
ODTUG
Kscope25
dallas - ft worth
june 15 - 19
Welcome

ODTUG Kscope25
dallas - ft worth june 15 - 19

@eaespinoza0

# Supercharging APEX with Oracle 23ai

## Enhance User Experience with Smart AI Tools


Erik Espinoza

Solutions Architect

VISCOSITY NORTH AMERICA

viscosityna.com

@ViscosityNa

# Why User Experience (UX) matters?

Users expect intuitive, responsive, and intelligent applications

AI tools enable smart and more personalized interfaces

# Why APEX or Oracle 23ai?

- Native support for AI features
  - Use natural language to interact with your data
  - Similarity searching (Vector store)
- Seamless integration with APEX components
  - The new Search region type
  - The new Generate Text Dynamic Action
  - The new Show AI Assistant Dynamic Action
  - Declarative support for Vector Stores
- Brings ”smart touch” to your apps

# System Prompt, Vector Store, Embeddings, Cosine … what is all this?

- In simple terms, a **system prompt** is like giving your AI assistant a **job description** and **set of rules** before it starts working. It's **not visible to the end user**, but it shapes how the AI behaves.
- A **vector store** is like a **specialized search engine** that finds things based on **meaning** rather than exact words.
- **Embeddings** are how AI turns words, sentences, or other data into **numbers** — specifically, into **vectors** (ordered lists of numbers). These numbers capture the **meaning** of the data in a way that computers can understand and compare.
- **Cosine search** (or **cosine similarity**) is a method to compare **how similar two vectors are** based on the **angle between them**, not their length.
- **RAG (Retrieval-Augmented Generation)** is an AI pattern that **combines search and generation**. It helps large language models (like ChatGPT) give **more accurate and up-to-date answers** — even if they weren't trained on specific information.

# Oracle 23ai Features – Embeddings

- Let's you **store and search vector embeddings** (like those from OpenAI or Hugging Face) in Oracle Database.
- Enables **AI-powered similarity search** (e.g., find similar documents, images, invoices).
- Ideal for **semantic search**, **recommendation systems**, or **AI-enhanced applications**.

```
SELECT t.*
  FROM ap_invoice_embeddings t
 ORDER BY vector_distance(vector_embedding,
                          vector_embedding(
                            ALL_MINILM_L12_V2 USING 'mechanical parts' AS DATA),
                          COSINE)
 FETCH FIRST 10 ROWS ONLY;
```

# Oracle 23ai Features – Select AI

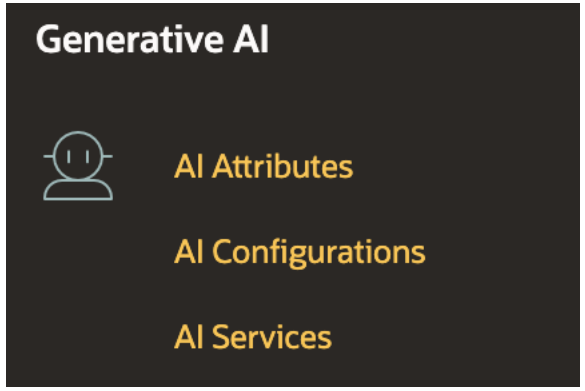
- Allows you to run **natural language queries** (e.g., “Show me all orders over $100”) directly in SQL.
- Oracle uses an **LLM (like OpenAI)** to convert plain English into optimized SQL automatically.
- Useful for **non-technical users** or **chatbot integrations**.

```
select ai 'how many invoices were received this month';
```

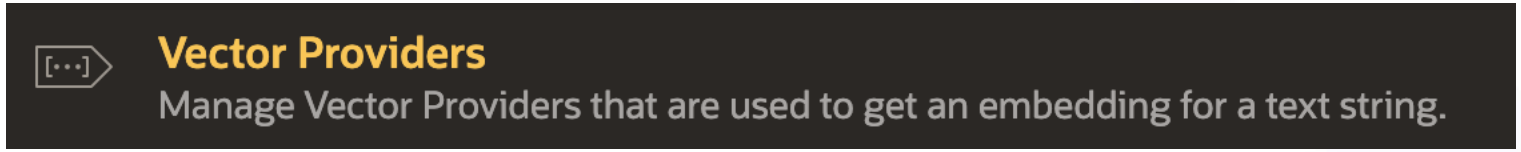
```
select ai chat 'what is the most expensive item we purchased this month?';
```

