Microsoft

PGConf India

# Using Postgres to locate best coffee near you! ☕📍

Varun Dhawan
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](data-nerd.blog)

→ Find me on social

linkedin.com/in/varundhawan

@iVarund

varun-dhawan

Varun Dhawan

# Coffee, I had today morning was already the best!

However, we’ll try to find something better nearby!

# Our agenda – Find coffee!

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

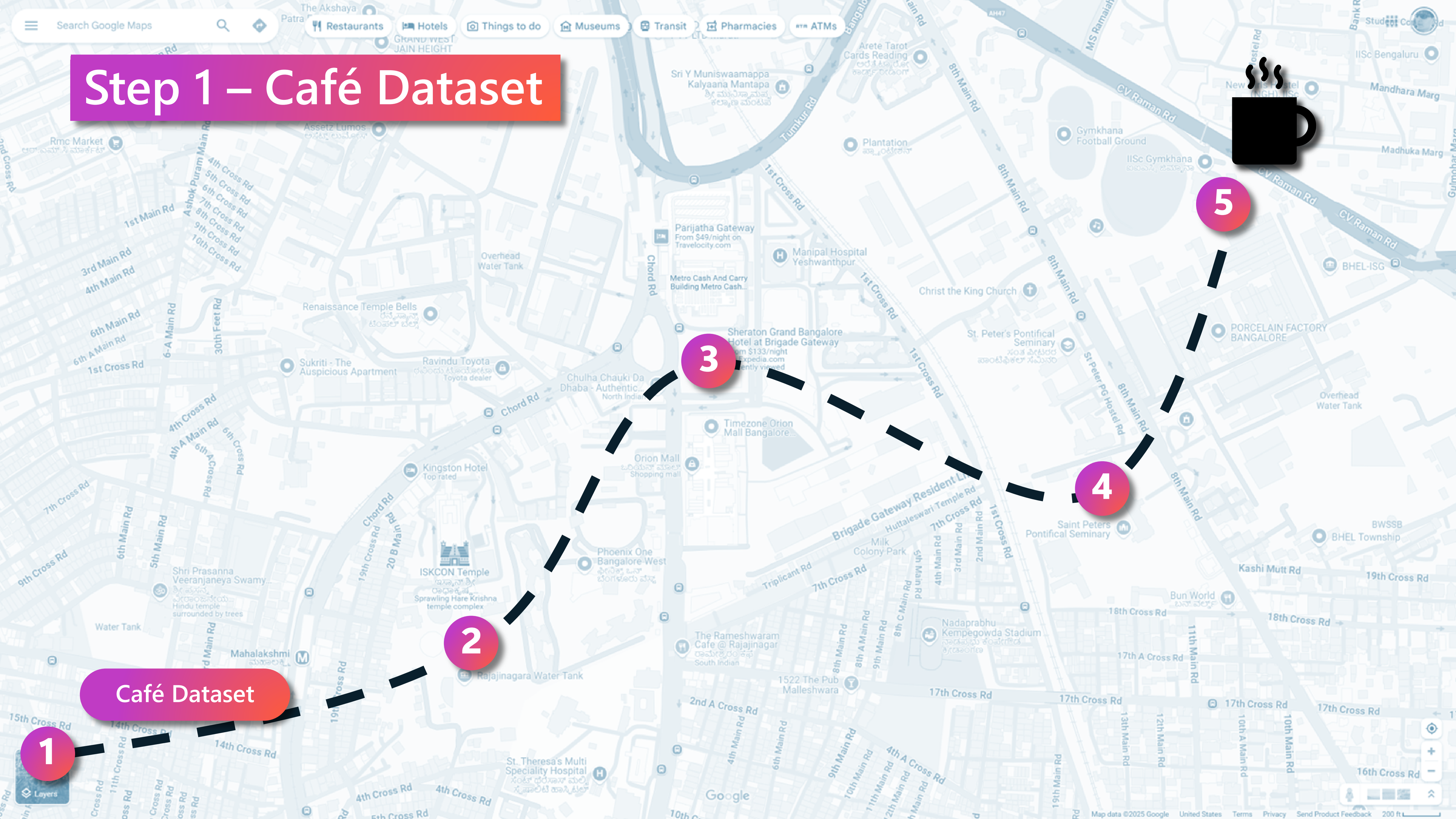
## Story Premise

- After a full day of insightful sessions at PGConf India, you're ready to unwind.
- You and your fellow attendees are looking for a coffee spot nearby to relax and chat.
- But you're in the mood for a place with cozy vibes, tasty snacks, and great ambience.
- And it should be within **5 km** of the hotel, so its easy to walk back.

