

Generating SQL for SQLite using Ollama, ChromaDB

This notebook runs through the process of using the `vanna` Python package to generate SQL using AI (RAG + LLMs) including connecting to a database and training. If you're not ready to train on your own database, you can still try it using a sample [SQLite database](#).

Which LLM do you want to use?

- [OpenAI via Vanna.AI \(Recommended\)](#)
Use Vanna.AI for free to generate your queries
- [OpenAI](#)
Use OpenAI with your own API key
- [Azure OpenAI](#)
If you have OpenAI models deployed on Azure
- [\[Selected\] Ollama](#)
Use Ollama locally for free. Requires additional setup.
- [Mistral via Mistral API](#)
If you have a Mistral API key
- [Other LLM](#)
If you have a different LLM model

Where do you want to store the 'training' data?

- [Vanna Hosted Vector DB \(Recommended\)](#)
Use Vanna.AI's hosted vector database (pgvector) for free. This is usable across machines with no additional setup.
- [\[Selected\] ChromaDB](#)
Use ChromaDB's open-source vector database for free locally. No additional setup is necessary -- all database files will be created and stored locally.
- [Marqo](#)
Use Marqo locally for free. Requires additional setup. Or use their hosted option.
- [Other VectorDB](#)

Use any other vector database. Requires additional setup.

Setup

```
!pwd!pip install vanna!pip install 'vanna[chromadb]'!pip install ollama!pip show vanna # 0.5.5, 0.2.1!pip show ollama # 0.2.0
```

```
In [1]: from vanna.ollama import Ollama
        from vanna.chromadb.chromadb_vector import ChromaDB_VectorStore
```

```
In [2]: class MyVanna(ChromaDB_VectorStore, Ollama):
        def __init__(self, config=None):
            ChromaDB_VectorStore.__init__(self, config=config)
            Ollama.__init__(self, config=config)

        config = {
            'model': 'llama3' # 'mistral' # "starcoder2"
        }
        vn = MyVanna(config=config)
```

Which database do you want to query?

- [Postgres](#)
- [Microsoft SQL Server](#)
- [DuckDB](#)
- [Snowflake](#)
- [BigQuery](#)
- [\[Selected\] SQLite](#)
- [Other Database](#)

Use Vanna to generate queries for any SQL database

```
In [3]: import os.path
        # file_db = "./db/gpt3sql.sqlite"

        file_db = "~/Downloads/chinook.sqlite"
        file_db = os.path.abspath(os.path.expanduser(file_db))
        vn.connect_to_sqlite(file_db)
```

```
In [4]: vn.run_sql_is_set
```

```
Out[4]: True
```

```
In [5]: def remove_collections(collection_name=None, ACCEPTED_TYPES = ["sql", "ddl", "documentation"]):
    if not collection_name:
        collections = ACCEPTED_TYPES
    elif isinstance(collection_name, str):
        collections = [collection_name]
    elif isinstance(collection_name, list):
        collections = collection_name
    else:
        print(f"\t{collection_name} is unknown: Skipped")
        return

    for c in collections:
        if not c in ACCEPTED_TYPES:
            print(f"\t{c} is unknown: Skipped")
            continue

        # print(f"vn.remove_collection('{c}')"")
        vn.remove_collection(c)
```

```
In [6]: if False:
        remove_collections()
```

Training

You only need to train once. Do not train again unless you want to add more training data.

```
In [7]: # show training data
training_data = vn.get_training_data()
training_data
```

Out[7]:

	id	question	content	training_data_type
0	f80850a9-c302-503b-a436-37abbfe15d4b-sql	SELECT * FROM t_person WHERE name = 'John Doe';	SELECT * FROM t_person WHERE name = 'John Doe'	sql
0	044ba63a-a15d-5339-b4f1-950e93c541be-ddl	None	CREATE TABLE [Invoice]\n(\n [InvoiceId] INT...	ddl
1	291a67cf-a386-5e8a-b858-dee5a9060a31-ddl	None	CREATE TABLE [Artist]\n(\n [ArtistId] INTEG...	ddl
2	3337d9c1-6447-541d-b8d3-9a90f3f85fc8-ddl	None	CREATE INDEX [IFK_AlbumArtistId] ON [Album] ([...	ddl
3	344578ab-d80f-52d4-976e-78a3e54d3c6d-ddl	None	CREATE TABLE [Employee]\n(\n [EmployeeId] I...	ddl
4	3858dfec-459a-53b5-afc9-8d854748161f-ddl	None	CREATE TABLE [MediaType]\n(\n [MediaTypeId]...	ddl
5	52c8d5a3-d118-50af-bb23-2cd719fc02d2-ddl	None	CREATE INDEX [IFK_TrackMediaTypeId] ON [Track]...	ddl
6	5665d55f-7b6d-5d98-8e43-6653428695fa-ddl	None	CREATE TABLE [Album]\n(\n [AlbumId] INTEGER...	ddl
7	5793b49d-25f0-56ac-8557-45389b6ad864-ddl	None	CREATE TABLE [InvoiceLine]\n(\n [InvoiceLin...	ddl
8	58b2d5f7-d7a5-5c6c-8708-7eb383fabb4c-ddl	None	CREATE INDEX [IFK_PlaylistTrackTrackId] ON [Pl...	ddl
9	73b20f77-f874-5c2d-bf6f-94a43bdf5664-ddl	None	CREATE TABLE [PlaylistTrack]\n(\n [Playlist...	ddl
10	7bd3c98d-b531-567d-b9c1-693c7cbbd5e3-ddl	None	CREATE TABLE [Genre]\n(\n [GenreId] INTEGER...	ddl
11	898268bb-3aa1-5c1d-ab43-56a338bdef6f-ddl	None	CREATE TABLE [Customer]\n(\n [CustomerId] I...	ddl
12	8dfd08d5-7eff-5b94-bb42-9a14d6a6501e-ddl	None	CREATE INDEX [IFK_InvoiceCustomerId] ON [Invoi...	ddl
13	93c72821-a041-5508-a9fc-ce75311da7fd-ddl	None	CREATE INDEX [IFK_CustomerSupportRepId] ON [Cu...	ddl
14	9444acf2-6428-5e3c-8ab1-24f92477e88e-ddl	None	CREATE TABLE [Playlist]\n(\n [PlaylistId] I...	ddl
15	9ece6fd1-fa46-5372-b62e-	None	CREATE INDEX [IFK_InvoiceLineInvoiceId]	ddl

	id	question	content	training_data_type
	cdfcecc05ec3-ddl		ON [In...	
16	a4e8be30-d5dd-5a8a-aa99-64c74783e17d-ddl	None	\n CREATE TABLE IF NOT EXISTS t_person (\n ...	ddl
17	a7b444d3-8364-5e23-bafc-3baabf6cb81e-ddl	None	CREATE INDEX [IFK_TrackAlbumId] ON [Track] ([A...	ddl
18	b1fd0669-8857-586b-a285-a7b64d74ad60-ddl	None	CREATE INDEX [IFK_EmployeeReportsTo] ON [Emplo...	ddl
19	bec00095-80eb-5b53-ba91-f82738314c17-ddl	None	CREATE INDEX [IFK_TrackGenreId] ON [Track] ([G...	ddl
20	cc32717f-78e3-5bea-a82e-de699b7a1d47-ddl	None	CREATE INDEX [IFK_InvoiceLineTrackId] ON [Invo...	ddl
21	e6de8429-fd12-5c09-9c76-7725dda2664-ddl	None	CREATE TABLE [Track]\n(\n [TrackId] INTEGER...	ddl
0	51cf1d6d-7637-5b87-b9e7-31c577fbde59-doc	None	Our business defines OTIF score as the percent...	documentation

```
df_ddl = vn.run_sql("SELECT type, sql FROM sqlite_master WHERE sql is not null")df_ddlfor ddl in df_ddl['sql'].to_list(): vn.train(ddl=ddl)
```

```
In [ ]:
```

Asking the AI

Whenever you ask a new question, it will find the 10 most relevant pieces of training data and use it as part of the LLM prompt to generate the SQL.

```
In [8]: vn.ask(question="Show me a list of tables in the SQLite database")
```

Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1

```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE TABLE [Playlist]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Playlist] PRIMARY KEY ([PlaylistId])\n)\n\nCREATE TABLE IF NOT EXISTS t_person (\n    id INT PRIMARY KEY,\n    name VARCHAR(100),\n    email text,\n    age INT\n)\n\nCREATE TABLE [Artist]\n(\n    [ArtistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Artist] PRIMARY KEY ([ArtistId])\n)\n\nCREATE TABLE [PlaylistTrack]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    CONSTRAINT [PK_PlaylistTrack] PRIMARY KEY ([PlaylistId], [TrackId]),\n    FOREIGN KEY ([PlaylistId]) REFERENCES [Playlist] ([PlaylistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [MediaType]\n(\n    [MediaTypeId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_MediaType] PRIMARY KEY ([MediaTypeId])\n)\n\nCREATE TABLE [Album]\n(\n    [AlbumId] INTEGER NOT NULL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER NOT NULL,\n    CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Genre]\n(\n    [GenreId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Genre] PRIMARY KEY ([GenreId])\n)\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(220),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Additional Context\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql\n3. If the provided context is insufficient, please explain why it can't be generated.\n4. Please use the most relevant table(s).\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before.\n"}], {'role': 'user', 'content': " SELECT * FROM t_person WHERE name = 'John Doe';"}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}]
```

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE [Playlist]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Playlist] PRIMARY KEY ([PlaylistId])\n)\n\nCREATE TABLE IF NOT EXISTS t_person (\n    id INT PRIMARY KEY,\n    name VARCHAR(100),\n    email text,\n    age INT\n)\n\nCREATE TABLE [Artist]\n(\n    [ArtistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Artist] PRIMARY KEY ([ArtistId])\n)\n\nCREATE TABLE [PlaylistTrack]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    CONSTRAINT [PK_PlaylistTrack] PRIMARY KEY ([PlaylistId], [TrackId]),\n    FOREIGN KEY ([PlaylistId]) REFERENCES [Playlist] ([PlaylistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [MediaType]\n(\n    [MediaTypeId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_MediaType] PRIMARY KEY ([MediaTypeId])\n)\n\nCREATE TABLE [Album]\n(\n    [AlbumId] INTEGER NOT NULL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER NOT NULL,\n    CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Genre]\n(\n    [GenreId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Genre] PRIMARY KEY ([GenreId])\n)\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(220),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment s
```

aying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " SELECT * FROM t_person WHERE name = 'John Doe';"}, {"role": "assistant", "content": "SELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content": "Show me a list of tables in the SQLite database"}]

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T19:44:21.96272562Z', 'message': {'role': 'assistant', 'content': 'Here is the list of tables:\n\n1. Artist\n2. Album\n3. Genre\n4. MediaType\n5. Playlist\n6. PlaylistTrack\n7. Track\n8. Customer\n9. Invoice\n10. InvoiceLine\n11. t_person'}, 'done_reason': 'stop', 'done': True, 'total_duration': 88766727226, 'load_duration': 1561141376, 'prompt_eval_count': 1247, 'prompt_eval_duration': 77814653000, 'eval_count': 54, 'eval_duration': 9245323000}
```

Here is the list of tables:

1. Artist
2. Album
3. Genre
4. MediaType
5. Playlist
6. PlaylistTrack
7. Track
8. Customer
9. Invoice
10. InvoiceLine
11. t_person

Here is the list of tables:

1. Artist
2. Album
3. Genre
4. MediaType
5. Playlist
6. PlaylistTrack
7. Track
8. Customer
9. Invoice
10. InvoiceLine
11. t_person

Couldn't run sql: Execution failed on sql 'Here is the list of tables:

1. Artist
2. Album
3. Genre

4. MediaType
5. Playlist
6. PlaylistTrack
7. Track
8. Customer
9. Invoice
10. InvoiceLine
11. t_person': near "Here": syntax error

In [9]: `vn.ask(question="How many records are in table called customer")`

Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1

```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE IF NOT EXISTS t_person (\n    id INT PRIMARY KEY,\n    name VARCHAR(100),\n    email text,\n    age INT\n)\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n    FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Album]\n(\n    [AlbumId] INTEGER NOT NULL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER NOT NULL,\n    CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE TABLE [Artist]\n(\n    [ArtistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Artist] PRIMARY KEY ([ArtistId])\n)\n\nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(220),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n\n===Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines\n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with
```

h a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': "SELECT * FROM t_person WHERE name = 'John Doe';"}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': 'How many records are in table called customer'}]

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\t\tON DELETE NO ACTION\nON UPDATE NO ACTION\n)\n\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION\nON UPDATE NO ACTION\n)\n\nCREATE TABLE IF NOT EXISTS t_person (\n    id INT PRIMARY KEY,\n    name VARCHAR(100),\n    email text,\n    age INT\n)\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION\nON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION\nON UPDATE NO ACTION\n)\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n    FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION\nON UPDATE NO ACTION\n)\n\nCREATE TABLE [Album]\n(\n    [AlbumId] INTEGER NOT NULL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER NOT NULL,\n    CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\t\tON DELETE NO ACTION\nON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE TABLE [Artist]\n(\n    [ArtistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Artist] PRIMARY KEY ([ArtistId])\n)\n\nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaType]
```

```

ID INTEGER NOT NULL,\n      [GenreId] INTEGER,\n      [Composer] NVARCHAR(220),\n      [Milliseconds] INTEGER\nNOT NULL,\n      [Bytes] INTEGER,\n      [UnitPrice] NUMERIC(10,2) NOT NULL,\n      CONSTRAINT [PK_Track] PRIMARY\nKEY ([TrackId]),\n      FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\tON DELETE NO ACTION ON\nUPDATE NO ACTION,\n      FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\tON DELETE NO ACTION ON\nUPDATE NO ACTION,\n      FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\tON DELETE N\nO ACTION ON UPDATE NO ACTION\n)\n\n\n===Additional Context \n\nOur business defines OTIF score as the perce\nntage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided conte\nxt is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the\nprovided context is almost sufficient but requires knowledge of a specific string in a particular column, p\nlease generate an intermediate SQL query to find the distinct strings in that column. Prepend the query wit\nh a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it ca\nn't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answe\nred before, please repeat the answer exactly as it was given before. \n"}}, {"role": "user", "content": " SE\nLECT * FROM t_person WHERE name = 'John Doe';"}, {"role": "assistant", "content": "SELECT * FROM t_person W\nHERE name = 'John Doe'"}, {"role": "user", "content": "How many records are in table called customer"}]
```

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T19:45:50.88299427Z', 'message': {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM [Customer];'}, 'done_reason': 'stop', 'done': True, 'total_duration': 88865856942, 'load_duration': 752307, 'prompt_eval_count': 1390, 'prompt_eval_duration': 87523463000, 'eval_count': 8, 'eval_duration': 11880960000}
```

```
SELECT COUNT(*) FROM [Customer];
```

Output from LLM: `SELECT COUNT(*) FROM [Customer];`

Extracted SQL: SELECT COUNT(*) FROM

```
SELECT COUNT(*) FROM
```

Couldn't run sql: Execution failed on sql 'SELECT COUNT(*) FROM ': incomplete input

In []:

```
In [10]: vn.ask(question="what are the top 5 countries that customers come from?")
```

Number of requested results 10 is greater than number of elements in index 1, updating n results = 1

Number of requested results 10 is greater than number of elements in index 1, updating n results = 1

```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n    FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [MediaType]\n(\n    [MediaTypeId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_MediaType] PRIMARY KEY ([MediaTypeId])\n)\n\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE TABLE [Playlist]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Playlist] PRIMARY KEY ([PlaylistId])\n)\n\nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(20),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Album]\n(\n    [AlbumId] INTEGER NOT NULL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER NOT NULL,\n    CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Additional Context\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prep
```

end the query with a comment saying `intermediate_sql \n3`. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': " SELECT * FROM t_person WHERE name = 'John Doe';"}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}]

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n    FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [MediaType]\n(\n    [MediaTypeId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_MediaType] PRIMARY KEY ([MediaTypeId])\n)\n\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE TABLE [Playlist]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Playlist] PRIMARY KEY ([PlaylistId])\n)\n\nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(220),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL
```

```
Extracted SQL: SELECT Country, COUNT(*) AS Count
FROM Customer
GROUP BY Country
ORDER BY Count DESC
LIMIT 5
```

```
SELECT Country, COUNT(*) AS Count
FROM Customer
GROUP BY Country
ORDER BY Count DESC
LIMIT 5
```

	Country	Count
0	USA	13
1	Canada	8
2	France	5
3	Brazil	5
4	Germany	4

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

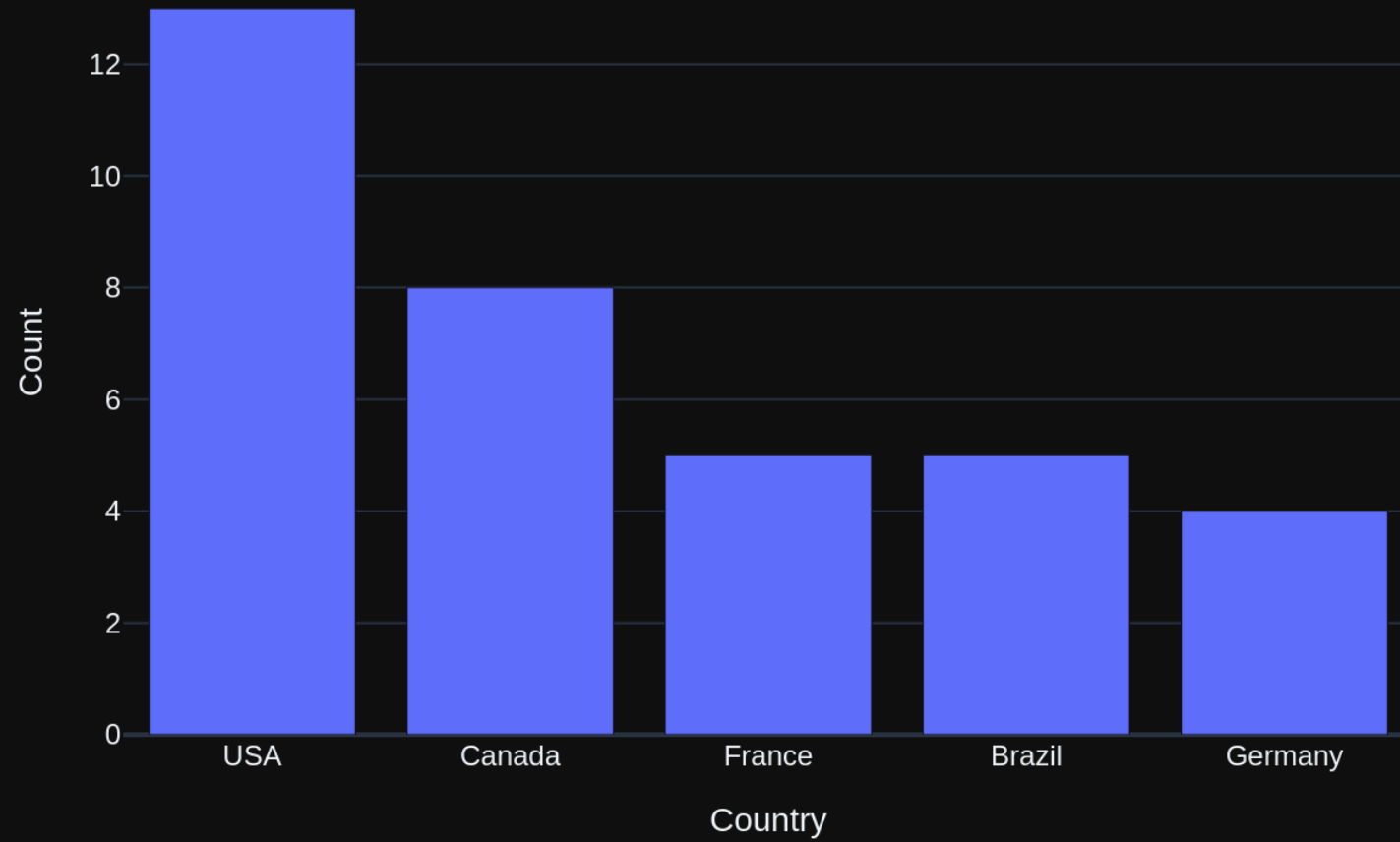
Prompt Content:

```
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: 'what are the top 5 countries that customers come from?'\n\nThe DataFrame was produced using this query: SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5\n\nThe following is information about the resulting pandas DataFrame 'df':\n\nRunning df.dtypes gives:\nCountry    object\nCount      int64\nndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
```

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T19:47:20.855518676Z', 'message': {'role': 'assistant', 'content': "\n\nimport plotly.express as px\n\nfig = px.bar(df, x='Country', y='Count', title='Top 5 Countries by Customer Count')\n\nfig.update_traces(textposition='auto')\nfig.show()\n\n"}, 'done_reason': 'stop', 'done': True, 'total_duration': 18284331905, 'load_duration': 2296769, 'prompt_eval_count': 177, 'prompt_eval_duration': 10540123000, 'eval_count': 48, 'eval_duration': 7636864000}
```


Top 5 Countries by Customer Count



```
Out[10]: ('SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5',
          Country Count
0      USA      13
1    Canada      8
2    France      5
3    Brazil      5
4    Germany      4,
          Figure({
            'data': [{'alignmentgroup': 'True',
                      'hovertemplate': 'Country=%{x}<br>Count=%{y}<extra></extra>',
                      'legendgroup': '',
                      'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                      'name': '',
                      'offsetgroup': '',
                      'orientation': 'v',
                      'showlegend': False,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array(['USA', 'Canada', 'France', 'Brazil', 'Germany'], dtype=object),
                      'xaxis': 'x',
                      'y': array([13, 8, 5, 5, 4]),
                      'yaxis': 'y'}],
            'layout': {'barmode': 'relative',
                      'legend': {'tracegroupgap': 0},
                      'template': '...',
                      'title': {'text': 'Top 5 Countries by Customer Count'},
                      'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Country'}},
                      'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Count'}}
          })
```

More SQL questions

see [sample-sql-queries-sqlite-chinook.ipynb](#)

```
In [11]: question = """
          List all albums and their corresponding artist names
          """

          vn.ask(question=question)
```

```
Number of requested results 10 is greater than number of elements in index 2, updating n_results = 2  
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX [IFK_AlbumArtistId] ON [Album] ([ArtistId])\n\nCREATE INDEX [IFK_TrackAlbumId] ON [Track] ([AlbumId])\n\nCREATE TABLE [Album]\n(\n    [AlbumId] INTEGER NOT NULL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER NOT NULL,\n    CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(220),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Artist]\n(\n    [ArtistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Artist] PRIMARY KEY ([ArtistId])\n)\n\nCREATE INDEX [IFK_TrackGenreId] ON [Track] ([GenreId])\n\nCREATE INDEX [IFK_PlaylistTrackTrackId] ON [PlaylistTrack] ([TrackId])\n\nCREATE INDEX [IFK_TrackMediaTypeId] ON [Track] ([MediaTypeId])\n\nCREATE TABLE [Playlist]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Playlist] PRIMARY KEY ([PlaylistId])\n)\n\nCREATE TABLE [PlaylistTrack]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    CONSTRAINT [PK_PlaylistTrack] PRIMARY KEY ([PlaylistId], [TrackId]),\n    FOREIGN KEY ([PlaylistId]) REFERENCES [Playlist] ([PlaylistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}], {'role': 'user', 'content': " SELECT * FROM t_person WHERE name = 'John Doe';"}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n    List all albums and their corresponding artist names \n'}]
```

Ollama parameters:

```
model=llama3:latest,
options={},
keep_alive=None
```

Prompt Content:

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX [IFK_AlbumArtistId] ON [Album] ([ArtistId])\n\nCREATE INDEX [IFK_TrackAlbumId] ON [Track] ([AlbumId])\n\nCREATE TABLE [Album]\n(\n    [AlbumId] INTEGER NOT NULL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER NOT NULL,\n    CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(220),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Artist]\n(\n    [ArtistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Artist] PRIMARY KEY ([ArtistId])\n)\n\nCREATE INDEX [IFK_TrackGenreId] ON [Track] ([GenreId])\n\nCREATE INDEX [IFK_PlaylistTrackTrackId] ON [PlaylistTrack] ([TrackId])\n\nCREATE INDEX [IFK_TrackMediaTypeId] ON [Track] ([MediaTypeId])\n\nCREATE TABLE [Playlist]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Playlist] PRIMARY KEY ([PlaylistId])\n)\n\nCREATE TABLE [PlaylistTrack]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    CONSTRAINT [PK_PlaylistTrack] PRIMARY KEY ([PlaylistId], [TrackId]),\n    FOREIGN KEY ([PlaylistId]) REFERENCES [Playlist] ([PlaylistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}], {"role": "user", "content": " SELECT * FROM t_person WHERE name = 'John Doe';"}, {"role": "assistant", "content": "SELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content": 'what are the top 5 countries that customers come from?'}, {"role": "assistant", "content": 'SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5'}, {"role": "user", "content": ' \n    List all albums and their corresponding artist names \n'}]
```

```

[IFK_TrackAlbumId] ON [Track] ([AlbumId])\n\nCREATE TABLE [Album]\n(\n    [AlbumId] INTEGER NOT NULL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER NOT NULL,\n    CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(220),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Artist]\n(\n    [ArtistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Artist] PRIMARY KEY ([ArtistId])\n)\n\nCREATE INDEX [IFK_TrackGenreId] ON [Track] ([GenreId])\n\nCREATE INDEX [IFK_PlaylistTrackTrackId] ON [PlaylistTrack] ([TrackId])\n\nCREATE INDEX [IFK_TrackMediaTypeId] ON [Track] ([MediaTypeId])\n\nCREATE TABLE [Playlist]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Playlist] PRIMARY KEY ([PlaylistId])\n)\n\nCREATE TABLE [PlaylistTrack]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    CONSTRAINT [PK_PlaylistTrack] PRIMARY KEY ([PlaylistId], [TrackId]),\n    FOREIGN KEY ([PlaylistId]) REFERENCES [Playlist] ([PlaylistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n\n===Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines\n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql\n3. If the provided context is insufficient, please explain why it can't be generated.\n4. Please use the most relevant table(s).\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before.\n"}
{"role": "user", "content": "SELECT * FROM t_person WHERE name = 'John Doe';"},
{"role": "assistant", "content": "SELECT * FROM t_person WHERE name = 'John Doe'"},
{"role": "user", "content": "what are the top 5 countries that customers come from?"},
{"role": "assistant", "content": "SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5"},
{"role": "user", "content": "\n    List all albums and their corresponding artist names\n"}

```

Ollama Response:

```

{'model': 'llama3:latest', 'created_at': '2024-06-08T19:48:25.135322138Z', 'message': {'role': 'assistant', 'content': 'SELECT A.Title, A.ArtistId, ART.Name\nFROM Album A\nJOIN Artist ART ON A.ArtistId = ART.ArtistId;', 'done_reason': 'stop', 'done': True, 'total_duration': 63620730653, 'load_duration': 597873, 'prompt_eval_count': 956, 'prompt_eval_duration': 58352289000, 'eval_count': 31, 'eval_duration': 5081171000}
SELECT A.Title, A.ArtistId, ART.Name
FROM Album A
JOIN Artist ART ON A.ArtistId = ART.ArtistId;
Output from LLM: SELECT A.Title, A.ArtistId, ART.Name
FROM Album A
JOIN Artist ART ON A.ArtistId = ART.ArtistId;

```

Extracted SQL: SELECT A.Title, A.ArtistId, ART.Name
 FROM Album A
 JOIN Artist ART ON A.ArtistId = ART.ArtistId
 SELECT A.Title, A.ArtistId, ART.Name
 FROM Album A
 JOIN Artist ART ON A.ArtistId = ART.ArtistId

	Title	ArtistId	\
0	For Those About To Rock We Salute You	1	
1	Balls to the Wall	2	
2	Restless and Wild	2	
3	Let There Be Rock	1	
4	Big Ones	3	
..	
342	Respighi:Pines of Rome	226	
343	Schubert: The Late String Quartets & String Qu...	272	
344	Monteverdi: L'Orfeo	273	
345	Mozart: Chamber Music	274	
346	Koyaanisqatsi (Soundtrack from the Motion Pict...	275	

	Name
0	AC/DC
1	Accept
2	Accept
3	AC/DC
4	Aerosmith
..	...
342	Eugene Ormandy
343	Emerson String Quartet
344	C. Monteverdi, Nigel Rogers - Chiaroscuro; Lon...
345	Nash Ensemble
346	Philip Glass Ensemble

[347 rows x 3 columns]

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n List all albums and their corresponding artist names \n'\n\nThe DataFrame was produced using this query: SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJ OIN Artist ART ON A.ArtistId = ART.ArtistId\n\nThe following is information about the resulting pandas Data
```

```

Frame 'df': \nRunning df.dtypes gives:\n Title      object\nArtistId    int64\nName      object\nndtype:
object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the
dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the datafram
e, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the cod
e."}]

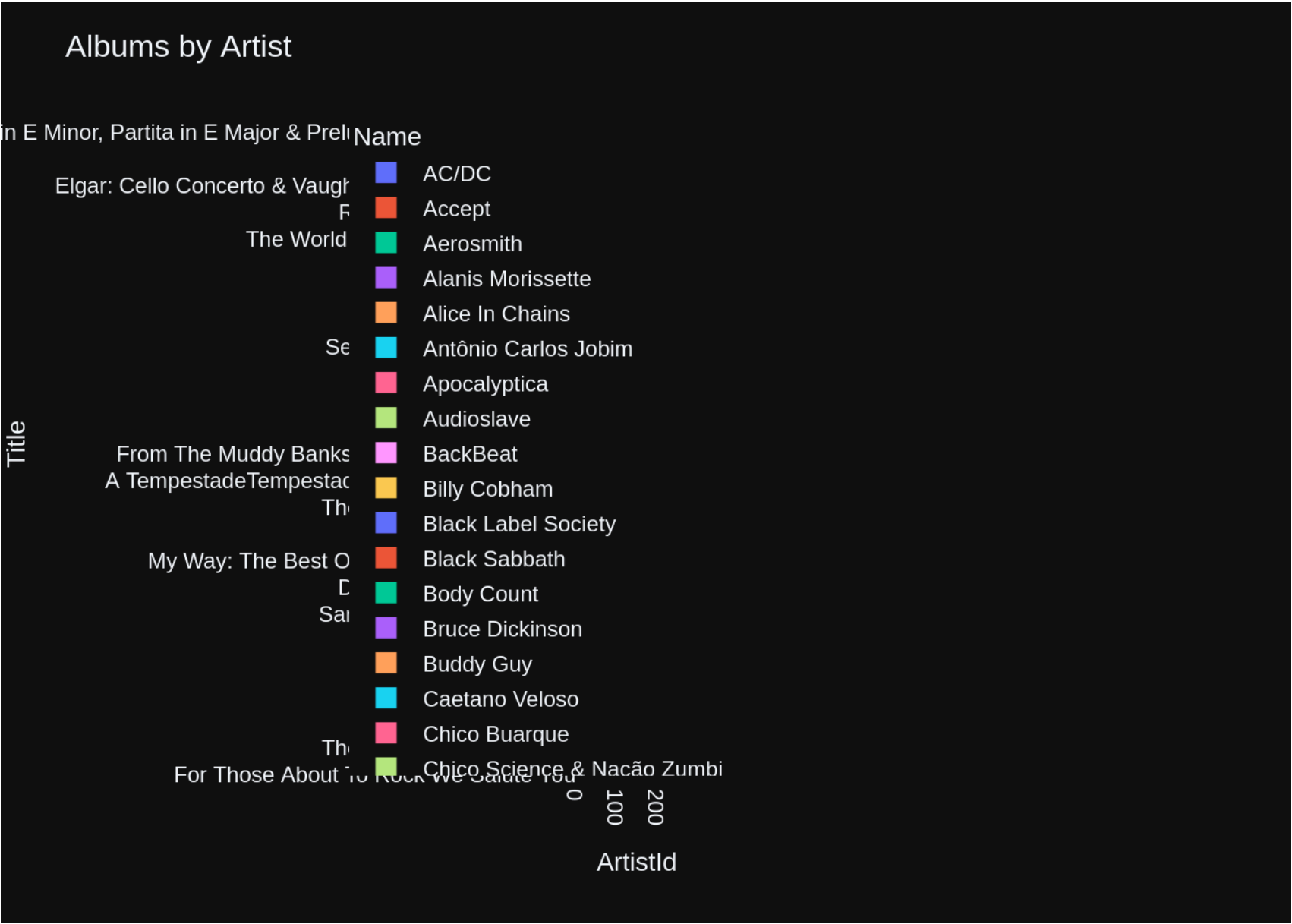
```

Ollama Response:

```

{'model': 'llama3:latest', 'created_at': '2024-06-08T19:48:44.857642836Z', 'message': {'role': 'assistant',
'content': '```\nimport plotly.express as px\nimport numpy as np\n\nfig = px.bar(df, x=\'ArtistId\', y=\'Ti
tle\', color=\'Name\')\nfig.update_layout(title_text="Albums by Artist")\nfig.show()\n```'}, 'done_reason':
'stop', 'done': True, 'total_duration': 19591388932, 'load_duration': 2390420, 'prompt_eval_count': 193, 'p
rompt_eval_duration': 11533911000, 'eval_count': 50, 'eval_duration': 7993266000}

```




```
Out[11]: ('SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJOIN Artist ART ON A.ArtistId = ART.ArtistId',
```

	Title	ArtistId
0	For Those About To Rock We Salute You	1
1	Balls to the Wall	2
2	Restless and Wild	2
3	Let There Be Rock	1
4	Big Ones	3
..
342	Respighi:Pines of Rome	226
343	Schubert: The Late String Quartets & String Qu...	272
344	Monteverdi: L'Orfeo	273
345	Mozart: Chamber Music	274
346	Koyaanisqatsi (Soundtrack from the Motion Pict...	275

	Name
0	AC/DC
1	Accept
2	Accept
3	AC/DC
4	Aerosmith
..	...
342	Eugene Ormandy
343	Emerson String Quartet
344	C. Monteverdi, Nigel Rogers - Chiaroscuro; Lon...
345	Nash Ensemble
346	Philip Glass Ensemble

```
[347 rows x 3 columns],
```

```
Figure({
  'data': [{ 'alignmentgroup': 'True',
             'hovertemplate': 'Name=AC/DC<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
             'legendgroup': 'AC/DC',
             'marker': { 'color': '#636efa', 'pattern': { 'shape': '' } },
             'name': 'AC/DC',
             'offsetgroup': 'AC/DC',
             'orientation': 'h',
             'showlegend': True,
             'textposition': 'auto',
             'type': 'bar',
             'x': array([1, 1]),
             'xaxis': 'x',
             'y': array(['For Those About To Rock We Salute You', 'Let There Be Rock']),
```

```

        dtype=object),
    'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Accept<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Accept',
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': 'Accept',
 'offsetgroup': 'Accept',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([2, 2]),
 'xaxis': 'x',
 'y': array(['Balls to the Wall', 'Restless and Wild'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Aerosmith<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Aerosmith',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'Aerosmith',
 'offsetgroup': 'Aerosmith',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([3]),
 'xaxis': 'x',
 'y': array(['Big Ones'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Alanis Morissette<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Alanis Morissette',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Alanis Morissette',
 'offsetgroup': 'Alanis Morissette',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([4]),
 'xaxis': 'x',

```

```

'y': array(['Jagged Little Pill'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Alice In Chains<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Alice In Chains',
'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
'name': 'Alice In Chains',
'offsetgroup': 'Alice In Chains',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([5]),
'xaxis': 'x',
'y': array(['Facelift'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Antônio Carlos Jobim<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Antônio Carlos Jobim',
'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
'name': 'Antônio Carlos Jobim',
'offsetgroup': 'Antônio Carlos Jobim',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([6, 6]),
'xaxis': 'x',
'y': array(['Warner 25 Anos', 'Chill: Brazil (Disc 2)'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Apocalyptica<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Apocalyptica',
'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
'name': 'Apocalyptica',
'offsetgroup': 'Apocalyptica',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([7]),
'xaxis': 'x',

```

```

'y': array(['Plays Metallica By Four Cellos'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Audioslave<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Audioslave',
'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
'name': 'Audioslave',
'offsetgroup': 'Audioslave',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([8, 8, 8]),
'xaxis': 'x',
'y': array(['Audioslave', 'Out Of Exile', 'Revelations'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=BackBeat<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'BackBeat',
'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
'name': 'BackBeat',
'offsetgroup': 'BackBeat',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([9]),
'xaxis': 'x',
'y': array(['BackBeat Soundtrack'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Billy Cobham<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Billy Cobham',
'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
'name': 'Billy Cobham',
'offsetgroup': 'Billy Cobham',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([10]),
'xaxis': 'x',

```

```

'y': array(['The Best Of Billy Cobham'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Black Label Society<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Black Label Society',
'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
'name': 'Black Label Society',
'offsetgroup': 'Black Label Society',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([11, 11]),
'xaxis': 'x',
'y': array(['Alcohol Fueled Brewtality Live! [Disc 1]',
            'Alcohol Fueled Brewtality Live! [Disc 2]'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Black Sabbath<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Black Sabbath',
'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
'name': 'Black Sabbath',
'offsetgroup': 'Black Sabbath',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([12, 12]),
'xaxis': 'x',
'y': array(['Black Sabbath', 'Black Sabbath Vol. 4 (Remaster)'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Body Count<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Body Count',
'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
'name': 'Body Count',
'offsetgroup': 'Body Count',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([13]),