# APEX AI Features

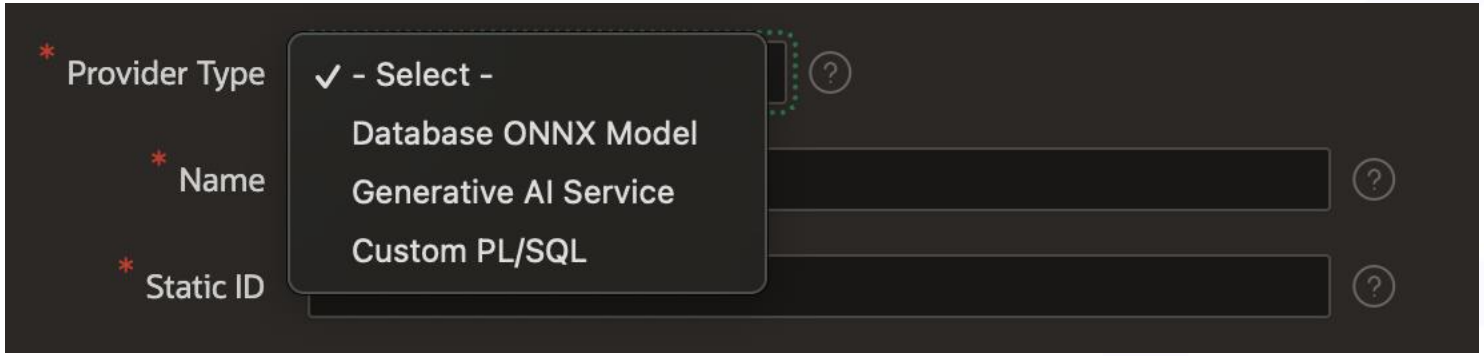
## Generative AI

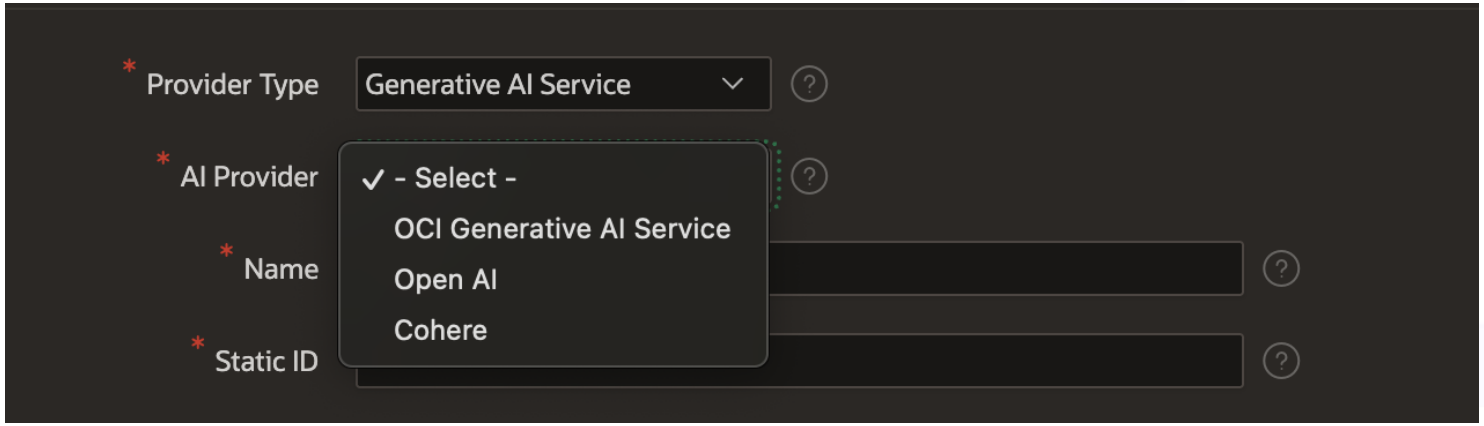



## Vector Providers



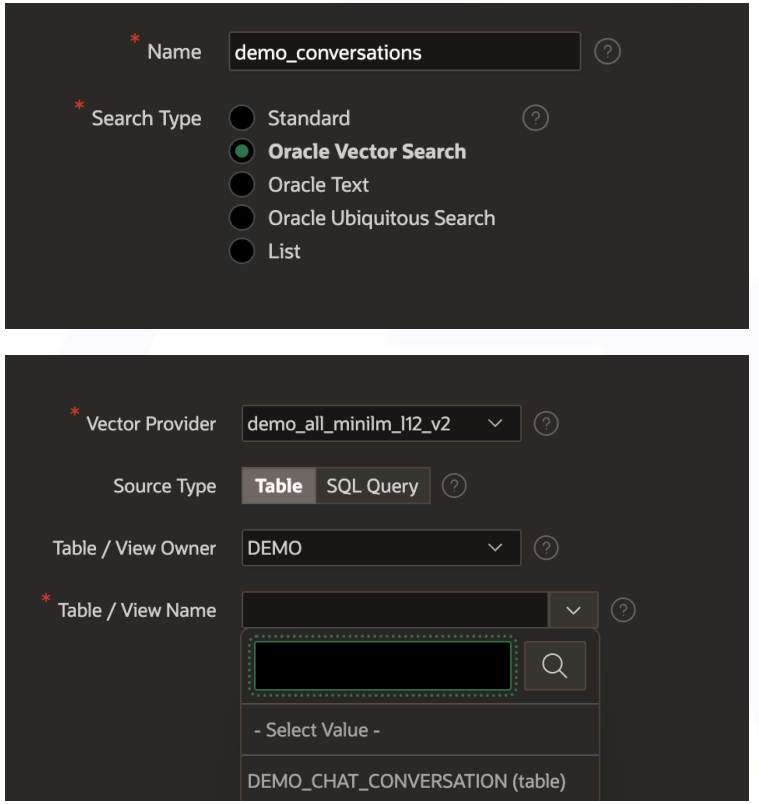
# APEX AI Features – Vector Providers



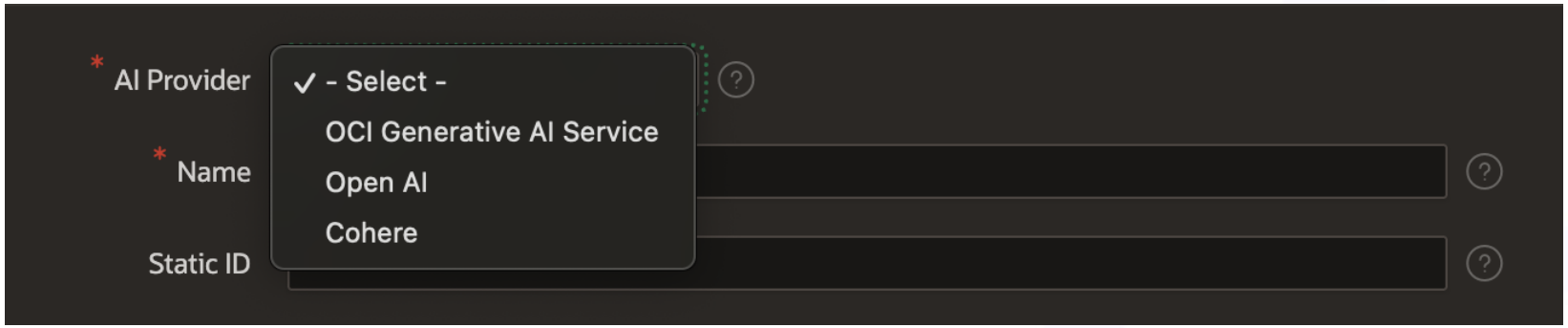



# APEX AI Features – Search Configurations

- Allows an APEX application to define declaratively configurations to be used in components that support similarity search out-the-box.
- There are specific PL/SQL APIs available that leverage this configurations if you have more custom use cases through the **APEX_SEACH** API.