## 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 zomato

## For this demo, we will use open dataset

50,000 reviews

8,000 businesses

Bangalore metro area

https://www.kaggle.com/datasets/rajeshrampure/zomato-dataset/data

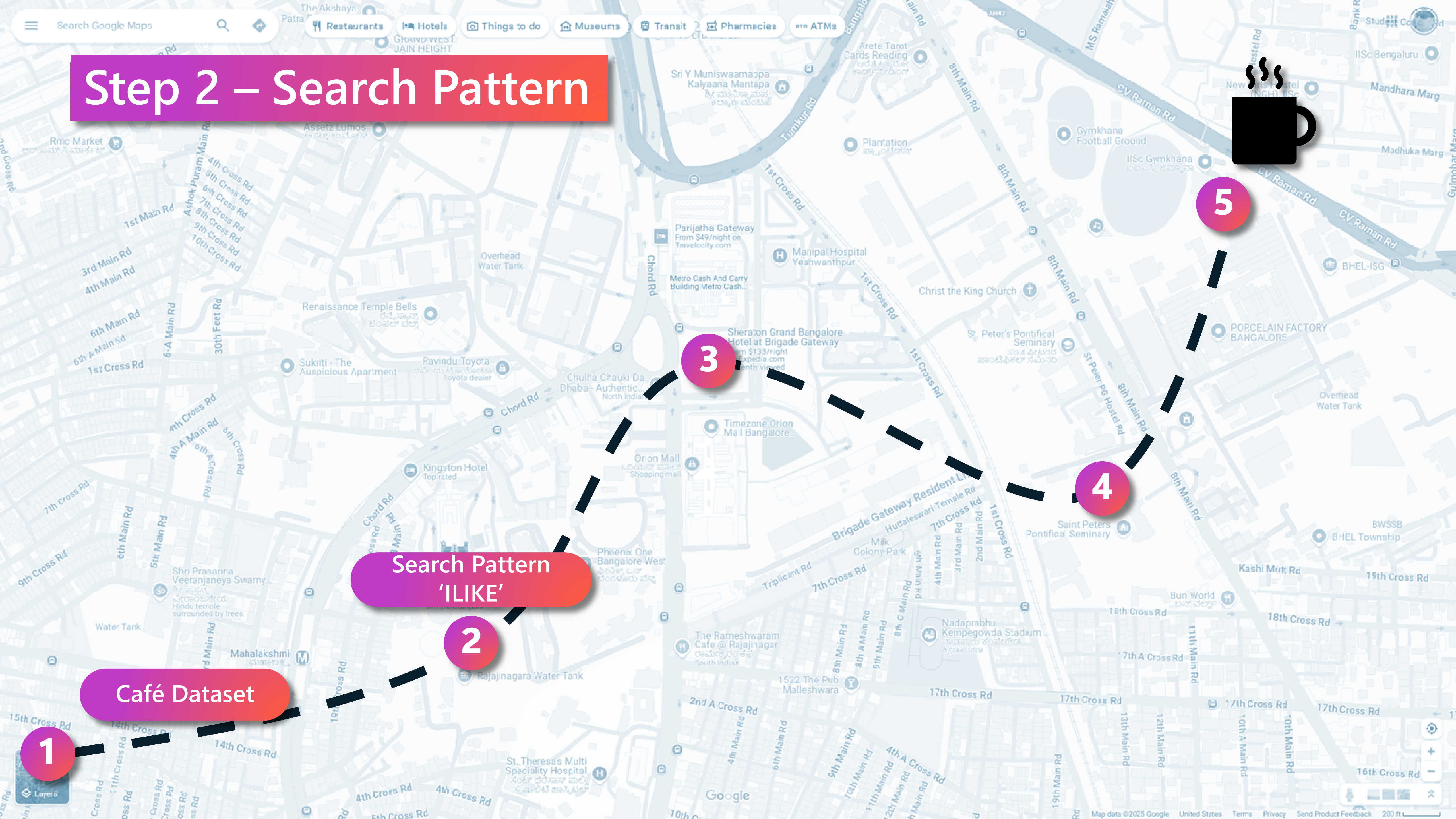
# zomato Dataset

## zomato.csv

Data including URL, business name, address, rating, and review

| url<br>URL of the restaurant | address<br>Complete address of the restaurant | name<br>Name of the restaurant | online_order<br>Do they accept online order (True/False) | book_table<br>Can we book table at the restaurant | rate<br>Rating given on zomato app | votes<br>Number of people gave rating | phone<br>Phone Number of the restaurant | location<br>Area of the restaurant | rest_type<br>Restaurant Type(Casual Dining/Cafe/Quick |
|---|---|---|---|---|---|---|---|---|---|
| 51717<br>unique values | 11495<br>unique values | 8792<br>unique values | true 30.4k 59%<br>false 21.3k 41% | true 6449 12%<br>false 45.3k 88% | [null] 15%<br>NEW 4%<br>Other (41734) 81% | 0 16.8k | [null] 2%<br>080 43334321 0%<br>Other (50293) 97% | BTM 10%<br>HSR 5%<br>Other (44070) 85% | Quick Bites 37%<br>Casual Dining 20%<br>Other (22255) 43% |
| https://www.zomato.com/bangalore/jalsa-banashankari?context=eyJzZSI6eyJlIjpbNTg2OTQsIjE4Mzc1NDc0Iiwi... | 942, 21st Main Road, 2nd Stage, Banashankari, Bangalore | Jalsa | Yes | Yes | 4.1/5 | 775 | 080 42297555 +91 9743772233 | Banashankari | Casual Dining |
| https://www.zomato.com/bangalore/spice-elephant-banashankari?context=eyJzZSI6eyJlIjpbIjU4Njk0IiwxODM... | 2nd Floor, 80 Feet Road, Near Big Bazaar, 6th Block, Kathriguppe, 3rd Stage, Banashankari, Bangalore | Spice Elephant | Yes | No | 4.1/5 | 787 | 080 41714161 | Banashankari | Casual Dining |
| https://www.zomato.com/SanchurroBangalore?context=eyJzZSI6eyJlIjpbIjU4Njk0IiwiMTgzNzU0NzQiLDU5MDkwLC... | 1112, Next to KIMS Medical College, 17th Cross, 2nd Stage, Banashankari, Bangalore | San Churro Cafe | Yes | No | 3.8/5 | 918 | +91 9663487993 | Banashankari | Cafe, Casual Dining |
| https://www.zomato.com/bangalore/addhuri-udupi-bhojana-banashankari?context=eyJzZSI6eyJlIjpbIjU4Njk0... | 1st Floor, Annakuteera, 3rd Stage, Banashankari, Bangalore | Addhuri Udupi Bhojana | No | No | 3.7/5 | 88 | +91 9620009302 | Banashankari | Quick Bites |
| https://www.zomato.com/bangalore/grand-village-basavanagudi?context=eyJzZSI6eyJlIjpbIjU4Njk0IiwiMTgz... | 10, 3rd Floor, Lakshmi Associates, Gandhi Bazaar Main Road, Above Reliance Trends, Basavanagudi, Ban... | Grand Village | No | No | 3.8/5 | 166 | +91 8026612447 +91 9901210005 | Basavanagudi | Casual Dining |
| https://www.zomato.com/bangalore/timepass-dinner-basavanagudi?context=eyJzZSI6eyJlIjpbIjE4Mzc1NDc0Ii... | 37, 5-1, 4th Floor, Bosco Court, Gandhi Bazaar Main Road, Basavanagudi, Bangalore | Timepass Dinner | Yes | No | 3.8/5 | 286 | +91 9980040002 +91 9980063005 | Basavanagudi | Casual Dining |
| https://www.zomato.com/bangalore/rosewood-international-hotel-bar-restaurant-mysore-road-bangalore?c... | 19/1, New Timberyard Layout, Beside Satellite Bus Stop, Mysore Road, Bangalore | Rosewood International Hotel - Bar & Restaurant | No | No | 3.6/5 | 8 | +91 9731716688 080 26740366 | Mysore Road | Casual Dining |

# 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%'.

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

# DEMO -  ILIKE ‘% search-pattern %’

```sql
-- Search for mentions of 'restaurant' and 'cafe' in the menu_item column using ILIKE.
SELECT
    name AS restaurant_name, rate, votes, rest_type, cuisines, approx_cost, menu_item
FROM zomato_data
WHERE
    rest_type ILIKE '%casual%'  -- Searching for restaurants mentioning 'restaurant'
    OR rest_type ILIKE '%coffee%'      -- And also those mentioning 'cafe'
ORDER BY rate DESC;
```

mato_data 1 ×

LECT name AS restaurant_name, rate, votes, rest_type, cuisines, ap | Enter a SQL expression to filter results (use Ctrl+Space)

| | restaurant_name | rate | votes | rest_type | cuisines | approx_cost | |
|---|---|---|---|---|---|---|---|
| | Dhana Lakshmi Family Restaurant | NaN | 0 | Casual Dining | Chinese, North Indian | 650 | [] |
| | Dhana Lakshmi Family Restaurant | NaN | 0 | Casual Dining | Chinese, North Indian | 650 | [] |
| | Stories | 4.6 /5 | 2,423 | Casual Dining, Bar | Continental, North Indian, Chinese | 1,100 | Na |
| | Stories | 4.6 /5 | 2,315 | Casual Dining, Bar | Continental, North Indian, Chinese | 1,100 | Na |
| | Stories | 4.6 /5 | 2,338 | Casual Dining, Bar | Continental, North Indian, Chinese | 1,100 | Na |
| | Stories | 4.6 /5 | 2,396 | Casual Dining, Bar | Continental, North Indian, Chinese | 1,100 | Na |
| | Stories | 4.6 /5 | 2,315 | Casual Dining, Bar | Continental, North Indian, Chinese | 1,100 | Na |
| | Stories | 4.6 /5 | 2,423 | Casual Dining, Bar | Continental, North Indian, Chinese | 1,100 | Na |
| | Barbeque Nation | 4.5 /5 | 1,282 | Casual Dining | North Indian, European, Mediterranean, BBQ, Kel | 1,600 | [] |
| 0 | Barbeque Nation | 4.5 /5 | 1,284 | Casual Dining | North Indian, European, Mediterranean, BBQ, Kel | 1,600 | [] |
| 1 | Sagar Hotel | 4.3 /5 | 818 | Casual Dining | North Indian, Chinese, Street Food, Juices | 600 | [] |
| 2 | Sagar Hotel | 4.3 /5 | 813 | Casual Dining | North Indian, Chinese, Street Food, Juices | 600 | [] |

```sql
-- Let's compare this pattern matching with a semantic search query for "coffee with sandwich".
SELECT id, name, rate, votes, rest_type, cuisines, approx_cost, menu_item
FROM zomato_data
WHERE
    menu_item ILIKE '%coffee with sandwich%'; -- Search for cafes in menu item text

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

ato_data 1 ×

LECT id, name, rate, votes, rest_type, cuisines, approx_cost, menu_i | Enter a SQL expression to filter results (use Ctrl+Space)

| id | name | rate | votes | rest_type | cuisines | approx_cost | menu_item |
|---|---|---|---|---|---|---|---|

Save | Cancel | Script | 200 | 0 | Rows: 0 | No data- 210ms, on 2025-03-07 at 09:35:32

IST | en | Writable | Smart Insert | 30 : 96 : 1130 | Sel: 0 | 0

# Search Pattern

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

```sql
WHERE r.text ILIKE '%cafe with chill vibes%’;
-- Search for cafes in review text
```

# Step 3 – Full-Text search for a phrase

1 Café Dataset

2 Search Pattern ‘ILIKE’

3 Search Phrase Full-Text

4

5

# 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 Bangalore'.

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

# DEMO - Full Text Search

*<DiskANN> zomato_coffee_search.sql | <DiskANN> 2_yelp_AI_setup.sql

```sql
-- PART B >> FULL TEXT SEARCH
-- SUBTITLE: Advanced Text Search Capabilities
-- DESCRIPTION: This section introduces full-text search capabilities using PostgreSQL's text search features.
*************************************************************/

-- 1⃣ Add a `menu_item_search` column of type `tsvector` for full-text search
ALTER TABLE zomato_data
ADD COLUMN menu_item_search tsvector;

-- 2⃣ Populate the `menu_item_search` column using existing menu items.
-- `to_tsvector()` converts text into a tsvector format for efficient full-text search.
UPDATE zomato_data
SET menu_item_search = to_tsvector('english', coalesce(menu_item, ''));

-- 3⃣ Create a GIN (Generalized Inverted Index) for efficient full-text search
CREATE INDEX idx_zomato_menu_search ON zomato_data USING GIN (menu_item_search);

-- Sample records
SELECT name, menu_item, menu_item_search FROM zomato_data;
```

zomato_data 1

SELECT name, menu_item, menu_item_search FROM zomato_data W | Enter a SQL expression to filter results (use Ctrl+Space)

| | name | menu_item | menu_item_search |
|---|---|---|---|
| 9 | Aramane Restaura | ['Idly with Vada', | '1':484 '2':24,34,53 '3':38 '65':102 'aloo':120,138,159,163,165,167,228,236,239,250,254,256,308,441 'american':411 'appl':456,486 'araman':87 |
| 10 | Baskin Robbins | ['Vanilla Ice Crea | '200':197,202 '300':177,182,187,192 '90':233,240,247,254 'affair':257 'almond':194 'banana':208,222,282 'bavarian':51,117,235,265 'belgian':2 |
| 11 | Behrouz Biryani | ['Dum Gosht Biry | '1':245 '2':53,257,266 '4':273,281 '8':288,296 'ayran':230 'bhuna':19,30,36,72,116,181 'biryan':151 'biryani':3,8,12,16,21,27,32,38,43,48,56,61,65 |
| 12 | Bhairava Deluxe H | [] | |
| 13 | 1947 | [] | |
| 14 | Biriyani Adda | [] | |

Save | Cancel | Script | 200 | 200+ | Rows: 1 | 200 row(s) fetched - 290ms, on 2025-03-07 at 09:39:36

*<DiskANN> zomato_coffee_search.sql ×   <DiskANN> 2_yelp_AI_setup.sql

```sql
ORDER BY rate DESC;

