



AI and the Enterprise Application



February 19th, 2025

Bill Reynolds, CTO  **Qarbine**



AWS Modern Databases and AI Challenges

AWS Neptune/DocDB plus Gen AI are powerful, but...

Legacy enterprise tools:

- use SQL vs. Cypher or MongoDB QL
- want tabular results vs. graphs or documents

Let's see how the duo of AWS Bedrock Gen AI services and AWS DBs can be applied across the app lifecycle...

Simplified code examples

Qarbine interactions showing

1. native DB querying and AI service integrations
2. handling of complex modern data results
3. delivering publication quality analytics



Bedrock AI Benefit Areas Across the Lifecycle

1. Using AI Assistants for
 1. Authoring queries
 2. Explaining existing queries
2. Leveraging vector indexes for queries
3. Analyzing answer set content
4. Using ad hoc prompts for flexible vector retrievals
5. Retrieval Augmented Generation (RAG) for analysis



Various Technologies

1. Databases

1. AWS Neptune and Neo4j use Cypher for querying
2. AWS DocumentDB and MongoDB use MongoDB Query Language for queries
3. AWS DynamoDB native is a JSON specification
4. AWS RDS is SQL oriented

2. Bedrock AI Services

1. AWS Bedrock supports a variety of LLMs
 1. Completions- ~ chatGPT
 2. Embeddings- similarity array “vector” used by DBs for querying



AWS Database Considerations

Neptune support 2 querying languages

1. openCypher
2. Gremlin

Neptune Analytics provides the vector search features

<https://docs.aws.amazon.com/neptune-analytics/latest/userguide/vector-index.html>

You can only create a vector search index for a Neptune Analytics graph **at the time the graph is created**. Neptune Analytics lets you create only one vector index for a graph, with a fixed dimension between 1 and 65,535 inclusive.

AWS DocumentDB

1. Sort of a work-a-like to MongoDB
2. Slightly different AI vector search querying syntax



Core Bedrock AI Concepts

1. Completions

1. ChatGPT and co-pilot in nature

2. Provide context and then “ask AI”

Most enterprise use cases are text oriented

3. Bedrock API sits on top of underlying LLMs

Considerations

1. LLMs conversational input rules vary * 😞

Claude- Human: `{humanInput}`

Question: `{promptArg}`\nAssistant:

Llama-

`<s>[INST]\n<<SYS>>{humanInput}<</SYS>>`

Question: `{promptArg}`\n[`/INST`]