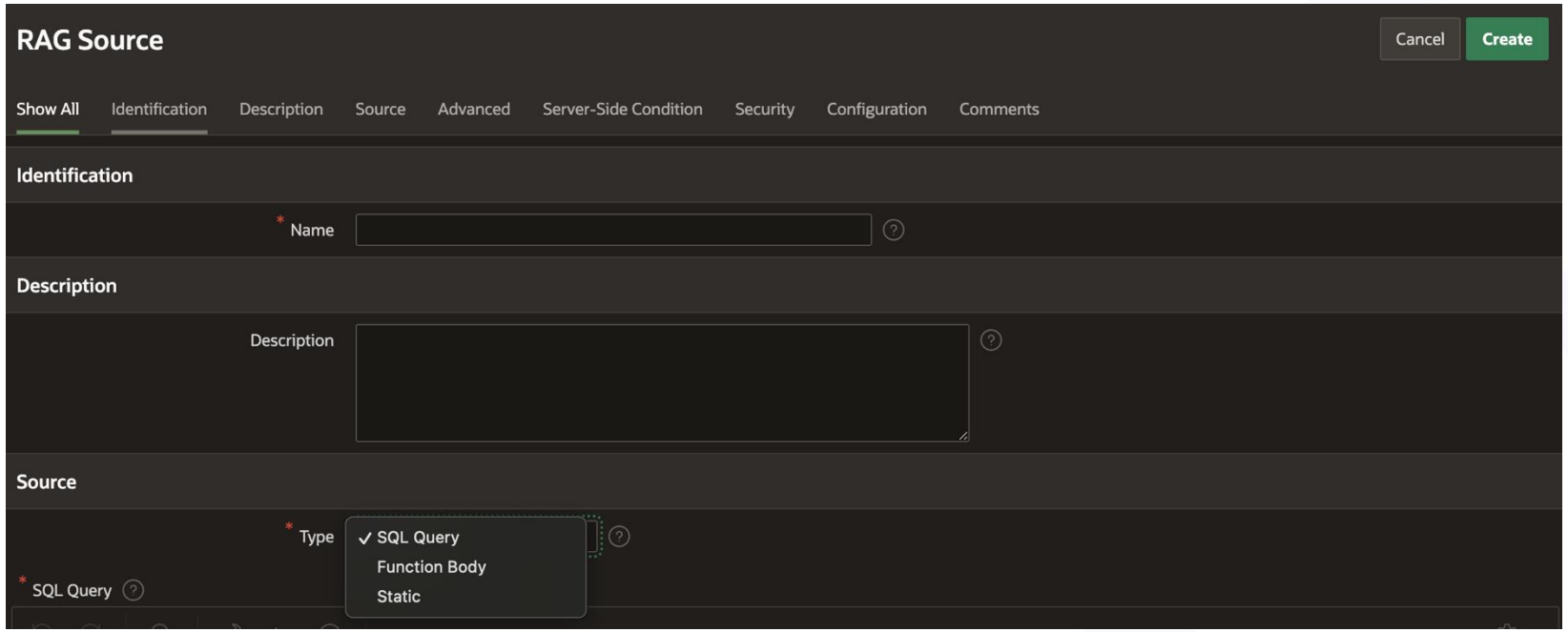
# APEX AI Features – Generative AI




- Support for an abstraction to commercial LLM providers such as OCI Generative AI and OpenAI.
- Configured services available through the PL/SQL API APEX_AI
- Stored at Workspace level, available to other applications.
- Generative AI services are included in your application export

Learn more at APEX Documentation

# **APEX AI Features –** Retrieval Augmented Generation (RAG)



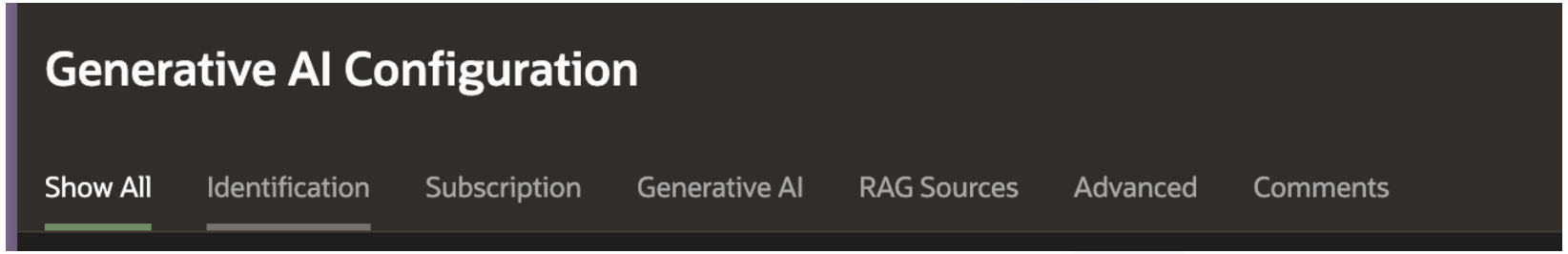

Allows APEX Applications to provide context to the AI model for improved generation.