-- 5⃣ PHRASE SEARCH - Using web-style search queries
-- Searching for restaurants offering 'grilled sandwich' with websearch-like syntax
SELECT
    name AS restaurant_name, address, rate, votes, cuisines, approx_cost, menu_item
FROM zomato_data
WHERE
    menu_item_search @@ websearch_to_tsquery('english', 'coffee')
ORDER BY rate DESC
LIMIT 5;
```

zomato_data 1 ×

SELECT name AS restaurant_name, address, rate, votes, cuisines, app | Enter a SQL expression to filter results (use Ctrl+Space)

| | restaurant_name | address | rate | votes | cuisines | approx_co |
|---|---|---|---|---|---|---|
| 1 | Mitraa Da Pizza | 18, Siddheshwara Complex, Ground Floor, Dr Raj Kumar Road, | NEW | 0 | Fast Food, Pizza | |
| 2 | The Juice Station | 2942, Akshaya Arcade Second Stage, Rajajinagar, Bangalore | NaN | 0 | Juices, Sandwich, Burger, Beverages, Continental | |
| 3 | Empire Restaurant | 701, Sri Lakshmi Narayana Arcade, West of Chord Road 2nd Sta | 4.0 /5 | 630 | North Indian, Mughlai, South Indian, Chinese | |
| 4 | Frozenheit? | 31/1, 20th Main Road, BVR Plaza, 8th cross, 1st N Block, Rajajin | 4.0 /5 | 154 | Ice Cream | |
| 5 | Frozenheit? | 31/1, 20th Main Road, BVR Plaza, 8th cross, 1st N Block, Rajajin | 4.0 /5 | 154 | Ice Cream | |

Save  Cancel  Script  200  5  Rows: 1  5 row(s) fetched - 214ms, on 2025-03-07 at 09:42:01

*<DiskANN> zomato_coffee_search.sql | <DiskANN> 2_yelp_AI_setup.sql

```sql
-- 6 NATURAL LANGUAGE SEARCH EXAMPLE
-- Searching for restaurants that serve "large masala fries"
SELECT
    name AS restaurant_name, address, rate, votes, cuisines, approx_cost, menu_item
FROM zomato_data
WHERE
    menu_item_search @@ phraseto_tsquery('english', 'coffee with sandwich')
ORDER BY rate DESC
LIMIT 5;
```

zomato_data 1

SELECT name AS restaurant_name, address, rate, votes, cuisines, app | Enter a SQL expression to filter results (use Ctrl+Space)

| restaurant_name | address | rate | votes | cuisines | approx_co |
|---|---|---|---|---|---|

Save | Cancel | Script | 200 | 0 | Rows: 0 | No data- 211ms, on 2025-03-07 at 09:43:39

# Search Phrase – FULL TEXT

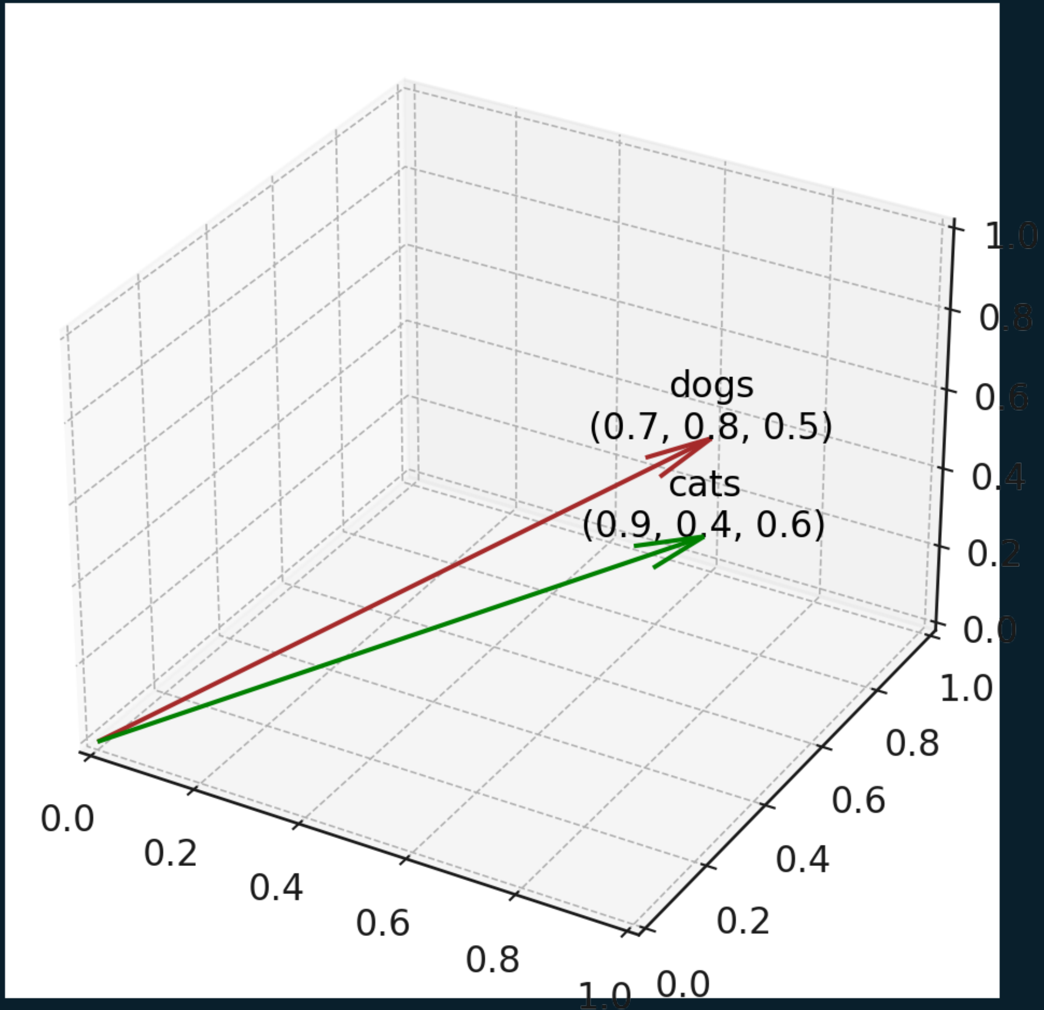
This is better for phrase-based searches but struggles with nuanced queries like ‘cafe with chill vibes’.

```
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):

# 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, 0.039895348250865936, 0.0012036223197355866, 0.009372025728225708, -0.012178223580121994, -0.019853007048368454, 0.006024873349815607, 0.011319459415972233, -0.025167878717184067, -0.00759363966062665, 0.010284884832799435, 0.009831836447119713, -0.008492975495755672, -0.005639444105327129, -0.009446406736969948, 0.007444877177476883, -0.009277358651161194, -0.025289593264460564, -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, -0.00038881058571860194, -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 |

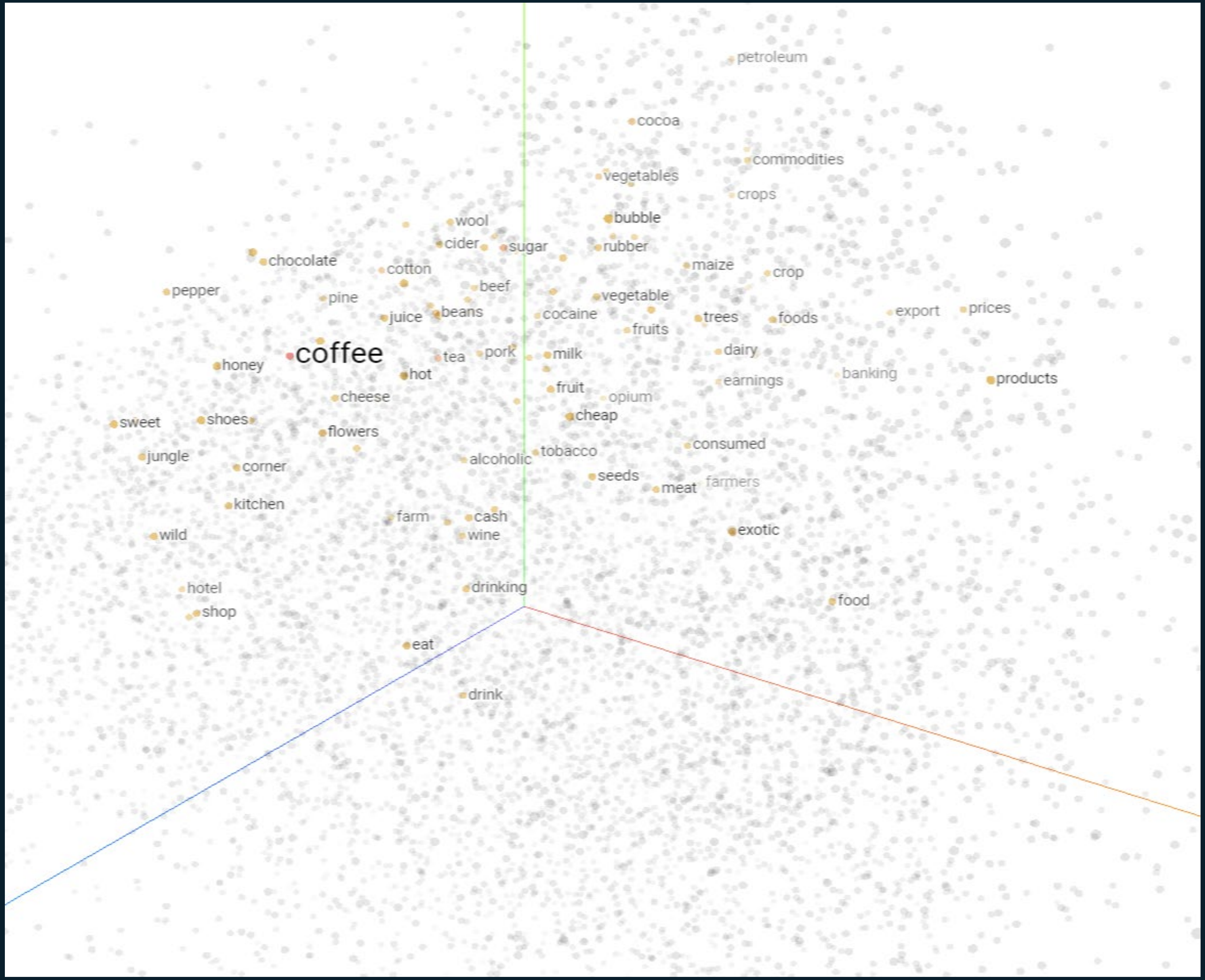
#### Least similar:

| | |
|---|---|
| catalonia | 0.7746281384075008 |
| anymore | 0.7745343111964632 |
| netherlands | 0.7744193510029177 |
| worse | 0.774271453446651 |
| shouldn | 0.7741518238387108 |

