



Understanding Oracle 23ai's Vector Search

Janis Griffin

Senior Database Consultant, Quest Software



Janis Griffin
Oracle ACE Pro

Janis Griffin

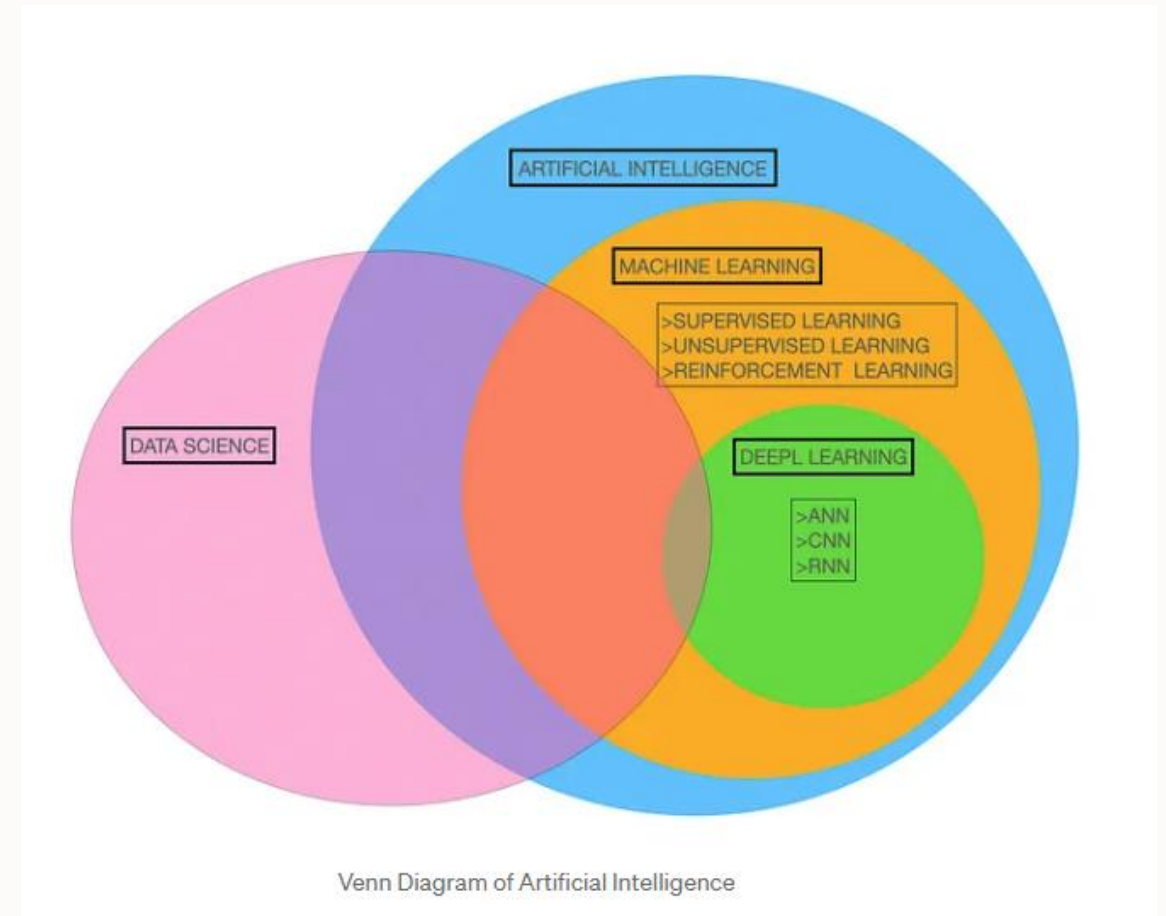
Senior Database Consultant
Quest Software
April 2, 2025

Machine Learning (ML)

Key Concepts

- Considered a subset of AI unlike traditional programming
 - ML algorithms learn by analyzing data patterns
 - > Data can be numbers, text, images, & audio
- ML Models are mathematical representations
 - Of real-world processes
 - > When exposed to more data,
 - > Can refine predictions
 - > Make further decision making

[More on Machine Learning](#)



What are Vectors

Vectors are building blocks representing & manipulating data in ML models

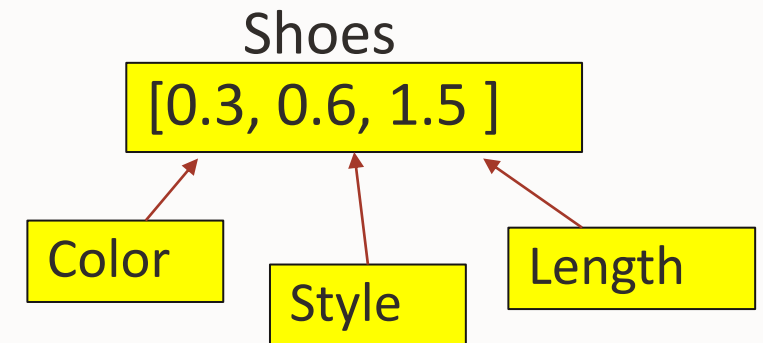
Need to understand Vector Search

- It's a technique used in information retrieval for machine learning
- Looks at items in large datasets
- Stores & groups items based on their vector representation – called **vector embeddings**

