MARCH 13, 2023

DATABASE BACKEND FOR A WEBGIS

STEREO PICNIC MUSIC FESTIVAL

MARÍA PAULA RODRÍGUEZ NAVARRETE
MSC- CDE, ZGIS | UNIVERSITY OF SALZBURG

1. OBJECTIVE

The aim of this project is to develop a spatial database that serves as a backend for a WebGIS system for a Music Festival. The WebGIS should enable festival visitors to access a wealth of information about the event, while also providing the organizers with valuable data to improve their overall workflows. As such, the database must be designed to efficiently manage and organize data on various festival aspects, including stages, music events, artists, as well as the locations, opening times, and attributes of facilities like bathrooms, cash stations, information points, food & drink stalls, stores, and more.

2. CONTEXT

The proposed database was inspired on the "Estéreo Picnic" music festival, which is held annually since 2010 at the "Campo de Golf Briceño 18" near Bogotá, Colombia (Figure 1). It is one of the largest and most popular festivals in Colombia, featuring a diverse lineup of local and international artists across various genres such as rock, pop, electronic, hip-hop, and more.

Figure 1. Location of Stereo Picnic Festival

As this is a huge festival designed for more than 40000 people, for the purpose of this project, the original Festival Map and lineup were highly modified.

- **Note:** The original Festival Map and lineup for Estéreo Picnic 2022 can be found at the following link: Festival Estéreo Picnic 2022

3. WORKFLOW

To construct the database the following stages were followed (Figure 2). Each of them will be discussed in detail in the subsequent sections.

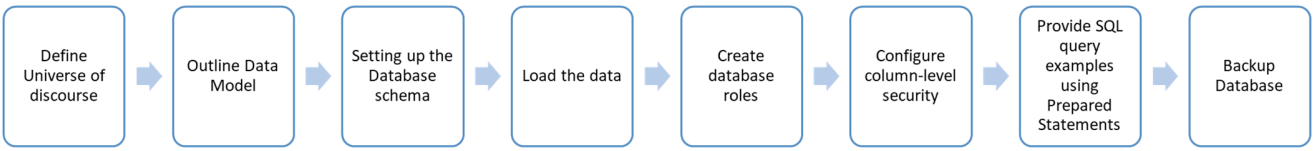

Figure 2. Workflow implemented to build the spatial database

4. DEFINE UNIVERSE OF DISCOURSE

Imaginary Universe → The Stereo Picnic music festival is coming up in three days, from March 24 to March 26, 2023 at the "Campo de Golf Briceño 18". The festival features four stages named after its sponsors: Adidas, Paramo, Bogota Bank, and the Principal Stage. Each stage has a dedicated team of staff members responsible for organizing logistics. While additional staff members can be requested if necessary, the maximum number of staff members for each stage cannot be exceeded due to limited space. The festival schedule (Figure 3) has already been announced to the public, and it is important to follow the plan to avoid confusion.