```

```

    'xaxis': 'x',
    'y': array(['Body Count'], dtype=object),
    'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Bruce Dickinson<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Bruce Dickinson',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Bruce Dickinson',
 'offsetgroup': 'Bruce Dickinson',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([14]),
 'xaxis': 'x',
 'y': array(['Chemical Wedding'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Buddy Guy<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Buddy Guy',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'Buddy Guy',
 'offsetgroup': 'Buddy Guy',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([15]),
 'xaxis': 'x',
 'y': array(['The Best Of Buddy Guy - The Millenium Collection'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Caetano Veloso<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Caetano Veloso',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'Caetano Veloso',
 'offsetgroup': 'Caetano Veloso',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([16, 16]),

```

```

    'xaxis': 'x',
    'y': array(['Prenda Minha', 'Sozinho Remix Ao Vivo'], dtype=object),
    'yaxis': 'y'},
    {'alignmentgroup': 'True',
    'hovertemplate': 'Name=Chico Buarque<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
    'legendgroup': 'Chico Buarque',
    'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
    'name': 'Chico Buarque',
    'offsetgroup': 'Chico Buarque',
    'orientation': 'h',
    'showlegend': True,
    'textposition': 'auto',
    'type': 'bar',
    'x': array([17]),
    'xaxis': 'x',
    'y': array(['Minha Historia'], dtype=object),
    'yaxis': 'y'},
    {'alignmentgroup': 'True',
    'hovertemplate': 'Name=Chico Science & Nação Zumbi<br>ArtistId=%{x}<br>Title=%{y}<extra></e
xtra>',
    'legendgroup': 'Chico Science & Nação Zumbi',
    'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
    'name': 'Chico Science & Nação Zumbi',
    'offsetgroup': 'Chico Science & Nação Zumbi',
    'orientation': 'h',
    'showlegend': True,
    'textposition': 'auto',
    'type': 'bar',
    'x': array([18, 18]),
    'xaxis': 'x',
    'y': array(['Afrociberdelia', 'Da Lama Ao Caos'], dtype=object),
    'yaxis': 'y'},
    {'alignmentgroup': 'True',
    'hovertemplate': 'Name=Cidade Negra<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
    'legendgroup': 'Cidade Negra',
    'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
    'name': 'Cidade Negra',
    'offsetgroup': 'Cidade Negra',
    'orientation': 'h',
    'showlegend': True,
    'textposition': 'auto',
    'type': 'bar',

```

```

    'x': array([19, 19]),
    'xaxis': 'x',
    'y': array(['Acústico MTV [Live]', 'Cidade Negra - Hits'], dtype=object),
    'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovernplate': 'Name=Cláudio Zoli<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Cláudio Zoli',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'Cláudio Zoli',
 'offsetgroup': 'Cláudio Zoli',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([20]),
 'xaxis': 'x',
 'y': array(['Na Pista'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovernplate': 'Name=Various Artists<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Various Artists',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': 'Various Artists',
 'offsetgroup': 'Various Artists',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([21, 21, 21, 21]),
 'xaxis': 'x',
 'y': array(['Axé Bahia 2001', 'Carnaval 2001', 'Sambas De Enredo 2001',
            'Vozes do MPB'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovernplate': 'Name=Led Zeppelin<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Led Zeppelin',
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': 'Led Zeppelin',
 'offsetgroup': 'Led Zeppelin',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',

```



```

'type': 'bar',
'x': array([22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22]),
'xaxis': 'x',
'y': array(['BBC Sessions [Disc 1] [Live]', 'Physical Graffiti [Disc 1]',
           'BBC Sessions [Disc 2] [Live]', 'Coda', 'Houses Of The Holy',
           'In Through The Out Door', 'IV', 'Led Zeppelin I', 'Led Zeppelin II',
           'Led Zeppelin III', 'Physical Graffiti [Disc 2]', 'Presence',
           'The Song Remains The Same (Disc 1)',
           'The Song Remains The Same (Disc 2)'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovernplate': ('Name=Frank Zappa & Captain Bee' ... ')<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Frank Zappa & Captain Beefheart',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'Frank Zappa & Captain Beefheart',
 'offsetgroup': 'Frank Zappa & Captain Beefheart',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([23]),
 'xaxis': 'x',
 'y': array(['Bongo Fury'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovernplate': 'Name=Marcos Valle<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Marcos Valle',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Marcos Valle',
 'offsetgroup': 'Marcos Valle',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([24]),
 'xaxis': 'x',
 'y': array(['Chill: Brazil (Disc 1)'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovernplate': 'Name=Metallica<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Metallica',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},

```

```

'name': 'Metallica',
'offsetgroup': 'Metallica',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([50, 50, 50, 50, 50, 50, 50, 50, 50, 50]),
'xaxis': 'x',
'y': array(['Garage Inc. (Disc 1)', 'Black Album', 'Garage Inc. (Disc 2)',
           'Kill 'Em All', 'Load', 'Master Of Puppets', 'ReLoad',
           'Ride The Lightning', 'St. Anger', '...And Justice For All'],
          dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Queen<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Queen',
'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
'name': 'Queen',
'offsetgroup': 'Queen',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([51, 51, 51]),
'xaxis': 'x',
'y': array(['Greatest Hits II', 'Greatest Hits I', 'News Of The World'],
          dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Kiss<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Kiss',
'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
'name': 'Kiss',
'offsetgroup': 'Kiss',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([52, 52]),
'xaxis': 'x',
'y': array(['Greatest Kiss', 'Unplugged [Live]'], dtype=object),
'yaxis': 'y'},

```

```

{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Spyro Gyra<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Spyro Gyra',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'Spyro Gyra',
 'offsetgroup': 'Spyro Gyra',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([53, 53]),
 'xaxis': 'x',
 'y': array(['Heart of the Night', 'Morning Dance'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Green Day<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Green Day',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'Green Day',
 'offsetgroup': 'Green Day',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([54, 54]),
 'xaxis': 'x',
 'y': array(['International Superhits', 'American Idiot'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=David Coverdale<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'David Coverdale',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'David Coverdale',
 'offsetgroup': 'David Coverdale',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([55]),
 'xaxis': 'x',
 'y': array(['Into The Light'], dtype=object),
 'yaxis': 'y'},

```

```

{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Gonzaguinha<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Gonzaguinha',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': 'Gonzaguinha',
 'offsetgroup': 'Gonzaguinha',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([56]),
 'xaxis': 'x',
 'y': array(['Meus Momentos'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Os Mutantes<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Os Mutantes',
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': 'Os Mutantes',
 'offsetgroup': 'Os Mutantes',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([57]),
 'xaxis': 'x',
 'y': array(['Minha História'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Deep Purple<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Deep Purple',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'Deep Purple',
 'offsetgroup': 'Deep Purple',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58]),
 'xaxis': 'x',
 'y': array(['MK III The Final Concerts [Disc 1]', 'The Final Concerts (Disc 2)',
            'Come Taste The Band', 'Deep Purple In Rock', 'Fireball'],

```

```

        "Knocking at Your Back Door: The Best Of Deep Purple in the 80's",
        'Machine Head', 'Purpendicular', 'Slaves And Masters', 'Stormbringer',
        'The Battle Rages On'], dtype=object),
    'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Santana<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Santana',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Santana',
 'offsetgroup': 'Santana',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([59, 59, 59]),
 'xaxis': 'x',
 'y': array(['Supernatural', 'Santana - As Years Go By', 'Santana Live'],
            dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Ed Motta<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Ed Motta',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'Ed Motta',
 'offsetgroup': 'Ed Motta',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([37]),
 'xaxis': 'x',
 'y': array(['The Best of Ed Motta'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Miles Davis<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Miles Davis',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'Miles Davis',
 'offsetgroup': 'Miles Davis',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',

```

```

        'type': 'bar',
        'x': array([68, 68, 68]),
        'xaxis': 'x',
        'y': array(['The Essential Miles Davis [Disc 1]',
                    'The Essential Miles Davis [Disc 2]', 'Miles Ahead'], dtype=object),
        'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Gene Krupa<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Gene Krupa',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'Gene Krupa',
 'offsetgroup': 'Gene Krupa',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([69]),
 'xaxis': 'x',
 'y': array(["Up An' Atom"], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Toquinho & Vinícius<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Toquinho & Vinícius',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'Toquinho & Vinícius',
 'offsetgroup': 'Toquinho & Vinícius',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([70]),
 'xaxis': 'x',
 'y': array(['Vinícius De Moraes - Sem Limite'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Creedence Clearwater Revival<br>ArtistId=%{x}<br>Title=%{y}<extra></
extra>',
 'legendgroup': 'Creedence Clearwater Revival',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'Creedence Clearwater Revival',
 'offsetgroup': 'Creedence Clearwater Revival',
 'orientation': 'h',

```

```

'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([76, 76]),
'xaxis': 'x',
'y': array(['Chronicle, Vol. 1', 'Chronicle, Vol. 2'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Cássia Eller<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Cássia Eller',
'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
'name': 'Cássia Eller',
'offsetgroup': 'Cássia Eller',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([77, 77]),
'xaxis': 'x',
'y': array(['Cássia Eller - Coleção Sem Limite [Disc 2]',
'Cássia Eller - Sem Limite [Disc 1]'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Def Leppard<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Def Leppard',
'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
'name': 'Def Leppard',
'offsetgroup': 'Def Leppard',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([78]),
'xaxis': 'x',
'y': array(["Vault: Def Leppard's Greatest Hits"], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Dennis Chambers<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Dennis Chambers',
'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
'name': 'Dennis Chambers',
'offsetgroup': 'Dennis Chambers',

```

```

'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([79]),
'xaxis': 'x',
'y': array(['Outbreak'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Djavan<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Djavan',
'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
'name': 'Djavan',
'offsetgroup': 'Djavan',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([80, 80]),
'xaxis': 'x',
'y': array(['Djavan Ao Vivo - Vol. 02', 'Djavan Ao Vivo - Vol. 1'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Elis Regina<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Elis Regina',
'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
'name': 'Elis Regina',
'offsetgroup': 'Elis Regina',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([41]),
'xaxis': 'x',
'y': array(['Elis Regina-Minha História'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Eric Clapton<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Eric Clapton',
'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
'name': 'Eric Clapton',
'offsetgroup': 'Eric Clapton',

```



```

'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([81, 81]),
'xaxis': 'x',
'y': array(['The Cream Of Clapton', 'Unplugged'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Faith No More<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Faith No More',
'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
'name': 'Faith No More',
'offsetgroup': 'Faith No More',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([82, 82, 82, 82]),
'xaxis': 'x',
'y': array(['Album Of The Year', 'Angel Dust', 'King For A Day Fool For A Lifetime',
            'The Real Thing'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Falamansa<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Falamansa',
'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
'name': 'Falamansa',
'offsetgroup': 'Falamansa',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([83]),
'xaxis': 'x',
'y': array(['Deixa Entrar'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Foo Fighters<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Foo Fighters',
'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
'name': 'Foo Fighters',

```

```

'offsetgroup': 'Foo Fighters',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([84, 84, 84, 84]),
'xaxis': 'x',
'y': array(['In Your Honor [Disc 1]', 'In Your Honor [Disc 2]', 'One By One',
           'The Colour And The Shape'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovernplate': 'Name=Frank Sinatra<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Frank Sinatra',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'Frank Sinatra',
 'offsetgroup': 'Frank Sinatra',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([85]),
 'xaxis': 'x',
 'y': array(['My Way: The Best Of Frank Sinatra [Disc 1]'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovernplate': 'Name=Funk Como Le Gusta<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Funk Como Le Gusta',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'Funk Como Le Gusta',
 'offsetgroup': 'Funk Como Le Gusta',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([86]),
 'xaxis': 'x',
 'y': array(['Roda De Funk'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovernplate': 'Name=Gilberto Gil<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Gilberto Gil',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},

```

```

'name': 'Gilberto Gil',
'offsetgroup': 'Gilberto Gil',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([27, 27, 27]),
'xaxis': 'x',
'y': array(['As Canções de Eu Tu Eles', 'Quanta Gente Veio Ver (Live)',
           'Quanta Gente Veio ver--Bônus De Carnaval'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Godsmack<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Godsmack',
'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
'name': 'Godsmack',
'offsetgroup': 'Godsmack',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([87]),
'xaxis': 'x',
'y': array(['Faceless'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': "Name=Guns N' Roses<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>",
'legendgroup': "Guns N' Roses",
'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
'name': "Guns N' Roses",
'offsetgroup': "Guns N' Roses",
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([88, 88, 88]),
'xaxis': 'x',
'y': array(['Appetite for Destruction', 'Use Your Illusion I',
           'Use Your Illusion II'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Incognito<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',

```

```

'legendgroup': 'Incognito',
'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
'name': 'Incognito',
'offsetgroup': 'Incognito',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([89]),
'xaxis': 'x',
'y': array(['Blue Moods'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Iron Maiden<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Iron Maiden',
'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
'name': 'Iron Maiden',
'offsetgroup': 'Iron Maiden',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90,
          90, 90, 90]),
'xaxis': 'x',
'y': array(['A Matter of Life and Death', 'A Real Dead One', 'A Real Live One',
          'Brave New World', 'Dance Of Death', 'Fear Of The Dark', 'Iron Maiden',
          'Killers', 'Live After Death', 'Live At Donington 1992 (Disc 1)',
          'Live At Donington 1992 (Disc 2)', 'No Prayer For The Dying',
          'Piece Of Mind', 'Powerslave', 'Rock In Rio [CD1]', 'Rock In Rio [CD2]',
          'Seventh Son of a Seventh Son', 'Somewhere in Time',
          'The Number of The Beast', 'The X Factor', 'Virtual XI'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=James Brown<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'James Brown',
'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
'name': 'James Brown',
'offsetgroup': 'James Brown',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',

```

```

    'type': 'bar',
    'x': array([91]),
    'xaxis': 'x',
    'y': array(['Sex Machine'], dtype=object),
    'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Jamiroquai<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Jamiroquai',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'Jamiroquai',
 'offsetgroup': 'Jamiroquai',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([92, 92, 92]),
 'xaxis': 'x',
 'y': array(['Emergency On Planet Earth', 'Synkronized',
            'The Return Of The Space Cowboy'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=JET<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'JET',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'JET',
 'offsetgroup': 'JET',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([93]),
 'xaxis': 'x',
 'y': array(['Get Born'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Jimi Hendrix<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Jimi Hendrix',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'Jimi Hendrix',
 'offsetgroup': 'Jimi Hendrix',
 'orientation': 'h',
 'showlegend': True,

```

```

'textposition': 'auto',
'type': 'bar',
'x': array([94]),
'xaxis': 'x',
'y': array(['Are You Experienced?'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Joe Satriani<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Joe Satriani',
'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
'name': 'Joe Satriani',
'offsetgroup': 'Joe Satriani',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([95]),
'xaxis': 'x',
'y': array(['Surfing with the Alien (Remastered)'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Jorge Ben<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Jorge Ben',
'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
'name': 'Jorge Ben',
'offsetgroup': 'Jorge Ben',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([46]),
'xaxis': 'x',
'y': array(['Jorge Ben Jor 25 Anos'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Jota Quest<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Jota Quest',
'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
'name': 'Jota Quest',
'offsetgroup': 'Jota Quest',
'orientation': 'h',
'showlegend': True,

```

```

'textposition': 'auto',
'type': 'bar',
'x': array([96]),
'xaxis': 'x',
'y': array(['Jota Quest-1995'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=João Suplicy<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'João Suplicy',
'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
'name': 'João Suplicy',
'offsetgroup': 'João Suplicy',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([97]),
'xaxis': 'x',
'y': array(['Cafezinho'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Judas Priest<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Judas Priest',
'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
'name': 'Judas Priest',
'offsetgroup': 'Judas Priest',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([98]),
'xaxis': 'x',
'y': array(['Living After Midnight'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Legião Urbana<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Legião Urbana',
'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
'name': 'Legião Urbana',
'offsetgroup': 'Legião Urbana',
'orientation': 'h',
'showlegend': True,

```

```

'textposition': 'auto',
'type': 'bar',
'x': array([99, 99]),
'xaxis': 'x',
'y': array(['A TempestadeTempestade Ou 0 Livro Dos Dias', 'Mais Do Mesmo'],
          dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Lenny Kravitz<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Lenny Kravitz',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'Lenny Kravitz',
 'offsetgroup': 'Lenny Kravitz',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([100]),
 'xaxis': 'x',
 'y': array(['Greatest Hits'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Lulu Santos<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Lulu Santos',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'Lulu Santos',
 'offsetgroup': 'Lulu Santos',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([101, 101]),
 'xaxis': 'x',
 'y': array(['Lulu Santos - RCA 100 Anos De Música - Álbum 01',
            'Lulu Santos - RCA 100 Anos De Música - Álbum 02'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Marillion<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Marillion',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'Marillion',
 'offsetgroup': 'Marillion',

```



```

    'orientation': 'h',
    'showlegend': True,
    'textposition': 'auto',
    'type': 'bar',
    'x': array([102]),
    'xaxis': 'x',
    'y': array(['Misplaced Childhood'], dtype=object),
    'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovernplate': 'Name=Marisa Monte<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Marisa Monte',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'Marisa Monte',
 'offsetgroup': 'Marisa Monte',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([103]),
 'xaxis': 'x',
 'y': array(['Barulhinho Bom'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovernplate': 'Name=Marvin Gaye<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Marvin Gaye',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'Marvin Gaye',
 'offsetgroup': 'Marvin Gaye',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([104]),
 'xaxis': 'x',
 'y': array(['Seek And Shall Find: More Of The Best (1963-1981)'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovernplate': 'Name=Men At Work<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Men At Work',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': 'Men At Work',
 'offsetgroup': 'Men At Work',

```

```

'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([105]),
'xaxis': 'x',
'y': array(['The Best Of Men At Work'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Milton Nascimento<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Milton Nascimento',
'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
'name': 'Milton Nascimento',
'offsetgroup': 'Milton Nascimento',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([42, 42]),
'xaxis': 'x',
'y': array(['Milton Nascimento Ao Vivo', 'Minas'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Motörhead<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Motörhead',
'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
'name': 'Motörhead',
'offsetgroup': 'Motörhead',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([106]),
'xaxis': 'x',
'y': array(['Ace Of Spades'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Mônica Marianno<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Mônica Marianno',
'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
'name': 'Mônica Marianno',
'offsetgroup': 'Mônica Marianno',

```

```

'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([108]),
'xaxis': 'x',
'y': array(['Demorou...'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Mötley Crüe<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Mötley Crüe',
'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
'name': 'Mötley Crüe',
'offsetgroup': 'Mötley Crüe',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([109]),
'xaxis': 'x',
'y': array(['Motley Crue Greatest Hits'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Nirvana<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Nirvana',
'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
'name': 'Nirvana',
'offsetgroup': 'Nirvana',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([110, 110]),
'xaxis': 'x',
'y': array(['From The Muddy Banks Of The Wishkah [Live]', 'Nevermind'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=0 Terço<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': '0 Terço',
'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
'name': '0 Terço',
'offsetgroup': '0 Terço',

```

```

        'orientation': 'h',
        'showlegend': True,
        'textposition': 'auto',
        'type': 'bar',
        'x': array([111]),
        'xaxis': 'x',
        'y': array(['Compositores'], dtype=object),
        'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovernplate': 'Name=Olodum<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
     'legendgroup': 'Olodum',
     'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
     'name': 'Olodum',
     'offsetgroup': 'Olodum',
     'orientation': 'h',
     'showlegend': True,
     'textposition': 'auto',
     'type': 'bar',
     'x': array([112]),
     'xaxis': 'x',
     'y': array(['Olodum'], dtype=object),
     'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovernplate': 'Name=Os Paralamas Do Sucesso<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>
>',
     'legendgroup': 'Os Paralamas Do Sucesso',
     'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
     'name': 'Os Paralamas Do Sucesso',
     'offsetgroup': 'Os Paralamas Do Sucesso',
     'orientation': 'h',
     'showlegend': True,
     'textposition': 'auto',
     'type': 'bar',
     'x': array([113, 113, 113]),
     'xaxis': 'x',
     'y': array(['Acústico MTV', 'Arquivo II', 'Arquivo Os Paralamas Do Sucesso'],
                 dtype=object),
     'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovernplate': 'Name=Ozzy Osbourne<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
     'legendgroup': 'Ozzy Osbourne',
     'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},

```

```

'name': 'Ozzy Osbourne',
'offsetgroup': 'Ozzy Osbourne',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([114, 114, 114, 114, 114, 114]),
'xaxis': 'x',
'y': array(['Bark at the Moon (Remastered)', 'Blizzard of Ozz',
          'Diary of a Madman (Remastered)', 'No More Tears (Remastered)',
          'Tribute', 'Speak of the Devil'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Page & Plant<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Page & Plant',
'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
'name': 'Page & Plant',
'offsetgroup': 'Page & Plant',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([115]),
'xaxis': 'x',
'y': array(['Walking Into Clarksdale'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Passengers<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Passengers',
'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
'name': 'Passengers',
'offsetgroup': 'Passengers',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([116]),
'xaxis': 'x',
'y': array(['Original Soundtracks 1'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': "Name=Paul D'Ianno<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>",

```

```

'legendgroup': "Paul D'Ianno",
'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
'name': "Paul D'Ianno",
'offsetgroup': "Paul D'Ianno",
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([117]),
'xaxis': 'x',
'y': array(['The Beast Live'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Pearl Jam<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Pearl Jam',
'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
'name': 'Pearl Jam',
'offsetgroup': 'Pearl Jam',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([118, 118, 118, 118, 118]),
'xaxis': 'x',
'y': array(['Live On Two Legs [Live]', 'Pearl Jam', 'Riot Act', 'Ten', 'Vs.'],
dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Pink Floyd<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Pink Floyd',
'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
'name': 'Pink Floyd',
'offsetgroup': 'Pink Floyd',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([120]),
'xaxis': 'x',
'y': array(['Dark Side Of The Moon'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',

```

```

'hovertemplate': 'Name=Planet Hemp<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Planet Hemp',
'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
'name': 'Planet Hemp',
'offsetgroup': 'Planet Hemp',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([121]),
'xaxis': 'x',
'y': array(['0s Cães Ladram Mas A Caravana Não Pára'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=R.E.M. Feat. Kate Pearson<br>ArtistId=%{x}<br>Title=%{y}<extra></ext
ra>',
'legendgroup': 'R.E.M. Feat. Kate Pearson',
'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
'name': 'R.E.M. Feat. Kate Pearson',
'offsetgroup': 'R.E.M. Feat. Kate Pearson',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([122]),
'xaxis': 'x',
'y': array(['0ut Of Time'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=R.E.M.<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'R.E.M.',
'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
'name': 'R.E.M.',
'offsetgroup': 'R.E.M.',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([124, 124, 124]),
'xaxis': 'x',
'y': array(['Green', 'New Adventures In Hi-Fi', 'The Best Of R.E.M.: The IRS Years'],
dtype=object),

```

```

'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Raimundos<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Raimundos',
'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
'name': 'Raimundos',
'offsetgroup': 'Raimundos',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([125]),
'xaxis': 'x',
'y': array(['Cesta Básica'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Raul Seixas<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Raul Seixas',
'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
'name': 'Raul Seixas',
'offsetgroup': 'Raul Seixas',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([126]),
'xaxis': 'x',
'y': array(['Raul Seixas'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Red Hot Chili Peppers<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Red Hot Chili Peppers',
'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
'name': 'Red Hot Chili Peppers',
'offsetgroup': 'Red Hot Chili Peppers',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([127, 127, 127]),
'xaxis': 'x',

```



```

'y': array(['Blood Sugar Sex Magik', 'By The Way', 'Californication'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Rush<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Rush',
'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
'name': 'Rush',
'offsetgroup': 'Rush',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([128]),
'xaxis': 'x',
'y': array(['Retrospective I (1974-1980)'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Skank<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Skank',
'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
'name': 'Skank',
'offsetgroup': 'Skank',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([130, 130]),
'xaxis': 'x',
'y': array(['Maquinarama', 'O Samba Poconé'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Smashing Pumpkins<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Smashing Pumpkins',
'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
'name': 'Smashing Pumpkins',
'offsetgroup': 'Smashing Pumpkins',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([131, 131]),
'xaxis': 'x',

```

```

'y': array(['Judas 0: B-Sides and Rarities', 'Rotten Apples: Greatest Hits'],
          dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Soundgarden<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Soundgarden',
'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
'name': 'Soundgarden',
'offsetgroup': 'Soundgarden',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([132]),
'xaxis': 'x',
'y': array(['A-Sides'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Stevie Ray Vaughan & Doub' ... ')<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Stevie Ray Vaughan & Double Trouble',
'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
'name': 'Stevie Ray Vaughan & Double Trouble',
'offsetgroup': 'Stevie Ray Vaughan & Double Trouble',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([133]),
'xaxis': 'x',
'y': array(['In Step'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Stone Temple Pilots<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Stone Temple Pilots',
'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
'name': 'Stone Temple Pilots',
'offsetgroup': 'Stone Temple Pilots',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([134]),

```

```

    'xaxis': 'x',
    'y': array(['Core'], dtype=object),
    'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=System Of A Down<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'System Of A Down',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'System Of A Down',
 'offsetgroup': 'System Of A Down',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([135]),
 'xaxis': 'x',
 'y': array(['Mezmerize'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('Name=Terry Bozzio, Tony Levin ' ... ')<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Terry Bozzio, Tony Levin & Steve Stevens',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'Terry Bozzio, Tony Levin & Steve Stevens',
 'offsetgroup': 'Terry Bozzio, Tony Levin & Steve Stevens',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([136]),
 'xaxis': 'x',
 'y': array(['[1997] Black Light Syndrome'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=The Black Crowes<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'The Black Crowes',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'The Black Crowes',
 'offsetgroup': 'The Black Crowes',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([137, 137]),

```

```

    'xaxis': 'x',
    'y': array(['Live [Disc 1]', 'Live [Disc 2]'], dtype=object),
    'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovertemplate': 'Name=The Clash<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
     'legendgroup': 'The Clash',
     'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
     'name': 'The Clash',
     'offsetgroup': 'The Clash',
     'orientation': 'h',
     'showlegend': True,
     'textposition': 'auto',
     'type': 'bar',
     'x': array([138]),
     'xaxis': 'x',
     'y': array(['The Singles'], dtype=object),
     'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovertemplate': 'Name=The Cult<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
     'legendgroup': 'The Cult',
     'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
     'name': 'The Cult',
     'offsetgroup': 'The Cult',
     'orientation': 'h',
     'showlegend': True,
     'textposition': 'auto',
     'type': 'bar',
     'x': array([139, 139]),
     'xaxis': 'x',
     'y': array(['Beyond Good And Evil',
                  'Pure Cult: The Best Of The Cult (For Rockers, Ravers, Lovers & Sinners) [U
K]'],
                  dtype=object),
     'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovertemplate': 'Name=The Doors<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
     'legendgroup': 'The Doors',
     'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
     'name': 'The Doors',
     'offsetgroup': 'The Doors',
     'orientation': 'h',
     'showlegend': True,

```

```

'textposition': 'auto',
'type': 'bar',
'x': array([140]),
'xaxis': 'x',
'y': array(['The Doors'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=The Police<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'The Police',
'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
'name': 'The Police',
'offsetgroup': 'The Police',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([141]),
'xaxis': 'x',
'y': array(['The Police Greatest Hits'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=The Rolling Stones<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'The Rolling Stones',
'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
'name': 'The Rolling Stones',
'offsetgroup': 'The Rolling Stones',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([142, 142, 142]),
'xaxis': 'x',
'y': array(['Hot Rocks, 1964-1971 (Disc 1)', 'No Security', 'Voodoo Lounge'],
dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=The Tea Party<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'The Tea Party',
'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
'name': 'The Tea Party',
'offsetgroup': 'The Tea Party',
'orientation': 'h',

```

```

'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([143, 143]),
'xaxis': 'x',
'y': array(['Tangents', 'Transmission'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=The Who<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'The Who',
'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
'name': 'The Who',
'offsetgroup': 'The Who',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([144]),
'xaxis': 'x',
'y': array(['My Generation - The Very Best Of The Who'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Tim Maia<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Tim Maia',
'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
'name': 'Tim Maia',
'offsetgroup': 'Tim Maia',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([145, 145]),
'xaxis': 'x',
'y': array(['Serie Sem Limite (Disc 1)', 'Serie Sem Limite (Disc 2)'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Titãs<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Titãs',
'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
'name': 'Titãs',
'offsetgroup': 'Titãs',
'orientation': 'h',

```

```

'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([146, 146]),
'xaxis': 'x',
'y': array(['Acústico', 'Volume Dois'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Battlestar Galactica<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Battlestar Galactica',
'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
'name': 'Battlestar Galactica',
'offsetgroup': 'Battlestar Galactica',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([147, 147]),
'xaxis': 'x',
'y': array(['Battlestar Galactica: The Story So Far',
            'Battlestar Galactica, Season 3'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Heroes<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Heroes',
'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
'name': 'Heroes',
'offsetgroup': 'Heroes',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([148]),
'xaxis': 'x',
'y': array(['Heroes, Season 1'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Lost<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Lost',
'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
'name': 'Lost',
'offsetgroup': 'Lost',

```

```

'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([149, 149, 149, 149]),
'xaxis': 'x',
'y': array(['Lost, Season 3', 'Lost, Season 1', 'Lost, Season 2', 'LOST, Season 4'],
          dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=U2<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'U2',
'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
'name': 'U2',
'offsetgroup': 'U2',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([150, 150, 150, 150, 150, 150, 150, 150, 150, 150]),
'xaxis': 'x',
'y': array(['Achtung Baby', 'All That You Can't Leave Behind', 'B-Sides 1980-1990',
          'How To Dismantle An Atomic Bomb', 'Pop', 'Rattle And Hum',
          'The Best Of 1980-1990', 'War', 'Zooropa',
          'Instant Karma: The Amnesty International Campaign to Save Darfur'],
          dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=UB40<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'UB40',
'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
'name': 'UB40',
'offsetgroup': 'UB40',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([151]),
'xaxis': 'x',
'y': array(['UB40 The Best Of - Volume Two [UK]'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',

```



```

'hovertemplate': 'Name=Van Halen<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Van Halen',
'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
'name': 'Van Halen',
'offsetgroup': 'Van Halen',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([152, 152, 152, 152]),
'xaxis': 'x',
'y': array(['Diver Down', 'The Best Of Van Halen, Vol. I', 'Van Halen',
            'Van Halen III'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Velvet Revolver<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Velvet Revolver',
'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
'name': 'Velvet Revolver',
'offsetgroup': 'Velvet Revolver',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([153]),
'xaxis': 'x',
'y': array(['Contraband'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Vinícius De Moraes<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Vinícius De Moraes',
'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
'name': 'Vinícius De Moraes',
'offsetgroup': 'Vinícius De Moraes',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([72]),
'xaxis': 'x',
'y': array(['Vinicius De Moraes'], dtype=object),
'yaxis': 'y'},

```

```

{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Zeca Pagodinho<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Zeca Pagodinho',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'Zeca Pagodinho',
 'offsetgroup': 'Zeca Pagodinho',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([155]),
 'xaxis': 'x',
 'y': array(['Ao Vivo [IMPORT]'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=The Office<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'The Office',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'The Office',
 'offsetgroup': 'The Office',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([156, 156, 156]),
 'xaxis': 'x',
 'y': array(['The Office, Season 1', 'The Office, Season 2', 'The Office, Season 3'],
            dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Dread Zeppelin<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Dread Zeppelin',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'Dread Zeppelin',
 'offsetgroup': 'Dread Zeppelin',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([157]),
 'xaxis': 'x',
 'y': array(['Un-Led-Ed'], dtype=object),

```

```

'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Battlestar Galactica (Cla' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Battlestar Galactica (Classic)',
'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
'name': 'Battlestar Galactica (Classic)',
'offsetgroup': 'Battlestar Galactica (Classic)',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([158]),
'xaxis': 'x',
'y': array(['Battlestar Galactica (Classic), Season 1'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Aquaman<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Aquaman',
'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
'name': 'Aquaman',
'offsetgroup': 'Aquaman',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([159]),
'xaxis': 'x',
'y': array(['Aquaman'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Scorpions<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Scorpions',
'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
'name': 'Scorpions',
'offsetgroup': 'Scorpions',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([179]),
'xaxis': 'x',
'y': array(['20th Century Masters - The Millennium Collection: The Best of Scorpions'],

```

```

        dtype=object),
    'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=House Of Pain<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'House Of Pain',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'House Of Pain',
 'offsetgroup': 'House Of Pain',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([180]),
 'xaxis': 'x',
 'y': array(['House of Pain'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=0 Rappa<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': '0 Rappa',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': '0 Rappa',
 'offsetgroup': '0 Rappa',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([36]),
 'xaxis': 'x',
 'y': array(['Radio Brasil (0 Som da Jovem Vanguarda) - Seleccao de Henrique Amaro'],
           dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Cake<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Cake',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'Cake',
 'offsetgroup': 'Cake',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([196]),

```

```

    'xaxis': 'x',
    'y': array(['Cake: B-Sides and Rarities'], dtype=object),
    'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovertemplate': 'Name=Aisha Duo<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
     'legendgroup': 'Aisha Duo',
     'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
     'name': 'Aisha Duo',
     'offsetgroup': 'Aisha Duo',
     'orientation': 'h',
     'showlegend': True,
     'textposition': 'auto',
     'type': 'bar',
     'x': array([197]),
     'xaxis': 'x',
     'y': array(['Quiet Songs'], dtype=object),
     'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovertemplate': 'Name=Habib Koité and Bamada<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>
>',
     'legendgroup': 'Habib Koité and Bamada',
     'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
     'name': 'Habib Koité and Bamada',
     'offsetgroup': 'Habib Koité and Bamada',
     'orientation': 'h',
     'showlegend': True,
     'textposition': 'auto',
     'type': 'bar',
     'x': array([198]),
     'xaxis': 'x',
     'y': array(['Muso Ko'], dtype=object),
     'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovertemplate': 'Name=Karsh Kale<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
     'legendgroup': 'Karsh Kale',
     'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
     'name': 'Karsh Kale',
     'offsetgroup': 'Karsh Kale',
     'orientation': 'h',
     'showlegend': True,
     'textposition': 'auto',
     'type': 'bar',

```

```

        'x': array([199]),
        'xaxis': 'x',
        'y': array(['Realize'], dtype=object),
        'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovertemplate': 'Name=The Posies<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
     'legendgroup': 'The Posies',
     'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
     'name': 'The Posies',
     'offsetgroup': 'The Posies',
     'orientation': 'h',
     'showlegend': True,
     'textposition': 'auto',
     'type': 'bar',
     'x': array([200]),
     'xaxis': 'x',
     'y': array(['Every Kind of Light'], dtype=object),
     'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovertemplate': 'Name=Luciana Souza/Romero Lubambo<br>ArtistId=%{x}<br>Title=%{y}<extra></
extra>',
     'legendgroup': 'Luciana Souza/Romero Lubambo',
     'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
     'name': 'Luciana Souza/Romero Lubambo',
     'offsetgroup': 'Luciana Souza/Romero Lubambo',
     'orientation': 'h',
     'showlegend': True,
     'textposition': 'auto',
     'type': 'bar',
     'x': array([201]),
     'xaxis': 'x',
     'y': array(['Duos II'], dtype=object),
     'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovertemplate': 'Name=Aaron Goldberg<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
     'legendgroup': 'Aaron Goldberg',
     'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
     'name': 'Aaron Goldberg',
     'offsetgroup': 'Aaron Goldberg',
     'orientation': 'h',
     'showlegend': True,
     'textposition': 'auto',

```

```

        'type': 'bar',
        'x': array([202]),
        'xaxis': 'x',
        'y': array(['Worlds'], dtype=object),
        'yaxis': 'y'},
        {'alignmentgroup': 'True',
        'hovernplate': 'Name=Nicolaus Esterhazy Sinfonia<br>ArtistId=%{x}<br>Title=%{y}<extra></e
xtra>',
        'legendgroup': 'Nicolaus Esterhazy Sinfonia',
        'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
        'name': 'Nicolaus Esterhazy Sinfonia',
        'offsetgroup': 'Nicolaus Esterhazy Sinfonia',
        'orientation': 'h',
        'showlegend': True,
        'textposition': 'auto',
        'type': 'bar',
        'x': array([203]),
        'xaxis': 'x',
        'y': array(['The Best of Beethoven'], dtype=object),
        'yaxis': 'y'},
        {'alignmentgroup': 'True',
        'hovernplate': 'Name=Temple of the Dog<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
        'legendgroup': 'Temple of the Dog',
        'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
        'name': 'Temple of the Dog',
        'offsetgroup': 'Temple of the Dog',
        'orientation': 'h',
        'showlegend': True,
        'textposition': 'auto',
        'type': 'bar',
        'x': array([204]),
        'xaxis': 'x',
        'y': array(['Temple of the Dog'], dtype=object),
        'yaxis': 'y'},
        {'alignmentgroup': 'True',
        'hovernplate': 'Name=Chris Cornell<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
        'legendgroup': 'Chris Cornell',
        'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
        'name': 'Chris Cornell',
        'offsetgroup': 'Chris Cornell',
        'orientation': 'h',
        'showlegend': True,