# 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?

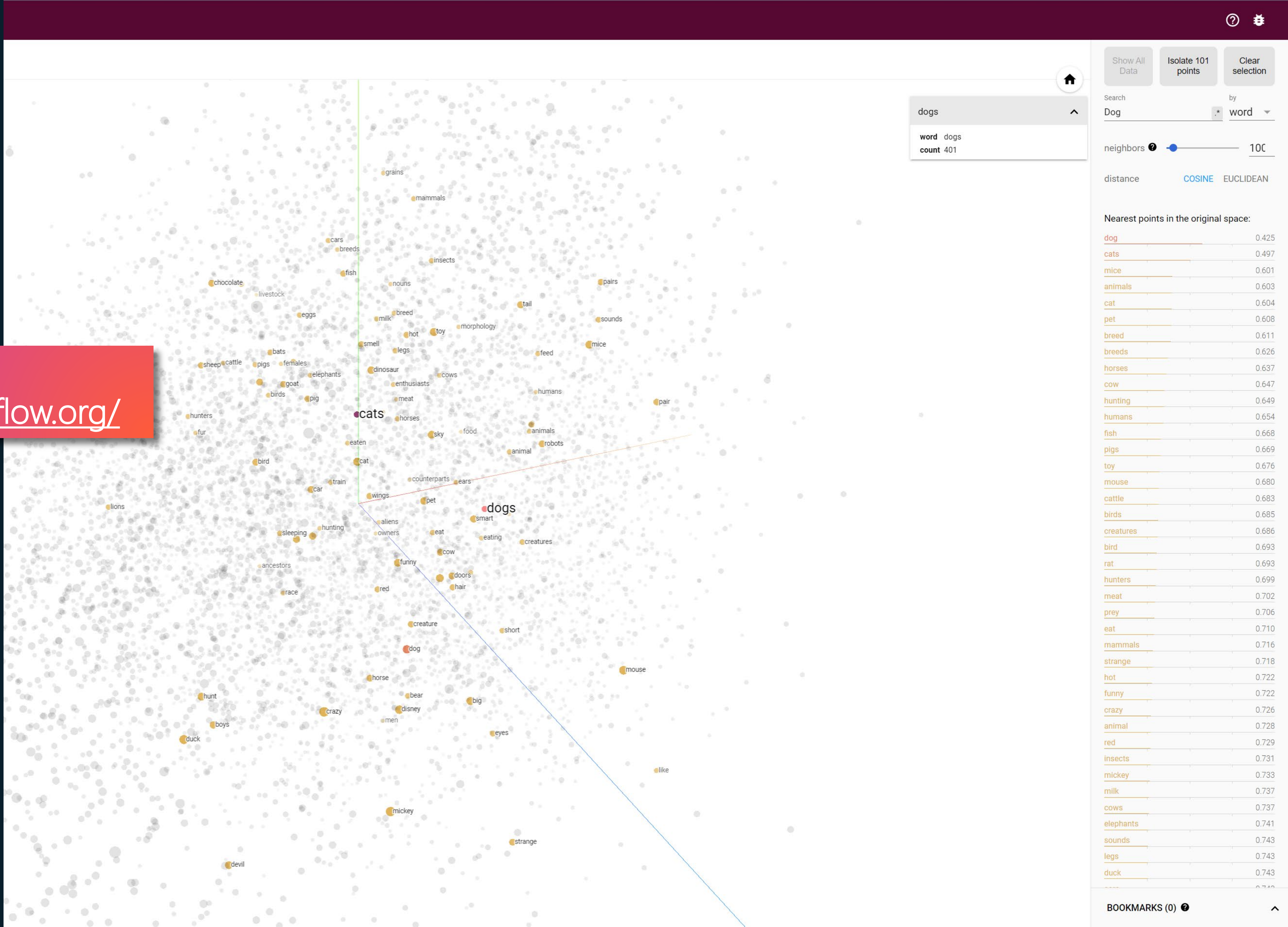
## Search

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

# Example:

Visualize Vector

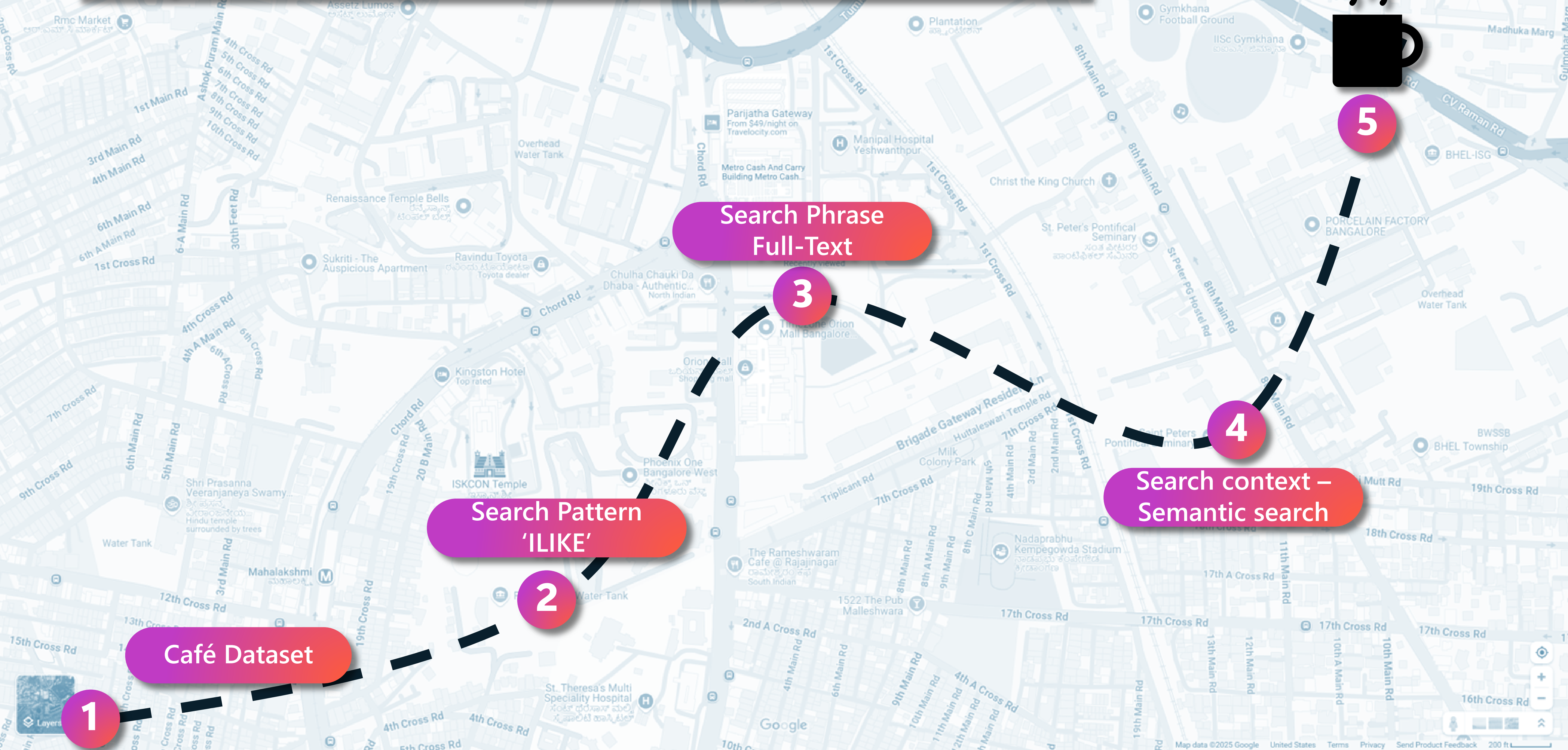
https://projector.tensorflow.org/



# Storing vectors in Postgres table

| restaurant_name | description_vector |
| --- | --- |
| Cold Station | [-0.021632893,-0.032019176,-0.02327874,-0.069424756,-0.027605318,-0.0205606,0.009401269,0.043340597,0.0377547,0.008416256,0.043540094,-0.054861516,-0.001209447,-0.000607451,-0.004906365,0.01972521,-0.01146481,0.020934656,-0.05306605,0.033216156,0.04102145,0.04588 |
| Empire Restaurant | [-0.009561688,-0.020217897,-0.026563661,-0.027203156,-0.0024995666,-0.011295705,-0.0014711472,0.023784315,0.01187371,0.027104773,0.015655342,-0.05637399,0.014536225,-0.010951361,-0.012267246,0.012900593,0.021472292,0.012814507,-0.049733076,0.043313526,0.0226529, |
| Lassi Shop | [-0.0021951075,-0.064719774,-0.04115865,-0.057583004,-0.019222848,-0.004213018,0.00036795167,0.043969363,0.039081164,-0.014102459,0.027447246,-0.06643064,-0.009202038,-0.023573346,-0.016424354,0.016412133,0.012128848,0.016571,-0.064622015,0.030624576,0.03380190 |
| Mitraa Da Pizza | [-0.027787942,-0.023954302,-0.027002165,-0.040312756,-0.028478473,-0.028478473,0.0024897829,0.05905235,0.023906678,0.008191133,0.02857372,-0.052813757,-0.015191694,0.0016102481,-0.0092626475,0.006649343,-0.012108114,0.024323378,-0.045051232,0.015751263,0.040693 |
| South Idlis | [0.0039331196,-0.043976244,0.01068478,-0.015825845,-0.05597401,-0.029947728,0.036693554,0.00967524,0.054386757,0.022781748,0.005870502,-0.072966956,0.024929209,0.009646063,0.008064645,-0.0056808484,0.029200787,0.0106906155,-0.021883084,0.101397455,0.050698727,0 |