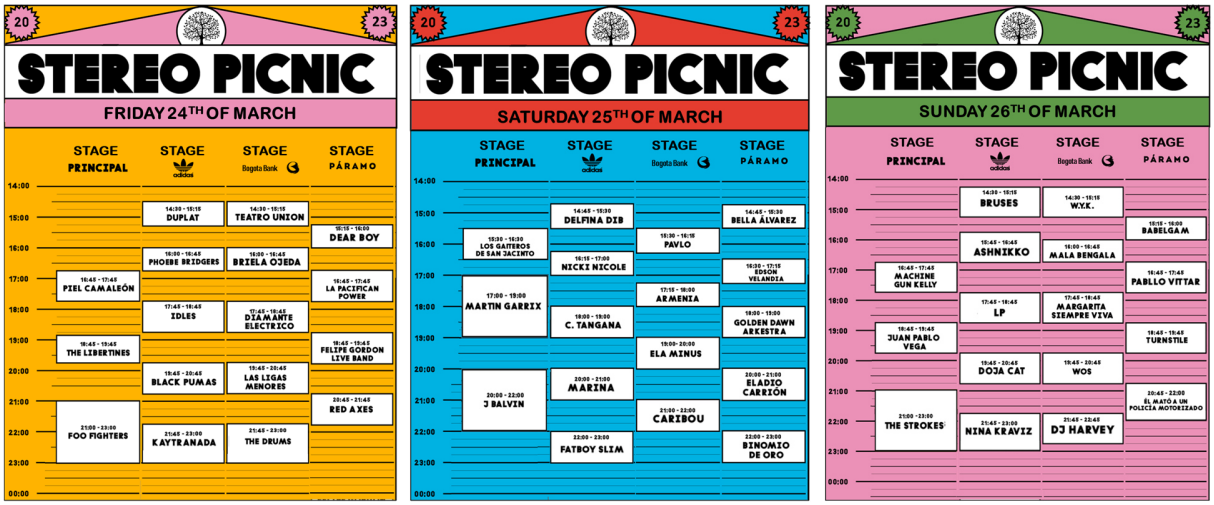

Figure 3. Schedules for the music events per day per stage

To streamline transactions, the festival will adopt a cashless system. Upon entering the festival, each attendee will receive a wristband which can be recharged at any of the cash stations within the festival area. These cash stations accept cash, card, or both payment methods, and will occasionally undergo maintenance to ensure optimal functionality.

The recharged money can be used to purchase items in the Store and Food & Drink zones. The festival will offer a wide variety of food and drinks from major brands and entrepreneurs, as well as merchandise from commercial and artisan vendors. Prices will range from high to low.

Just like the stages, the Store and Food & Drink zones will have a dedicated cleaning staff to ensure a hygienic and pleasant environment for festivalgoers. While additional staff can be requested if necessary, the number of available staff members is limited.

Multiple bathroom cubicles will be located throughout the festival area, with some offering disability access. These cubicles may be in maintenance as the goal is to keep them clean and well-maintained.

If attendees have any questions that cannot be answered through the Web GIS, they can visit the distributed information points for assistance. Staff members at these points can even call colleagues at other locations if needed to provide the best possible help and support.

The intended location and names of the festival facilities are provided in the map – Coordinate System: 3857 WGS 84 / Pseudo-Mercator (Figure 4).