Vector embeddings are strings of numbers

- That correspond to the many attributes of an item
 - Can be a document, image, audio, or video file
- Vector embeddings are arrays of real numbers
 - Can be 100s or 1000s of dimensions
- The process for generating a vector for a data object is **vectorization**

Vector Search = Similarity Search = Nearest Neighbor Search



Where are Vectors Stored & Accessed

Vectors are stored in databases that support vectors: vector databases

- Optimized for similarity searches
- Designed to contain many dimensions with millions or billions of vectors
 - Which are the result of embedded process & machine learning models
- Oracle stores vector embeddings as a VECTOR data type

Vector databases use advance indexing techniques

- **Inverted File Index (IVF), Hierarchical Navigable Small World (HNSW)**, Locality-Sensitive Hashing (LSH), Tree-based indexes (like ANNOY), Product Quantization (PQ), & Hash-based indexes
 - Each type offers different trade-offs
 - > Search speed vs. accuracy depending on the application & data

Oracle recently added, new Hybrid Vector Index

- Combination of Oracle Text index & Vector index on your unstructured data

```
CREATE HYBRID VECTOR INDEX moviev_hybrid_idx on moviev(summary)
PARAMETERS ('model ALL_MINILM_L12_V2 vector_idxtype HNSW');
```

- [hybrid-vector-index-the-combination-of-full-text-and-semantic-vector-search](#)

Large Language Models (LLMs)

LLMs are designed to understand & generate human language

- Trained on large amounts of Text data
- Consists of million/billions of parameters learned from training data
- The underlying transformer is a set of neural networks ([What's a Neural Network](#))
 - > That consist of an encoder & decoder
 - > Encoder extracts meanings from a sequence of text
 - Understands relationships between words & phrases
 - > Decoder takes the processed data & steps through it to produce output

Transformer LLMs are capable of self-learning

- Through this process, transformers learn basic grammar, languages & knowledge
- [Transformer Architecture in LLMs](#)

[More Information on LLMS by AWS](#) / [Key Takeaways From Azure](#)

What is RAG & how does it work?

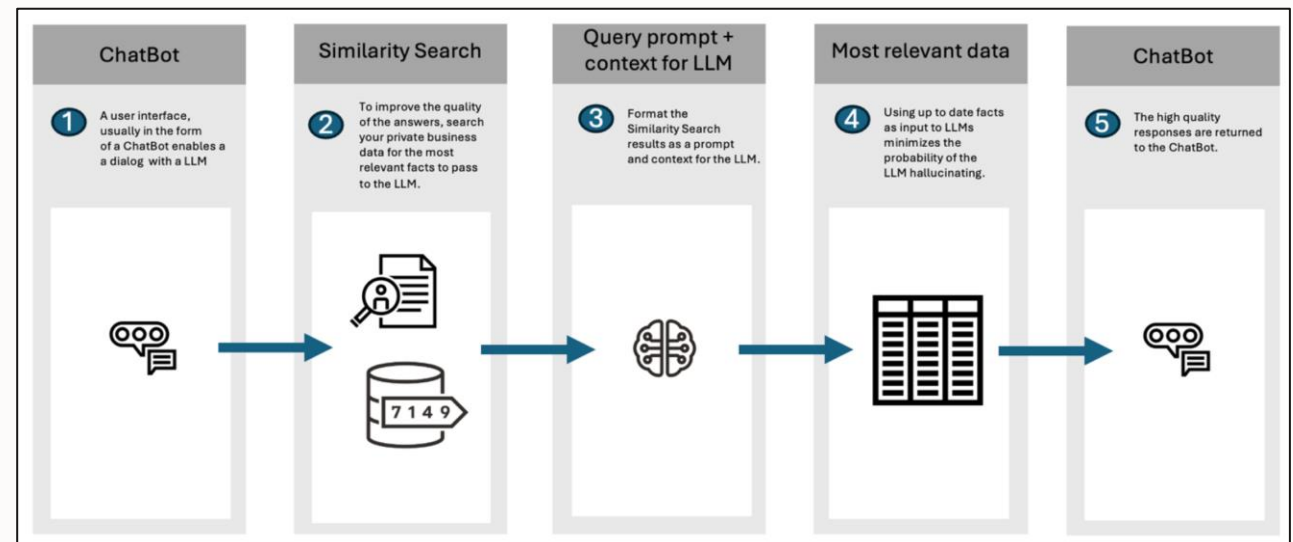
Retrieval-augmented Generation (RAG) is a technique within ML

- Gets relevant information from external knowledge bases during generation
 - User query: User provides a question or input
 - Query encoder: Query is converted into a dense vector representation
 - Retrieval process: System searches the external knowledge base using the query vector
 - > Identifies the most relevant documents based on similarity scores
 - Contextual input: Retrieved documents with original query are input to the language model
 - Response generation: Language model generates response
 - Using both the retrieved information & its own knowledge

Benefits of RAG over standard LLM

- By accessing external information
 - it provides more up-to-date info
- RAG can use different knowledge sources
 - by adjusting the retrieval process

How RAG Works



Oracle Database Is a ML Engine

Oracle acquired ML technology in 1999 from Thinking Machines Corp.

