Adam Tegart – CA270 – Data Warehouse Project

In order to give my project some background I decided it would be best to briefly explain the overall idea and plan so the rest will be easy to follow.

My chosen domain

I have decided to create a data warehouse for a group of healthcare practices that offer both GP visits and a pharmacy. Below a data warehouse bus matrix can be seen for such a group.

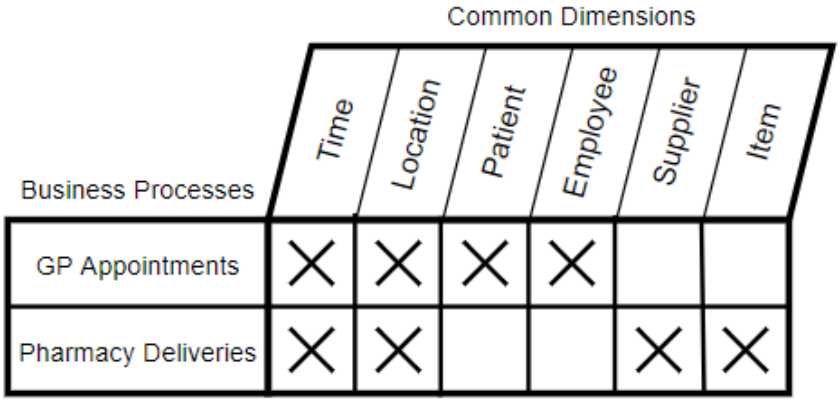


Fig 0. Data Warehouse Bus Matrix

It can be seen that the GP Appointments fact table/data mart will consist of time, location, patient and employee (only those that are associated with a patient as their GP). Pharmacy Deliveries will have supplier and item as opposed to patient and employee.

My problems

I plan to look at average age of the patients having a GP visit by time, by location and by specific GP. It is not possible to have the average value present at all times as it is not additive, i.e. you cannot add 2 averages to get the average of the 2 sets (unless they are averages of the same number of values). To solve this I will instead get the sum of all ages and then divide by the count of rows that have been grouped. I don't mind counting a persons age twice as I just want to know the average for every visit. This will help the practices understand what age group (probably older) frequently visit and which groups should visit more (young), also might find areas with a smaller average, i.e. younger people do go often.

