Microsoft

Using Postgres to locate best coffee near you! ☕📍

Varun Dhawan | @iVarund

Principal Product Manager @ Azure Postgres

About me

→ Principal Product Manager, Azure Postgres

- 20+ years in relational database systems (Postgres, Oracle, SQL)
- Previously: DevOps-Lead @Target, DBA @McKinsey&Co.
- Based in Minnesota, I enjoy hiking and blogging about tech
- Blog: **data-nerd.blog**

Varun Dhawan

→ Find me on social

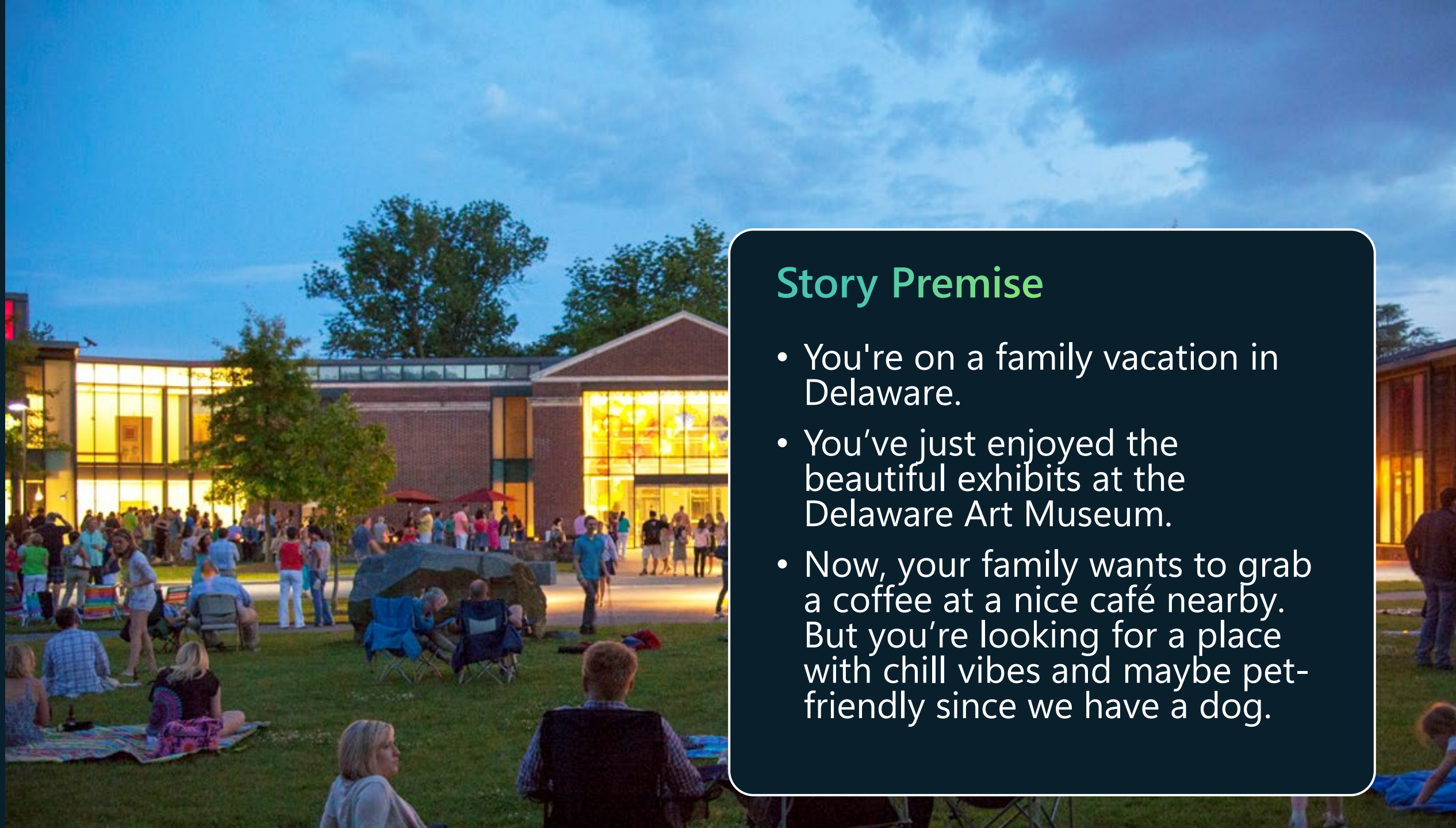
linkedin.com/in/varundhawan

@iVarund

varun-dhawan

Agenda – Find coffee!

1. Café Dataset
2. Search Pattern ‘ILIKE’
3. Search Phrase Full-Text
4. Search context – Semantic search
5. RAG (Retrieval Augmented Generation)



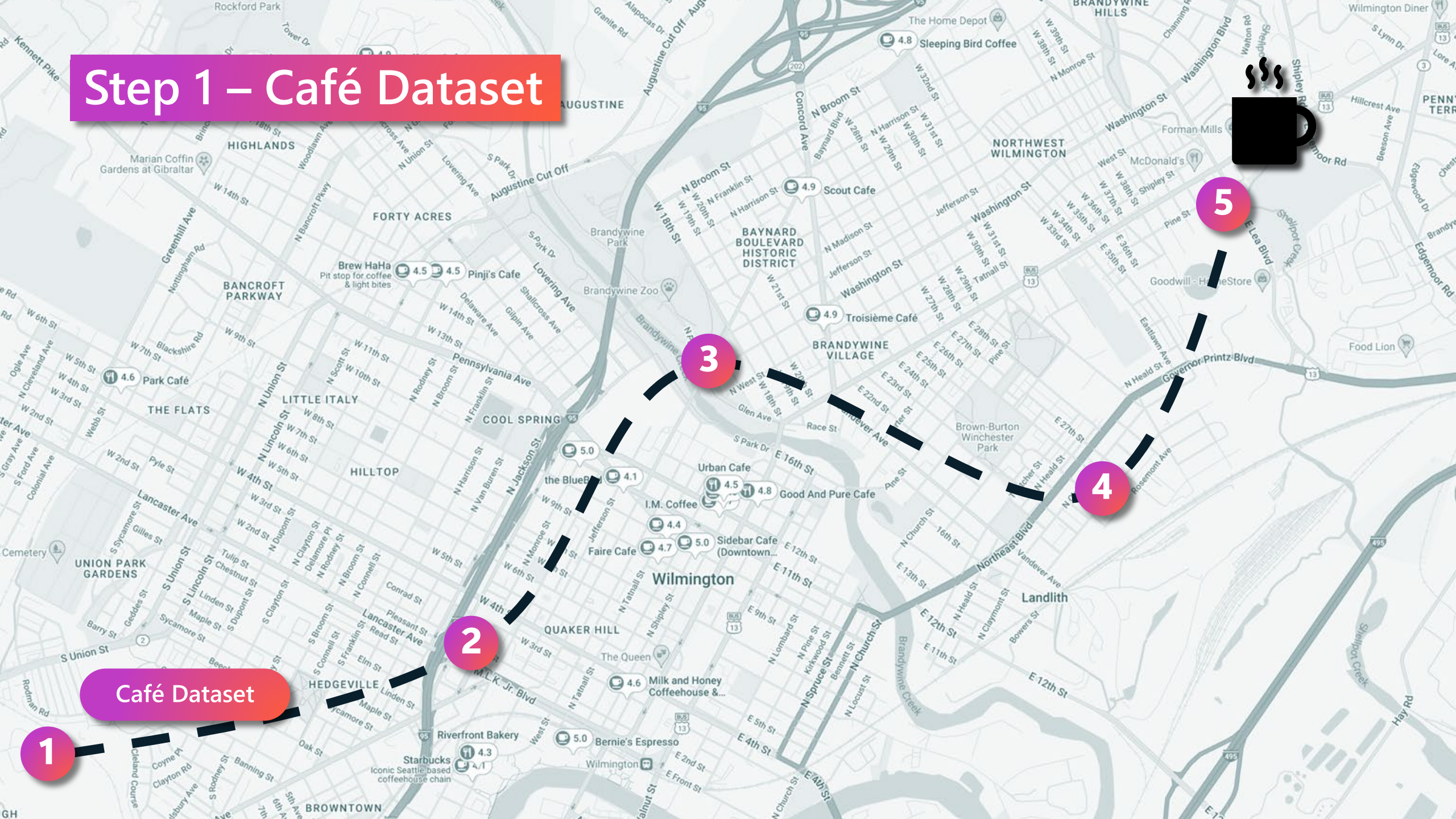
Story Premise

- You're on a family vacation in Delaware.
- You've just enjoyed the beautiful exhibits at the Delaware Art Museum.
- Now, your family wants to grab a coffee at a nice café nearby. But you're looking for a place with chill vibes and maybe pet-friendly since we have a dog.

How can Postgres help?

- With PostgreSQL's robust geospatial and vector capabilities, combined with RAG implementation, we can find the perfect coffee spot near you.
- Let's walk through the journey of turning basic text search into a powerful, context-aware search solution.

