Boosting MySQL with Vector Search

Fosdem 2025

Introducing the MyVector Plugin

Alkin Tezuysal / Shankar Iyer

Let’s get connected!

Alkin Tezuysal - Director of Services @AltinityDB

- Linkedin : https://www.linkedin.com/in/askdba/

Open Source Database Evangelist

- Previously ChistaDATA, PlanetScale, Percona and Pythian as Senior Technical Manager, SRE, DBA
- Earlier in life Enterprise DBA , Informix, Oracle, DB2 , SQL Server
- Recent Recognitions:
  - Most Influential in Database Community 2022 - The Redgate 100
  - MySQL Cookbook, 4th Edition 2022 - O'Reilly Media, Inc.
  - MySQL Rockstar 2023 - Oracle (MySQL Community)
  - Database Design and Modeling with PostgreSQL and MySQL 2024 - <Packt>

Born to Sail, Forced to Work!

@ask_dba

Sailing Trivia

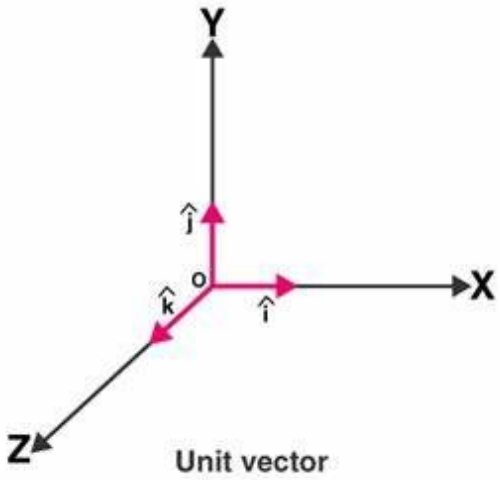
1 nautical mile (nm) equals to ...

Let’s get connected!

Shankar Iyer

- Linkedin : https://www.linkedin.com/in/shankar-iyer-a4a1563/
- Earlier in life : Engineer in Oracle RAC Database Development Team (link)
- Recently worked in AWS Aurora MySQL, Google Cloud MySQL, Clickhouse
- Interests : MySQL for GenAI, Research & Experimentations in MySQL performance/replication/availability, SQL optimizations for BI

-



Vectors and Dimensions

- Mathematical objects representing magnitude and direction (velocity, force)
- Vectors can be represented as arrays or ordered lists of numbers
  - 2D (x, y), 3D (x, y, z), n-dimensional (AI)
- Basically data points
- When dimensions increase data points increase and they turn into embeddings

Understanding Vector Embeddings

- Vector Embeddings: Mathematical representations of data in a high-dimensional space.
- Capture semantic meaning and relationships between words, phrases, or other data.
- Enable more accurate and relevant search results compared to traditional keyword search.
- Power AI applications like semantic search, recommendation systems, and chatbots.
- Data analysis and application functionality

Vector Embeddings

_id string · lengths 25 39 46.7%	title string · lengths 15 29 46.7%	text string · lengths 1 1.26k 99.9%	openai sequence
<dbpedia:Animalia_(book)>	Animalia (book)	Animalia is an illustrated children's book by Graeme Base. It was originally published in 1986, followed by a tenth anniversary edition in 1996, and a 25th anniversary edition in 2012. Over three million copies have been sold. A special numbered and signed anniversary edition was also published in 1996, with an embossed gold	[0.017398979514837265, -0.01408793 -0.010348621755838394, -0.022282455 -0.010668220929801464, 0.0253889597 -0.030783794820308685, -0.027204282 0.005008119158446789, -0.0129885114 0.007529756985604763, 0.00643353164 -0.007580892648547888, 0.0024209639 0.006503843702375889, 0.02431510575 0.02932642214000225, 0.027229851111 -0.013397597707808018, -0.002714995 0.0036338428035378456, 0.0462523959 0.0095304474234581, -0.020863434299 0.012438800185918808, -0.0020134749 0.018587889149785042, -0.0307070910 0.006999222096055746, -0.0254912320 0.011416083201766014, -0.0182555057

