



Postgres for Search

Standard, Full-Text and Semantic Search.

13th June 2024

Ajit Gadge
Field CTO

Marc Linster, Ph. D.
EDB Fellow

Speakers



Ajit Gadge

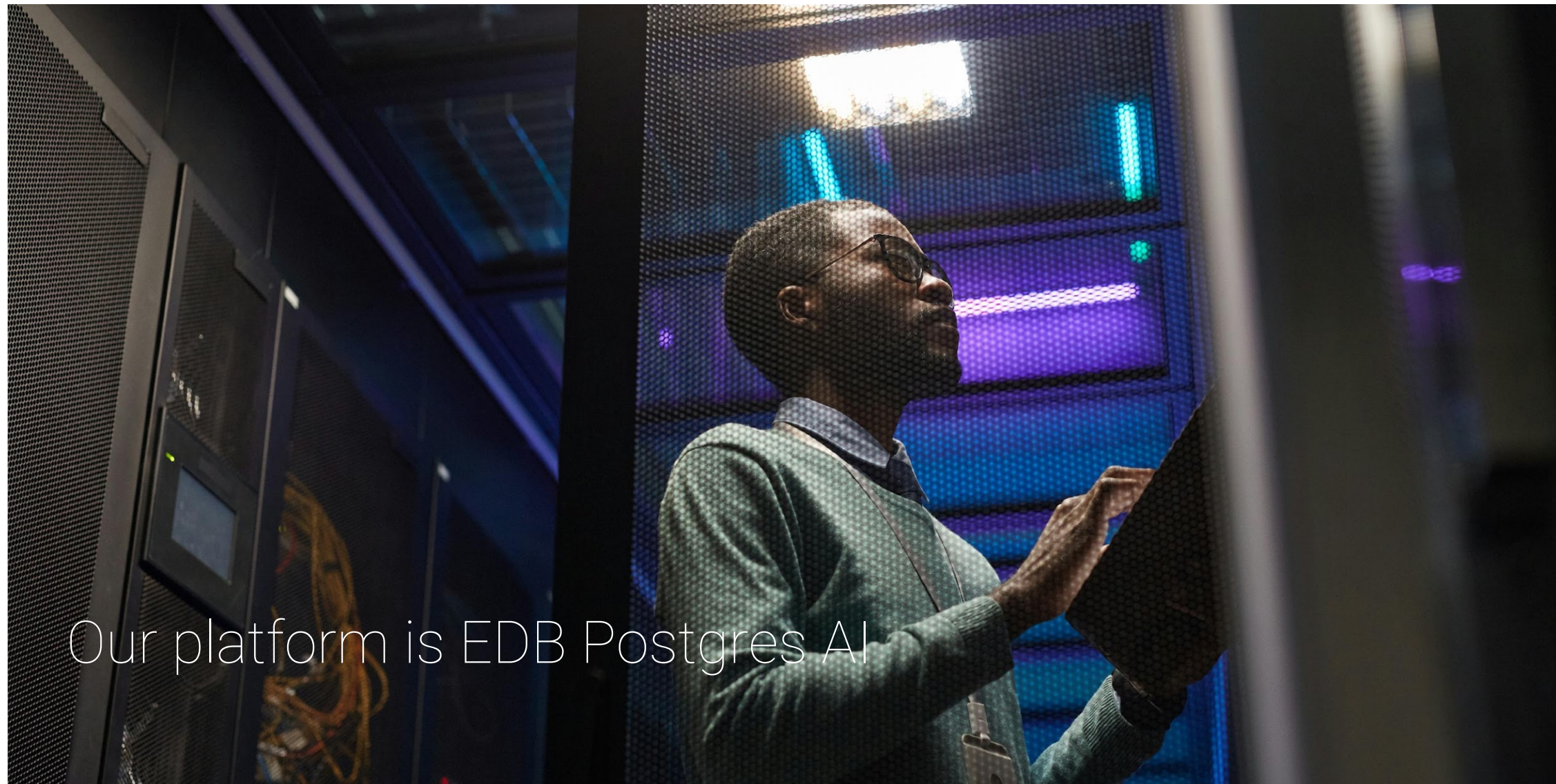
Field CTO, APJ, EDB



Marc Linster, Ph.D.