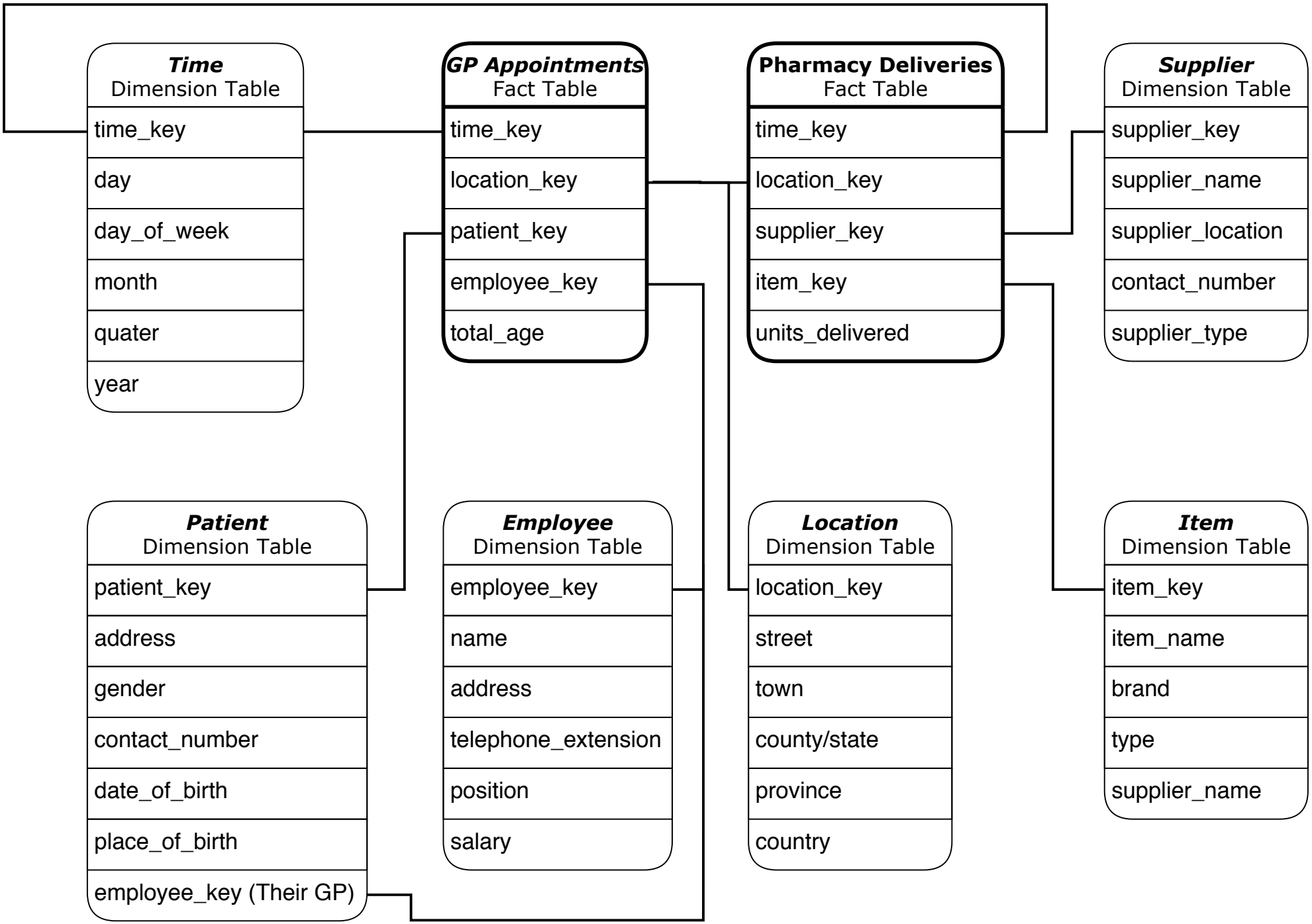
Our overall problem will be, "I want to know the average age of patients for each GP in Dublin and Cork, for all quarters of 2019 split by male and female patients."

For pharmacy deliveries I will look at the how many items I receive of certain types, by time of year, by location, by supplier. This will help to find trends in what medicines/equipment are typically needed in what places and when along with what supplier supplies these.

Our overall problem will be, "I want to know the total amount of items by type that were received each month, in Dublin and Galway from Medi-tech and Healthpro in 2018".

This concludes my brief overview.

Section 1
Constellation Schema for Healthcare Practices

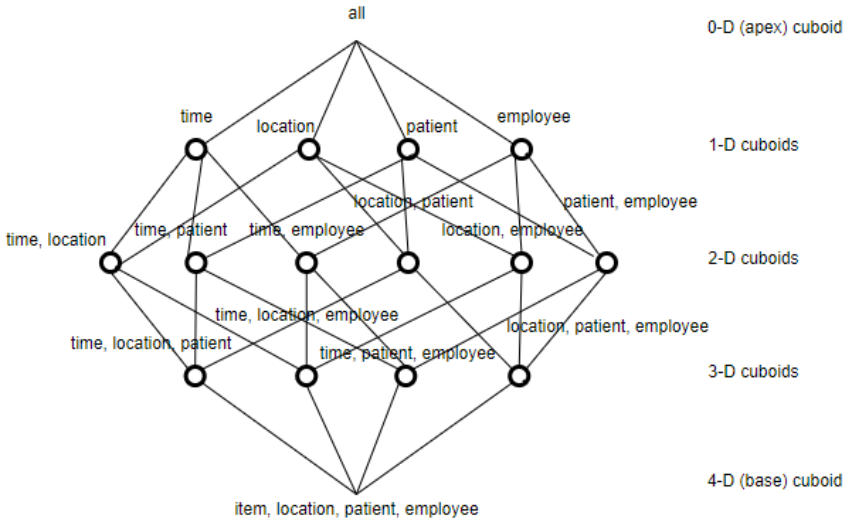


Data Warehouse Schema

Lattice of Cuboids

To be able to properly visualize all the possible combinations of dimensions in the data cubes created I made a lattice of cuboids to represent all these combinations. The snippet of this can be seen below.