‹ Previous 1 2 3 ... 10,000 Next ›

1M OpenAI Embeddings -- 1536 dimensions

Created: June 2023. Text used for Embedding: title (string) + text (string) Embedding Model: text-embedding-ada-002

Transforming data into vectors in a way that captures the semantic meaning or relationships between the data points. *(The image on the left shows a OpenAI vector embedding for a Wikipedia article using text-embedding-ada-002 Vector dimension is **1536**)*

OpenAI has another model text-embedding-3-large model provides 3,072-dimensional vector embeddings. P.S: Deep Seek can generate embedding dimensions upto 8K!

Benefits of Vector Search

- Efficient and relevant search results.
- Improved relevance for text and image search.
- Semantic understanding.
- Supports AI-based applications.
- Leverage and unlock value from live and historical business data in the MySQL database : e.g *conversation logs, clinical records, support tickets, persona descriptions, medical images, product catalog, legal filings* etc.

Vector Support Helps

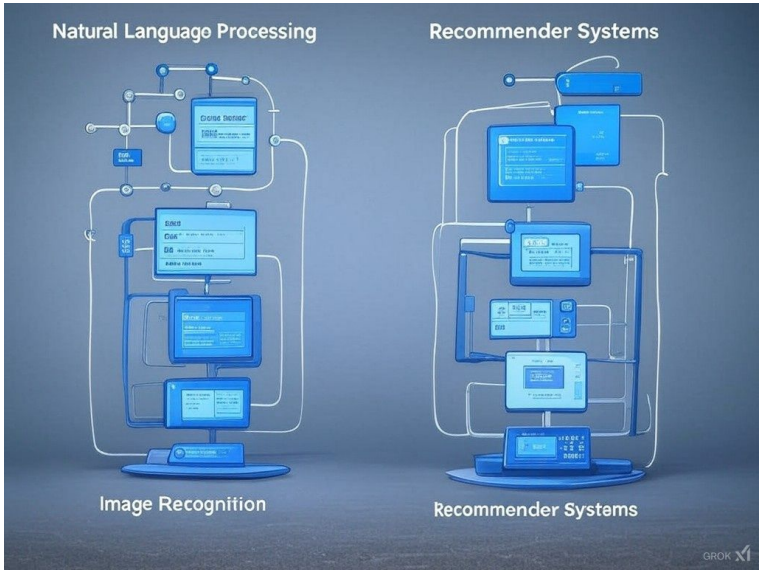
- Traditional search struggles with high-dimensional data and semantic understanding.
- Inefficient for AI and ML applications that rely on vector embeddings.
- Vector support enhances performance and relevance for similarity searches.
- Enables new AI-driven applications directly within MySQL.
- Unlocks advanced data analysis techniques using vector embeddings.

Why Vector Search in MySQL?

- MySQL World's number one open-source relational database
- Vector search allows for more efficient and relevant search results.
- Improved relevance for text and image search with semantic understanding.
- MyVector plugin extends MySQL with vector search capabilities.
- Supports a wide range of AI-based applications.

P.S: MySQL does not come with extensibility interfaces to add new data types, and no support to add new index types and to add new access methods