- Rational: Data is big - algorithms are small
 - So move algorithms to the data – don't move the data to the algorithms

Oracle 12.2 introduced more advanced & integrated ML features

- Internally, Oracle uses ML for Adaptive Plans & 19c Automatic Indexing features
- Fully integrated in 19c+ with SQL, R (via OML4R) & Python (via OML4Py)
 - Over 30 free algorithms

Oracle 23ai

- Uses SQL to build models & run ML directly on business data
 - Implemented VECTORS as a data type
 - > Vector_col_name vector(# of dimensions, format, dense) or (*.*)
 - No need to move data to separate ML engine if you have Oracle23ai
- Supports Large Language Model (LLM) & Retrieval-Augmented Generation (RAG)
 - > has ONNX format support, API & SQL Support of vectors
- Extends applications for both structured & unstructured data types
 - By including popular LLMs with your own data sets with RAG

Oracle AI Vector Search Details

Needs Oracle Version 23.4 or higher (Version 23.6.0.24.10)

- Works in 23ai Free, Autonomous database, Oracle engineered systems

Similarity searches leverage data semantics

- By generating vectors using transformer models
- Goes beyond basic text matching to get data with more meaning & context

Works on Unstructured data

Uses Hugging Face's all-MiniLM-L12-v2 model in ONNX format

- Can be loaded directly into the database via `DBMS_VECTOR.LOAD_ONNX_MODEL`
- All-MiniLM-L12-v2 is a sentence transformer model
 - > Used for optimizing natural language processing (NLP) tasks
- [AI Vector Search Users Guide](#)

ONNX = open-source format designed for ML models - ensures cross-platform compatibility

AI Vector Search uses Distance metrics

To measure semantic similarity between text documents via Vectors

Cosine similarity metric

- When not explicitly specified (DEFAULT)
- Focuses on the angle (direction) between vectors & ignores their magnitude (size)

Euclidean distance metric

- This metric measures the straight-line distance between them
- Considers both magnitude & direction

Euclidean squared,
Hamming distance,
Manhattan distance,
DOT product
[Supported Distance Metrics](#)

VECTOR_DISTANCE Function

- Oracle provides a VECTOR_DISTANCE function to calculate the distance between two vectors

Advance indexing techniques

- **Inverted File Index (IVF), Hierarchical Navigable Small World (HNSW),**
 - Each type offers different trade-offs
 - > Search speed vs. accuracy depending on the application & data

Steps to use all-MiniLM-L12-v2 – cont.

```
DECLARE
  ONNX_MOD_FILE VARCHAR2(100) := 'all_MiniLM_L12_v2.onnx';
  MODNAME VARCHAR2(500);
  LOCATION_URI VARCHAR2(200) := 'https://adwc4pm.objectstorage.us-ashburn-1.oci.customer-
oci.com/p/eLddQappgBJ7jNi6Guz9m9LOtYe2u8LWY19GfgU8flFK4N9YgP4kTlrE9Px3pE12/n/adwc4pm/b/OML-Resources/o/';
BEGIN
  DBMS_OUTPUT.PUT_LINE('ONNX model file name in Object Storage is: '||ONNX_MOD_FILE);
  -----
  -- Define a model name for the loaded model
  -----
  SELECT UPPER(REGEXP_SUBSTR(ONNX_MOD_FILE, '[^.]+')) INTO MODNAME from dual;
  DBMS_OUTPUT.PUT_LINE('Model will be loaded and saved with name: '||MODNAME);
  -----
  -- Read the ONNX model file from Object Storage into
  -- the Autonomous Database data pump directory
  -----
  BEGIN DBMS_DATA_MINING.DROP_MODEL(model_name => MODNAME);
  EXCEPTION WHEN OTHERS THEN NULL; END;

  DBMS_CLOUD.GET_OBJECT(
    credential_name => 'V_CRED',
    directory_name => 'DATA_PUMP_DIR',
    object_uri => LOCATION_URI||ONNX_MOD_FILE);
  -----
  -- Load the ONNX model to the database
  -----
  DBMS_VECTOR.LOAD_ONNX_MODEL(
    directory => 'DATA_PUMP_DIR',
    file_name => ONNX_MOD_FILE,
    model_name => MODNAME);

  DBMS_OUTPUT.PUT_LINE('New model successfully loaded with name: '||MODNAME);
END;
```

```
BEGIN
  DBMS_CLOUD.CREATE_CREDENTIAL(
    credential_name => 'V_CRED',
    username => 'MOVIESTREAM',
    password => 'yourPassWordhere##');
END;
```

Load the model to
the database from
Object Storage

[Pre-built Embedding Generation Model](#)

[by Sherry LaMonica](#)

Steps to use all-MiniLM-L12-v2

Validate that model is in the database

```
SQL> select model_name, algorithm, mining_function
  2  from user_mining_models
  3  where model_name='ALL_MINILM_L12_V2';
```

MODEL_NAME	ALGORITHM	MINING_FUNCTION
ALL_MINILM_L12_V2	ONNX	EMBEDDING

```
SQL> █
```

Create vector embeddings

- With function VECTOR_EMBEDDING

```
select VECTOR_EMBEDDING(ALL_MINILM_L12_V2 USING 'How now brown cow' as DATA) as embedding_col;
```

```
SQL>
SQL> select vector_embedding(ALL_MINILM_L12_V2 USING 'How now brown cow' as DATA) as embedding_col;
```