```

```

'textposition': 'auto',
'type': 'bar',
'x': array([205]),
'xaxis': 'x',
'y': array(['Carry On'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Alberto Turco & Nova Scho' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Alberto Turco & Nova Schola Gregoriana',
'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
'name': 'Alberto Turco & Nova Schola Gregoriana',
'offsetgroup': 'Alberto Turco & Nova Schola Gregoriana',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([206]),
'xaxis': 'x',
'y': array(['Adorate Deum: Gregorian Chant from the Proper of the Mass'],
dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Richard Marlow & The Choi' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Richard Marlow & The Choir of Trinity College, Cambridge',
'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
'name': 'Richard Marlow & The Choir of Trinity College, Cambridge',
'offsetgroup': 'Richard Marlow & The Choir of Trinity College, Cambridge',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([207]),
'xaxis': 'x',
'y': array(['Alleghi: Miserere'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=English Concert & Trevor ' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'English Concert & Trevor Pinnock',
'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
'name': 'English Concert & Trevor Pinnock',
'offsetgroup': 'English Concert & Trevor Pinnock',
'orientation': 'h',

```



```

'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([208, 208]),
'xaxis': 'x',
'y': array(['Pachelbel: Canon & Gigue',
            'Handel: Music for the Royal Fireworks (Original Version 1749)'],
            dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovernplate': ('Name=Anne-Sophie Mutter, Herbe' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Anne-Sophie Mutter, Herbert Von Karajan & Wiener Philharmoniker',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'Anne-Sophie Mutter, Herbert Von Karajan & Wiener Philharmoniker',
 'offsetgroup': 'Anne-Sophie Mutter, Herbert Von Karajan & Wiener Philharmoniker',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([209]),
 'xaxis': 'x',
 'y': array(['Vivaldi: The Four Seasons'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovernplate': ('Name=Hilary Hahn, Jeffrey Kaha' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Hilary Hahn, Jeffrey Kahane, Los Angeles Chamber Orchestra & Margaret Batje
r',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'Hilary Hahn, Jeffrey Kahane, Los Angeles Chamber Orchestra & Margaret Batjer',
 'offsetgroup': 'Hilary Hahn, Jeffrey Kahane, Los Angeles Chamber Orchestra & Margaret Batje
r',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([210]),
 'xaxis': 'x',
 'y': array(['Bach: Violin Concertos'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovernplate': 'Name=Wilhelm Kempff<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Wilhelm Kempff',

```

```

'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
'name': 'Wilhelm Kempff',
'offsetgroup': 'Wilhelm Kempff',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([211]),
'xaxis': 'x',
'y': array(['Bach: Goldberg Variations'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Yo-Yo Ma<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Yo-Yo Ma',
'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
'name': 'Yo-Yo Ma',
'offsetgroup': 'Yo-Yo Ma',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([212]),
'xaxis': 'x',
'y': array(['Bach: The Cello Suites'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Scholars Baroque Ensemble<br>ArtistId=%{x}<br>Title=%{y}<extra></ext
ra>',
'legendgroup': 'Scholars Baroque Ensemble',
'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
'name': 'Scholars Baroque Ensemble',
'offsetgroup': 'Scholars Baroque Ensemble',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([213]),
'xaxis': 'x',
'y': array(['Handel: The Messiah (Highlights)'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Academy of St. Martin in ' ... ')<br>Title=%{y}<extra></extra>'),

```

```

'legendgroup': 'Academy of St. Martin in the Fields & Sir Neville Marriner',
'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
'name': 'Academy of St. Martin in the Fields & Sir Neville Marriner',
'offsetgroup': 'Academy of St. Martin in the Fields & Sir Neville Marriner',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([214]),
'xaxis': 'x',
'y': array(['The World of Classical Favourites'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Academy of St. Martin in ' ... ')<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Academy of St. Martin in the Fields Chamber Ensemble & Sir Neville Marrine
r',

'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
'name': 'Academy of St. Martin in the Fields Chamber Ensemble & Sir Neville Marriner',
'offsetgroup': 'Academy of St. Martin in the Fields Chamber Ensemble & Sir Neville Marrine
r',

'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([215]),
'xaxis': 'x',
'y': array(['Sir Neville Marriner: A Celebration'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Berliner Philharmoniker, ' ... ')<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Berliner Philharmoniker, Claudio Abbado & Sabine Meyer',
'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
'name': 'Berliner Philharmoniker, Claudio Abbado & Sabine Meyer',
'offsetgroup': 'Berliner Philharmoniker, Claudio Abbado & Sabine Meyer',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([216]),
'xaxis': 'x',
'y': array(['Mozart: Wind Concertos'], dtype=object),
'yaxis': 'y'},

```

```

{'alignmentgroup': 'True',
 'hovertemplate': ('Name=Royal Philharmonic Orches' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Royal Philharmonic Orchestra & Sir Thomas Beecham',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'Royal Philharmonic Orchestra & Sir Thomas Beecham',
 'offsetgroup': 'Royal Philharmonic Orchestra & Sir Thomas Beecham',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([217]),
 'xaxis': 'x',
 'y': array(['Haydn: Symphonies 99 - 104'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('Name=Orchestre Révolutionnaire' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Orchestre Révolutionnaire et Romantique & John Eliot Gardiner',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'Orchestre Révolutionnaire et Romantique & John Eliot Gardiner',
 'offsetgroup': 'Orchestre Révolutionnaire et Romantique & John Eliot Gardiner',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([218]),
 'xaxis': 'x',
 'y': array(['Beethoven: Symhonies Nos. 5 & 6'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('Name=Britten Sinfonia, Ivor Bo' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Britten Sinfonia, Ivor Bolton & Lesley Garrett',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'Britten Sinfonia, Ivor Bolton & Lesley Garrett',
 'offsetgroup': 'Britten Sinfonia, Ivor Bolton & Lesley Garrett',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([219]),
 'xaxis': 'x',
 'y': array(['A Soprano Inspired'], dtype=object),
 'yaxis': 'y'},

```

```

{'alignmentgroup': 'True',
 'hovertemplate': ('Name=Chicago Symphony Chorus, ' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Chicago Symphony Chorus, Chicago Symphony Orchestra & Sir Georg Solti',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'Chicago Symphony Chorus, Chicago Symphony Orchestra & Sir Georg Solti',
 'offsetgroup': 'Chicago Symphony Chorus, Chicago Symphony Orchestra & Sir Georg Solti',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([220]),
 'xaxis': 'x',
 'y': array(['Great Opera Choruses'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('Name=Sir Georg Solti & Wiener ' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Sir Georg Solti & Wiener Philharmoniker',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': 'Sir Georg Solti & Wiener Philharmoniker',
 'offsetgroup': 'Sir Georg Solti & Wiener Philharmoniker',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([221]),
 'xaxis': 'x',
 'y': array(['Wagner: Favourite Overtures'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('Name=Academy of St. Martin in ' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': ('Academy of St. Martin in the F' ... 'ville Marriner & Sylvia McNair'),
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': ('Academy of St. Martin in the F' ... 'ville Marriner & Sylvia McNair'),
 'offsetgroup': ('Academy of St. Martin in the F' ... 'ville Marriner & Sylvia McNair'),
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([222]),
 'xaxis': 'x',
 'y': array(['Fauré: Requiem, Ravel: Pavane & Others'], dtype=object),
 'yaxis': 'y'},

```

```

{'alignmentgroup': 'True',
 'hovertemplate': ('Name=London Symphony Orchestra' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'London Symphony Orchestra & Sir Charles Mackerras',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'London Symphony Orchestra & Sir Charles Mackerras',
 'offsetgroup': 'London Symphony Orchestra & Sir Charles Mackerras',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([223]),
 'xaxis': 'x',
 'y': array(['Tchaikovsky: The Nutcracker'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('Name=Barry Wordsworth & BBC Co' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Barry Wordsworth & BBC Concert Orchestra',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Barry Wordsworth & BBC Concert Orchestra',
 'offsetgroup': 'Barry Wordsworth & BBC Concert Orchestra',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([224]),
 'xaxis': 'x',
 'y': array(['The Last Night of the Proms'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('Name=Herbert Von Karajan, Mire' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Herbert Von Karajan, Mirella Freni & Wiener Philharmoniker',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'Herbert Von Karajan, Mirella Freni & Wiener Philharmoniker',
 'offsetgroup': 'Herbert Von Karajan, Mirella Freni & Wiener Philharmoniker',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([225]),
 'xaxis': 'x',
 'y': array(['Puccini: Madama Butterfly - Highlights'], dtype=object),
 'yaxis': 'y'},

```

```

{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Eugene Ormandy<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Eugene Ormandy',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'Eugene Ormandy',
 'offsetgroup': 'Eugene Ormandy',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([226, 226, 226]),
 'xaxis': 'x',
 'y': array(['Holst: The Planets, Op. 32 & Vaughan Williams: Fantasies',
            'Strauss: Waltzes', 'Respighi:Pines of Rome'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Luciano Pavarotti<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Luciano Pavarotti',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'Luciano Pavarotti',
 'offsetgroup': 'Luciano Pavarotti',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([227]),
 'xaxis': 'x',
 'y': array(["Pavarotti's Opera Made Easy"], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('Name=Leonard Bernstein & New Y' ... ')<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Leonard Bernstein & New York Philharmonic',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'Leonard Bernstein & New York Philharmonic',
 'offsetgroup': 'Leonard Bernstein & New York Philharmonic',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([228]),
 'xaxis': 'x',
 'y': array(["Great Performances - Barber's Adagio and Other Romantic Favorites for String

```

s"],

```

dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Boston Symphony Orchestra' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Boston Symphony Orchestra & Seiji Ozawa',
'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
'name': 'Boston Symphony Orchestra & Seiji Ozawa',
'offsetgroup': 'Boston Symphony Orchestra & Seiji Ozawa',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([229]),
'xaxis': 'x',
'y': array(['Carmina Burana'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Aaron Copland & London Sy' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Aaron Copland & London Symphony Orchestra',
'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
'name': 'Aaron Copland & London Symphony Orchestra',
'offsetgroup': 'Aaron Copland & London Symphony Orchestra',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([230]),
'xaxis': 'x',
'y': array(['A Copland Celebration, Vol. I'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Ton Koopman<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Ton Koopman',
'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
'name': 'Ton Koopman',
'offsetgroup': 'Ton Koopman',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([231]),

```



```

'xaxis': 'x',
'y': array(['Bach: Toccata & Fugue in D Minor'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Sergei Prokofiev & Yuri T' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Sergei Prokofiev & Yuri Temirkanov',
'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
'name': 'Sergei Prokofiev & Yuri Temirkanov',
'offsetgroup': 'Sergei Prokofiev & Yuri Temirkanov',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([232]),
'xaxis': 'x',
'y': array(['Prokofiev: Symphony No.1'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Chicago Symphony Orchestr' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Chicago Symphony Orchestra & Fritz Reiner',
'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
'name': 'Chicago Symphony Orchestra & Fritz Reiner',
'offsetgroup': 'Chicago Symphony Orchestra & Fritz Reiner',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([233]),
'xaxis': 'x',
'y': array(['Scheherazade'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Orchestra of The Age of E' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Orchestra of The Age of Enlightenment',
'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
'name': 'Orchestra of The Age of Enlightenment',
'offsetgroup': 'Orchestra of The Age of Enlightenment',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([234]),

```

```

'xaxis': 'x',
'y': array(['Bach: The Brandenburg Concertos'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Emanuel Ax, Eugene Ormand' ... ')<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Emanuel Ax, Eugene Ormandy & Philadelphia Orchestra',
'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
'name': 'Emanuel Ax, Eugene Ormandy & Philadelphia Orchestra',
'offsetgroup': 'Emanuel Ax, Eugene Ormandy & Philadelphia Orchestra',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([235]),
'xaxis': 'x',
'y': array(['Chopin: Piano Concertos Nos. 1 & 2'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=James Levine<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'James Levine',
'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
'name': 'James Levine',
'offsetgroup': 'James Levine',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([236]),
'xaxis': 'x',
'y': array(['Mascagni: Cavalleria Rusticana'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Berliner Philharmoniker &' ... ')<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Berliner Philharmoniker & Hans Rosbaud',
'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
'name': 'Berliner Philharmoniker & Hans Rosbaud',
'offsetgroup': 'Berliner Philharmoniker & Hans Rosbaud',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([237]),

```

```

'xaxis': 'x',
'y': array(['Sibelius: Finlandia'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Maurizio Pollini<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Maurizio Pollini',
'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
'name': 'Maurizio Pollini',
'offsetgroup': 'Maurizio Pollini',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([238]),
'xaxis': 'x',
'y': array(['Beethoven Piano Sonatas: Moonlight & Pastorale'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Gustav Mahler<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Gustav Mahler',
'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
'name': 'Gustav Mahler',
'offsetgroup': 'Gustav Mahler',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([240]),
'xaxis': 'x',
'y': array(['Great Recordings of the Century - Mahler: Das Lied von der Erde'],
dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Felix Schmidt, London Sym' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Felix Schmidt, London Symphony Orchestra & Rafael Frühbeck de Burgos',
'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
'name': 'Felix Schmidt, London Symphony Orchestra & Rafael Frühbeck de Burgos',
'offsetgroup': 'Felix Schmidt, London Symphony Orchestra & Rafael Frühbeck de Burgos',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',

```

```

        'x': array([241]),
        'xaxis': 'x',
        'y': array(['Elgar: Cello Concerto & Vaughan Williams: Fantasias'], dtype=object),
        'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovernplate': ('Name=Edo de Waart & San Franci' ... '}<br>Title=%{y}<extra></extra>'),
     'legendgroup': 'Edo de Waart & San Francisco Symphony',
     'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
     'name': 'Edo de Waart & San Francisco Symphony',
     'offsetgroup': 'Edo de Waart & San Francisco Symphony',
     'orientation': 'h',
     'showlegend': True,
     'textposition': 'auto',
     'type': 'bar',
     'x': array([242]),
     'xaxis': 'x',
     'y': array(['Adams, John: The Chairman Dances'], dtype=object),
     'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovernplate': ('Name=Antal Doráti & London Sym' ... '}<br>Title=%{y}<extra></extra>'),
     'legendgroup': 'Antal Doráti & London Symphony Orchestra',
     'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
     'name': 'Antal Doráti & London Symphony Orchestra',
     'offsetgroup': 'Antal Doráti & London Symphony Orchestra',
     'orientation': 'h',
     'showlegend': True,
     'textposition': 'auto',
     'type': 'bar',
     'x': array([243]),
     'xaxis': 'x',
     'y': array(['Tchaikovsky: 1812 Festival Overture, Op.49, Capriccio Italien & Beethoven: Wel
lington's Victory'],
                dtype=object),
     'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovernplate': ('Name=Choir Of Westminster Abbe' ... '}<br>Title=%{y}<extra></extra>'),
     'legendgroup': 'Choir Of Westminster Abbey & Simon Preston',
     'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
     'name': 'Choir Of Westminster Abbey & Simon Preston',
     'offsetgroup': 'Choir Of Westminster Abbey & Simon Preston',
     'orientation': 'h',
     'showlegend': True,

```

```

'textposition': 'auto',
'type': 'bar',
'x': array([244]),
'xaxis': 'x',
'y': array(['Palestrina: Missa Papae Marcelli & Allegri: Miserere'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Michael Tilson Thomas & S' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Michael Tilson Thomas & San Francisco Symphony',
'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
'name': 'Michael Tilson Thomas & San Francisco Symphony',
'offsetgroup': 'Michael Tilson Thomas & San Francisco Symphony',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([245, 245]),
'xaxis': 'x',
'y': array(['Prokofiev: Romeo & Juliet', 'Berlioz: Symphonie Fantastique'],
          dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Chor der Wiener Staatsope' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Chor der Wiener Staatsoper, Herbert Von Karajan & Wiener Philharmoniker',
'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
'name': 'Chor der Wiener Staatsoper, Herbert Von Karajan & Wiener Philharmoniker',
'offsetgroup': 'Chor der Wiener Staatsoper, Herbert Von Karajan & Wiener Philharmoniker',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([246]),
'xaxis': 'x',
'y': array(['Bizet: Carmen Highlights'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': "Name=The King's Singers<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>",
'legendgroup': "The King's Singers",
'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
'name': "The King's Singers",
'offsetgroup': "The King's Singers",
'orientation': 'h',

```

```

'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([247]),
'xaxis': 'x',
'y': array(['English Renaissance'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Berliner Philharmoniker & ' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Berliner Philharmoniker & Herbert Von Karajan',
'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
'name': 'Berliner Philharmoniker & Herbert Von Karajan',
'offsetgroup': 'Berliner Philharmoniker & Herbert Von Karajan',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([248, 248, 248]),
'xaxis': 'x',
'y': array(['Grieg: Peer Gynt Suites & Sibelius: Pell  as et M  lisande',
'Mozart: Symphonies Nos. 40 & 41',
'Prokofiev: Symphony No.5 & Stravinsky: Le Sacre Du Printemps'],
dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Sir Georg Solti, Sumi Jo ' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Sir Georg Solti, Sumi Jo & Wiener Philharmoniker',
'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
'name': 'Sir Georg Solti, Sumi Jo & Wiener Philharmoniker',
'offsetgroup': 'Sir Georg Solti, Sumi Jo & Wiener Philharmoniker',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([249]),
'xaxis': 'x',
'y': array(['Mozart Gala: Famous Arias'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': "Name=Christopher O'Riley<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>",
'legendgroup': "Christopher O'Riley",
'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},

```

```

'name': "Christopher O'Riley",
'offsetgroup': "Christopher O'Riley",
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([250]),
'xaxis': 'x',
'y': array(['SCRIABIN: Vers la flamme'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Fretwork<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Fretwork',
'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
'name': 'Fretwork',
'offsetgroup': 'Fretwork',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([251]),
'xaxis': 'x',
'y': array(['Armada: Music from the Courts of England and Spain'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Amy Winehouse<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Amy Winehouse',
'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
'name': 'Amy Winehouse',
'offsetgroup': 'Amy Winehouse',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([252, 252]),
'xaxis': 'x',
'y': array(['Back to Black', 'Frank'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Callexico<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Callexico',
'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},

```

```

'name': 'Calexico',
'offsetgroup': 'Calexico',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([253]),
'xaxis': 'x',
'y': array(['Carried to Dust (Bonus Track Version)'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Otto Klemperer & Philharm' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Otto Klemperer & Philharmonia Orchestra',
'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
'name': 'Otto Klemperer & Philharmonia Orchestra',
'offsetgroup': 'Otto Klemperer & Philharmonia Orchestra',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([254]),
'xaxis': 'x',
'y': array(["Beethoven: Symphony No. 6 'Pastoral' Etc."], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Yehudi Menuhin<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Yehudi Menuhin',
'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
'name': 'Yehudi Menuhin',
'offsetgroup': 'Yehudi Menuhin',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([255]),
'xaxis': 'x',
'y': array(['Bartok: Violin & Viola Concertos'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Philharmonia Orchestra & ' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Philharmonia Orchestra & Sir Neville Marriner',
'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},

```



```

'name': 'Philharmonia Orchestra & Sir Neville Marriner',
'offsetgroup': 'Philharmonia Orchestra & Sir Neville Marriner',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([256]),
'xaxis': 'x',
'y': array(["Mendelssohn: A Midsummer Night's Dream"], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Academy of St. Martin in ' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Academy of St. Martin in the Fields, Sir Neville Marriner & Thurston Dart',
'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
'name': 'Academy of St. Martin in the Fields, Sir Neville Marriner & Thurston Dart',
'offsetgroup': 'Academy of St. Martin in the Fields, Sir Neville Marriner & Thurston Dart',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([257]),
'xaxis': 'x',
'y': array(['Bach: Orchestral Suites Nos. 1 - 4'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Les Arts Florissants & Wi' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Les Arts Florissants & William Christie',
'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
'name': 'Les Arts Florissants & William Christie',
'offsetgroup': 'Les Arts Florissants & William Christie',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([258]),
'xaxis': 'x',
'y': array(['Charpentier: Divertissements, Airs & Concerts'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=The 12 Cellists of The Be' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'The 12 Cellists of The Berlin Philharmonic',
'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},

```

```

'name': 'The 12 Cellists of The Berlin Philharmonic',
'offsetgroup': 'The 12 Cellists of The Berlin Philharmonic',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([259]),
'xaxis': 'x',
'y': array(['South American Getaway'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Adrian Leaper & Doreen de' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Adrian Leaper & Doreen de Feis',
'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
'name': 'Adrian Leaper & Doreen de Feis',
'offsetgroup': 'Adrian Leaper & Doreen de Feis',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([260]),
'xaxis': 'x',
'y': array(['Górecki: Symphony No. 3'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Roger Norrington, London ' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Roger Norrington, London Classical Players',
'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
'name': 'Roger Norrington, London Classical Players',
'offsetgroup': 'Roger Norrington, London Classical Players',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([261]),
'xaxis': 'x',
'y': array(['Purcell: The Fairy Queen'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ("Name=Charles Dutoit & L'Orches" ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': "Charles Dutoit & L'Orchestre Symphonique de Montréal",
'marker': {'color': '#636efa', 'pattern': {'shape': ''}},

```

```

'name': "Charles Dutoit & L'Orchestre Symphonique de Montréal",
'offsetgroup': "Charles Dutoit & L'Orchestre Symphonique de Montréal",
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([262]),
'xaxis': 'x',
'y': array(['The Ultimate Relaxation Album'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Equale Brass Ensemble, Jo' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': ('Equale Brass Ensemble, John El' ... 'Monteverdi Orchestra and Choir'),
'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
'name': ('Equale Brass Ensemble, John El' ... 'Monteverdi Orchestra and Choir'),
'offsetgroup': ('Equale Brass Ensemble, John El' ... 'Monteverdi Orchestra and Choir'),
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([263]),
'xaxis': 'x',
'y': array(['Purcell: Music for the Queen Mary'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=Kent Nagano and Orchestre' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': "Kent Nagano and Orchestre de l'Opéra de Lyon",
'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
'name': "Kent Nagano and Orchestre de l'Opéra de Lyon",
'offsetgroup': "Kent Nagano and Orchestre de l'Opéra de Lyon",
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([264]),
'xaxis': 'x',
'y': array(['Weill: The Seven Deadly Sins'], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Julian Bream<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Julian Bream',
'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},

```

```

        'name': 'Julian Bream',
        'offsetgroup': 'Julian Bream',
        'orientation': 'h',
        'showlegend': True,
        'textposition': 'auto',
        'type': 'bar',
        'x': array([265]),
        'xaxis': 'x',
        'y': array(['J.S. Bach: Chaconne, Suite in E Minor, Partita in E Major & Prelude, Fugue and
Allegro'],
                dtype=object),
        'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovernplate': 'Name=Martin Roscoe<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
     'legendgroup': 'Martin Roscoe',
     'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
     'name': 'Martin Roscoe',
     'offsetgroup': 'Martin Roscoe',
     'orientation': 'h',
     'showlegend': True,
     'textposition': 'auto',
     'type': 'bar',
     'x': array([266]),
     'xaxis': 'x',
     'y': array(['Szymanowski: Piano Works, Vol. 1'], dtype=object),
     'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovernplate': ('Name=Göteborgs Symfoniker & Ne' ... ')<br>Title=%{y}<extra></extra>'),
     'legendgroup': 'Göteborgs Symfoniker & Neeme Järvi',
     'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
     'name': 'Göteborgs Symfoniker & Neeme Järvi',
     'offsetgroup': 'Göteborgs Symfoniker & Neeme Järvi',
     'orientation': 'h',
     'showlegend': True,
     'textposition': 'auto',
     'type': 'bar',
     'x': array([267]),
     'xaxis': 'x',
     'y': array(['Nielsen: The Six Symphonies'], dtype=object),
     'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovernplate': 'Name=Itzhak Perlman<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',

```

```

'legendgroup': 'Itzhak Perlman',
'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
'name': 'Itzhak Perlman',
'offsetgroup': 'Itzhak Perlman',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([268]),
'xaxis': 'x',
'y': array(["Great Recordings of the Century: Paganini's 24 Caprices"], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Michele Campanella<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Michele Campanella',
'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
'name': 'Michele Campanella',
'offsetgroup': 'Michele Campanella',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([269]),
'xaxis': 'x',
'y': array(["Liszt - 12 Études D'Execution Transcendante"], dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Gerald Moore<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
'legendgroup': 'Gerald Moore',
'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
'name': 'Gerald Moore',
'offsetgroup': 'Gerald Moore',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([270]),
'xaxis': 'x',
'y': array(['Great Recordings of the Century - Shubert: Schwanengesang, 4 Lieder'],
dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',

```

```

'hovertemplate': ('Name=Mela Tenenbaum, Pro Music' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': 'Mela Tenenbaum, Pro Musica Prague & Richard Kapp',
'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
'name': 'Mela Tenenbaum, Pro Musica Prague & Richard Kapp',
'offsetgroup': 'Mela Tenenbaum, Pro Musica Prague & Richard Kapp',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([271]),
'xaxis': 'x',
'y': array(['Locatelli: Concertos for Violin, Strings and Continuo, Vol. 3'],
          dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'Name=Emerson String Quartet<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>
>',
'legendgroup': 'Emerson String Quartet',
'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
'name': 'Emerson String Quartet',
'offsetgroup': 'Emerson String Quartet',
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([272]),
'xaxis': 'x',
'y': array(["Schubert: The Late String Quartets & String Quintet (3 CD's)"],
          dtype=object),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': ('Name=C. Monteverdi, Nigel Roge' ... '}<br>Title=%{y}<extra></extra>'),
'legendgroup': ('C. Monteverdi, Nigel Rogers - ' ... 'roque; London Cornett & Sackbu'),
'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
'name': ('C. Monteverdi, Nigel Rogers - ' ... 'roque; London Cornett & Sackbu'),
'offsetgroup': ('C. Monteverdi, Nigel Rogers - ' ... 'roque; London Cornett & Sackbu'),
'orientation': 'h',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array([273]),
'xaxis': 'x',

```

```

        'y': array(["Monteverdi: L'Orfeo"], dtype=object),
        'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovertemplate': 'Name=Nash Ensemble<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
     'legendgroup': 'Nash Ensemble',
     'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
     'name': 'Nash Ensemble',
     'offsetgroup': 'Nash Ensemble',
     'orientation': 'h',
     'showlegend': True,
     'textposition': 'auto',
     'type': 'bar',
     'x': array([274]),
     'xaxis': 'x',
     'y': array(['Mozart: Chamber Music'], dtype=object),
     'yaxis': 'y'},
    {'alignmentgroup': 'True',
     'hovertemplate': 'Name=Philip Glass Ensemble<br>ArtistId=%{x}<br>Title=%{y}<extra></extra>',
     'legendgroup': 'Philip Glass Ensemble',
     'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
     'name': 'Philip Glass Ensemble',
     'offsetgroup': 'Philip Glass Ensemble',
     'orientation': 'h',
     'showlegend': True,
     'textposition': 'auto',
     'type': 'bar',
     'x': array([275]),
     'xaxis': 'x',
     'y': array(['Koyaanisqatsi (Soundtrack from the Motion Picture)'], dtype=object),
     'yaxis': 'y'}],
    'layout': {'barmode': 'relative',
               'legend': {'title': {'text': 'Name'}, 'tracegroupgap': 0},
               'margin': {'t': 60},
               'template': '...',
               'title': {'text': 'Albums by Artist'},
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'ArtistId'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Title'}}
    })

```

```

In [12]: question = """
         Find all tracks with a name containing "What" (case-insensitive)

```

```
"""
```

```
vn.ask(question=question)
```

```
Number of requested results 10 is greater than number of elements in index 3, updating n_results = 3  
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```



```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX [IFK_TrackGenreId] ON [Track] ([GenreId])\n\nCREATE INDEX [IFK_TrackAlbumId] ON [Track] ([AlbumId])\n\nCREATE INDEX [IFK_PlaylistTrackTrackId] ON [PlaylistTrack] ([TrackId])\n\nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(20),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_TrackMediaTypeId] ON [Track] ([MediaTypeId])\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE INDEX [IFK_AlbumArtistId] ON [Album] ([ArtistId])\n\nCREATE TABLE [PlaylistTrack]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    CONSTRAINT [PK_PlaylistTrack] PRIMARY KEY ([PlaylistId], [TrackId]),\n    FOREIGN KEY ([PlaylistId]) REFERENCES [Playlist] ([PlaylistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Album]\n(\n    [AlbumId] INTEGER NOT NULL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER NOT NULL,\n    CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Playlist]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Playlist] PRIMARY KEY ([PlaylistId])\n)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n\n===Response Guidelines \n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n\n3. If the provided context is insufficient, please explain why it can't be generated. \n\n4. Please use the most relevant table(s). \n\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n\n"}], {'role': 'user', 'content': ' \n    List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJOIN Artist ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': " SELECT * FROM t_person WHERE name = 'John Doe';"}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n    Find all tracks with a name containing "What" (case-insensitive)\n'}]
```

Ollama parameters:

```
model=llama3:latest,
options={},
keep_alive=None
```

Prompt Content:

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo
```

```

rmat instructions. \n===Tables \nCREATE INDEX [IFK_TrackGenreId] ON [Track] ([GenreId])\n\nCREATE INDEX [IFK_TrackAlbumId] ON [Track] ([AlbumId])\n\nCREATE INDEX [IFK_PlaylistTrackTrackId] ON [PlaylistTrack] ([TrackId])\n\nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(20),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_TrackMediaTypeId] ON [Track] ([MediaTypeId])\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE INDEX [IFK_AlbumArtistId] ON [Album] ([ArtistId])\n\nCREATE TABLE [PlaylistTrack]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    CONSTRAINT [PK_PlaylistTrack] PRIMARY KEY ([PlaylistId], [TrackId]),\n    FOREIGN KEY ([PlaylistId]) REFERENCES [Playlist] ([PlaylistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Album]\n(\n    [AlbumId] INTEGER NOT NULL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER NOT NULL,\n    CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Playlist]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Playlist] PRIMARY KEY ([PlaylistId])\n)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n\n3. If the provided context is insufficient, please explain why it can't be generated. \n\n4. Please use the most relevant table(s). \n\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}; {"role": "user", "content": " \n    List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJOIN Artist ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": " SELECT * FROM t_person WHERE name = 'John Doe';"}, {"role": "assistant", "content": "SELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5"}, {"role": "user", "content": " \n    Find all tracks with a name containing \"What\" (case-insensitive)\n"}]

```

Ollama Response:

```

{'model': 'llama3:latest', 'created_at': '2024-06-08T19:49:50.396655881Z', 'message': {'role': 'assistant', 'content': "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%';"}, 'done_reason': 'stop', 'done': True, 'total_duration': 64390640532, 'load_duration': 542078, 'prompt_eval_count': 998, 'prompt_eval_duration': 61746068000, 'eval_count': 15, 'eval_duration': 2407898000}

```

```

SELECT *
FROM Track
WHERE LOWER(Name) LIKE '%what%';
Output from LLM: SELECT *

```

```

FROM Track
WHERE LOWER(Name) LIKE '%what%';
Extracted SQL: SELECT *
FROM Track
WHERE LOWER(Name) LIKE '%what%'
SELECT *
FROM Track
WHERE LOWER(Name) LIKE '%what%'

```

	TrackId	Name	AlbumId	\
0	26	What It Takes	5	
1	88	What You Are	10	
2	130	Do what cha wanna	13	
3	342	What is and Should Never Be	30	
4	607	So What	48	
5	960	What A Day	76	
6	1000	What If I Do?	80	
7	1039	What Now My Love	83	
8	1145	Whatsername	89	
9	1440	Whatever It Is, I Just Can't Stop	116	
10	1469	Look What You've Done	119	
11	1470	Get What You Need	119	
12	1628	What Is And What Should Never Be	133	
13	1778	You're What's Happening (In The World Today)	146	
14	1823	So What	149	
15	2772	I Don't Know What To Do With Myself	223	
16	2884	What Kate Did	231	
17	2893	Whatever the Case May Be	230	
18	2992	I Still Haven't Found What I'm Looking for	237	
19	3007	I Still Haven't Found What I'm Looking For	238	
20	3258	Whatever Gets You Thru the Night	255	
21	3475	What Is It About Men	322	

	MediaTypeId	GenreId	Composer	\
0	1	1	Steven Tyler, Joe Perry, Desmond Child	
1	1	1	Audioslave/Chris Cornell	
2	1	2	George Duke	
3	1	1	Jimmy Page/Robert Plant	
4	1	2	Miles Davis	
5	1	1	Mike Bordin, Billy Gould, Mike Patton	
6	1	1	Dave Grohl, Taylor Hawkins, Nate Mendel, Chris...	
7	1	12	carl sigman/gilbert becaud/pierre leroyer	
8	1	4	Green Day	

9	1	1	Jay Kay/Kay, Jay
10	1	4	N. Cester
11	1	4	C. Cester/C. Muncey/N. Cester
12	1	1	Jimmy Page, Robert Plant
13	1	14	Allen Story/George Gordy/Robert Gordy
14	1	3	Culmer/Exalt
15	1	7	None
16	3	19	None
17	3	19	None
18	1	1	Bono/Clayton, Adam/Mullen Jr., Larry/The Edge
19	1	1	U2
20	2	9	None
21	2	9	Delroy "Chris" Cooper, Donovan Jackson, Earl C...