Vector Search Use Cases



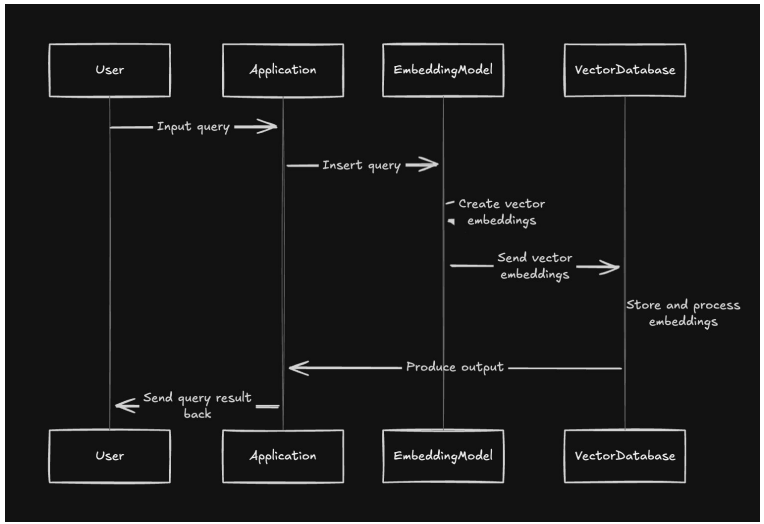
Use Cases:

Natural Language Processing (NLP): Words or sentences are converted into vectors where similar words or phrases are closer in vector space.

Recommender Systems: Items or users can be embedded to find similarities or predict preferences.

Image Recognition: Images can be encoded into vectors for tasks like similarity search or classification.

Vector Databases



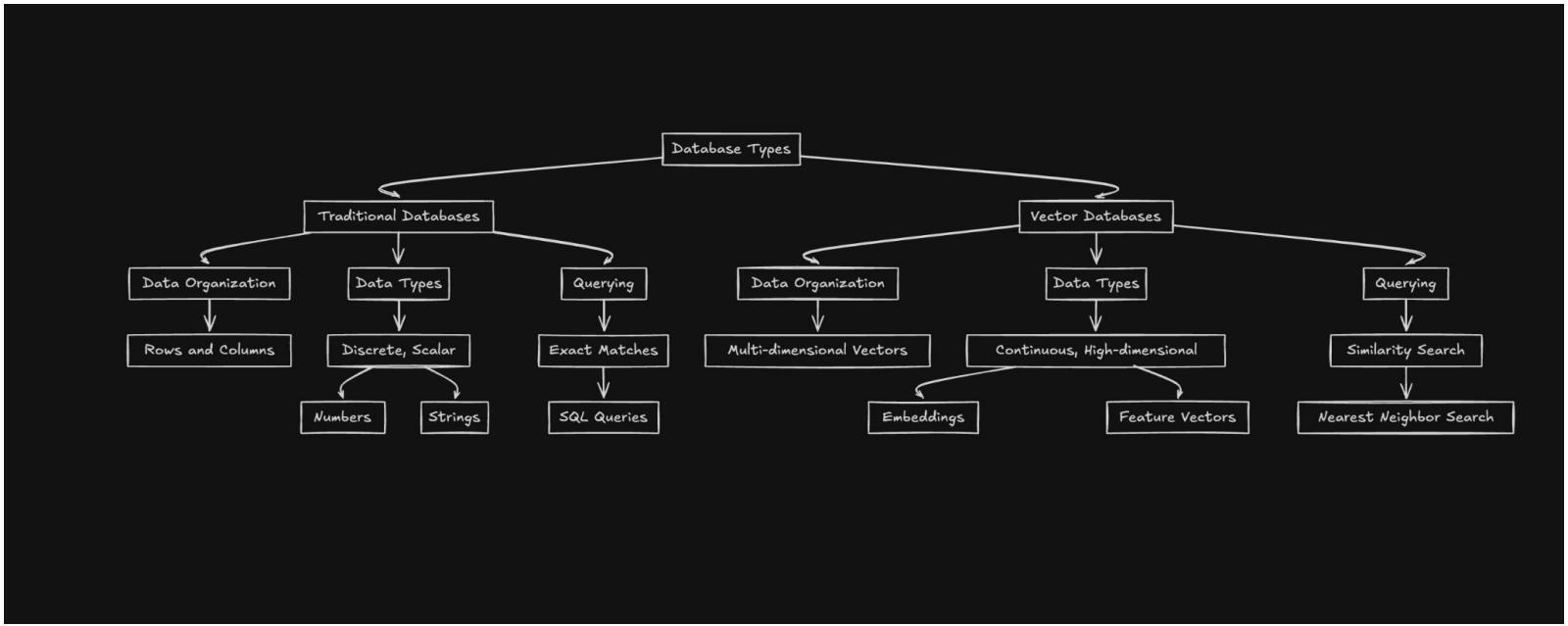
Vector Database Providers

- Pinecone
- Qdrant
- Milvus
- Weavite
- Chroma

Vector Store Providers

- ClickHouse
- Elasticsearch
- Singlestore
- Typesense
- Others
  - MySQL
  - MariaDB
  - PostgreSQL.

