Tutorial 2

for

Data Mining - Transformation Version

1.0

Adam Ryan (14395076)

COMP47530

October 7, 2021

Contents

1 Question 1

1.1 Questions and answers

Consider a dataset D that consists of customers purchasing tour packages to various places at different prices. We can perform different kinds of data operations on the dataset D. Try to categorise the following operations into a) simple query of data retrieval, b) Online Analytical Processing, or c) Data Mining.

1. Find names of customers who have purchased tours that cost less than €500.
   - Simple Query of Data Retrieval. This is a standard query on a database selecting the distinct customer names filtered by tour price.
2. List the names of the customers, the number of tour packages that the customers have purchased, and the total cost for the tours.
   a) OLAP. This requires rolling up tour-level data to customer level data, summing the tour costs and counting the distinct number of tour IDs for the customer.
3. Calculate the difference in quarterly sales of tours between this year and the previous two years.
   - OLAP. This requires rolling up tour sales by quarter by year (and may, depending on your approach, involve pivoting the data).
4. Find a rule such as "IF customers purchase a tour package to France, THEN it is 80% likely that the same customers also purchase a tour package to Spain.
   - Data Mining. The identification of an appropriate rule will involve the full data mining process as you are developing some form of rule-based model.
5. From the customer purchase history, build a model for predicting the kinds of customer who are likely to purchase tours to a certain country.
   - Data Mining. The development of the model may be classified as data mining if the model is appropriately sophisticated to build out customer clusters and appropriately analyse the purchase history (of course, simplified models could be developed relying solely on OLAP such as a simple frequent travel and total payment segmentation model, but a 'proper' completion of the process will be data mining).

2 Question 2

2.1 Questions and Answers

Consider the dataset given in DW_Dataset.csv, representing the employee data of a company.

1. If the attribute "Salary" needs to be discretised into three pay bands, suggest a simple yet sensible solution for the discretisation based with a valid argument.
    - A simple way of banding the saleries based on the information provided would be categorise them as Low corresponding with under €2800, Medium corresponding with €2800 to €5600, and High corresponding with over 5600. The logic of this split would be based on splitting the total salary range in the data present into three groups. These groups appropriately capture Juniors into the lower tier (where we see there's a low standard deviation in salary between people in the same position), seniors into the medium range, and directors or executives into the high range. Using qcut or cut on the data results in ranges which fail to appropriately capture the key differentiator in salary (position). At lower levels, we see the salary is more concentrated with a lower deviation, while at higher levels (director) there is a larger salary deviation which suggests that the 'low' level should encompass a tighter band than upper bands, however we have limited information at the senior level to identify how varied salaries may be. As such, using the categorisation suggested keeps the low bracket small (as in reality the range is not 0 to 2800 but minimum wage to 2800), the medium range larger, and the high range unbounded.

2. Miss Davis's salary is unknown, and the unknown value needs to be imputed, what is a sensible replacement value and why?
    - I would replace this with 3300 as we see she is a senior technician and based on the junior technician and director salaries we see that salaries broadly correlate to your level of experience.

3. Among the employee records, which record can be considered as an outlier? What harm can an outlier cause to the understanding of the dataset?
    - I would classify Jones is an outlier in the dataset. There is a positive correlation between Age and Salary (explained via seniority) within the dataset, yet despite being 44 years old Jones is a Technician on a salary of 1800. This is a significant divergence from the other datapoints (the one point which might

be considered an exception is Moore who, at 25, is a senior technician, however this is not as significant a divergence as Jones). We see this as an outlier when we plot the values:

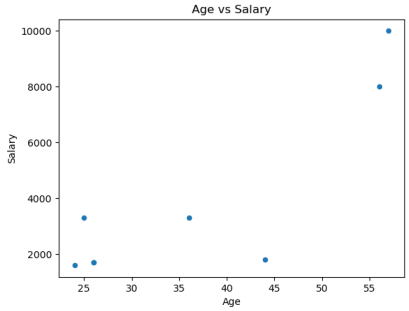

Figure 2.1: Age vs Salary

Online Analytical Processing (OLAP) perceives a dataset in a multi-dimensional space. For the dataset given in DW_Dataset.csv, perform the following tasks/operations:

4. Draw a diagram of a 3D view using the following attributes: Year of Birth, Status, and Salary.
   - I have created a 3D view using matplotlib, graphing Year of Birth, Status, and Salary. Please refer to figure 3.1.