* Many AI services are OpenAI API look-alikes which simplifies integrations



Simplified Bedrock node.js Completion I/O

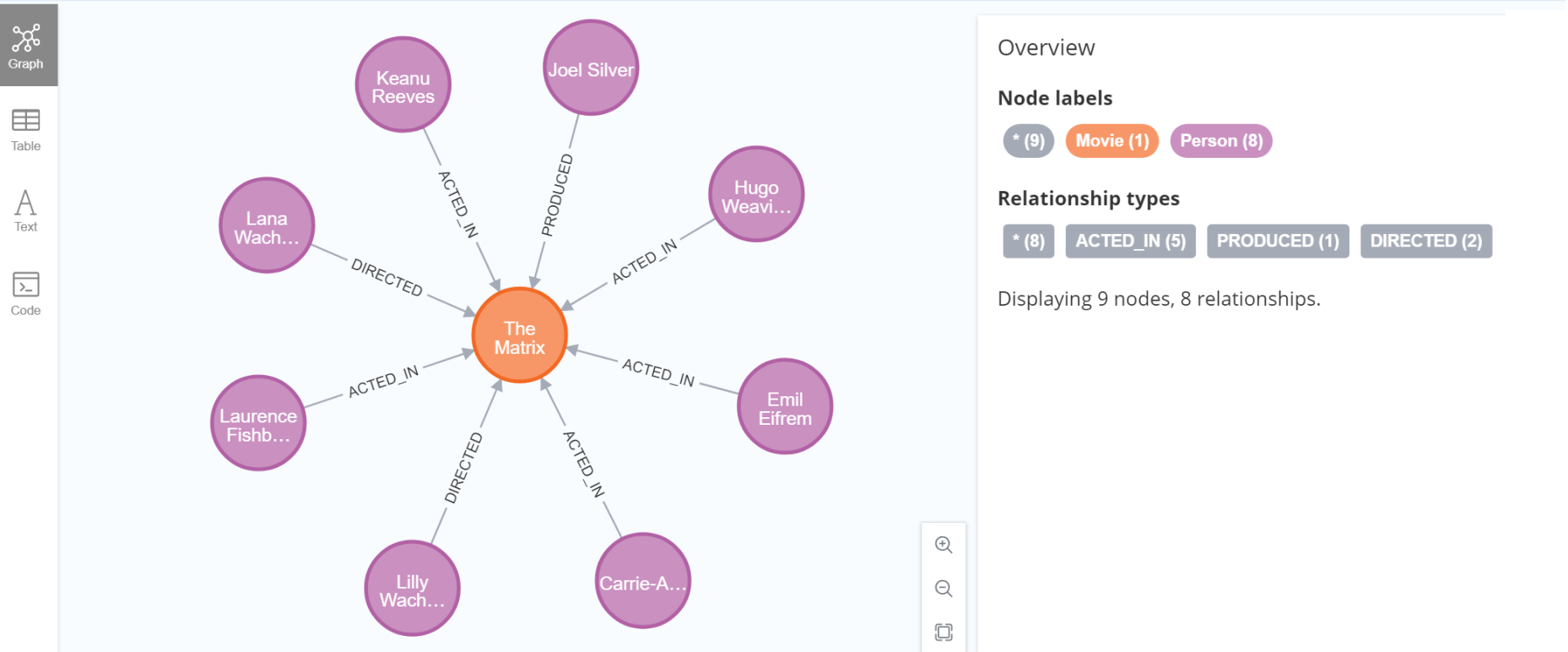
```
// Claude
var payload = { "prompt": ` Human: ${humanInput*}\nQuestion: ${promptArg}\nAssistant: `,
  "max_tokens_to_sample":300,
  "temperature":1,
  "top_k":250,
  "top_p":0.999,
  "stop_sequences":[]
};
var params = {
  modelId: "anthropic.claude-v2",
  contentType: "application/json",
  body: JSON.stringify(payload)
};
var data = await bedrockRuntime.invokeModel(params);
const asciiDecoder = new TextDecoder('utf-8');
const responseBodyJson = asciiDecoder.decode(replyBodyBlob);
var responseBody = JSON.parse(replyBodyJson);
var completionText = responseBody.completion;
```

*** Answer the question below as truthfully as possible considering the provided context. If you don't know the answer, just say "I don't know"**



Sample Movies Query and Graph Data Result

```
neo4j$ MATCH (person: Person)-[relatedTo]- (movie: Movie {title: "The Matrix"}) RETURN * order by person.name
```



Visualizations are very useful to depict relationships.

But many business decisions require the details behind the diagrams.

Why not benefit from having both! Gen AI can help us on the way!



AI Helps Devs Understand & Query Graph Data

Properties

▼ [...] Movie

- 🔑 _identity
- # budget
- T countries
- T imdbId
- T imdbRating
- # imdbVotes
- T languages
- T movieId
- T plot
- T plotEmbedding
- T poster
- T posterEmbedding
- T released
- # revenue
- # runtime
- T title
- T tmdbId
- T url
- # year

- ▼ [...] <~~ ACTED_IN
 - 📄 Actor_Director_Person
- ▼ [...] <~~ DIRECTED
 - 📄 Director_Person
- ▼ [...] <~~ RATED
 - 📄 User
- ▼ [...] ~~~> IN_GENRE
 - 📄 Genre

“<~~ “are relationships pointing into a Movie:

An Actor_Director_Person ACTED_IN a Movie.

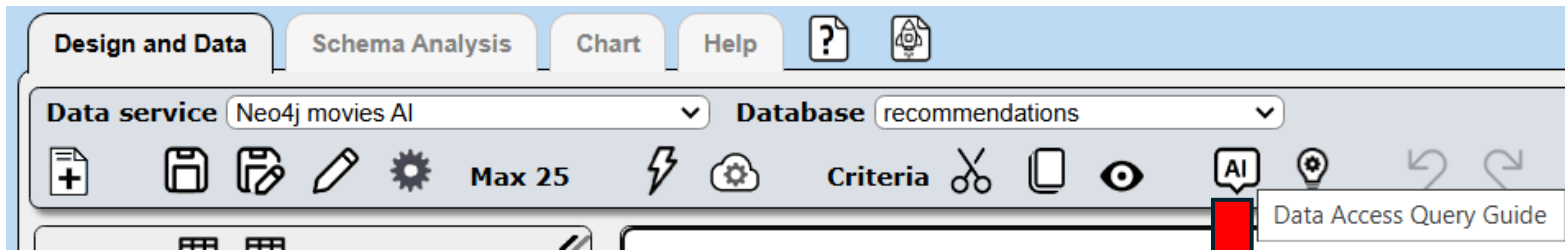
‘~~> ‘are relationships that a Movie points to:

A Movie is “IN_GENRE” Genre.

