to convert the data into a format that facilitates Online Analytical Processing (OLAP) queries. The following is an initial dimensional model that is submitted for your review.



Shape Dimension
Shape-Key
Shape-name
Summary

UFO FACT
Reported-Date-key
Shape-Key
Location-Key
Duration

Location Dimension
Location-Key
City
State
Country
Region (e.g. *North America, Africa, Asia*)

Reported-Date Dimension
Reported-Date-key
Date
Week
Month
Year
Weekend (Y/N)
Season (Winter, Fall, Spring, Summer)
Special Event (*e.g. Halloween, Valentines, etc.*)

Answer all the following questions.

1. Declare the grain of this dimensional model. (2)

The grain of this data mart is a single report of UFO activity, as per location and the date it was included in the data mart. (Note, as discussion in class, that the "summary" text box makes this a bit of an awkward design, since all summaries will be appended. See (2) below.

2. There is a dimension missing from this model. Provide this dimension and motivate your answer. (4)

Either <u>one</u> of the following two answers are correct.

   a. **The dimensional model does not include the date of the sighting. This makes it very difficult to navigate. One should add the sighting data as an additional dimension. Note that this is an example of a role-playing dimension, where one who typically have one "Data" dimension that is linked via two foreign keys in the Fact table. (This is similar to the co-pilot and pilot dimensions as discussion in class.)**
   b. **One could add a "Description" dimension, that contains the summary (removed from the Shape), together with some keywords. Such a dimension will facilitate analytics based on keywords such as "Dark", "Clear", Rain", "Halloween", etc.**

3. Define what a measure (or fact) is and suggest one more measure that may be added to this design. (2)

A measure (also known as a fact) is an additive, often numeric, attribute that is used for aggregate operations such as Sum(), Count(), Average(), Min() and Max(). It is added in the Fact table.

- **In this data mart, one may want to add a measure TIME, that indicates the exact time of the observation.**
- **Alternatively, the TIME may be converted into hourly BANDS, e.g. 00:00-00:59, 01:00-01:59, and so on.**

(You were asked for only <u>one answer</u>.)

4. Explain what a concept hierarchy is and provide an example of a concept hierarchy in the UFO data mart. (2)

A concept hierarchy defines a relationship between different attributes within a dimension, where the different levels of the hierarchy correspond to different levels of details or aggregated information. Concept hierarchies are typically used in order to roll up, or drill down, levels of details.

In this case, two concept hierarchies are Country → State → City or Year → Month → Day.

(You were asked for only <u>one answer</u>.)

5. Aggregation is one way that may be used to speed up OLAP queries against a data mart. Give an example of an aggregate that you would potentially create against the UFO dimensional model. (4)

The potential aggregates are associated with the concept hierarchies. For example (You were asked for only one answer):

Aggregate by month
Aggregate by year
Aggregate by state
Aggregate by country
Aggregate by month and state
Aggregate by year and state
Aggregate by month and country
Aggregate by year and country

6. The data contain some missing values. Give an example of an attribute with missing values and explain how you would handle such data during the analytical process. (2)

Shape contains some empty values. An approach could be to look at the summary and then extract the shape from there.

Duration is the most problematic. Contacting the people who reported the sighting would be time consuming and most probably unrealistic. The best solution would be, when running queries that e.g. report average durations of a particular shape (or in a specific state), to add a note that contains the % of the sighting did not report the duration.

7. Draw a scatter plot in order to determine whether there is a correlation between the time of occurrence and the duration. (4)