5. What do the data points inside the cube represent? Use the cube as an example to discuss the meaning of OLAP operations such as pivoting, slicing and dicing, rolling up and drilling down.
   - Each point represents a 3-dimensional vector such that the first element is an element of the set of values of Status, the second element is an element of the sest of values of Year of Birth, and the third element is an element of the set of values of Salary.
     - Pivoting corresponds with rotating the cube for an alternative view of the data.
     - Slicing and dicing corresponds with filtering dimensions of the cube. I.e. if one only wished for the 'Status' attribute to correspond with 'Technician', Slicing would involve taking the plane of the cube which corresponds with this value, while dicing involves cutting the cube along multiple planes to extract a subcube, for example the sub-cube where 'Status'=Technician and 'Status' = 'Senior Technician'.
     - Rolling up involves aggregating the data over a dimension e.g. finding the total salary by status.

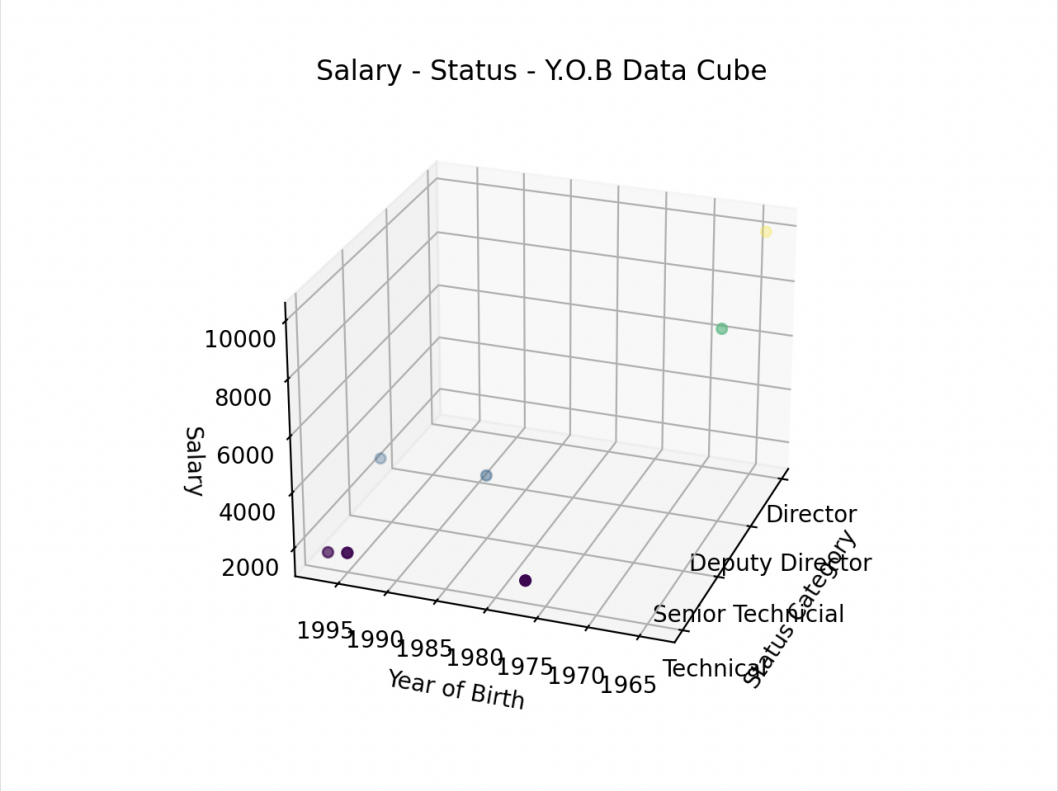

Figure 2.2: 3D View of Data

– Drilling down involves subdividing the data into more granularity e.g. if we had Date of Birth instead of Year of birth, drilling down into this column might involve subdividing Date of Birth a more granular combination of Year, quarter, Month, or Day of Birth or the day number of date of birth since 1900.

