DEVELOPMENT GUIDE

Davide A. Zollo
Shahrzad Fakour Fard

June 2020

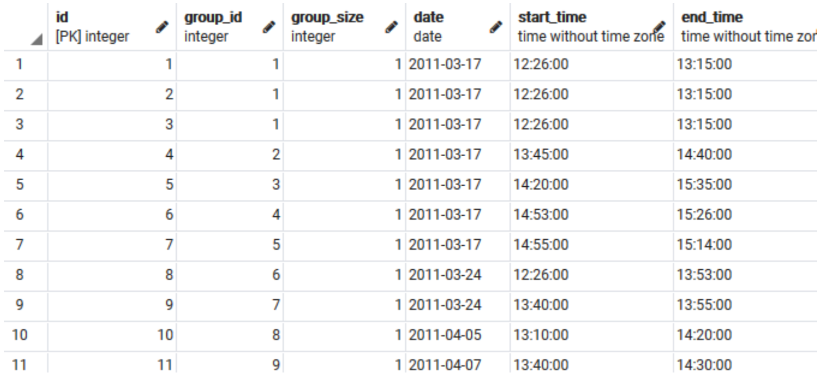
1 Introduction

The following guide will lead the user through the web service, helping him understand the work and reasoning done by the students. The project goal was to replicate a study published by the PUC that analyzed the behaviour of the visitors of the Reuben Hecht Museum, implementing a web service to illustrate the statistics and analysis done.

2 Data

The data storage was implemented through a PostgresSQL database that contained 3 tables.

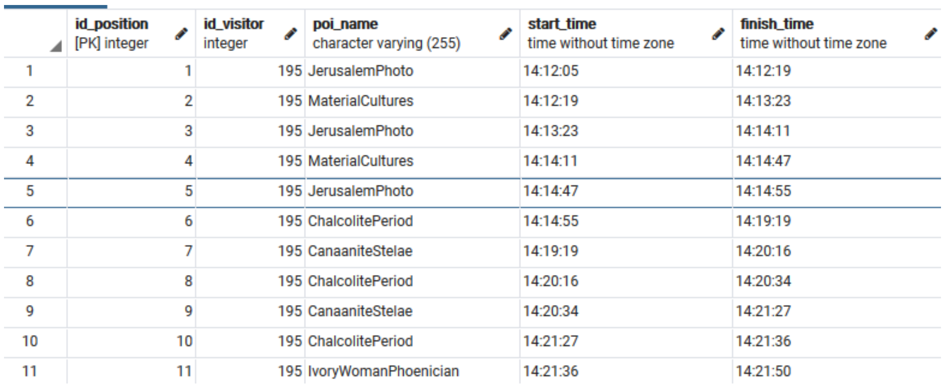
The visitor table is the transcription of the data proposed in the "*Visitors grouping.xlsx*" file. It contains the visitors basic information. The import was done through a java class that parsed every row of the excel file and added it as a tuple in the table. Only the first six columns of the excel file were imported since the other two contained "*holes*" in the file (null values) and were not deemed to be important for our project. Furthermore the column *group_size* wasn't used because in some cases it had mistaken values (this can be seen in Figure 1, group 1), thus the size was calculated dynamically.



	id [PK] integer	group_id integer	group_size integer	date date	start_time time without time zone	end_time time without time zor
1	1	1	1	2011-03-17	12:26:00	13:15:00
2	2	1	1	2011-03-17	12:26:00	13:15:00
3	3	1	1	2011-03-17	12:26:00	13:15:00
4	4	2	1	2011-03-17	13:45:00	14:40:00
5	5	3	1	2011-03-17	14:20:00	15:35:00
6	6	4	1	2011-03-17	14:53:00	15:26:00
7	7	5	1	2011-03-17	14:55:00	15:14:00
8	8	6	1	2011-03-24	12:26:00	13:53:00
9	9	7	1	2011-03-24	13:40:00	13:55:00
10	10	8	1	2011-04-05	13:10:00	14:20:00
11	11	9	1	2011-04-07	13:40:00	14:30:00