Step 1 – Café Dataset



Café Dataset yelp

For this demo, we will use Yelp Open Dataset

6,990,280 reviews

150,346 businesses

200,100 pictures

11 metro areas

https://www.yelp.com/dataset

yelp Dataset

business.json

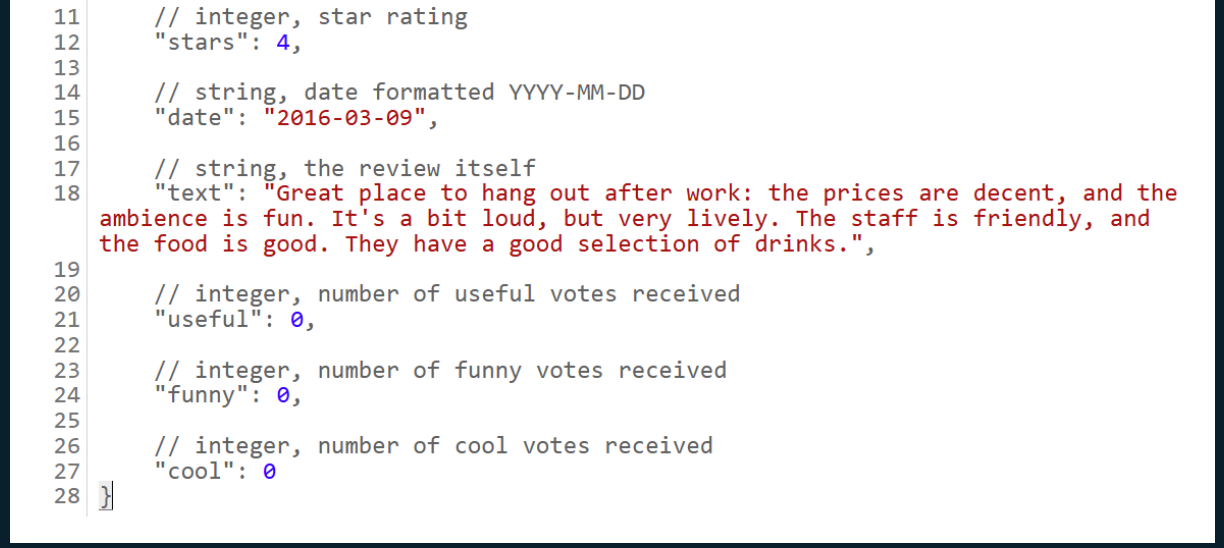
Data including location data, attributes, and categories

```
{
    // string, 22 character unique string business id
    "business_id": "tnhfDv5Il8EaGSXZGiuQGg",

    // string, the business's name
    "name": "Garaje",

    // string, the full address of the business
    "address": "475 3rd St",

    // string, the city
    "city": "San Francisco",

    // string, 2 character state code, if applicable
    "state": "CA",

    // string, the postal code
    "postal code": "94107",

    // float, latitude
    "latitude": 37.7817529521,

    // float, longitude
    "longitude": -122.39612197,

    // float, star rating, rounded to half-stars
    "stars": 4.5,

    // integer, number of reviews
    "review_count": 1198,

    // integer, 0 or 1 for closed or open, respectively
    "is_open": 1,

    // object, business attributes to values. note: some attribute values might be objects
    "attributes": {
        "RestaurantsTakeOut": true,
        "BusinessParking": {
            "garage": false,
            "street": true,
            "validated": false,
            "lot": false,
            "valet": false
        },
    },

    // an array of strings of business categories
    "categories": [
        "Mexican",
        "Burgers",
        "Gastropubs"
    ],

    // an object of key day to value hours, hours are using a 24hr clock
    "hours": {
        "Monday": "10:00-21:00",
        "Tuesday": "10:00-21:00",
        "Friday": "10:00-21:00",
        "Wednesday": "10:00-21:00",
        "Thursday": "10:00-21:00",
        "Sunday": "11:00-18:00",
        "Saturday": "10:00-21:00"
    }
}
```

yelp Dataset

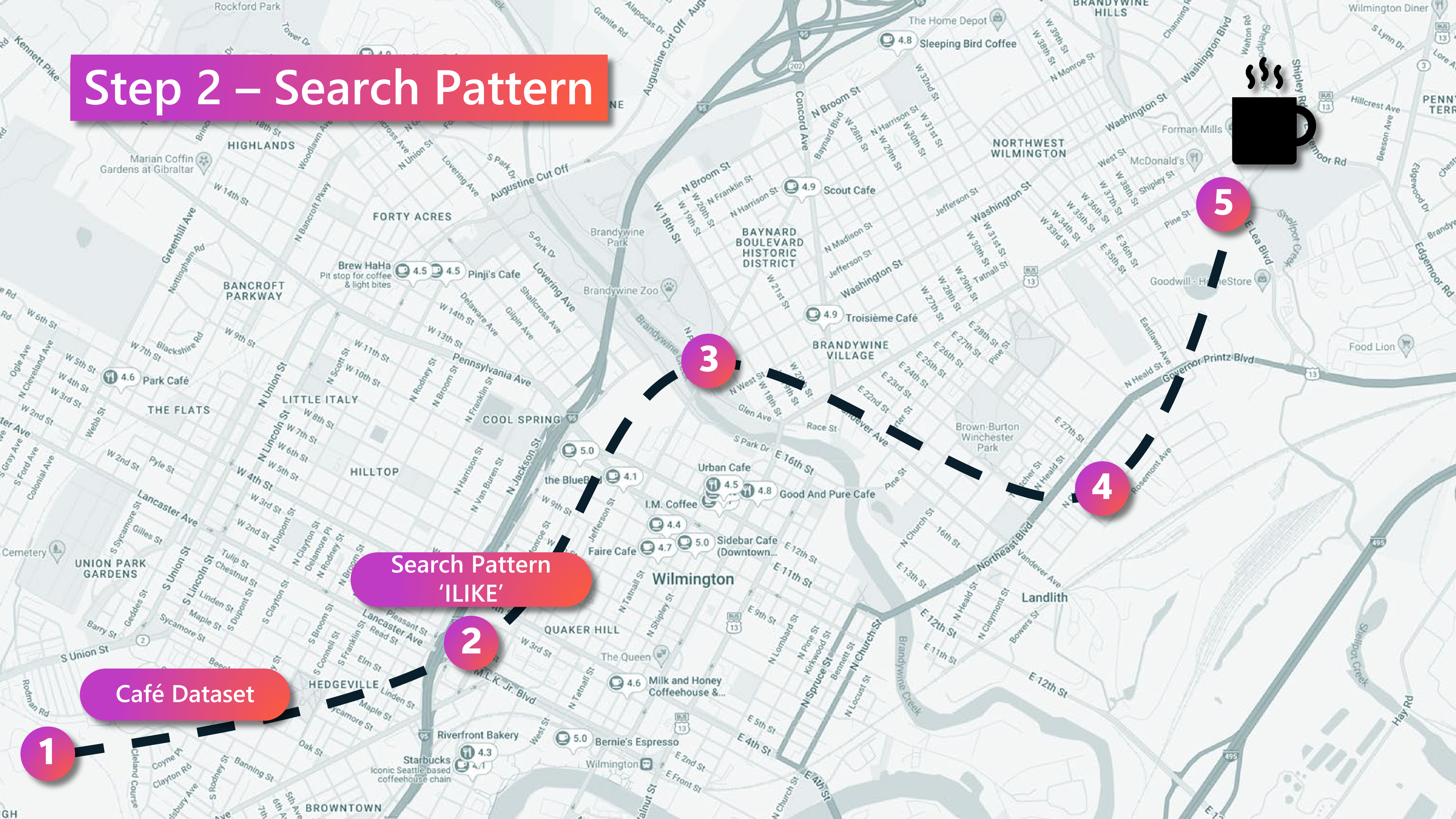
review.json

Review text including the details of business the review is written for.



```
{
    // string, 22 character unique review id
    "review_id": "zdSx_SD6obEhz9VrW9uAWA",

    // string, 22 character unique user id, maps to the user in user.json
    "user_id": "Ha3iJu77CxlrFm-vQRs_8g",

    // string, 22 character business id, maps to business in business.json
    "business_id": "tnhfDv5Il8EaGSXZGiuQGg",

    // integer, star rating
    "stars": 4,

    // string, date formatted YYYY-MM-DD
    "date": "2016-03-09",

    // string, the review itself
    "text": "Great place to hang out after work: the prices are decent, and the ambience is fun. It's a bit loud, but very lively. The staff is friendly, and the food is good. They have a good selection of drinks.",

    // integer, number of useful votes received
    "useful": 0,

    // integer, number of funny votes received
    "funny": 0,

    // integer, number of cool votes received
    "cool": 0
}
```

Step 2 – Search Pattern



Search Pattern

- Let's start by using Postgres's "ILIKE" for pattern matching.
- Search reviews to see if they mention words like '%restaurant%' and '%cafe%'.

```
WHERE r.text ILIKE '%restaurant%'
-- Searching for reviews mentioning 'restaurant'.
AND r.text ILIKE '%cafe%'
-- And those mentioning 'cafe'.
```

```sql
-- Postgres Coffee Finder
/*********************************************************
-- PART A >> TEXT MATCH USING ILIKE
-- SUBTITLE: Simple Pattern Matching in Reviews
-- DESCRIPTION: This part demonstrates using ILIKE for text matching to filter review data.
*********************************************************/
-- SHOW server_version;
-- SHOW azure.extensions;

--Sample records
SELECT * FROM review_de r LIMIT 5;

-- Now, let's search for mentions of 'restaurant' and 'cafe' in the reviews using ILIKE.
SELECT
    b.name AS business_name, r.stars,  r.text, b.city, b.address
FROM review_de r
JOIN business b ON r.business_id = b.business_id
WHERE
    r.text ILIKE '%restaurant%'  -- Searching for reviews mentioning 'restaurant'.
    OR r.text ILIKE '%cafe%'    -- And also those mentioning 'cafe'.
    AND b.city = 'Wilmington'
ORDER BY b.stars DESC;

-- Let's compare this pattern matching with a semantic search query for "cafe in Wilmington with chill vibes".
SELECT r.review_id, r.business_id, r.stars,
    r."date", r.text, b.name AS business_name, b.city
FROM review_de r
JOIN business b ON r.business_id = b.business_id
WHERE
    r.text ILIKE '%cafe with chill vibes%'; -- Search for cafes in review text

-- As we see, pattern matching can find results based on exact words, but it doesn't handle phrases or understand context well.
```

review_de(+) 1

SELECT r.review_id, r.business_id, r.stars, r."date", r.text, b.name AS business_name, b.city FROM review_de r JOIN busi