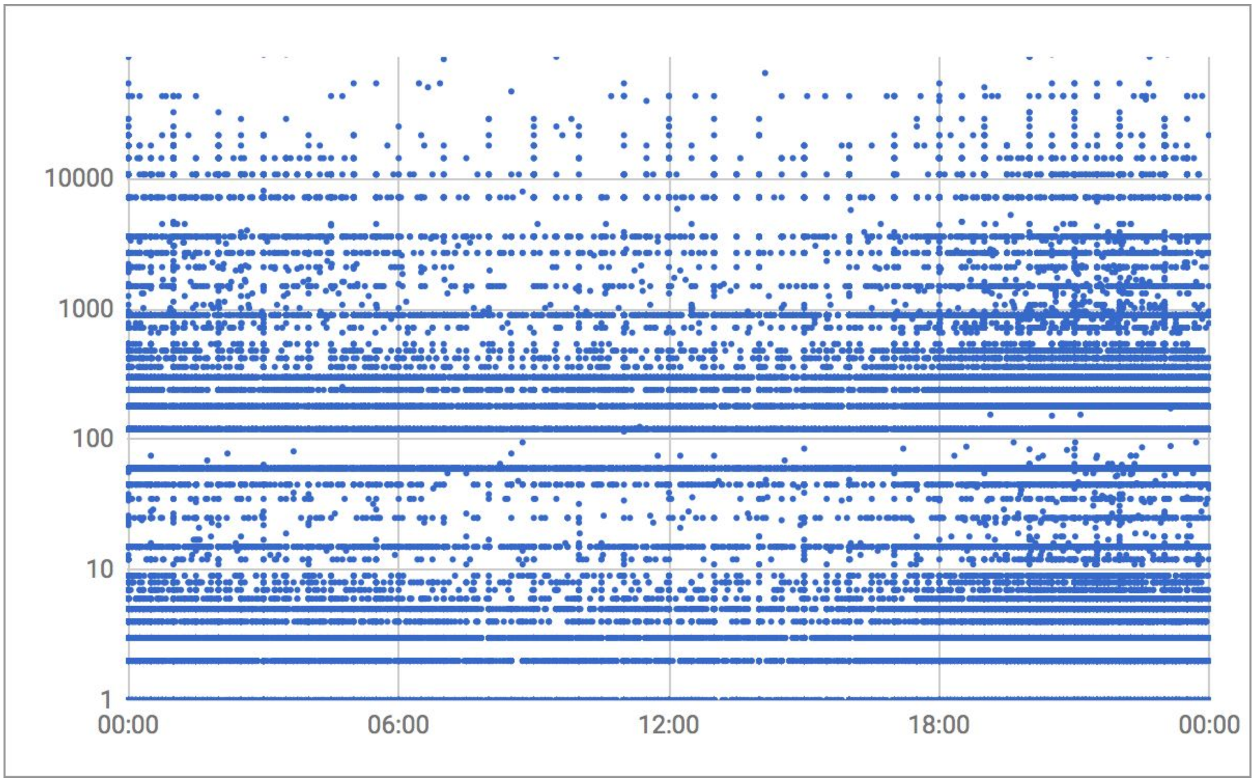
The first step would be to convert the duration into the same unit. In this case, the smallest units are seconds.

The data contains a large number of rows with durations of 0. This seems impossible and counterintuitive. An explanation could be that these durations were not recorded, and that the data engineer set the default value to 0.

There are also a number of outliers, which could be removed from the dataset prior to plotting.

In short, there is no correlation between duration and time of day. This thus goes against popular belief, where one would expect to see UFOs at night.

(This gives a correlation coefficient of -0.0105, indicating no linear correlation.)



8. Consider a query that returns the names of the States that reported the eight viewings of DELTA-based shapes. Show how you would use bitmap indexes to answer this query.
   (2)

In this question, the best approach would be to construct an aggregate (or cube) based on Shape and State.

In this case, the data in the Fact table may be converted into bitmaps that shows the occurrence (or not) of the Shapes and States. The bitwise AND for the DELTA shape and STATE keys would yield the correct result.

	California	**Florida**	**Alberta**	**Ontario**
Shape: Delta	1	0	0	1
Shape: Round	0	0	0	1
Shape: Sphere	0	**1**	**0**	0

9. Consider a query that returns the total number of sightings of DIAMOND shapes in Florida, in 2017. Show how you would use semi-joins and bitmaps to optimize this query. (4)

For the Shape dimension: Obtain the Fact table <record identifiers> where DIAMOND shape-key occurs, and use a Semi-join to link them to the Shape dimension. This will return the FACT table <record identifiers>.

For Florida, obtain the Fact table <record identifiers> where the Location-keys occurs.

For 2017, obtain the Fact table <record identifiers> for this year.

