# 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?

- OpenAl via Vanna.Al (Recommended)
   Use Vanna.Al for free to generate your queries
- OpenAl
   . .

Use OpenAl with your own API key

- Azure OpenAl
  - 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. Als hosted vector database (pgvector) for free. This is usable across machines with no additional setup.
- [Selected] ChromaDB
  - Use ChromaDBs 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]: import warnings
        import re
        warnings.filterwarnings('ignore', category=DeprecationWarning, message='^Number of requested results')
        # warnings.filterwarnings('ignore', category=DeprecationWarning, message=re.escape(r'^Some regex pattern')
        import os
        import re
        from time import time
        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)
In [3]: file db = "~/Downloads/chinook.sqlite"
        model name = 'aya'
In [4]: config = {
            'model': model name, # 'mistral' # "starcoder2"
        vn = MyVanna(config=config)
In [5]: hostname = os.uname().nodename
        print("Hostname:", hostname)
       Hostname: ducklover1
```

```
In [6]: file db = os.path.abspath(os.path.expanduser(file db))
        vn.connect to sqlite(file db)
In [7]: vn.run sql is set
Out[7]: True
In [8]: | 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 [9]: def strip brackets(ddl):
            This function removes square brackets from table and column names in a DDL script.
            Args:
                 ddl (str): The DDL script containing square brackets.
             Returns:
                 str: The DDL script with square brackets removed.
             0.00
            # Use regular expressions to match and replace square brackets
            pattern = r"\setminus [([^{]}]+)]" # Match any character except ] within square brackets
            return re.sub(pattern, r"\1", ddl)
```

```
In [10]: if False:
    remove_collections()
```

## Training

## SQLite sample database

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

```
In [11]: # show training data
    training_data = vn.get_training_data()
    training_data
```

Out[11]:		id	question	content	training_data_type
	0	01c4a964-460b-5e1c-af1e- 622c8210b835-sql	\n Hint: album quantity is found in invoi	SELECT i.CustomerId, COUNT(ii.InvoiceLineId) A	sql
	1	03e56919-8e65-58f6-b8b5- 803d8078f4b7-sql	\n List all albums and their correspondin	SELECT a.Title, ar.Name AS ArtistName\nFROM "a	sql
	2	0658ba3d-98ff-51f4-9006- a24f87045858-sql	How many customers are there	SELECT COUNT(*) FROM "customers"	sql
	3	0d9348d9-1384-5029-983f- 0f4456db33b5-sql	\n Find the top 5 customers who spent th	SELECT c.CustomerId, SUM(i.Total) AS TotalSpen	sql
	4	0e1a2b7b-d65e-53de-b839- edb7afcf4ab1-sql	\n Hint: album quantity is found in invoi	SELECT i.CustomerId, COUNT(ii.TrackId) AS Tota	sql
	•••				
	19	d654f328-dc36-549e-84c3- 06ee0db7e0f7-ddl	None	CREATE TABLE "playlist_track"\r\n(\r\n Play	ddl
	20	d93f0d68-023d-5afb-8121- ba346699d318-ddl	None	CREATE TABLE "customers"\r\n(\r\n Customerl	ddl
	21	e5879308-329e-543f-a693- 0c14e2f9972e-ddl	None	CREATE INDEX IFK_InvoiceLineTrackId ON "invoic	ddl
	22	ea84418b-1a28-59b4-a1f4- 2fb674208adc-ddl	None	CREATE TABLE sqlite_sequence(name,seq)	ddl
	0	9d2550eb-8e22-54cd-9fad- 9e1be65ab03a-doc	None	In the SQLite database invoice means order	documentation

74 rows × 4 columns