Sources used by Generative AI to improve response quality, such as:

- SQL Queries
- PL/SQL Functions
- Static

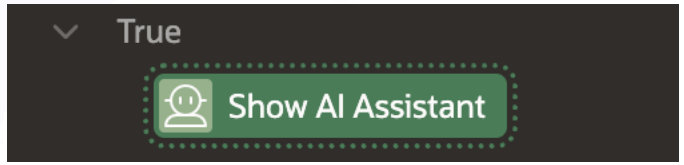
# APEX AI Features – AI Configurations

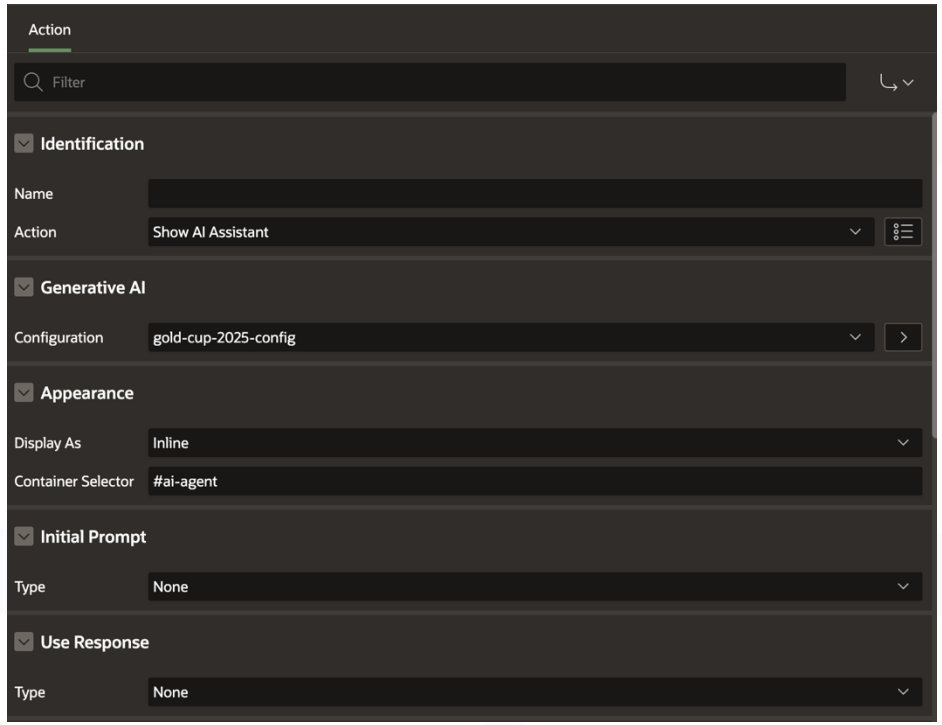
Allows APEX Applications to define many system prompts for specialized use cases, to associate specific RAG sources and contextualize this for specific pages or reports in your application.



