WS 2021/22

Department of Geoinformatics Z_GIS

Spatial Databases

Supervisor: Mr. Martin Sudmanns

Exercise: Final Project

Database as a backend for a WebGIS Festival Application

By Edah Šahinović (12047186)

February 15th, 2022

Goal: Create a SQL database as a WebGIS backend for city festival in city of Visoko, Bosnia and Herzegovina to organize festival content, their working hours as well as starting and ending times for stage events.

Approach: PgAdmin4, QGIS, MySQL Workbench

DATA MODEL

First, prior to implementing the festival database a logical structure of a database is conceptualized through database model in *MYSQL Workbench*. It demonstrates how the database will be organized for the given task.

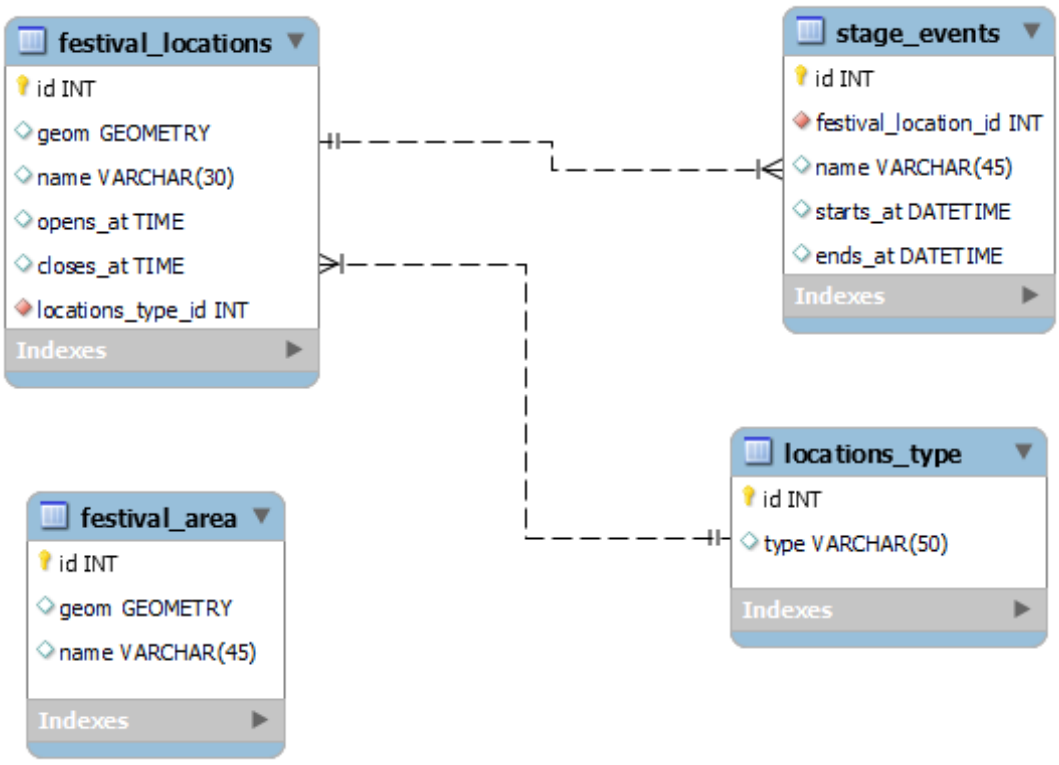

Figure 1 Database model for Visoko's city festival

Presented database model contains 4 entities and their relations:

- *Festival_locations* is a point entity that holds all locations that are considered as a part of the festival with their geographical location, type and name;
- *Locations_type* stores the type values of the festival locations.
- *Festival_area* is a polygon entity which decomposes the festival area into festival parts and it helps out the user to determine whether he/she is in the festival area and in which its respectful part he is located in;
- *Stage_events* is an entity that comprises of events for each of two festival stages. This table will help the user to discover at which day and when the events take place.

SETTING UP THE DATABASE

After the logical model for the database has been defined and city festival organizers are satisfied with the given concept, a phase of database creation is initiated. With the help of *QGIS* tables that contain geometries that are mapped as either point or polygon shapefiles and are connected to the database in *PgAdmin4* where final two entities and their overall relationships are created.