You can use this context to “Ask Bedrock AI” for guidance on authoring your Cypher queries.



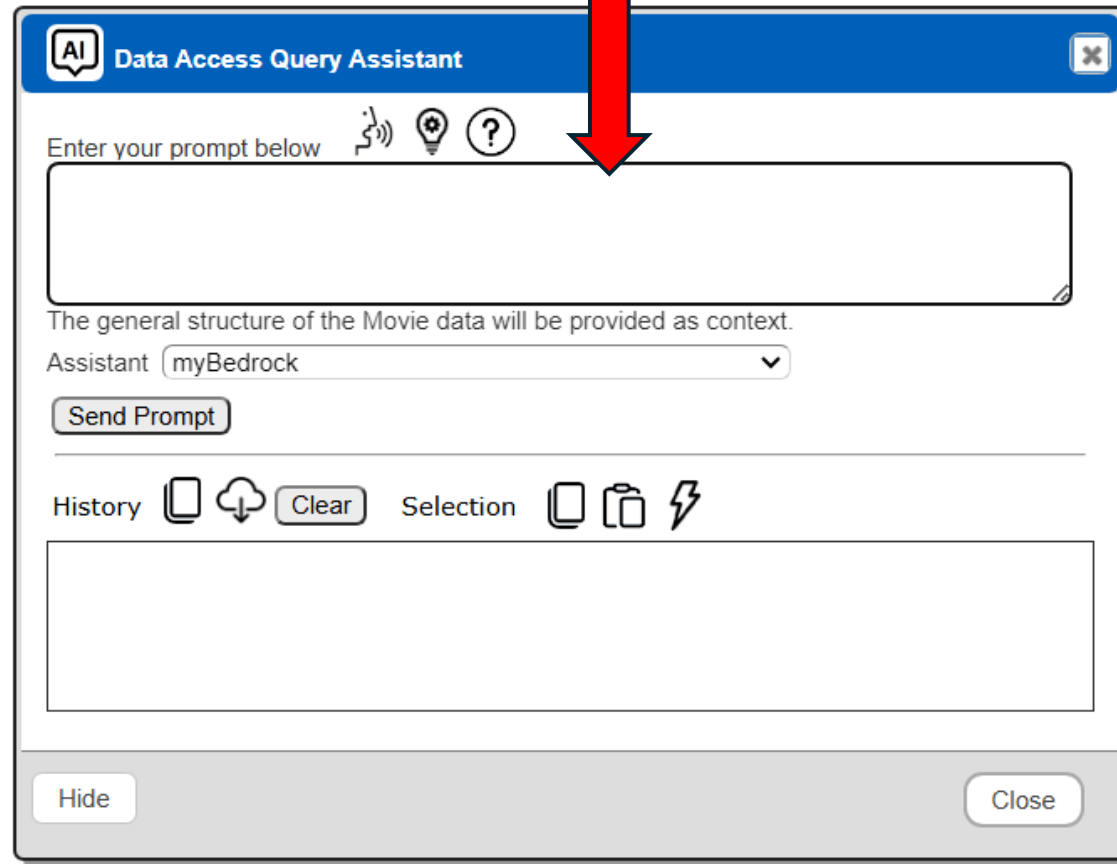
Qarbine has an Integrated Query Co-pilot



The context given to AI to answer the user's prompt varies across AWS Neptune, RDS, DocumentDB, DynamoDB, etc.

Query goal:

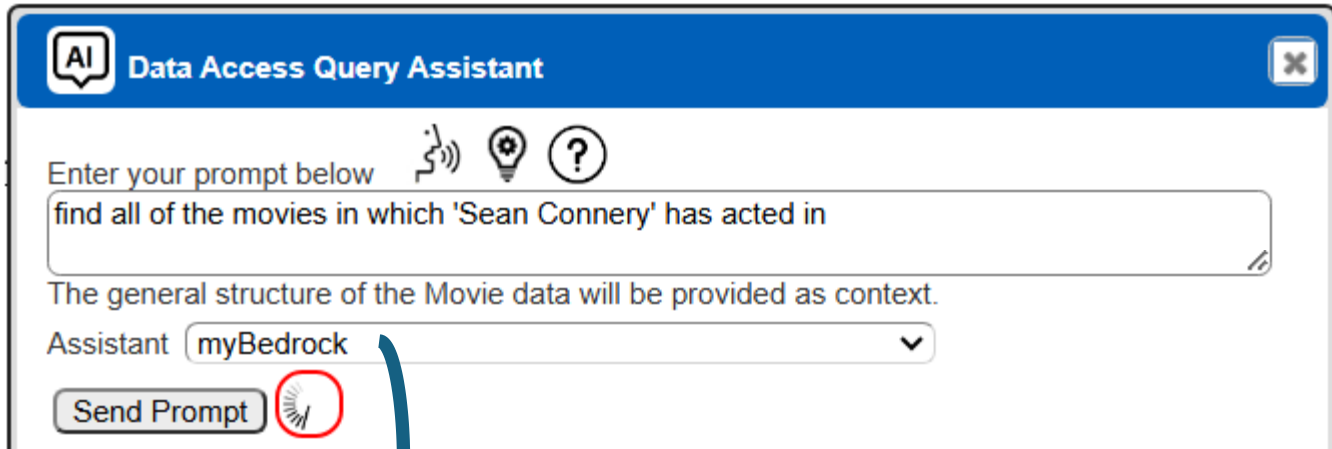
Find all of the movies in which "Sean Connery" has acted in





AI Helps Devs Understand & Use Cypher

Find all of the movies in which "Sean Connery" has acted in



The “myBedrock” value refers to a Qarbine “AI Assistant”.

Multiple may be defined based on AI service, model, etc.

Qarbine supports many popular Gen AI services.



“Ask AI” for an Appropriate Cypher Query

History Selection

1 find all of the movies in which 'Sean Connery' has acted in

Based on the sample document provided, to find all the movies in which 'Sean Connery' has acted in, we would query the ACTED_IN relationship for documents where Actor_Director_Person equals 'Sean Connery'. In Cypher query language this would be:

```
MATCH (m:Movie)-[:ACTED_IN]->(p:Person)
WHERE p.name = 'Sean Connery'
RETURN m
```

This queries for Movie nodes that have an ACTED_IN relationship to Person nodes where the name property equals 'Sean Connery', and returns the Movie nodes.

YOMV
Your output may vary!

Copy the query

Set the query

Set the query and run it

History Selection

1 find all of the movies in which 'sean connery' has acted in

Based on the sample document provided, to find all the movies that 'sean connery' has acted in, we would query the ACTED_IN relationship field for documents where the Actor_Director_Person is 'sean connery'.

In Cypher query language, it would look like:

```
MATCH (m:Movie)-[:ACTED_IN]->(p:Actor_Director_Person)
WHERE p.name = 'sean connery'
RETURN m
```

This matches movies (m) with an ACTED_IN relationship to person nodes (p) where the name is 'sean connery', and returns the movie nodes.



Results of Running the Generated Cypher Query

```
MATCH (a:Person {name: "Sean Connery"})-[:ACTED_IN]->(m:Movie)
RETURN m
```

Filled in Cypher query

_identity	_labels	languages	plotEmbedding	year	imdbId	runtime	imdbRating	movieId	countries	imdbVotes	title
248	Array	Array	Array	1995	0113501	102	6.4	257	Array	19111	Just Cause
2280	Array	Array	Array	1992	0104839	106	6	2822	Array	17135	Medicine Man
482	Array	Array	Array	1993	0107969	129	6.2	517	Array	34019	Rising Sun
3632	Array	Array	Array	1989	0097328	110	5.6	4593	Array	9235	Family Business
2588	Array	Array	Array	1988	0095897	97	5.9	3197	Array	12700	Presidio, The
3918	Array	Array	Array	1990	0100530	123	6.1	5033	Array	11059	Russia House, The
1300	Array	Array	Array	1990	0099810	135	7.6	1610	Array	139120	Hunt for Red October, The

Cypher query results



Cypher Query Result Element

1300 Array Array Array 1990 0099810 135 7.6 1610 Array 139120 Hunt for Red October, The

```
Selection
{
  _identity: 1300,
  labels: [
    "Movie"
  ],
  languages: [
    "English",
    "Russian"
  ],
  plotEmbedding: [ 1536 fields ],
  year: 1990,
  imdbId: "0099810",
  runtime: 135,
  imdbRating: 7.6,
  movieId: "1610",
  countries: [
    "USA"
  ],
  imdbVotes: 139120,
  title: "Hunt for Red October, The",
  url:
    "https://themoviedb.org/movie/1669",
  revenue: 200512643,
  tmdbId: "1669",
  plot: "In November 1984, the Soviet Union's best submarine captain in their newest sub violates orders and heads for the USA. Is he trying to defect or to start a war?",
  posterEmbedding: [ 512 fields ],
  poster:
    "https://image.tmdb.org/t/p/w440_and_h660_face/yVl7zidse4KiWtGMqHFtZCx4X3N.jpg",
  released: "1990-03-02",
  budget: 30000000
}
```

Notice the JSON structure has many embedded and arrays

The plotEmbedding vector is 1536 in size and the postEmbedding vector 512 in size.