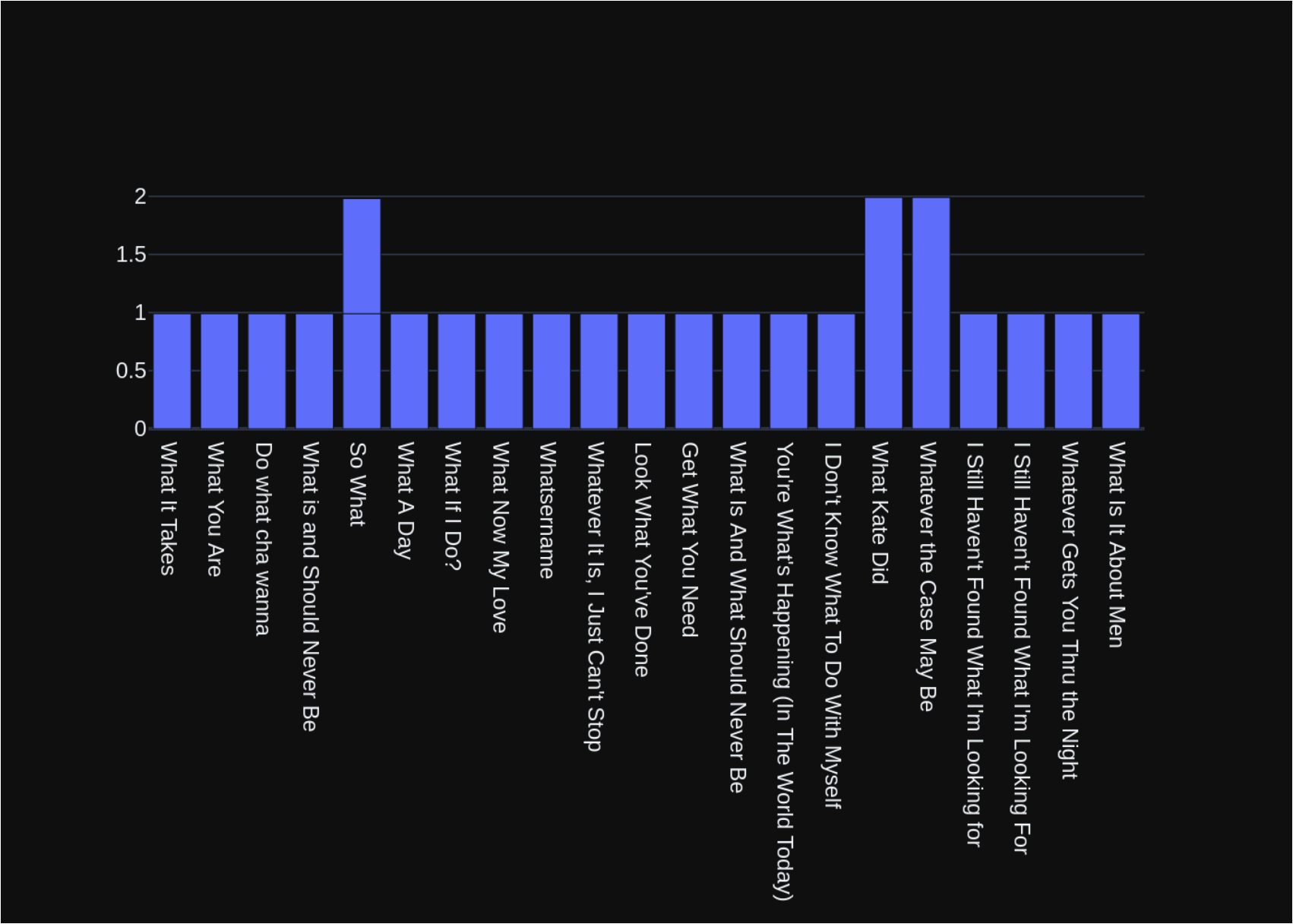
	Milliseconds	Bytes	UnitPrice
0	310622	10144730	0.99
1	249391	5988186	0.99
2	274155	9018565	0.99
3	260675	8497116	0.99
4	564009	18360449	0.99
5	158275	5203430	0.99
6	302994	9929799	0.99
7	149995	4913383	0.99
8	252316	8244843	0.99
9	247222	8249453	0.99
10	230974	7517083	0.99
11	247719	8043765	0.99
12	287973	9369385	0.99
13	142027	4631104	0.99
14	189152	6162894	0.99
15	221387	7251478	0.99
16	2610250	484583988	1.99
17	2616410	183867185	1.99
18	353567	11542247	0.99
19	280764	9306737	0.99
20	215084	3499018	0.99
21	209573	3426106	0.99

Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:

```
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n Find all tracks with a name containing \"What\" (case-insensitive)\n'\n\nThe DataFrame was produced using this query: SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n TrackId          int64\nName              object\nAlbumId          int64\nMediaTypeId      int64\nGenreId          int64\nComposer         object\nMilliseconds     int64\nBytes            int64\nUnitPrice        float64\nndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
```

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T19:50:10.86894395Z', 'message': {'role': 'assistant', 'content': "\n\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = go.Figure(data=[go.Bar(x=df['Name'], y=df['UnitPrice'])])\n\nfig.show()\n\n"}, 'done_reason': 'stop', 'done': True, 'total_duration': 20332085749, 'load_duration': 2349270, 'prompt_eval_count': 220, 'prompt_eval_duration': 13221516000, 'eval_count': 44, 'eval_duration': 7000965000}
```



Out[12]: ("SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'",

	TrackId	Name	AlbumId \
0	26	What It Takes	5
1	88	What You Are	10
2	130	Do what cha wanna	13
3	342	What is and Should Never Be	30
4	607	So What	48
5	960	What A Day	76
6	1000	What If I Do?	80
7	1039	What Now My Love	83
8	1145	Whatsername	89
9	1440	Whatever It Is, I Just Can't Stop	116
10	1469	Look What You've Done	119
11	1470	Get What You Need	119
12	1628	What Is And What Should Never Be	133
13	1778	You're What's Happening (In The World Today)	146
14	1823	So What	149
15	2772	I Don't Know What To Do With Myself	223
16	2884	What Kate Did	231
17	2893	Whatever the Case May Be	230
18	2992	I Still Haven't Found What I'm Looking for	237
19	3007	I Still Haven't Found What I'm Looking For	238
20	3258	Whatever Gets You Thru the Night	255
21	3475	What Is It About Men	322

	MediaTypeId	GenreId	Composer \
0	1	1	Steven Tyler, Joe Perry, Desmond Child
1	1	1	Audioslave/Chris Cornell
2	1	2	George Duke
3	1	1	Jimmy Page/Robert Plant
4	1	2	Miles Davis
5	1	1	Mike Bordin, Billy Gould, Mike Patton
6	1	1	Dave Grohl, Taylor Hawkins, Nate Mendel, Chris...
7	1	12	carl sigman/gilbert becaud/pierre leroyer
8	1	4	Green Day
9	1	1	Jay Kay/Kay, Jay
10	1	4	N. Cester
11	1	4	C. Cester/C. Muncey/N. Cester
12	1	1	Jimmy Page, Robert Plant
13	1	14	Allen Story/George Gordy/Robert Gordy
14	1	3	Culmer/Exalt
15	1	7	None

16	3	19	None
17	3	19	None
18	1	1	Bono/Clayton, Adam/Mullen Jr., Larry/The Edge
19	1	1	U2
20	2	9	None
21	2	9	Delroy "Chris" Cooper, Donovan Jackson, Earl C...

	Milliseconds	Bytes	UnitPrice
0	310622	10144730	0.99
1	249391	5988186	0.99
2	274155	9018565	0.99
3	260675	8497116	0.99
4	564009	18360449	0.99
5	158275	5203430	0.99
6	302994	9929799	0.99
7	149995	4913383	0.99
8	252316	8244843	0.99
9	247222	8249453	0.99
10	230974	7517083	0.99
11	247719	8043765	0.99
12	287973	9369385	0.99
13	142027	4631104	0.99
14	189152	6162894	0.99
15	221387	7251478	0.99
16	2610250	484583988	1.99
17	2616410	183867185	1.99
18	353567	11542247	0.99
19	280764	9306737	0.99
20	215084	3499018	0.99
21	209573	3426106	0.99 ,

```
Figure({
  'data': [{ 'type': 'bar',
    'x': array(['What It Takes', 'What You Are', 'Do what cha wanna',
      'What is and Should Never Be', 'So What', 'What A Day', 'What If I Do?',
      'What Now My Love', 'Whatsername', "Whatever It Is, I Just Can't Stop",
      "Look What You've Done", 'Get What You Need',
      'What Is And What Should Never Be',
      "You're What's Happening (In The World Today)", 'So What',
      "I Don't Know What To Do With Myself", 'What Kate Did',
      'Whatever the Case May Be',
      "I Still Haven't Found What I'm Looking for",
      "I Still Haven't Found What I'm Looking For",
```



```
        'Whatever Gets You Thru the Night', 'What Is It About Men'],  
        dtype=object),  
        'y': array([0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99,  
                    0.99, 0.99, 0.99, 0.99, 1.99, 1.99, 0.99, 0.99, 0.99, 0.99])),  
        'layout': {'template': '...'}  
    ))
```

```
In [13]: question = """  
        Get the total number of invoices for each customer  
        """  
  
        vn.ask(question=question)
```

```
Number of requested results 10 is greater than number of elements in index 4, updating n_results = 4  
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineInvoiceId] ON [InvoiceLine] ([InvoiceId])\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n    FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_EmployeeReportsTo] ON [Employee] ([ReportsTo])\n\nCREATE TABLE IF NOT EXISTS t_person (\n    id INT PRIMARY KEY,\n    name VARCHAR(100),\n    email text,\n    age INT\n)\n\n\n===Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines\n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql\n3. If the provided context is insufficient, please explain why it can't be generated.\n4. Please use the most relevant table(s).\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before.\n"}], {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5'}, {'role': 'user', 'content': 'List all albums and their corresponding artist names'}, {'role': 'assistant', 'content': 'SELECT A.Title, A.ArtistId, ART.Name\nFROM Album A\nJOIN Artist ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': 'Find all tracks with a name containing "What" (case-insensitive)'}, {'role': 'assistant', 'content': 'SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%wh'"}]
```

```
at%"}}, {'role': 'user', 'content': " SELECT * FROM t_person WHERE name = 'John Doe';"}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}}, {'role': 'user', 'content': ' \n    Get the total number of invoices for each customer\n'}]
```

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineInvoiceId] ON [InvoiceLine] ([InvoiceId])\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n    FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_EmployeeReportsTo] ON [Employee] ([ReportsTo])\n\nCREATE TABLE IF NOT EXISTS t_person (\n    id INT PRIMARY KEY,\n    name VARCHAR(100),\n    email text,\n    age INT\n)\n\n\n\n===Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n\n===Response Guidelines\n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql\n\n3. If the provided context is insufficie
```

nt, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJOIN Artist ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": " \n Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": " SELECT * FROM t_person WHERE name = 'John Doe';"}, {"role": "assistant", "content": "SELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content": " \n Get the total number of invoices for each customer\n"}]

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T19:51:41.848077571Z', 'message': {'role': 'assistant', 'content': 'SELECT [Customer].FirstName, [Customer].LastName, COUNT([Invoice].[InvoiceId]) AS TotalInvoices\nFROM [Customer]\nJOIN [Invoice] ON [Customer].CustomerId = [Invoice].CustomerId\nGROUP BY [Customer].FirstName, [Customer].LastName', 'done_reason': 'stop', 'done': True, 'total_duration': 90817477415, 'load_duration': 642575, 'prompt_eval_count': 1305, 'prompt_eval_duration': 81484816000, 'eval_count': 54, 'eval_duration': 9043837000}
```

```
SELECT [Customer].FirstName, [Customer].LastName, COUNT([Invoice].[InvoiceId]) AS TotalInvoices
FROM [Customer]
```

```
JOIN [Invoice] ON [Customer].CustomerId = [Invoice].CustomerId
GROUP BY [Customer].FirstName, [Customer].LastName
```

Output from LLM: SELECT [Customer].FirstName, [Customer].LastName, COUNT([Invoice].[InvoiceId]) AS TotalInvoices

```
FROM [Customer]
```

```
JOIN [Invoice] ON [Customer].CustomerId = [Invoice].CustomerId
GROUP BY [Customer].FirstName, [Customer].LastName
```

Extracted SQL: SELECT

SELECT

Couldn't run sql: Execution failed on sql 'SELECT ': incomplete input

```
In [14]: question = """
        Find the total number of invoices per country:
        """

        vn.ask(question=question)
```

Number of requested results 10 is greater than number of elements in index 4, updating n_results = 4
 Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1

```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE INDEX [IFK_InvoiceLineInvoiceId] ON [InvoiceLine] ([InvoiceId])\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n    FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_EmployeeReportsTo] ON [Employee] ([ReportsTo])\n\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE TABLE [Album]\n(\n    [AlbumId] INTEGER NOT NULL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER NOT NULL,\n    CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines\n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql\n\n3. If the provided context is insufficient, please explain why it can't be generated.\n\n4. Please use the most relevant table(s).\n\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before.\n"}], {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5'}, {'role': 'user', 'content': 'List all albums and their corresponding artist names'}, {'role': 'assistant', 'content': 'SELECT A.Title, A.ArtistId, ART.Name\nFROM Album A\nJOIN Artist ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': 'Find all tra
```

```
cks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'",}, {'role': 'user', 'content': " SELECT * FROM t_person WHERE name = 'John Doe';"}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': ' \n Find the total number of invoices per country:\n'}]
```

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\t\tON DELETE NO ACTION\nON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE INDEX [IFK_InvoiceLineInvoiceId] ON [InvoiceLine] ([InvoiceId])\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION\nON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION\nON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n    FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION\nON UPDATE NO ACTION\n)\n\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION\nON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_EmployeeReportsTo] ON [Employee] ([ReportsTo])\n\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE TABLE [Album]\n(\n    [AlbumId] INTEGER NOT NULL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER NOT NULL,\n    CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\t\tON DELETE NO ACTION\nON UPDATE NO ACTION\n)\n\n===Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines\n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but
```

t requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying `intermediate_sql` \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJOIN Artist ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": " \n Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": " SELECT * FROM t_person WHERE name = 'John Doe';"}, {"role": "assistant", "content": "SELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content": " \n Find the total number of invoices per country:\n"}]

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T19:53:12.071561745Z', 'message': {'role': 'assistant', 'content': '```\nSELECT \n I.[BillingCountry], COUNT(*) AS TotalInvoices\nFROM \n [Invoice] I\nGROUP BY \n I.[BillingCountry]\nORDER BY \n TotalInvoices DESC;\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 90182031251, 'load_duration': 560030, 'prompt_eval_count': 1318, 'prompt_eval_duration': 82349888000, 'eval_count': 45, 'eval_duration': 7498922000}
```

```
SELECT
    I.[BillingCountry], COUNT(*) AS TotalInvoices
FROM
    [Invoice] I
GROUP BY
    I.[BillingCountry]
ORDER BY
    TotalInvoices DESC;
```
```

Output from LLM: ``

```
SELECT
 I.[BillingCountry], COUNT(*) AS TotalInvoices
FROM
 [Invoice] I
GROUP BY
 I.[BillingCountry]
ORDER BY
 TotalInvoices DESC;
```
```

Extracted SQL: SELECT
I.
SELECT

I.
Couldn't run sql: Execution failed on sql 'SELECT
I.': incomplete input

```
In [15]: question = """  
         List all invoices with a total exceeding $10:  
         """>  
  
         vn.ask(question=question)
```

Number of requested results 10 is greater than number of elements in index 4, updating n_results = 4
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1


```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineInvoiceId] ON [InvoiceLine] ([InvoiceId])\n\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(220),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n    FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_EmployeeReportsTo] ON [Employee] ([ReportsTo])\n\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n\n===Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\nResponse Guidelines\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql\n3. If the provided context is insufficient, please explain why it can't be generated.\n4. Please use the most relevant table(s).\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before.\n"}], {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'a
```

```
ssistant', 'content': 'SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count D\nESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n    List all albums and their corresponding artist names\n\n'}, {'role': 'assistant', 'content': 'SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJOIN Artist A\nRT ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': ' \n    Find all tracks with a name contain\ning "What" (case-insensitive)\n\n'}, {'role': 'assistant', 'content': "SELECT *\nFROM Track\nWHERE LOWER(Nam\n e) LIKE '%what%'"}, {'role': 'user', 'content': " SELECT * FROM t_person WHERE name = 'John Doe';"}, {'rol\n e': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content':\n ' \n    List all invoices with a total exceeding $10:\n\n']}
```

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the\nquestion. Your response should ONLY be based on the given context and follow the response guidelines and fo\nrmat instructions. \n===Tables\n\nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [Custom\nerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [B\nillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [Billi\nngPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY\n([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\tON DELETE NO ACTI\nON ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineInvoiceId] ON [InvoiceLine] ([InvoiceId])\n\nCREA\nTE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE TABLE [InvoiceLine]\n(\n    [Invoice\nLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [Unit\nPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY\nKEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\tON DELETE N\nO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE N\nO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCR\nEATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId]\nINTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(220),\n    [Millise\nconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONS\nTRAIT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([Medi\naTypeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId]\nINTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [T\nitle] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Ad\ndress] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [Postal\nCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n    FOREIGN KEY ([ReportsTo]) REFERENCES [Employee]\n([EmployeeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_EmployeeReportsTo] ON\n[Employee] ([ReportsTo])\n\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREAT\nE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [L
```

```

astName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVAR
CHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Pho
ne] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGE
R,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES
[Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n\n\n===Additional Context \n\nO
ur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Res
ponse Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any
explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of
a specific string in a particular column, please generate an intermediate SQL query to find the distinct st
rings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context
is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5.
If the question has been asked and answered before, please repeat the answer exactly as it was given befor
e. \n"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "a
ssistant", "content": "SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count D
ESC\nLIMIT 5"}, {"role": "user", "content": " \n    List all albums and their corresponding artist names
\n"}, {"role": "assistant", "content": "SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJOIN Artist A
RT ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": " \n    Find all tracks with a name contain
ing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT *\nFROM Track\nWHERE LOWER(Na
me) LIKE '%what%'", {"role": "user", "content": " SELECT * FROM t_person WHERE name = 'John Doe'"}, {"rol
e": "assistant", "content": "SELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content":
" \n    List all invoices with a total exceeding $10:\n"}]

```

Ollama Response:

```

{'model': 'llama3:latest', 'created_at': '2024-06-08T19:54:33.8166476Z', 'message': {'role': 'assistant',
'content': 'SELECT * \nFROM Invoice'}, 'done_reason': 'stop', 'done': True, 'total_duration': 81704951265,
'load_duration': 635728, 'prompt_eval_count': 1274, 'prompt_eval_duration': 80492136000, 'eval_count': 6,
'eval_duration': 857856000}

```

SELECT *

FROM Invoice

SELECT *

FROM Invoice

	InvoiceId	CustomerId	InvoiceDate \
0	1	2	2009-01-01 00:00:00
1	2	4	2009-01-02 00:00:00
2	3	8	2009-01-03 00:00:00
3	4	14	2009-01-06 00:00:00
4	5	23	2009-01-11 00:00:00
..
407	408	25	2013-12-05 00:00:00
408	409	29	2013-12-06 00:00:00
409	410	35	2013-12-09 00:00:00
410	411	44	2013-12-14 00:00:00
411	412	58	2013-12-22 00:00:00

	BillingAddress	BillingCity	BillingState	\
0	Theodor-Heuss-Straße 34	Stuttgart	None	
1	Ullevålsveien 14	Oslo	None	
2	Grétrystraat 63	Brussels	None	
3	8210 111 ST NW	Edmonton	AB	
4	69 Salem Street	Boston	MA	
..	
407	319 N. Frances Street	Madison	WI	
408	796 Dundas Street West	Toronto	ON	
409	Rua dos Campeões Europeus de Viena, 4350	Porto	None	
410	Porthaninkatu 9	Helsinki	None	
411	12,Community Centre	Delhi	None	

	BillingCountry	BillingPostalCode	Total
0	Germany	70174	1.98
1	Norway	0171	3.96
2	Belgium	1000	5.94
3	Canada	T6G 2C7	8.91
4	USA	2113	13.86
..
407	USA	53703	3.96
408	Canada	M6J 1V1	5.94
409	Portugal	None	8.91
410	Finland	00530	13.86
411	India	110017	1.99

[412 rows x 9 columns]

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n List all invoices with a total exceeding $10:\n'\n\nThe DataFrame was produced using this query: SELECT * \nFROM Invoice\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\nInvoiceId int64\nCustomerId int64\nInvoiceDate object\nBillingAddress object\nBillingCity object\nBillingState object\nBillingCountry object\nBillingPostalCode object\nTotal float64\nndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just t
```

he code."}]

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T19:54:56.865546876Z', 'message': {'role': 'assistant',  
'content': "```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = px.bar(df, x='InvoiceID', y='Total')\nfig.update_layout(title='Invoices with Total Exceeding $10',\n                    xaxis_title='Invoice ID',\n                    yaxis_title='Total')\nfig.show()\n```"}, 'done_reason': 'stop',  
'done': True, 'total_duration': 22888728032, 'load_duration': 2595903, 'prompt_eval_count': 202, 'prompt_eval_duration': 11958846000, 'eval_count': 68, 'eval_duration': 10864077000}
```

Invoices with Total Exceeding \$10



Out[15]: ('SELECT * \nFROM Invoice',

	InvoiceId	CustomerId	InvoiceDate	\
0	1	2	2009-01-01 00:00:00	
1	2	4	2009-01-02 00:00:00	
2	3	8	2009-01-03 00:00:00	
3	4	14	2009-01-06 00:00:00	
4	5	23	2009-01-11 00:00:00	
..	
407	408	25	2013-12-05 00:00:00	
408	409	29	2013-12-06 00:00:00	
409	410	35	2013-12-09 00:00:00	
410	411	44	2013-12-14 00:00:00	
411	412	58	2013-12-22 00:00:00	

	BillingAddress	BillingCity	BillingState	\
0	Theodor-Heuss-Straße 34	Stuttgart	None	
1	Ullevålsveien 14	Oslo	None	
2	Grétrystraat 63	Brussels	None	
3	8210 111 ST NW	Edmonton	AB	
4	69 Salem Street	Boston	MA	
..	
407	319 N. Frances Street	Madison	WI	
408	796 Dundas Street West	Toronto	ON	
409	Rua dos Campeões Europeus de Viena, 4350	Porto	None	
410	Porthaninkatu 9	Helsinki	None	
411	12,Community Centre	Delhi	None	

	BillingCountry	BillingPostalCode	Total
0	Germany	70174	1.98
1	Norway	0171	3.96
2	Belgium	1000	5.94
3	Canada	T6G 2C7	8.91
4	USA	2113	13.86
..
407	USA	53703	3.96
408	Canada	M6J 1V1	5.94
409	Portugal	None	8.91
410	Finland	00530	13.86
411	India	110017	1.99

[412 rows x 9 columns],

Figure({

```

'data': [{ 'alignmentgroup': 'True',
           'hovertemplate': 'InvoiceId=%{x}<br>Total=%{y}<extra></extra>',
           'legendgroup': '',
           'marker': { 'color': '#636efa', 'pattern': { 'shape': '' } },
           'name': '',
           'offsetgroup': '',
           'orientation': 'v',
           'showlegend': False,
           'textposition': 'auto',
           'type': 'bar',
           'x': array([ 1, 2, 3, ..., 410, 411, 412]),
           'xaxis': 'x',
           'y': array([ 1.98, 3.96, 5.94, ..., 8.91, 13.86, 1.99]),
           'yaxis': 'y' }],
'layout': { 'barmode': 'relative',
            'legend': { 'tracegroupgap': 0 },
            'margin': { 't': 60 },
            'template': '...',
            'title': { 'text': 'Invoices with Total Exceeding $10' },
            'xaxis': { 'anchor': 'y', 'domain': [0.0, 1.0], 'title': { 'text': 'Invoice ID' } },
            'yaxis': { 'anchor': 'x', 'domain': [0.0, 1.0], 'title': { 'text': 'Total' } } }
}))

```

```

In [16]: question = """
         Find all invoices since 2010 and the total amount invoiced:
         """

vn.ask(question=question)

```

Number of requested results 10 is greater than number of elements in index 5, updating n_results = 5
 Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1


```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineInvoiceId] ON [InvoiceLine] ([InvoiceId])\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n    FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(220),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_EmployeeReportsTo] ON [Employee] ([ReportsTo])\n\nCREATE TABLE IF NOT EXISTS t_person (\n    id INT PRIMARY KEY,\n    name VARCHAR(100),\n    email text,\n    age INT\n)\n\n\n===Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n\n===Response Guidelines\n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql\n3. If the provided context is insufficient, please explain why it can't be generated.\n4. Please use the most relevant table(s).\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before.\n"}], {'role': 'use
```

```
r', 'content': ' \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content':
'SELECT * \nFROM Invoice'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come f
rom?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Countr
y\nORDER BY Count DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n    Find all tracks with a name contain
ing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT *\nFROM Track\nWHERE LOWER(Nam
e) LIKE '%what%'"}, {'role': 'user', 'content': ' \n    List all albums and their corresponding artist nam
es \n'}, {'role': 'assistant', 'content': 'SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJOIN Arti
st ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': " SELECT * FROM t_person WHERE name = 'J
ohn Doe';"}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role':
'user', 'content': ' \n    Find all invoices since 2010 and the total amount invoiced:\n'}}
```

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [Custom
erId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [B
illingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [Billi
ngPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY
([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\tON DELETE NO ACTI
ON ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineInvoiceId] ON [InvoiceLine] ([InvoiceId])\n\nCREA
TE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n
[TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n
CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invo
ice] ([InvoiceId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES
[Track] ([TrackId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceCustomerI
d] ON [Invoice] ([CustomerId])\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCRE
ATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n
[FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] D
ATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVAR
CHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fa
x] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n
FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO
N\n)\n\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT
NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n
[City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(1
0),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportR
epId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId])
REFERENCES [Employee] ([EmployeeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Trac
k]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n
```

```

[MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(220),\n    [Milliseconds]
INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track]
PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_EmployeeReportsTo] ON [Employee] ([ReportsTo])\n\n\nCREATE TABLE IF NOT EXISTS t_person (\n    id INT PRIMARY KEY,\n    name VARCHAR(100),\n    email text,\n    age INT\n)\n\n\n====Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n====Response Guidelines\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.
\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql\n3. If the provided context is insufficient, please explain why it can't be generated.
\n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM Invoice"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5"}, {"role": "user", "content": " \n    Find all tracks with a name containing 'What' (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": " \n    List all albums and their corresponding artist names\n"}, {"role": "assistant", "content": "SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJOIN Artist ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": " SELECT * FROM t_person WHERE name = 'John Doe';"}, {"role": "assistant", "content": "SELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content": " \n    Find all invoices since 2010 and the total amount invoiced:\n"}]

```

Ollama Response:

```

{'model': 'llama3:latest', 'created_at': '2024-06-08T19:56:40.116008088Z', 'message': {'role': 'assistant', 'content': '\n\nSELECT InvoiceDate, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE YEAR(InvoiceDate) >= 2010\nGROUP BY InvoiceDate;\n\n'}, 'done_reason': 'stop', 'done': True, 'total_duration': 102989129771, 'load_duration': 597309, 'prompt_eval_count': 1533, 'prompt_eval_duration': 96822209000, 'eval_count': 35, 'eval_duration': 5838178000}

```

```

SELECT InvoiceDate, SUM(Total) AS TotalAmount
FROM Invoice
WHERE YEAR(InvoiceDate) >= 2010
GROUP BY InvoiceDate;

```

Output from LLM: ```

```

SELECT InvoiceDate, SUM(Total) AS TotalAmount
FROM Invoice
WHERE YEAR(InvoiceDate) >= 2010
GROUP BY InvoiceDate;

```

```

```
Extracted SQL: SELECT InvoiceDate, SUM(Total) AS TotalAmount
FROM Invoice
WHERE YEAR(InvoiceDate) >= 2010
GROUP BY InvoiceDate
SELECT InvoiceDate, SUM(Total) AS TotalAmount
FROM Invoice
WHERE YEAR(InvoiceDate) >= 2010
GROUP BY InvoiceDate
Couldn't run sql: Execution failed on sql 'SELECT InvoiceDate, SUM(Total) AS TotalAmount
FROM Invoice
WHERE YEAR(InvoiceDate) >= 2010
GROUP BY InvoiceDate': no such function: YEAR
```

```
In [17]: question = """
 List all employees and their reporting manager's name (if any):
 """

 vn.ask(question=question)
```

Number of requested results 10 is greater than number of elements in index 5, updating n\_results = 5  
Number of requested results 10 is greater than number of elements in index 1, updating n\_results = 1

```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE INDEX [IFK_EmployeeReportsTo] ON [Employee] ([ReportsTo])\n\nCREATE TABLE [Employee]\n(\n [EmployeeId] INTEGER NOT NULL,\n [LastName] NVARCHAR(20) NOT NULL,\n [FirstName] NVARCHAR(20) NOT NULL,\n [Title] NVARCHAR(30),\n [ReportsTo] INTEGER,\n [BirthDate] DATETIME,\n [HireDate] DATETIME,\n [Address] NVARCHAR(70),\n [City] NVARCHAR(40),\n [State] NVARCHAR(40),\n [Country] NVARCHAR(40),\n [PostalCode] NVARCHAR(10),\n [Phone] NVARCHAR(24),\n [Fax] NVARCHAR(24),\n [Email] NVARCHAR(60),\n CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Customer]\n(\n [CustomerId] INTEGER NOT NULL,\n [FirstName] NVARCHAR(40) NOT NULL,\n [LastName] NVARCHAR(20) NOT NULL,\n [Company] NVARCHAR(80),\n [Address] NVARCHAR(70),\n [City] NVARCHAR(40),\n [State] NVARCHAR(40),\n [Country] NVARCHAR(40),\n [PostalCode] NVARCHAR(10),\n [Phone] NVARCHAR(24),\n [Fax] NVARCHAR(24),\n [Email] NVARCHAR(60) NOT NULL,\n [SupportRepId] INTEGER,\n CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE TABLE IF NOT EXISTS t_person (\n id INT PRIMARY KEY,\n name VARCHAR(100),\n email text,\n age INT\n)\n\nCREATE TABLE [Invoice]\n(\n [InvoiceId] INTEGER NOT NULL,\n [CustomerId] INTEGER NOT NULL,\n [InvoiceDate] DATETIME NOT NULL,\n [BillingAddress] NVARCHAR(70),\n [BillingCity] NVARCHAR(40),\n [BillingState] NVARCHAR(40),\n [BillingCountry] NVARCHAR(40),\n [BillingPostalCode] NVARCHAR(10),\n [Total] NUMERIC(10,2) NOT NULL,\n CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE TABLE [Artist]\n(\n [ArtistId] INTEGER NOT NULL,\n [Name] NVARCHAR(120),\n CONSTRAINT [PK_Artist] PRIMARY KEY ([ArtistId])\n)\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE TABLE [InvoiceLine]\n(\n [InvoiceLineId] INTEGER NOT NULL,\n [InvoiceId] INTEGER NOT NULL,\n [TrackId] INTEGER NOT NULL,\n [UnitPrice] NUMERIC(10,2) NOT NULL,\n [Quantity] INTEGER NOT NULL,\n CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines\n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql\n\n3. If the provided context is insufficient, please explain why it can't be generated.\n\n4. Please use the most relevant table(s).\n\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before.\n\n"}], {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n\nList all albums and their corresponding artist names\n\n'}, {'role': 'assistant', 'content': 'SELECT A.Title, A.ArtistId, ART.Name\nFROM Album A\nJOIN Artist ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': '\n\nList all invoices with a total exceeding $10:\n\n'}, {'role': 'assistant', 'content': 'SELECT InvoiceId, Total\nFROM Invoice\nWHERE Total > 10\nORDER BY Total DESC\nLIMIT 5'}
```

```
t': 'SELECT * \nFROM Invoice'}}, {'role': 'user', 'content': " SELECT * FROM t_person WHERE name = 'John Doe';"}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': ' \n Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': " \n List all employees and their reporting manager's name (if any):\n"}]
```

Ollama parameters:

model=llama3:latest,

options={},

keep\_alive=None

Prompt Content:

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX [IFK_EmployeeReportsTo] ON [Employee] ([ReportsTo])\n\nCREATE TABLE [Employee]\n(\n [EmployeeId] INTEGER NOT NULL,\n [LastName] NVARCHAR(20) NOT NULL,\n [FirstName] NVARCHAR(20) NOT NULL,\n [Title] NVARCHAR(30),\n [ReportsTo] INTEGER,\n [BirthDate] DATETIME,\n [HireDate] DATETIME,\n [Address] NVARCHAR(70),\n [City] NVARCHAR(40),\n [State] NVARCHAR(40),\n [Country] NVARCHAR(40),\n [PostalCode] NVARCHAR(10),\n [Phone] NVARCHAR(24),\n [Fax] NVARCHAR(24),\n [Email] NVARCHAR(60),\n CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Customer]\n(\n [CustomerId] INTEGER NOT NULL,\n [FirstName] NVARCHAR(40) NOT NULL,\n [LastName] NVARCHAR(20) NOT NULL,\n [Company] NVARCHAR(80),\n [Address] NVARCHAR(70),\n [City] NVARCHAR(40),\n [State] NVARCHAR(40),\n [Country] NVARCHAR(40),\n [PostalCode] NVARCHAR(10),\n [Phone] NVARCHAR(24),\n [Fax] NVARCHAR(24),\n [Email] NVARCHAR(60) NOT NULL,\n [SupportRepId] INTEGER,\n CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE TABLE IF NOT EXISTS t_person (\n id INT PRIMARY KEY,\n name VARCHAR(100),\n email text,\n age INT\n)\n\nCREATE TABLE [Invoice]\n(\n [InvoiceId] INTEGER NOT NULL,\n [CustomerId] INTEGER NOT NULL,\n [InvoiceDate] DATETIME NOT NULL,\n [BillingAddress] NVARCHAR(70),\n [BillingCity] NVARCHAR(40),\n [BillingState] NVARCHAR(40),\n [BillingCountry] NVARCHAR(40),\n [BillingPostalCode] NVARCHAR(10),\n [Total] NUMERIC(10,2) NOT NULL,\n CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE TABLE [Artist]\n(\n [ArtistId] INTEGER NOT NULL,\n [Name] NVARCHAR(120),\n CONSTRAINT [PK_Artist] PRIMARY KEY ([ArtistId])\n)\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE TABLE [InvoiceLine]\n(\n [InvoiceLineId] INTEGER NOT NULL,\n [InvoiceId] INTEGER NOT NULL,\n [TrackId] INTEGER NOT NULL,\n [UnitPrice] NUMERIC(10,2) NOT NULL,\n [Quantity] INTEGER NOT NULL,\n CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided
```

ed context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying `intermediate_sql \n3`. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(\*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5"}, {"role": "user", "content": "\n\nList all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJOIN Artist ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": "\n\nList all invoices with a total exceeding \$10:\n"}, {"role": "assistant", "content": "SELECT \* \nFROM Invoice"}, {"role": "user", "content": " SELECT \* FROM t\_person WHERE name = 'John Doe';"}, {"role": "assistant", "content": "SELECT \* FROM t\_person WHERE name = 'John Doe'"}, {"role": "user", "content": "\n\nFind all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT \*\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": "\n\nList all employees and their reporting manager's name (if any):\n"}]

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T19:58:11.541035285Z', 'message': {'role': 'assistant', 'content': "SELECT E.LastName, E.FirstName, \n CASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END AS ReportingManager\nFROM Employee E"}, 'done_reason': 'stop', 'done': True, 'total_duration': 91375306004, 'load_duration': 632073, 'prompt_eval_count': 1324, 'prompt_eval_duration': 82831152000, 'eval_count': 49, 'eval_duration': 8199177000}
```

```
SELECT E.LastName, E.FirstName,
 CASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END AS ReportingManager
FROM Employee E
```

```
SELECT E.LastName, E.FirstName,
```

```
 CASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END AS ReportingManager
FROM Employee E
```

|   | LastName | FirstName | ReportingManager |
|---|----------|-----------|------------------|
| 0 | Adams    | Andrew    | N/A              |
| 1 | Edwards  | Nancy     | 0                |
| 2 | Peacock  | Jane      | 0                |
| 3 | Park     | Margaret  | 0                |
| 4 | Johnson  | Steve     | 0                |
| 5 | Mitchell | Michael   | 0                |
| 6 | King     | Robert    | 0                |
| 7 | Callahan | Laura     | 0                |

Ollama parameters:

```
model=llama3:latest,
options={},
keep_alive=None
```

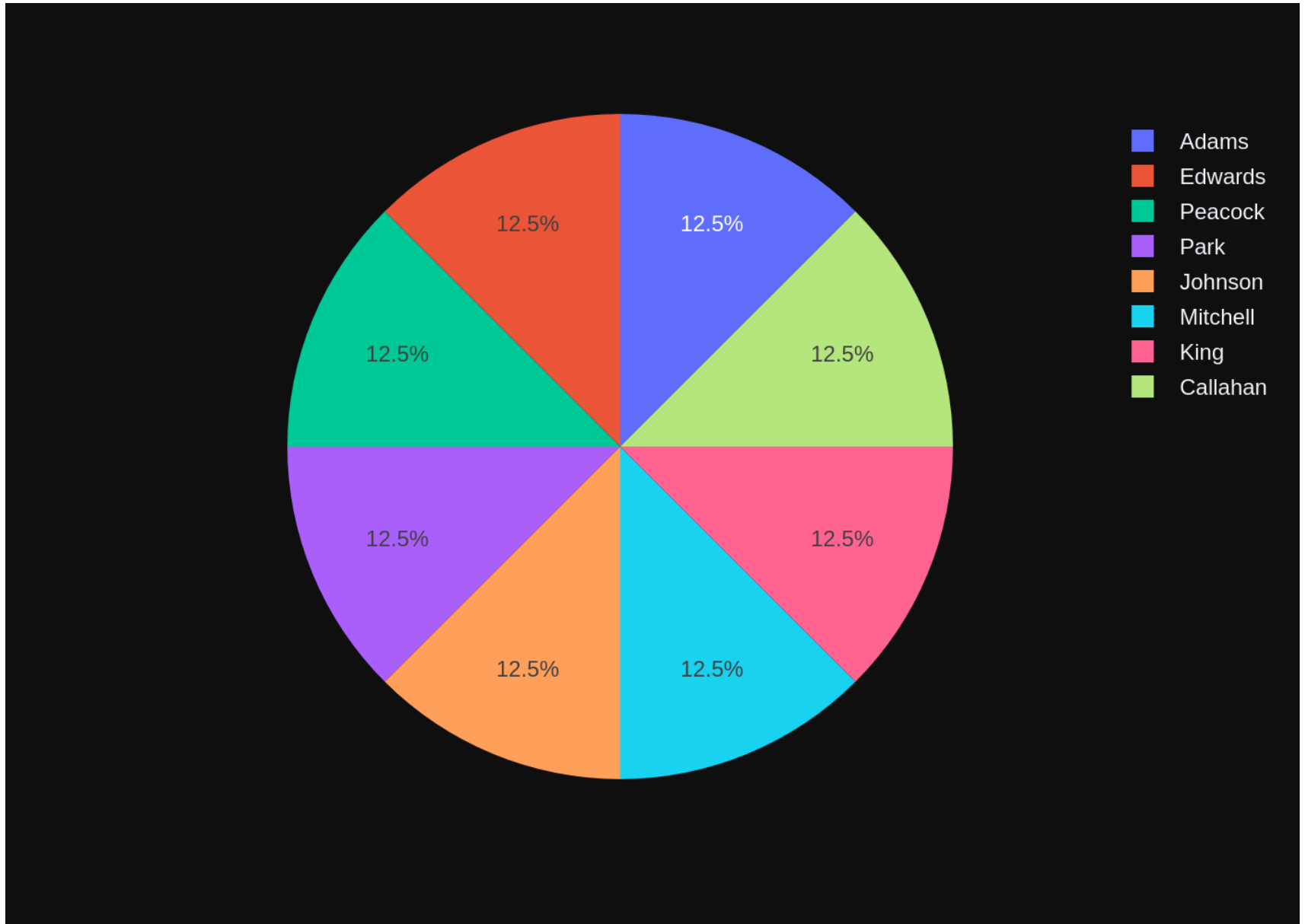
Prompt Content:

```
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n List all employees and their reporting manager's name (if any):\n'\n\nThe DataFrame was produced using this query: SELECT E.LastName, E.FirstName, \n CASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END AS ReportingManager\nFROM Employee E\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n LastName object\n FirstName object\n ReportingManager object\n dtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
```