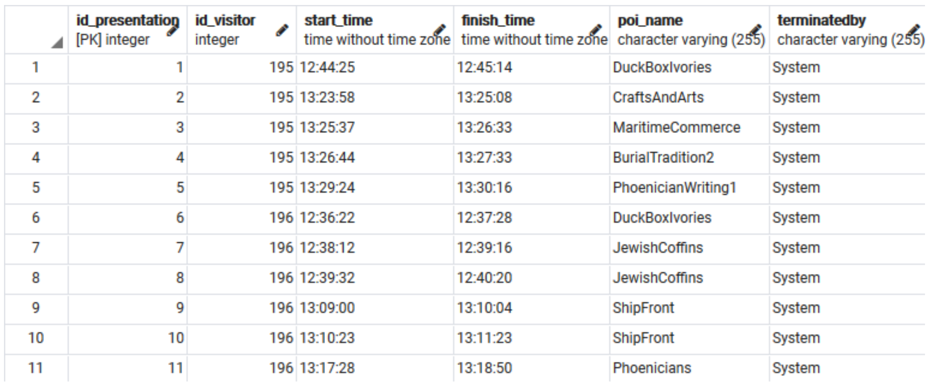
Figure 1: Visitor Table.

The positions table and the presentations table were imported from the log files of the users. This was done implementing a script that took the name of the files (which indicates the visitor id and the group id of the the subject of the log file) and positioned it as a value in the CSV file, then importing it as a table. The positions table includes 12.692 rows while the presentations table includes 2.460. Additionally, with the help of some SQL queries we identified 53 unique positions and 46 unique presentations.



	id_position [PK] integer	id_visitor integer	poi_name character varying (255)	start_time time without time zone	finish_time time without time zone
1	1	195	JerusalemPhoto	14:12:05	14:12:19
2	2	195	MaterialCultures	14:12:19	14:13:23
3	3	195	JerusalemPhoto	14:13:23	14:14:11
4	4	195	MaterialCultures	14:14:11	14:14:47
5	5	195	JerusalemPhoto	14:14:47	14:14:55
6	6	195	ChalcolitePeriod	14:14:55	14:19:19
7	7	195	CanaaniteStelae	14:19:19	14:20:16
8	8	195	ChalcolitePeriod	14:20:16	14:20:34
9	9	195	CanaaniteStelae	14:20:34	14:21:27
10	10	195	ChalcolitePeriod	14:21:27	14:21:36
11	11	195	IvoryWomanPhoenician	14:21:36	14:21:50

Figure 2: Position Table.



	id_presentation [PK] integer	id_visitor integer	start_time time without time zone	finish_time time without time zone	poi_name character varying (255)	terminatedby character varying (255)
1	1	195	12:44:25	12:45:14	DuckBoxIvories	System
2	2	195	13:23:58	13:25:08	CraftsAndArts	System
3	3	195	13:25:37	13:26:33	MaritimeCommerce	System
4	4	195	13:26:44	13:27:33	BurialTradition2	System
5	5	195	13:29:24	13:30:16	PhoenicianWriting1	System
6	6	196	12:36:22	12:37:28	DuckBoxIvories	System
7	7	196	12:38:12	12:39:16	JewishCoffins	System
8	8	196	12:39:32	12:40:20	JewishCoffins	System
9	9	196	13:09:00	13:10:04	ShipFront	System
10	10	196	13:10:23	13:11:23	ShipFront	System
11	11	196	13:17:28	13:18:50	Phoenicians	System

Figure 3: Presentation Table.

3 Web Service

In this section we will explain how the web service works, and how we achieved the results obtained. The service was comprised of three main features which will be discussed further above. The framework employed for the development of the web service was Spring Boot with the use of Thymeleaf for the creation of the pages.

3.1 Visitor Summary

The goal of this feature was to quickly review and summarize one of the viewers visit to the museum illustrating what we retained to be the most important statistics.