Cypher and such nested structures are both non-starters for legacy analytics tools.



Asking AI to Explain Existing Cypher Queries



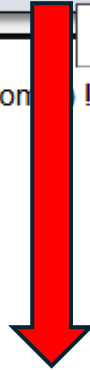
Criteria



Result

```
MATCH (node:Movie)
WITH node, vector.similarity.euclidean( [! embeddings(@prompt) !], node.plotEmbedding) AS score
where score is not null
RETURN node, score
ORDER BY score DESCENDING
LIMIT 3
```

Explain the tool's current query



1 translate this query into plain english.

```
MATCH (node:Movie)
WITH node, vector.similarity.cosine( [! embeddings(@prompt) !], node.plotEmbedding) AS
score
where score is not null
RETURN node, score
ORDER BY score DESCENDING
LIMIT 5
```

Find the top 5 movies that have the closest similarity to the given prompt based on their plot embeddings.

Provide AI context such as the type of database.

Bedrock AI query explanation via completions



Asking AI About a More Complex Cypher Query

```
MATCH (m:Movie {title: 'Inception'})-[:IN_GENRE|ACTED_IN|DIRECTED]-
      (t)-[:IN_GENRE|ACTED_IN|DIRECTED]-(other:Movie)
WITH m, other, count(t) AS intersection, collect(t.name) AS common,
      [(m)-[:IN_GENRE|ACTED_IN|DIRECTED]-(mt) | mt.name] AS set1,
      [(other)-[:IN_GENRE|ACTED_IN|DIRECTED]-(ot) | ot.name] AS set2

WITH m,other,intersection, common, set1, set2,
      set1 + [x IN set2 WHERE NOT x IN set1] AS union

RETURN m.title, other.title, common, set1,set2,
      ((1.0*intersection)/size(union)) AS jaccard
ORDER BY jaccard DESC LIMIT 25
```

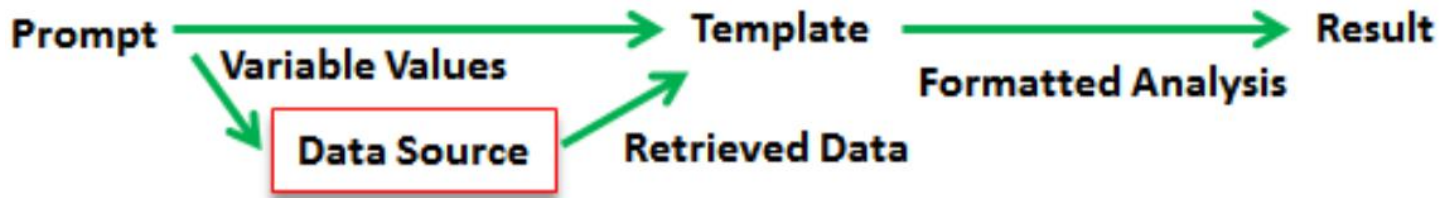
Bedrock AI query explanation via completions

Find movies that are most similar to "Inception" based on the Jaccard similarity of their genres. This query compares the genres of "Inception" with other movies by looking at the actors, directors, and genres they are associated with. It calculates the Jaccard similarity coefficient by looking at the intersection and union of genres between the two movies. The query returns the title of "Inception," the title of the similar movie, the common genres, the genres associated with "Inception," the genres associated with the similar movie, and the Jaccard similarity coefficient. The results are sorted in descending order of similarity and limited to the top 25 matches.