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T19:58:39.464719374Z', 'message': {'role': 'assistant', 'content': '\n\nimport plotly.express as px\n\nfig = px.bar(df, x=\'LastName\', y=\'ReportingManager\', title=\'Employee Reporting Manager\')\nfig.update_layout(yaxis_title=\'Reporting Manager Name\')\n\nif df.shape[0] == 1:\n fig = px.line_pivot_table(df, keys="ReportingManager", values="FirstName", title=\'Single Employee Reporting Manager\')\nelse:\n fig.show()\n\n'}, 'done_reason': 'stop', 'done': True, 'total_duration': 27806730757, 'load_duration': 712415, 'prompt_eval_count': 223, 'prompt_eval_duration': 13288191000, 'eval_count': 90, 'eval_duration': 14470524000}
```





```

Out[17]: ("SELECT E.LastName, E.FirstName, \n CASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (SELECT LastName +
', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END AS ReportingManager\nFROM Employee E",
 LastName FirstName ReportingManager
0 Adams Andrew N/A
1 Edwards Nancy 0
2 Peacock Jane 0
3 Park Margaret 0
4 Johnson Steve 0
5 Mitchell Michael 0
6 King Robert 0
7 Callahan Laura 0,
Figure({
 'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
 'hovertemplate': 'LastName=%{label}<extra></extra>',
 'labels': array(['Adams', 'Edwards', 'Peacock', 'Park', 'Johnson', 'Mitchell', 'King',
 'Callahan'], dtype=object),
 'legendgroup': '',
 'name': '',
 'showlegend': True,
 'type': 'pie'}],
 'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'template': '...'}
}))

```

```

In [18]: question = """
 Get the average invoice total for each customer:
 """

vn.ask(question=question)

```

Number of requested results 10 is greater than number of elements in index 6, updating n\_results = 6  
 Number of requested results 10 is greater than number of elements in index 1, updating n\_results = 1

```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE TABLE [Invoice]\n(\n [InvoiceId] INTEGER NOT NULL,\n [CustomerId] INTEGER NOT NULL,\n [InvoiceDate] DATETIME NOT NULL,\n [BillingAddress] NVARCHAR(70),\n [BillingCity] NVARCHAR(40),\n [BillingState] NVARCHAR(40),\n [BillingCountry] NVARCHAR(40),\n [BillingPostalCode] NVARCHAR(10),\n [Total] NUMERIC(10,2) NOT NULL,\n CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineInvoiceId] ON [InvoiceLine] ([InvoiceId])\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE TABLE [InvoiceLine]\n(\n [InvoiceLineId] INTEGER NOT NULL,\n [InvoiceId] INTEGER NOT NULL,\n [TrackId] INTEGER NOT NULL,\n [UnitPrice] NUMERIC(10,2) NOT NULL,\n [Quantity] INTEGER NOT NULL,\n CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE INDEX [IFK_EmployeeReportsTo] ON [Employee] ([ReportsTo])\n\nCREATE TABLE [Employee]\n(\n [EmployeeId] INTEGER NOT NULL,\n [LastName] NVARCHAR(20) NOT NULL,\n [FirstName] NVARCHAR(20) NOT NULL,\n [Title] NVARCHAR(30),\n [ReportsTo] INTEGER,\n [BirthDate] DATETIME,\n [HireDate] DATETIME,\n [Address] NVARCHAR(70),\n [City] NVARCHAR(40),\n [State] NVARCHAR(40),\n [Country] NVARCHAR(40),\n [PostalCode] NVARCHAR(10),\n [Phone] NVARCHAR(24),\n [Fax] NVARCHAR(24),\n [Email] NVARCHAR(60),\n CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE IF NOT EXISTS t_person (\n id INT PRIMARY KEY,\n name VARCHAR(100),\n email text,\n age INT\n)\n\nCREATE TABLE [Customer]\n(\n [CustomerId] INTEGER NOT NULL,\n [FirstName] NVARCHAR(40) NOT NULL,\n [LastName] NVARCHAR(20) NOT NULL,\n [Company] NVARCHAR(80),\n [Address] NVARCHAR(70),\n [City] NVARCHAR(40),\n [State] NVARCHAR(40),\n [Country] NVARCHAR(40),\n [PostalCode] NVARCHAR(10),\n [Phone] NVARCHAR(24),\n [Fax] NVARCHAR(24),\n [Email] NVARCHAR(60) NOT NULL,\n [SupportRepId] INTEGER,\n CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines\n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql\n\n3. If the provided context is insufficient, please explain why it can't be generated.\n\n4. Please use the most relevant table(s).\n\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before.\n\n", {'role': 'user', 'content': '\n List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM Invoice'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n List all employees and the reporting manager's name (if any):\n'}, {'role': 'assistant', 'content': 'SELECT E.LastName, E.FirstName, M.LastName, M.FirstName\nFROM Employee E\nLEFT JOIN Employee M ON E.ReportsTo = M.EmployeeId\nCASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee W\n WHERE W.EmployeeId = E.ReportsTo) END\nORDER BY E.LastName, E.FirstName\n"}]
```

```
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
```

132/214

ines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate\_sql\n3. If the provided context is insufficient, please explain why it can't be generated.\n4. Please use the most relevant table(s).\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before.\n"}, {"role": "user", "content": " \n List all invoices with a total exceeding \$10:\n"}, {"role": "assistant", "content": "SELECT \* \nFROM Invoice"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(\*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all employees and the reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT E.LastName, E.FirstName, \n CASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END AS ReportingManager\nFROM Employee E"}, {"role": "user", "content": " \n Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT \* \nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": " \n List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJOIN Artist ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": " \n SELECT \* FROM t\_person WHERE name = 'John Doe';"}, {"role": "assistant", "content": "SELECT \* FROM t\_person WHERE name = 'John Doe'"}, {"role": "user", "content": " \n Get the average invoice total for each customer:\n"}]

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T20:00:15.098873637Z', 'message': {'role': 'assistant', 'content': 'SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName', 'done_reason': 'stop', 'done': True, 'total_duration': 95445858565, 'load_duration': 613625, 'prompt_eval_count': 1424, 'prompt_eval_duration': 89208140000, 'eval_count': 35, 'eval_duration': 5803964000}
```

```
SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal
FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
GROUP BY C.LastName
SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal
FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
GROUP BY C.LastName
```

|   | LastName | AverageInvoiceTotal |
|---|----------|---------------------|
| 0 | Almeida  | 5.374286            |
| 1 | Barnett  | 6.231429            |
| 2 | Bernard  | 5.517143            |
| 3 | Brooks   | 5.374286            |
| 4 | Brown    | 5.374286            |
| 5 | Chase    | 5.374286            |

|    |            |          |
|----|------------|----------|
| 6  | Cunningham | 6.802857 |
| 7  | Dubois     | 5.374286 |
| 8  | Fernandes  | 5.660000 |
| 9  | Francis    | 5.374286 |
| 10 | Girard     | 5.660000 |
| 11 | Gonçalves  | 5.660000 |
| 12 | Gordon     | 5.374286 |
| 13 | Goyer      | 5.517143 |
| 14 | Gray       | 5.374286 |
| 15 | Gruber     | 6.088571 |
| 16 | Gutiérrez  | 5.374286 |
| 17 | Hansen     | 5.660000 |
| 18 | Harris     | 5.374286 |
| 19 | Holý       | 7.088571 |
| 20 | Hughes     | 5.374286 |
| 21 | Hämäläinen | 5.945714 |
| 22 | Johansson  | 5.517143 |
| 23 | Jones      | 5.374286 |
| 24 | Kovács     | 6.517143 |
| 25 | Köhler     | 5.374286 |
| 26 | Leacock    | 5.660000 |
| 27 | Lefebvre   | 5.517143 |
| 28 | Mancini    | 5.374286 |
| 29 | Martins    | 5.374286 |
| 30 | Mercier    | 5.802857 |
| 31 | Miller     | 5.660000 |
| 32 | Mitchell   | 5.374286 |
| 33 | Murray     | 5.374286 |
| 34 | Muñoz      | 5.374286 |
| 35 | Nielsen    | 5.374286 |
| 36 | O'Reilly   | 6.517143 |
| 37 | Pareek     | 5.517143 |
| 38 | Peeters    | 5.374286 |
| 39 | Peterson   | 5.517143 |
| 40 | Philips    | 5.374286 |
| 41 | Ralston    | 6.231429 |
| 42 | Ramos      | 5.374286 |
| 43 | Rocha      | 5.374286 |
| 44 | Rojas      | 6.660000 |
| 45 | Sampaio    | 5.374286 |
| 46 | Schneider  | 5.374286 |
| 47 | Schröder   | 5.374286 |

|    |              |          |
|----|--------------|----------|
| 48 | Silk         | 5.374286 |
| 49 | Smith        | 5.660000 |
| 50 | Srivastava   | 6.106667 |
| 51 | Stevens      | 6.088571 |
| 52 | Sullivan     | 5.374286 |
| 53 | Taylor       | 5.374286 |
| 54 | Tremblay     | 5.660000 |
| 55 | Van der Berg | 5.802857 |
| 56 | Wichterlová  | 5.802857 |
| 57 | Wójcik       | 5.374286 |
| 58 | Zimmermann   | 6.231429 |

Ollama parameters:

model=llama3:latest,

options={},

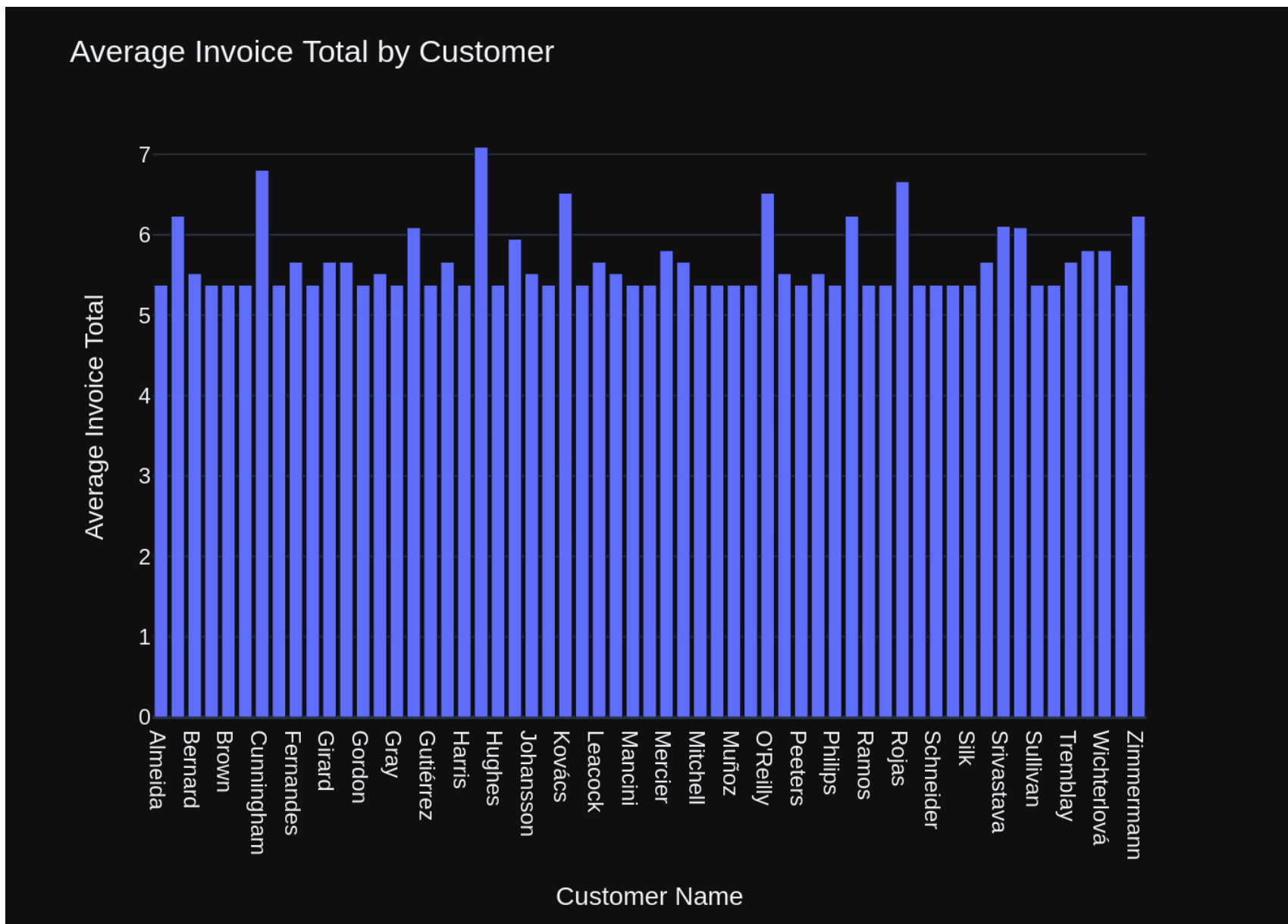
keep\_alive=None

Prompt Content:

```
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n Get the average invoice total for each customer:\n'\n\nThe DataFrame was produced using this query: SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\nLastName object\nAverageInvoiceTotal float64\nndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
```

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T20:00:36.382523615Z', 'message': {'role': 'assistant', 'content': "\n\nimport plotly.express as px\n\nfig = px.bar(df, x='LastName', y='AverageInvoiceTotal')\n\nfig.update_layout(title='Average Invoice Total by Customer',\n\n yaxis_title='Average Invoice Total',\n\n xaxis_title='Customer Name')\n\nfig.show()\n\n"}, 'done_reason': 'stop', 'done': True, 'total_duration': 21135077298, 'load_duration': 955256, 'prompt_eval_count': 195, 'prompt_eval_duration': 11592755000, 'eval_count': 59, 'eval_duration': 9391230000}
```





```
Out[18]: ('SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId\n= I.CustomerId\nGROUP BY C.LastName',
```

|    | LastName   | AverageInvoiceTotal |
|----|------------|---------------------|
| 0  | Almeida    | 5.374286            |
| 1  | Barnett    | 6.231429            |
| 2  | Bernard    | 5.517143            |
| 3  | Brooks     | 5.374286            |
| 4  | Brown      | 5.374286            |
| 5  | Chase      | 5.374286            |
| 6  | Cunningham | 6.802857            |
| 7  | Dubois     | 5.374286            |
| 8  | Fernandes  | 5.660000            |
| 9  | Francis    | 5.374286            |
| 10 | Girard     | 5.660000            |
| 11 | Gonçalves  | 5.660000            |
| 12 | Gordon     | 5.374286            |
| 13 | Goyer      | 5.517143            |
| 14 | Gray       | 5.374286            |
| 15 | Gruber     | 6.088571            |
| 16 | Gutiérrez  | 5.374286            |
| 17 | Hansen     | 5.660000            |
| 18 | Harris     | 5.374286            |
| 19 | Holý       | 7.088571            |
| 20 | Hughes     | 5.374286            |
| 21 | Hämäläinen | 5.945714            |
| 22 | Johansson  | 5.517143            |
| 23 | Jones      | 5.374286            |
| 24 | Kovács     | 6.517143            |
| 25 | Köhler     | 5.374286            |
| 26 | Leacock    | 5.660000            |
| 27 | Lefebvre   | 5.517143            |
| 28 | Mancini    | 5.374286            |
| 29 | Martins    | 5.374286            |
| 30 | Mercier    | 5.802857            |
| 31 | Miller     | 5.660000            |
| 32 | Mitchell   | 5.374286            |
| 33 | Murray     | 5.374286            |
| 34 | Muñoz      | 5.374286            |
| 35 | Nielsen    | 5.374286            |
| 36 | O'Reilly   | 6.517143            |
| 37 | Pareek     | 5.517143            |
| 38 | Peeters    | 5.374286            |

|    |              |           |
|----|--------------|-----------|
| 39 | Peterson     | 5.517143  |
| 40 | Philips      | 5.374286  |
| 41 | Ralston      | 6.231429  |
| 42 | Ramos        | 5.374286  |
| 43 | Rocha        | 5.374286  |
| 44 | Rojas        | 6.660000  |
| 45 | Sampaio      | 5.374286  |
| 46 | Schneider    | 5.374286  |
| 47 | Schröder     | 5.374286  |
| 48 | Silk         | 5.374286  |
| 49 | Smith        | 5.660000  |
| 50 | Srivastava   | 6.106667  |
| 51 | Stevens      | 6.088571  |
| 52 | Sullivan     | 5.374286  |
| 53 | Taylor       | 5.374286  |
| 54 | Tremblay     | 5.660000  |
| 55 | Van der Berg | 5.802857  |
| 56 | Wichterlová  | 5.802857  |
| 57 | Wójcik       | 5.374286  |
| 58 | Zimmermann   | 6.231429, |

```
Figure({
 'data': [{'alignmentgroup': 'True',
 'hovertemplate': 'LastName=%{x}
AverageInvoiceTotal=%{y}<extra></extra>',
 'legendgroup': '',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': '',
 'offsetgroup': '',
 'orientation': 'v',
 'showlegend': False,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['Almeida', 'Barnett', 'Bernard', 'Brooks', 'Brown', 'Chase',
 'Cunningham', 'Dubois', 'Fernandes', 'Francis', 'Girard', 'Gonçalves',
 'Gordon', 'Goyer', 'Gray', 'Gruber', 'Gutiérrez', 'Hansen', 'Harris',
 'Holý', 'Hughes', 'Hämäläinen', 'Johansson', 'Jones', 'Kovács',
 'Köhler', 'Leacock', 'Lefebvre', 'Mancini', 'Martins', 'Mercier',
 'Miller', 'Mitchell', 'Murray', 'Muñoz', 'Nielsen', "O'Reilly",
 'Pareek', 'Peeters', 'Peterson', 'Philips', 'Ralston', 'Ramos', 'Rocha',
 'Rojas', 'Sampaio', 'Schneider', 'Schröder', 'Silk', 'Smith',
 'Srivastava', 'Stevens', 'Sullivan', 'Taylor', 'Tremblay',
 'Van der Berg', 'Wichterlová', 'Wójcik', 'Zimmermann'], dtype=object),
 'xaxis': 'x',
```

```

'y': array([5.37428571, 6.23142857, 5.51714286, 5.37428571, 5.37428571, 5.37428571,
 6.80285714, 5.37428571, 5.66 , 5.37428571, 5.66 , 5.66 ,
 5.37428571, 5.51714286, 5.37428571, 6.08857143, 5.37428571, 5.66 ,
 5.37428571, 7.08857143, 5.37428571, 5.94571429, 5.51714286, 5.37428571,
 6.51714286, 5.37428571, 5.66 , 5.51714286, 5.37428571, 5.37428571,
 5.80285714, 5.66 , 5.37428571, 5.37428571, 5.37428571, 5.37428571,
 6.51714286, 5.51714286, 5.37428571, 5.51714286, 5.37428571, 6.23142857,
 5.37428571, 5.37428571, 6.66 , 5.37428571, 5.37428571, 5.37428571,
 5.37428571, 5.66 , 6.10666667, 6.08857143, 5.37428571, 5.37428571,
 5.66 , 5.80285714, 5.80285714, 5.37428571, 6.23142857]),
'yaxis': 'y'}],
'layout': {'barmode': 'relative',
'legend': {'tracegroupgap': 0},
'margin': {'t': 60},
'template': '...',
'title': {'text': 'Average Invoice Total by Customer'},
'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Customer Name'}},
'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Average Invoice Tota
l'}}}]
}))

```

```

In [19]: question = """
 Find the top 5 most expensive tracks (based on unit price):
 """

 vn.ask(question=question)

```

Number of requested results 10 is greater than number of elements in index 7, updating n\_results = 7  
 Number of requested results 10 is greater than number of elements in index 1, updating n\_results = 1

```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE TABLE [Track]\n(\n [TrackId] INTEGER NOT NULL,\n [Name] NVARCHAR(200) NOT NULL,\n [AlbumId] INTEGER,\n [MediaTypeId] INTEGER NOT NULL,\n [GenreId] INTEGER,\n [Composer] NVARCHAR(220),\n [Milliseconds] INTEGER NOT NULL,\n [Bytes] INTEGER,\n [UnitPrice] NUMERIC(10,2) NOT NULL,\n CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_TrackAlbumId] ON [Track] ([AlbumId])\n\nCREATE INDEX [IFK_TrackGenreId] ON [Track] ([GenreId])\n\nCREATE INDEX [IFK_PlaylistTrackTrackId] ON [PlaylistTrack] ([TrackId])\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE INDEX [IFK_TrackMediaTypeId] ON [Track] ([MediaTypeId])\n\nCREATE INDEX [IFK_AlbumArtistId] ON [Album] ([ArtistId])\n\nCREATE TABLE [InvoiceLine]\n(\n [InvoiceLineId] INTEGER NOT NULL,\n [InvoiceId] INTEGER NOT NULL,\n [TrackId] INTEGER NOT NULL,\n [UnitPrice] NUMERIC(10,2) NOT NULL,\n [Quantity] INTEGER NOT NULL,\n CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [PlaylistTrack]\n(\n [PlaylistId] INTEGER NOT NULL,\n [TrackId] INTEGER NOT NULL,\n CONSTRAINT [PK_PlaylistTrack] PRIMARY KEY ([PlaylistId], [TrackId]),\n FOREIGN KEY ([PlaylistId]) REFERENCES [Playlist] ([PlaylistId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Album]\n(\n [AlbumId] INTEGER NOT NULL,\n [Title] NVARCHAR(160) NOT NULL,\n [ArtistId] INTEGER NOT NULL,\n CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines\n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql\n\n3. If the provided context is insufficient, please explain why it can't be generated.\n\n4. Please use the most relevant table(s).\n\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before.\n"}], {'role': 'user', 'content': ' \n Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': ' \n List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT *\nFROM Invoice'}, {'role': 'user', 'content': ' \n List all albums and their corresponding artist names\n'}, {'role': 'assistant', 'content': 'SELECT A.Title, A.ArtistId, ART.Name\nFROM Album A\nJOIN Artist ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': ' \n Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5'}, {'role': 'user', 'content': " SELECT * FROM t_person WHERE name = 'John Doe';"}], {'ro
```

```

le': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'", {'role': 'user', 'content':
" \n List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content':
t': "SELECT E.LastName, E.FirstName, \n CASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (SELECT LastName
+ ' ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END AS ReportingManager\nFROM Employee
E"}, {'role': 'user', 'content': ' \n Find the top 5 most expensive tracks (based on unit price):\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE [Track]\n(\n [TrackId] INTEGER NOT NULL,\n [Name] NVAR
CHAR(200) NOT NULL,\n [AlbumId] INTEGER,\n [MediaTypeId] INTEGER NOT NULL,\n [GenreId] INTEGE
R,\n [Composer] NVARCHAR(220),\n [Milliseconds] INTEGER NOT NULL,\n [Bytes] INTEGER,\n [UnitPr
ice] NUMERIC(10,2) NOT NULL,\n CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n FOREIGN KEY ([Album
Id]) REFERENCES [Album] ([AlbumId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([Genre
Id]) REFERENCES [Genre] ([GenreId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([Media
TypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE
INDEX [IFK_TrackAlbumId] ON [Track] ([AlbumId])\n\nCREATE INDEX [IFK_TrackGenreId] ON [Track] ([GenreId])\n
\nCREATE INDEX [IFK_PlaylistTrackTrackId] ON [PlaylistTrack] ([TrackId])\n\nCREATE INDEX [IFK_InvoiceLineTr
ackId] ON [InvoiceLine] ([TrackId])\n\nCREATE INDEX [IFK_TrackMediaTypeId] ON [Track] ([MediaTypeId])\n\nCR
EATE INDEX [IFK_AlbumArtistId] ON [Album] ([ArtistId])\n\nCREATE TABLE [InvoiceLine]\n(\n [InvoiceLineI
d] INTEGER NOT NULL,\n [InvoiceId] INTEGER NOT NULL,\n [TrackId] INTEGER NOT NULL,\n [UnitPric
e] NUMERIC(10,2) NOT NULL,\n [Quantity] INTEGER NOT NULL,\n CONSTRAINT [PK_InvoiceLine] PRIMARY KEY
([InvoiceLineId]),\n FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACT
ION ON UPDATE NO ACTION,\n FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACT
ION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [PlaylistTrack]\n(\n [PlaylistId] INTEGER NOT NULL,\n [Tr
ackId] INTEGER NOT NULL,\n CONSTRAINT [PK_PlaylistTrack] PRIMARY KEY ([PlaylistId], [TrackId]),\n F
OREIGN KEY ([PlaylistId]) REFERENCES [Playlist] ([PlaylistId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTIO
N,\n FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTIO
N\n)\n\nCREATE TABLE [Album]\n(\n [AlbumId] INTEGER NOT NULL,\n [Title] NVARCHAR(160) NOT NULL,\n
 [ArtistId] INTEGER NOT NULL,\n CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n FOREIGN KEY ([Artis
tId]) REFERENCES [Artist] ([ArtistId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n\n===Additional
Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in f
ull\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query
without any explanations for the question. \n2. If the provided context is almost sufficient but requires k
nowledge of a specific string in a particular column, please generate an intermediate SQL query to find the
distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provi
ded context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant ta
ble(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was
given before. \n"}, {"role": "user", "content": " \n Find all tracks with a name containing \"What\" (c

```

```
ase-insensitive)\n"}, {"role": "assistant", "content": "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%wha
t%'"}, {"role": "user", "content": " \n List all invoices with a total exceeding $10:\n"}, {"role": "as
sistant", "content": "SELECT *\nFROM Invoice"}, {"role": "user", "content": " \n List all albums and t
heir corresponding artist names \n"}, {"role": "assistant", "content": "SELECT A.Title, A.ArtistId, ART.Na
me \nFROM Album A \nJOIN Artist ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": " \n Ge
t the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT C.LastName, AV
G(I.Total) AS AverageInvoiceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY
C.LastName"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"rol
e": "assistant", "content": "SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY C
ount DESC\nLIMIT 5"}, {"role": "user", "content": " SELECT * FROM t_person WHERE name = 'John Doe';"}, {"ro
le": "assistant", "content": "SELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content":
" \n List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "conten
t": "SELECT E.LastName, E.FirstName, \n CASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (SELECT LastName
+ ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END AS ReportingManager\nFROM Employee
E"}, {"role": "user", "content": " \n Find the top 5 most expensive tracks (based on unit price):\n"}]
```

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T20:02:02.311579523Z', 'message': {'role': 'assistant',
'content': 'SELECT TOP 5 *\nFROM Track\nORDER BY UnitPrice DESC;'}, 'done_reason': 'stop', 'done': True, 't
otal_duration': 85766730990, 'load_duration': 623586, 'prompt_eval_count': 1328, 'prompt_eval_duration': 82
839725000, 'eval_count': 15, 'eval_duration': 2393108000}
```

```
SELECT TOP 5 *
```

```
FROM Track
```

```
ORDER BY UnitPrice DESC;
```

```
Output from LLM: SELECT TOP 5 *
```

```
FROM Track
```

```
ORDER BY UnitPrice DESC;
```

```
Extracted SQL: SELECT TOP 5 *
```

```
FROM Track
```

```
ORDER BY UnitPrice DESC
```

```
SELECT TOP 5 *
```

```
FROM Track
```

```
ORDER BY UnitPrice DESC
```

```
Couldn't run sql: Execution failed on sql 'SELECT TOP 5 *
```

```
FROM Track
```

```
ORDER BY UnitPrice DESC': near "5": syntax error
```

```
In [20]: question = """
 List all genres and the number of tracks in each genre:
 """

 vn.ask(question=question)
```

```
Number of requested results 10 is greater than number of elements in index 7, updating n_results = 7
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE INDEX [IFK_TrackGenreId] ON [Track] ([GenreId])\n\nCREATE TABLE [Track]\n(\n [TrackId] INTEGER NOT NULL,\n [Name] NVARCHAR(200) NOT NULL,\n [AlbumId] INTEGER,\n [MediaTypeId] INTEGER NOT NULL,\n [GenreId] INTEGER,\n [Composer] NVARCHAR(220),\n [Milliseconds] INTEGER NOT NULL,\n [Bytes] INTEGER,\n [UnitPrice] NUMERIC(10,2) NOT NULL,\n CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_TrackAlbumId] ON [Track] ([AlbumId])\n\nCREATE TABLE [Genre]\n(\n [GenreId] INTEGER NOT NULL,\n [Name] NVARCHAR(120),\n CONSTRAINT [PK_Genre] PRIMARY KEY ([GenreId])\n)\n\nCREATE INDEX [IFK_PlaylistTrackTrackId] ON [PlaylistTrack] ([TrackId])\n\nCREATE INDEX [IFK_TrackMediaTypeId] ON [Track] ([MediaTypeId])\n\nCREATE INDEX [IFK_AlbumArtistId] ON [Album] ([ArtistId])\n\nCREATE TABLE [Album]\n(\n [AlbumId] INTEGER NOT NULL,\n [Title] NVARCHAR(160) NOT NULL,\n [ArtistId] INTEGER NOT NULL,\n CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [PlaylistTrack]\n(\n [PlaylistId] INTEGER NOT NULL,\n [TrackId] INTEGER NOT NULL,\n CONSTRAINT [PK_PlaylistTrack] PRIMARY KEY ([PlaylistId], [TrackId]),\n FOREIGN KEY ([PlaylistId]) REFERENCES [Playlist] ([PlaylistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Playlist]\n(\n [PlaylistId] INTEGER NOT NULL,\n [Name] NVARCHAR(120),\n CONSTRAINT [PK_Playlist] PRIMARY KEY ([PlaylistId])\n)\n\n\n===Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines\n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql\n3. If the provided context is insufficient, please explain why it can't be generated.\n4. Please use the most relevant table(s).\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before.\n"}], {'role': 'user', 'content': '\n List all albums and their corresponding artist names\n'}, {'role': 'assistant', 'content': 'SELECT A.Title, A.ArtistId, ART.Name\nFROM Album A\nJOIN Artist ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': '\n Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': 'SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName'}, {'role': 'user', 'content': '\n List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT *\nFROM Invoice'}, {'role': 'user', 'content': '\n List all employees and their reporting manager's name (if any):\n'}, {'role': 'assistant', 'content': 'SELECT E.LastName, E.FirstName, \n CASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END AS ReportingManager\nFROM Employee E'}, {'role': 'user', 'content': '

```



```
" SELECT * FROM t_person WHERE name = 'John Doe';"}}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}}, {'role': 'user', 'content': ' \n List all genres and the number of tracks in each genre:\n'}]
```

Ollama parameters:

model=llama3:latest,

options={},

keep\_alive=None

Prompt Content:

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX [IFK_TrackGenreId] ON [Track] ([GenreId])\n\nCREATE TABLE [Track]\n(\n [TrackId] INTEGER NOT NULL,\n [Name] NVARCHAR(200) NOT NULL,\n [AlbumId] INTEGER,\n [MediaTypeId] INTEGER NOT NULL,\n [GenreId] INTEGER,\n [Composer] NVARCHAR(220),\n [Milliseconds] INTEGER NOT NULL,\n [Bytes] INTEGER,\n [UnitPrice] NUMERIC(10,2) NOT NULL,\n CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_TrackAlbumId] ON [Track] ([AlbumId])\n\nCREATE TABLE [Genre]\n(\n [GenreId] INTEGER NOT NULL,\n [Name] NVARCHAR(120),\n CONSTRAINT [PK_Genre] PRIMARY KEY ([GenreId])\n)\n\nCREATE INDEX [IFK_PlaylistTrackTrackId] ON [PlaylistTrack] ([TrackId])\n\nCREATE INDEX [IFK_TrackMediaTypeId] ON [Track] ([MediaTypeId])\n\nCREATE INDEX [IFK_AlbumArtistId] ON [Album] ([ArtistId])\n\nCREATE TABLE [Album]\n(\n [AlbumId] INTEGER NOT NULL,\n [Title] NVARCHAR(160) NOT NULL,\n [ArtistId] INTEGER NOT NULL,\n CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [PlaylistTrack]\n(\n [PlaylistId] INTEGER NOT NULL,\n [TrackId] INTEGER NOT NULL,\n CONSTRAINT [PK_PlaylistTrack] PRIMARY KEY ([PlaylistId], [TrackId]),\n FOREIGN KEY ([PlaylistId]) REFERENCES [Playlist] ([PlaylistId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Playlist]\n(\n [PlaylistId] INTEGER NOT NULL,\n [Name] NVARCHAR(120),\n CONSTRAINT [PK_Playlist] PRIMARY KEY ([PlaylistId])\n)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}], {"role": "user", "content": " \n List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJOIN Artist ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": " \n Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS Count\nFR
```

```
OM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5"}, {"role": "user", "content": " \n Get the
average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT C.LastName, AVG(I.To
tal) AS AverageInvoiceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.Las
tName"}, {"role": "user", "content": " \n List all invoices with a total exceeding $10:\n"}, {"role":
"assistant", "content": "SELECT * \nFROM Invoice"}, {"role": "user", "content": " \n List all employees
and their reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT E.LastName, E.Fi
rstName, \n CASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Empl
oyee WHERE EmployeeId = E.ReportsTo) END AS ReportingManager\nFROM Employee E"}, {"role": "user", "conten
t": " SELECT * FROM t_person WHERE name = 'John Doe';"}, {"role": "assistant", "content": "SELECT * FROM t_
person WHERE name = 'John Doe'"}, {"role": "user", "content": " \n List all genres and the number of tr
acks in each genre:\n"}]]
```

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T20:03:23.855543745Z', 'message': {'role': 'assistant',
'content': 'SELECT G.Name, COUNT(T.TrackId) AS TrackCount \nFROM Genre G \nLEFT JOIN Track T ON G.GenreId =
T.GenreId \nGROUP BY G.Name'}, 'done_reason': 'stop', 'done': True, 'total_duration': 81502125975, 'load_du
ration': 1091557, 'prompt_eval_count': 1202, 'prompt_eval_duration': 74913045000, 'eval_count': 37, 'eval_d
uration': 6095847000}
```

```
SELECT G.Name, COUNT(T.TrackId) AS TrackCount
FROM Genre G
LEFT JOIN Track T ON G.GenreId = T.GenreId
GROUP BY G.Name
SELECT G.Name, COUNT(T.TrackId) AS TrackCount
FROM Genre G
LEFT JOIN Track T ON G.GenreId = T.GenreId
GROUP BY G.Name
```

|    | Name               | TrackCount |
|----|--------------------|------------|
| 0  | Alternative        | 40         |
| 1  | Alternative & Punk | 332        |
| 2  | Blues              | 81         |
| 3  | Bossa Nova         | 15         |
| 4  | Classical          | 74         |
| 5  | Comedy             | 17         |
| 6  | Drama              | 64         |
| 7  | Easy Listening     | 24         |
| 8  | Electronica/Dance  | 30         |
| 9  | Heavy Metal        | 28         |
| 10 | Hip Hop/Rap        | 35         |
| 11 | Jazz               | 130        |
| 12 | Latin              | 579        |
| 13 | Metal              | 374        |
| 14 | Opera              | 1          |
| 15 | Pop                | 48         |

|    |                  |      |
|----|------------------|------|
| 16 | R&B/Soul         | 61   |
| 17 | Reggae           | 58   |
| 18 | Rock             | 1297 |
| 19 | Rock And Roll    | 12   |
| 20 | Sci Fi & Fantasy | 26   |
| 21 | Science Fiction  | 13   |
| 22 | Soundtrack       | 43   |
| 23 | TV Shows         | 93   |
| 24 | World            | 28   |