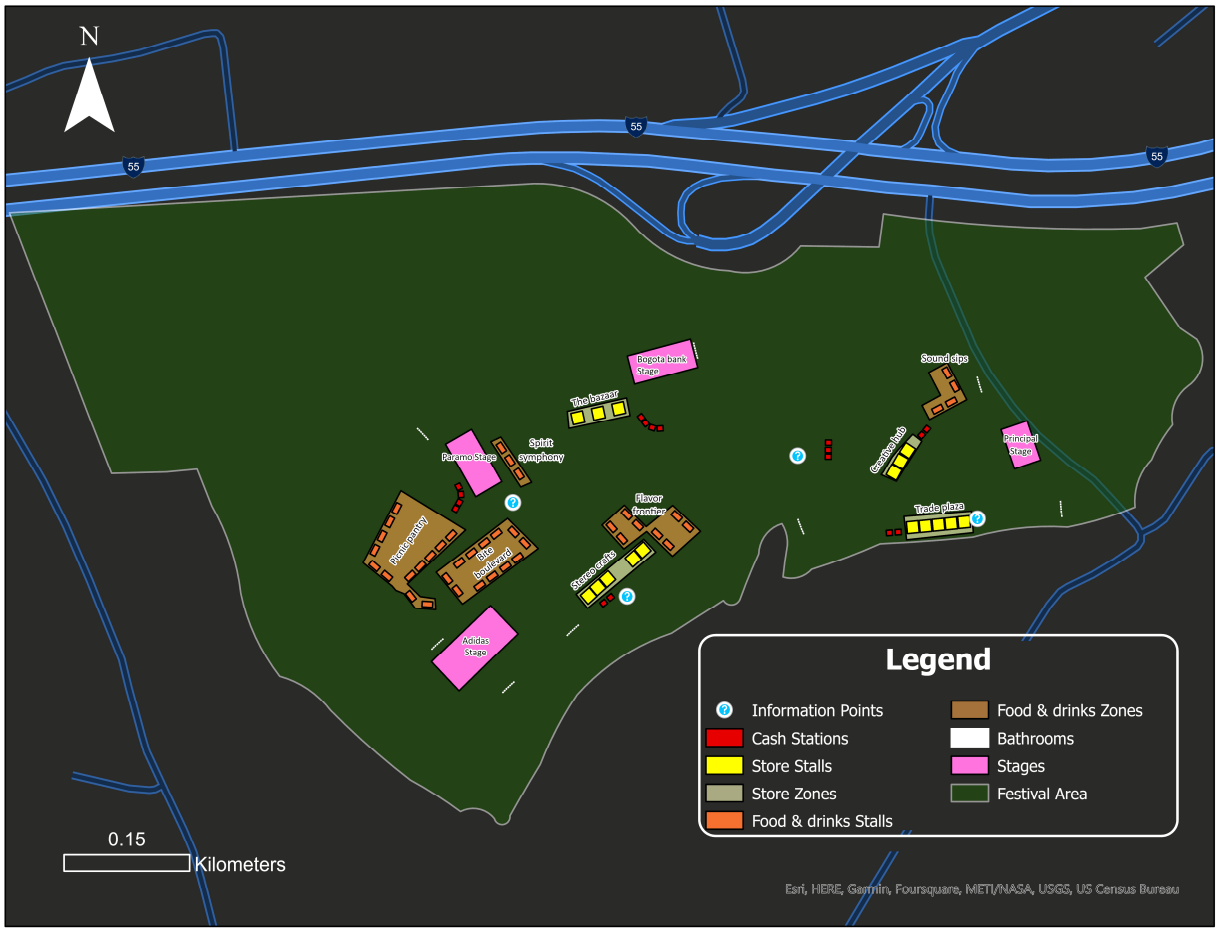

Figure 4. Stereo Picnic Map with facilities location and names

5. OUTLINE DATA MODEL

Considering the universe of discourse, the following entity relationship model was built with 10 entities representing the real objects or concepts:

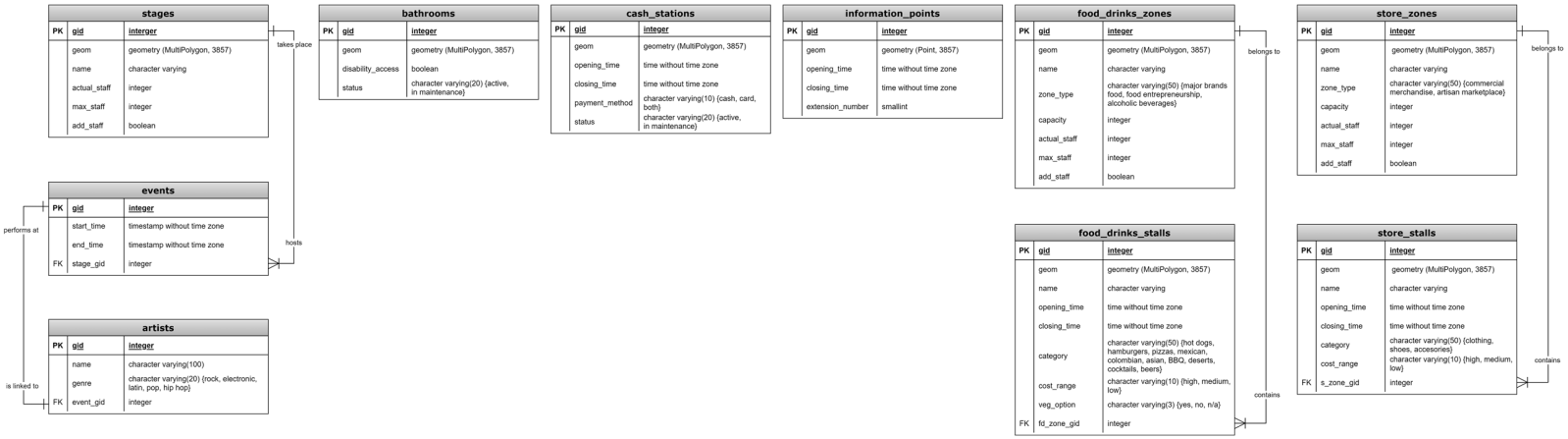

Figure 5. Data Model Stereo Picnic Festival

6. SETTING UP THE DATABASE SCHEMA AND 7. LOAD THE DATA

For simplicity, all the table names, fields and data were provided in lowercase and without any accents for Spanish words.

- **In pgAdmin:**
  - Created a database called “stereopicnic” with the POSTGIS extension enabled
  - Created the artists and events tables, as they do not have any spatial field
- **In QGIS:**
  - Georeferenced the provided map and use it to create the spatial layers (Coordinate System: 3857) with the proper geometry: stages, bathrooms, cash_stations, information points, food_drinks_zones, food_drinks_stalls, store_zones and store_stalls.
  - Imported the layers to the database with the QGIS DB Manager, leaving the “create spatial index option” marked in the process
- **Back in pgAdmin:**
  - The data for all the tables was added using the Data Manipulation Language (DML)
  - Additionally, the relationships identified in the data model were set up.

8. CREATE DATABASE ROLES

Based on the requirements, the final Web GIS will not only serve as an information resource for festival attendees, but it will also aid the staff in improving their workflow organization. Considering the latter, two roles without privileges were added to the database in pgAdmin: **attendees and staff**.

- attendees password: StereoPicnic2023_attendees
- staff password: StereoPicnic2023_staff

9. CONFIGURE COLUMN-LEVEL SECURITY

Instead of relying on views that can be problematic, built-in functions of column-level were configured as follows:

Table	staff	attendees
stages	**SELECT:** all fields **UPDATE:** only actual_staff and add_staff	**SELECT:** all fields except actual_staff, max_staff and add_staff
events	**SELECT:** all fields	**SELECT:** all fields
artists	**SELECT:** all fields	**SELECT:** all fields
bathrooms	**SELECT:** all fields **UPDATE:** only status	**SELECT:** all fields
cash_stations	**SELECT:** all fields **UPDATE:** only status	**SELECT:** all fields
information_points	**SELECT:** all fields	**SELECT:** all fields except extension_number
food_drinks_zones	**SELECT:** all fields **UPDATE:** only actual_staff and add_staff	**SELECT:** all fields except actual_staff, max_staff and add_staff
food_drinks_stalls	**SELECT:** all fields	**SELECT:** all fields
store_zones	**SELECT:** all fields **UPDATE:** only actual_staff and add_staff	**SELECT:** all fields except actual_staff, max_staff and add_staff
store_stalls	**SELECT:** all fields	**SELECT:** all fields

For example, attendees cannot SELECT ALL the fields of the stages table (Figure 6), only the gid, geom and name fields (Figure 6)

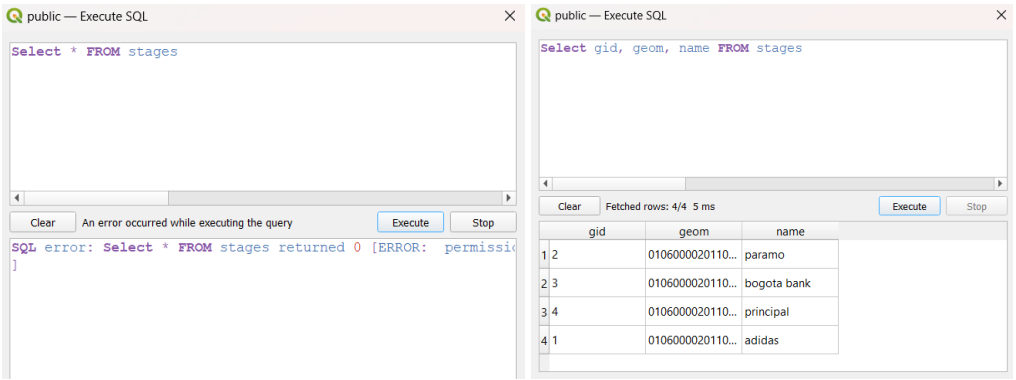

Figure 6. attendees unable to query ALL fields, only the ones with SELECT privilege