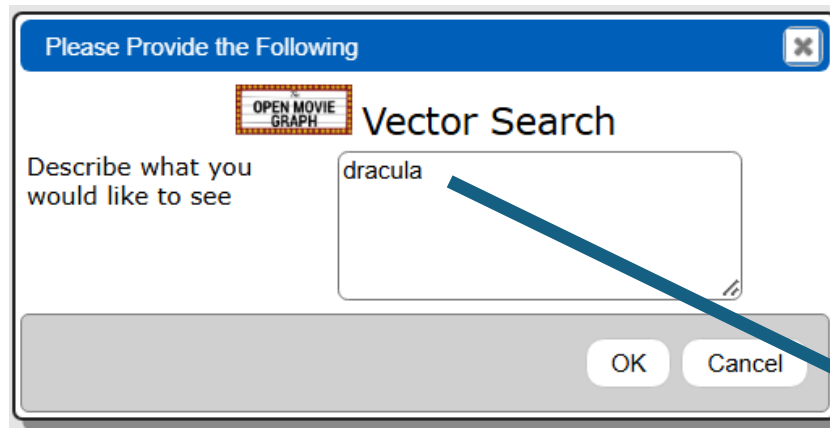


Vector Query Flow for AWS DB and Gen AI Analytics

Vector searching using dynamic Bedrock AI embeddings



A “vector” is an array of numbers representing a point in n-space. Various ways to compute distance/similarity of 2 embeddings. Algorithms: cosine, Euclidean, approximate nearest neighbor...



WITH node, vector.similarity.cosine([! embeddings(@input, "myOpenAI") !],



Simplified Bedrock node.js Embedding I/O

```
var bedrockRuntime = new ClientBedrockRuntime.BedrockRuntime(accessArgs);
var params = { "modelId": "amazon.titan-embed-text-v1", }
params.body = JSON.stringify( { "inputText": "this is where you place your text", } );
var data = await bedrockRuntime.invokeModel(params);
console.log(data);
// { '$metadata': {...}
//   body: Uint8ArrayBlobAdapter(16954) [Uint8Array] [
//     123, 34, 101, 109, 98, 101, 100, 100, 105, 110, 103, 34, ... 16854 more items
//   ] }
var replyBodyBlob = data.body;
const asciiDecoder = new TextDecoder('utf-8');
const replyJson = asciiDecoder.decode(replyBodyBlob);
var embeddingHolder = JSON.parse(replyJson);
var vectors = embeddingHolder.embedding;
// { "embedding": [0.123...], "inputTextTokenCount": 8 }
```

The embedding values are:

1. Stored as the field/column value of the DB document/row
2. Used to query the DB for ***similar*** objects



Queries with Vector Similarity Indexes

```
MATCH (node:Movie)
WITH node, vector.similarity.cosine( [! embeddings("dracula", "myBedrock") !],
node.plotEmbedding) AS score
where score is not null
RETURN node, score
ORDER BY score DESCENDING
LIMIT 5
```

Qarbine dynamically determines this value using the Bedrock AI Assistant.

Sample Results

node	score
Document	0.9510183334350586
Document	0.9500061273574829
Document	0.947650671005249
Document	0.9435795545578003
Document	0.942032516002655

Sample Result Element

```
Selection
{
  node: {
    _identity: 1102,
    _labels: [ 1 field ],
    languages: [ 5 fields ],
    plotEmbedding: [ 1536 fields ],
    year: 1992,
    imdbId: "0103874",
    runtime: 128,
    imdbRating: 7.5,
    movieId: "1339",
    countries: [ 1 field ],
    imdbVotes: 138668,
    title: "Dracula (Bram Stoker's Dracula)",
    url: "https://themoviedb.org/movie/6114",
  },
}
```

```
revenue: 215862692,
tmdbId: "6114",
plot: "The vampire comes to England to seduce a visitor's fiancée and inflict havoc in the foreign land.",
posterEmbedding: [ 512 fields ],
poster: "https://image.tmdb.org/t/p/w440_and_h660_face/n39glC4GkBeCbwdenES8ZBodim8.jpg",
released: "1992-11-13",
budget: 40000000
},
score: 0.9500061273574829
```

NOTE - Your query embedding model must match that used to compute the stored embedding values in the first place!



Home >




Data Source

Vector embedding with prompt with embedding

Prompt for user input from which we obtain a vector

Please Provide the Following

 **Vector Search**

Describe what you would like to see

dracula

Base query

```

MATCH (node:Movie)
WITH node, vector.similarity.euclidean([! embeddings(@prompt) !],
node.plotEmbedding) AS score where score is not null
RETURN node, score ORDER BY score DESCENDING LIMIT 3

```

We call Bedrock to obtain the embedding for the user input to form the final Cypher query which is then run.

score	_identity	_labels	languages	plotEmbedding	year	imdbId	runtime
0.0017155787209048867	7609	Array	Array	Array	2009	0485947	141

Selection

```

{
  score: 0.0017155787,
  _identity: 7609,
  _labels: [
    "Movie"
  ],
  languages: [
    "English",
    " Mohawk"
  ],
  plotEmbedding: [
    0.007695260923355818,
    -0.030911803245544434,