GP Appointments



Pharmacy Deliveries

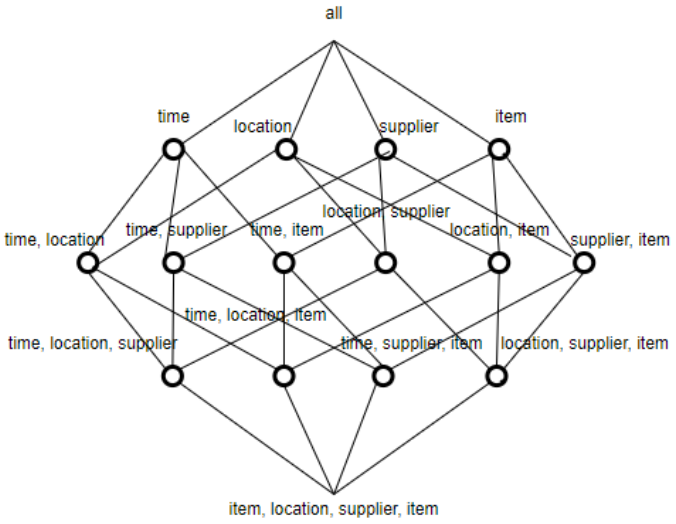


Fig 1. Lattice of Cuboids for the concept Data Warehouse

As a lattice of cuboids is said to form a data cube, this is a visualization of our 2 data cubes, one for GP Appointment and one for Pharmacy Deliveries.

Section 2

Query spec – first query

The query for my first problem will involve joining the time, location, patient and employee dimensions onto the GP Appointments fact table on the keys shows in the Data Warehouse Constellation Schema. Once I have the whole table joined I want to look at the sum of the ages and the count of the rows as I group the data.

In SQL as long as I have the most granular form of the data (each appointment and so a patients age), I can use the avg() function. Although, with a warehouse these groupings are manipulated with roll-ups and drill downs and so we will lose track of the set of numbers the average represents. It is for this reason I decided to keep the attributes separate and compute the average when needed.

The problem that the query will solve is "I want to know the average age of patients for each GP in Dublin and Cork, for all quarters of 2019 split by male and female patients". The data mart would look something like the following after rolling up on time to get quarters and on employees to get GP's (only staff that are GP's are used as they will only be associated with an appointment if they are a patients GP). Then, slicing the locations to have only Dublin and Cork, lastly, a drill down is carried out on the patients to split them by gender.