10. PROVIDE SAMPLE SQL QUERIES AS PREPARED STATEMENTS (TO AVOID SQL INJECTIONS)

For each of the roles different useful queries were proposed based on their assigned privileges. These queries can also be checked in a sql file attached (sample_queries.sql)

THEMATIC: STAGES/EVENTS/ARTISTS

Role: staff

```
-- 1. Request for extra staff for a particular stage.
Input: stage name
PREPARE update_stage_staff AS
  UPDATE stages SET add_staff = true WHERE name =
LOWER($1);

EXECUTE update_stage_staff('AdIdas');

-- 2. Return how many additional staff members can be sent
to a stage.
Input: stage name
PREPARE count_additional_staff AS
  SELECT max_staff - actual_staff AS additional_staff
  FROM stages
  WHERE LOWER(name) = LOWER($1);

EXECUTE count_additional_staff('PARAMO');
```

Role: attendees

```
-- 3. Retrieve upcoming events today of a specified genre,
including the start time, end time, artist name, stage
name and distance (meters) to the stage.
Input: attendee location (GPS coordinates), genre
PREPARE retrieve_upcoming_events AS
  SELECT
    e.start_time,
    e.end_time,
    a.name AS artist,
    s.name AS stage,
    ST_Distance(s.geom,
ST_Transform(ST_SetSRID(ST_MakePoint($2, $3), 4326),
3857)) AS distance_meters
  FROM
    public.events e
    JOIN public.stages s ON e.stage_gid = s.gid
    JOIN public.artists a ON e.gid = a.event_gid
  WHERE
    a.genre = LOWER($1)
    AND DATE(e.start_time) = DATE(NOW())
    AND e.start_time >= NOW()
  ORDER BY
    e.start_time ASC;

EXECUTE retrieve_upcoming_events('Rock', -73.969537,
4.935142);

------ example to test:

PREPARE retrieve_upcoming_events AS
  SELECT
    e.start_time,
    e.end_time,
    a.name AS artist,
    s.name AS stage,
    ST_Distance(s.geom,
ST_Transform(ST_SetSRID(ST_MakePoint($2, $3), 4326),
3857)) AS distance_meters
  FROM
    public.events e
    JOIN public.stages s ON e.stage_gid = s.gid
    JOIN public.artists a ON e.gid = a.event_gid
  WHERE
    a.genre = LOWER($1)
    AND DATE(e.start_time) = '2023-03-24'
    AND e.start_time >= '2023-03-24 15:00:00'
  ORDER BY
    e.start_time ASC;

EXECUTE retrieve_upcoming_events('Rock', -73.969537,
4.935142);
```

	-- 4. Provide information on the date, start time, end time, and stage name where a particular artist will be performing.
	Input: artist name
	PREPARE give_artist_event_info AS
	  SELECT e.start_time, e.end_time, s.name AS stage_name
	  FROM public.events e
	  JOIN public.stages s ON e.stage_gid = s.gid
	  JOIN public.artists a ON e.gid = a.event_gid
	  WHERE LOWER(a.name) = LOWER($1);
	
	EXECUTE give_artist_event_info('J balvin');

THEMATIC: BATHROOMS

Role: staff	Role: attendees

-- 5. Retrieve a list of bathrooms that are active or that require maintenance.
Input: status
PREPARE return_bathrooms_by_status AS
  SELECT *
  FROM public.bathrooms
  WHERE status = LOWER($1);
 	