```



Execution Flow Across AWS Components

Qarbine “Prompt” for Runtime Values

Please Provide the Following

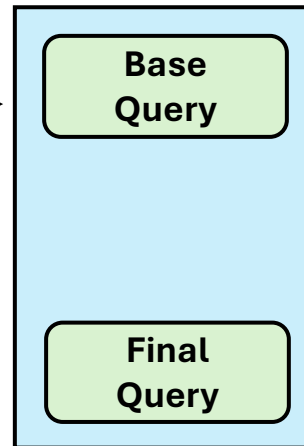
OPEN MOVIE GRAPH Vector Search

Describe what you would like to see

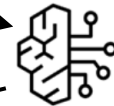
dracula

OK Cancel

Qarbine “Data Source”



Dynamic embedding lookup
vector: [! embedding("dracula") !]



Amazon Bedrock

vector: [embedding numbers]



AWS NoSQL DB

Interactive Analysis Result

Similar Movies Based On- dracula

Year	Movie Title	Revenues	Score
1931	Dracula	\$700,000	0.84
1992	Dracula (Bram Stoker's Dracula)	\$215,862,692	0.83
1979	Dracula	\$216,562,692	0.83



Total Revenue \$216,562,692



Home >



Template

About similar movies

Cypher vector similarity retrieved data formatted for interactive analytics

Design Result Help ? ? ?

⚡ ☁️ 📁 📄 ✎ Page 1 / 1 ⏪ ⏩ ⏴ ⏵

Similar Movies Based On- dracula

1931	Dracula	Revenues	\$700,000		
	IMDBRating 7.60 Runtime 85 Countries USA		Score 0.84		
	The ancient vampire Count Dracula arrives in England and begins to prey upon the virtuous young Mina.				
1992	Dracula (Bram Stoker's Dracula)	Revenues	\$215,862,692		
	IMDBRating 7.50 Runtime 128 Countries USA		Score 0.83		
	The vampire comes to England to seduce a visitor's fiancée and inflict havoc in the foreign land.				
1979	Dracula	Revenues			
	IMDBRating 6.40 Runtime 109 Countries USA, UK		Score 0.83		
	Romanticized adaptation of Bram Stoker's 1897 classic. Set in 1913 England, the bloodsucking, but handsome, charming and seductive, Count Dracula seeks an immortal bride.				
			Total Revenue	\$216,562,692	

Next, given the retrieved and formatted content, let's apply retrieval augmented generation (RAG) to ask Bedrock AI to comment their similarity and revenues



Adding RAG Commentary to the Analysis

Qarbine templates define how to process graph data

You can gain RAG commentary with just 1 Qarbine formula!

=completion(concat("Below is a listing of movies. Comment on their similarity and revenues", formattedResult(false)), "myBedrock", true)

Similar Movies Based On- dracula

1931	Dracula	Revenues	\$700,000	
	IMDBRating	7.60	Runtime	85
	Countries	USA	Score	0.84
The ancient vampire Count Dracula arrives in England and begins to prey upon the virtuous young Mina.				
1992	Dracula (Bram Stoker's Dracula)	Revenues	\$215,862,692	
	IMDBRating	7.50	Runtime	128
	Countries	USA	Score	0.83
The vampire comes to England to seduce a visitor's fiancée and inflict havoc in the foreign land.				
1979	Dracula	Revenues		
	IMDBRating	6.40	Runtime	109
	Countries	USA, UK	Score	0.83
Romanticized adaptation of Bram Stoker's 1897 classic. Set in 1913 England, the bloodsucking, but handsome, charming and seductive, Count Dracula seeks an immortal bride.				




Total Revenue \$216,562,692




Similar Movies Based On- dracula

1931 Dracula Revenues \$700,000 
 IMDBRating 7.60 Runtime 85 Countries USA Score 0.84
 The ancient vampire Count Dracula arrives in England and begins to prey upon the virtuous young Mina.



1992 Dracula (Bram Stoker's Dracula) Revenues \$215,862,692 
 IMDBRating 7.50 Runtime 128 Countries USA Score 0.83
 The vampire comes to England to seduce a visitor's fiancée and inflict havoc in the foreign land.



1979 Dracula Revenues 
 IMDBRating 6.40 Runtime 109 Countries USA, UK Score 0.83
 Romanticized adaptation of Bram Stoker's 1897 classic. Set in 1913 England, the bloodsucking, but handsome, charming and seductive, Count Dracula seeks an immortal bride.