```
In [12]: df_ddl = vn.run_sql("SELECT type, sql FROM sqlite_master WHERE sql is not null")
In [13]: df_ddl
```

Out[13]:		type	sql
	0	table	CREATE TABLE "albums"\r\n(\r\n [AlbumId] IN
	1	table	CREATE TABLE sqlite_sequence(name,seq)
	2	table	CREATE TABLE "artists"\r\n(\r\n [ArtistId]
	3	table	CREATE TABLE "customers"\r\n(\r\n [Customer
	4	table	CREATE TABLE "employees"\r\n(\r\n [Employee
	5	table	CREATE TABLE "genres"\r\n(\r\n [GenreId] IN
	6	table	CREATE TABLE "invoices"\r\n(\r\n [InvoiceId
	7	table	CREATE TABLE "invoice_items"\r\n(\r\n [Invo
	8	table	CREATE TABLE "media_types"\r\n(\r\n [MediaT
	9	table	CREATE TABLE "playlists"\r\n(\r\n [Playlist
	10	table	CREATE TABLE "playlist_track"\r\n(\r\n [Pla
	11	table	CREATE TABLE "tracks"\r\n(\r\n [TrackId] IN
	12	index	CREATE INDEX [IFK_AlbumArtistId] ON "albums" (
	13	index	CREATE INDEX [IFK_CustomerSupportRepId] ON "cu
	14	index	CREATE INDEX [IFK_EmployeeReportsTo] ON "emplo
	15	index	CREATE INDEX [IFK_InvoiceCustomerId] ON "invoi
	16	index	CREATE INDEX [IFK_InvoiceLineInvoiceId] ON "in
	17	index	CREATE INDEX [IFK_InvoiceLineTrackId] ON "invo
	18	index	CREATE INDEX [IFK_PlaylistTrackTrackId] ON "pl
	19	index	CREATE INDEX [IFK_TrackAlbumId] ON "tracks" ([
	20	index	CREATE INDEX [IFK_TrackGenreId] ON "tracks" ([
	21	index	CREATE INDEX [IFK_TrackMediaTypeId] ON "tracks
	22	table	CREATE TABLE sqlite_stat1(tbl,idx,stat)