EXECUTE return_bathrooms_by_status('in maintenance');
 	
-- 6. Update the status of a bathroom from active to in maintenance or vice versa.
Input: bathroom gid, new status
PREPARE update_bathroom_status AS
  UPDATE public.bathrooms SET status = LOWER($2)
  WHERE gid = $1;
 	
EXECUTE update_bathroom_status(1234, 'Active');
	-- 7. Find the nearest active bathroom with disability access and return its gid and distance (meters) from the attendee's location. Input: attendee location (GPS coordinates).
	PREPARE find_nearest_bathroom_da (float8, float8) AS
	  SELECT gid, ST_Distance(geom, ST_Transform(ST_SetSRID(ST_MakePoint($1, $2), 4326), 3857)) AS distance_meters
	  FROM public.bathrooms
	  WHERE disability_access = true
	  AND status = 'active'
	  ORDER BY distance_meters
	  LIMIT 1;
	
	EXECUTE find_nearest_bathroom_da (-73.969537, 4.935142);

THEMATIC: CASH STATIONS

Role: staff	Role: attendees

-- 8. Check for all the cash_stations that require maintenance.
SELECT gid
FROM public.cash_stations
WHERE status = 'in maintenance';
	-- 9. Nearest 5 active cash stations that are active, open and accept a specified payment method, along with their distance from the attendee's location.
	Input: attendee location (GPS coordinates), payment method
	PREPARE find_nearest_cash_stations AS
	  SELECT gid, ST_Distance(geom, ST_Transform(ST_SetSRID(ST_MakePoint($1, $2), 4326), 3857)) AS distance_meters
	  FROM public.cash_stations
	  WHERE status = 'active'
	  AND payment_method = $3
	  AND opening_time <= CURRENT_TIME
	  AND closing_time > CURRENT_TIME
	  ORDER BY distance
	  LIMIT 5;
	
	EXECUTE find_nearest_bathroom_da (-73.969537, 4.935142,'both’);
	
	------ example to test:
	PREPARE find_nearest_cash_stations AS
	  SELECT gid, ST_Distance(geom, ST_Transform(ST_SetSRID(ST_MakePoint($1, $2), 4326), 3857)) AS distance_meters
	  FROM public.cash_stations
	  WHERE status = 'active'

	AND payment_method = $3 AND opening_time <= '13:51:00' AND closing_time > '13:51:00' ORDER BY distance_meters LIMIT 5; EXECUTE find_nearest_cash_stations (-73.969537, 4.935142,'both');

Thematic: Information Points

Role: staff	Role: attendees
-- 10. Retrieve the extension number of all the information points open. SELECT extension_number FROM public.information_points WHERE opening_time <= CURRENT_TIME AND closing_time >= CURRENT_TIME;	-- 11. Return the nearest open information point along with its distance. Input: attendee location (GPS coordinates) PREPARE find_nearest_information_point AS SELECT gid, ST_Distance(geom, ST_Transform(ST_SetSRID(ST_MakePoint($1, $2), 4326), 3857)) AS distance_meters FROM public.information_points WHERE opening_time <= CURRENT_TIME AND closing_time > CURRENT_TIME ORDER BY distance_meters LIMIT 1; EXECUTE find_nearest_information_point (-73.969537, 4.935142);