EMBEDDING_COL
[-9.78076085E-003,3.92418988E-002,-3.00500635E-002,-4.72754793E-004,9.2594428E-003,2.0322552E-002,-8.62419978E-003,5.93701713E-002,5.01224771E-003,-1.38688786E-002,-2.17656177E-002,-6.02688082E-002,-1.0191308E-001,-2.77633313E-002,-3.10682524E-002,-3.49388868E-002,3.89614478E-002,-2.70192623E-002,-6.73687384E-002,-1.29006609E-001,-1.07883021E-001,-2.0933127E-002,-4.70857322E-002,2.47860178E-002,-8.00913945E-002,-3.00575607E-002,1.29399952E-002,2.91546676E-002,-2.45270934E-002,-1.06916383E-001,8.7348707E-003,-6.13076948E-002,1.31290965E-002,2.16532429E-003,-9.11920

Steps to use all-MiniLM-L12-v2 – cont.

Perform a similarity search

- Create table & Load data

```
create table vdata (id number not null, info varchar2(120), v vector(*,*));
```

```
insert into vdata (id, info) values (1,'San Francisco is in California.');
```

```
insert into vdata (id, info) values (2,'San Jose is in California.');
```

```
insert into vdata (id, info) values (102,'Mini Coopers are small.');
```

```
insert into vdata (id, info) values (910,'Oracle CloudWorld Sao Paulo is on 4 April 2024');
```

```
insert into vdata (id, info) values (912,'Oracle CloudWorld Singapore is on 16 April 2024');
```

...

Update table 'vdata' with vector embeddings

```
update vdata set v=VECTOR_EMBEDDING(ALL_MINILM_L12_V2 USING info as data);
```

```
SQL> select id, info, v from vdata where rownum <6;
```

ID	INFO	V
402	Gunma is in Kanto.	[3.17411907E-002,4.85073635E-003,-2.07891446E-002,2.5316935E-002,
403	Saitama is in Kanto.	[-8.80532898E-003,-8.98086205E-002,3.48317437E-002,-3.20804641E-002,
404	Chiba is in Kanto.	[5.99674508E-003,-1.3688487E-002,-2.67483555E-002,5.90551691E-003,
405	Tokyo is in Kanto.	[4.96507585E-002,2.06358498E-003,2.67245919E-002,-1.53107466E-002,
324	Kangaroos can hop.	[7.44743794E-002,-8.70305393E-003,-2.96329465E-002,2.60136072E-002,

Similarity Query Example

Do a similarity search for 'Oracle' string

- Look up the ID of 'Oracle' description

```
SELECT ID, INFO FROM VDATA  
WHERE ID = 909;
```

```
select id, info from vdata where id <> 909  
order by vector_distance(v, to_vector(  
1  
6  
0  
4  
3  
2  
0  
7  
8  
6  
0  
2  
3  
794291E-002,2.00050343E-002,4.37186025E-002,2.38271207E-002,2.03080103E-002,5.3220503E-002,8.81564692E-002,-3.48353991E-003,-4.53762151E-002,-4.43401895E-002,  
-3.25705446E-002,7.52080157E-002,-7.63299912E-002,7.64165223E-002,2.4561299E-002,7.42148682E-002,-6.21137908E-003,2.9130647E-002,1.81009881E-002,8.32014009E-0  
02,-3.61453509E-003,-2.68469024E-002,-4.79419455E-002,-9.98866837E-003,4.05372307E-002,-3.71959689E-003,-8.74776393E-002,1.44836461E-033,2.97428835E-002,-8.21  
415409E-002,-2.42466442E-002,-3.5681136E-002,1.66083239E-002,2.26499815E-003,3.8979169E-002,8.02413449E-002,-6.48359433E-002,4.99817133E-002,1.55769009E-002,4  
.31471467E-002,8.51256028E-002,-6.88268542E-002,-1.35851214E-002,1.2305107E-002,3.36682573E-002,3.97662632E-002,-7.80334473E-002,1.39786929E-001,5.28790355E-0  
02,-3.76427025E-002,-7.84885697E-003,2.98213889E-003,-5.00591099E-002,-1.58992456E-003,4.57127988E-002,-2.14038044E-002,9.39843059E-003,1.82048324E-002,-1.309  
14539E-001,1.01908715E-002,-9.77061614E-002,9.93210226E-002,2.58489046E-002,-2.7765099E-002,2.64408886E-002,-9.72326919E-002,-1.22394092E-001,-2.32343823E-002  
,9.02670994E-003,-2.16798875E-002,-4.44828644E-002,4.03261781E-002,-6.16931869E-003,5.20048626E-002,3.464045E-002,1.51240509E-002,-1.55729288E-002,-9.1569826E  
-002,-1.18203303E-002,-4.72370684E-002,6.32990431E-003,2.41887532E-002,6.49466589E-002,8.09864551E-002,-1.76854879E-002,-2.98077278E-002,-2.60046925E-002,-4.0  
2988605E-002,4.94032241E-002,-3.20423767E-002,-7.28437863E-003,6.84087574E-002,-2.47318819E-002,6.58546314E-002,1.44245895E-002,1.73984855E-001,-1.08724199E-0  
02,3.534  
65E-002,  
-001,-1.  
034011E-  
02,-2.43  
5181E-00  
.5603099  
E-002,5.  
02248E-0  
,8.26732069E-002,-1.16442284E-002,1.56327486E-002,-4.83360477E-002,1.249367E-002]], COSINE)  
fetch approximate first 3 rows only;
```