Total Revenue \$216,562,692

All three movies are based on the famous character of Count Dracula from Bram Stoker's novel. They all depict Dracula preying upon young women and causing havoc in various settings.

In terms of revenues, the 1992 version of Dracula (Bram Stoker's Dracula) stands out with a total revenue of \$215,862,692, while the 1931 version of Dracula brought in \$700,000. Unfortunately, the revenue for the 1979 version is not provided in the listing.

Despite their similarities in being adaptations of the same source material, the different versions of Dracula likely drew in different audiences based on factors such as their release year, cast, and director. The 1992 version, directed by Francis Ford Coppola, likely had a larger budget and more marketing push, leading to its higher revenue compared to the 1931 version.

AI insights driven by Retrieval Augmented Generation (RAG) from on-the-fly analytic content using Bedrock AI completions



Interactive "Ask Bedrock AI"

Similar Movies Based On- dracula

1931 Dracula Revenues \$700,000
 IMDBRating 7.60 Runtime 85 Countries USA Score 0.84

The ancient vampire Count Dracula arrives in England and begins to prey upon the virtuous young Mina.



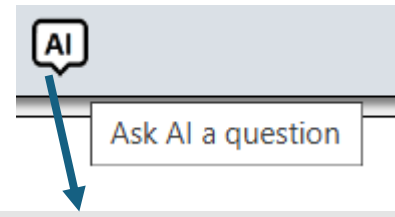
1992 Dracula (Bram Stoker's Dracula) Revenues \$215,862,692
 IMDBRating 7.50 Runtime 128 Countries USA Score 0.83

The vampire comes to England to seduce a visitor's fiancée and inflict havoc in the foreign land.



1979 Dracula Revenues
 IMDBRating 6.40 Runtime 109 Countries USA, UK Score 0.83

Romanticized adaptation of Bram Stoker's 1897 classic. Set in 1913 England, the bloodsucking, but handsome, charming and seductive, Count Dracula seeks an immortal bride.



Ask AI Assistant ✕

Enter your prompt below. Typically it starts by telling the AI Assistant their role followed by your question. For example, "You are a portfolio analyst..."

How are these 3 movies similar?

Which text do you want to append to your prompt?

The selections

Assistant

myBedrock

OK
Cancel

Information ✕

You prompted:
How are these 3 movies similar?

Based on the movie descriptions provided, here are two ways these 3 movies are similar:

1. They all feature Count Dracula as a vampire who comes to England.
2. They are all adaptations or based on Bram Stoker's 1897 novel Dracula.

OK



Interactive "Ask Bedrock AI"

GREATWORTH FINANCIAL Client Portfolio Status Report Thomas Smith, Sr. Advisor

Sandra Danielson Is IRA 12 Burgundy Street A conservative asset mix seeking ...
Personal Account Los Angeles, CA 90210

Cash	\$41,000.00				
Stocks					
Coca-Cola	500	\$45.00	\$47.25 ▲	\$23,625.00	39%
Proctor and Gamble	200	\$80.00	\$105.50 ▲	\$21,100.00	35%
DuPont	300	\$48.25	\$39.50 ▼	\$11,850.00	19%
PG&E	200	\$45.00	\$21.50 ▼	\$4,300.00	7%
					\$60,875.00
Bonds					
2030 New York Municipal			4.5%	\$20,000.00	
2035 California Municipal			5.5%	\$28,000.00	
2025 California Municipal			7.5%	\$25,000.00	
					\$73,000.00
Total portfolio value					\$174,875.00

Ask AI a question

Ask AI Assistant ✕

Enter your prompt below. Typically it starts by telling the AI Assistant their role followed by your question. For example, "You are a portfolio analyst..."

You are a financial analyst. Comment on this portfolio's stock diversity

Which text do you want to append to your prompt?

The selections ▼

? **Information** ✕

You prompted:
You are a financial analyst. Comment of this portfolio's stock diversity.

Based on the limited data provided, I do not have enough information to fully evaluate the stock diversity of this portfolio. I can say that it contains shares in 4 different companies, but I don't know details about the companies, their industries, market caps, etc that would allow me to comment meaningfully on the portfolio's diversity. Without more context, I don't have sufficient information to provide an informed assessment.

📄
OK



Wrap Up - Bedrock AI Broadly Benefits the Lifecycle

1. Using AI Assistants for
 1. Authoring queries
 2. Explaining existing queries
2. Leveraging vector indexes for queries
3. Analyzing answer set content
4. Using ad hoc prompts for vector retrievals
5. Retrieval Augmented Generation (RAG) for analysis