Vector Databases vs Traditional Databases



MySQL Vector Plugin Architecture

The MyVector Plugin, discussed earlier, leverages this architecture. Here's how it fits:

- UDFs in the plugin: Adds vector-related SQL functions.
- Data storage : Use varbinary in MySQL 8.x and VECTOR in MySQL 9.x.
- Indexing: Integrates vector similarity search via HNSW indexing.
- Similarity Search - Intuitive syntax for search using SQL
- Administration - Simple MySQL stored procedures

MySQL Vector Plugin Build

```
/// get the MyVector sources

$ cd mysql-server/src/plugin

$ git clone https://github.com/p3io/myvector-dev/
./myvector

/// Generate makefile for the new plugin

$ cd mysql-server/bld

$ cmake .. <other options used for this build >

/// Build the plugin

$ cd mysql-server/bld/plugin/myvector

$ make
```

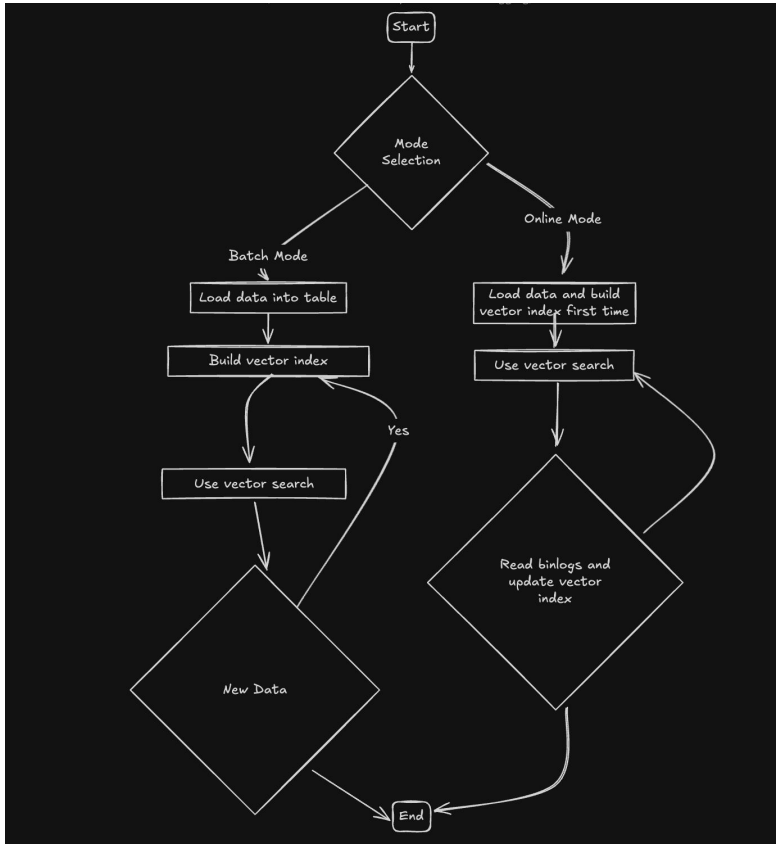
MySQL Vector Plugin Install

```
/// Copy the MyVector plugin shared library to the
MySQL installation plugins

$ cp
mysql-server/bld/plugin_output_directory/myvector.so
/usr/local/mysql/lib/plugin/

/// Register the MyVector plugin and create MyVector
stored procedures.

$ cd mysql-server/plugin/myvector

/// Connect to 'mysql' database as 'root'

$ mysql -u root -p   mysql

mysql> source myvectorplugin.sql
```