Enter a SQL expression to filter results (use Ctrl+Space)

review_id	business_id	stars	date	text	business_name	city

Value

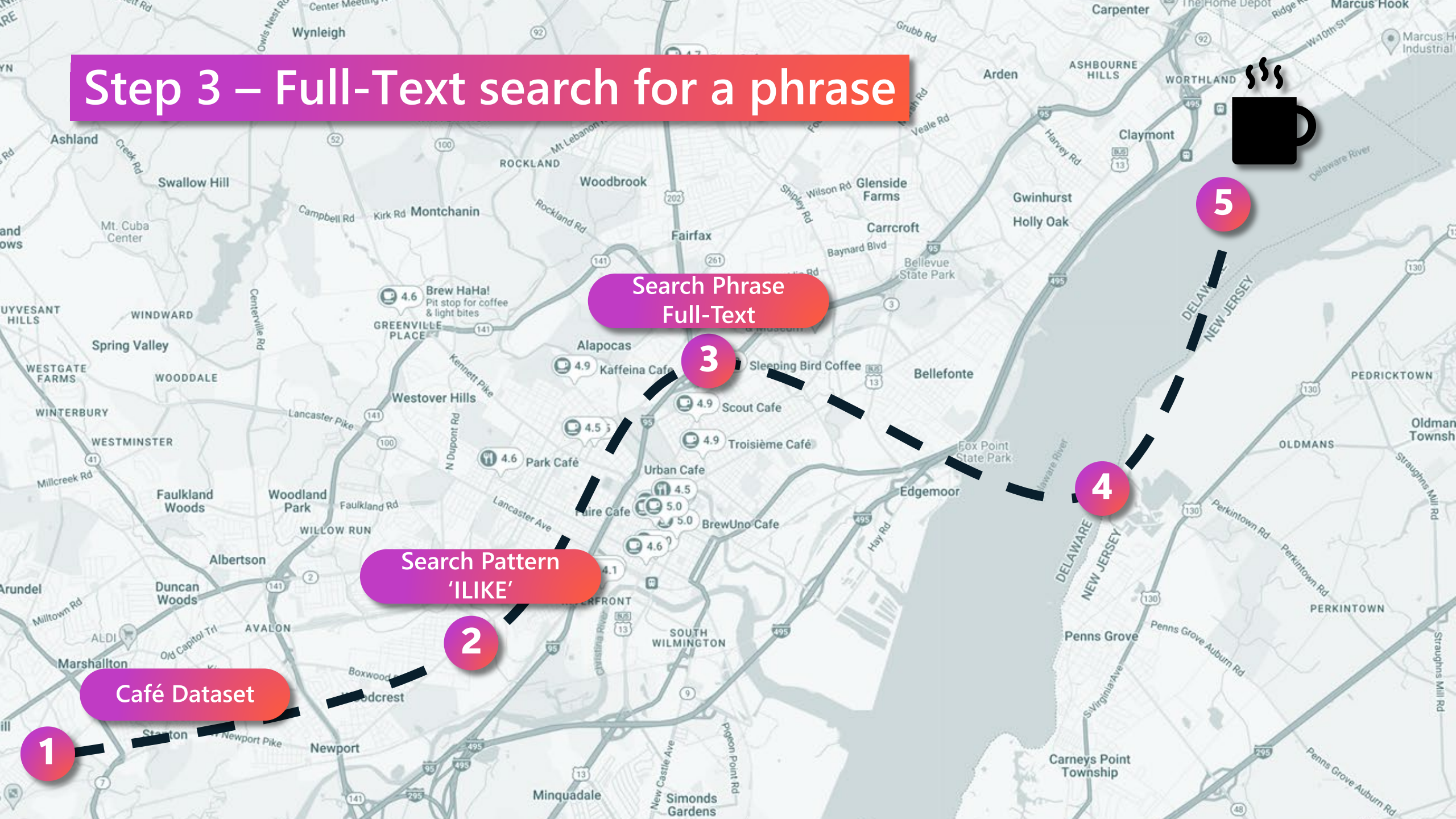
Close

Save | Cancel | Script | 200 | 0 | Rows: 0 | No data- 402ms, on 2024-10-17 at 16:13:46

Search Pattern

This approach is simple but has limitations when it comes to complex phrases or contextual search.

```
WHERE r.text ILIKE '%cafe with
chill vibes%’;

-- Search for cafes in review text
```


Step 3 – Full-Text search for a phrase
Café Dataset
Search Pattern
'ILIKE'
Search Phrase
Full-Text
1
2
3
4
5
Brew HaHa!
Pit stop for coffee
& light bites
Kaffeina Cafe
Sleeping Bird Coffee
Scout Cafe
Troisième Café
Park Café
Urban Cafe
BrewUno Cafe
Wynleigh
Ashland
Swallow Hill
Mt. Cuba
Center
Montchanin
ROCKLAND
Woodbrook
Fairfax
Glenside
Farms
Carrcroft
Arden
ASHBOURNE
HILLS
WORTHLAND
Claymont
Gwinhurst
Holly Oak
Bellevue
State Park
Bellefonte
Fox Point
State Park
Edgemoor
Marcus Hook
Carpenter
WINDWARD
Spring Valley
WESTGATE
FARMS
WOODDALE
WINTERBURY
WESTMINSTER
GREENVILLE
PLACE
Westover Hills
Faulkland
Woods
Woodland
Park
WILLOW RUN
Albertson
Duncan
Woods
AVALON
Marshallton
ALDI
SOUTH
WILMINGTON
Minquadale
Simonds
Gardens
Newport
Carneys Point
Township
Penns Grove
PERKINTOWN
PEDRICKTOWN
OLDMANS
Oldman
Townsh
DELAWARE
NEW JERSEY

Search Phrase – FULL TEXT

- Next, we use PostgreSQL's FULL TEXT search capabilities.
- Full-text search helps us find phrases like 'restaurant' or 'cafe in Wilmington'.

```
WHERE r.text_search @@
to_tsquery('english', 'restaurant |
cafe')
-- Search for "restaurant" OR "cafe"
```

```sql
-- Postgres Coffee Finder
/*********************************************************
-- PART B >> FULL TEXT SEARCH
-- SUBTITLE: Advanced Text Search Capabilities
-- DESCRIPTION: This section introduces full-text search capabilities using PostgreSQL's text search features.
*********************************************************/
-- SHOW server_version;
-- SHOW azure.extensions;

--  Add a `text_search` column of type `tsvector` for full-text search
-- `tsvector` data type is designed to handle full-text search
ALTER TABLE review_de
ADD COLUMN text_search tsvector;

-- Populate the `text_search` column using existing review texts.
-- We use the `to_tsvector()` function which converts the review text into a tsvector type
UPDATE review_de
SET text_search = to_tsvector('english', coalesce(text, ''));

-- Create a GIN (Generalized Inverted Index) index for efficient search
-- GIN is optimized for fast full-text search
CREATE INDEX idx_review_de_text_search ON review_de USING GIN (text_search);

-- Sample records
SELECT r.text, r.text_search FROM review_de r LIMIT 5;

-- FULL-TEXT SEARCH QUERY
-- This query performs a full-text search on reviews mentioning either 'restaurant' or 'cafe'.
SELECT
    b.name AS business_name, r.stars, r.text, b.city
FROM review_de r
JOIN business b ON r.business_id = b.business_id
WHERE
    r.text_search @@ to_tsquery('english', 'restaurant | cafe')
    -- Search for "restaurant" OR "cafe"
ORDER BY r."date" DESC;
```

Results 1

SHOW server_version | Enter a SQL expression to filter results (use Ctrl+Space)

	server_version
1	16.4

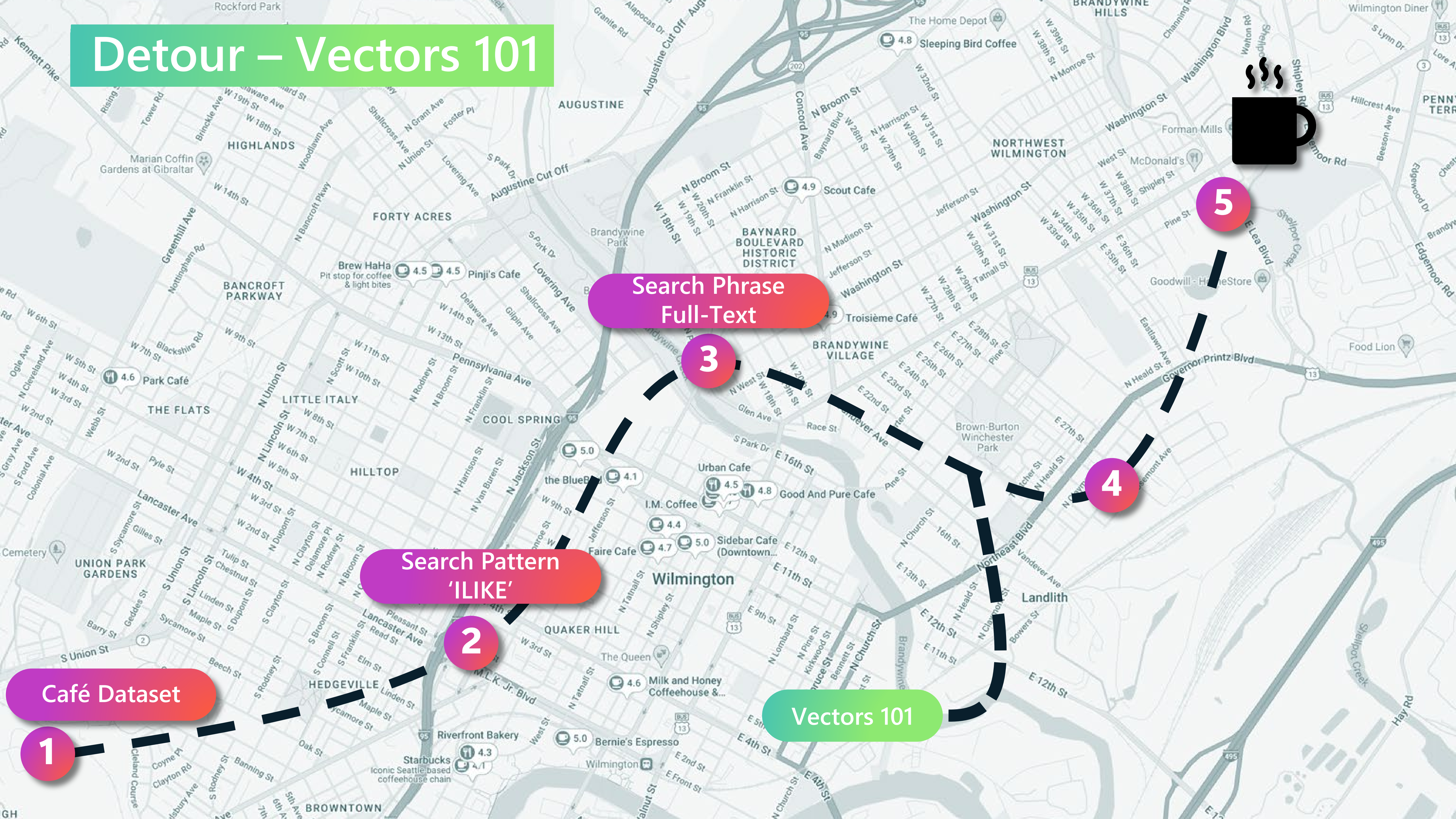
Save | Cancel | Script | 200 | 1 | Rows: 1 | 1 row(s) fetched - 37ms, on 2024-10-17 at 16:39:58

Search Phrase – FULL TEXT

This is better for phrase-based searches but struggles with nuanced queries like ‘cafe with chill vibes’.