Ollama parameters:

model=llama3:latest,

options={},

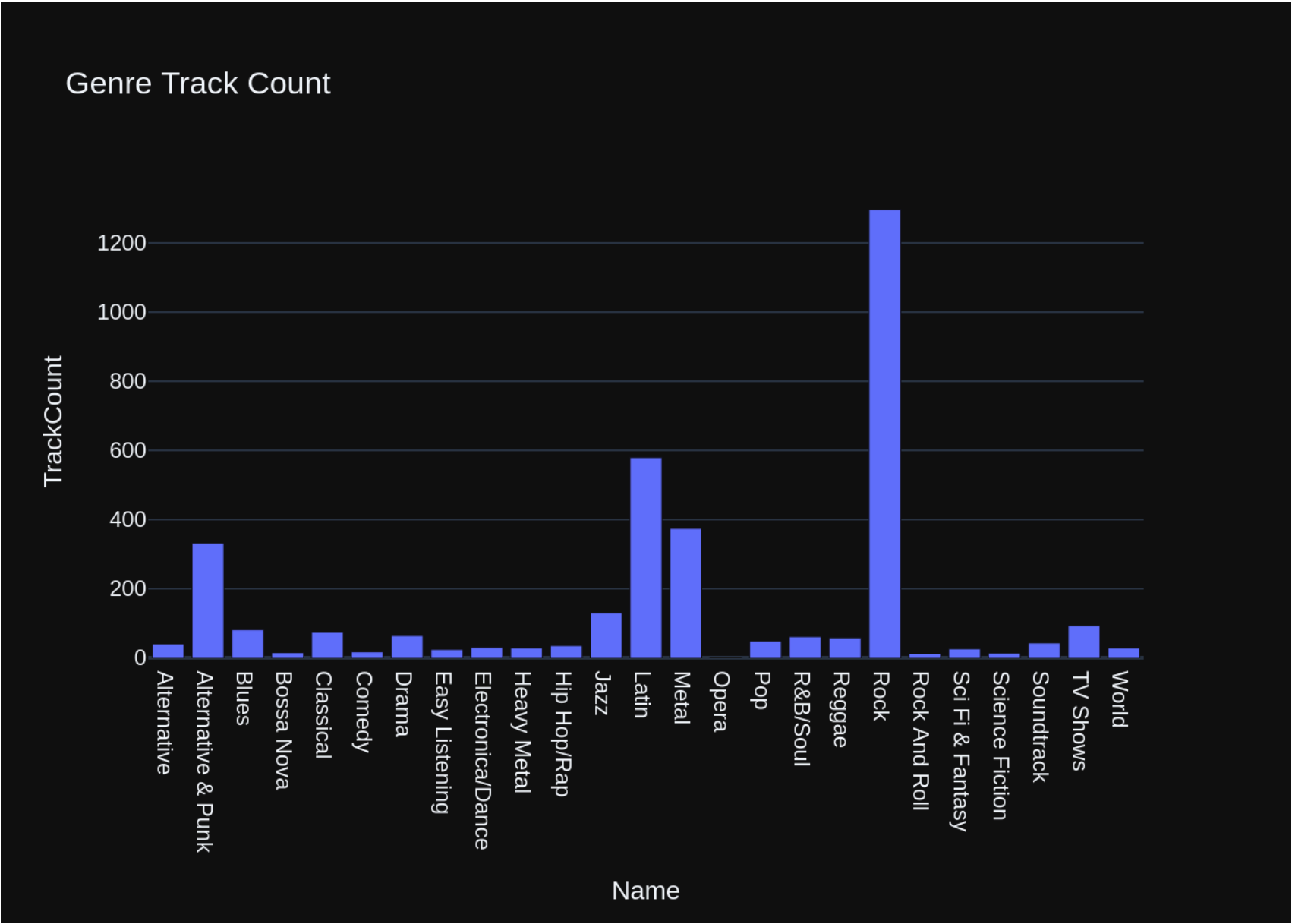
keep\_alive=None

Prompt Content:

```
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n List all genres and the number of tracks in each genre:\n'\n\nThe DataFrame was produced using this query: SELECT G.Name, COUNT(T.TrackId) AS TrackCount \nFROM Genre G \nLEFT JOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name object\nTrackCount in t64\nndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
```

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T20:03:43.058916498Z', 'message': {'role': 'assistant', 'content': "```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = px.bar(df, x='Name', y='TrackCount', title='Genre Track Count')\nfig.show()\n```", 'done_reason': 'stop', 'done': True, 'total_duration': 19095372873, 'load_duration': 2400732, 'prompt_eval_count': 203, 'prompt_eval_duration': 12071964000, 'eval_count': 44, 'eval_duration': 6960466000}
```



```
Out[20]: ('SELECT G.Name, COUNT(T.TrackId) AS TrackCount \nFROM Genre G \nLEFT JOIN Track T ON G.GenreId = T.GenreI
d \nGROUP BY G.Name',
```

|    | Name               | TrackCount |
|----|--------------------|------------|
| 0  | Alternative        | 40         |
| 1  | Alternative & Punk | 332        |
| 2  | Blues              | 81         |
| 3  | Bossa Nova         | 15         |
| 4  | Classical          | 74         |
| 5  | Comedy             | 17         |
| 6  | Drama              | 64         |
| 7  | Easy Listening     | 24         |
| 8  | Electronica/Dance  | 30         |
| 9  | Heavy Metal        | 28         |
| 10 | Hip Hop/Rap        | 35         |
| 11 | Jazz               | 130        |
| 12 | Latin              | 579        |
| 13 | Metal              | 374        |
| 14 | Opera              | 1          |
| 15 | Pop                | 48         |
| 16 | R&B/Soul           | 61         |
| 17 | Reggae             | 58         |
| 18 | Rock               | 1297       |
| 19 | Rock And Roll      | 12         |
| 20 | Sci Fi & Fantasy   | 26         |
| 21 | Science Fiction    | 13         |
| 22 | Soundtrack         | 43         |
| 23 | TV Shows           | 93         |
| 24 | World              | 28,        |

```
Figure({
 'data': [{ 'alignmentgroup': 'True',
 'hovertemplate': 'Name=%{x}
TrackCount=%{y}<extra></extra>',
 'legendgroup': '',
 'marker': { 'color': '#636efa', 'pattern': { 'shape': '' } },
 'name': '',
 'offsetgroup': '',
 'orientation': 'v',
 'showlegend': False,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['Alternative', 'Alternative & Punk', 'Blues', 'Bossa Nova', 'Classical',
 'Comedy', 'Drama', 'Easy Listening', 'Electronica/Dance', 'Heavy Metal',
 'Hip Hop/Rap', 'Jazz', 'Latin', 'Metal', 'Opera', 'Pop', 'R&B/Soul',
```

```

 'Reggae', 'Rock', 'Rock And Roll', 'Sci Fi & Fantasy',
 'Science Fiction', 'Soundtrack', 'TV Shows', 'World'], dtype=object),
 'xaxis': 'x',
 'y': array([40, 332, 81, 15, 74, 17, 64, 24, 30, 28, 35, 130,
 579, 374, 1, 48, 61, 58, 1297, 12, 26, 13, 43, 93,
 28]),
 'yaxis': 'y'}],
 'layout': {'barmode': 'relative',
 'legend': {'tracegroupgap': 0},
 'template': '...',
 'title': {'text': 'Genre Track Count'},
 'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
 'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TrackCount'}}}
))

```

```

In [21]: question = """
 Get all genres that do not have any tracks associated with them:
 """

 vn.ask(question=question)

```

Number of requested results 10 is greater than number of elements in index 8, updating n\_results = 8  
 Number of requested results 10 is greater than number of elements in index 1, updating n\_results = 1

```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX [IFK_TrackGenreId] ON [Track] ([GenreId])\n\nCREATE TABLE [Track]\n(\n [TrackId] INTEGER NOT NULL,\n [Name] NVARCHAR(200) NOT NULL,\n [AlbumId] INTEGER,\n [MediaTypeId] INTEGER NOT NULL,\n [GenreId] INTEGER,\n [Composer] NVARCHAR(220),\n [Milliseconds] INTEGER NOT NULL,\n [Bytes] INTEGER,\n [UnitPrice] NUMERIC(10,2) NOT NULL,\n CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_TrackMediaTypeId] ON [Track] ([MediaTypeId])\n\nCREATE INDEX [IFK_PlaylistTrackTrackId] ON [PlaylistTrack] ([TrackId])\n\nCREATE INDEX [IFK_TrackAlbumId] ON [Track] ([AlbumId])\n\nCREATE TABLE [Album]\n(\n [AlbumId] INTEGER NOT NULL,\n [Title] NVARCHAR(160) NOT NULL,\n [ArtistId] INTEGER NOT NULL,\n CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Genre]\n(\n [GenreId] INTEGER NOT NULL,\n [Name] NVARCHAR(120),\n CONSTRAINT [PK_Genre] PRIMARY KEY ([GenreId])\n)\n\nCREATE INDEX [IFK_AlbumArtistId] ON [Album] ([ArtistId])\n\nCREATE TABLE [PlaylistTrack]\n(\n [PlaylistId] INTEGER NOT NULL,\n [TrackId] INTEGER NOT NULL,\n CONSTRAINT [PK_PlaylistTrack] PRIMARY KEY ([PlaylistId], [TrackId]),\n FOREIGN KEY ([PlaylistId]) REFERENCES [Playlist] ([PlaylistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Artist]\n(\n [ArtistId] INTEGER NOT NULL,\n [Name] NVARCHAR(120),\n CONSTRAINT [PK_Artist] PRIMARY KEY ([ArtistId])\n)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}], {'role': 'user', 'content': ' \n List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT G.Name, COUNT(T.TrackId) AS TrackCount \nFROM Genre G \nLEFT JOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name'}, {'role': 'user', 'content': ' \n Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'",}, {'role': 'user', 'content': ' \n List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJOIN Artist ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': " SELECT * FROM t_person WHERE name = 'John Doe';"}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM Invoice'}, {'role': 'user', 'content': ' \n List all employees and their reporting manager's name (if any):\n'}, {'role': 'assistant', 'content': "SELECT E.LastName, E.FirstName, \n CASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (S
```

```

ELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END AS ReportingManager\nFROM Employee E"}, {'role': 'user', 'content': ' \n Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName'}, {'role': 'user', 'content': ' \n Get all genres that do not have any tracks associated with them:\n'}]

```

Ollama parameters:

model=llama3:latest,

options={},

keep\_alive=None

Prompt Content:

```

[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE INDEX [IFK_TrackGenreId] ON [Track] ([GenreId])\n\nCREATE TABLE [Track]\n(\n [TrackId] INTEGER NOT NULL,\n [Name] NVARCHAR(200) NOT NULL,\n [AlbumId] INTEGER,\n [MediaTypeId] INTEGER NOT NULL,\n [GenreId] INTEGER,\n [Composer] NVARCHAR(220),\n [Milliseconds] INTEGER NOT NULL,\n [Bytes] INTEGER,\n [UnitPrice] NUMERIC(10,2) NOT NULL,\n CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_TrackMediaTypeId] ON [Track] ([MediaTypeId])\n\nCREATE INDEX [IFK_PlaylistTrackTrackId] ON [PlaylistTrack] ([TrackId])\n\nCREATE INDEX [IFK_TrackAlbumId] ON [Track] ([AlbumId])\n\nCREATE TABLE [Album]\n(\n [AlbumId] INTEGER NOT NULL,\n [Title] NVARCHAR(160) NOT NULL,\n [ArtistId] INTEGER NOT NULL,\n CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Genre]\n(\n [GenreId] INTEGER NOT NULL,\n [Name] NVARCHAR(120),\n CONSTRAINT [PK_Genre] PRIMARY KEY ([GenreId])\n)\n\nCREATE INDEX [IFK_AlbumArtistId] ON [Album] ([ArtistId])\n\nCREATE TABLE [PlaylistTrack]\n(\n [PlaylistId] INTEGER NOT NULL,\n [TrackId] INTEGER NOT NULL,\n CONSTRAINT [PK_PlaylistTrack] PRIMARY KEY ([PlaylistId], [TrackId]),\n FOREIGN KEY ([PlaylistId]) REFERENCES [Playlist] ([PlaylistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Artist]\n(\n [ArtistId] INTEGER NOT NULL,\n [Name] NVARCHAR(120),\n CONSTRAINT [PK_Artist] PRIMARY KEY ([ArtistId])\n)\n\n\n===Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines\n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql\n3. If the provided context is insufficient, please explain why it can't be generated.\n4. Please use the most relevant table(s).\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before.\n"}, {"role": "user", "content": " \n List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT G.Name, COUNT(T.TrackId) AS TrackCount\nFROM Genre G\nLEFT JOIN Track T ON G.GenreId = T.GenreId\nGROUP BY G.Name"}, {"role": "user", "content": " \n Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "a

```



```
ssistant", "content": "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'", {"role": "user", "content": "
\n List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELE
CT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJOIN Artist ART ON A.ArtistId = ART.ArtistId"}, {"role":
"user", "content": " SELECT * FROM t_person WHERE name = 'John Doe';"}, {"role": "assistant", "content": "S
ELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content": "what are the top 5 countries
that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS Count\nFROM Cust
omer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all invoic
es with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM Invoice"}, {"role":
"user", "content": " \n List all employees and their reporting manager's name (if any):\n"}, {"role":
"assistant", "content": "SELECT E.LastName, E.FirstName, \n CASE WHEN ReportsTo IS NULL THEN 'N/A' EL
SE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END AS ReportingManage
r\nFROM Employee E"}, {"role": "user", "content": " \n Get the average invoice total for each custome
r:\n"}, {"role": "assistant", "content": "SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal\nFROM Cust
omer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName"}, {"role": "user", "content": "
\n Get all genres that do not have any tracks associated with them:\n"}]
```

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T20:05:11.156203217Z', 'message': {'role': 'assistant',
'content': '```\nSELECT G.Name\nFROM Genre G\nLEFT JOIN Track T ON G.GenreId = T.GenreId\nWHERE T.TrackId I
S NULL;\n`'}, 'done_reason': 'stop', 'done': True, 'total_duration': 87894622048, 'load_duration': 137208
1, 'prompt_eval_count': 1316, 'prompt_eval_duration': 81740551000, 'eval_count': 34, 'eval_duration': 56049
09000}
```
```

```
SELECT G.Name
FROM Genre G
LEFT JOIN Track T ON G.GenreId = T.GenreId
WHERE T.TrackId IS NULL;
```
```

Output from LLM: ```

```
SELECT G.Name
FROM Genre G
LEFT JOIN Track T ON G.GenreId = T.GenreId
WHERE T.TrackId IS NULL;
```
```

```
Extracted SQL: SELECT G.Name
FROM Genre G
LEFT JOIN Track T ON G.GenreId = T.GenreId
WHERE T.TrackId IS NULL
SELECT G.Name
FROM Genre G
LEFT JOIN Track T ON G.GenreId = T.GenreId
WHERE T.TrackId IS NULL
Empty DataFrame
```

Columns: [Name]

Index: []

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n    Get all genres that do not have any tracks associated with them:\n\n\nThe DataFrame was produced using this query: SELECT G.Name\nFROM Genre G\nLEFT JOIN Track T\nON G.GenreId = T.GenreId\nWHERE T.TrackId IS NULL\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name      object\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
```

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T20:05:29.738235683Z', 'message': {'role': 'assistant', 'content': "\n\nimport plotly.express as px\n\nfig = px.bar(df, x='Name', y='', title='Genres without Tracks')\nfig.update_layout(xaxis_title='Genre Name', yaxis_title='')\nfig.show()\n\n"}, 'done_reason': 'stop', 'done': True, 'total_duration': 18579805112, 'load_duration': 560733, 'prompt_eval_count': 186, 'prompt_eval_duration': 10877176000, 'eval_count': 48, 'eval_duration': 7609717000}
```



```
Out[21]: ('SELECT G.Name\nFROM Genre G\nLEFT JOIN Track T ON G.GenreId = T.GenreId\nWHERE T.TrackId IS NULL',
Empty DataFrame
Columns: [Name]
Index: [],
Figure({
  'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
            'hovernplate': 'Name=%{label}<extra></extra>',
            'labels': array([], dtype=object),
            'legendgroup': '',
            'name': '',
            'showlegend': True,
            'type': 'pie'}],
  'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'template': '...'}
}))
```

```
In [22]: question = """
        List all customers who have not placed any orders:
        """

        vn.ask(question=question)
```

Number of requested results 10 is greater than number of elements in index 8, updating n_results = 8
 Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1

```
{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE TABLE IF NOT EXISTS t_person (\n    id INT PRIMARY KEY,\n    name VARCHAR(100),\n    email text,\n    age INT\n)\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n    FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [PlaylistTrack]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    CONSTRAINT [PK_PlaylistTrack] PRIMARY KEY ([PlaylistId], [TrackId]),\n    FOREIGN KEY ([PlaylistId]) REFERENCES [Playlist] ([PlaylistId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE TABLE [Album]\n(\n    [AlbumId] INTEGER NOT NULL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER NOT NULL,\n    CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(220),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n\n===Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines\n\n1. If the provided context is sufficient, please generate a valid SQL query without a
```

ny explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName'}, {'role': 'user', 'content': ' \n List all invoices with a total exceeding \$10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM Invoice'}, {'role': 'user', 'content': " \n List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': "SELECT E.LastName, E.FirstName, \n CASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END AS ReportingManager\nFROM Employee E"}, {'role': 'user', 'content': ' \n List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A\nJOIN Artist ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': ' \n List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT G.Name, COUNT(T.TrackId) AS TrackCount \nFROM Genre G\nLEFT JOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name'}, {'role': 'user', 'content': " SELECT * FROM t_person WHERE name = 'John Doe';"}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': ' \n Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': ' \n List all customers who have not placed any orders:\n'}]

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n)\n\nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n)\n\nCREATE
```

```
ny):\n"}}, {"role": "assistant", "content": "SELECT E.LastName, E.FirstName, \n      CASE WHEN ReportsTo IS
```

```

NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END
AS ReportingManager\nFROM Employee E"}, {"role": "user", "content": " \n    List all albums and their corr
esponding artist names \n"}, {"role": "assistant", "content": "SELECT A.Title, A.ArtistId, ART.Name \nFROM
Album A \nJOIN Artist ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": " \n    List all gen
res and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT G.Name, COUNT(T.T
rackId) AS TrackCount \nFROM Genre G \nLEFT JOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name"}, {"ro
le": "user", "content": " SELECT * FROM t_person WHERE name = 'John Doe';"}, {"role": "assistant", "conten
t": "SELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content": " \n    Find all tracks
with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT *\nFROM Tr
ack\nWHERE LOWER(Name) LIKE '%what%'", {"role": "user", "content": " \n    List all customers who have no
t placed any orders:\n"]}

```

Ollama Response:

```

{'model': 'llama3:latest', 'created_at': '2024-06-08T20:07:47.299622439Z', 'message': {'role': 'assistant',
'content': "There is no direct table relationship between Customer and Invoice. To find the list of custome
rs who have not placed any orders, you would need to use a LEFT JOIN with a subquery or a derived table.\n
\nHere's an example using a subquery:\n\n```\nSELECT C.*\nFROM Customer C\nLEFT JOIN (\n    SELECT Customer
Id\n    FROM Invoice\n) I ON C.CustomerId = I.CustomerId\nWHERE I.CustomerId IS NULL;\n```"}, 'done_reaso
n': 'stop', 'done': True, 'total_duration': 137448644620, 'load_duration': 609255, 'prompt_eval_count': 191
1, 'prompt_eval_duration': 121184695000, 'eval_count': 91, 'eval_duration': 15687985000}

```

There is no direct table relationship between Customer and Invoice. To find the list of customers who have not placed any orders, you would need to use a LEFT JOIN with a subquery or a derived table.

Here's an example using a subquery:

```

```

```

```

SELECT C.*
FROM Customer C
LEFT JOIN (
 SELECT CustomerId
 FROM Invoice
) I ON C.CustomerId = I.CustomerId
WHERE I.CustomerId IS NULL;
```

```

Output from LLM: There is no direct table relationship between Customer and Invoice. To find the list of customers who have not placed any orders, you would need to use a LEFT JOIN with a subquery or a derived table.

Here's an example using a subquery:

```

```

```

```

SELECT C.*
FROM Customer C

```



```

LEFT JOIN (
 SELECT CustomerId
 FROM Invoice
) I ON C.CustomerId = I.CustomerId
WHERE I.CustomerId IS NULL;
```

```

```

Extracted SQL: SELECT C.*
FROM Customer C
LEFT JOIN (
    SELECT CustomerId
    FROM Invoice
) I ON C.CustomerId = I.CustomerId
WHERE I.CustomerId IS NULL
SELECT C.*
FROM Customer C
LEFT JOIN (
    SELECT CustomerId
    FROM Invoice
) I ON C.CustomerId = I.CustomerId
WHERE I.CustomerId IS NULL

```

Empty DataFrame

Columns: [CustomerId, FirstName, LastName, Company, Address, City, State, Country, PostalCode, Phone, Fax, Email, SupportRepId]

Index: []

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```

[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n    List all customers who have not placed any orders:\n'\n\nThe DataFrame was produced using this query: SELECT C.*\nFROM Customer C\nLEFT JOIN (\n    SELECT CustomerId\n    FROM Invoice\n) I ON C.CustomerId = I.CustomerId\nWHERE I.CustomerId IS NULL\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\nCustomerId      object\nFirstName      object\nLastName      object\nCompany      object\nAddress      object\nCity      object\nState      object\nCountry      object\nPostalCode      object\nPhone      object\nFax      object\nEmail      object\nSupportRepId      object\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]

```


Ollama Response:

```

{'model': 'llama3:latest', 'created_at': '2024-06-08T20:08:07.353586774Z', 'message': {'role': 'assistant',

```

```
'content': "```\nimport plotly.express as px\nimport pandas as pd\n\nfig = px.bar(df, x='ColumnName', y='ValueColumn')\nfig.show()\n```", 'done_reason': 'stop', 'done': True, 'total_duration': 20051339110, 'load_duration': 613645, 'prompt_eval_count': 243, 'prompt_eval_duration': 14384942000, 'eval_count': 35, 'eval_duration': 5518162000}
```



```

Out[22]: ('SELECT C.*\nFROM Customer C\nLEFT JOIN (\n    SELECT CustomerId\n    FROM Invoice\n) I ON C.CustomerId =
I.CustomerId\nWHERE I.CustomerId IS NULL',
Empty DataFrame
Columns: [CustomerId, FirstName, LastName, Company, Address, City, State, Country, PostalCode, Phone, Fa
x, Email, SupportRepId]
Index: [],
Figure({
  'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
    'hovertemplate': 'CustomerId=%{label}<extra></extra>',
    'labels': array([], dtype=object),
    'legendgroup': '',
    'name': '',
    'showlegend': True,
    'type': 'pie'}],
  'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'template': '...'}
}))

```

```

In [23]: question = """
        Get the top 10 most popular artists (based on the number of tracks):
        """

        vn.ask(question=question)

```

Number of requested results 10 is greater than number of elements in index 8, updating n_results = 8
 Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1

```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE INDEX [IFK_AlbumArtistId] ON [Album] ([ArtistId])\n\nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(220),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_TrackAlbumId] ON [Track] ([AlbumId])\n\nCREATE INDEX [IFK_TrackGenreId] ON [Track] ([GenreId])\n\nCREATE TABLE [Artist]\n(\n    [ArtistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Artist] PRIMARY KEY ([ArtistId])\n)\n\nCREATE TABLE [Album]\n(\n    [AlbumId] INTEGER NOT NULL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER NOT NULL,\n    CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_PlaylistTrackTrackId] ON [PlaylistTrack] ([TrackId])\n\nCREATE INDEX [IFK_TrackMediaTypeId] ON [Track] ([MediaTypeId])\n\nCREATE TABLE [Genre]\n(\n    [GenreId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Genre] PRIMARY KEY ([GenreId])\n)\n\nCREATE TABLE [Playlist]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Playlist] PRIMARY KEY ([PlaylistId])\n)\n\n===Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n==Response Guidelines\n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql\n\n3. If the provided context is insufficient, please explain why it can't be generated.\n\n4. Please use the most relevant table(s).\n\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before.\n"}], {'role': 'user', 'content': ' \n    List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT G.Name, COUNT(T.TrackId) AS TrackCount\nFROM Genre G\nLEFT JOIN Track T ON G.GenreId = T.GenreId\nGROUP BY G.Name'}, {'role': 'user', 'content': ' \n    List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT A.Title, A.ArtistId, ART.Name\nFROM Album A\nJOIN Artist ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n    Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': ' \n    Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName'}, {'role': 'user', 'content': ' \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT *\nFROM Invoice'}, {'role': 'user', 'content': " SELECT * FROM t_person WHERE name = 'John Doe';"}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': " \n    List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': "SELECT E.LastName, E.FirstName, \n    CASE WHEN ReportsTo IS NOT NULL THEN M.LastName ELSE E.LastName END AS ManagerName\nFROM Employee E\nLEFT JOIN Employee M ON E.ReportsTo = M.EmployeeId"}]
```

```
S NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) EN
D AS ReportingManager\nFROM Employee E"}, {"role": "user", "content": " \n    Get the top 10 most popular
artists (based on the number of tracks):\n"}]
```

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables\nCREATE INDEX [IFK_AlbumArtistId] ON [Album] ([ArtistId])\n\nCREATE TABLE
[Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(220),\n    [Milliseconds]
INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track]
PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_TrackAlbumId] ON [Track] ([AlbumId])\n\nCREA
TE INDEX [IFK_TrackGenreId] ON [Track] ([GenreId])\n\nCREATE TABLE [Artist]\n(\n    [ArtistId] INTEGER NOT
NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Artist] PRIMARY KEY ([ArtistId])\n)\n\nCREATE TABLE
[Album]\n(\n    [AlbumId] INTEGER NOT NULL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER
NOT NULL,\n    CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Ar
tist] ([ArtistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_PlaylistTrackTrack
Id] ON [PlaylistTrack] ([TrackId])\n\nCREATE INDEX [IFK_TrackMediaTypeId] ON [Track] ([MediaTypeId])\n\nCRE
ATE TABLE [Genre]\n(\n    [GenreId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Genr
e] PRIMARY KEY ([GenreId])\n)\n\nCREATE TABLE [Playlist]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [Nam
e] NVARCHAR(120),\n    CONSTRAINT [PK_Playlist] PRIMARY KEY ([PlaylistId])\n)\n\n\n===Additional Context
\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n=
==Response Guidelines\n1. If the provided context is sufficient, please generate a valid SQL query without
any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge
of a specific string in a particular column, please generate an intermediate SQL query to find the distinct
strings in that column. Prepend the query with a comment saying intermediate_sql\n3. If the provided conte
xt is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s).
\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given b
efore. \n"}, {"role": "user", "content": " \n    List all genres and the number of tracks in each genr
e:\n"}, {"role": "assistant", "content": "SELECT G.Name, COUNT(T.TrackId) AS TrackCount\nFROM Genre G\nLE
FT JOIN Track T ON G.GenreId = T.GenreId\nGROUP BY G.Name"}, {"role": "user", "content": " \n    List all
albums and their corresponding artist names\n"}, {"role": "assistant", "content": "SELECT A.Title, A.Arti
stId, ART.Name\nFROM Album A\nJOIN Artist ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content":
"what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Countr
y, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5"}, {"role": "user", "co
ntent": " \n    Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistan
```

```
t", "content": "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'", {"role": "user", "content": " \n
Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT C.LastName,
AVG(I.Total) AS AverageInvoiceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP
BY C.LastName"}, {"role": "user", "content": " \n    List all invoices with a total exceeding $10:\n"},
{"role": "assistant", "content": "SELECT * \nFROM Invoice"}, {"role": "user", "content": " SELECT * FROM t_
person WHERE name = 'John Doe';"}, {"role": "assistant", "content": "SELECT * FROM t_person WHERE name = 'J
ohn Doe'"}, {"role": "user", "content": " \n    List all employees and their reporting manager's name (if
any):\n"}, {"role": "assistant", "content": "SELECT E.LastName, E.FirstName, \n        CASE WHEN ReportsTo I
S NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) EN
D AS ReportingManager\nFROM Employee E"}, {"role": "user", "content": " \n    Get the top 10 most popular
artists (based on the number of tracks):\n"}]
```

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T20:09:33.051446083Z', 'message': {'role': 'assistant',
'content': 'SELECT A.Name, COUNT(T.TrackId) AS TrackCount\nFROM Artist A\nJOIN Track T ON A.ArtistId = T.Ar
tistId\nGROUP BY A.Name\nORDER BY TrackCount DESC\nLIMIT 10'}, 'done_reason': 'stop', 'done': True, 'total_
duration': 85612308947, 'load_duration': 606837, 'prompt_eval_count': 1251, 'prompt_eval_duration': 7740830
9000, 'eval_count': 46, 'eval_duration': 7631172000}
```

```
SELECT A.Name, COUNT(T.TrackId) AS TrackCount
FROM Artist A
JOIN Track T ON A.ArtistId = T.ArtistId
GROUP BY A.Name
ORDER BY TrackCount DESC
LIMIT 10
SELECT A.Name, COUNT(T.TrackId) AS TrackCount
FROM Artist A
JOIN Track T ON A.ArtistId = T.ArtistId
GROUP BY A.Name
ORDER BY TrackCount DESC
LIMIT 10
```

```
Couldn't run sql: Execution failed on sql 'SELECT A.Name, COUNT(T.TrackId) AS TrackCount
FROM Artist A
JOIN Track T ON A.ArtistId = T.ArtistId
GROUP BY A.Name
ORDER BY TrackCount DESC
LIMIT 10': no such column: T.ArtistId
```

```
In [24]: question = """
        List all customers from Canada and their email addresses:
        """

vn.ask(question=question)
```

```
Number of requested results 10 is greater than number of elements in index 8, updating n_results = 8  
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE IF NOT EXISTS t_person (\n    id INT PRIMARY KEY,\n    name VARCHAR(100),\n    email text,\n    age INT\n)\n\nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n    FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineInvoiceId] ON [InvoiceLine] ([InvoiceId])\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE INDEX [IFK_EmployeeReportsTo] ON [Employee] ([ReportsTo])\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines\n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql\n3. If the provided context is insufficient, please explain why it can't be generated.\n4. Please use the most relevant table(s).\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before.\n\n\"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5'}, {'role': 'user', 'content': 'List all invoices with a total exceeding $10:'}, {'role': 'assistant', 'content': 'SELECT * \nFROM Invoice'}, {'role': 'user', 'content': 'Get the average invoice total for each customer:'}, {'role': 'assistant', 'content': 'SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName'}, {'role': 'user',
```



```
'content': " \n      List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': "SELECT E.LastName, E.FirstName, \n      CASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END AS ReportingManager\nFROM Employee E"}, {'role': 'user', 'content': ' \n      List all albums and their corresponding artist names\n'}, {'role': 'assistant', 'content': 'SELECT A.Title, A.ArtistId, ART.Name\nFROM Album A\nJOIN Artist ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': " SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': ' \n      List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT G.Name, COUNT(T.TrackId) AS TrackCount\nFROM Genre G\nLEFT JOIN Track T ON G.GenreId = T.GenreId\nGROUP BY G.Name'}, {'role': 'user', 'content': ' \n      Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': ' \n      List all customers from Canada and their email addresses:\n'}]
```

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId])\n)\n\nCREATE TABLE IF NOT EXISTS t_person (\n    id INT PRIMARY KEY,\n    name VARCHAR(100),\n    email text,\n    age INT\n)\n\nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId])\n)\n\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n    FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId])\n)\n\nCREATE INDEX [IFK_InvoiceLineInvoiceId] ON [InvoiceLine] ([InvoiceId])\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE INDEX [IFK_EmployeeReportsTo] ON [Employee] ([ReportsTo])\n\nCREATE TABLE [InvoiceLine]
```

```
e)\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5"}, {"role": "user", "content": " \n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM Invoice"}, {"role": "user", "content": " \n    Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName"}, {"role": "user", "content": " \n    List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT E.LastName, E.FirstName, \n    CASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END AS ReportingManager\nFROM Employee E"}, {"role": "user", "content": " \n    List all albums and their corresponding artist names\n"}, {"role": "assistant", "content": "SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJOIN Artist ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": " SELECT * FROM t_person WHERE name = 'John Doe';"}, {"role": "assistant", "content": "SELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content": " \n    List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT G.Name, COUNT(T.TrackId) AS TrackCount \nFROM Genre G \nLEFT JOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name"}, {"role": "user", "content": " \n    Find all tracks with a name containing 'What' (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": " \n    List all customers from Canada and their email addresses:\n"}]
```

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T20:11:11.509852558Z', 'message': {'role': 'assistant', 'content': "```\nSELECT Email, City, Country\nFROM Customer\nWHERE Country = 'Canada'\n```\n"}, 'done_reason': 'stop', 'done': True, 'total_duration': 98388885009, 'load_duration': 783272, 'prompt_eval_count': 1507, 'prompt_eval_duration': 94543280000, 'eval_count': 20, 'eval_duration': 3261709000}\n```\n
```

```
SELECT Email, City, Country
FROM Customer
WHERE Country = 'Canada'
```
```

Output from LLM: ```

```
SELECT Email, City, Country
FROM Customer
WHERE Country = 'Canada'
```

```

```
Extracted SQL: SELECT Email, City, Country
FROM Customer
WHERE Country = 'Canada'
```

```
SELECT Email, City, Country
FROM Customer
WHERE Country = 'Canada'
```

	Email	City	Country
0	ftremblay@gmail.com	Montréal	Canada
1	mphilips12@shaw.ca	Edmonton	Canada
2	jenniferp@rogers.ca	Vancouver	Canada
3	robbrown@shaw.ca	Toronto	Canada
4	edfrancis@yachoo.ca	Ottawa	Canada
5	marthasilk@gmail.com	Halifax	Canada
6	aaronmitchell@yahoo.ca	Winnipeg	Canada
7	ellie.sullivan@shaw.ca	Yellowknife	Canada

Ollama parameters:

model=llama3:latest,

options={},

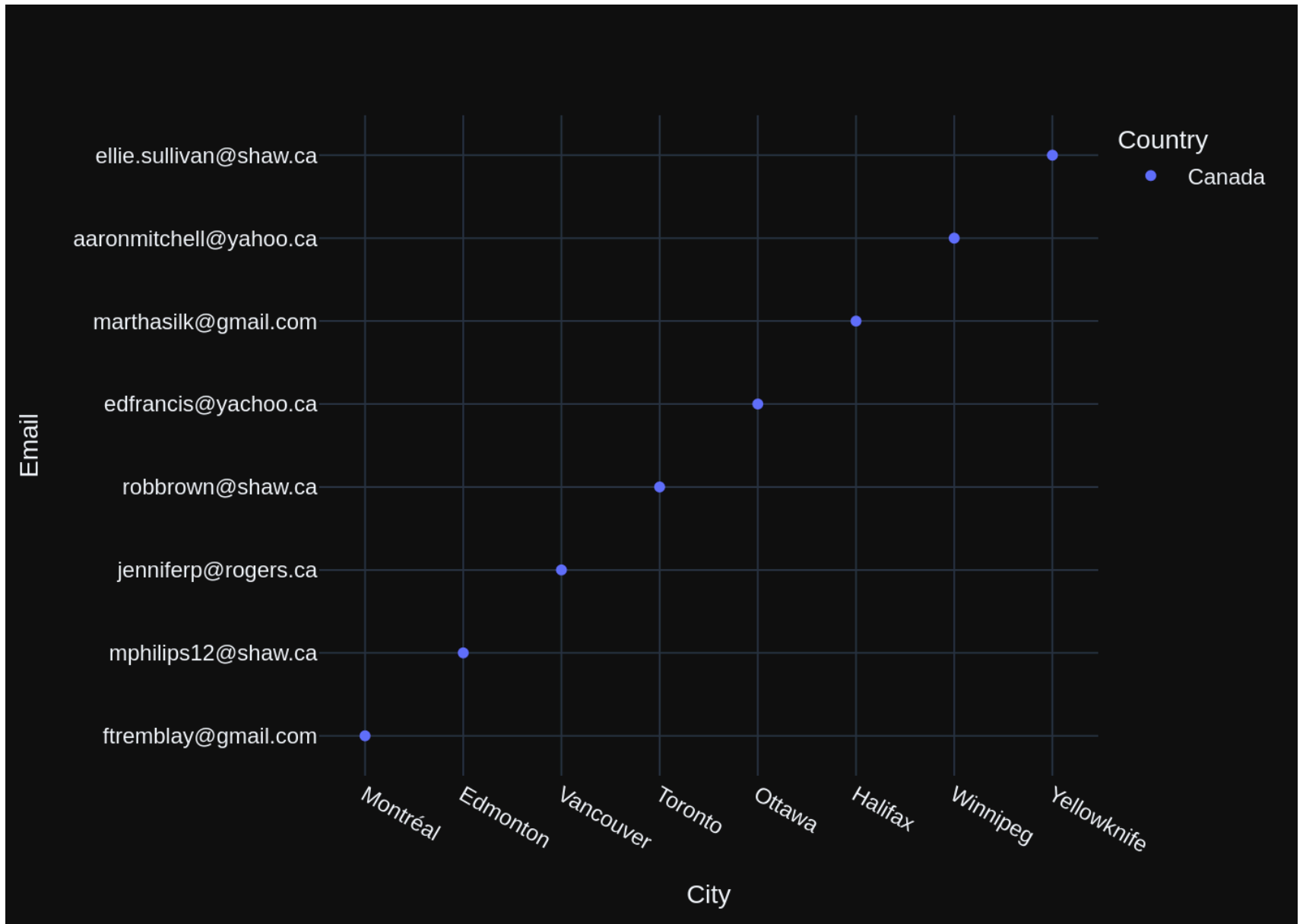
keep_alive=None

Prompt Content:

```
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n      List all customers from Canada and their email addresses:\n'\n\nThe DataFrame was produced using this query: SELECT Email, City, Country\nFROM Customer\nWHERE Country = 'Canada'\n\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\nEmail      object\nCity       object\nCountry    object\nndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
```

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T20:11:36.620380381Z', 'message': {'role': 'assistant', 'content': '\n\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nif len(df) == 1:\n    fig = go.Figure(data=[go.Indicator(\n        mode="number+delta",\n        value=df[\'Email\'].values[0],\n        title="Customer Email"\n    )])\nelse:\n    fig = px.scatter(df, x=\'City\', y=\'Email\', color=\'Country\')\nfig.show()\n\n'}, 'done_reason': 'stop', 'done': True, 'total_duration': 24944848833, 'load_duration': 2520797, 'prompt_eval_count': 177, 'prompt_eval_duration': 10422980000, 'eval_count': 90, 'eval_duration': 14412902000}
```



```
Out[24]: ("SELECT Email, City, Country\nFROM Customer\nWHERE Country = 'Canada'\n",
          Email      City Country
0    ftremblay@gmail.com    Montréal Canada
1    mphilips12@shaw.ca    Edmonton Canada
2    jenniferp@rogers.ca    Vancouver Canada
3    robbrown@shaw.ca    Toronto Canada
4    edfrancis@yachoo.ca    Ottawa Canada
5    marthasilk@gmail.com    Halifax Canada
6    aaronmitchell@yahoo.ca    Winnipeg Canada
7    ellie.sullivan@shaw.ca    Yellowknife Canada,
Figure({
  'data': [{'hovertemplate': 'Country=Canada<br>City=%{x}<br>Email=%{y}<extra></extra>',
    'legendgroup': 'Canada',
    'marker': {'color': '#636efa', 'symbol': 'circle'},
    'mode': 'markers',
    'name': 'Canada',
    'orientation': 'v',
    'showlegend': True,
    'type': 'scatter',
    'x': array(['Montréal', 'Edmonton', 'Vancouver', 'Toronto', 'Ottawa', 'Halifax',
      'Winnipeg', 'Yellowknife'], dtype=object),
    'xaxis': 'x',
    'y': array(['ftremblay@gmail.com', 'mphilips12@shaw.ca', 'jenniferp@rogers.ca',
      'robbrown@shaw.ca', 'edfrancis@yachoo.ca', 'marthasilk@gmail.com',
      'aaronmitchell@yahoo.ca', 'ellie.sullivan@shaw.ca'], dtype=object),
    'yaxis': 'y'}],
  'layout': {'legend': {'title': {'text': 'Country'}}, 'tracegroupgap': 0},
    'margin': {'t': 60},
    'template': '...',
    'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'City'}},
    'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Email'}}}
  )))
```

```
In [25]: question = """
          Find the customer with the most invoices
          """

          vn.ask(question=question)
```

Number of requested results 10 is greater than number of elements in index 9, updating n_results = 9
 Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1

```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineInvoiceId] ON [InvoiceLine] ([InvoiceId])\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n    FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE IF NOT EXISTS t_person (\n    id INT PRIMARY KEY,\n    name VARCHAR(100),\n    email text,\n    age INT\n)\n\nCREATE INDEX [IFK_EmployeeReportsTo] ON [Employee] ([ReportsTo])\n\n\n===Additional Context\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n\n===Response Guidelines\n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql\n3. If the provided context is insufficient, please explain why it can't be generated.\n4. Please use the most relevant table(s).\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before.\n"}], {'role': 'user', 'content': '\n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM Invoice'}, {'role': 'user', 'content': '\n    Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5'}, {'role': 'use
```

```

r', 'content': ' \n      List all customers from Canada and their email addresses:\n'}, {'role': 'assistant', 'content': "SELECT Email, City, Country\nFROM Customer\nWHERE Country = 'Canada'\n"}, {'role': 'user', 'content': " \n      List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': "SELECT E.LastName, E.FirstName, \n      CASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END AS ReportingManager\nFROM Employee E"}, {'role': 'user', 'content': ' \n      List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT G.Name, COUNT(T.TrackId) AS TrackCount\nFROM Genre G\nLEFT JOIN Track T ON G.GenreId = T.GenreId\nGROUP BY G.Name'}, {'role': 'user', 'content': " SELECT * FROM t_person WHERE name = 'John Doe';"}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': ' \n      List all albums and their corresponding artist names\n'}, {'role': 'assistant', 'content': 'SELECT A.Title, A.ArtistId, ART.Name\nFROM Album A\nJOIN Artist ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': ' \n      Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': ' \n      Find the customer with the most invoices\n'}]

```

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```

[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineInvoiceId] ON [InvoiceLine] ([InvoiceId])\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHA

```

```

R(70),\n      [City] NVARCHAR(40),\n      [State] NVARCHAR(40),\n      [Country] NVARCHAR(40),\n      [PostalCode] NVARCHAR(10),\n      [Phone] NVARCHAR(24),\n      [Fax] NVARCHAR(24),\n      [Email] NVARCHAR(60),\n      CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n      FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId])\n    )\n  )\n  ON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n\nCREATE TABLE IF NOT EXISTS t_person (\n  id INT PRIMARY KEY,\n  name VARCHAR(100),\n  email text,\n  age INT\n)\n\n\nCREATE INDEX [IFK_EmployeeReportsTo] ON [Employee] ([ReportsTo])\n\n\n\n===Additional Context\n\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n\n===Response Guidelines\n\n\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql\n\n3. If the provided context is insufficient, please explain why it can't be generated.\n\n4. Please use the most relevant table(s).\n\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before.\n\n"}\n\n{"role": "user", "content": "\n      List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM Invoice"}, {"role": "user", "content": "\n      Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5"}, {"role": "user", "content": "\n      List all customers from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SELECT Email, City, Country\nFROM Customer\nWHERE Country = 'Canada'\n"}, {"role": "user", "content": "\n      List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT E.LastName, E.FirstName, \n      CASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END AS ReportingManager\nFROM Employee E"}, {"role": "user", "content": "\n      List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT G.Name, COUNT(T.TrackId) AS TrackCount\nFROM Genre G\nLEFT JOIN Track T ON G.GenreId = T.GenreId\nGROUP BY G.Name"}, {"role": "user", "content": "SELECT * FROM t_person WHERE name = 'John Doe';"}, {"role": "assistant", "content": "SELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content": "\n      List all albums and their corresponding artist names\n"}, {"role": "assistant", "content": "SELECT A.Title, A.ArtistId, ART.Name\nFROM Album A\nJOIN Artist ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": "\n      Find all tracks with a name containing 'What' (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'\n"}, {"role": "user", "content": "\n      Find the customer with the most invoices\n"}]

```

Ollama Response:

```

{'model': 'llama3:latest', 'created_at': '2024-06-08T20:13:27.094971486Z', 'message': {'role': 'assistant', 'content': '\n\nSELECT C.LastName, COUNT(I.InvoiceId) AS TotalInvoices\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.CustomerId, C.LastName\nORDER BY TotalInvoices DESC\nLIMIT 1;\n\n'}, 'done_reason': 'stop', 'done': True, 'total_duration': 110317939603, 'load_duration': 1156886, 'prompt_eval_count': 1596, 'prompt_eval_duration': 100237393000, 'eval_count': 56, 'eval_duration': 9458827000}

```

```

SELECT C.LastName, COUNT(I.InvoiceId) AS TotalInvoices
FROM Customer C

```



```
JOIN Invoice I ON C.CustomerId = I.CustomerId
GROUP BY C.CustomerId, C.LastName
ORDER BY TotalInvoices DESC
LIMIT 1;
```

```

Output from LLM: ```

```
SELECT C.LastName, COUNT(I.InvoiceId) AS TotalInvoices
FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
GROUP BY C.CustomerId, C.LastName
ORDER BY TotalInvoices DESC
LIMIT 1;
```

```

```
Extracted SQL: SELECT C.LastName, COUNT(I.InvoiceId) AS TotalInvoices
FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
GROUP BY C.CustomerId, C.LastName
ORDER BY TotalInvoices DESC
LIMIT 1
SELECT C.LastName, COUNT(I.InvoiceId) AS TotalInvoices
FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
GROUP BY C.CustomerId, C.LastName
ORDER BY TotalInvoices DESC
LIMIT 1
```

```
    LastName  TotalInvoices
0  Gonçalves              7
```

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

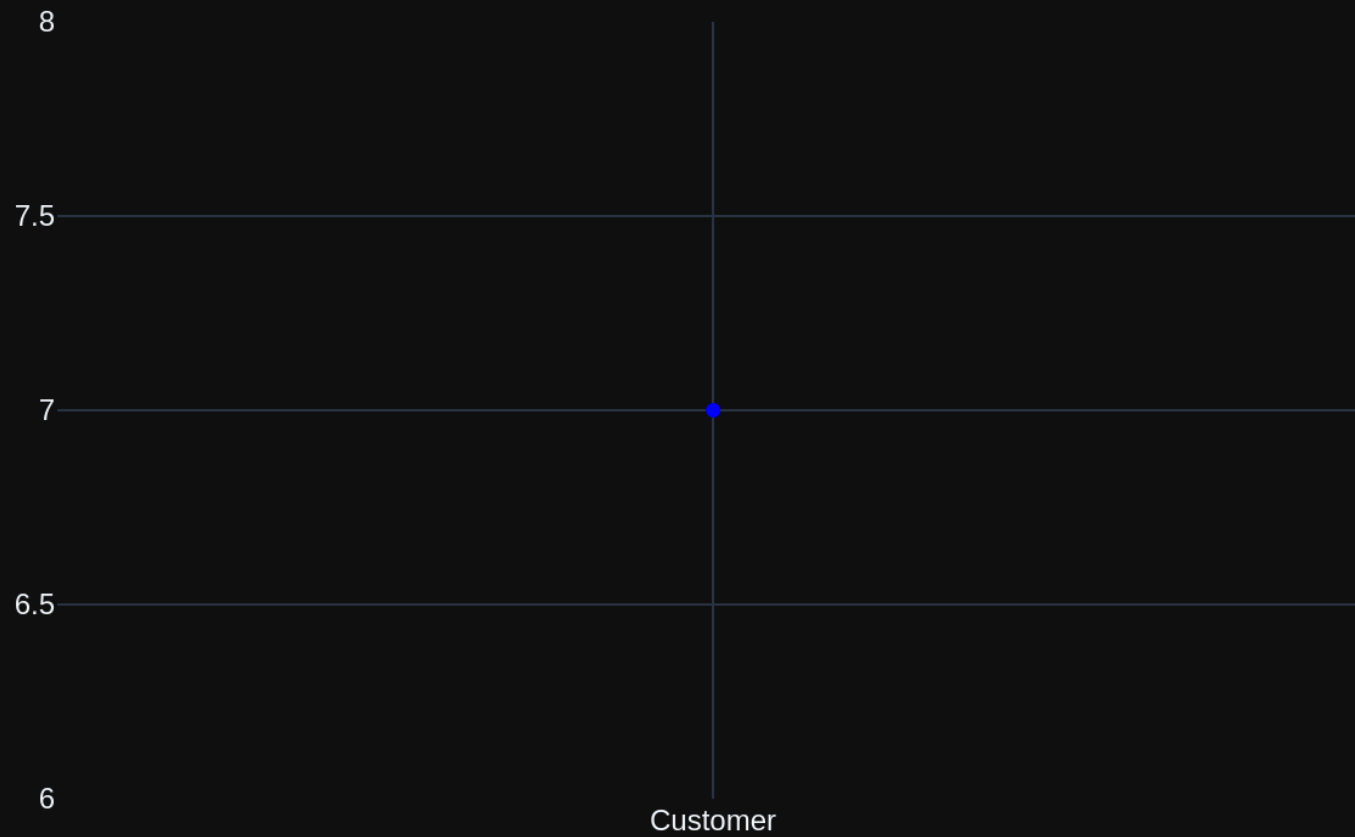
Prompt Content:

```
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n      Find the customer with the most invoices \n'\n\nThe DataFrame was produced using this query: SELECT C.LastName, COUNT(I.InvoiceId) AS TotalInvoices\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.CustomerId, C.LastName\nORDER BY TotalInvoices DESC\nLIMIT 1\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n LastName      object\nTotalInvoices  int64\nndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
```

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T20:13:58.532559292Z', 'message': {'role': 'assistant',  
'content': "```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = go.Figure()\n\nif  
df.shape[0] == 1:\n    fig.add_scatter(x=['Customer'], y=[df['TotalInvoices'].values[0]], mode='markers', m  
arker=dict(color='blue'))\n    fig.update_layout(title='Most Invoices by Customer')\nelse:\n    fig.add_bar  
(x=df['LastName'], y=df['TotalInvoices'], textposition='auto')\n    fig.update_layout(title='Most Invoices  
by Customer')\n\nfig.show()\n```\n}", 'done_reason': 'stop', 'done': True, 'total_duration': 31286231147, 'lo  
ad_duration': 2324904, 'prompt_eval_count': 209, 'prompt_eval_duration': 12355927000, 'eval_count': 117, 'e  
val_duration': 18822127000}
```

Most Invoices by Customer



```
Out[25]: ('SELECT C.LastName, COUNT(I.InvoiceId) AS TotalInvoices\nFROM Customer C\nJOIN Invoice I ON C.CustomerId\n= I.CustomerId\nGROUP BY C.CustomerId, C.LastName\nORDER BY TotalInvoices DESC\nLIMIT 1',
  LastName TotalInvoices
0 Gonçalves          7,
Figure({
  'data': [{'marker': {'color': 'blue'}, 'mode': 'markers', 'type': 'scatter', 'x': ['Customer'], 'y':
[7]}],
  'layout': {'template': '...', 'title': {'text': 'Most Invoices by Customer'}}
}))
```

In []:

Advanced SQL questions

```
In [26]: question = """
  Find the customer who bought the most albums in total quantity (across all invoices):
  """
  vn.ask(question=question)
```

Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1

ame/Downloads/sqlite-ollama-chromadb-papagame-test-2-ok.html 1

```
A.ArtistId = ART.ArtistId'}}, {'role': 'user', 'content': ' \n    List all genres and the number of tracks
in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT G.Name, COUNT(T.TrackId) AS TrackCount \nFROM
Genre G \nLEFT JOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name'}}, {'role': 'user', 'content': 'what
are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUN
T(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5'}}, {'role': 'user', 'content':
' \n    Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'conte
nt': "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': ' \n    List a
ll customers from Canada and their email addresses:\n'}, {'role': 'assistant', 'content': "SELECT Email, Ci
ty, Country\nFROM Customer\nWHERE Country = 'Canada'\n"}, {'role': 'user', 'content': " SELECT * FROM t_per
son WHERE name = 'John Doe';"}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John
Doe'"}, {'role': 'user', 'content': " \n    List all employees and their reporting manager's name (if an
y):\n"}, {'role': 'assistant', 'content': "SELECT E.LastName, E.FirstName, \n        CASE WHEN ReportsTo IS
NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END
AS ReportingManager\nFROM Employee E"}, {'role': 'user', 'content': ' \n    Find the customer who bought
the most albums in total quantity (across all invoices): \n'}]
```

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVAR
CHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGE
R,\n    [Composer] NVARCHAR(220),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPr
ice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([Album
Id]) REFERENCES [Album] ([AlbumId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([Genre
Id]) REFERENCES [Genre] ([GenreId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([Media
TypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE
INDEX [IFK_AlbumArtistId] ON [Album] ([ArtistId])\n\nCREATE TABLE [Album]\n(\n    [AlbumId] INTEGER NOT NU
LL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER NOT NULL,\n    CONSTRAINT [PK_Album] PRI
MARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\t\tON DELETE NO AC
TION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE T
ABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDat
e] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingSt
ate] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total]
NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([Custom
erId]) REFERENCES [Customer] ([CustomerId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDE
X [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineI
d] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPric
e] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY
([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACT
```

```

ION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE NO ACT
ION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineInvoiceId] ON [InvoiceLine] ([InvoiceId])\n\nCRE
ATE INDEX [IFK_TrackAlbumId] ON [Track] ([AlbumId])\n\nCREATE TABLE [Artist]\n(\n    [ArtistId] INTEGER NO
T NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Artist] PRIMARY KEY ([ArtistId])\n)\n\n\n====Additio
nal Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and
in full\n\n====Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL q
uery without any explanations for the question. \n2. If the provided context is almost sufficient but requi
res knowledge of a specific string in a particular column, please generate an intermediate SQL query to fin
d the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the
provided context is insufficient, please explain why it can't be generated. \n4. Please use the most releva
nt table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as i
t was given before. \n"}, {"role": "user", "content": " \n    Find the customer with the most invoices
\n"}, {"role": "assistant", "content": "SELECT C.LastName, COUNT(I.InvoiceId) AS TotalInvoices\nFROM Custom
er C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.CustomerId, C.LastName\nORDER BY TotalInvoi
ces DESC\nLIMIT 1"}, {"role": "user", "content": " \n    Get the average invoice total for each custome
r:\n"}, {"role": "assistant", "content": "SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal\nFROM Cust
omer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName"}, {"role": "user", "content": "
\n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM
Invoice"}, {"role": "user", "content": " \n    List all albums and their corresponding artist names \n"},
{"role": "assistant", "content": "SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJOIN Artist ART ON
A.ArtistId = ART.ArtistId"}, {"role": "user", "content": " \n    List all genres and the number of tracks
in each genre:\n"}, {"role": "assistant", "content": "SELECT G.Name, COUNT(T.TrackId) AS TrackCount \nFROM
Genre G \nLEFT JOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name"}, {"role": "user", "content": "what
are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUN
T(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5"}, {"role": "user", "content":
" \n    Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "con
tent": "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": " \n    List
all customers from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SELECT Email, C
ity, Country\nFROM Customer\nWHERE Country = 'Canada'\n"}, {"role": "user", "content": " SELECT * FROM t_pe
rson WHERE name = 'John Doe';"}, {"role": "assistant", "content": "SELECT * FROM t_person WHERE name = 'Joh
n Doe'"}, {"role": "user", "content": " \n    List all employees and their reporting manager's name (if an
y):\n"}, {"role": "assistant", "content": "SELECT E.LastName, E.FirstName, \n    CASE WHEN ReportsTo IS
NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END
AS ReportingManager\nFROM Employee E"}, {"role": "user", "content": " \n    Find the customer who bought
the most albums in total quantity (across all invoices): \n"}]

```

Ollama Response:

```

{'model': 'llama3:latest', 'created_at': '2024-06-08T20:15:47.748091544Z', 'message': {'role': 'assistant',
'content': '```\nSELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums\nFROM Customer C\nJOIN InvoiceLine IL O
N C.CustomerId = IL.InvoiceId\nGROUP BY C.LastName\nORDER BY TotalAlbums DESC\nLIMIT 1;\n```'}, 'done_reaso
n': 'stop', 'done': True, 'total_duration': 109060841817, 'load_duration': 714557, 'prompt_eval_count': 158
6, 'prompt_eval_duration': 99565268000, 'eval_count': 52, 'eval_duration': 8799981000}
```

```

```
SELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums
FROM Customer C
JOIN InvoiceLine IL ON C.CustomerId = IL.InvoiceId
GROUP BY C.LastName
ORDER BY TotalAlbums DESC
LIMIT 1;
```

```

Output from LLM: ```

```
SELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums
FROM Customer C
JOIN InvoiceLine IL ON C.CustomerId = IL.InvoiceId
GROUP BY C.LastName
ORDER BY TotalAlbums DESC
LIMIT 1;
```

```

```
Extracted SQL: SELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums
FROM Customer C
JOIN InvoiceLine IL ON C.CustomerId = IL.InvoiceId
GROUP BY C.LastName
ORDER BY TotalAlbums DESC
LIMIT 1
SELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums
FROM Customer C
JOIN InvoiceLine IL ON C.CustomerId = IL.InvoiceId
GROUP BY C.LastName
ORDER BY TotalAlbums DESC
LIMIT 1
```

```
 LastName TotalAlbums
0 Wichterlová 14
```

Ollama parameters:  
model=llama3:latest,  
options={},  
keep\_alive=None  
Prompt Content:

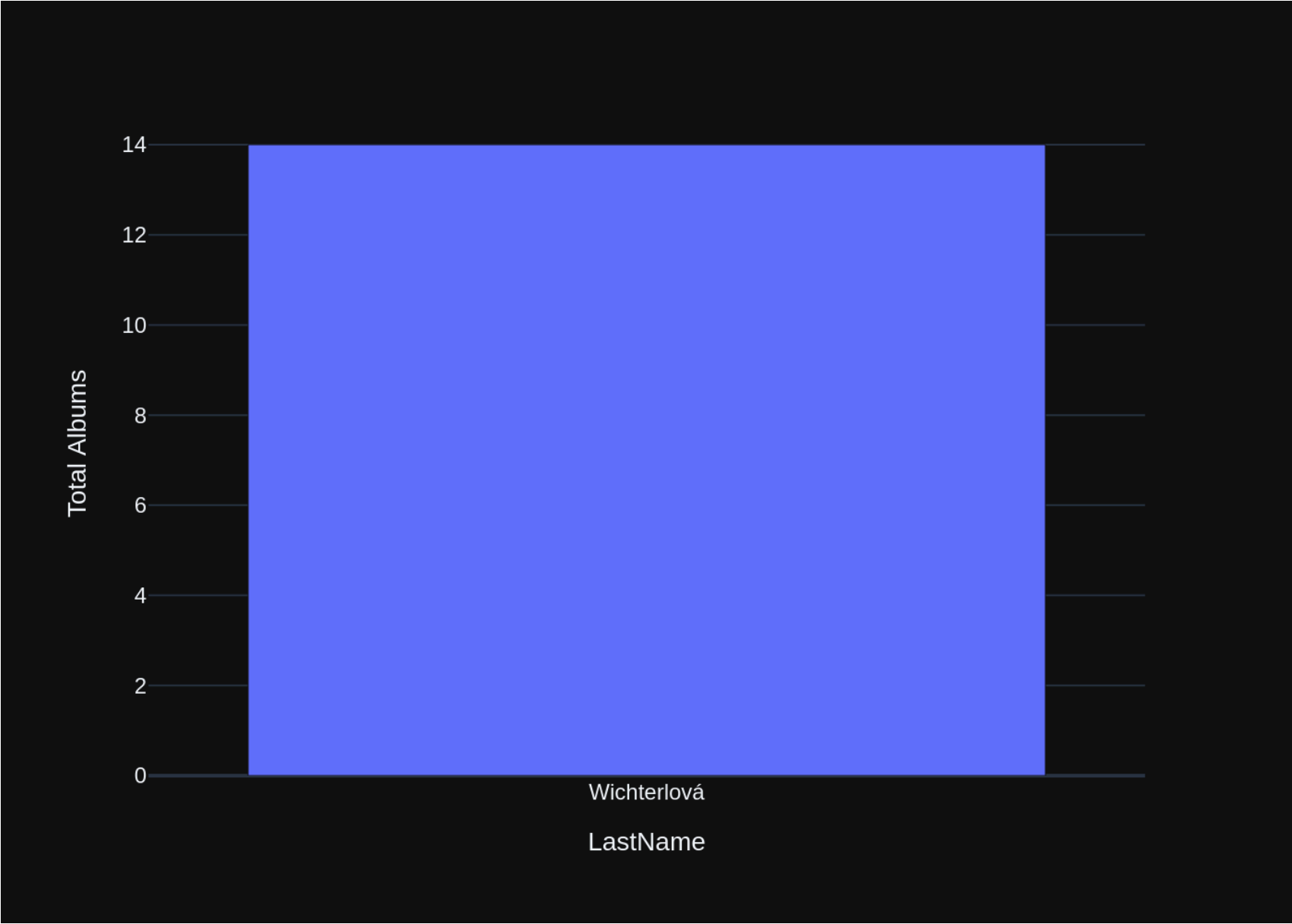
```
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n Find the customer who bought the most albums in total q\nquantity (across all invoices): \n'\n\nThe DataFrame was produced using this query: SELECT C.LastName, SUM(I\nL.Quantity) AS TotalAlbums\nFROM Customer C\nJOIN InvoiceLine IL ON C.CustomerId = IL.InvoiceId\nGROUP BY\nC.LastName\nORDER BY TotalAlbums DESC\nLIMIT 1\n\nThe following is information about the resulting pandas D\nataFrame 'df': \nRunning df.dtypes gives:\n LastName object\nTotalAlbums int64\nndtype: object"},\n{"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an I
```



ndicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T20:16:12.673090795Z', 'message': {'role': 'assistant',
'content': '```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = px.bar(df, x=\'LastName\', y=\'TotalAlbums\')\n\nif len(df) == 1:\n fig.update_layout(yaxis_title="Total Albums")\nelse:\n fig.update_layout(title_text="Customer who bought the most albums")\n\nfig.show()\n```'}, 'done_reason': 's
top', 'done': True, 'total_duration': 24756471534, 'load_duration': 2392490, 'prompt_eval_count': 217, 'pro
mpt_eval_duration': 12825054000, 'eval_count': 74, 'eval_duration': 11821524000}
```



```

Out[26]: ('SELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums\nFROM Customer C\nJOIN InvoiceLine IL ON C.CustomerId
= IL.InvoiceId\nGROUP BY C.LastName\nORDER BY TotalAlbums DESC\nLIMIT 1',
 LastName TotalAlbums
0 Wichterlová 14,
Figure({
 'data': [{'alignmentgroup': 'True',
 'hovernplate': 'LastName=%{x}
TotalAlbums=%{y}<extra></extra>',
 'legendgroup': '',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': '',
 'offsetgroup': '',
 'orientation': 'v',
 'showlegend': False,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['Wichterlová'], dtype=object),
 'xaxis': 'x',
 'y': array([14]),
 'yaxis': 'y'}],
 'layout': {'barmode': 'relative',
 'legend': {'tracegroupgap': 0},
 'margin': {'t': 60},
 'template': '...',
 'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'LastName'}},
 'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total Albums'}}}
}))

```

```

In [27]: question = """
 Find the top 5 customer who bought the most albums in total quantity (across all invoices):
 """
 vn.ask(question=question)

```

Number of requested results 10 is greater than number of elements in index 1, updating n\_results = 1

ame/Downloads/sqlite-ollama-chromadb-papagame-test-2-ok.html 1

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions."}]

===Tables
CREATE TABLE [Track]
(
 [TrackId] INTEGER NOT NULL,
 [Name] NVARCHAR(200) NOT NULL,
 [AlbumId] INTEGER,
 [MediaTypeId] INTEGER NOT NULL,
 [GenreId] INTEGER,
 [Composer] NVARCHAR(220),
 [Milliseconds] INTEGER NOT NULL,
 [Bytes] INTEGER,
 [UnitPrice] NUMERIC(10,2) NOT NULL,
 CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),
 FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,
 FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,
 FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
)

CREATE INDEX [IFK_AlbumArtistId] ON [Album] ([ArtistId])

CREATE TABLE [Album]
(
 [AlbumId] INTEGER NOT NULL,
 [Title] NVARCHAR(160) NOT NULL,
 [ArtistId] INTEGER NOT NULL,
 CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),
 FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
)

CREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])

CREATE TABLE [Invoice]
(
 [InvoiceId] INTEGER NOT NULL,
 [CustomerId] INTEGER NOT NULL,
 [InvoiceDate] DATETIME NOT NULL,
 [BillingAddress] NVARCHAR(70),
 [BillingCity] NVARCHAR(40),
 [BillingState] NVARCHAR(40),
 [BillingCountry] NVARCHAR(40),
 [BillingPostalCode] NVARCHAR(10),
 [Total] NUMERIC(10,2) NOT NULL,
 CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),
 FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
)

CREATE INDEX [IFK_TrackAlbumId] ON [Track] ([AlbumId])

CREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])

CREATE INDEX [IFK_InvoiceLineInvoiceId] ON [InvoiceLine] ([InvoiceId])

CREATE TABLE [InvoiceLine]
(
 [InvoiceLineId] INTEGER NOT NULL,
 [InvoiceId] INTEGER NOT NULL,
 [TrackId] INTEGER NOT NULL,
 [UnitPrice] NUMERIC(10,2) NOT NULL,
 [Quantity] INTEGER NOT NULL,
 CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),
 FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,
 FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
)
```

```

InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceI
d]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackI
d]) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Artist]\n(\n [ArtistId] INTEGER N
OT NULL,\n [Name] NVARCHAR(120),\n CONSTRAINT [PK_Artist] PRIMARY KEY ([ArtistId])\n)\n\n\n===Additi
onal Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and
in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL q
uery without any explanations for the question. \n2. If the provided context is almost sufficient but requi
res knowledge of a specific string in a particular column, please generate an intermediate SQL query to fin
d the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the
provided context is insufficient, please explain why it can't be generated. \n4. Please use the most releva
nt table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as i
t was given before. \n"}, {"role": "user", "content": " \n Find the customer who bought the most album
s in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT C.LastName, SUM(I
L.Quantity) AS TotalAlbums\nFROM Customer C\nJOIN InvoiceLine IL ON C.CustomerId = IL.InvoiceId\nGROUP BY
C.LastName\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n Find the customer w
ith the most invoices \n"}, {"role": "assistant", "content": "SELECT C.LastName, COUNT(I.InvoiceId) AS Tota
lInvoices\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.CustomerId, C.LastNam
e\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n Get the average invoice tot
al for each customer:\n"}, {"role": "assistant", "content": "SELECT C.LastName, AVG(I.Total) AS AverageInvo
iceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName"}, {"role":
"user", "content": " \n List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "conte
nt": "SELECT * \nFROM Invoice"}, {"role": "user", "content": "what are the top 5 countries that customers c
ome from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY C
ountry\nORDER BY Count DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all albums and their corr
esponding artist names \n"}, {"role": "assistant", "content": "SELECT A.Title, A.ArtistId, ART.Name \nFROM
Album A \nJOIN Artist ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": " \n List all gen
res and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT G.Name, COUNT(T.T
rackId) AS TrackCount \nFROM Genre G \nLEFT JOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name"}, {"ro
le": "user", "content": " \n Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"
role": "assistant", "content": "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user",
"content": " \n List all customers from Canada and their email addresses:\n"}, {"role": "assistant",
"content": "SELECT Email, City, Country\nFROM Customer\nWHERE Country = 'Canada'\n"}, {"role": "user", "con
tent": " SELECT * FROM t_person WHERE name = 'John Doe';"}, {"role": "assistant", "content": "SELECT * FROM
t_person WHERE name = 'John Doe'"}, {"role": "user", "content": " \n Find the top 5 customer who bough
t the most albums in total quantity (across all invoices):\n"}]

```

Ollama Response:

```

{'model': 'llama3:latest', 'created_at': '2024-06-08T20:18:01.794209073Z', 'message': {'role': 'assistant',
'content': '```\nSELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums\nFROM Customer C\nJOIN InvoiceLine IL O
N C.CustomerId = IL.InvoiceId\nGROUP BY C.LastName\nORDER BY TotalAlbums DESC\nLIMIT 5;\n```'}, 'done_reaso
n': 'stop', 'done': True, 'total_duration': 109013160945, 'load_duration': 632213, 'prompt_eval_count': 158
5, 'prompt_eval_duration': 99564151000, 'eval_count': 52, 'eval_duration': 8759299000}
```

```

```

SELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums
FROM Customer C
JOIN InvoiceLine IL ON C.CustomerId = IL.InvoiceId
GROUP BY C.LastName
ORDER BY TotalAlbums DESC
LIMIT 5;
```

```

Output from LLM: ```

```

SELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums
FROM Customer C
JOIN InvoiceLine IL ON C.CustomerId = IL.InvoiceId
GROUP BY C.LastName
ORDER BY TotalAlbums DESC
LIMIT 5;
```

```

```

Extracted SQL: SELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums
FROM Customer C
JOIN InvoiceLine IL ON C.CustomerId = IL.InvoiceId
GROUP BY C.LastName
ORDER BY TotalAlbums DESC
LIMIT 5
SELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums
FROM Customer C
JOIN InvoiceLine IL ON C.CustomerId = IL.InvoiceId
GROUP BY C.LastName
ORDER BY TotalAlbums DESC
LIMIT 5

```

	LastName	TotalAlbums
0	Wichterlová	14
1	Sullivan	14
2	Murray	14
3	Mancini	14
4	Lefebvre	14

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```

[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n      Find the top 5 customer who bought the most albums in total quantity (across all invoices):\n'\n\nThe DataFrame was produced using this query: SELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums\nFROM Customer C\nJOIN InvoiceLine IL ON C.CustomerId = IL.InvoiceId\nGROUP

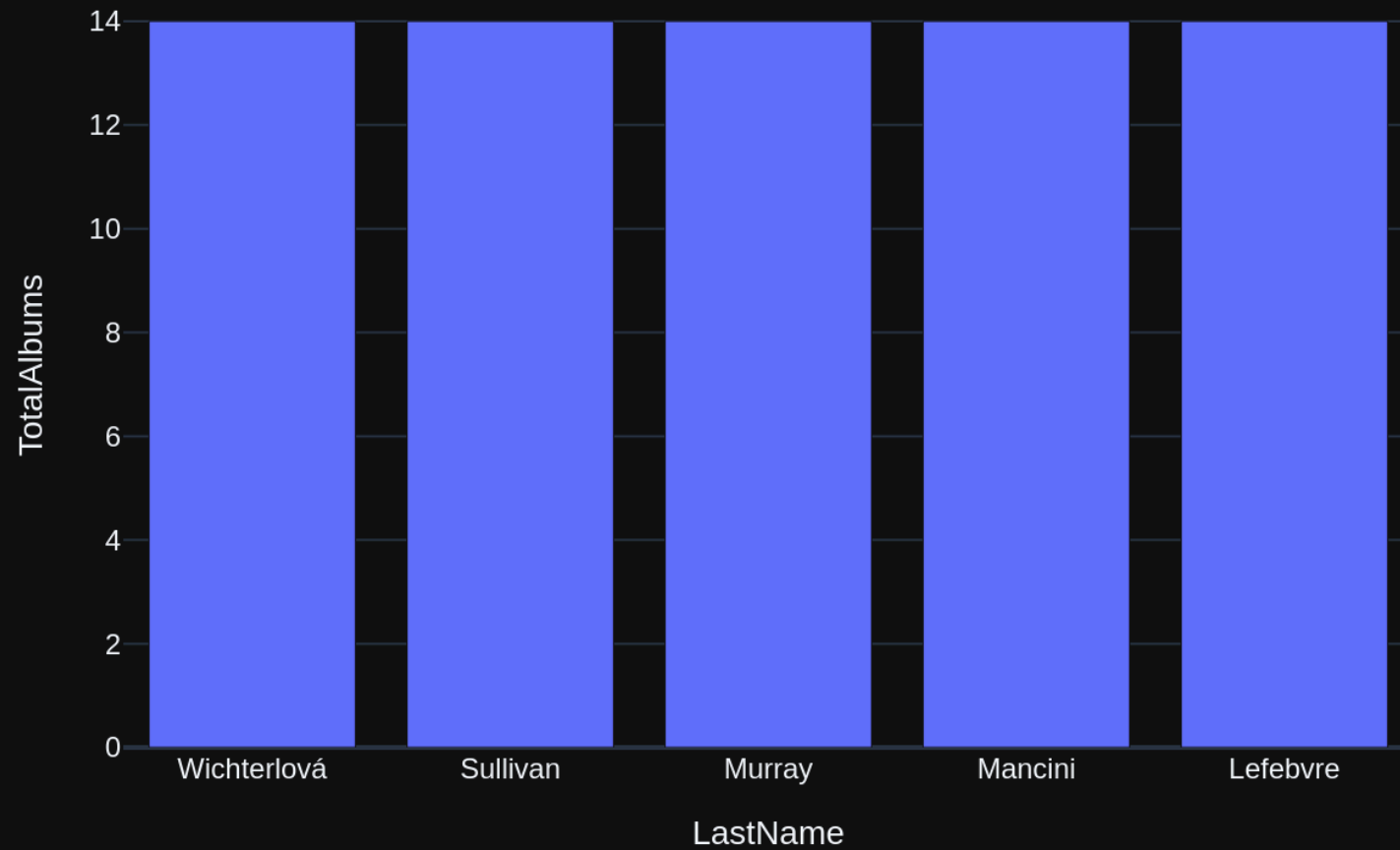
```

BY C.LastName\nORDER BY TotalAlbums DESC\nLIMIT 5\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n LastName object\nTotalAlbums int64\nndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T20:18:32.648770148Z', 'message': {'role': 'assistant',
'content': '```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = px.bar(df, x="Last\nName", y="TotalAlbums", title=\'Top 5 Customers by Total Album Purchases\')\n\nif len(df) == 1:\n    fig =\ngo.Figure(data=[go.Indicator(\n        mode = "number",\n        value = df["TotalAlbums"].values[0],\n        number={\'value\': int(df["TotalAlbums"].values[0])}\n    )])\n\nfig.show()\n```'}, 'done_reason': 'stop',
'done': True, 'total_duration': 30724961045, 'load_duration': 819430, 'prompt_eval_count': 220, 'prompt_eva\nl_duration': 12935308000, 'eval_count': 110, 'eval_duration': 17694919000}
```


Top 5 Customers by Total Album Purchases



```
Out[27]: ('SELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums\nFROM Customer C\nJOIN InvoiceLine IL ON C.CustomerId\n= IL.InvoiceId\nGROUP BY C.LastName\nORDER BY TotalAlbums DESC\nLIMIT 5',
```