Select all the other rows that are similar

```
select id, info from vdata  
where id <> 909  
order by  
vector_distance(v, to_vector(  
['-3.2,1.5,0.9,...'], COSINE)  
fetch approximate first 3 rows only;
```

SP2-0027: Input is too long (> 4999 characters) - line ignored
Help: <https://docs.oracle.com/error-help/db/sp2-0027/>

Similarity Query using Function

Do a similarity search for 'Foiling sailboats are fast'

- Look up the ID of 'Foiling sailboats are fast' row
- Select all the other rows that are similar

```
SELECT ID, INFO FROM VDATA
WHERE ID <> 602
ORDER BY VECTOR_DISTANCE(V, VDATAF(602), COSINE)
FETCH APPROXIMATE FIRST 3 ROWS ONLY;
```

```
SQL> select id, info from vdata where id =602;
```

```
      ID INFO
-----
```

```
      602 Foiling sailboats are fast.
```

```
SQL> SELECT ID, INFO
FROM VDATA
WHERE ID <> 602
ORDER BY VECTOR_DISTANCE(V, VDATAF(602), COSINE)
FETCH APPROXIMATE FIRST 3 ROWS ONLY;
```

```
      2      3      4      5
-----
```

```
      ID INFO
-----
```

```
      605 Sloops have one mast.
```

```
      600 Catamarans have two hulls.
```

```
      504 Man overboard.
```

```
create or replace function vdataf
  (vid in integer) return vector is
  v_return vector;
begin
  select v into v_return
  from vdata
  where id =vid;
  return v_return;
end;
/
```

Another Similarity Search on Movies

```
SQL> @movie_vector_searchf.sql
Enter value for id: 3270

TITLE                YEAR SUMMARY
-----
The Handmaid's Tale  1990 The Handmaid's Tale is a 1990 dystopian film adapted from Canadian author Margaret Atwood's 1985 novel of the same name. Directed by Volker Schlöndorff, the film stars Natasha Richardson (Offred), Faye Dunaway (Serena Joy), Robert Duvall (The Commander), Aidan Quinn (Nick), and Elizabeth McGovern (Moirra). The screenplay was written by playwright Harold Pinter. The original music score was composed by Ryuichi Sakamoto. The film was entered into the 40th Berlin International Film Festival. It is the first filmed adaptation of the novel, succeeded by the Hulu television series which began streaming in 2017.

TITLE                YEAR SUMMARY
-----
A Woman's Tale       1991 A Woman's Tale is a 1991 Australian film directed by Paul Cox and starring Sheila Florance, Gosia Dobrowolska, Norman Kaye, Chris Haywood and Ernie Gray.

Union Maids          1976 Union Maids is a 1976 American documentary film directed by Jim Klein, Julia Reichert and Miles Mogulescu. It was nominated for an Academy Award for Best Documentary Feature. The film was based on the three women from Chicago in the labor history book Rank and File by Staughton Lynd and Alice Lynd.

Maid to Order        1987 Maid to Order is a 1987 American comedy/fantasy film. It is a variation on the Cinderella formula where the fairy godmother is not the means to a better life for the heroine but rather the nemesis. However, rather than doing so out of malice, the fairy godmother hopes to teach the heroine there is more to life than financial security.
```

undefine id

Vector Indexes

Vector Indexes Speed Up Vector Search

- Can be **exact search** indexes which cost because of heavy compute resources
 - Or can be **approximate** indexes which may not be as accurate but less costly
- Vectors are grouped together based on similarity
- 2 types of vector indexes supported
 - **IVF (Inverted File) Flat** = partitioned-based index
 - > Classified as Neighbor Partition Vector Index
 - **HNSW (Hierarchical Navigable Small Worlds)** = graph-based index
 - > In-Memory Neighbor Graph Vector Index

HNSW Vector Index

- Strengths - **Higher Search Quality** - graph-based index that finds nearest neighbors with high precision
 - **Scalability** - can handle large datasets effectively
 - **Pure in-memory index** - meaning it is stored and operated entirely in RAM – so faster search speeds
- Weaknesses - **Slower Index Creation** - Creating index takes longer than creating an IVF index
 - **Higher Memory Requirements** - requires more memory to store the graph structure.