Thematic: Food and Drinks Zones and Stalls

Role: staff	Role: attendees
-- 12. Provide the zones with the least staff members at the moment. SELECT name, actual_staff FROM public.food_drinks_zones ORDER BY actual_staff ASC LIMIT 2;	-- 13. Give information of the zone where attendee is now located. Input: attendee location (GPS coordinates) PREPARE current_fd_zone_info AS SELECT name, zone_type, capacity FROM public.food_drinks_zones WHERE ST_Intersects(geom, ST_Transform(ST_SetSRID(ST_Point($1, $2), 4326), 3857)); EXECUTE current_fd_zone_info (-73.967807, 4.935218); -- 14. Retrieve the 3 nearest food and drink stalls that are currently open, offer vegan options, have a low cost range, and provide their name, category, distance in meters, and associated food and drinks zone. Input: attendee location (GPS coordinates) PREPARE find_nearest_specified_fd_stall AS SELECT s.name AS stall_name, s.category, ST_Distance(s.geom, ST_Transform(ST_SetSRID(ST_MakePoint($1, $2), 4326), 3857)) AS distance_meters, s.name AS zone_name FROM public.food_drinks_stalls s JOIN public.food_drinks_zones z ON s.fd_zone_gid = z.gid WHERE s.opening_time <= CURRENT_TIME AND s.closing_time >= CURRENT_TIME AND s.cost_range = 'low' AND s.veg_option = 'yes' ORDER BY

	distance_meters ASC LIMIT 3; EXECUTE find_nearest_specified_fd_stall (-73.968716, 4.935845);

THEMATIC: STORES ZONES AND STALLS

Role: staff	Role: attendees
-- 15. Provide the zones with the highest staff members at the moment. SELECT name, actual_staff FROM public.store_zones ORDER BY actual_staff DESC LIMIT 2;	-- 16. Return closest open store_stall where a festival attendee can look for products of a specific category done by artisans. Input: attendee location (GPS coordinates), and stall category (shoes, clothing or accessories) PREPARE find_nearest_specified_s_stall AS SELECT s.name AS stall_name, s.category, s.cost_range, ST_Distance(s.geom, ST_Transform(ST_SetSRID(ST_MakePoint($1, $2), 4326), 3857)) AS distance_meters, z.name AS zone_name, s.opening_time, s.closing_time FROM public.store_stalls s JOIN public.store_zones z ON s.s_zone_gid = z.gid WHERE s.category = $3 AND z.zone_type = 'artisan marketplace' AND s.opening_time <= CURRENT_TIME AND s.closing_time >= CURRENT_TIME ORDER BY distance_meters ASC LIMIT 1; EXECUTE find_nearest_specified_s_stall (-73.968716, 4.935845, 'shoes');

For instance, logged in to the database as a staff member in QGIS, the query No. 2 returns:

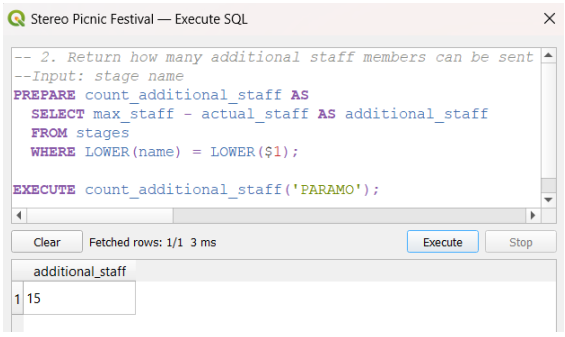

Figure 7. Test Query No. 2 with staff role in QGIS

Or logged in to the database as an attendee in QGIS, the query No. 7 returns:

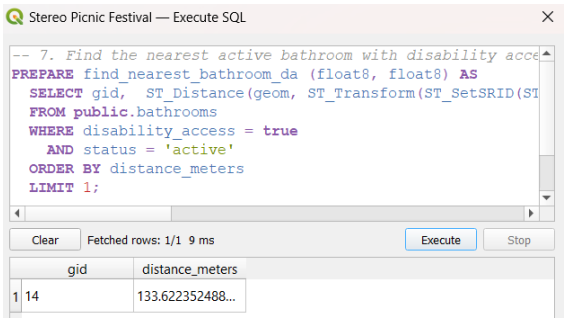

Figure 8. Test Query No. 7 with attendee role in QGIS

11. BACKUP DATABASE

As a security measure in case of data loss or corruption, a backup of the completed database was created using the pg_dump utility. The backup file can be checked attached (Final_backup.sql).

To restore the database:

- First, create 2 users called “staff” and “attendees” without privileges
- Restore the database with the attached file