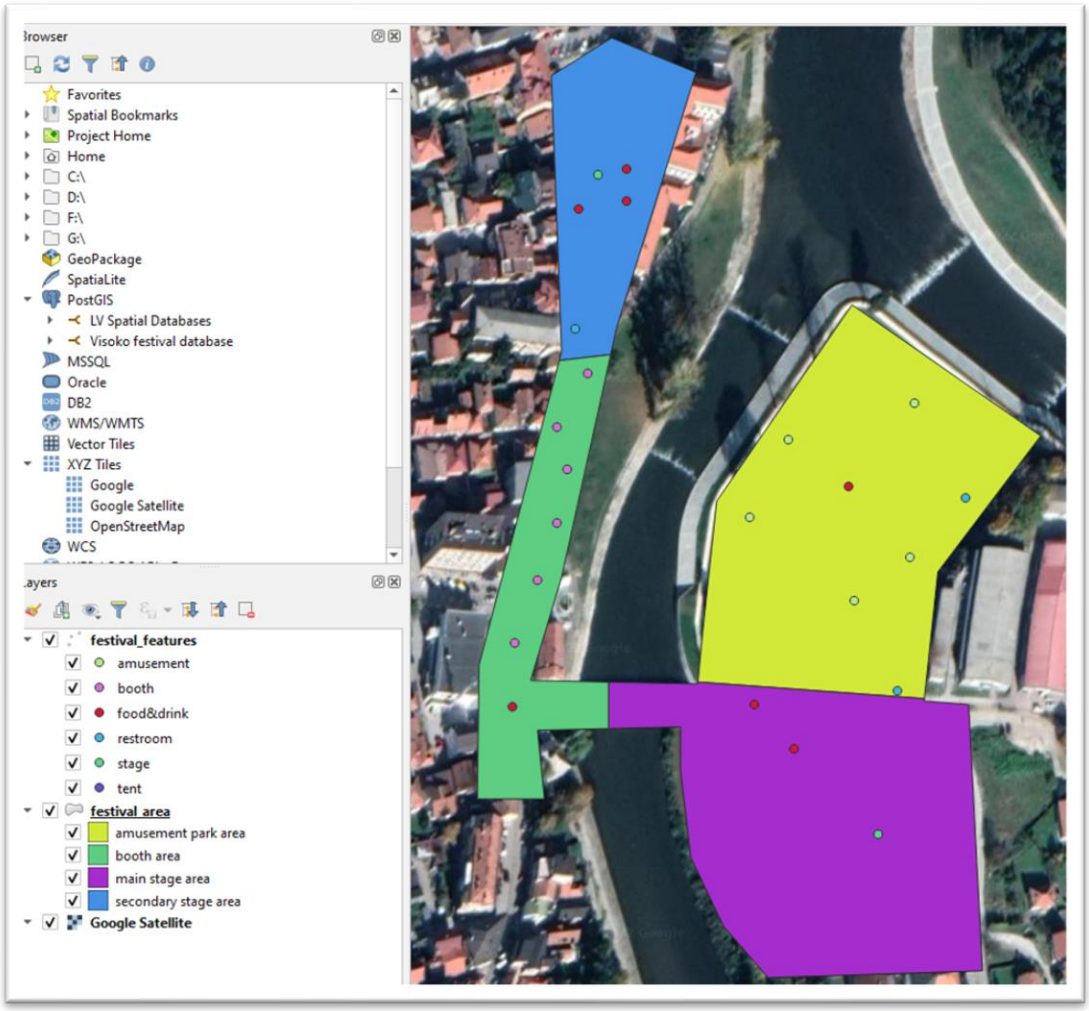

Figure 2 QGIS spatial entities creation

SQL QUERIES

As now the data has been created, populated, and linked together, it is time to perform standard and spatial SQL queries. Here we put ourselves in shoes of a city festival visitor that is not familiar with its content and has taken out he's phone to inspect and navigate through the event. This festival lasts for two days (2nd and 3rd of August 2022), and this visitor planned to spend its first day of festival before leaving the city. Here the queries are performed in *QGIS DB Manager* to simulate the mobile application which will provide the desired output followed by a digital map with manually categorized features.

First, the festival visitor wants to retrieve a list of festival locations to get familiarized with the festival content. Figure 3 depicts the query results, exhibiting the festival location names, their type, opening and closing hours.