```
    LastName TotalAlbums
0  Wichterlová      14
1    Sullivan      14
2     Murray      14
3    Mancini      14
4   Lefebvre      14,
Figure({
  'data': [{'alignmentgroup': 'True',
            'hovertemplate': 'LastName=%{x}<br>TotalAlbums=%{y}<extra></extra>',
            'legendgroup': '',
            'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
            'name': '',
            'offsetgroup': '',
            'orientation': 'v',
            'showlegend': False,
            'textposition': 'auto',
            'type': 'bar',
            'x': array(['Wichterlová', 'Sullivan', 'Murray', 'Mancini', 'Lefebvre'],
                       dtype=object),
            'xaxis': 'x',
            'y': array([14, 14, 14, 14, 14]),
            'yaxis': 'y'}],
  'layout': {'barmode': 'relative',
             'legend': {'tracegroupgap': 0},
             'template': '...',
             'title': {'text': 'Top 5 Customers by Total Album Purchases'},
             'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'LastName'}},
             'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalAlbums'}}}
}))
```

```
In [28]: question = """
        Find the top 3 customers who spent the most money overall:
        """

        vn.ask(question=question)
```

Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1

```
{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE [Invoice]\n(\n    [InvoiceId] INTEGER NOT NULL,\n    [CustomerId] INTEGER NOT NULL,\n    [InvoiceDate] DATETIME NOT NULL,\n    [BillingAddress] NVARCHAR(70),\n    [BillingCity] NVARCHAR(40),\n    [BillingState] NVARCHAR(40),\n    [BillingCountry] NVARCHAR(40),\n    [BillingPostalCode] NVARCHAR(10),\n    [Total] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Invoice] PRIMARY KEY ([InvoiceId]),\n    FOREIGN KEY ([CustomerId]) REFERENCES [Customer] ([CustomerId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_CustomerSupportRepId] ON [Customer] ([SupportRepId])\n\nCREATE TABLE [Customer]\n(\n    [CustomerId] INTEGER NOT NULL,\n    [FirstName] NVARCHAR(40) NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [Company] NVARCHAR(80),\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceCustomerId] ON [Invoice] ([CustomerId])\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NULL,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineId]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(20) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGER,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(40),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeId]),\n    FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_EmployeeReportsTo] ON [Employee] ([ReportsTo])\n\nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(220),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\nCREATE INDEX [IFK_InvoiceLineInvoiceId] ON [InvoiceLine] ([InvoiceId])\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}}, {'role': 'user', 'content': 'Find the top 5 customer who bought the most al
```

ame/Downloads/sqlite-ollama-chromadb-papagame-test-2-ok.html 196/214

```

hone] NVARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60) NOT NULL,\n    [SupportRepId] INTEGE
ER,\n    CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),\n    FOREIGN KEY ([SupportRepId]) REFERENCES
[Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceCust
omerId] ON [Invoice] ([CustomerId])\n\nCREATE TABLE [InvoiceLine]\n(\n    [InvoiceLineId] INTEGER NOT NUL
L,\n    [InvoiceId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    [UnitPrice] NUMERIC(10,2) NO
T NULL,\n    [Quantity] INTEGER NOT NULL,\n    CONSTRAINT [PK_InvoiceLine] PRIMARY KEY ([InvoiceLineI
d]),\n    FOREIGN KEY ([InvoiceId]) REFERENCES [Invoice] ([InvoiceId]) \n\t\t\tON DELETE NO ACTION ON UPDATE
NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE
NO ACTION\n)\n\nCREATE TABLE [Employee]\n(\n    [EmployeeId] INTEGER NOT NULL,\n    [LastName] NVARCHAR(2
0) NOT NULL,\n    [FirstName] NVARCHAR(20) NOT NULL,\n    [Title] NVARCHAR(30),\n    [ReportsTo] INTEGE
R,\n    [BirthDate] DATETIME,\n    [HireDate] DATETIME,\n    [Address] NVARCHAR(70),\n    [City] NVARCHAR(4
0),\n    [State] NVARCHAR(40),\n    [Country] NVARCHAR(40),\n    [PostalCode] NVARCHAR(10),\n    [Phone] NV
ARCHAR(24),\n    [Fax] NVARCHAR(24),\n    [Email] NVARCHAR(60),\n    CONSTRAINT [PK_Employee] PRIMARY KEY
([EmployeeId]),\n    FOREIGN KEY ([ReportsTo]) REFERENCES [Employee] ([EmployeeId]) \n\t\t\tON DELETE NO ACTI
ON ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_EmployeeReportsTo] ON [Employee] ([ReportsTo])\n\nCREATE TAB
LE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGE
R,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(220),\n    [Milli
seconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT
[PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\t\tON D
ELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\t\tON D
ELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]
)\n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLi
ne] ([TrackId])\n\nCREATE INDEX [IFK_InvoiceLineInvoiceId] ON [InvoiceLine] ([InvoiceId])\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n    Find the top 5 customer who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums\nFROM Customer C\nJOIN InvoiceLine IL ON C.CustomerId = IL.InvoiceId\nGROUP BY C.LastName\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n    Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT C.LastName, COUNT(I.InvoiceId) AS TotalInvoices\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.CustomerId, C.LastName\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n    Find the customer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums\nFROM Customer C\nJOIN InvoiceLine IL ON C.CustomerId = IL.InvoiceId\nGROUP BY C.LastName\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n    Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY"}

```

```
C.LastName"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5"}, {"role": "user", "content": " \n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM Invoice"}, {"role": "user", "content": " \n    List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT G.Name, COUNT(T.TrackId) AS TrackCount \nFROM Genre G \nLEFT JOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name"}, {"role": "user", "content": " \n    List all customers from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SELECT Email, City, Country\nFROM Customer\nWHERE Country = 'Canada'\n"}, {"role": "user", "content": " \n    List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT E.LastName, E.FirstName, \n    CASE WHEN ReportsTo IS NULL THEN 'N/A' ELSE (SELECT LastName + ', ' + FirstName FROM Employee WHERE EmployeeId = E.ReportsTo) END AS ReportingManager\nFROM Employee E"}, {"role": "user", "content": " \n    Find all tracks with a name containing 'What' (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": " \n    Find the top 3 customers who spent the most money overall:\n"}]
```

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T20:20:44.175373553Z', 'message': {'role': 'assistant', 'content': '```\nSELECT C.LastName, SUM(I.Total) AS TotalSpent\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName\nORDER BY TotalSpent DESC\nLIMIT 3\n`'}, 'done_reason': 'stop', 'done': True, 'total_duration': 131418101316, 'load_duration': 578075, 'prompt_eval_count': 1925, 'prompt_eval_duration': 122201195000, 'eval_count': 50, 'eval_duration': 8519333000}
```

```
SELECT C.LastName, SUM(I.Total) AS TotalSpent
FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
GROUP BY C.LastName
ORDER BY TotalSpent DESC
LIMIT 3
```
```

Output from LLM: ```

```
SELECT C.LastName, SUM(I.Total) AS TotalSpent
FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
GROUP BY C.LastName
ORDER BY TotalSpent DESC
LIMIT 3
```
```

```
Extracted SQL: SELECT C.LastName, SUM(I.Total) AS TotalSpent
FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
GROUP BY C.LastName
ORDER BY TotalSpent DESC
```

LIMIT 3

```
SELECT C.LastName, SUM(I.Total) AS TotalSpent
FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
GROUP BY C.LastName
ORDER BY TotalSpent DESC
LIMIT 3
```

	LastName	TotalSpent
0	Holý	49.62
1	Cunningham	47.62
2	Rojas	46.62

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

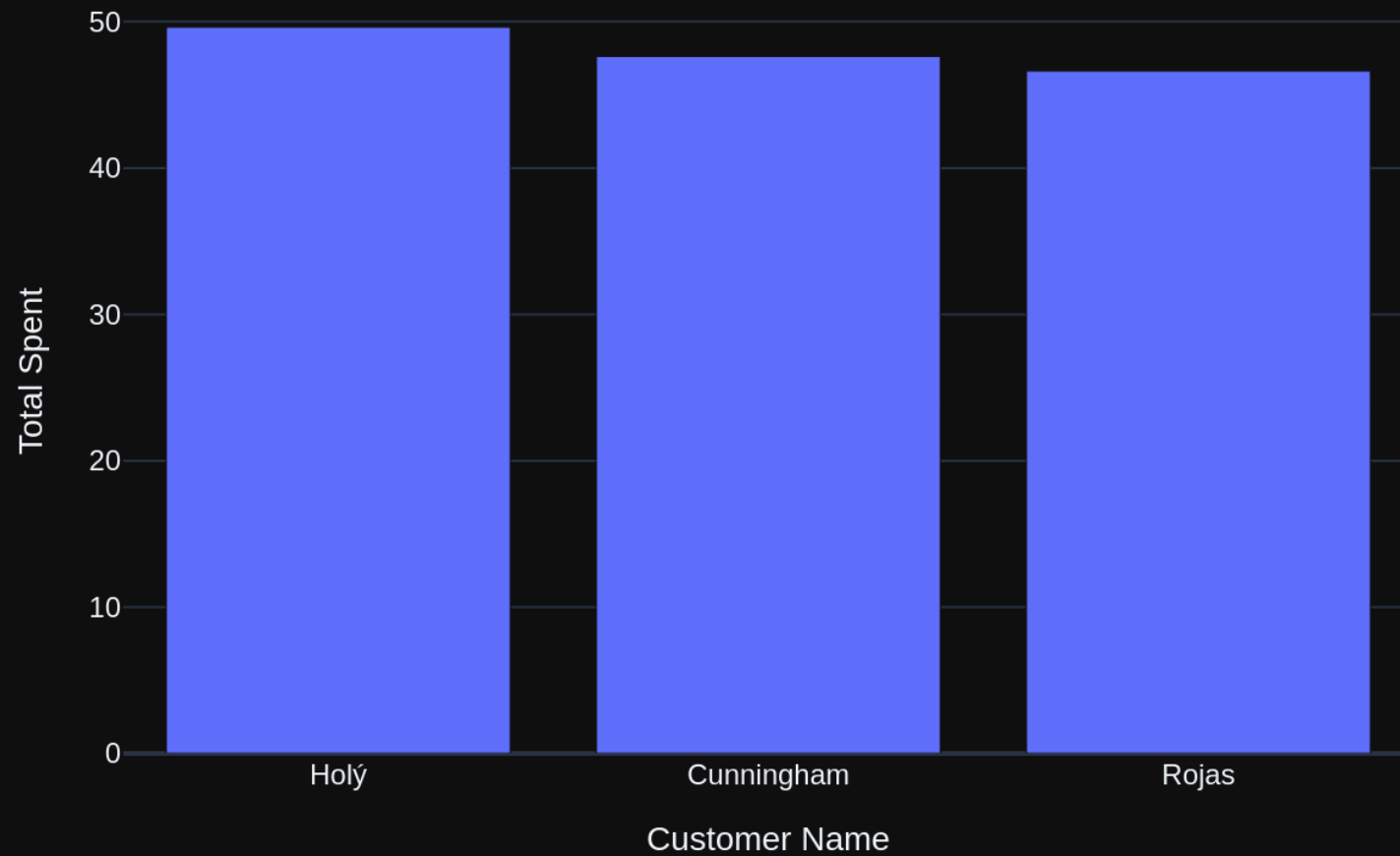
Prompt Content:

```
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n      Find the top 3 customers who spent the most money overall:\n'\n\nThe DataFrame was produced using this query: SELECT C.LastName, SUM(I.Total) AS TotalSpent\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName\nORDER BY TotalSpent DESC\nLIMIT 3\n'\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n LastName      object\n TotalSpent      float64\n dtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
```

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T20:21:15.280651789Z', 'message': {'role': 'assistant', 'content': "\n\nimport plotly.express as px\n\nfig = px.bar(df, x='LastName', y='TotalSpent', title='Top 3 Customers Who Spent the Most Money Overall')\nfig.update_layout(xaxis_title='Customer Name', yaxis_title='Total Spent')\n\nif df.shape[0] == 1:\n    fig = px.bar(x=[''], y=[df['TotalSpent'].values[0]], title='Only One Customer!')\n    fig.update_layout(xaxis_title='', yaxis_title='Total Spent')\n\nfig.show()\n\n"}}, {'done_reason': 'stop', 'done': True, 'total_duration': 30945142661, 'load_duration': 1762987, 'prompt_eval_count': 210, 'prompt_eval_duration': 12349096000, 'eval_count': 115, 'eval_duration': 18497077000}
```

Top 3 Customers Who Spent the Most Money Overall




```
Out[28]: ('SELECT C.LastName, SUM(I.Total) AS TotalSpent\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName\nORDER BY TotalSpent DESC\nLIMIT 3\n',
```

```
    LastName TotalSpent
0      Holý      49.62
1  Cunningham      47.62
2      Rojas      46.62,
```

```
Figure({
  'data': [{'alignmentgroup': 'True',
            'hovertemplate': 'LastName=%{x}<br>TotalSpent=%{y}<extra></extra>',
            'legendgroup': '',
            'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
            'name': '',
            'offsetgroup': '',
            'orientation': 'v',
            'showlegend': False,
            'textposition': 'auto',
            'type': 'bar',
            'x': array(['Holý', 'Cunningham', 'Rojas'], dtype=object),
            'xaxis': 'x',
            'y': array([49.62, 47.62, 46.62]),
            'yaxis': 'y'}],
  'layout': {'barmode': 'relative',
            'legend': {'tracegroupgap': 0},
            'template': '...',
            'title': {'text': 'Top 3 Customers Who Spent the Most Money Overall'},
            'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Customer Name'}},
            'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total Spent'}}
}))
```

```
In [29]: question = """
        Get all playlists containing at least 10 tracks and the total duration of those tracks:
        """
        vn.ask(question=question)
```

Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1

202/214

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions.\n\n===Tables\nCREATE INDEX [IFK_PlaylistTrackTrackId] ON [PlaylistTrack] ([TrackId])\n\nCREATE INDEX [IFK_TrackGenreId] ON [Track] ([GenreId])\n\nCREATE INDEX [IFK_TrackAlbumId] ON [Track] ([AlbumId])\n\nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVARCHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGER,\n    [Composer] NVARCHAR(20),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPrice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([AlbumId]) REFERENCES [Album] ([AlbumId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([GenreId]) REFERENCES [Genre] ([GenreId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([MediaTypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE [Playlist]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Playlist] PRIMARY KEY ([PlaylistId])\n)\n\nCREATE INDEX [IFK_TrackMediaTypeId] ON [Track] ([MediaTypeId])\n\nCREATE TABLE [PlaylistTrack]\n(\n    [PlaylistId] INTEGER NOT NULL,\n    [TrackId] INTEGER NOT NULL,\n    CONSTRAINT [PK_PlaylistTrack] PRIMARY KEY ([PlaylistId], [TrackId]),\n    FOREIGN KEY ([PlaylistId]) REFERENCES [Playlist] ([PlaylistId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([TrackId]) REFERENCES [Track] ([TrackId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_AlbumArtistId] ON [Album] ([ArtistId])\n\nCREATE TABLE [Album]\n(\n    [AlbumId] INTEGER NOT NULL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER NOT NULL,\n    CONSTRAINT [PK_Album] PRIMARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])\n\n\n===Additional Context\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n\n===Response Guidelines\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question.\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column.
```

mn. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT G.Name, COUNT(T.TrackId) AS TrackCount \nFROM Genre G \nLEFT JOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name"}, {"role": "user", "content": " \n Find the top 5 customer who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums\nFROM Customer C\nJOIN InvoiceLine IL ON C.CustomerId = IL.InvoiceId\nGROUP BY C.LastName\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find all tracks with a name containing 'What' (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": " \n Find the customer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums\nFROM Customer C\nJOIN InvoiceLine IL ON C.CustomerId = IL.InvoiceId\nGROUP BY C.LastName\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT A.Title, A.ArtistId, ART.Name \nFROM Album A \nJOIN Artist ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": " \n Find the top 3 customers who spent the most money overall:\n"}, {"role": "assistant", "content": "SELECT C.LastName, SUM(I.Total) AS TotalSpent\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName\nORDER BY TotalSpent DESC\nLIMIT 3\n"}, {"role": "user", "content": " \n Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT C.LastName, COUNT(I.InvoiceId) AS TotalInvoices\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.CustomerId, C.LastName\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT C.LastName, AVG(I.Total) AS AverageInvoiceTotal\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.LastName"}, {"role": "user", "content": " \n List all invoices with a total exceeding \$10:\n"}, {"role": "assistant", "content": "SELECT *\nFROM Invoice"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5"}, {"role": "user", "content": " \n Get all playlists containing at least 10 tracks and the total duration of those tracks:\n"}]

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T20:22:59.510887409Z', 'message': {'role': 'assistant', 'content': '```\nSELECT P.Name, SUM(T.Milliseconds) AS TotalDuration\nFROM PlaylistTrack PT\nJOIN Track T ON PT.TrackId = T.TrackId\nJOIN Playlist P ON PT.PlaylistId = P.PlaylistId\nGROUP BY P.Name\nHAVING COUNT(*)\n>= 10;\n```\n'}, 'done_reason': 'stop', 'done': True, 'total_duration': 104014370452, 'load_duration': 604210, 'prompt_eval_count': 1487, 'prompt_eval_duration': 93063789000, 'eval_count': 61, 'eval_duration': 10295313000}\n```\n
```

```
SELECT P.Name, SUM(T.Milliseconds) AS TotalDuration
FROM PlaylistTrack PT
JOIN Track T ON PT.TrackId = T.TrackId
JOIN Playlist P ON PT.PlaylistId = P.PlaylistId
GROUP BY P.Name
```

```
HAVING COUNT(*) >= 10;
```
```

Output from LLM: ```

```
SELECT P.Name, SUM(T.Milliseconds) AS TotalDuration
FROM PlaylistTrack PT
JOIN Track T ON PT.TrackId = T.TrackId
JOIN Playlist P ON PT.PlaylistId = P.PlaylistId
GROUP BY P.Name
HAVING COUNT(*) >= 10;
```
```

Extracted SQL: SELECT P.Name, SUM(T.Milliseconds) AS TotalDuration
FROM PlaylistTrack PT
JOIN Track T ON PT.TrackId = T.TrackId
JOIN Playlist P ON PT.PlaylistId = P.PlaylistId
GROUP BY P.Name
HAVING COUNT(*) >= 10
SELECT P.Name, SUM(T.Milliseconds) AS TotalDuration
FROM PlaylistTrack PT
JOIN Track T ON PT.TrackId = T.TrackId
JOIN Playlist P ON PT.PlaylistId = P.PlaylistId
GROUP BY P.Name
HAVING COUNT(*) >= 10

	Name	TotalDuration
0	90's Music	398705153
1	Brazilian Music	9486559
2	Classical	21770592
3	Classical 101 - Deep Cuts	6755730
4	Classical 101 - Next Steps	7575051
5	Classical 101 - The Basics	7439811
6	Grunge	4122018
7	Heavy Metal Classic	8206312
8	Music	1755366166
9	TV Shows	1002189914

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

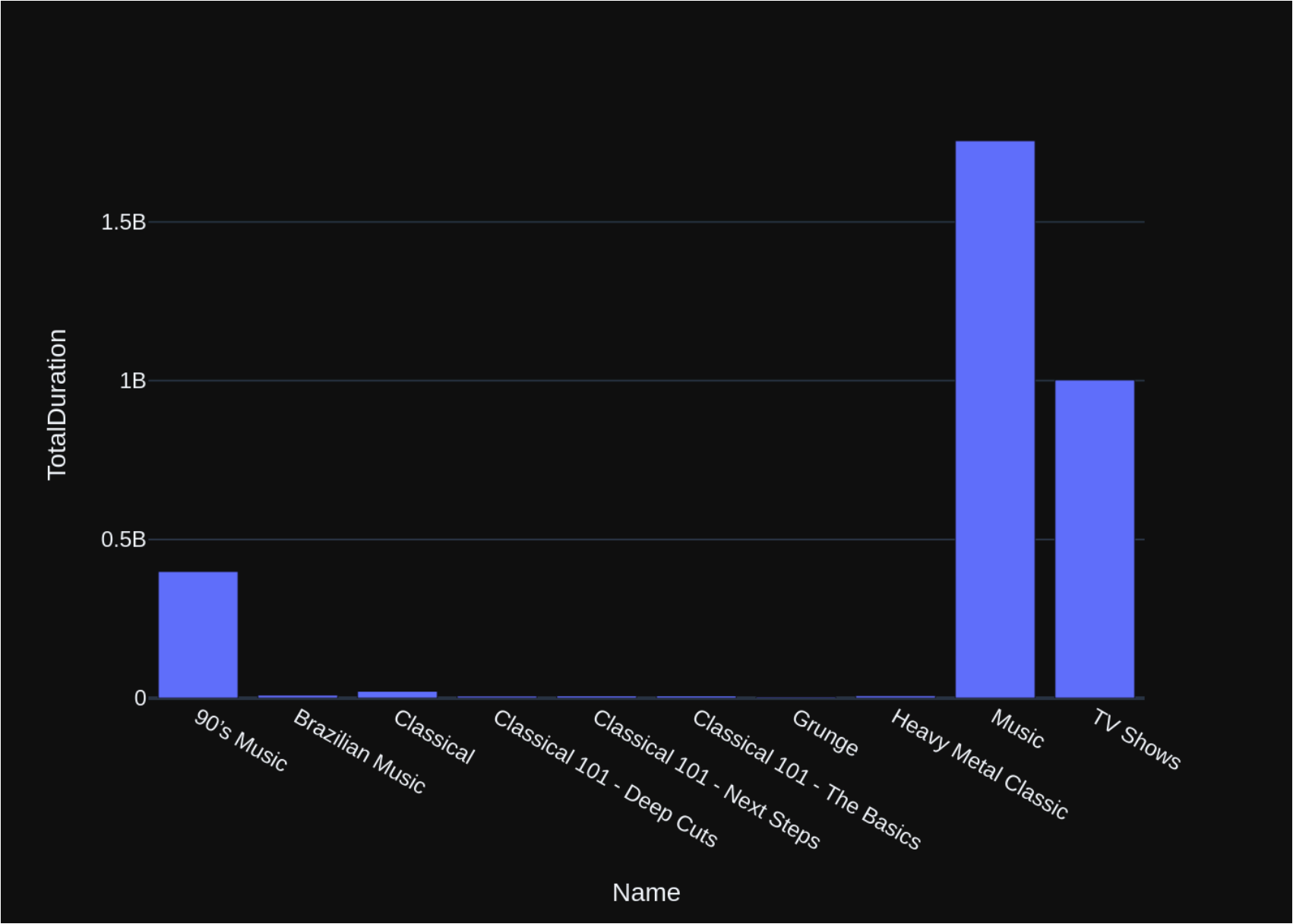
Prompt Content:

```
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n      Get all playlists containing at least 10 tracks and the total duration of those tracks:\n'\n\nThe DataFrame was produced using this query: SELECT P.Name, SUM(T.Miliseconds) AS TotalDuration\nFROM PlaylistTrack PT\nJOIN Track T ON PT.TrackId = T.TrackId\nJOIN Playlist P
```

ON PT.PlaylistId = P.PlaylistId\nGROUP BY P.Name\nHAVING COUNT(*) >= 10\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name object\nTotalDuration int64\nndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]

Ollama Response:

```
{'model': 'llama3:latest', 'created_at': '2024-06-08T20:23:28.975303497Z', 'message': {'role': 'assistant',
'content': '```\nimport plotly.express as px\nimport numpy as np\n\nfig = px.bar(df, x=\'Name\', y=\'TotalDuration\')\n\nif len(df) == 1:\n    fig.update_layout(yaxis={\'visible\': False})\n    fig.add_annotation(text="Total Duration: " + str(df[\'TotalDuration\'].values[0]), \n                        x=0.5, \n                        y=-0.15,\n                        font=dict(size=12))\n\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 29325918068, 'load_duration': 804038, 'prompt_eval_count': 228, 'prompt_eval_duration': 13460035000, 'eval_count': 98, 'eval_duration': 15772725000}
```



```
Out[29]: ('SELECT P.Name, SUM(T.Milliseconds) AS TotalDuration\nFROM PlaylistTrack PT\nJOIN Track T ON PT.TrackId =\nT.TrackId\nJOIN Playlist P ON PT.PlaylistId = P.PlaylistId\nGROUP BY P.Name\nHAVING COUNT(*) >= 10',
```

```

      Name  TotalDuration
0      90's Music      398705153
1    Brazilian Music      9486559
2      Classical      21770592
3 Classical 101 - Deep Cuts      6755730
4 Classical 101 - Next Steps      7575051
5 Classical 101 - The Basics      7439811
6      Grunge      4122018
7 Heavy Metal Classic      8206312
8      Music      1755366166
9      TV Shows      1002189914,
```

```
Figure({
  'data': [{'alignmentgroup': 'True',
            'hovertemplate': 'Name=%{x}<br>TotalDuration=%{y}<extra></extra>',
            'legendgroup': '',
            'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
            'name': '',
            'offsetgroup': '',
            'orientation': 'v',
            'showlegend': False,
            'textposition': 'auto',
            'type': 'bar',
            'x': array(['90's Music', 'Brazilian Music', 'Classical',
                       'Classical 101 - Deep Cuts', 'Classical 101 - Next Steps',
                       'Classical 101 - The Basics', 'Grunge', 'Heavy Metal Classic', 'Music',
                       'TV Shows'], dtype=object),
            'xaxis': 'x',
            'y': array([ 398705153,    9486559,    21770592,    6755730,    7575051,    7439811,
                       4122018,    8206312, 1755366166, 1002189914]),
            'yaxis': 'y'}],
  'layout': {'barmode': 'relative',
            'legend': {'tracegroupgap': 0},
            'margin': {'t': 60},
            'template': '...',
            'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
            'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalDuration'}}
})
```

```
In [30]: question = ""
```

Identify artists who have albums with tracks appearing in multiple genres:


```
"""
```

```
vn.ask(question=question)
```

```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

210/214

```
tent': ' \n      Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT C.
LastName, COUNT(I.InvoiceId) AS TotalInvoices\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.Customer
Id\nGROUP BY C.CustomerId, C.LastName\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content':
' \n      Find the top 3 customers who spent the most money overall:\n'}, {'role': 'assistant', 'content':
'SELECT C.LastName, SUM(I.Total) AS TotalSpent\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.Custome
rId\nGROUP BY C.LastName\nORDER BY TotalSpent DESC\nLIMIT 3\n'}, {'role': 'user', 'content': 'what are the
top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS
Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5'}, {'role': 'user', 'content': " SELEC
T * FROM t_person WHERE name = 'John Doe';"}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHER
E name = 'John Doe'"}, {'role': 'user', 'content': ' \n      Identify artists who have albums with tracks a
ppearing in multiple genres:\n'}]]
```

Ollama parameters:

model=llama3:latest,

options={},

keep_alive=None

Prompt Content:

```
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE [Track]\n(\n    [TrackId] INTEGER NOT NULL,\n    [Name] NVAR
CHAR(200) NOT NULL,\n    [AlbumId] INTEGER,\n    [MediaTypeId] INTEGER NOT NULL,\n    [GenreId] INTEGE
R,\n    [Composer] NVARCHAR(220),\n    [Milliseconds] INTEGER NOT NULL,\n    [Bytes] INTEGER,\n    [UnitPr
ice] NUMERIC(10,2) NOT NULL,\n    CONSTRAINT [PK_Track] PRIMARY KEY ([TrackId]),\n    FOREIGN KEY ([Album
Id]) REFERENCES [Album] ([AlbumId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([Genre
Id]) REFERENCES [Genre] ([GenreId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY ([Media
TypeId]) REFERENCES [MediaType] ([MediaTypeId]) \n\t\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE
INDEX [IFK_AlbumArtistId] ON [Album] ([ArtistId])\n\nCREATE INDEX [IFK_TrackAlbumId] ON [Track] ([AlbumId])
\n\nCREATE INDEX [IFK_TrackGenreId] ON [Track] ([GenreId])\n\nCREATE TABLE [Album]\n(\n    [AlbumId] INTEGE
R NOT NULL,\n    [Title] NVARCHAR(160) NOT NULL,\n    [ArtistId] INTEGER NOT NULL,\n    CONSTRAINT [PK_A
lbum] PRIMARY KEY ([AlbumId]),\n    FOREIGN KEY ([ArtistId]) REFERENCES [Artist] ([ArtistId]) \n\t\t\tON DEL
ETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX [IFK_TrackMediaTypeId] ON [Track] ([MediaTypeId])\n\nC
REATE INDEX [IFK_PlaylistTrackTrackId] ON [PlaylistTrack] ([TrackId])\n\nCREATE TABLE [Artist]\n(\n    [Art
istId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT [PK_Artist] PRIMARY KEY ([ArtistId])
\n)\n\nCREATE TABLE [Genre]\n(\n    [GenreId] INTEGER NOT NULL,\n    [Name] NVARCHAR(120),\n    CONSTRAINT
[PK_Genre] PRIMARY KEY ([GenreId])\n)\n\nCREATE INDEX [IFK_InvoiceLineTrackId] ON [InvoiceLine] ([TrackId])
\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are del
ivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please gen
erate a valid SQL query without any explanations for the question. \n2. If the provided context is almost s
ufficient but requires knowledge of a specific string in a particular column, please generate an intermedia
te SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermedia
te_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please
use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the
answer exactly as it was given before. \n"}, {"role": "user", "content": " \n      List all albums and their
```

```

corresponding artist names \n"}, {"role": "assistant", "content": "SELECT A.Title, A.ArtistId, ART.Name \n
FROM Album A \nJOIN Artist ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": " \n  List al
l genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT G.Name, COUN
T(T.TrackId) AS TrackCount \nFROM Genre G \nLEFT JOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name"},
{"role": "user", "content": " \n  Get all playlists containing at least 10 tracks and the total duratio
n of those tracks:\n"}, {"role": "assistant", "content": "SELECT P.Name, SUM(T.Milliseconds) AS TotalDurati
on\nFROM PlaylistTrack PT\nJOIN Track T ON PT.TrackId = T.TrackId\nJOIN Playlist P ON PT.PlaylistId = P.Pla
ylistId\nGROUP BY P.Name\nHAVING COUNT(*) >= 10"}, {"role": "user", "content": " \n  Find the top 5 cus
tomer who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "conte
nt": "SELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums\nFROM Customer C\nJOIN InvoiceLine IL ON C.Custome
rId = IL.InvoiceId\nGROUP BY C.LastName\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content":
" \n  Find the customer who bought the most albums in total quantity (across all invoices): \n"}, {"rol
e": "assistant", "content": "SELECT C.LastName, SUM(IL.Quantity) AS TotalAlbums\nFROM Customer C\nJOIN Invo
iceLine IL ON C.CustomerId = IL.InvoiceId\nGROUP BY C.LastName\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"rol
e": "user", "content": " \n  Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"r
ole": "assistant", "content": "SELECT *\nFROM Track\nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "c
ontent": " \n  Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT
C.LastName, COUNT(I.InvoiceId) AS TotalInvoices\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.Custome
rId\nGROUP BY C.CustomerId, C.LastName\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "conten
t": " \n  Find the top 3 customers who spent the most money overall:\n"}, {"role": "assistant", "conten
t": "SELECT C.LastName, SUM(I.Total) AS TotalSpent\nFROM Customer C\nJOIN Invoice I ON C.CustomerId = I.Cus
tomerId\nGROUP BY C.LastName\nORDER BY TotalSpent DESC\nLIMIT 3\n"}, {"role": "user", "content": "what are
the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*)
AS Count\nFROM Customer\nGROUP BY Country\nORDER BY Count DESC\nLIMIT 5"}, {"role": "user", "content": " SE
LECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "assistant", "content": "SELECT * FROM t_person W
HERE name = 'John Doe'"}, {"role": "user", "content": " \n  Identify artists who have albums with track
s appearing in multiple genres:\n"}]

```

Ollama Response:

```

{'model': 'llama3:latest', 'created_at': '2024-06-08T20:25:13.114016312Z', 'message': {'role': 'assistant',
'content': '```\nSELECT A.Name, COUNT(DISTINCT T.GenreId) AS GenresCount\nFROM Artist A\nJOIN AlbumTrack AT
ON A.ArtistId = AT.ArtistId\nJOIN Track T ON AT.TrackId = T.TrackId\nGROUP BY A.Name\nHAVING COUNT(DISTINCT
T.GenreId) > 1;\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 104014703681, 'load_duratio
n': 569102, 'prompt_eval_count': 1458, 'prompt_eval_duration': 90984604000, 'eval_count': 73, 'eval_duratio
n': 12330632000}
```

```

```

SELECT A.Name, COUNT(DISTINCT T.GenreId) AS GenresCount
FROM Artist A
JOIN AlbumTrack AT ON A.ArtistId = AT.ArtistId
JOIN Track T ON AT.TrackId = T.TrackId
GROUP BY A.Name
HAVING COUNT(DISTINCT T.GenreId) > 1;
```

```

Output from LLM: ```

```
SELECT A.Name, COUNT(DISTINCT T.GenreId) AS GenresCount
FROM Artist A
JOIN AlbumTrack AT ON A.ArtistId = AT.ArtistId
JOIN Track T ON AT.TrackId = T.TrackId
GROUP BY A.Name
HAVING COUNT(DISTINCT T.GenreId) > 1;
```
```

```
Extracted SQL: SELECT A.Name, COUNT(DISTINCT T.GenreId) AS GenresCount
FROM Artist A
JOIN AlbumTrack AT ON A.ArtistId = AT.ArtistId
JOIN Track T ON AT.TrackId = T.TrackId
GROUP BY A.Name
HAVING COUNT(DISTINCT T.GenreId) > 1
SELECT A.Name, COUNT(DISTINCT T.GenreId) AS GenresCount
FROM Artist A
JOIN AlbumTrack AT ON A.ArtistId = AT.ArtistId
JOIN Track T ON AT.TrackId = T.TrackId
GROUP BY A.Name
HAVING COUNT(DISTINCT T.GenreId) > 1
```

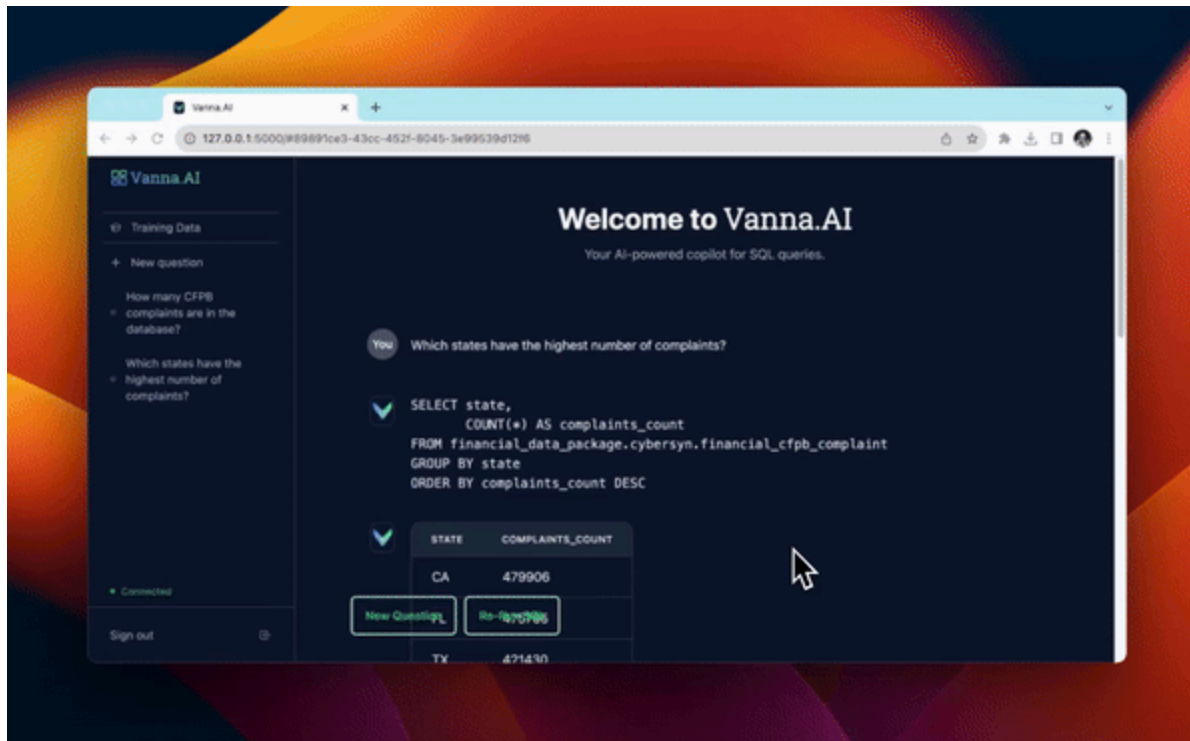
```
Couldn't run sql: Execution failed on sql 'SELECT A.Name, COUNT(DISTINCT T.GenreId) AS GenresCount
FROM Artist A
JOIN AlbumTrack AT ON A.ArtistId = AT.ArtistId
JOIN Track T ON AT.TrackId = T.TrackId
GROUP BY A.Name
HAVING COUNT(DISTINCT T.GenreId) > 1': no such table: AlbumTrack
```

In [ ]:

In [ ]:

In [ ]:

## Launch the User Interface



```
from vanna.flask import VannaFlaskApp app = VannaFlaskApp(vn) app.run()
```

## Next Steps

Using Vanna via Jupyter notebooks is great for getting started but check out additional customizable interfaces like the

- [Streamlit app](#)
- [Flask app](#)
- [Slackbot](#)