Cebby Wrapped: Data Queries and Technical Implementation

To build "Cebby Wrapped," we need a structured way to pull specific, personalized metrics from the database. Given the stack, we will focus on **Supabase** for data queries and **Astro** for the frontend presentation.

Phase 1: Core Data Queries

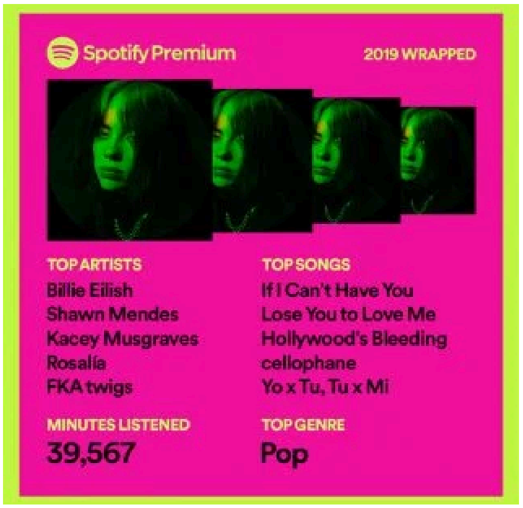
Since we are using Supabase, these queries would be executed as **PostgreSQL functions (using pl/pgsql) or as Supabase Edge Functions** (Deno) that query the database.

Metric	Required Data Fields	Database Query / Logic
Total Events Attended	event_registrations.profile_id, event_registrations.check_in_at	**Query:** Count all records in event_registrations for a given user where check_in_at is !NULL and the event date is within the Wrapped period (e.g., Jan 1 - Dec 31).
First Event Attended	event_registrations.profile_id, events.name, events.start_time	**Query:** Select the events.name associated with the oldest event_registrations record (smallest events.start_time) where the user was checked in.
Total Hours Spent Learning	event_registrations.profile_id, events.start_time, events.end_time	**Logic:** Sum the difference between events.end_time and events.start_time for all attended events.
"Tech Stack Focus" (Category)	event_registrations.profile_id, events.category (added to the db or let AI categorize it)	**Query:** Group attended events by events.category (e.g., Web Dev, AI, Design) and count the frequency.

Top Month for Attendance	event_registrations.profile_id, events.start_time	**Query:** Group attended events by month (MONTH(events.start_time)) and count the frequency.

Phase 2: Advanced Logic & Storytelling

To make the Wrapped more engaging, we can run comparative queries across the whole community. Here, we can create summary visualizations.

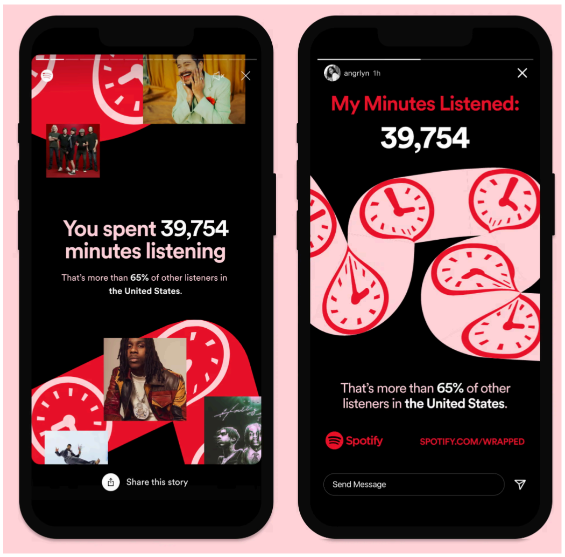

Spotify Premium 2019 WRAPPED

TOP ARTISTS
Billie Eilish
Shawn Mendes
Kacey Musgraves
Rosalía
FKA twigs

TOP SONGS
If I Can't Have You
Lose You to Love Me
Hollywood's Bleeding
cellophane
Yo x Tu, Tu x Mi

MINUTES LISTENED
39,567

TOP GENRE
Pop


You spent 39,754 minutes listening

That's more than 65% of other listeners in the United States.