```
In [14]: if False:
    for ddl in df_ddl['sql'].to_list():
```

## Asking the Al

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 [16]: ts_start = time()

SELECT name FROM sqlite_master WHERE type = 'table';
In [17]: vn.ask(question="Can you list all tables in the SQLite database catalog?")
```

```
Add of existing embedding ID: d8a2f948-dffa-5524-a5f9-174ccla8da73-sql
Add of existing embedding ID: 0658ba3d-98ff-51f4-9006-a24f87045858-sql
Add of existing embedding ID: 127fd4bd-b9af-539d-9313-1d0234d073b7-sql
Add of existing embedding ID: 32b99e7b-31ab-55d8-8431-fb010fa7af85-sql
Add of existing embedding ID: d8a2f948-dffa-5524-a5f9-174ccla8da73-sql
Add of existing embedding ID: 0658ba3d-98ff-51f4-9006-a24f87045858-sql
Add of existing embedding ID: d8a37163-5ce5-58cd-a316-ea5598d44d27-sql
Add of existing embedding ID: a7185c88-7417-5b75-a52e-4eaef5f9deca-sql
Add of existing embedding ID: 6f22268c-5062-5f11-ba2d-8555f06b409d-sql
Add of existing embedding ID: 49e67df3-a604-51f8-ad01-b8f5a2043eac-sql
Add of existing embedding ID: dd282d7c-a4ef-5e3a-87e0-cb45fac50808-sql
Add of existing embedding ID: aea89953-21b2-55d1-9dda-431ee6033c3d-sql
Add of existing embedding ID: fd25ebba-4066-5a0f-8613-7b1c2ace0339-sql
Add of existing embedding ID: 6bed484b-9a80-57f4-ad89-5f775b5df252-sql
Add of existing embedding ID: f33f8cb6-1b12-5ea7-8d9a-aef8166b9970-sql
Add of existing embedding ID: f626b681-4d8f-563a-beee-lea759baaa82-sql
Add of existing embedding ID: 127fd4bd-b9af-539d-9313-1d0234d073b7-sql
Add of existing embedding ID: 584873f8-1904-50f1-8f80-7ccf08059264-sql
Add of existing embedding ID: 3013d1b4-feb2-519d-bfb9-114500436e3d-sql
Add of existing embedding ID: d1d70c18-f5d9-5970-a32c-914deeca1087-sql
Add of existing embedding ID: e7c4b3aa-664f-5f87-8b25-449a4482f3fd-sql
Add of existing embedding ID: 9a9c970b-b94c-5f22-b54c-b86921a38b65-sql
Add of existing embedding ID: d8a2f948-dffa-5524-a5f9-174ccla8da73-sql
Add of existing embedding ID: 0658ba3d-98ff-51f4-9006-a24f87045858-sql
Add of existing embedding ID: d8a37163-5ce5-58cd-a316-ea5598d44d27-sql
Add of existing embedding ID: a7185c88-7417-5b75-a52e-4eaef5f9deca-sql
Add of existing embedding ID: 6f22268c-5062-5f11-ba2d-8555f06b409d-sql
Add of existing embedding ID: 49e67df3-a604-51f8-ad01-b8f5a2043eac-sql
Add of existing embedding ID: dd282d7c-a4ef-5e3a-87e0-cb45fac50808-sql
Add of existing embedding ID: aea89953-21b2-55d1-9dda-431ee6033c3d-sql
Add of existing embedding ID: fd25ebba-4066-5a0f-8613-7b1c2ace0339-sql
Add of existing embedding ID: 6bed484b-9a80-57f4-ad89-5f775b5df252-sql
Add of existing embedding ID: f33f8cb6-1b12-5ea7-8d9a-aef8166b9970-sql
Add of existing embedding ID: f626b681-4d8f-563a-beee-lea759baaa82-sql
Add of existing embedding ID: 127fd4bd-b9af-539d-9313-1d0234d073b7-sql
Add of existing embedding ID: 584873f8-1904-50f1-8f80-7ccf08059264-sql
Add of existing embedding ID: 3013d1b4-feb2-519d-bfb9-114500436e3d-sql
Add of existing embedding ID: e7c4b3aa-664f-5f87-8b25-449a4482f3fd-sql
Add of existing embedding ID: d8a2f948-dffa-5524-a5f9-174ccla8da73-sql
Add of existing embedding ID: a7185c88-7417-5b75-a52e-4eaef5f9deca-sql
Add of existing embedding ID: 6f22268c-5062-5f11-ba2d-8555f06b409d-sql
Add of existing embedding ID: 49e67df3-a604-51f8-ad01-b8f5a2043eac-sql
```

```
Add of existing embedding ID: 49e67df3-a604-51f8-ad01-b8f5a2043eac-sql
Add of existing embedding ID: 5b0b32a6-7f1d-544f-8c5b-9448cbc635ac-sql
Add of existing embedding ID: 584873f8-1904-50f1-8f80-7ccf08059264-sql
Add of existing embedding ID: 0e1a2b7b-d65e-53de-b839-edb7afcf4ab1-sql
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 guery 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 sqlite stat1(tbl,idx,stat)\n\nCREATE TABLE sqlite sequence(na PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n me,seq)\n\nCREATE TABLE "playlists"\r\n(\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "genres"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT Name NVARCHAR(120) $\r\n)\n\n$ CREATE TABLE "tracks" $\r\n(\r\n$ TrackId INTEGER PRIMARY KEY AUTOI NCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER.\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" Bytes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "med ia types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "media type Name NVARCHAR(120)\r\n)\n\nCR  $s"\r\n(\r\n$ MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n EATE TABLE "artists"\r\n(\r\n (120)\r\n)\n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10.2) NOT NULL,\r\n Ouantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceI d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (Track Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\cREATE TABLE "playlist track"\r\n(\r\n Plavl istId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELET (PlavlistId, TrackId),\r\n E NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELE TE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTO ArtistId INTEGER NOT NULL.\r\n INCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n== =Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the pr ovided 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 particul ar column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend t he guery with a comment saying intermediate sql \n3. If the provided context is insufficient, please explai n why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been ask ed and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'co ntent': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELEC T name FROM sqlite master WHERE type='table'"}, {'role': 'user', 'content': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'assistant', 'content': " SELECT name FROM sqlite master WHERE type='t able'"}, {'role': 'user', 'content': '\n There are 3 tables: artists, albums and tracks, where albums a Can you find the top 10 mo nd artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n st popular artists based on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = a l.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTracks DE SC\nLIMIT 10'}, {'role': 'user', 'content': '\n There are 3 tables: artists, albums and tracks, where a lbums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the to

p 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT a.Nam

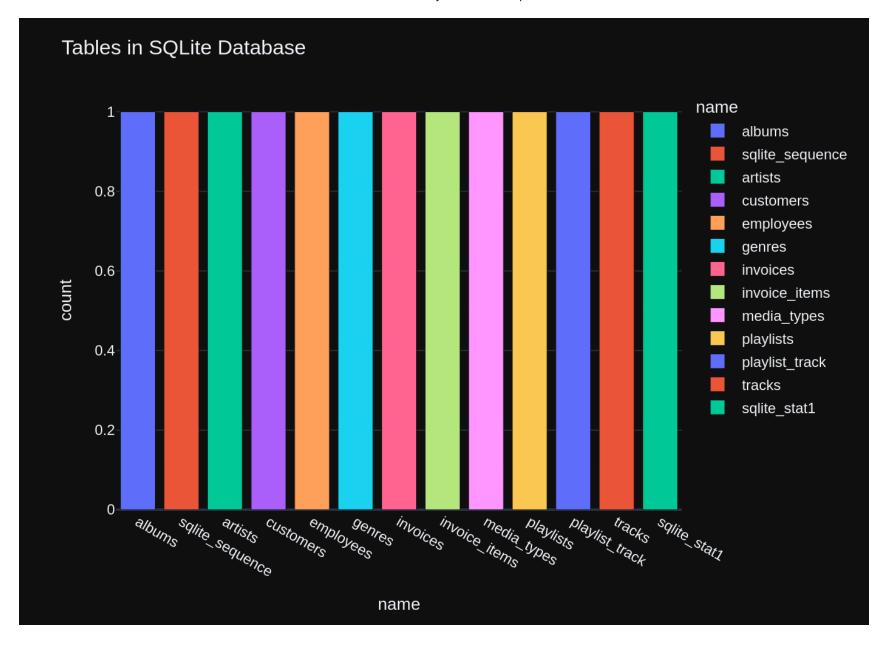
e, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "t racks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'use r', 'content': ' \n List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.Arti stId = ar.ArtistId'}, {'role': 'user', 'content': ' \n List all customers from Canada and their email addresses:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent \nFROM Customers c\nJOIN Invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSp ent DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n List all customers from Canada and their email a ddresses:\n'}, {'role': 'assistant', 'content': 'SELECT c.Email, c.Country\nFROM "customers" c\nWHERE c.Cou ntry = \'Canada\''}, {'role': 'user', 'content': ' \n List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" q\nJOIN "tracks" t ON q.GenreId = t.GenreId\nGROUP BY q.Name\nORDER BY TotalTracks DESC\nLIMIT 5'}, {'rol e': 'user', 'content': ' \n List all albums and their corresponding artist names \n'}, {'role': 'assis tant', 'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId\n-----'}, {'role': 'user', 'content': ' \n Find the customer who bou ght the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.C ustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbu ms DESC\nLIMIT 1'}, {'role': 'user', 'content': 'Can you list all tables in the SQLite database catalog?'}] Ollama parameters: model=aya:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery 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 sqlite stat1(tbl,idx,stat)\n\nCREATE TABLE sqlite sequence(na me,seq)\n\nCREATE TABLE \"playlists\"\r\n(\r\n PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREM Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY ENT NOT NULL,\r\n KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL.\r\n AlbumId INTEGER.\r\n MediaTvpeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL,\r\n Bvtes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENC ES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFER ENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeI d) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"media types\"\r\n(\r\n MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n  $HAR(120)\r\n)\n\nCREATE TABLE \"artists\"\r\n(\r\n$ ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL L.\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMA RY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERE

NCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"play list track\"\r\n(\r\n PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId).\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists \" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Title NVARCHAR(160) NOT NULL.\r\n stId INTEGER NOT NULL.\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE N O ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SOLite database invoice means order \n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query wi thout any explanations for the question. \n2. If the provided context is almost sufficient but requires kno wledge of a specific string in a particular column, please generate an intermediate SQL query to find the d istinct strings in that column. Prepend the query with a comment saying intermediate sql \n3. If the provid ed context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant tab le(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": "Can you list all tables in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite master WHERE type='table'"}, {"role": "user", "co ntent": "Can you list all tables in the SQLite database catalog?"}, {"role": "assistant", "content": " SELE CT name FROM sqlite master WHERE type='table'"}, {"role": "user", "content": " \n There are 3 tables: ar tists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based on the number of tracks\n"}, {"role": "ass istant", "content": "SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM \"artis ts\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks ar e linked by AlbumId.\n Can you find the top 10 most popular artists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nOR DER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n List all albums and their corresp onding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name AS ArtistN ame\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " List all customers from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SEL ECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent\nFROM Customers c\nJOIN Invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 1"}, {"role": "user", "content": " \n List all customers from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SELECT c.E mail, c.Country\nFROM \"customers\" c\nWHERE c.Country = 'Canada'"}, {"role": "user", "content": " \n ist all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Nam e, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY q.Name\nORDER BY TotalTracks DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all albums and thei r corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, ar.Name AS ArtistName \nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId\n-----"}, {"role": Find the customer who bought the most albums in total quantity (across all inv "user", "content": " \n

oices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM
\"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice\_items\" ii ON i.Invoice
Id = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content":
"Can you list all tables in the SQLite database catalog?"}]

Add of existing embedding ID: d8a2f948-dffa-5524-a5f9-174ccla8da73-sql Insert of existing embedding ID: d8a2f948-dffa-5524-a5f9-174ccla8da73-sql