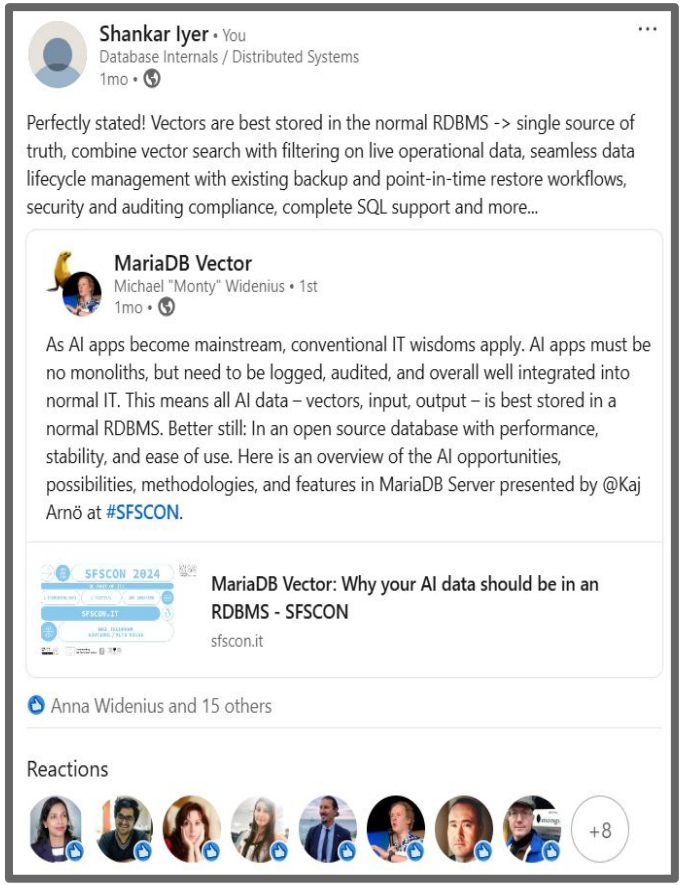
MySQL Vector Plugin Architecture

- MySQL V8.X - InnoDB + Plugin + Varbinary Data Type
- MySQL V9.X - InnoDB + Plugin + Vector Data Type

```
mysql> select udf_name from
performance_schema.user_defined_functions where udf_name like
"myvector%" ;
+-----------------------------+
| udf_name                    |
+-----------------------------+
|                             |
| myvector_is_valid           |
| myvector_distance           |
| myvector_display            |
| myvector_ann_set            |
| myvector_row_distance       |
| myvector_construct          |
+-----------------------------+
```