HNSW Indexes

Create HNSW Vector Index on MOVIE(v)

```
select title, year, summary from moviev
where movie_id = &&id;

select title, year, summary
from moviev
where movie_id <> &id
order by vector_distance(v,movie_vdataf(&id), COSINE)
fetch approximate first 3 rows only;

undefine id
```

```
CREATE VECTOR INDEX moviev_vector_idx ON moviev (v) ORGANIZATION INMEMORY NEIGHBOR GRAPH
DISTANCE COSINE
WITH TARGET ACCURACY 95;
```

```
CREATE VECTOR INDEX moviev_vector_idx ON moviev (v) ORGANIZATION INMEMORY NEIGHBOR GRAPH
DISTANCE COSINE
WITH TARGET ACCURACY 95 PARAMETERS (type HNSW, neighbors 40, efconstruction 500);
```

```
select index_name, index_subtype from user_indexes where index_type = 'VECTOR';

MOVIEV_VECTOR_IDX                INMEMORY_NEIGHBOR_GRAPH_HNSW
```

Execution Plan

NO INDEX

Plan hash value: 4221754071

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)
0	SELECT STATEMENT		3	6351		1528 (1)
* 1	COUNT STOPKEY					
2	VIEW		3799	7853K		1528 (1)
* 3	SORT ORDER BY STOPKEY		3799	4288K	5080K	1528 (1)
* 4	TABLE ACCESS FULL	MOVIEV	3799	4288K		605 (0)

Predicate Information (identified by operation id):

- 1 - filter(ROWNUM<=3)
- 3 - filter(ROWNUM<=3)
- 4 - filter("MOVIE_ID"<>3270)

Statistics

```

3809 recursive calls
0 db block gets
8318222 consistent gets
2183 physical reads
0 redo size
1734 bytes sent via SQL*Net to client
1468 bytes received via SQL*Net from client
2 SQL*Net roundtrips to/from client
1 sorts (memory)
0 sorts (disk)
3 rows processed
    
```

Execution Plan

HNSW INDEX

Plan hash value: 1749717704

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		3	6351	3 (34)
* 1	COUNT STOPKEY				
2	VIEW		3	6351	3 (34)
* 3	SORT ORDER BY STOPKEY		3	8241	3 (34)
* 4	TABLE ACCESS BY INDEX ROWID	MOVIEV	3	8241	2 (0)
5	VECTOR INDEX HNSW SCAN IN-FILTER	MOVIEV_VECTOR_IDX	3	8241	2 (0)
6	VIEW	VW_HIJ_983C7328	1		1 (0)
* 7	TABLE ACCESS BY USER ROWID	MOVIEV	1	2747	1 (0)

Predicate Information (identified by operation id):

- 1 - filter(ROWNUM<=3)
- 3 - filter(ROWNUM<=3)
- 4 - filter("MOVIEV"."MOVIE_ID"<>3270)
- 7 - filter("MOVIEV"."MOVIE_ID"<>3270)

Statistics

```

25 recursive calls
0 db block gets
2811 consistent gets
0 physical reads
0 redo size
1734 bytes sent via SQL*Net to client
1468 bytes received via SQL*Net from client
2 SQL*Net roundtrips to/from client
5 sorts (memory)
0 sorts (disk)
3 rows processed
    
```

IVF Indexes

IVF (Inverted File) Flat index = partitioned-based index

- Classified as Neighbor Partition Vector Index
- **Strengths**
 - Faster Index Creation - IVF is faster to create than HNSW
 - Lower Memory Requirements - IVF requires less memory to store the index structure
- **Weaknesses**
 - Lower Search Quality - IVF may have slightly lower search quality than HNSW if complex
 - Less Scalable - IVF might not scale as well as HNSW for extremely large datasets
- Use IVF for faster index creation time
 - Or have memory constraints or can accept lower search quality

```
CREATE VECTOR INDEX moviev_vector_IVF_idx on moviev (v) ORGANIZATION NEIGHBOR PARTITIONS  
DISTANCE COSINE  
WITH TARGET ACCURACY 95 PARAMETERS (type IVF, neighbor partitions 10);
```

```
select index_name, index_subtype from user_indexes where index_type = 'VECTOR';  
  
MOVIEV_VECTOR_IVF_IDX          NEIGHBOR_PARTITIONS_IVF
```

Execution Plan

IVF INDEX

Plan hash value: 1452756757

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)
0	SELECT STATEMENT		3	6351		677 (1)
* 1	COUNT STOPKEY					
2	VIEW		315	651K		677 (1)
* 3	SORT ORDER BY STOPKEY		315	1337K	2528K	677 (1)
4	NESTED LOOPS		315	1337K		389 (1)
* 5	HASH JOIN		315	492K		74 (2)
6	PART JOIN FILTER CREATE	:BF0000	14	56		14 (8)
7	VIEW	VW_IVCR_2D77159E	14	56		14 (8)
* 8	COUNT STOPKEY					
9	VIEW	VW_IVCN_9A1D2119	169	2197		14 (8)
* 10	SORT ORDER BY STOPKEY		169	1690		14 (8)
11	TABLE ACCESS FULL	VECTOR\$MOVIEV_VECTOR_IVF_IDX\$75717_78442_0\$IVF_FLAT_CENTROIDS	169	1690		13 (0)
12	PARTITION LIST JOIN-FILTER		3800	5930K		4 (0)
13	TABLE ACCESS FULL	VECTOR\$MOVIEV_VECTOR_IVF_IDX\$75717_78442_0\$IVF_FLAT_CENTROID_PARTITIONS	3800	5930K		4 (0)
* 14	TABLE ACCESS BY USER ROWID	MOVIEV	1	2747		1 (0)