```
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:32:32.882201431Z', 'message': {'role': 'assistant', 'c
ontent': "SELECT name FROM sqlite master WHERE type='table'"}, 'done reason': 'stop', 'done': True, 'total
duration': 58589911906, 'load duration': 2086082988, 'prompt eval count': 1660, 'prompt eval duration': 536
63694000, 'eval count': 12, 'eval duration': 2183006000}
SELECT name FROM sqlite master WHERE type='table'
SELECT name FROM sqlite master WHERE type='table'
               name
0
             albums
    sqlite_sequence
1
2
            artists
3
          customers
4
          employees
5
             genres
6
           invoices
7
      invoice items
8
        media types
9
          playlists
10
     playlist track
11
             tracks
12
       sqlite stat1
Ollama parameters:
model=aya: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: 'Can you list all tables in the SQLite database catalog?'\n\nThe
DataFrame was produced using this guery: SELECT name FROM sglite master WHERE type='table'\n\nThe following
is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n name
e: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of t
he dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataf
rame, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the cod
e."}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:32:52.822650957Z', 'message': {'role': 'assistant', 'c
ontent': "```python\nimport plotly.express as px\n\n# Create a bar chart\nfig = px.bar(df, x='name', color
='name')\n\n# Add a title to the chart\nfig.update layout(title='Tables in SQLite Database')\n\n# Display t
he chart\nfig.show()\n``"}, 'done reason': 'stop', 'done': True, 'total duration': 19912562254, 'load dura
tion': 636768, 'prompt eval count': 157, 'prompt eval duration': 5889286000, 'eval count': 71, 'eval durati
on': 13933724000}
```