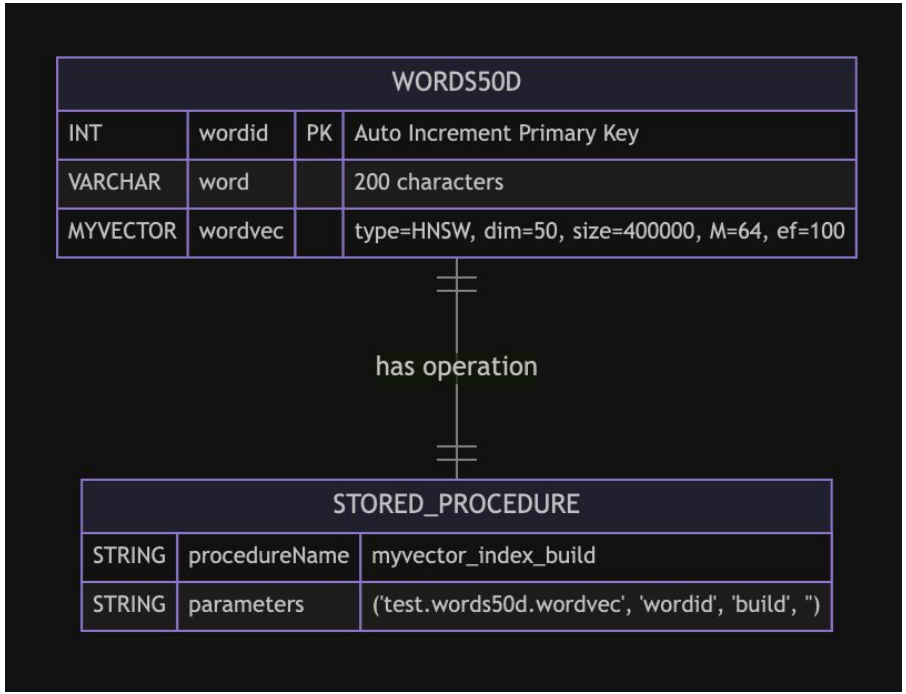
MyVector Features



- High speed, parallel build of HNSW index
- Specify recall v/s latency parameters : M, ef, ef_search
- HNSW index is incrementally persisted after initial build
- Online update of index via binlog read & parse of DMLs
- HNSW Index is *crash-safe* and is recovered on MySQL instance crash
- Write amplification to redo log/undo log avoided

MySQL Vector Plugin Examples MySQL 8.X

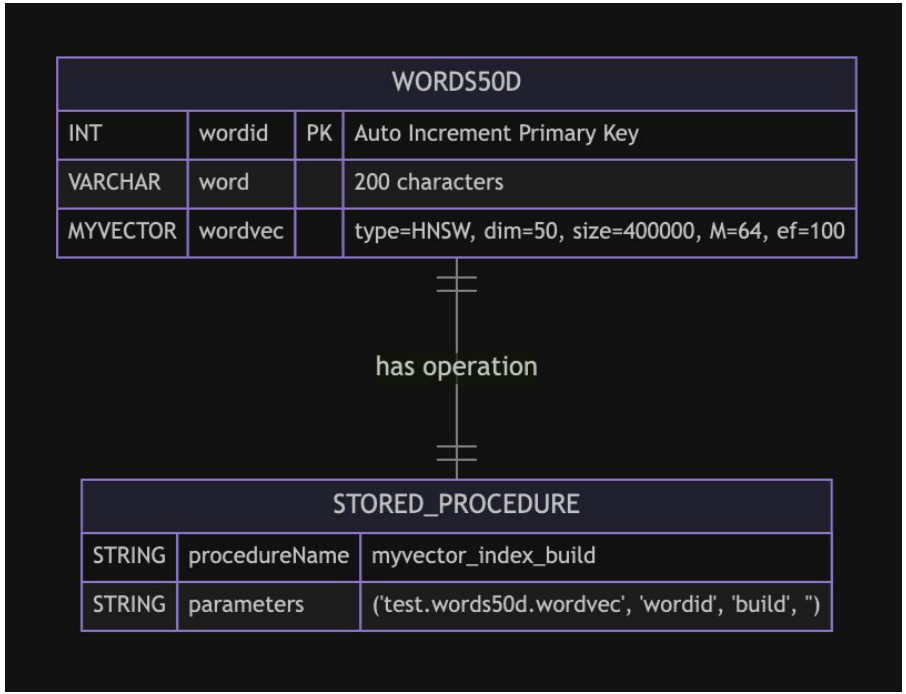
```
mysql > create table words50d(
wordid  int primary key,
word    varchar(200),
wordvec MYVECTOR(type=HNSW,
        dim=50,size=400000,
        M=64,ef=100)

);

- Load the Data (insert …)

mysql> call
myvector_index_build
('test.words50d.wordvec','wordid');
```



Ref: https://nlp.stanford.edu/projects/glove/

MySQL Vector Plugin Examples MySQL 9.1

```
mysql > create table words50d(
wordid  int primary key,
word    varchar(200),
wordvec MYVECTOR(type=HNSW,
        dim=50,size=400000,
        M=64,ef=100)

);

- Load the Data (insert …)

mysql> call
myvector_index_build
('test.words50d.wordvec','wordid');
```



MySQL Vector Plugin Examples

`myvector_construct(vector_string VARCHAR):`

Purpose: Converts a human-readable vector string into a serialized binary format suitable for storage in a VARBINARY / VECTOR column.