Predicate Information (identified by operation id):

- 1 - filter(ROWNUM<=3)
- 3 - filter(ROWNUM<=3)
- 5 - access("VW_IVCR_2D77159E"."CENTROID_ID"="VTIX_CNPART"."CENTROID_ID")
- 8 - filter(ROWNUM<=14)
- 10 - filter(ROWNUM<=14)
- 14 - filter("MOVIEV"."MOVIE_ID"<>3270)

Statistics

707 recursive calls
 22 db block gets
 1447687 consistent gets
 175 physical reads
 3888 redo size
 1743 bytes sent via SQL*Net to client
 1468 bytes received via SQL*Net from client
 2 SQL*Net roundtrips to/from client
 2 sorts (memory)
 0 sorts (disk)
 3 rows processed

Cost 3 vs 677

Id	Operation	Name	Rows
0	SELECT STATEMENT		3
* 1	COUNT STOPKEY		
2	VIEW		3
* 3	SORT ORDER BY STOPKEY		3
* 4	TABLE ACCESS BY INDEX ROWID	MOVIEV	3
5	VECTOR INDEX HNSW SCAN IN-FILTER	MOVIEV_VECTOR_IDX	3
6	VIEW	VW_HIJ_983C7328	1
* 7	TABLE ACCESS BY USER ROWID	MOVIEV	1

Statistics

25 recursive calls
 0 db block gets
 2811 consistent gets
 0 physical reads
 0 redo size

HNSW IDX

Select AI with RAG

New feature in Autonomous Database

- Allows users to ask “normal” questions about their data
- Retrieves content using AI Vector Search
- With RAG, it leverages information from the database
 - Reduces hallucinations thus generating better responses
- [Select AI with RAG Documentation](#)

LiveLabs can show you how to set it up

- [Chat with your data in Autonomous Database using generative AI](#)
- MUST subscribe to Specific Tenancy region to use it
 - > US Midwest (Chicago)
 - > Germany Central (Frankfurt)
 - > UK South (London)

High Level Steps to Install Select AI

Use different LLMS for natural language

- Oracle OCI GenAI,
- OpenAI, Azure OpenAI & Google Gemini
 - Need paid account or subscription

Use SQL Worksheet or SQLPlus

- Create Credential
- AI Profile To Connect to AI providers

```
begin
  -- Drop the profile if already exists
  dbms_cloud_ai.drop_profile(
    profile_name => 'genai',
    force => true
  );

  --AI profile uses the default LLAMA model on OCI
  dbms_cloud_ai.create_profile(
    profile_name => 'genai',
    attributes =>
      '{"provider": "oci",
      "credential_name": "OCI$RESOURCE_PRINCIPAL",
      "comments": "true",
      "object_list": [
        {"owner": "MOVIESTREAM", "name": "GENRE"},
        {"owner": "MOVIESTREAM", "name": "CUSTOMER"},
        {"owner": "MOVIESTREAM", "name": "PIZZA_SHOP"},
        {"owner": "MOVIESTREAM", "name": "STREAMS"},
        {"owner": "MOVIESTREAM", "name": "MOVIES"},
        {"owner": "MOVIESTREAM", "name": "ACTORS"}
      ] }' );
end;
/
```

SQLPlus

```
SELECT DBMS_CLOUD_AI.GENERATE(  
  prompt    => '&Question',  
  profile_name => 'GENAI',  
  action    => 'chat');
```

```
SQL> 1  
1  SELECT DBMS_CLOUD_AI.GENERATE(  
2     prompt      => '&Question',  
3     profile_name => 'GENAI',  
4*    action      => 'chat')  
SQL> /  
  
old  2:      prompt      => '&Question',  
new  2:      prompt      => 'Tell me a story about a LION',
```

Once upon a time, in the scorching savannah of Africa, there lived a majestic lion named Kibo. Kibo was the king of the land, with a shaggy mane that rippled in the wind and eyes that shone like golden coins in the sunlight. He was a just ruler, respected by all the animals in the savannah, from the tiny ants to the mighty elephants.

Kibo lived with his pride, a group of lionesses who were his loyal companions and protectors. They were a fierce and formidable team, working together to hunt and defend their territory. Kibo's pride was known for its bravery and cunning, and they were feared by all who crossed their path.

One day, a severe drought struck the land, and the savannah began to wither and dry. The grasses turned brown, the rivers shriveled up, and the animals began to struggle to find food and water. Kibo's pride was no exception, and they grew weaker and hungrier with each passing day.

Kibo knew that he had to do something to save his pride, so he set out on a journey to find a new source of water and food. He traveled far and wide, braving the scorching sun and treacherous terrain, until he came to a hidden oasis deep in the heart of the savannah.

The oasis was a lush and verdant paradise, filled with juicy grasses and sparkling water. Kibo knew that he had found the solution to his pride's problems, and he quickly returned to tell them the good news.

Together, Kibo and his pride made their way to the oasis, where they feasted on the lush grasses and drank from the cool waters. They grew strong and healthy once again, and their bellies were full and content.

But Kibo's journey didn't end there. He knew that the oasis was a precious resource, and he wanted to make sure that it was protected for all the animals in the savannah. So, he made a pact with the other animals, promising to defend the oasis and keep it safe from harm.