```
Out[17]: ("SELECT name FROM sqlite_master WHERE type='table'",
                          name
           0
                        albums
           1
               sqlite sequence
           2
                       artists
           3
                     customers
           4
                     employees
           5
                        genres
           6
                      invoices
           7
                 invoice items
           8
                   media types
           9
                     playlists
           10
                playlist track
           11
                        tracks
           12
                  sqlite stat1,
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'name=%{x}<br/>br>count=%{y}<extra></extra>',
                          'legendgroup': 'albums',
                          'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                          'name': 'albums',
                          'offsetgroup': 'albums',
                         'orientation': 'v',
                          'showlegend': True,
                          'textposition': 'auto',
                         'type': 'bar',
                         'x': array(['albums'], dtype=object),
                         'xaxis': 'x',
                          'y': array([1]),
                         'yaxis': 'y'},
                        {'alignmentgroup': 'True',
                         'hovertemplate': 'name=%{x}<br/>br>count=%{y}<extra></extra>',
                         'legendgroup': 'sglite sequence',
                         'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
                          'name': 'sqlite sequence',
                          'offsetgroup': 'sglite sequence',
                         'orientation': 'v',
                          'showlegend': True,
                          'textposition': 'auto',
                         'type': 'bar',
                         'x': array(['sqlite_sequence'], dtype=object),
                         'xaxis': 'x',
```

```
'y': array([1]),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'name=%{x}<br>count=%{y}<extra></extra>',
 'legendgroup': 'artists',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'artists',
 'offsetgroup': 'artists',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['artists'], dtype=object),
 'xaxis': 'x',
 'y': array([1]),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'name=%{x}<br>count=%{y}<extra></extra>',
 'legendgroup': 'customers',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'customers',
 'offsetgroup': 'customers',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['customers'], dtype=object),
 'xaxis': 'x',
 'y': array([1]),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'name=%{x}<br>count=%{y}<extra></extra>',
 'legendgroup': 'employees',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'employees',
 'offsetgroup': 'employees',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['employees'], dtype=object),
 'xaxis': 'x',
```