Usage Example:

```
INSERT INTO vectors_table (vector_column)
VALUES (myvector_construct('[0.1, 0.2, 0.3, ...]'));
```

MySQL Vector Plugin Examples

`myvector_display(vector_col_expr VARBINARY/VECTOR):`

Purpose: Transforms a binary-stored vector back into a human-readable string representation.

`Usage Example:`

```
SELECT myvector_display(vector_column) AS readable_vector
FROM vectors_table;
```

MySQL Vector Plugin Examples

`myvector_distance(vec1 VARBINARY/VECTOR, vec2 VARBINARY/VECTOR, disttype VARCHAR):`

Purpose: Calculates the distance between two vectors using the specified distance metric ('L2', 'EUCLIDEAN', or 'IP' for inner product).

`Usage Example:`

```
mysql> select myvector_distance((select wordvec from words50d where word = 'school'),
(select wordvec from words50d where word='institute'))\G

*************************** 1. row ***************************

myvector_distance((select wordvec from words50d where word = 'school'), (select wordvec
from words50d where word='institute')): 25.51254653930664

1 row in set (0.36 sec)
```

MySQL Vector Plugin Examples

Similarity Search (ANN)

```
SELECT <column-list> FROM <table> WHERE
       MYVECTOR_IS_ANN('vector_column','key column',
                       <search vector>, options)


// search vector(s) should first be inserted to <query_table>
SELECT <column-list> FROM
       MYVECTOR_SEARCH[<base_table>, <key column>,
                       <vector_column>, <query_table>]
```

MySQL Vector Plugin Examples

`myvector_ann_set(veccol VARCHAR, options VARCHAR, searchvec VECTOR/VARBINARY):`

Purpose: Returns a comma-separated list of IDs corresponding to the nearest neighbors of a given search vector.

Usage Example:

```
SELECT myvector_ann_set('vector_column', 'options', myvector_construct('[0.1, 0.2, 0.3, ...]')) AS neighbors FROM vectors_table;
```

MySQL Vector Plugin Examples

Vector Index Administration - Stored Procedures

```
// build the index
CALL myvector_index_build('vector column', 'key column')

// check current status of the index - number of rows, parameters etc
CALL myvector_index_status('vector column')

// open/load the index into memory after a restart (if not marked online)
CALL myvector_index_load('vector column')

// drop the the index
CALL myvector_index_drop('vector column')
```

Demo

https://github.com/p3io/myvector-dev/tree/main/demo

References

https://medium.com/@shiyer22/introducing-myvector-plugin-vector-storage-similarity-search-in-mysql-a32a6f84755e
https://medium.com/@shiyer22/introducing-myvector-plugin-vector-storage-similarity-search-in-mysql-part-2-74a7195226e2
https://medium.com/@shiyer22/myvector-plugin-part-3-nearest-neighbour-search-with-other-predicates-and-the-sql-optimizer-7541bfaa3df9
https://blogs.oracle.com/mysql/post/mysql-vector-datatype-create-your-operations-part-1
https://lefred.be/content/mysql-vector-datatype-create-your-operations-part-2/
https://learning.oreilly.com/home/#:~:text=Vector%20Databases%20Deep%20Dive
https://asciinema.org/a/KZeSifbD0QpHYmYUep5SiaCMA
https://asciinema.org/a/YTIg12ClZAh38qUcgsBUAdSA7
https://asciinema.org/a/4uro9wYgFijuwP3EdJvp0tRMj
https://asciinema.org/a/O7rNs2OzLXyUja0bwWcldXsX9
https://asciinema.org/a/RJs3CFbuShQGeXkCSxDDG0n3i

altinity.com/slack