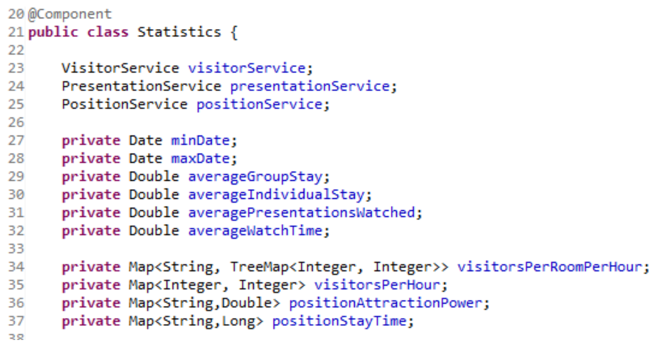
It starts with a form where you can search for the visit summary. This can be done using the visitor id or the group id of the subject. Once done, it will redirect you to the summary page. This page starts with some useful statistics about the visit. These are all computed dynamically through SQL commands and OOP techniques. Some of these are the subject total stay time in the museum, the number of points of interest visited, the total time watching presentations, etc.

This is followed by a user score that can be comprised between zero and three points, indicating the likening of the visit by subject. The score its calculated in the following way:

- If the user total stay time in the museum was above the average, he earns +1 score.
- If the user total presentation watch time was above the average, he earns +1 score.
- If the user didn't stop any presentation and watched each one through the end, he earns +1 score.

Lastly, we illustrated some charts that compared the subject visit to the average visit followed by the top ten most liked points of interest based on the user stay time.

As a quick note, all of the the museum general statistics, which includes all of the average visit statistics, are calculated at the launch of the web service and saved in a singleton class Statistics.



```java
@Component
public class Statistics {

    VisitorService visitorService;
    PresentationService presentationService;
    PositionService positionService;

    private Date minDate;
    private Date maxDate;
    private Double averageGroupStay;
    private Double averageIndividualStay;
    private Double averagePresentationsWatched;
    private Double averageWatchTime;

    private Map<String, TreeMap<Integer, Integer>> visitorsPerRoomPerHour;
    private Map<Integer, Integer> visitorsPerHour;
    private Map<String,Double> positionAttractionPower;
    private Map<String,Long> positionStayTime;
```

Figure 4: Quick view of the Statistics singleton class fields.

3.2 Museum Statistics

In this section we present the user with some useful statistics about the museum and visits.

For this purpose we thought that charts would be more explanatory than just the data. The first chart is a bar chart that illustrates the museum visitors "favourite" points of interest based on average stay time. The ten points showed are the ones with the biggest average stay time value calculated adding every stay time in that position and then dividing each one by the number of visitors it received. In this case what counts it's not how many visitors each point received, but how much time each one stayed. The next pie chart follows the same logic but this time the chart is calculated in terms of number of visits instead of stay time. Additionally, the rate of visitors each point receives is compared with the rest of the museum.

The spline area chart illustrates the average museum visitors per hour while the last one takes the top ten rooms positioned on the pie chart and explodes them into average visitors per hour in those rooms.

3.3 Replay a Visit

The main feature of the web service is the one that let's you accompany the visitor through his way in the museum.

First, the service consumer is introduced again with a form. Here he may search for the id of the visitor or group whose visit he wants to replay. Next up, we have a page with the map image of the museum and a pin pointing to the entrance, representing the start of the visit. Beneath the image we have a quick summary of every point of interest visited and the start visit button.

Once the visit has started the page the user is redirected to lets the user control each step of the visit. Using four buttons the user can control the navigation of the visit and in case he wants to skip many steps ahead or before, he can use the jump to button. On the bottom of the page there are some useful numbers about the current step. In the middle of the page, it's positioned the image of the museum map with the pinpointed location of the current point of interest. For every unique position there's an associated image which points its location in the museum.

As soon as the last point gets visited, the user is finally redirected to the home page.