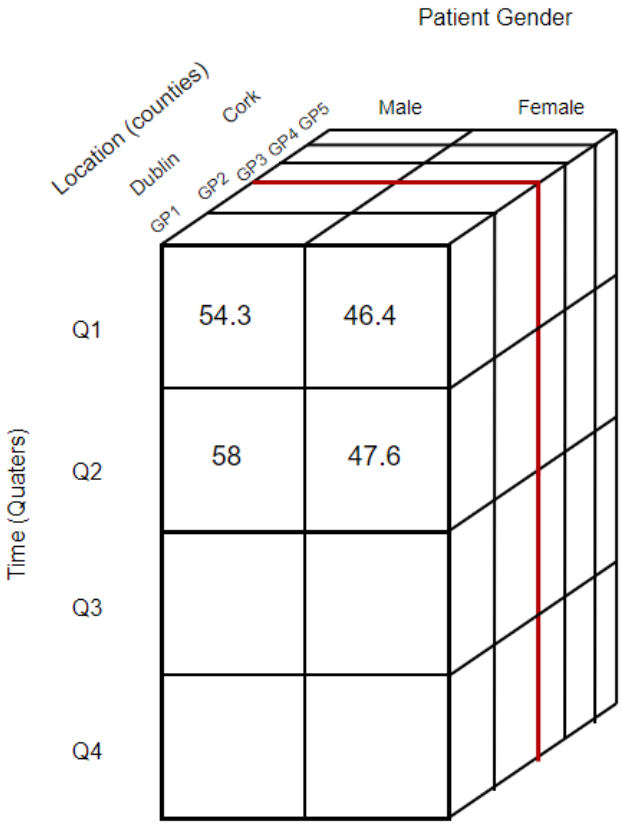


FIG 2.1. Slice of Data Cube corresponding to problem 1

The red line shows the split between Dublin and Cork, within that there is then a split again for every GP in the Dublin and Cork practice. In this way, the average age can be looked at for each GP, in each location at any given quarter in the year for both male and female patients(values are meaningless).

Query spec – second query

My second query relates to the Pharmacy Deliveries, it is will solve the problem "I want to know the total amount of items by type that were received each month, in Dublin and Galway from Medi-tech and Healthpro in 2018". It will be gotten by rolling up the times into months, slicing on location to get only Dublin and Galway and on supplier to get only Medi-tech and Healthpro and drilling down on items to show each type, as shown below.

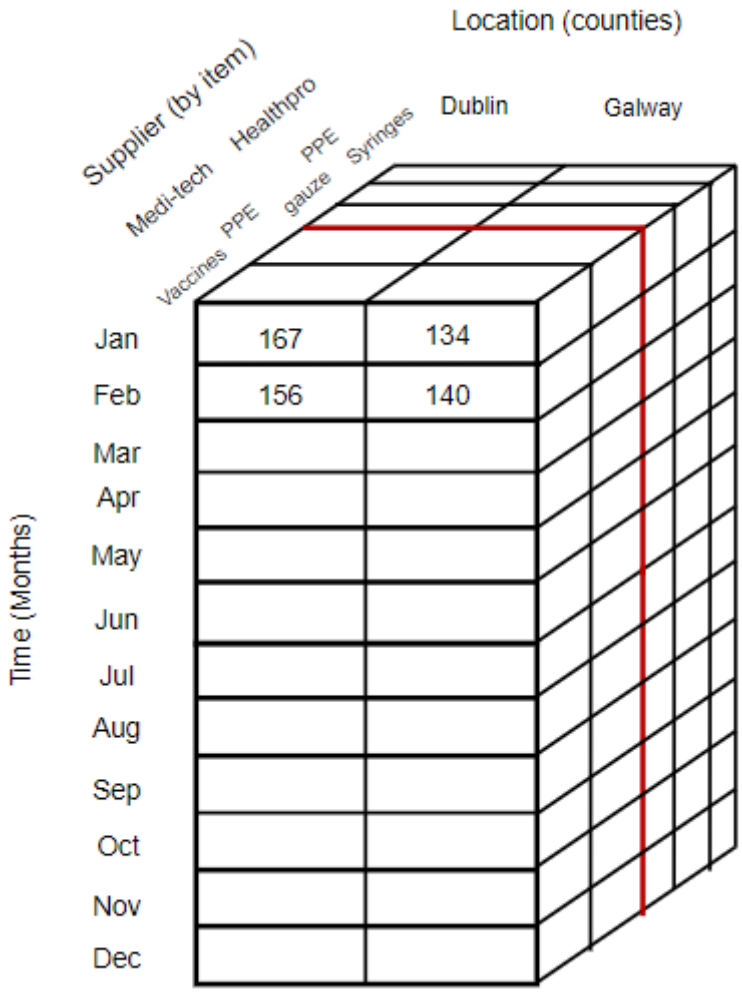


FIG 2.2. Slice of Data Cube corresponding to problem 2

The red line shows the split of items per every supplier. The values are again meaningless, just a sample of how the slice would look.

Knowing what the slice of the data marts that I want look like will definitely help to write the SQL to get them in this theoretical Data Warehouse.