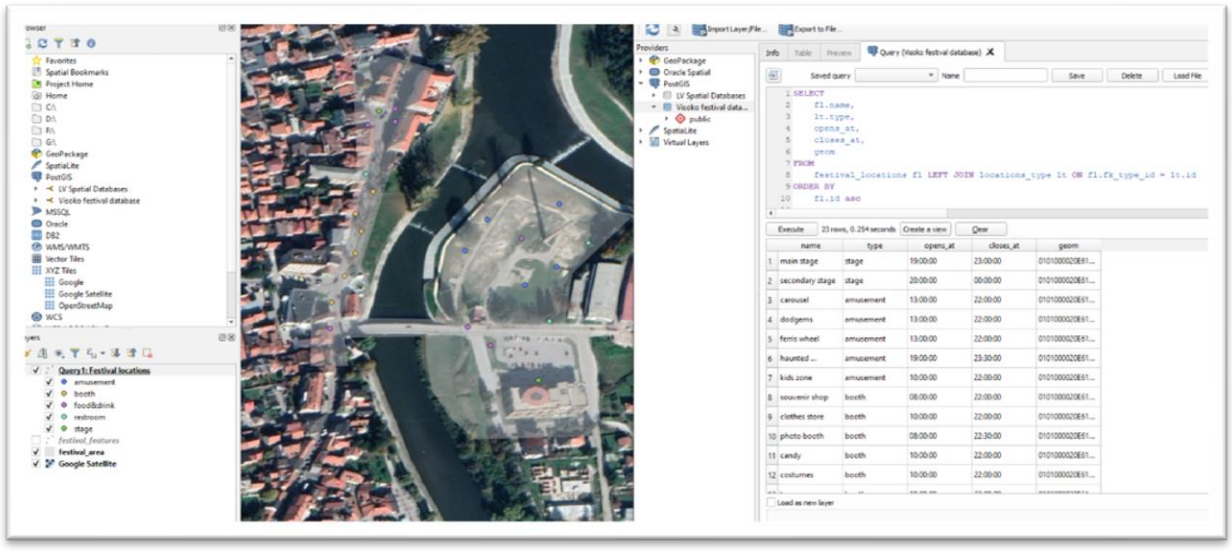

Figure 3 First query: festival locations

After the visitor has seen that working hours of festival locations differ, he wants to check which festival locations are open now. This temporal query is performed by *CURRENT_TIME()* function and lists all festival locations open at the time of executed query, in this case at 18:25. As previous query listed total of 23 records, this query listed 19 records, that is, 17 open locations. Locations such as main and secondary stage are left out as they open after this hour.

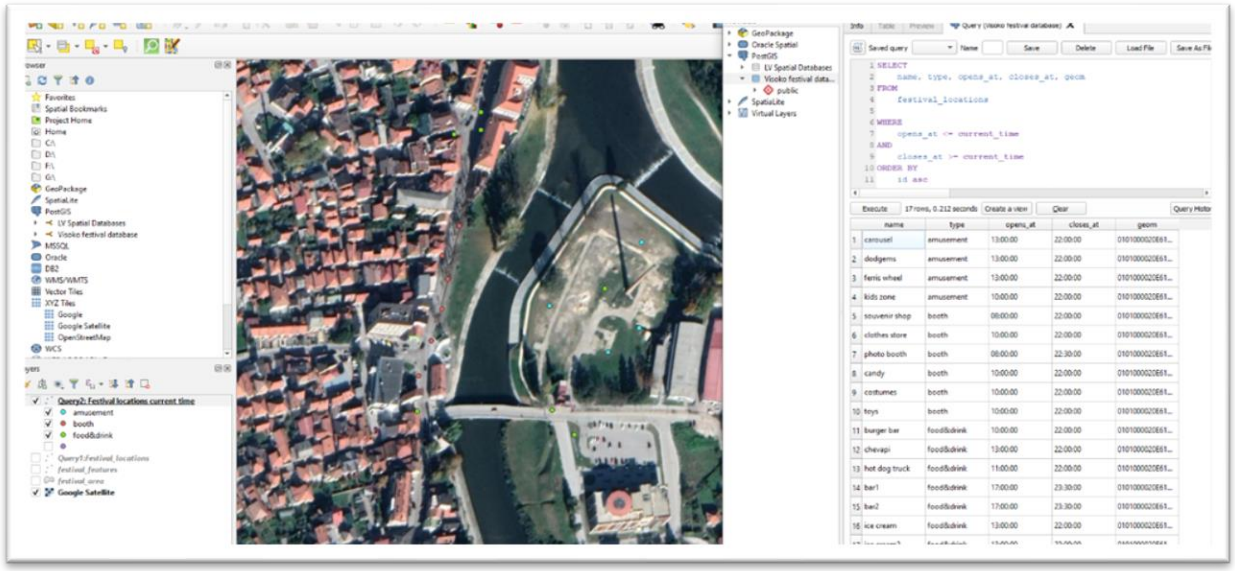

Figure 4 Second query: Festival locations working at 18:25

Next, the visitor wants to check if he is located within the festival area. User's location is mapped as a point input and is used in spatial query to retrieve the name of the festival area he is in. If the query does not return any result, the visitor knows he is outside the festival area.