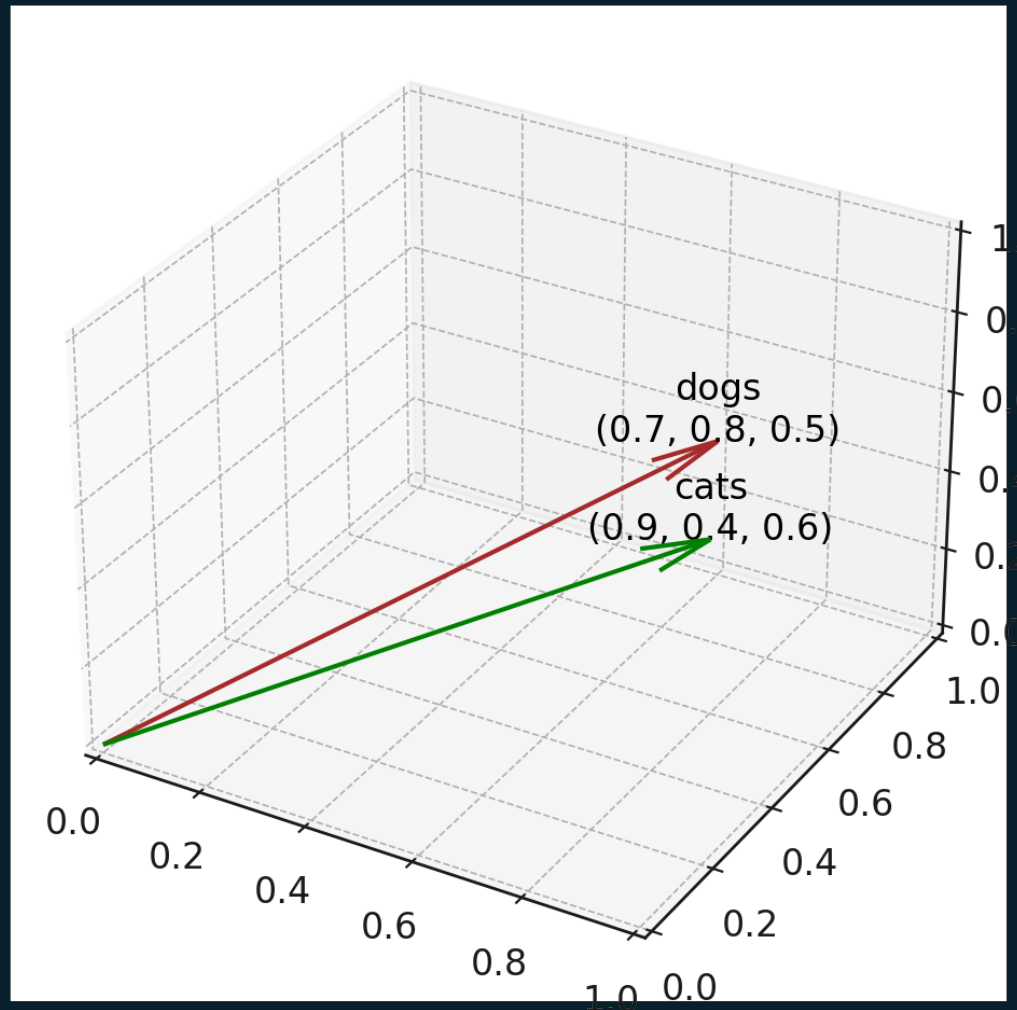
```
WHERE r.text_search @@
websearch_to_tsquery('english', 'cafe
with chill vibes’)
-- Attempting full-text search with a phrase
```

Detour – Vectors 101



Vector 101

- Lists of numbers that represent items in a high-dimensional space.
- For example, a vector representing the string “**dogs**" might be [0.7, 0.8, 0.5].
- Each number in the vector is a dimension of the space.



Generating vectors

Use a model to generate vectors for items:

Input	→	Model	→	Vector
"dog"		word2vec		[0.017198, -0.007493, -0.057982, ..]
"cat"		word2vec		[0.004059, 0.06719, -0.093874, ...]

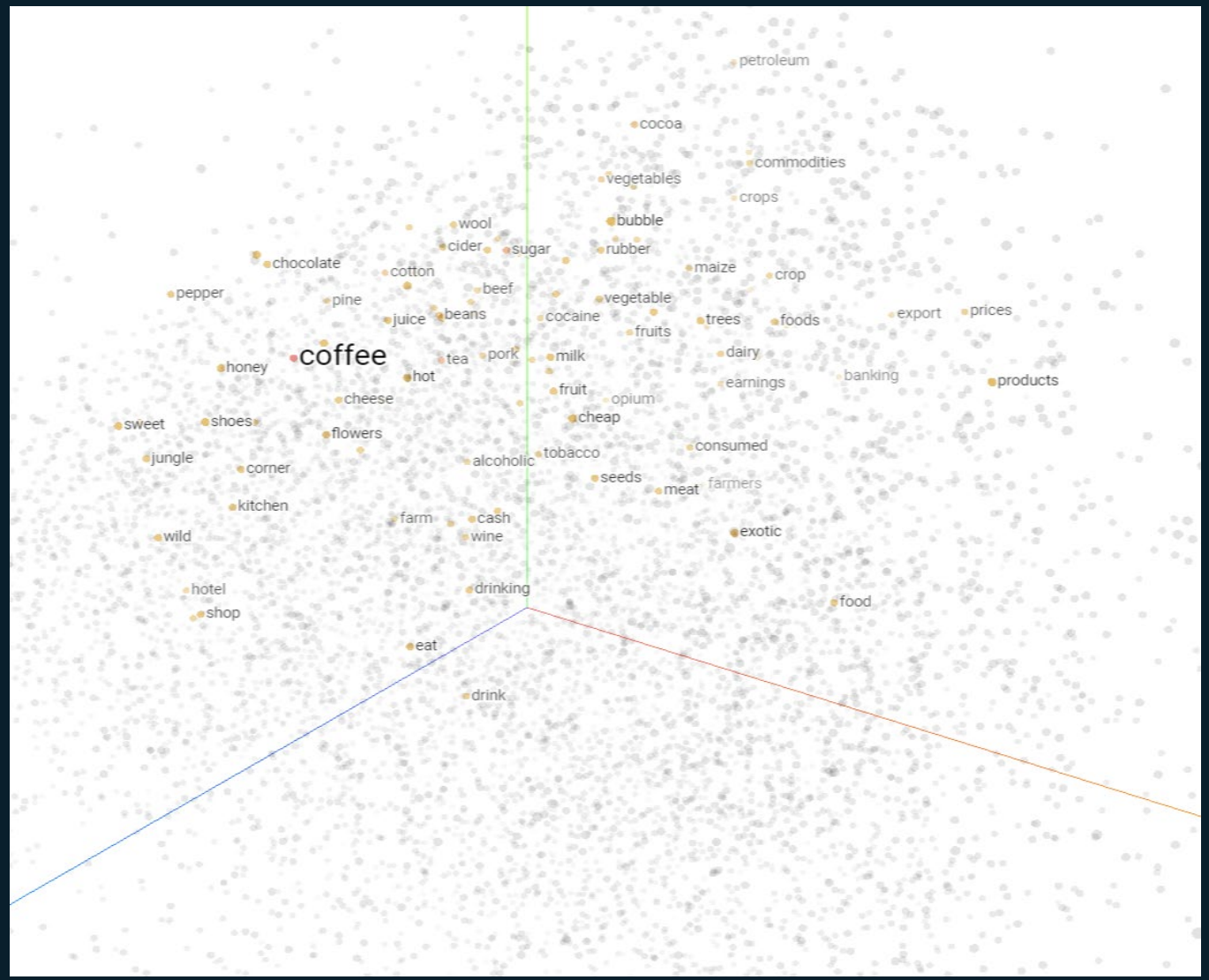
Model	Input types	Dimensions
Word2Vec	Word	50-300
OpenAI text-embedding-ada-002	Text	1536
OpenAI text-embedding-3	Text	256-3072
Azure Computer Vision Multi-modal	Text or Image	1024

Popular models (find more on HuggingFace):

So why should we care about vectors embeddings?

Similarity

Find similar items in a large dataset, useful for recommendations



So why should we care about vectors embeddings?

Search

Search other items that are similar to what you're querying.

Example

Generate Vector
https://pamelafox.github.io/vectors-comparison/

What is a vector?

Expore words from a dataset of 1000 words across two embedding models.