Section 3

GP Appointments Query

Below is my query to return the data mart corresponding to my problem, “I want to know the average age of patients for each GP in Dublin and Cork, for all quarters of 2019 split by male and female patients”.

```
SELECT l.county, e.name, t.quarter, p.gender, sum(YEAR(CURRENT_TIMESTAMP) - YEAR(p.date_of_birth)) / count(*) as AverageAge
FROM GP Appointments gp, Time t, Patient p, Location l, Employee e
WHERE GP.time_key = t.timekey AND GP.location_key = l.location_key
AND GP.patient_key = p.patient_key AND GP.employee_key = e.employee_key
AND l.county in ["Dublin"city, "Cork"] AND t.year = 2019
GROUP BY l.county, p.employee_key, t.quarter, p.gender;
```

county	name	quarter	gender	AverageAge
Dublin	Dr. Ross	Q1	Male	46.3
Dublin	Dr. Ross	Q1	Female	43.4
Dublin	Dr. Ross	Q2	Male	48.7

FIG 3.1. SQL Query for problem 1 and corresponding sample table

I am selecting the value of the current year – the year in which the patient is born to get their approximate age, then summing these when we are grouping and then dividing by the rows that we have grouped together. This is the same result you would get from the avg() function, but this will allow for changes in the view as the count of rows and sum of ages can be tracked. It can also be seen that the way I chose to join is very verbose, but I feel it makes it clear what is being joined on.

I specified in the where clause that only practices in Dublin and Cork were considered and that only the data in 2019 was used. Lastly, I grouped on the county, employee_key, quarter and gender. These are also in the select clause just so it can be seen in tabular format which values of these the average age corresponds to.

This cube consists of 4 dimensional tables joined onto the fact table, the cube itself is similar to that of the image from section 2(fig2.1.). It will consist of chunks relating to the different combinations of attributes.

Pharmacy Deliveries

The query for my Pharmacy Deliveries data mart that solves the problem, "I want to know the total amount of items by type that were received each month, in Dublin and Galway from Medi-tech and Healthpro in 2018".

```
SELECT l.county, s.name, t.month, i.type, sum(pd.units_delivered) as AmountDelivered
FROM Pharmacy Deliveries pd, Time t, Item i, Location l, Supplier s
WHERE pd.time_key = t.timekey AND pd.location_key = l.location_key
AND pd.item_key = i.item__key AND pd.supplier_key = s.supplier_key
AND s.name in ["Medi-tech", "Healthpro"] AND t.year = 2018
GROUP BY l.county, s.name, t.month, i.type;
```

county	name	month	type	AmountDelivered
Dublin	Medi-tech	Jan	Vaccines	167
Dublin	Medi-tech	Jan	PPE	205
Dublin	Medi-tech	Feb	Vaccines	171
Dublin	Medi-tech	Feb	PPE	203
Dublin	Medi-tech	Mar	Vaccines	153

FIG 3.2. SQL Query for problem 2 and corresponding sample table

In a similar way to before, I am selecting the attributes that will help to identify the meaning of the value in the tuples. I am getting the amount delivered by summing across the tuples that are grouped together. The join, like the last one is clear but quite long. I only look at tuples that correspond to 2018 and are related to either Medi-tech and Healthpro. Lastly, I group on county first, then supplier name, month and finally item type.

This gives me a table that is ordered by county, supplier and then the types of items received in a given month. This should result in about 120 rows, 60 per county relating to the deliveries of 5 items over 12 months, 2 x 5 x 12.

This cube is also 4 – dimensional and again has a similar structure to the image in section 2 (fig2.2).

Conclusion

Having completed this project, I can say that creating a data warehouse from actual data with actual issues seems much more approachable. I now have a better understanding of the uses and advantages of a data warehouse with OLAP as opposed to a relational database with OLTP.

The nice thing about these 2 problems above is that they can easily be altered and other attributes of the table can be used, we could look at how many times people older than a certain date visit the GP in certain months by location for example. This is the advantage of having a unified, non-volatile collection of consolidated data for analysis.