# 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 Sheraton Grand
- With the power of PostGIS extension, we can combine the semantic search results with spatial data, and ask "Coffee with sandwich, near Sheraton Grand".

```sql
WHERE
ST_DWithin(b.business_location::geography,ST_GeographyFromText('POINT(77.5556 13.0110)'), 5000)
-- Spatial filter: only include businesses within 5 km of Sheraton Grand Hotel.
```

+

```sql
ORDER BY  description_vector <=> azure_openai.create_embeddings('text-embedding-3-small', 'coffee near me with sandwich')::vector
-- Rank results by vector similarity to the given search query.
```

# DEMO - Vector semantic search

*<DiskANN> zomato_coffee_search.sql   <DiskANN> 2_yelp_AI_setup.sql

```sql
-- 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
*************************************************************/
-- Check PostgreSQL version and available extensions
SHOW server_version;
SHOW azure.extensions;

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

pg_extension 1

SELECT pge.extname, pge.extversion FROM pg_extension pge WHEI | Enter a SQL expression to filter results (use Ctrl+Space)

| | extnam | extversion |
|---|---|---|
| 1 | azure_ai | 1.1.0 |
| 2 | postgis | 3.3.3 |
| 3 | vector | 0.7.0 |
| 4 | pg_diskann | 0.3.2 |

Save Cancel Script 200 4 Rows: 1 4 row(s) fetched - 208ms, on 2025-03-07 at 09:44:54

*<DiskANN> zomato_coffee_search.sql ✕ <DiskANN> 2_yelp_AI_setup.sql

```sql
DO $$
DECLARE
    counter integer := (SELECT COUNT(*) FROM zomato_data WHERE menu_item <> '' AND description_vector IS NULL);
    r record;
BEGIN
    WHILE counter > 0 LOOP
        FOR r IN
            SELECT * FROM zomato_data
            WHERE menu_item <> '' AND description_vector IS NULL
        LOOP
            UPDATE zomato_data
            SET description_vector = azure_openai.create_embeddings(
                'text-embedding-3-small',
                COALESCE(r.name, '') || ' ' || COALESCE(r.menu_item, '') || ' ' || COALESCE(r.cuisines, '')
            )
            WHERE id = r.id;
            counter := (SELECT COUNT(*) FROM zomato_data WHERE menu_item <> '' AND description_vector IS NULL);
            IF counter % 25 = 0 THEN
                COMMIT;
            END IF;
        END LOOP;
    END LOOP;
END;
$$;

-- 3⃣ Test if the vector embeddings are populated
SELECT name AS restaurant_name, description_vector FROM zomato_data WHERE menu_item  ILIKE '%coffee%' LIMIT 5;
```

zomato_data 1 ✕

SELECT name AS restaurant_name, description_vector FROM zomato | Enter a SQL expression to filter results (use Ctrl+Space)

| | restaurant_name | description_vector |
|---|---|---|
| 1 | Aramane Restaurant | [-0.043055058,-0.033062674,-0.02295162,-0.019035367,-0.012342133,-0.03049931,0.021574998,0.022643067,0.029621119,0.0390675 |
| 2 | Baskin Robbins | [0.01925521,-0.032931935,-0.024281237,-0.04877807,0.0026005986,-0.01805597,-0.0043219794,0.054275706,0.011480363,-0.020535 |
| 3 | Cold Station | [-0.021632893,-0.032019176,-0.02327874,-0.069424756,-0.027605318,-0.0205606,0.009401269,0.043340597,0.0377547,0.008416256,( |

Save Cancel Script 200 5 Rows: 1 5 row(s) fetched - 260ms, on 2025-03-07 at 09:46:37

*<DiskANN> zomato_coffee_search.sql   <DiskANN> 2_yelp_AI_setup.sql

```sql
-- 3□ Test if the vector embeddings are populated
SELECT name AS restaurant_name, description_vector FROM zomato_data WHERE menu_item  ILIKE '%coffee%' LIMIT 5;

-- 4□ Create a vector index for efficient vector similarity search
-- DiskANN (Disk-based Approximate Nearest Neighbor) is an efficient index for searching large-scale vector data.
CREATE EXTENSION IF NOT EXISTS pg_diskann CASCADE;

CREATE INDEX description_vector_diskann_idx ON zomato_data USING diskann (description_vector vector_cosine_ops);

-- 5□ Simple vector semantic search example
-- This query searches for restaurants that are semantically similar to 'sandwich shop with vegetarian options' using v
SELECT name AS restaurant_name, rate, votes, rest_type, cuisines, approx_cost, menu_item
FROM zomato_data
ORDER BY description_vector <=> azure_openai.create_embeddings('text-embedding-3-small', 'coffee with sandwich')::vecto
LIMIT 10;
```

zomato_data 1

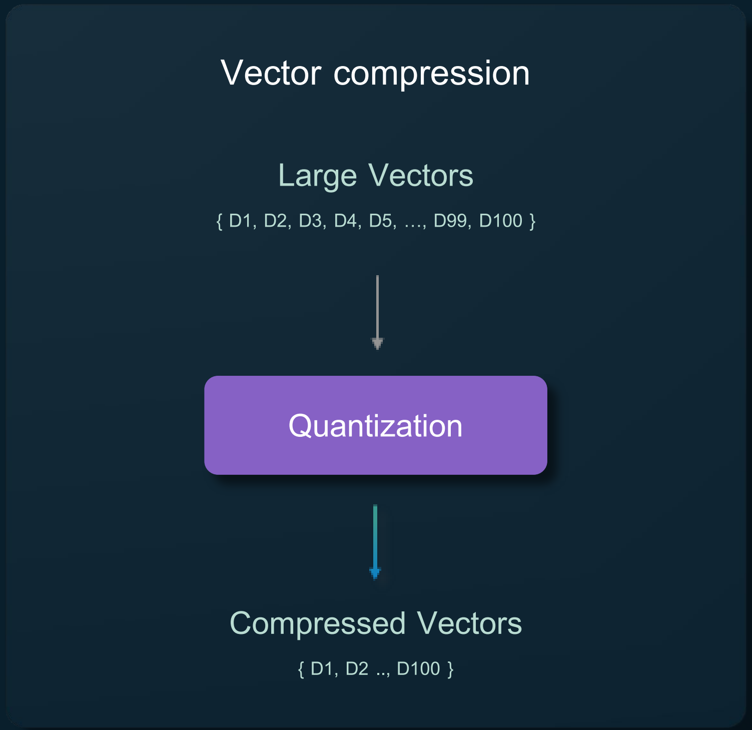
SELECT name AS restaurant_name, rate, votes, rest_type, cuisines, apr   Enter a SQL expression to filter results (use Ctrl+Space)

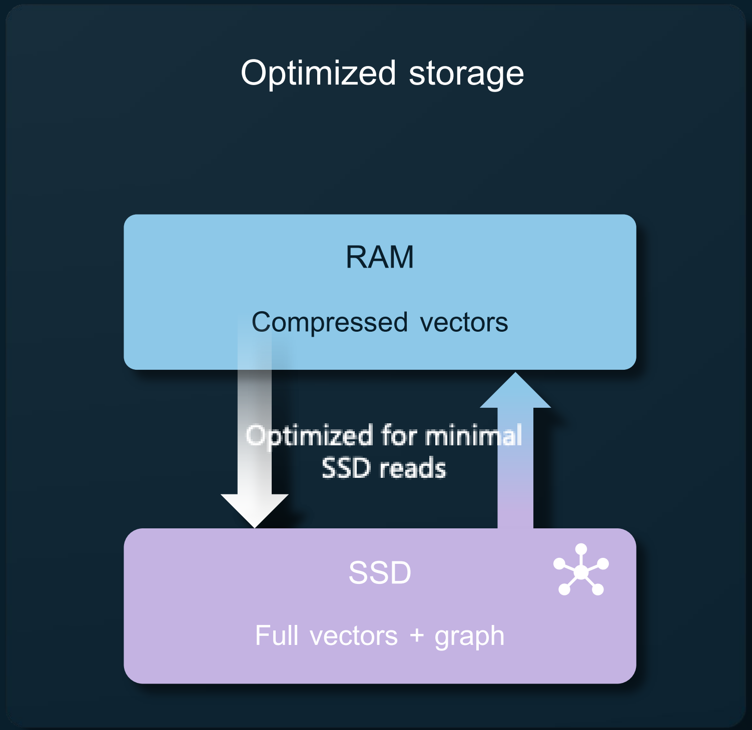
| | restaurant_name | rate | votes | rest_type | cuisines | approx_cost | menu_item |
|---|---|---|---|---|---|---|---|
| 1 | Hot Coffee | 3.3 /5 | 4 | Quick Bites | South Indian, Chinese, Beverages | 150 | [] |
| 2 | Hot Coffee | 3.3 /5 | 4 | Quick Bites | South Indian, Chinese, Beverages | 150 | [] |
| 3 | Cafe Coffee Day | 3.0 /5 | 20 | Cafe | Cafe, Fast Food | 900 | [] |
| 4 | Create Cafe | 3.4 /5 | 10 | Quick Bites | Italian | 300 | [] |
| 5 | Create Cafe | 3.4 /5 | 11 | Quick Bites | Italian | 300 | [] |
| 6 | Create Cafe | 3.4 /5 | 9 | Quick Bites | Italian | 300 | ['Veg Salad', ' |
| 7 | Cakeport | 3.4 /5 | 36 | Dessert Parlor | Bakery | 400 | [] |
| 8 | Cakeport | 3.4 /5 | 36 | Dessert Parlor | Bakery | 400 | [] |

Save  Cancel  Script  200  10  Rows: 1  10 row(s) fetched - 366ms, on 2025-03-07 at 09:48:10

# DiskANN – Vector Index for Performance and Accuracy

- Highly performant, scalable, and accurate index for vectors
- Superior to IVFLAT and HNSW
- Reduced memory footprint by storing vectors on SSD
- Compression and quantization improve speed and accuracy of vector search
- Accuracy retained as data changed



*<DiskANN> zomato_coffee_search.sql    <DiskANN> 2_yelp_AI_setup.sql