3 Question 3

3.1 Questions and Answers

Suppose that a data warehouse for Big University consists of the four dimensions student, course, semester, and instructor, and two measures count and avg_grade. At the lowest conceptual level (e.g., for a given student, course, semester, and instructor combination), the avg_grade measure stores the actual course grade of the student. At higher conceptual levels, avg_grade stores the average grade for the given combination.

1. Draw a snowflake schema diagram for the data warehouse.
    - I've created a high-level schema for big university focusing on four required dimensions, with a few optional dimension tables. In particular, to a student I've added on address (which itself has county and country references) to store the student's term address, registration status (as it's referred to in the question below), the degree major and the degree. To an instructor, I've added on their position (Assistant Lecturer, Lecturer, Associate Professor, etc.). To the semester I've added on the year of the semester.

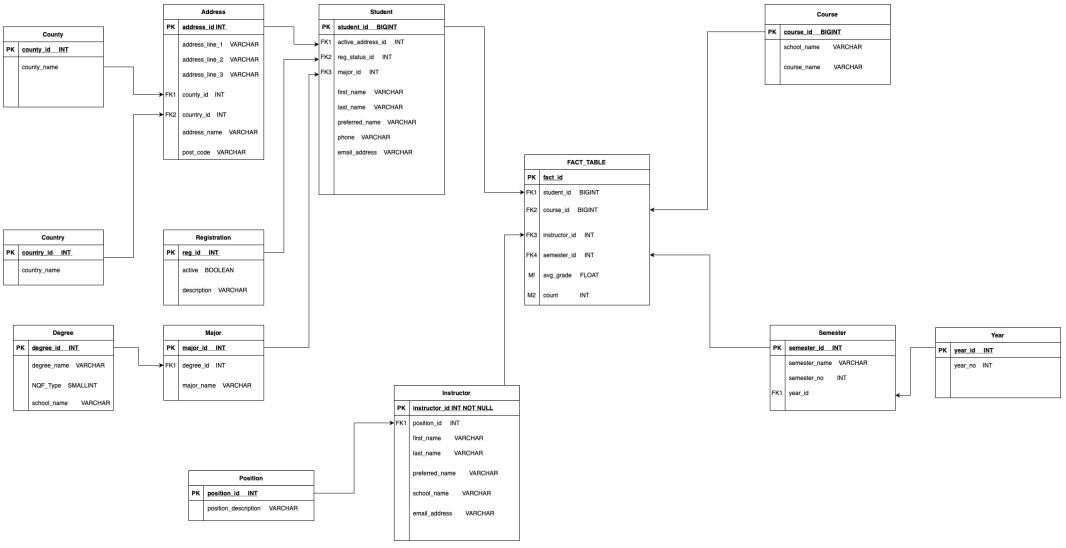

Figure 3.1: BigUniversity Schema

2. Starting with the base cuboid [student, course, semester, instructor], what specific

OLAP operations (e.g., roll-up from semester to year) should you perform to list the average grade of CS courses for each Big University student.

- Note, I am assuming it is looking for the student's average grade across ALL computer science courses. I.e. That it is not asking about course PER student. The specific OLAP operations are:
  - Roll up course to school name of CS.
  - Dice where course's school name is CS
  - Drill across to student first name and last name.
  - Take the average grade grouped by student.
- Specifically, you're taking the avg grade grouped by student from the table where the courses are inner joined on the course table where school name is CS.
- An MSSQL query to accomplish this would be:

```
1                   SELECT
2                           s.student_id
3                           ,s.first_name
4                           ,s.last_name
5                           ,AVG(ft.avg_grade)                          AS
6                   FROM
7                           (SELECT
8                                   course_id
9                           FROM
10                                  course
11                          WHERE
12                                  school_name='CS') cs
13
14                  INNER JOIN
15                          fact_table ft
16                  ON
17                          cs.course_id=ft.course_id
18
19                  INNER JOIN
20                          (SELECT
21                                  st.student_id
22                                  ,st.first_name
23                                  ,st.last_name
24                          FROM
25                                  student st
26                          ) s
27                  ON
28                          s.student_id=ft.student_id
```

```
29
30                      GROUP BY
31                              s.student_id
32                              ,s.first_name
33                              ,s.last_name
```

- If instead you are looking for the average grade per student per course where course is CS, then you do the exact same as above except you group by course_id and drill across to course name as well.
- I am assuming university is BigUniversity for all students and courses as this is stated in the question.