EDB Fellow





Our platform is EDB Postgres AI



A close-up, profile view of a woman with dark hair and black-rimmed glasses, looking intently at a screen. She is wearing a brown, ribbed turtleneck sweater. The lighting is warm and focused on her face, with the background being dark and out of focus. The text "Just solve it with Postgres" is overlaid in white on the lower left of the image.

Just solve it with Postgres





MISSION

Enable builders, data champions, and business leaders to realize and operationalize the near-infinite potential of Postgres and AI

VISION

Postgres and AI deliver real-world outcomes to drive economic and social prosperity

760+ employees

32 countries

#1 contributor to Postgres

20+ years of innovation



20+ years of Postgres innovation & adoption

- 3 Core Team members
- 7 Committers
- 9 Major Contributors
- 10 Contributors
- #1 site for desktop downloads

Agenda

- What are the typical challenges of searching data in different scenarios?
- Postgres has various options for Searching to address these challenges.
- Standard Search, Full-text Search and Semantic Search.
- Demo
- Future Trends with Semantic Search in AI with Postgre & EDB.
- Q & A



Challenges with Different Search Scenarios

Relevance

- Most relevant information and documents are at the top of the list
- Synonyms, Abbreviations, and phrases may cause the search to miss relevant results
- Stop words, noise words, and typos

Speed

- Quick responses keep users engaged

User Experience

- Easy to search with different options/filters
- Intent-based human-like search
- Modern Image-based or voice-based searching

Data Format

- Structure (Database, CSV, etc.) vs. unstructured (text, email documents etc.) format

Data Integration

- Data silos
- Unified view
- Data quality



Postgres Solutions for Search

Standard Search:

Basic keyword-based search that matches query terms exactly

Full-Text Search:

Searches entire text documents, indexing all words and phrases

Semantic Search:

Understands the intent and contextual meaning behind queries to deliver more relevant results



Standard Search

Advantages:

- Simplicity: The `LIKE` operator is straightforward to use and understand
- Basic Pattern Matching: It can match patterns within text data using wildcards like `%` and `_`

Limitations:

- Pattern: the `LIKE` pattern has to match the text, e.g., car vs. automobile; people vs. persons
- Performance: `LIKE` queries can be slow, especially when searching large datasets, as they often require full table scans
- Limited Functionality: Limited support for linguistic analysis, stemming, and relevance ranking
- Indexing Limitations: Indexing on text columns may not significantly improve performance for wildcard searches

When to use:

- Suitable for simple pattern matching or when full-text search features are not required
- When working with small datasets or when performance is not a critical concern



Full-Text Search with tsvector

Advantages:

- Fast and Efficient: Full-text search is optimised for searching large volumes of text data and can provide faster results compared to `LIKE` queries
- Linguistic Analysis: Provides support for stemming, ranking, and linguistic analysis, allowing for more accurate and relevant search results
- Index Support: tsvector indexes with GIN can significantly improve search performance, especially for complex queries

Limitation :

- Learning Curve: Requires understanding of tsvector data types and GIN or GIST indexes
- Setup Overhead: Initial setup and configuration for full-text search indexing may require more effort than standard search
- Context-based searching for text requires additional effort and could be more efficient

When to use:

- Suitable for applications requiring advanced search capabilities, such as searching within large documents or implementing relevance ranking
- When performance is a critical consideration for searching large volumes of text data



Semantic Search / Similarity Search with pgvector

Advantages:

- Enhanced Relevance: Semantic search techniques such as cosine similarity or Euclidean distance can provide more nuanced relevance ranking than basic text matching.
- Similarity Matching: Allows for searching based on the similarity between text fragments, which can be useful for applications like spell checking, autocomplete, or recommendation systems.
- Context-based searching applications like Chat Bot can use this method
- Better matching: Recognizes that 'car' and 'automobile' mean the same thing

Limitation :

- Complexity: Implementing semantic search requires more effort and expertise than standard or full-text search.
- Resource Intensive: Calculating similarity scores for large datasets is resource-intensive, potentially impacting performance.
- Requires more space to store the vector data.