Target word: dog Embedding model: Both (Comparison) Find word

Model: word2vec

Vector: 300 dimensions

0.017198, -0.007493, -0.057982, 0.054051, -0.028336, 0.019245, 0.019655, -0.027681, -0.005159, -0.021293, 0.060275, -0.142171, -0.007575, -0.055689, -0.008435, 0.036034, -0.066827, 0.053396, -0.062896, -0.040293, 0.052086, -0.03325, 0.047827, -0.055034, -0.029974, 0.067154, -0.05012, 0.107447, 0.110068, 0.00819, -0.032594, -0.027517, -0.012202, -0.028827, -0.033086, 0.00261, -0.004504, 0.017689, 0.049792, 0.112033, 0.005569, -0.071413, -0.005057, 0.017608, -0.036034, -0.02981, 0.083533, -0.023586, -0.005364, 0.025388, -0.023586, 0.039965, 0.076982,

Most similar:

cat	0.7609456296774421
horse	0.482580559367262
child	0.3701001015211071
bear	0.3660915748726983
someone	0.36170237677870604
baby	0.3560092821041511
boy	0.35216872587817744
woman	0.3511048220342392
mother	0.3455034314869205
girl	0.3426251584138038

Least similar:

bank	-0.02625562901048338
meet	-0.026630362532046314
met	-0.02771119328935793
of	-0.02891628801968331
switzerland	-0.040093095982862106
present	-0.0425287520326544
If	-0.04463080257045229
In	-0.04993156762830111
worked	-0.05088302787727771
high	-0.051125786415643575

Similarity histogram:

Model: openai

Vector: 1536 dimensions

-0.0033353185281157494, -0.017689190804958344, -0.01590404286980629, -0.01751338131725788, -0.018054334446787834, 0.021841011941432953, -0.012313461862504482, -0.02273358590900898, -0.021286534145474434, -0.01814900152385235, 0.012252604588866234, 0.038759343326091766, 0.0015408731997013092, -0.00691406661644578, -0.013638799078762531, 0.024153590202331543, 0.03989534825086596, 0.0012036223197355866, 0.009372025728225708, -0.012178223580121994, -0.019853007048368454, 0.006024873349815607, 0.011319459415972233, -0.025167878717184067, -0.007593639660626665, 0.010284884832799435, 0.009831836447119713, -0.008492975495755672, -0.005639444105327129, -0.009446406736969948, 0.007444877177476883, -0.009277358651161194, -0.025285932644605 64, -0.02119186706840992, -0.005906539969146252, -0.018906336277723312, -0.007539544254541397, -0.016066329553723335, -0.01171841286122799, -0.02093491330742836, 0.004608250688761473, 0.011042220517992973, 0.011549364775419235, -0.009541073814034462, 0.0025864355266094208, 0.0026202453300356865, -0.0036007240414619446, -0.011995651759207249, -0.02549245022237301, -0.007958783768117428, 0.015701185911893845, 0.016188044100999832, -0.005825396627187729, -0.00866878591477871, -0.0003888105857186019 4, -0.0006356207886710763, 0.0074110678397119045, 0.00766802066937089, -0.005419681314378977, -0.007674783002585173, 0.0086823096498847, -0.004740108270198107, -0.01406479999423027, 0.02170577272772789, -0.0029955320060253143, -0.008574118837714195,

Most similar:

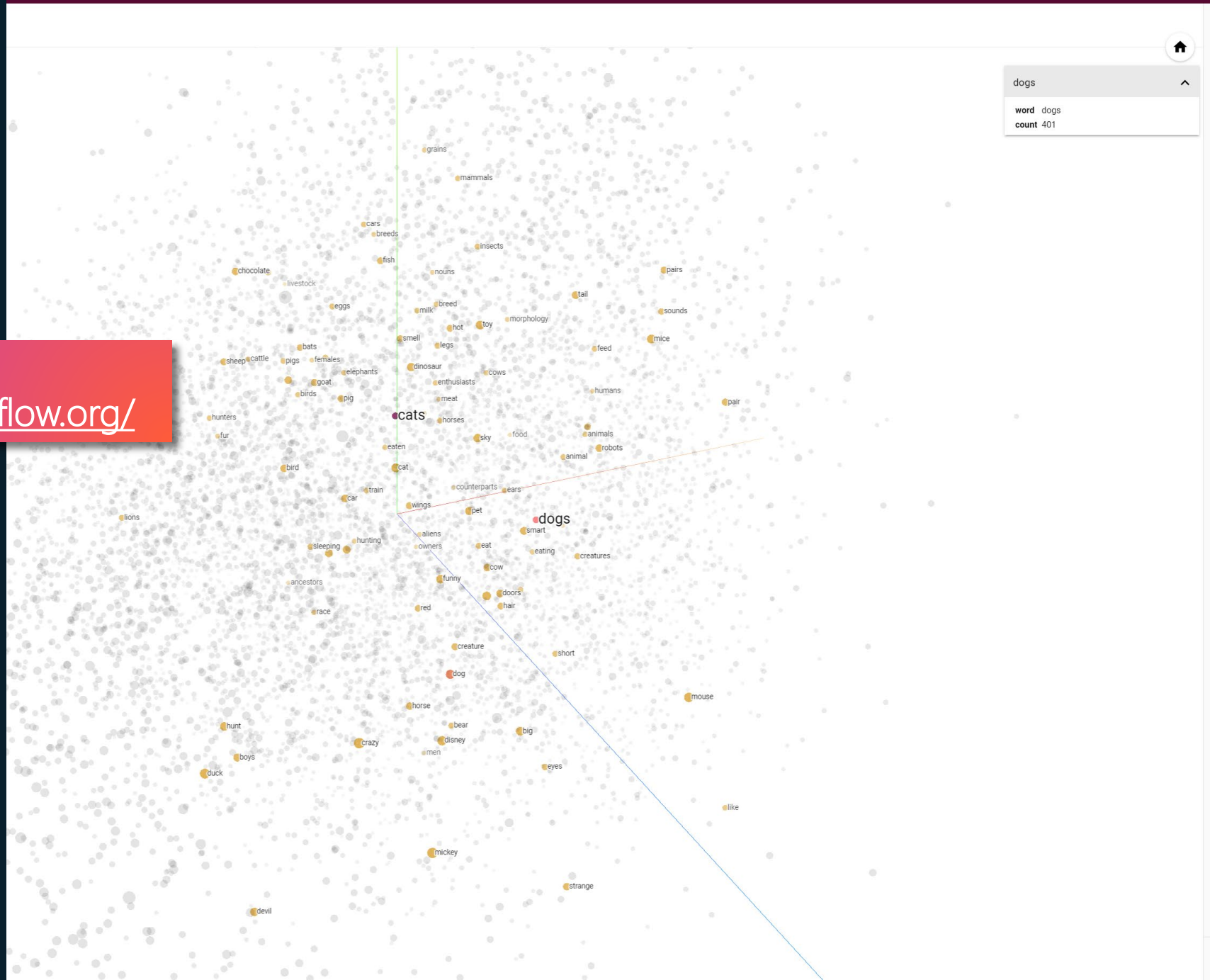
god	0.8661232217030437
cat	0.8635463285343138
kid	0.8633793412791264
boss	0.8616536488849736
fish	0.8567160061416755
do	0.8531014742976359
horse	0.8516590030182295
bear	0.8516394647209997
human	0.8500093809305883
gun	0.8492639208536553

Least similar:

catalonia	0.7746281384075008
anymore	0.7745343111964632
netherlands	0.7744193510029177
worse	0.774271453446651
shouldn	0.7741518238387108

Example:

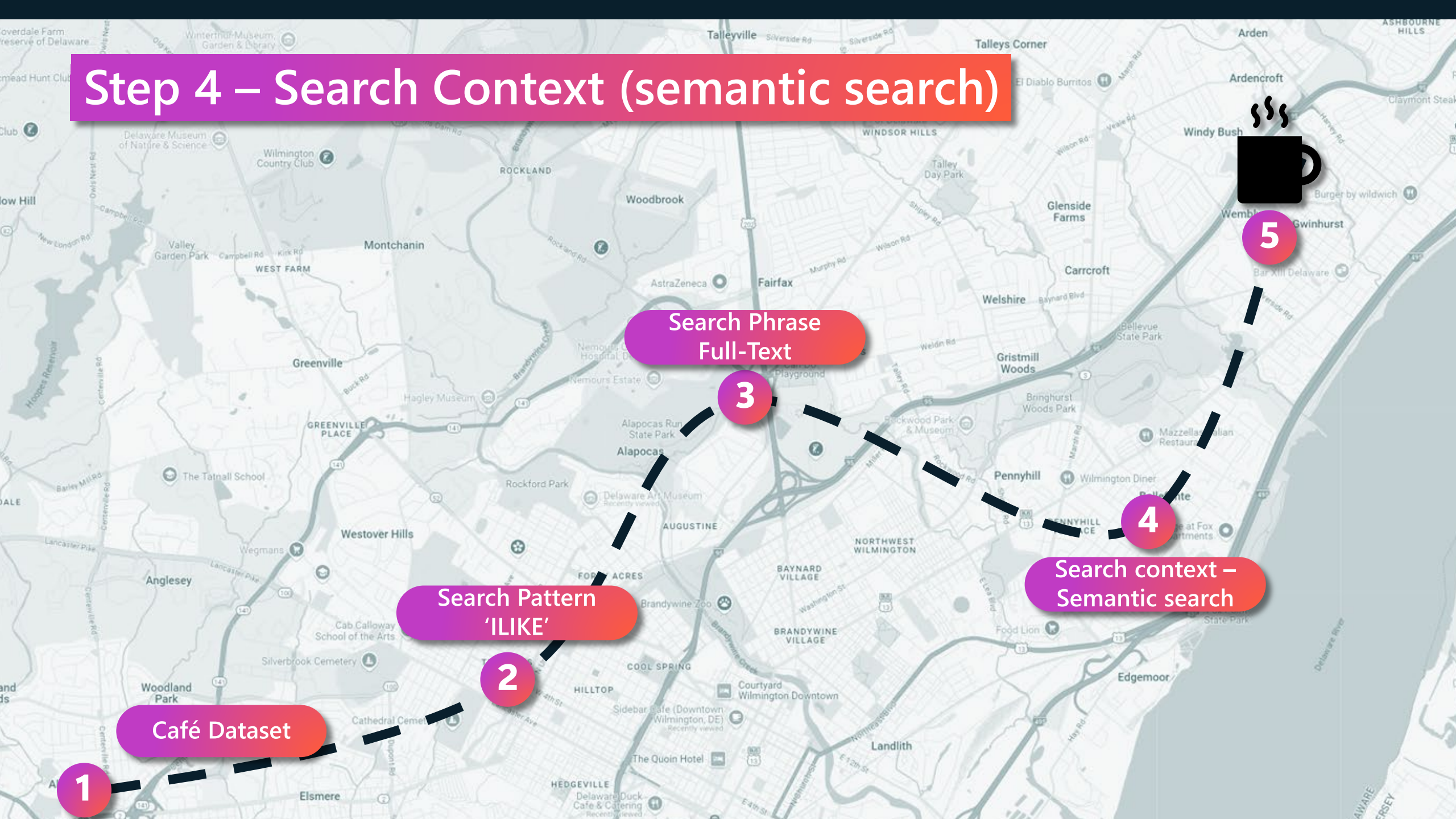
Visualize Vector
https://projector.tensorflow.org/