3. If each dimension has five levels (including all), such as "student < major < status < university < all", how many cuboids will this cube contain (including the base and apex cuboids)?
   - We have 4 dimensions and each dimension has 5 levels, therefore we have $5^4 = 625$ cuboids.

The following questions will guide you to implement a data warehouse using PostgreSQL. At this stage, we assume that you have already defined the DW schema, with subjects, dimensions, and measures.

4. First, you need to establish a connection with the database to create a table where you can store records and arrays of data. Make sure you follow the PostgreSQL naming convention.
   - Done. I've named the table input_dw_data with columns: [id, first_name, middle_name, last_name, favourite_number, location]
5. Second, you need to create another connection to the DW where you will store your datasets in.
   - Done.
6. Create a data file that contains 5 entries, called "input_DW_data.csv", which you need to store in your DW.
   - Done
7. Finally, you need to define the following functions to read, write, update, and list your data to / from the data warehouse.

   def read_record (field, name, engine): ...
   def write_record (name, details, engine): ...
   def update_record (field name, new value, engine): ...
   def read_dataset (name, engine): ...
   def write_dataset (name, dataset, engine): ...
   def list_datasets (engine): ...

- Note, I have created functions which accomplish this however, the functions which are here are poorly defined; how general are the functions expected to be, which tables should they operate on, what columns should they work with, what are the meanings of the inputs, etc. As such, in defining these functions I have kept them highly general, however to do this I have had to alter what inputs are present. For example, my functions will allow a user to read and write to any table in any database, update any column against any record in any table, etc. and protects against sql injection using psycopg2 and it's union functions to allow for a Dynamic SQL alternative. However, because the question has not defined the database requirements this may require modification for a demonstrator to run. As such, I have copied a full output of what this function looks like below because I cannot guarantee it will work for the demonstrator without proper configuration which was not provided.

```
        Write Dataset:

        Read Dataset:

        id first_name middle_name last_name  favourite_number
location
        0   1       Adam      Patrick       Ryan
7          Ireland
        1   2        Joe       Jemma      Smith
4          Ireland
        2   3      Peter        Paris   Petigrew
7  United Kingdom
        3   4        Fav            u      Orite
8            Spain
        4   5       Jill        Gilly   Gilligan
1          Ireland
        Insert:

        id first_name middle_name last_name  favourite_number
location
        0   1       Adam      Patrick       Ryan
7          Ireland
        1   2        Joe       Jemma      Smith
4          Ireland
        2   3      Peter        Paris   Petigrew
7  United Kingdom
        3   4        Fav            u      Orite
8            Spain
        4   5       Jill        Gilly   Gilligan
```

```
1          Ireland
        5   6        New       Sample       Data
6          Ireland
        Reading Record:


        id:6


        first_name:New


        middle_name:Sample


        last_name:Data


        favourite_number:6


        location:Ireland

        Updating Record:

        Read Dataset After update:

        id first_name middle_name last_name  favourite_number
location
        0   1        Adam      Patrick       Ryan
7          Ireland
        1   2         Joe       Jemma      Smith
4          Ireland
        2   3       Peter       Paris   Petigrew
7  United Kingdom
        3   4         Fav           u      Orite
8             Spain
        4   5        Jill       Gilly   Gilligan
1          Ireland
        5   6        New       Sample       Data
6  UpdateLocation

        Out[92]:
```

```
        {'input_dw_data':     id first_name middle_name last_name
favourite_number         location
                    0   1       Adam      Patrick       Ryan
7           Ireland
                    1   2        Joe        Jemma       Smith
4           Ireland
                    2   3      Peter        Paris    Petigrew
7    United Kingdom
                    3   4        Fav            u       Orite
8             Spain
                    4   5       Jill        Gilly    Gilligan
1           Ireland
                    5   6        New       Sample        Data
6    UpdateLocation}
```