Share this story

angrlyn 1h

My Minutes Listened:
39,754

That's more than 65% of other listeners in the United States.

Spotify SPOTIFY.COM/WRAPPED

Send Message

Source Example: Spotify Wrapped often shows users "You were in the top 5% of [Artist]'s listeners."

1. The "Top % of Attendees" Badge

- **Goal:** Tell the user they are highly engaged compared to others.
- **Metric: Top X% of Cebby Attendees**.
- **Logic:**
    1. Get the current user's **Total Events Attended** count (A).
    2. Get the **Total Events Attended** count for *every other user* (B).
    3. Calculate the user's percentile rank:

$$\text{Percentile} = \frac{\text{Count of users with attendance } < A}{\text{Total number of users}} \times 100$$

    4. If the result is 90, the user is in the **Top 10%** of attendees.

2. Your 'Most Popular' Event

- **Goal:** Show the user which of the events they attended was the biggest community-wide success.
- **Metric: The Crowd Favorite** (Name of event).
- **Logic:**
    1. Get a list of all events the user attended.
    2. For each event in that list, count the **total number of unique attendees** community-wide.
    3. The event with the **highest total attendance** is their "Crowd Favorite."

Phase 3: Implementation with the Tech Stack (Supabase & Astro)

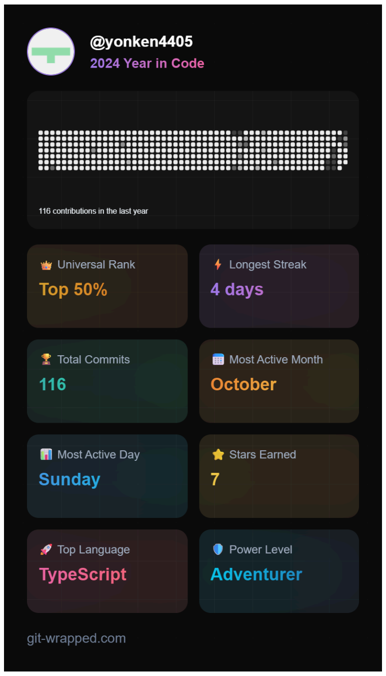
Backend (Supabase)

- **Data Query Method:** We have two excellent options in Supabase:
    1. **Supabase Edge Functions (Deno):** This is the recommended approach. Create a function (e.g., `get-user-wrapped`). This function would authenticate the user, run all the complex PostgreSQL queries from Phase 1 & 2, aggregate the data into a single JSON object, and return it. This keeps our Astro frontend clean (it just makes one API call).
    2. **PostgreSQL Functions (pl/pgsql):** We could create a complex function *inside* the database (e.g., `fn_get_wrapped_stats(user_id)`) and call it via `supabase.rpc()` from the frontend. This is very fast but moves business logic into the database.

Frontend (Astro)

Astro is perfect for this, as we can build a fast-loading static page (`/wrapped/2024`) that becomes dynamic when the user logs in.

- **Data Visualization:** Since Astro is component-based, we can use any UI-agnostic library. **Chart.js** or **Recharts** are still excellent choices that can be easily wrapped in an Astro component (using a `client:load` directive).
- **Image Generation (The Shareable Card):** This is the most critical part for virality.
    - **Client-Side:** Use a library like **html2canvas**. The user clicks a 'Share' button, the library screenshots a specific `<div>` on our Astro page, and provides a PNG for them to save. (This is easy, but less controlled).
    - **Server-Side (Recommended):** Use a Supabase Edge Function with a library like **Satori** (from Vercel, works in Deno) to dynamically generate an image. The user would go to `.../wrapped/share-card.png?user_id=123`, and our function would generate the image on the fly with their stats. This is how GitHub Wrapped and Spotify do it.



@yonken4405
2024 Year in Code

116 contributions in the last year

Universal Rank Top 50%	Longest Streak 4 days
Total Commits 116	Most Active Month October
Most Active Day Sunday	Stars Earned 7
Top Language TypeScript	Power Level Adventurer

git-wrapped.com

Source Example: Git Wrapped (a popular community project) generates a single image card that users post on Twitter/LinkedIn.