Storing vectors in Postgres table

business_name	description_vector
Border Cafe	[-0.028679002,-0.01258319,-0.009900554,-0.0068770316,0.00074013317,0.01134562,0.014057898,0.0
Cafe Verdi Restaurant	[-0.076620474,-0.029332073,-0.028847456,-0.048257638,0.0023896075,-0.029663652,-0.023644201,
Eeffoc Cafe	[-0.002310421,0.004936138,-0.034080025,-0.027507847,0.0045192465,0.03127739,-0.0007010963,0.0
Eeffoc Cafe	[-0.045164146,-0.014219385,-0.032155517,-0.05293553,0.03407021,-0.017274441,-0.010171788,0.02
Cafe Napoli Restaurant & Pizzeria	[-0.050793096,-0.008950297,-0.01286485,-0.029009983,0.010398233,-0.011077354,-0.02611411,0.00

Step 4 – Search Context (semantic search)



Vector Semantic Search

- Now, let's dive into semantic search using vector data
- With Azure OpenAI embeddings and PostgreSQL vectors, we can search reviews using natural language queries like: "cafe with chill vibes"

```
ORDER BY r.description_vector <=>
azure_openai.create_embeddings('t
ext-embedding-3-small','cafe with
chill vibes')::vector

-- Perform vector similarity search using Azure
OpenAI embeddings.

-- This searches for reviews "similar to" the
input phrase: 'cafe with chill vibes'.
```

Hybrid Search (vector search combined with geo-spatial data)

- But let's take it a step further by adding geospatial data to find results near the Art Museum"

- With the power of PostGIS extension, we can combine the semantic search results with spatial data, and ask "Find cafe with chill vibes, near Delaware Art Museum"

```
WHERE
ST_DWithin(b.business_location::geography,ST_GeographyFromText('POINT(-75.564039.7658)'), 5000)

-- Spatial filter: only include businesses within 5 km of the Delaware Art Museum.
```

```
ORDER BY  description_vector <=> azure_openai.create_embeddings('text-embedding-3-small', 'cafe near Delaware Art Museum with outdoor seating and pet friendly')::vector

-- Rank results by vector similarity to the given search query.
```

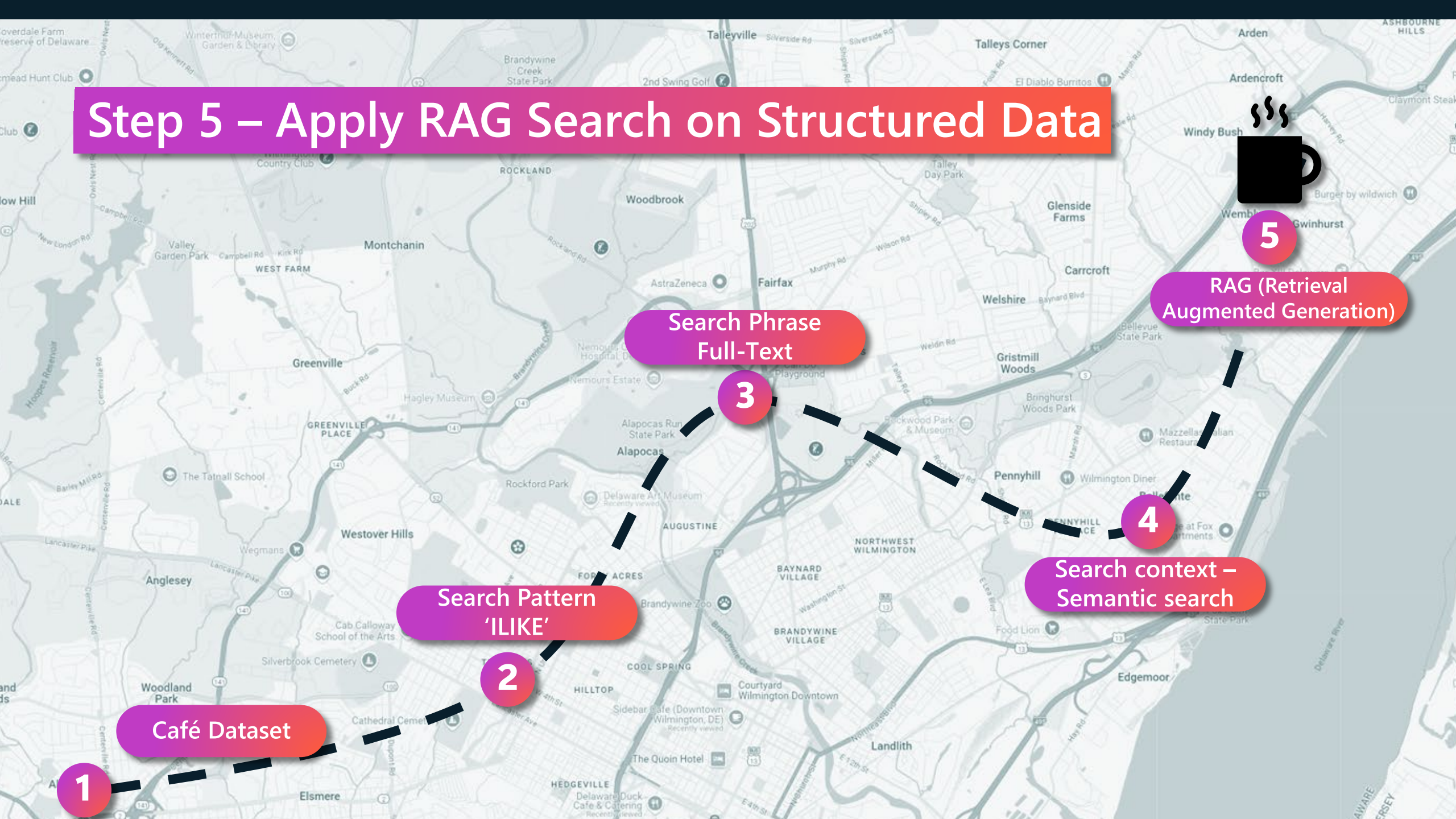
```sql
-- Postgres Coffee Finder
/*********************************************************
-- PART C >> VECTOR SIMILARITY AND HYBRID SEARCH
-- SUBTITLE: Combining Semantic and Spatial Searches
-- DESCRIPTION: This script sets up vector similarity search with Azure OpenAI embeddings and demonstrates both simple and hybrid semantic searches.
*********************************************************/
-- SHOW server_version;
-- SHOW azure.extensions;

-- List avilable extensions
SELECT pge.extname, pge.extversion FROM pg_extension pge
WHERE extname IN ('azure_ai','vector','postgis','pg_diskann');

-- Step 1: Add a new column for storing vector embeddings
-- Each vector will have 1536 dimensions and will represent the semantic meaning of the review and business data.
ALTER TABLE review_de
ADD COLUMN description_vector vector(1536);

-- Step 2: Populate the `description_vector` column with embeddings generated from the business name, review text, categories, and attributes.
-- This loop fetches reviews that are not yet embedded and generates embeddings using Azure OpenAI's embedding model.
DO $$
DECLARE
    counter integer := (SELECT COUNT(*) FROM review_de WHERE text <> '' AND description_vector IS NULL);
    r record;
BEGIN
    WHILE counter > 0 LOOP
        FOR r IN
            SELECT *
            FROM review_de de
            JOIN business b ON de.business_id = b.business_id
            WHERE de.text <> '' AND de.description_vector IS NULL
        LOOP
```

Results 1

SHOW server_version | Enter a SQL expression to filter results (use Ctrl+Space)