```sql
-- 6⃣ HYBRID SEARCH: Combine vector similarity with geospatial filtering
-- Add a new column `restaurant_location` to store the geographical coordinates.
ALTER TABLE zomato_data ADD COLUMN restaurant_location geometry(point, 4326);

-- Populate the `restaurant_location` with latitude and longitude.
UPDATE zomato_data
SET restaurant_location = ST_SetSRID(ST_Point(longitude, latitude), 4326);

-- Sample Data
SELECT name AS restaurant_name, restaurant_location, description_vector FROM zomato_data LIMIT 5
```

zomato_data 1

SELECT name AS restaurant_name, restaurant_location, description_v | Enter a SQL expression to filter results (use Ctrl+Space)

| | restaurant_name | restaurant_location | description_vector |
|---|---|---|---|
| 1 | Altaj Restaurant | POINT (77.5528973 12.9899287) | [-0.053034913,-0.037086163,0.051546365,-0.007985008,-0.026368603,-0.042317353,-0.0294 |
| 2 | Altaj Restaurant | POINT (77.5528973 12.9899287) | [-0.053034913,-0.037086163,0.051546365,-0.007985008,-0.026368603,-0.042317353,-0.0294 |
| 3 | Al-Taj Restaurant | POINT (77.5528973 12.9899287) | [-0.038120918,-0.058015533,0.03797494,0.00085044786,-0.03905934,-0.0313434,-0.0170376 |
| 4 | Coastal Spice | POINT (77.5601684 12.9870689) | [0.019696297,-0.017024228,0.04947411,0.019567944,-0.039579287,-0.027047403,-0.0106999 |
| 5 | Food & Friends | POINT (77.5512332999999 12.9971572) | [-0.03106937,-0.028591184,-0.011852809,-0.014423042,0.0288036,-0.028421251,-0.0025419 |

Save Cancel Script 200 5 Rows: 1 5 row(s) fetched - 262ms, on 2025-03-07 at 09:52:10

*<DiskANN> zomato_coffee_search.sql × <DiskANN> 2_yelp_AI_setup.sql

