

Towards DROP TABLE analytics;

Tylan O'Flynn

Lighthouse Labs

January 24, 2024

Outline

Project/Goals

Process

Analysis



Figure 1: Credit:

<https://www.wikihow.com/Destroy-Sensitive-Documents>

Project/Goals

The goal of this project was to take a toy dataset with messy values and convert it into a normalized relational database managed using PostgreSQL, then answer questions based on that data.

Import

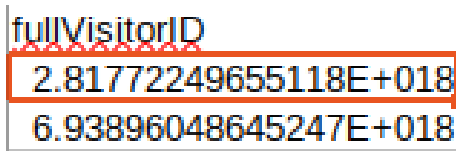
First, I imported the data in the .csvs 'without modification' into a PostgreSQL-managed database.

fullVisitorID	channelGrouping	time	country	city
2.81772249655118E+018	Direct	122213	Taiwan	(not set)
6.93896048645247E+018	Referral	268947	United States	not available in demo dataset
5.30755433175428E+018	Referral	61402	United States	not available in demo dataset
5.25694987374209E+018	Display	8655	United States	not available in demo dataset
7.54930869799597E+018	Referral	0	United States	London
5.44491396189027E+018	Organic Search	25667	(not set)	(not set)
9.14760062456296E+017	Organic Search	14737	El Salvador	not available in demo dataset
4.47681124563799E+018	Organic Search	123686	United States	not available in demo dataset
5.15303863541975E+018	Organic Search	8860	United Kingdom	not available in demo dataset
7.19635766852235E+017	Organic Search	39393	Australia	Sydney

Figure 2: A preview of the all_sessions.csv

But...

There was some modification.



The image shows a snippet of a data table. The first row has a header 'fullVisitorID' followed by a series of red triangles. The next two rows contain large scientific notation numbers. A red rectangular box highlights the two rows of numbers.

fullVisitorID
2.81772249655118E+018
6.93896048645247E+018

Figure 3: Zoomed in on four hours of my life. Wasted.

Cleaning

Next, I cleaned the raw data in all .csvs with the exception of analytics.csv, while endeavouring to minimize information loss, creating appropriate relationships between the tables for the purpose of normalizing the database.

Cleaning

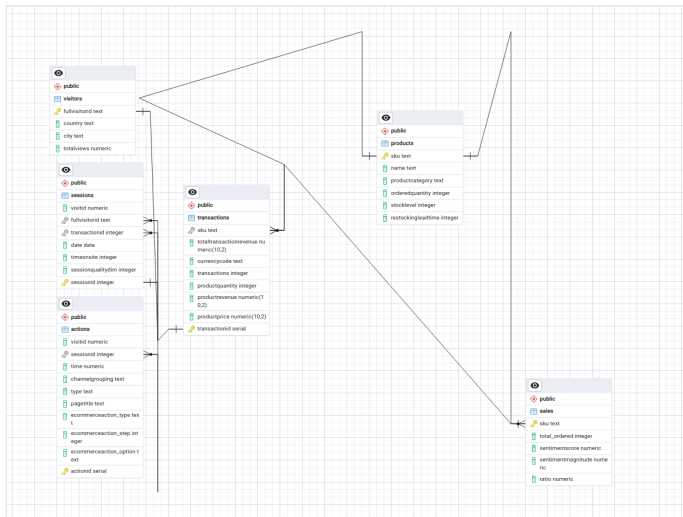


Figure 4: Does anyone know how to clean up these relationship arrows?

Import analytics.csv

Lastly, I split analytics up between the tables, while taking care not to add unnecessary data that was not sufficiently linked to a visitor/product/location.

```
CREATE TEMPORARY TABLE analytics_temp AS (  
    SELECT *  
    FROM analytics  
    WHERE fullvisitorid IN  
    (SELECT DISTINCT fullvisitorid  
    FROM analytics  
    JOIN visitors USING(fullvisitorid))  
);  
  
DROP table analytics;  
  
CREATE TABLE analytics AS (  
    SELECT DISTINCT * FROM analytics_temp  
);
```

Figure 5: Query to remove rows in analytics that don't match a visitor.

Interesting (and concerning) results

- ▶ A large proportion of all revenue earned \$1,145,722.53 (36.851%) came from one city with a population of 82,376: Mountain View, California.
- ▶ After the United States which makes up 99.487% of total revenue, the country that provided the most revenue was Czechia.
- ▶ Visitors that made purchases had 75% more page views than the average.
- ▶ Referrals were 42.5% more likely to be converted into sales than the next most likely channel, which was direct links.

What do my results mean?

- ▶ It was difficult to develop a non-destructive import pipeline
- ▶ It was difficult to normalize the database in a way that made sense and did not lose information.
- ▶ It was difficult to reconcile contradictory data between `all_sessions` and analytics.
- ▶ Analysis was limited by the lack of interpretable time data.
- ▶ Or in summary...

But at least...

I made it to the end goal of the project:

```
DROP TABLE analytics;
```

Thank you!