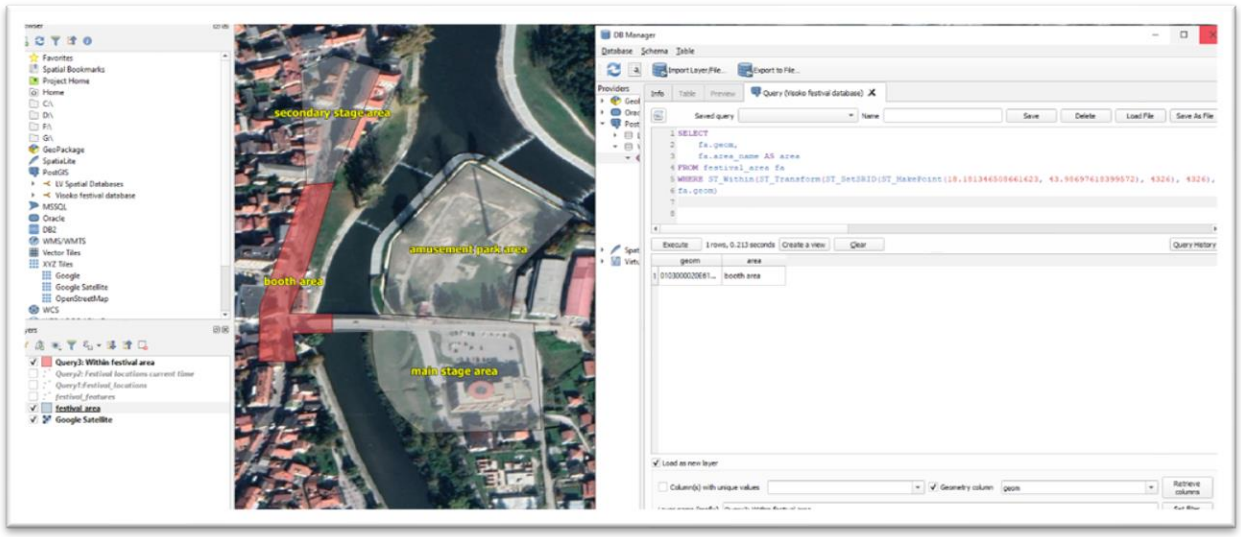

Figure 5 Third query: Visitor within festival area

The *ST_Within()* function takes users location, in this case an arbitrary point set near the bridge of festival area and a geometry column of festival area, providing with the column in which the visitor is in, in this case in the booth area.

At some point of his stay, the visitor would like to use a restroom. There are three restrooms spread across the festival area and using the closest one would be the most convenient choice.