```sql
-- 7⃣ Hybrid Search (Spatial + Vector Similarity)
-- Search for restaurants within a 5 km radius of a given location and rank them by vector similarity.
WITH coffee_places_cte AS (
    SELECT
        DISTINCT ON (z.name) z.id,  -- Ensuring unique restaurant names
        z.name AS "Coffee Place",
        z.restaurant_location,
        ROUND(AVG(CAST(NULLIF(REGEXP_REPLACE(z.rate, '[^0-9.]', '', 'g'), '') AS NUMERIC)), 1) AS "Average Rating",
        STRING_AGG(DISTINCT z.menu_item, ', ') AS "Menu Items",
        z.description_vector
    FROM zomato_data z
    WHERE
        -- Spatial filter: Find places within 5 km of Sheraton Grand Bangalore
        ST_DWithin(
            z.restaurant_location::geography,
            ST_GeographyFromText('POINT(77.5556 13.0110)'), 5000)
        AND CAST(NULLIF(REGEXP_REPLACE(z.rate, '[^0-9.]', '', 'g'), '') AS NUMERIC) > 4  -- Only highly rated places
    GROUP BY z.id, z.name, z.restaurant_location, z.description_vector
)
SELECT "Coffee Place", "Average Rating", "Menu Items"
FROM coffee_places_cte
ORDER BY "Average Rating" DESC,  -- First sort by highest rating
         description_vector <=> azure_openai.create_embeddings('text-embedding-3-small', 'coffee near me with sandwich')::vector
```

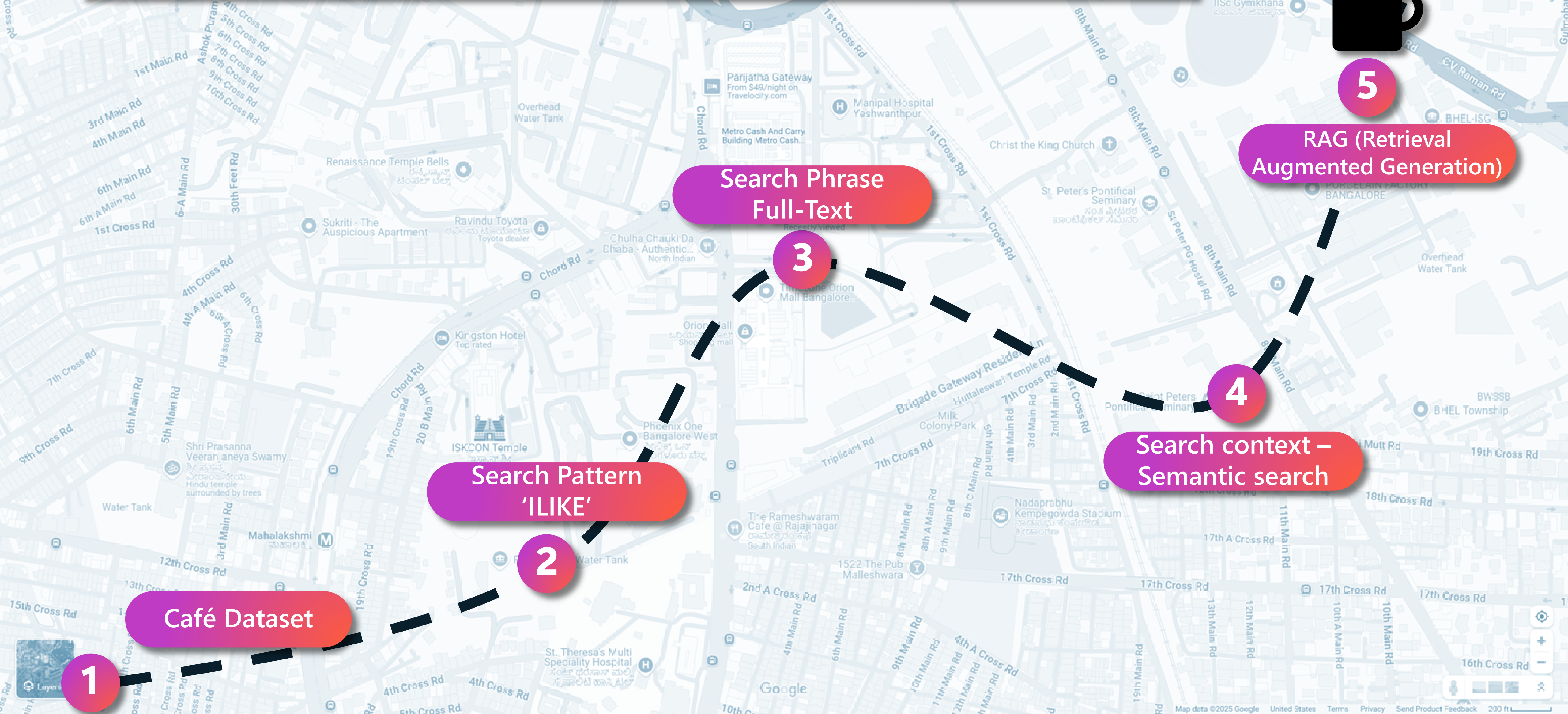
zomato_data 1 ×

WITH coffee_places_cte AS ( SELECT DISTINCT ON (z.name) z.id, z.n | Enter a SQL expression to filter results (use Ctrl+Space)

| | Coffee Place | Average Rating | Menu Items |
|---|---|---|---|
| 1 | O.G. Variar & Sons | 4.9 | [] |
| 2 | Stories | 4.7 | NaN |
| 3 | eat.fit | 4.6 | ['Masala Egg Curry & Vegetable Pulao', 'Nutty Dates Pudding', 'Spicy Arrabbiata-Stuffed Chicken & |
| 4 | Barbeque Nation | 4.6 | [] |

Save | Cancel | Script | 200 | 27 | Rows: 1 | 27 row(s) fetched - 338ms, on 2025-03-07 at 09:54:15

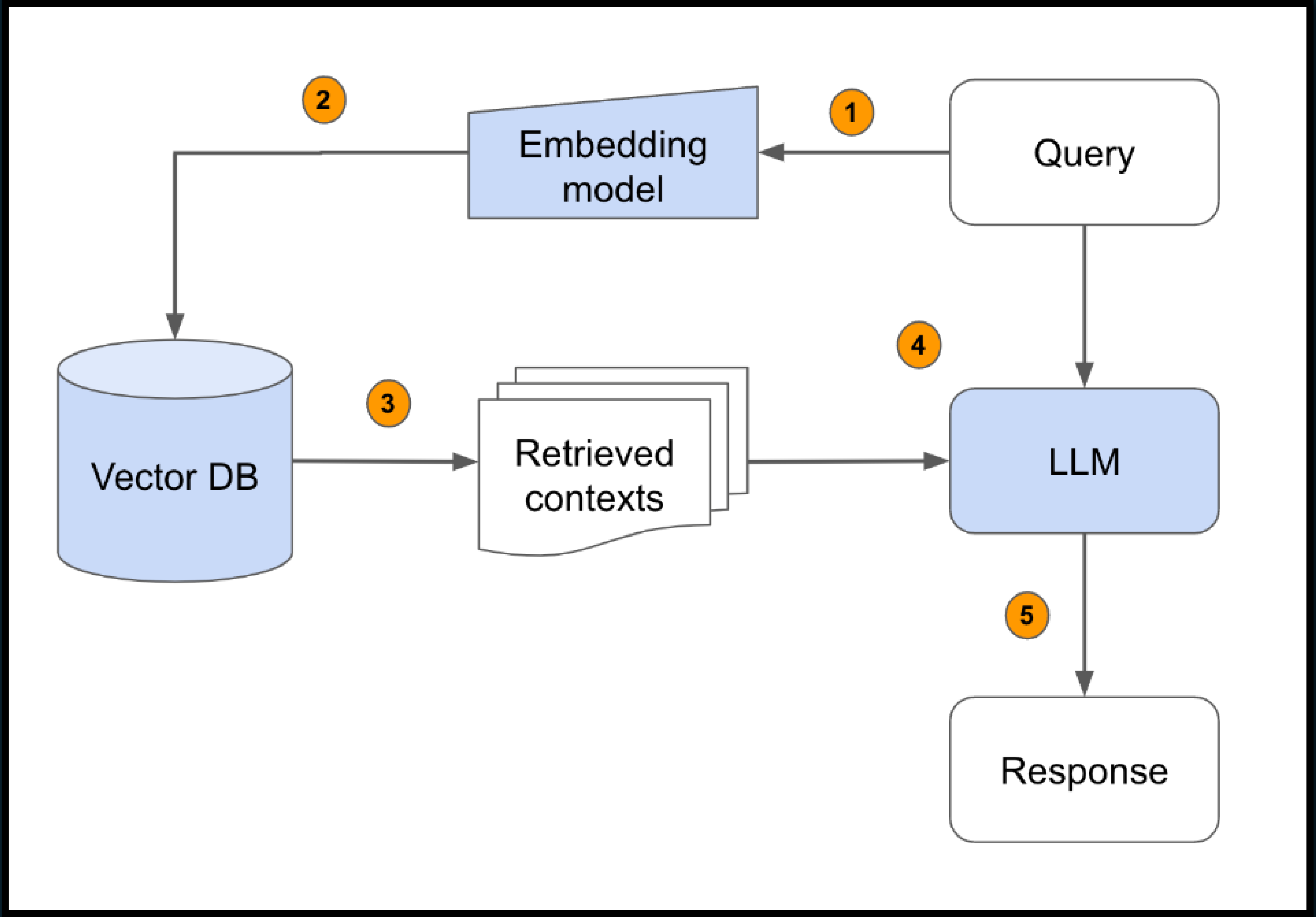
# Step 5 – Apply RAG Search on Structured Data



# What is RAG?

## Retrieval Augmented Generation

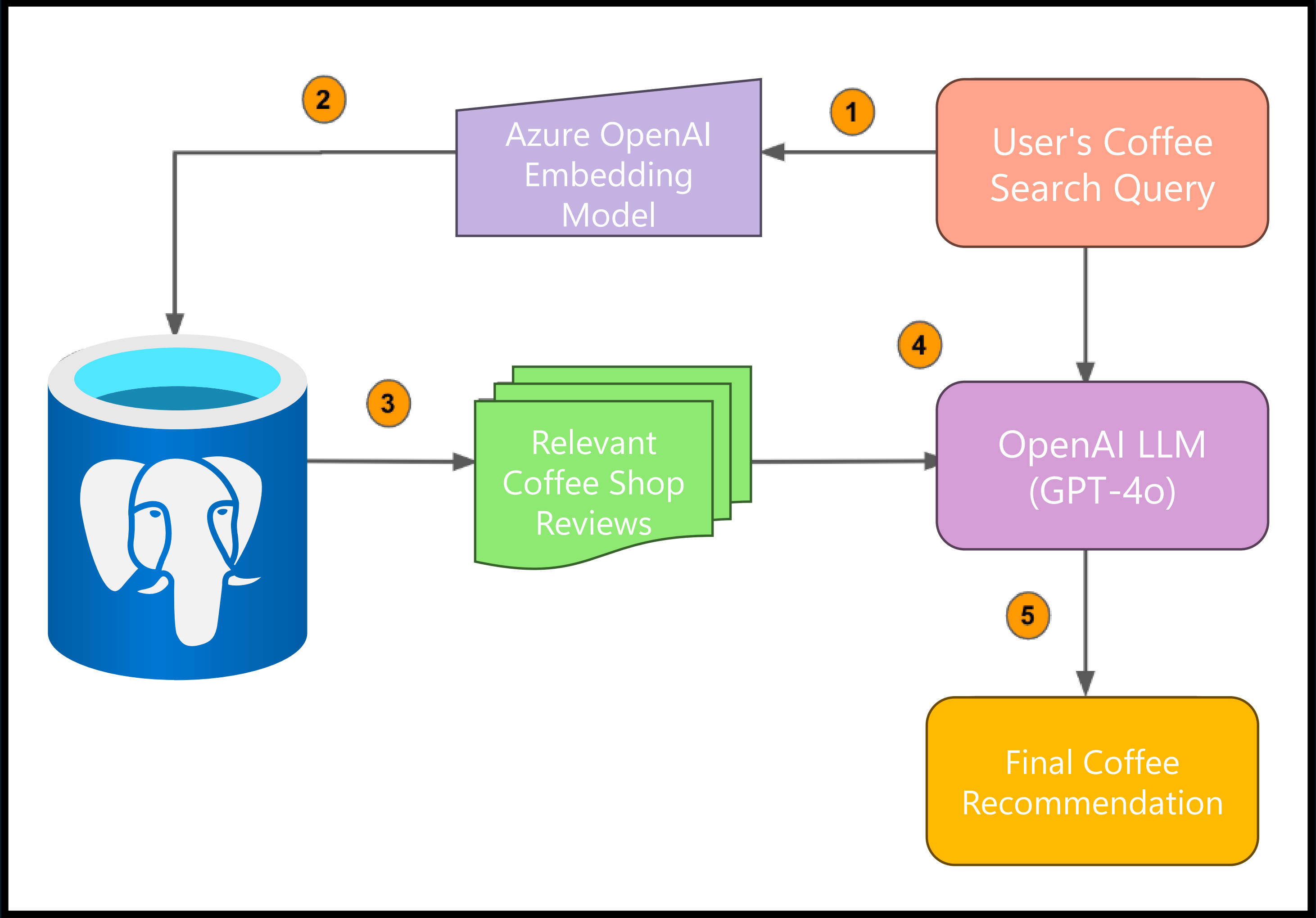
RAG is the process of retrieving relevant contextual information from out vectorized dataset and passes this information to a large language model (LLM) to generate answers.



# What is RAG?

## RAG Search on Structured Data

Instead of directly 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 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.

Index

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.

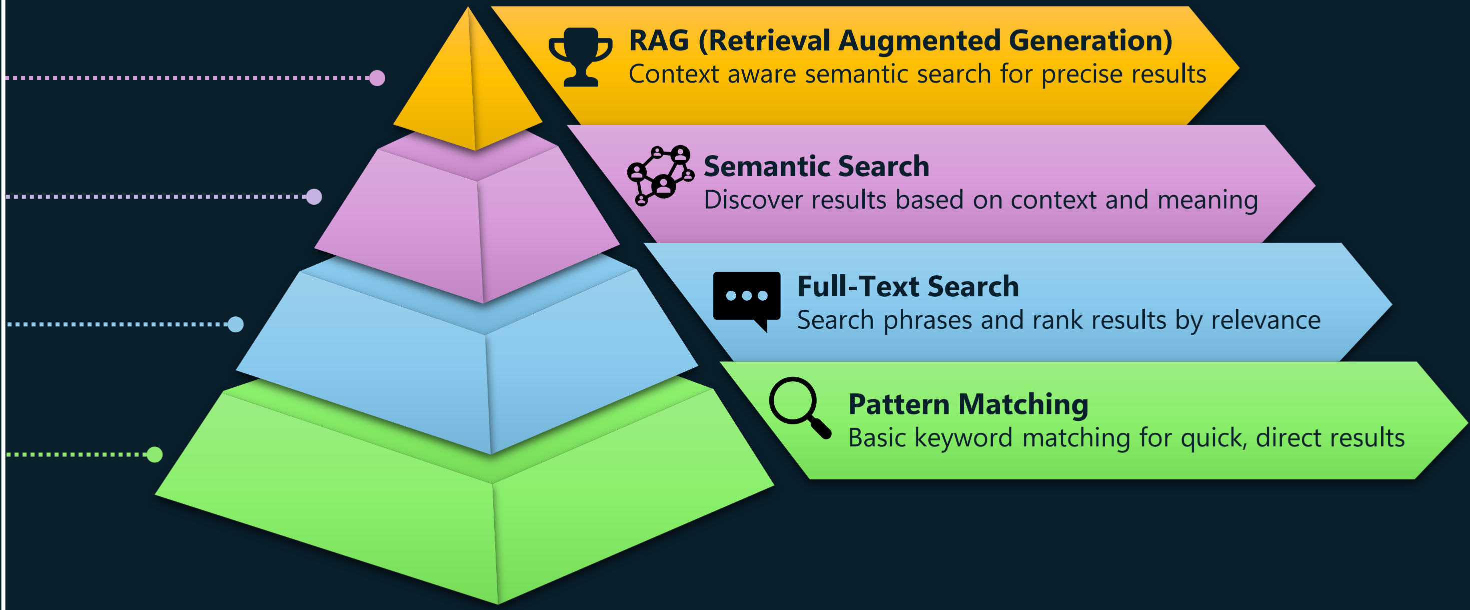