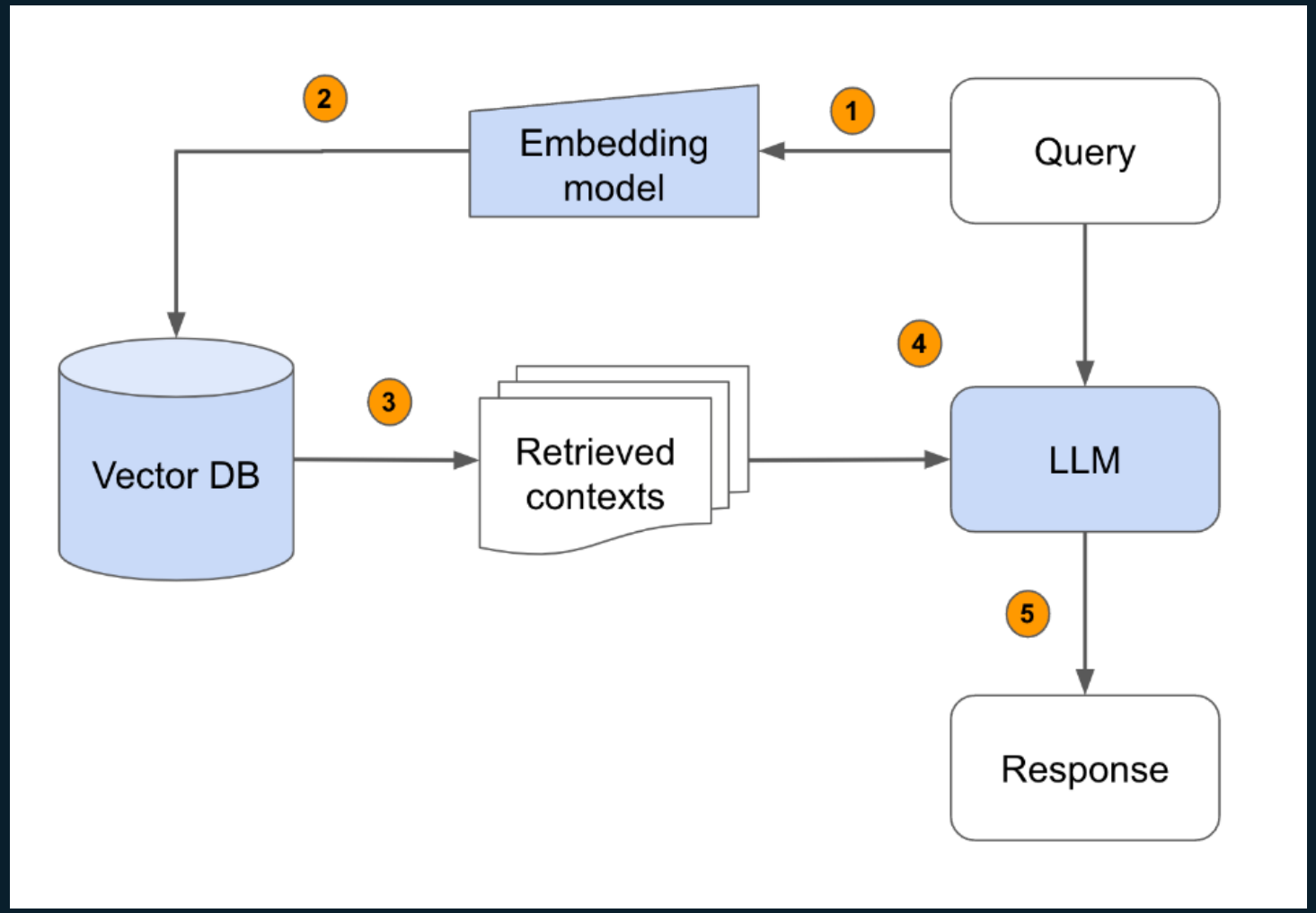
	server_version
1	16.4

Step 5 – Apply RAG Search on Structured Data



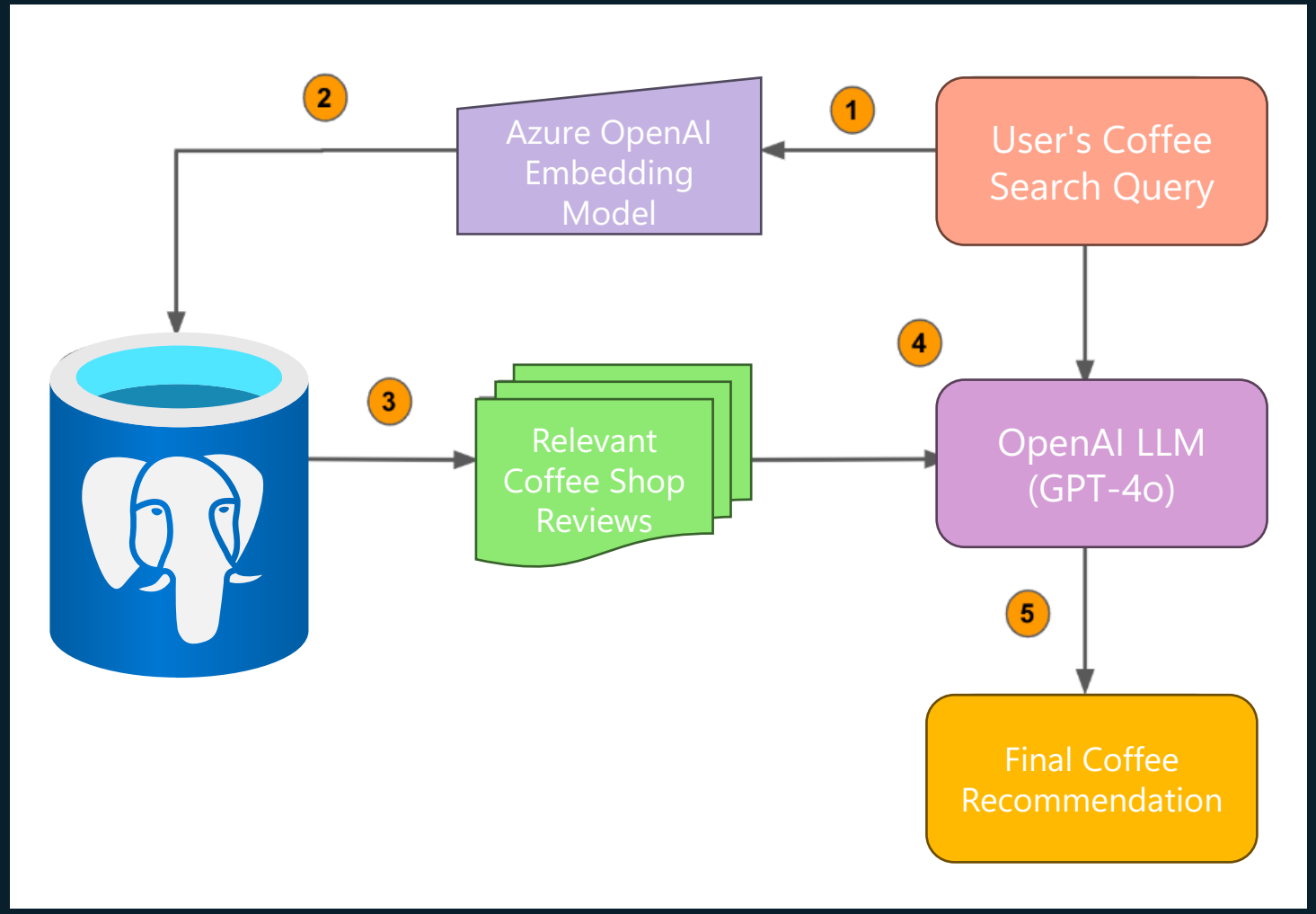
What is RAG?

RAG (Retrieval Augmented Generation) is the process of retrieving relevant contextual information from out vectorized dataset and passes this information to a large language model (LLM) to generate answers.



Apply RAG Search on Structured Data

So, Instead of asking LLM a question and hoping the answer lies somewhere in its *training-data*, we provide the context sourced from our "vectorized" dataset. LLM then references this context outside of its training data and generate more precise answers.



DEMO - Step 5

- Walk through the Coffee Finder (Python) app demo.
- Show how RAG combines all the elements: pattern matching, full-text search, vector search, and geospatial data.

In

No Index

Price Range

0 70

0 100

Check-in Date

2017/01/02

What are you looking for?

coffee within 5 mins walk and pet friendly a

Search for listings

Docs

Blog

GitHub

Seattle Airbnb Rentals Listing

Search SEA Rentals! The Streamlit app uses cosine similarity to semantically match your query with Airbnb listings and find matching properties in our database

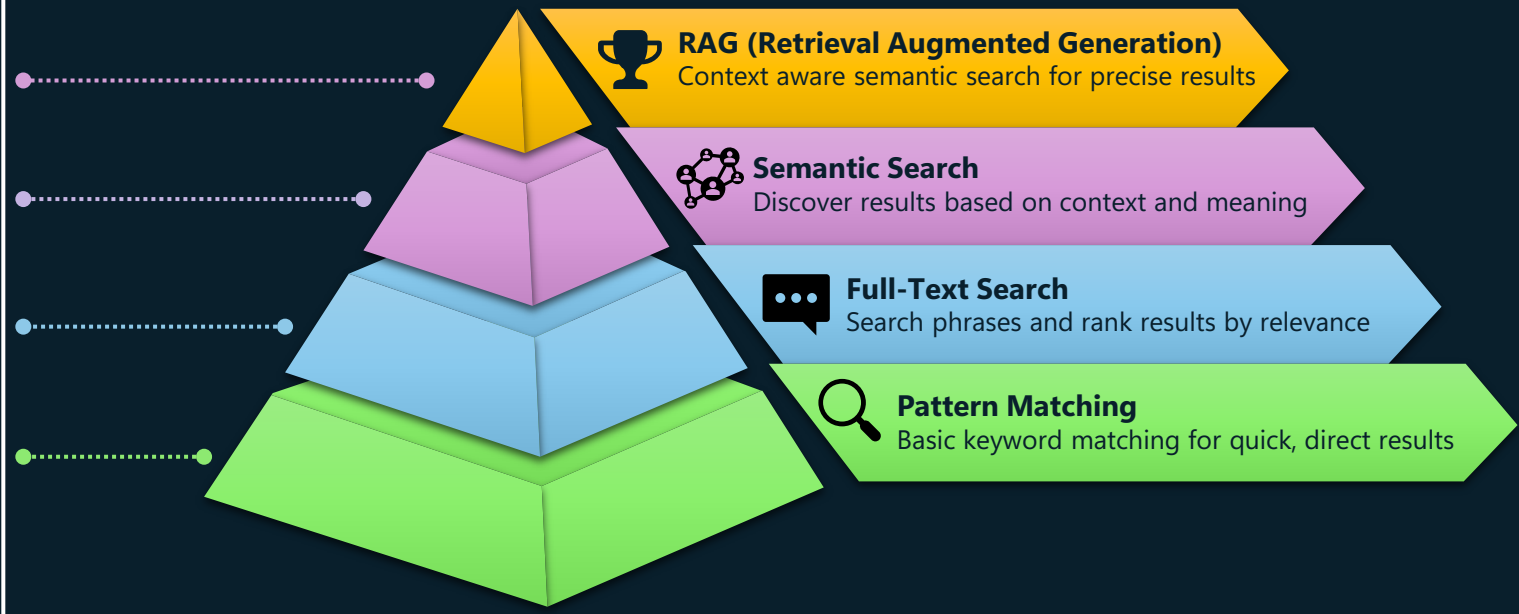
Found 10 listings.