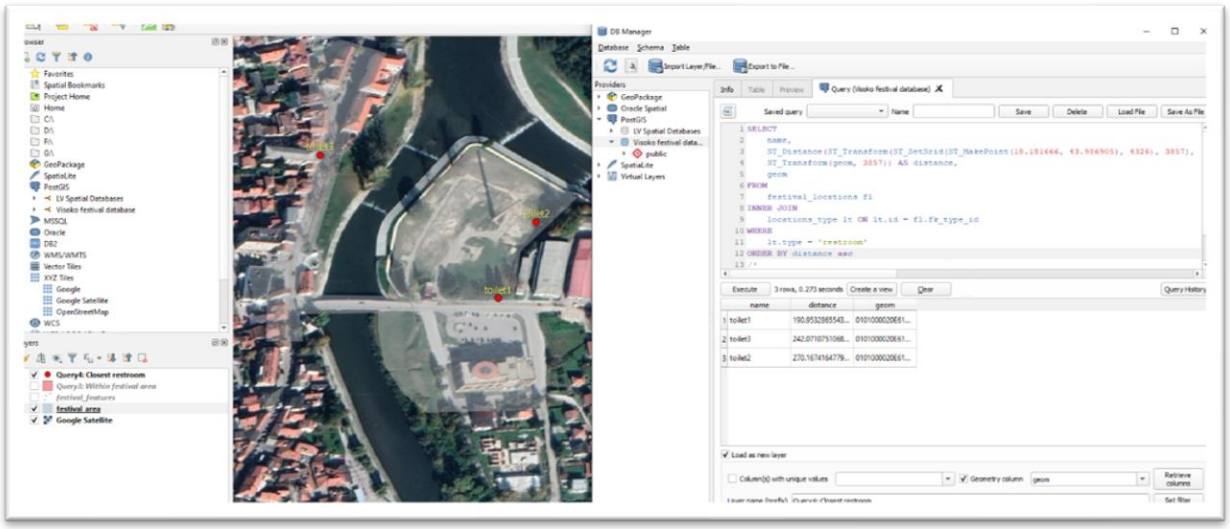

Figure 6 Fourth query: Closest restroom

Restrooms are mapped and labeled by their names. As we can see, the closest restroom from the visitor's same location being near the bridge is toilet number 1. If we'd like for the result to display only the closest record, we could do so with the *LIMIT 1* statement.

Lastly, this visitor has caught an eye on the festival stages and is interested what events and when they take place for the day he is in. First, he looks at all the events going happening throughout the complete festival.

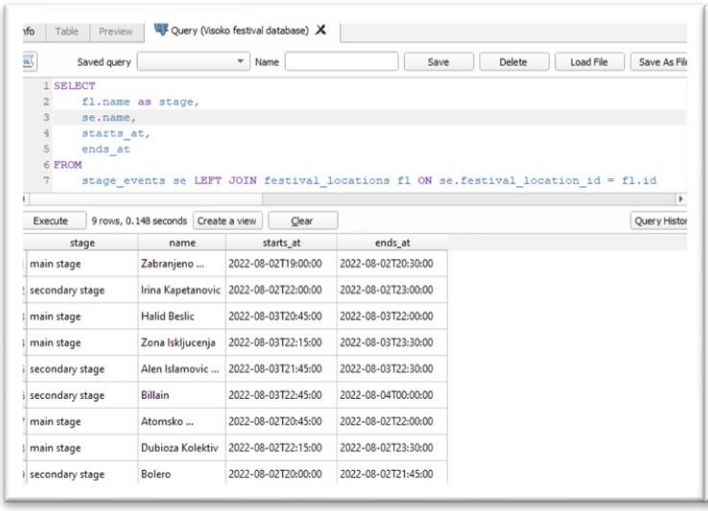

Figure 7 Fifth query: Entire stage events for the festival

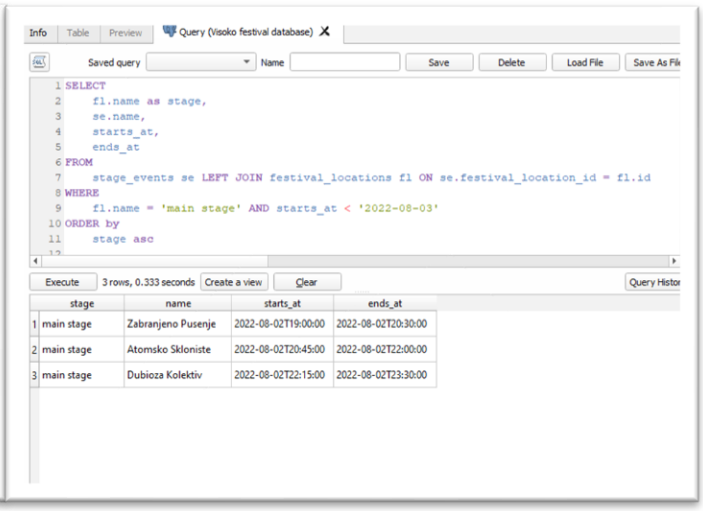

Figure 8 Sixth query: Stage events for main stage for first day of the festival

After the two-day list has been populated with both stage events, visitor decides to query only for those that are happening on the day he is on and is mainly interested in artists playing on main stage. Query results display the given criteria in Figure 8.

Now the mobile application is up and running, waiting for the visitors to start retrieving the festival information. Not only that the application users can access the information about the festival content but can also inspect the location as well as distance of the content and look over their and visitors' current location through use of digital map.

Each presented query has been stored in a separate view, the database, initial festival locations shapefile and database model have been included in the zipped folder.

CONCLUSION

In summary, the project involving the creation of a SQL database as a WebGIS backend for an imagined city festival in Visoko, Bosnia and Herzegovina, has been a significant learning experience and accomplishment. It allowed for the practical application of knowledge acquired during the course on spatial database systems, SQL language and GIS concepts. It signifies a promising start of my journey toward more advanced and user-centric GIS applications. It not only establishes a solid ground in in SQL but also sets the beginning of understanding of backend development for a broader, more interactive and sophisticated GIS application development path.