Query time: 0.21 seconds

| | 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 | 47196 | 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. |
| | | | | | 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 | |

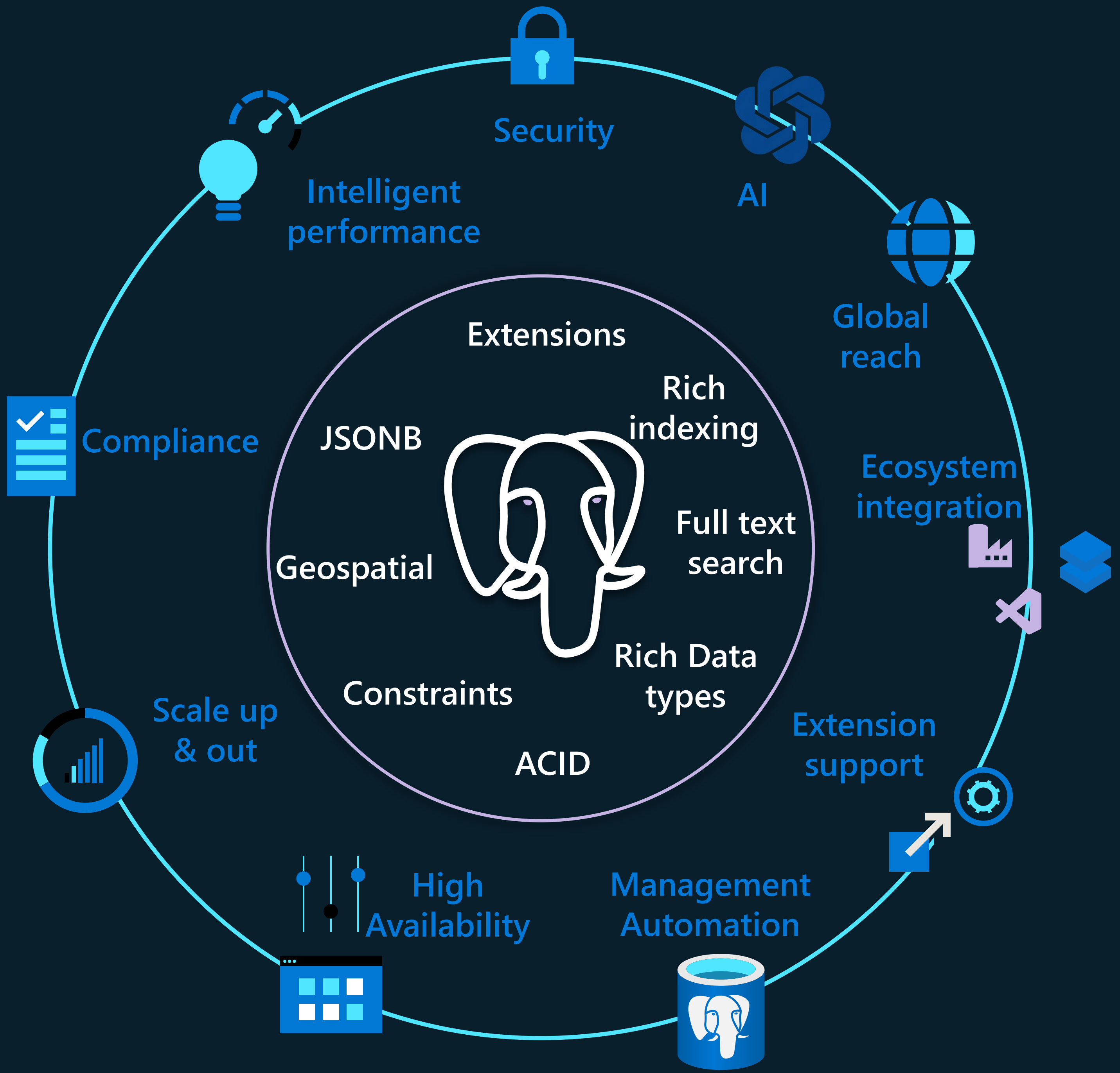
aka.ms/pg-diskann-demo

# 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



# Vectors: Important in the AI Apps

[ 1, 3, 5, 4, .. ]

[ 2, 8, 1, 0, .. ]

[ 9, 2, 5, 6, .. ]

**Raw Data** → **Deep Learning Model** → **High-dimensional Vectors** → **Vector Similarity Search**

Vector similarity search empowers Generative AI apps

**Recommendation**

**Question Answering**

**RAG apps**

# GraphRAG Solution Accelerator for Postgres

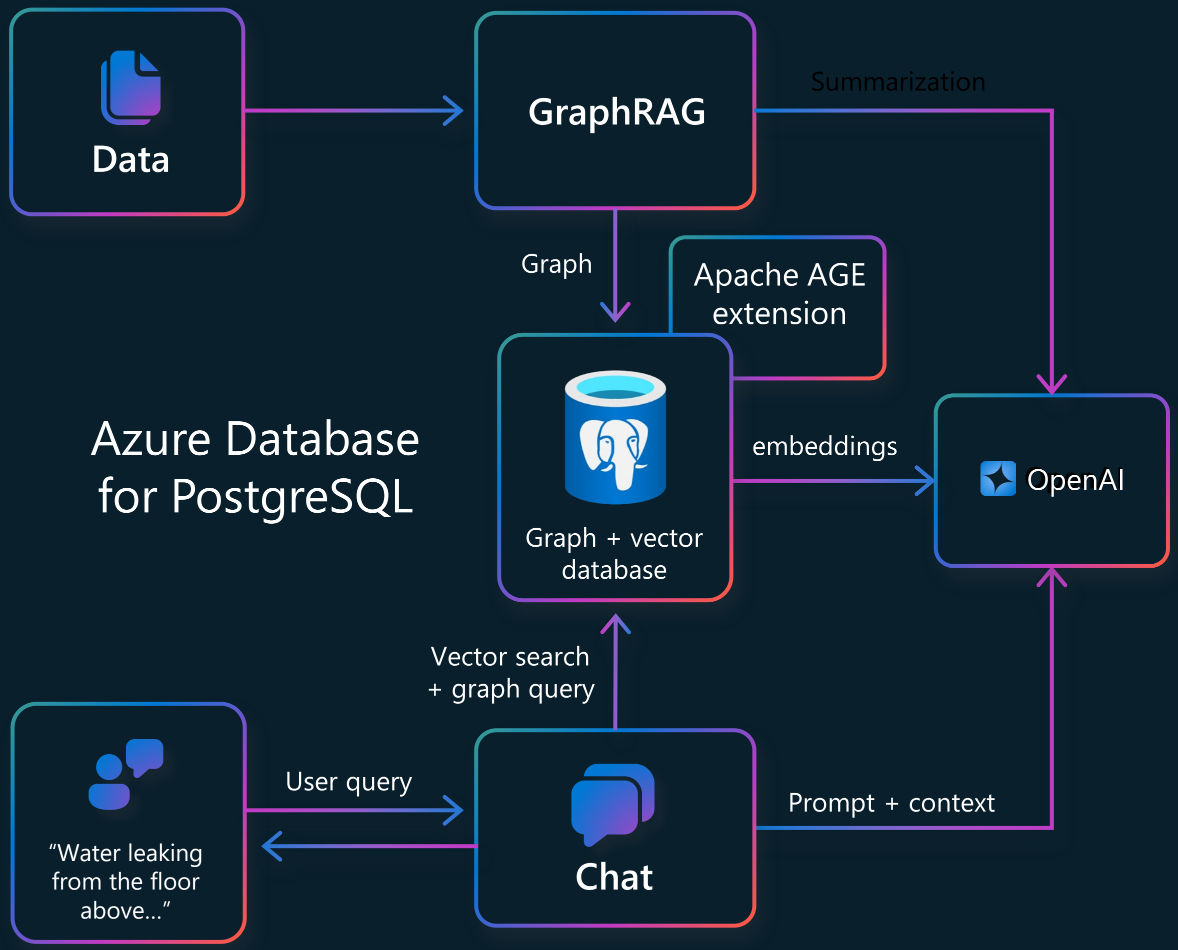
Preview

## Overview

- Legal Research Copilot app
- U.S. Case Law dataset (0.5 million cases)

## Available Now!

- Blog: aka.ms/pg-graphrag
- Repo: aka.ms/pg-graphrag-repo




This Lab: aka.ms/pg-ai-demo


"AI isn't the future,
It's already here."

Microsoft

# Talks by our Microsoft team

**Training: Developing RAG Apps with Azure Database for PostgreSQL & GraphRAG**

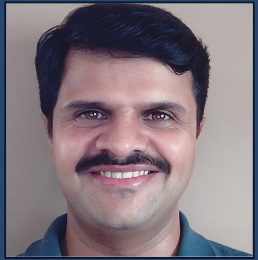
Varun Dhawan

Wed 5 Mar | 9:00

**Hacking Postgres Executor For Performance**

Amit Langote

Thu 6 Mar | 11:30



**Graph databases, PostgreSQL and SQL/PGQ**

Ashutosh Bapat

Thu 6 Mar | 14:00

**Unleashing the Power of Azure Database for PostgreSQL Flexible Server**

Shriram Muthukrishnan

Thu 6 Mar | 14:00

**Keynote: All the Postgres Things at Microsoft**

Sujit Kuruvilla

Thu 6 Mar | 16:45

**Using Postgres to locate the best coffee near you**

Varun Dhawan

Fri 7 Mar | 10:45

**Postgres: ServerLESS is more?**

Nikhil Sontakke

Fri 7 Mar | 11:30

**Beginner's Guide to Partitioning vs. Sharding in Postgres**

Claire Giordano

Fri 7 Mar | 14:45

Free
Socks
@ Booth

Microsoft

లదావ్యనధ
நன்றி
നന്ദി

# Thank you

ಧನ್ಯವಾದಗಳು
धन्यवाद
આભાર
ਤੁਹਾਡਾ ਧੰਨਵਾਦ
धन्यवाद
ଧନ୍ୟବାଦ


## Varun Dhawan

Principal Product Manager @Azure Postgres

linkedin.com/in/varundhawan/

# Would Love to Hear From You!

Thank you for being part of this session! It means a lot. I'd love to hear your thoughts - what you found helpful, what could be better, or anything that stood out.

Session Survey – Your feedback helps me improve! It'll be available shortly.

Join the conversation on LinkedIn – Share your thoughts here: LinkedIn Post

Really appreciate your time and insights. Looking forward to learning from you! 😊