10. Create the UFO data mart in PostgreSQL and insert all the data into your tables. (You will notice that some of the durations are uncertain. In this case, you should record the minimum value. That is, a duration of 10-15 seconds should be converted to 10 seconds.) (6)

For the exercise students, were encouraged to use scripts to insert their data. Note that it is important to include all attributes in the schema, including details such as weekend(y/n), season, and so on. Also, the conversion of the dates were a bit tricky.

Specific things to look out for:

- **logic to normalize the CSV into dimensions (and uniqueness in the dimension tables)**
- **(unless students use the latest CSV) reported sighting dates from the National UFO reporting centre database range from 1400 to 2018, but the scraped data lists dates using only two digits for the year. students will need to overcome this by changing the year for anything after xx/xx/18 to the previous century**
    1. **example result xx/xx/2045 changed to xx/xx/1945**
- **suggested methodology to handle duration (durations converted to single unit):**
    1. **using regular expressions (try to match # seconds, if fail then try to match # minutes, then # hours, then #)**
    2. **convert the matched number into seconds**
    3. **use 1 second for missing data**

11. Provide the SQL statement to list the number of sightings by Month. (2)

```
SELECT month, SUM(1)
FROM ufo_facts
    INNER JOIN reported_dates
        ON reported_date_id = reported_dates.id
GROUP BY month
```

12. Provide the SQL statement to list the number of sightings by Month and State. (4)

```
SELECT month, state,  SUM(1)
FROM ufo_facts
    INNER JOIN reported_dates
        ON reported_date_id = reported_dates.id
    INNER JOIN locations
        ON location_id = locations.id
GROUP BY month, state
```

13.

Provide the SQL statement to list the names and average durations of the five shapes that have the most sightings, overall. (4)

```
SELECT name, AVG(duration)
FROM ufo_facts
      INNER JOIN shapes
             ON shape_id = shapes.id
GROUP BY name
ORDER BY COUNT(1) DESC
LIMIT 5
```

14. Provide the SQL statements to list, for each shape, the State with the longest recorded duration of sighting, together with the average duration for each shape. For example, for the CRESCENT shape, the average duration was 47.5 seconds, the longest duration was 90 seconds and this occurred in WI (Wisconsin). (4)

- Using inner joins and group by (runs in 165ms)

```
SELECT DISTINCT name AS shape, avg, max, state
FROM ufo_facts
      INNER JOIN (SELECT shape_id, AVG(duration), MAX(duration) FROM ufo_facts
GROUP BY shape_id) AS t
             ON ufo_facts.shape_id = t.shape_id AND duration = t.max
      INNER JOIN locations
             ON location_id = locations.id
      INNER JOIN shapes
             ON t.shape_id = shapes.id
```

- Using the Window() function (runs in 1.7s)

```
SELECT shape, avg, max, state
FROM (  SELECT DISTINCT name as shape, AVG(duration) OVER w, state,
MAX(duration) OVER w, RANK() OVER w
      FROM ufo_facts
             INNER JOIN locations ON location_id = locations.id
             INNER JOIN shapes ON shape_id = shapes.id
      WINDOW w AS (PARTITION BY shape_id ORDER BY duration DESC) ) as t
WHERE rank < 2
```

15. Provide the SQL statement to list the total the number of sightings over the weekends, for every different shape, as recorded in California versus Florida.(4)

- Using inner joins and group by

```
SELECT state, name AS "shape", COUNT(*) AS "sightings on weekends"
FROM ufo_facts
      INNER JOIN reported_dates
            ON reported_date_id = reported_dates.id
      INNER JOIN locations
            ON location_id = locations.id
      INNER JOIN shapes
            ON shape_id = shapes.id
WHERE weekend = true
      AND state IN ('CA', 'FL')
GROUP BY state, name
ORDER BY shape, state
```

- Using the Window () function

```
SELECT distinct state, name AS "shape", COUNT(*) OVER (PARTITION BY
locations.state, name) AS "sightings on weekends"
FROM ufo_facts
      INNER JOIN reported_dates
            ON reported_date_id = reported_dates.id
      INNER JOIN locations
            ON location_id = locations.id
      INNER JOIN shapes
            ON shape_id = shapes.id
WHERE weekend = true
      AND state IN ('CA', 'FL')
ORDER BY shape, state
```