Query time: 0.21 seconds

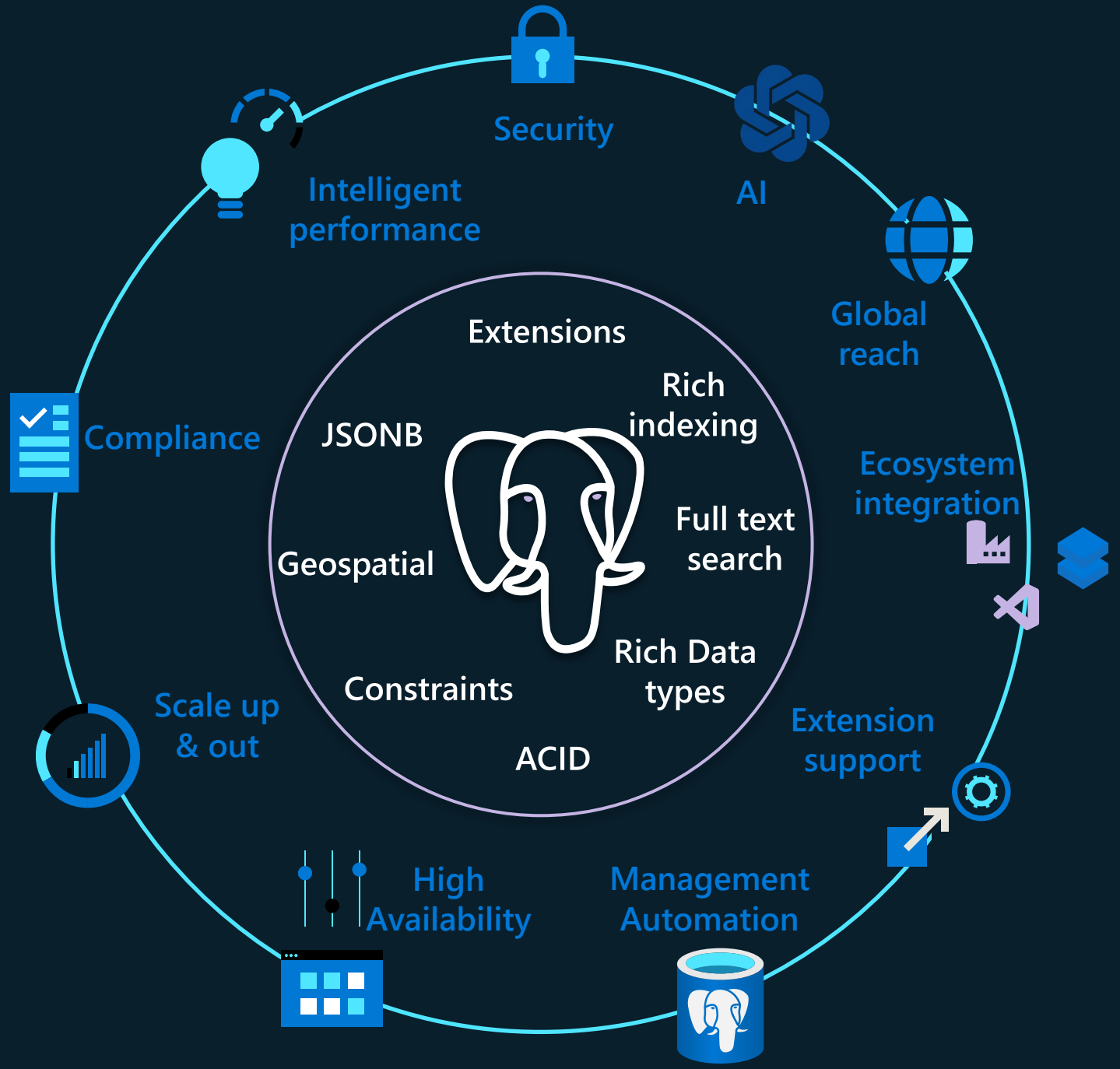
	Listing ID	Name	Price	Date	Summary	Description
0	5002964	10 Minutes from Downtown Seattle	50	2017-01-02	Easy access from the SEA-TAC. 10 minutes from Downtown Seattle, Ballard, Fremont, the Water Front and Discovery Park. Quiet apartment with a private bedroom and full bathroom. Continental breakfast provided. Reserved parking included. Upper floor balcony where you can enjoy squirrel watching with your morning coffee, this is Seattle afterall. Shared Living room, kitchen, and balcony. Coffee & Tea station provided. We will be available via txt once you check in for any questions you may have. Red Mill Burgers and QFC (grocery) are two short blocks away. There is also 9 hole golf course and mini golf just down road. Major bus line to Downtown Seattle, Space Needle, Ballard and Fremont is one block away. We have a Cat and a small Dog professionally trained.	Easy access from the SEA-TAC. 10 minutes from Downtown Seattle, Ballard, Fremont, the Water Front and Discovery Park. Quiet apartment with a private bedroom and full bathroom. Continental breakfast provided. Reserved parking included.
1	6701018	Seattle Pet/Family Friendly Living	67	2017-01-02	Diverse city location, conveniently 1block to bus line, nearby grocery/coffee shop. Easy access to I-5 & shopping! Furnishings throughout, detailed cleaning for your comfort.Owner on premise & available!We love letting well behaved/clean pets \stay\! The entire space is for your privacy, a dedicated parking spot for your vehicle. Leave your car and take the bus if you choose. * We hire professional staffing to clean the unit after each guest visits, every area and place the guest touches is sanitized and cleaned for the safety and cleanliness and peace of mind when you stay with us. All sheets and linens are laundered each and every time a guest leaves. The Oaktree Suite is on the ground level perfect for your cute little doggie to roam outside the enclosed patio area. The kitchen has full appliances for your stay and have a Blender, Juicer, George Foreman Grill, salad spinner, toaster, microwave, coffee maker, rice cooker, lots of pots and pans and cooking utensils, pizza cutter, kni	Diverse city location, conveniently 1block to bus line, nearby grocery/coffee shop. Easy access to I-5 & shopping! Furnishings throughout, detailed cleaning for your comfort.Owner on premise & available!We love letting well behaved/clean pets \stay\!
2	5020861	cozy balcony apt one block to UW	65	2017-01-02	This is a one bedroom apartment one block from the University of Washington. Its a quiet corner apartment with a balcony with coffee and modest breakfast. I sleep on the pull out couch and I have a small friendly dog. Close to buses going downtown. Proximity to the university is great. Walk one block (which takes me about 1 minute) and you are on the northeast end of campus, which is convenient to the Burke Museum and Paccar. There is laundry on my floor, 1.75 to wash and 1.75 to dry Please note I have a small dog. Guests have access to the entire apartment. I go to bed around 10 pm so cable would not be available at that time since the television is located in the living room. On nights where the price is listed as 100$ it is likely that I will not be in town for most of or all of the visit so the access changes to the entire apartment. Breakfast options include coffee, tea, cereal, toast with jam and cream cheese, instant grits and oatmeal. I am very busy and keep to myself but am	This is a one bedroom apartment one block from the University of Washington. Its a quiet corner apartment with a balcony with coffee and modest breakfast. I sleep on the pull out couch and I have a small friendly dog. Close to buses going downtown.
3	6130287	Great Location, Friendly and Clean!	65	2017-01-02	Come stay in Ballard! A great neighborhood with lots to do. Surrounded by great N.W. cuisine and breweries. I'm near the Burke Gilman Trail and bus stop which can take you to adjoining neighborhood's like University District, Fremont, Green Lake. My apartment is a big 1 bedroom apartment (750 sqft) that I share with my 2 lovely animals (1 dog and 1 cat.) Decorated in a little bit of a Latin flare and vintage, I like antiquing. So if you are looking for a modern flare, I'm sorry but this is not the place, but very cute and clean! The building was built in 1969 for the world fair, so it's a little vintage too! You have access to everything, but will be sharing common space with me and my pet's. I will try to stay out of way as much as possible or as much as you want. Again my is pretty large. I will interact with you ask much as you want or don't want. I am VERY social and can tell you where to get the best bites and libations. Just let me know!! You are sharing the apartment with me	Come stay in Ballard! A great neighborhood with lots to do. Surrounded by great N.W. cuisine and breweries. I'm near the Burke Gilman Trail and bus stop which can take you to adjoining neighborhood's like University District, Fremont, Green Lake.
4	47156	Suite + Free Parking	65	01-02	ars. Private bed & bath on separate floor, feels like your own flat. Free street parking & private patio. Easy, ds. Park, walk ,or bus easily to all parts of the city. **Free Street Parking**Free WiFi**Your Own Private Access**Pet Free** Our Walkscore is 91 & we are a block away from 4 bus lines that take you to the Market, Downtown & Space Needle. In addition, you have the option of walking to Downtown (the convention center is 1.3 miles away & will take you 25-30 minutes). We have an extra room downstairs with a full bed & an attached bathroom for your use. The private suite is on the ground level of a 3-story townhouse. We are less than half a mile from Elysian Brewpub, Rione XIII, Anchovies & Olives, Spinasse, Cafe Flora, Crush, Luc and Victrola Coffee. We are less than half a mile away from Seattle U and Seattle Central Community College We are a block away from Safeway & 3 blocks aw	Walk to Pike/Pine cafes, clubs & bars. Private bed & bath on separate floor, feels like your own flat. Free street parking & private patio. Easy, frequent bus service to other hoods. Park, walk ,or bus easily to all parts of the city.
5					We have a 2 night minimum. The room is good sized with a nice big window, and a small shared bathroom and kitchen. We have WiFi as well as a	

What Did We Achieve?

- We started with basic pattern matching and worked our way up to powerful RAG search using structured data.
- Each step built on the limitations of the previous one, resulting in a more refined and context-aware solution.
- This demonstrates the flexibility and power of PostgreSQL when paired with advanced extensions.



Azure Database for PostgreSQL: AI-Ready for Enterprise Applications




“AI isn't the future,
It’s already here.”


Microsoft
Postgres
Microsoft
Postgres
Microsoft
Postgres
MICROSOFT
MICROSOFT
MICROSOFT

And we ❤️ socks!

Thank You!

Varun Dhawan

LinkedIn	linkedin.com/in/varundhawan/
Twitter	@iVarund
GitHub	www.github.com/varun-dhawan
Blog	data-nerd.blog

Let me know about your experiences with Postgres and pgvector.