Select AI – How it works

Set AI Profile to Specific AI Provider

```
BEGIN
  DBMS_CLOUD_AI.SET_PROFILE(profile_name => 'genai');
END;
/
```

Select AI actions

- Chat: Use for general AI Chat
- Runsql: Ask a question & get structured result (default)
- Narrate: Ask a question & get a conversational result
- Showsql: Produces SQL statement for the result

Select AI 'Chat' Examples

```
SQL> select AI chat
  2  Who is president of the United States;
```

RESPONSE

As of my knowledge cutoff in 2023, the President of the United States is Joe Biden. However, please note that my information may not be up to date, and I recommend checking a reliable news source for the most current information.

```
SQL> select AI chat
  2  What date is today
  3  ;
```

RESPONSE

Today's date is February 18, 2025

```
SQL> select AI chat
  2  What is the date of your most recent knowledge;
```

RESPONSE

My most recent knowledge cutoff is December 2023.

Select AI 'Chat' Examples

```
SQL> select AI chat
      2  What is Angelina Jolie best known for;
```

RESPONSE

Angelina Jolie is a renowned American actress, filmmaker, and humanitarian. She is best known for her versatility and range in various film roles, as well as her philanthropic work. Here are some of the things she is most known for:

1. Acting career: Jolie has appeared in numerous successful films, including "Lara Croft: Tomb Raider," "Mr. & Mrs. Smith," "Maleficent," and "Girl, Interrupted," for which she won an Academy Award for Best Supporting Actress.
2. Humanitarian work: Jolie is a dedicated humanitarian and has worked with various organizations, including the United Nations High Commissioner for Refugees (UNHCR), to help refugees and displaced people around the world.
3. Adoption and family: Jolie has adopted three children from different countries (Cambodia, Ethiopia, and Vietnam) and has three biological children with her ex-husband Brad Pitt.
4. Directing and producing: Jolie has also ventured into directing and producing, with films like "In the Land of Blood and Honey" and "Unbroken," which she directed and produced.
5. Beauty and style: Jolie is known for her stunning beauty and has been named one of the most beautiful people in the world by various publications, including People magazine.

Overall, Angelina Jolie is a talented and influential figure in the entertainment industry, known for her impressive acting career, humanitarian work, and dedication to her family.

Select AI 'runsql' (default)

```
SQL> select AI runsql
```

```
SQL> select AI
```

```
2 What are the total streams per movie for Angelina Jolies movies;
```

Movie Title	Total Streams
Mr. & Mrs. Smith	1967
Maleficent	8684
Love Is All There Is	4
Foxfire	68
Cyborg 2	105

Select AI 'narrate'

```
SQL> select AI narrate  
2 What are the total streams per movie for Angelina Jolies movies;
```

RESPONSE

Angelina Jolie's movies have the following total streams:

- Mr. & Mrs. Smith: 1967 streams
- Maleficent: 8684 streams
- Love Is All There Is: 4 streams
- Foxfire: 68 streams
- Cyborg 2: 105 streams

Select AI 'showsql'

```
SQL> select AI showsql
  2  What are the total streams per movie for Angelina Jolies movies;
```

RESPONSE

SELECT

```
  T2.TITLE AS "Movie Title",
  SUM(T1.VIEWS) AS "Total Streams"
```

FROM

```
  "MOVIESTREAM"."STREAMS" T1
  INNER JOIN "MOVIESTREAM"."MOVIES" T2 ON T1.MOVIE_ID = T2.MOVIE_ID
  INNER JOIN "MOVIESTREAM"."ACTORS" T3 ON T2.MOVIE_ID = T3.MOVIE_ID
```

WHERE

```
  T3.ACTOR = 'Angelina Jolie'
```

GROUP BY

```
  T2.TITLE
```

Summary

AI consists of ML algorithms use LLMs & RAG

- To enhance accuracy of data by coming external data with business knowledge base

Oracle23ai Contributes to AI Landscape

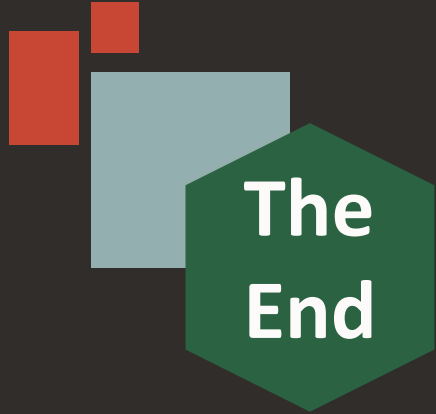
- Oracle's Machine Learning engine primarily leverages ONNX-formatted transformer models
- Integration of AI with existing business data (Brings the algorithms to the data)
- Allows natural language processing & text generation

Vectors are a mathematical representation of data features

- Oracle vectors are represented by the data type of "VECTOR"
- Use Vector Embeddings to convert words & sentences into arrays of real numbers
- Vector Indexes can speed up Vector Searches

Select AI works only in Autonomous Database

- Use natural language / create SQL statements
- Needs to be in specific Region - For US: Midwest (Chicago)



Questions?

Thank You!