# **APEX AI Features –** Show AI Assistant Dynamic Action

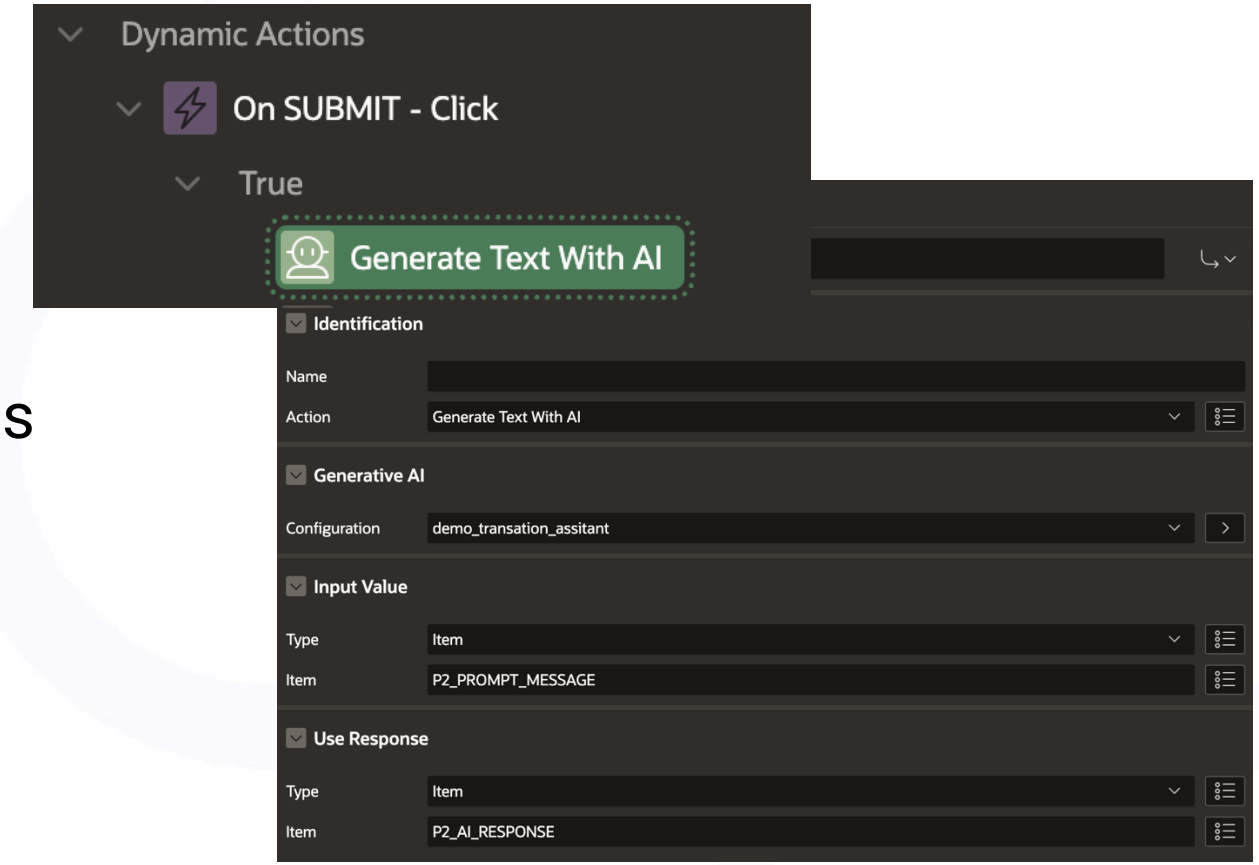
- What it is
  - Conversational UX/UI in APEX
- Use Cases
  - Help Desk
  - Guided Navigation
  - Form Assistants
- Tools
  - Large Language Model Service
  - APEX AI Dynamic Action



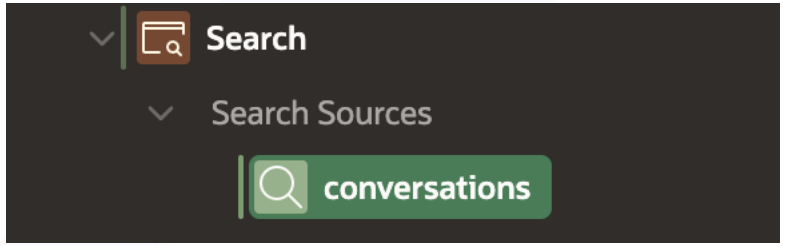
# **APEX AI Features –** Generate Text Dynamic Action

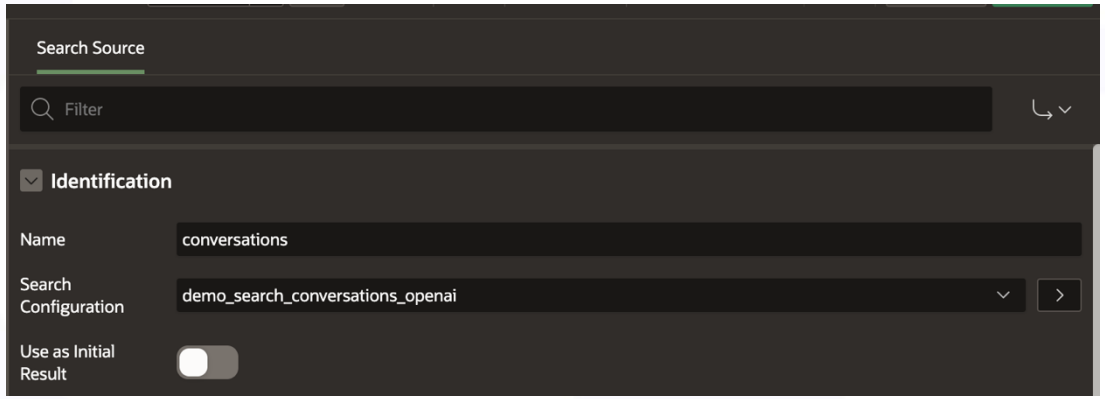
- What it is
  - Generative Text
- Use Cases
  - Translation
  - Generation of Product Descriptions
  - Summarization
- Tools
  - Large Language Model Service
  - APEX AI Dynamic Action