When to use:

- Suitable for applications requiring precise similarity matching or recommendation systems based on textual data.
- When relevance ranking based on semantic similarity is crucial for search results.
- Context-based searching with RAG and Generative AI



DEMO

Standard Search Vs Full-text Search Vs
Semantic Search

Searching the phrase 'PostgreSQL &
vector'



Future Trends With Semantic Search and AI

Context-Based AI Search:

Context-aware, more accurate search. Ex: Voice assistance

Personalize Search Experience:

Leverage user data and behaviour for a more personalised search for engaging users. Ex: Recommendation in E-Commerce Product that include personal search history and preferences

Multi-Model Search Capabilities:

A combination of Text, Images, Audio, and Video provides a diversified search. Ex: HealthCare, doctors search medical data using voice and receive images and text

Real-Time Search:

Faster and real-time search with predictive-based search. Ex : User searching news data on social platforms that also anticipate the predictive content

Fully integrated and easy to implement:

Easy to implement with diverse data and scale. Ex: All types of data (structured, unstructured, semi-structured) that can be queried in real-time, supported by an easy implementation.



What's next with AI @ EDB

1. AI will be everywhere
2. EDB Postgres AI makes it easier to create AI-enabled and AI infused applications
3. EDB Postgres AI provides an integrated platform to manage transactional, analytical and AI data



EDB POSTGRES AI PLATFORM

UNIFIED WORKLOAD MANAGEMENT

TRANSACTIONAL

ANALYTICAL

ARTIFICIAL INTELLIGENCE

SINGLE PANE OF GLASS ADMINISTRATION

HYBRID DATA ESTATE

INTELLIGENT OBSERVABILITY

ENTERPRISE SECURITY

HYBRID AND MULTI-CLOUD DEPLOYMENT

PUBLIC CLOUD
(MANAGED)

PRIVATE CLOUD
(SOFTWARE)

ON PREMISES
(APPLIANCE)

PLATFORM TOOLS AND SERVICES

MIGRATION
PORTAL

CONTINUOUS HIGH
AVAILABILITY

BACKUP AND
RECOVERY

EXTENSIBILITY

CSP INTEGRATIONS

DEVOPS TOOLING

KUBERNETES TOOLING

GENAI & LLM INTEGRATIONS

LAKEHOUSE INTEGRATIONS

Delivered with world-class
strategic partners:

carahsoft



Red Hat

IBM



NUTANIX

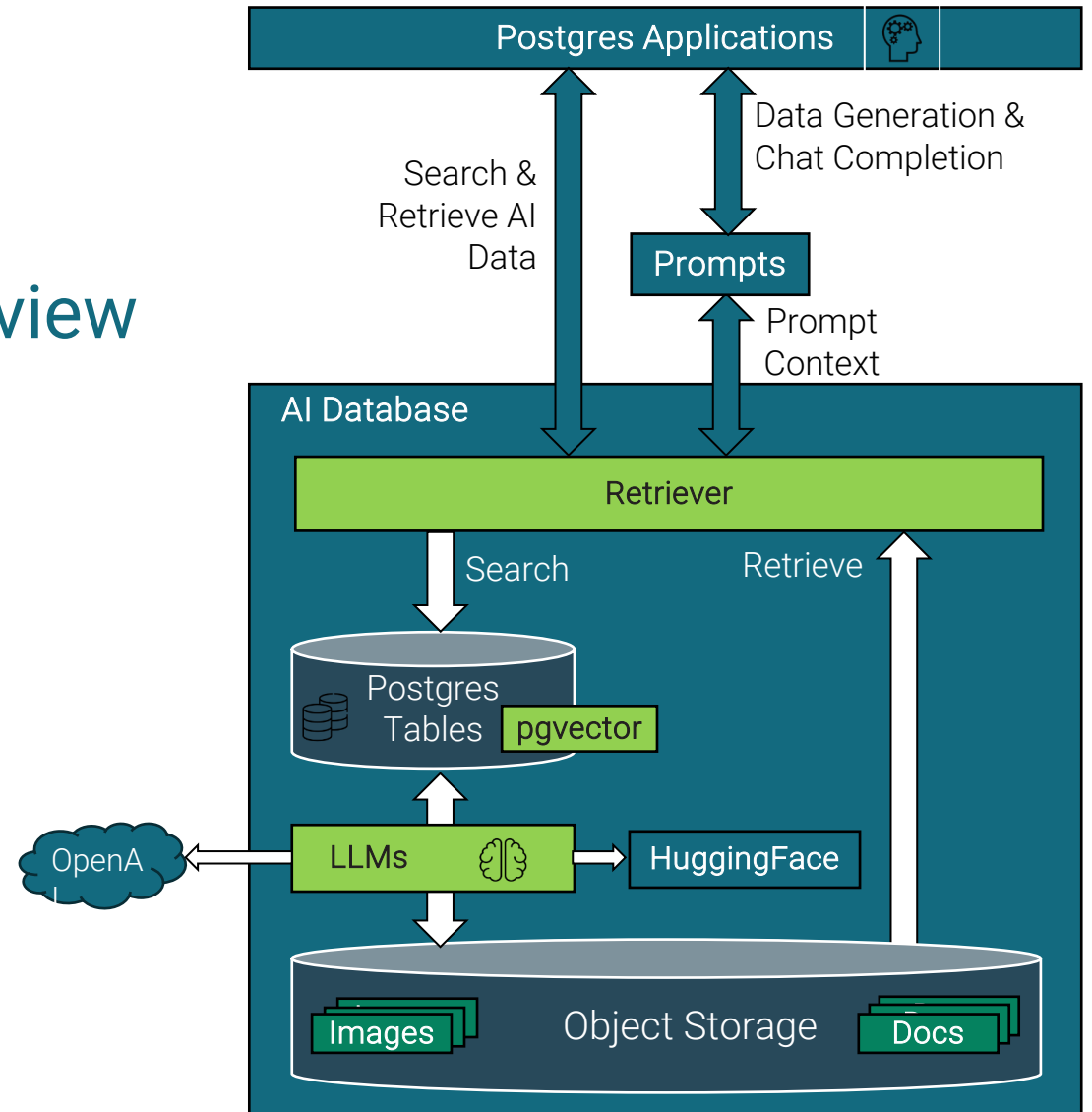


EDB Postgres AI - Tech Preview

Request Free Access at

<https://info.enterprisedb.com/pgai-preview>

```
SELECT pgai.create_pg_retriever(  
  'product_embeddings', -- Retriever name  
  'public', -- Schema  
  'product_id', -- Primary key  
  'text-embedding-3-small', -- embedding model  
  'text', -- data type  
  'products', -- Source table  
  ARRAY['product_name', 'description'], -- Columns to  
    vectorize  
  TRUE -- auto embeddings TRUE to set trigger  
);
```



Thank you

- Visit EDB Blogs to learn more:
 - Enhancing Search Capabilities with Postgres: From Standard to Semantic
<https://www.enterprisedb.com/blog/enhancing-search-capabilities-postgresql-standard-semantic>
 - RAG app with Postgres and pgvector
<https://www.enterprisedb.com/blog/rag-app-postgres-and-pgvector>
- Download EDB Postgres AI - Tech Preview
<https://info.enterprisedb.com/pgai-preview>