# **APEX AI Features** – Search

- What it is
  - Similarity Search
- Use Cases
  - Search with Natural Language
- Tools
  - Vector Store
  - APEX AI Region



# **Step by Step –** Grants

Just need a few grants before get started

```
GRANT EXECUTE ON DBMS_CLOUD TO DEMO;
GRANT DB_DEVELOPER_ROLE TO DEMO;
GRANT EXECUTE ON DBMS_CLOUD_AI TO DEMO;
GRANT CREATE MINING MODEL TO DEMO;
```

# **Step by Step –** Cloud Credentials

This is how the database AI features will be able to communicate with LLM services such as OCI Generative AI's models or OpenAI's model's.

```
BEGIN
    DBMS_CLOUD.create_credential(
    credential_name => 'MY_CREDENTIAL',
    user_ocid => 'ocid1.user. ...',
    tenancy_ocid => 'ocid1.tenancy. ...',
    private_key => q'~-----BEGIN PRIVATE KEY-----
...
...
-----END PRIVATE KEY-----
OCI_API_KEY~',
    fingerprint => 'aa:bb:33 ...');
END;
/
```

# **Step by Step –** Load Embedding Model

Loading a local embedding model allows the database to create embeddings faster for text, reduces latency and improves speed.

Use **DBMS_VECTOR** to load the embedding model.

```
DECLARE
  model_source BLOB := NULL;
BEGIN
  model_source := DBMS_CLOUD.get_object(credential_name => 'MY_CREDENTIAL',
    object_uri => 'https://objectstorage.us-ashburn-1.oraclecloud.com/n/<namespace>/b/<bucket name>/o/<filename>.onnx');
  DBMS_VECTOR.load_onnx_model('<model name<', model_source);
END;
/
```

# **Step by Step –** Create a table with VECTOR data type

The support for the new type VECTOR allows to store embeddings along with your data to enable similarity searching.

```
CREATE TABLE my_content(
  content_id NUMBER,
  content BLOB,
  content_embedding VECTOR
);
```

# **Step by Step –** Query Data using Similarity Search

The support for the new type **VECTOR** allows to store embeddings along with your data to enable similarity searching.

```
SELECT *
  FROM my_content
 ORDER BY vector_distance(content_embedding,
                          vector_embedding(ALL_MINILM_L12_V2 USING 'Your natural language query' AS DATA),
                         COSINE) ASC
FETCH FIRST 10 ROWS ONLY;
```

# **Step by Step –** Enabling SELECT AI with OCI Generative AI

Create a profile which saves the configuration needed for Oracle 23ai to use OCI Generative AI service to enable SELECT AI

```
BEGIN
  DBMS_CLOUD_AI.CREATE_PROFILE(
    profile_name    => '<PROFILE_NAME>'
   ,attributes      => json_object(
                         'credential_name' VALUE '<CREDENTIAL_NAME>'
                        ,'model'           VALUE '<MODEL_NAME>'
                        ,'provider'        VALUE 'oci'
                        ,'object_list'     VALUE json_array(
                                                   json_object(
                                                     'owner' VALUE '<SCHEMA_NAME>'
                                                   )
                                                 )
                        ,'object_list_mode' VALUE 'all'
                       )
   ,status          => 'enabled'
  );
END;
/
```

# **Step by Step –** Use SELECT AI

Before you can run any SELECT AI statement you MUST tell your SESSION wh ich AI profile to use.

```
BEGIN
  DBMS_CLOUD_AI.SET_PROFILE('<PROFILE_NAME>');
END;
/
```

```
SELECT AI 'how many vendors are there in the database?';
```

Yes, you don't need **FROM DUAL!**

# Any Questions?

erik.espinoza@ViscosityNA.com

@<username>

VISCOSITY NORTH AMERICA

All Viscosity hosted webinar recordings and conference sessions slide decks will be posted to www.OraPub.com for free and paid members!

# Follow Us Online!

- Facebook.com/ViscosityNA
- LinkedIn.com/company/Viscosity-North-America
- @ViscosityNA
- Viscosity North America
- @Viscosity_NA


ODTUG
Kscope25
dallas - ft worth
june 15 - 19
Don’t Forget To Fill Out Your Evals