```
'y': array([1]),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'name=%{x}<br>count=%{y}<extra></extra>',
 'legendgroup': 'genres',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'genres',
 'offsetgroup': 'genres',
 'orientation': 'v'.
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['genres'], dtype=object),
 'xaxis': 'x',
 'y': array([1]),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'name=%{x}<br>count=%{y}<extra></extra>',
 'legendgroup': 'invoices',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'invoices',
 'offsetgroup': 'invoices',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['invoices'], dtype=object),
 'xaxis': 'x',
 'y': array([1]),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'name=%{x}<br>count=%{y}<extra></extra>',
 'legendgroup': 'invoice items',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'invoice items',
 'offsetgroup': 'invoice items',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['invoice items'], dtype=object),
 'xaxis': 'x',
```

```
'y': array([1]),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'name=%{x}<br>count=%{y}<extra></extra>',
 'legendgroup': 'media types',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'media types',
 'offsetgroup': 'media types',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['media types'], dtype=object),
 'xaxis': 'x',
 'y': array([1]),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'name=%{x}<br>count=%{y}<extra></extra>',
 'legendgroup': 'playlists',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'playlists',
 'offsetgroup': 'playlists',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['playlists'], dtype=object),
 'xaxis': 'x',
 'y': array([1]),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'name=%{x}<br>count=%{y}<extra></extra>',
 'legendgroup': 'playlist track',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': 'playlist track',
 'offsetgroup': 'playlist track',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['playlist track'], dtype=object),
 'xaxis': 'x',
```

```
'y': array([1]),
         'yaxis': 'y'},
        {'alignmentgroup': 'True',
          'hovertemplate': 'name=%{x}<br>count=%{y}<extra></extra>',
          'legendgroup': 'tracks',
          'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
          'name': 'tracks',
          'offsetgroup': 'tracks',
          'orientation': 'v'.
          'showlegend': True,
          'textposition': 'auto',
          'type': 'bar',
          'x': array(['tracks'], dtype=object),
          'xaxis': 'x',
          'y': array([1]),
          'yaxis': 'y'},
        {'alignmentgroup': 'True',
          'hovertemplate': 'name=%{x}<br>count=%{y}<extra></extra>',
          'legendgroup': 'sqlite stat1',
          'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
          'name': 'sqlite stat1',
          'offsetgroup': 'sglite stat1',
          'orientation': 'v',
          'showlegend': True,
          'textposition': 'auto',
          'type': 'bar',
          'x': array(['sqlite stat1'], dtype=object),
          'xaxis': 'x',
         'y': array([1]),
          'yaxis': 'y'}],
'layout': {'barmode': 'relative',
           'legend': {'title': {'text': 'name'}, 'tracegroupgap': 0},
           'margin': {'t': 60},
           'template': '...',
           'title': {'text': 'Tables in SQLite Database'},
           'xaxis': {'anchor': 'y',
                     'categoryarray': [albums, sqlite sequence, artists,
                                       customers, employees, genres, invoices,
                                       invoice items, media types, playlists,
                                       playlist track, tracks, sqlite stat1],
                     'categoryorder': 'array',
                     'domain': [0.0, 1.0],
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery 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 "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR EMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n rv NVARCHAR(40),\r\n **FOREIG** N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n UnitPrice NUMERIC(10.2) NOT NULL.\r FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\t Quantity INTEGER NOT NULL,\r\n ON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t \tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER P RIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) N OT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n  $VARCHAR(40).\r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(24).\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL.\r\n SupportRepId INTEGER,\r\n FOREIGN KEY (Sup portRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREA EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TE TABLE "employees"\r\n(\r\n LastName NVA FirstName NVARCHAR(20) NOT NULL,\r\n RCHAR(20) NOT NULL,\r\n Title NVARCHAR(30).\r\n ReportsTo I NTEGER,\r\n BirthDate DATETIME.\r\n HireDate DATETIME.\r\n Address NVARCHAR(70).\r\n City NVARC State NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n  $HAR(40), \r\n$ Country NVARCHAR(40),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n ne NVARCHAR(24),\r\n FOREIGN KEY (ReportsTo) REFEREN CES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite se quence(name,seg)\n\nCREATE TABLE "playlists"\r\n(\r\n PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT N Name NVARCHAR(120)\r\n)\n\nCREATE TABLE sqlite stat1(tbl,idx,stat)\n\nCREATE TABLE "albums"\r\n ULL.\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n (\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\t0N DELETE rtistId INTEGER NOT NULL,\r\n NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist track"\r\n(\r\n PlavlistId INTEGER NOT NUL CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackI TrackId INTEGER NOT NULL.\r\n L,\r\n FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDA d), r nFOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\t0N DELETE NO ACTION ON UPD TE NO ACTION,\r\n ATE NO ACTION\r\n)\n\nCREATE TABLE "media types"\r\n(\r\n MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT Name NVARCHAR(120)\ $r\n$ ) $n\n$ ===Additional Context \ $n\n$ In the SQLite database invoice mean s order\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 guery 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 relev ant 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 customers who spent the mos Hint: order total can be found on invoices table, calculation using invoice t monev overall, \n \n items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total)

AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId \nORDER BY TotalSpent DESC\nLIMIT 5\n-----'}, {'role': 'user', 'content': '\n d the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on inv oices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'conte nt': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM "customers" c\nJOIN "invoices" i ON c.Custom erId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5'}, {'role': 'user', 'conte Find the top 5 customers who spent the most money overall, \n \n Hint: order total c an be found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoice s" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': Find the customer with the most invoices \n'}, {'role': 'assistant', 'conten 'user', 'content': ' \n t': 'SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER B Y NumberOfInvoices DESC\nLIMIT 1\n------'}, {'role': 'user', 'content': '\n he customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.Inv oiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c. CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': '\n Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(InvoiceId) as NumOf Invoices\nFROM "invoices"\nGROUP BY CustomerId\nORDER BY NumOfInvoices DESC\nLIMIT 1'}, {'role': 'user', 'c Find the customer who bought the most albums in total quantity (across all invoices): \n'\}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "custom ers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.Invoi ceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n t: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most album s in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUN T(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "i nvoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\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'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS To talAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId \nORDER BY TotalAlbums DESC\nLIMIT 1\n------'}, {'role': 'user', 'content': '\n t: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most album s in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUN T(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGRO UP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n-----'}, {'role': 'user', 'cont ent': "which table stores customer's orders"}] Ollama parameters: model=aya: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 \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL.\r\n InvoiceDate DATETIME NOT NULL.\r\n Billin aAddress NVARCHAR(70).\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40).\r\n BillinaCou ntry NVARCHAR(40),\r\n Total NUMERIC(10,2) NOT NULL,\r\n BillingPostalCode NVARCHAR(10),\r\n F0RE IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL TrackId INTEGER NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n L.\r\n UnitPrice NUMERIC(10.2) NO FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceI T NULL,\r\n Ouantity INTEGER NOT NULL.\r\n d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (Tra ckid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n erId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName N VARCHAR(20) NOT NULL.\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(4 0),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r\n  $ARCHAR(24), \r\n$ Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER.\r\n FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO A CTION\r\n)\n\nCREATE TABLE \"employees\"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL L.\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(3 BirthDate DATETIME.\r\n 0),\r\n ReportsTo INTEGER,\r\n HireDate DATETIME.\r\n Address NVARCHAR  $(70), \r\n$ City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode N VARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60).\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE sqlite sequence(name,seq)\n\nCREATE TABLE \"playlists\"\r\n(\r\n PlavlistId INTEGER PRI MARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE sqlite stat1(tbl,idx,stat) \n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n ARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES \"artist s\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist track\"\r\n PlavlistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n  $(\r\n$ CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackI d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"media types\"\r\n(\r\n peld INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Name NVARCHAR(120)\r\n)\n\n===Additional Context  $\n = \n \$ fficient, 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 gen erate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comme nt saying intermediate sql \n3. If the provided context is insufficient, please explain why it can't be gen erated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered befor e, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n d the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on inv oices table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "conte nt": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.Custo merId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n------

"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most money overall, \n Hint: order total can be found on invoices table, calculation using invoice items detail table is un necessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalS pending DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoices table, calculation using invoice it ems detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) A S TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Customer Id\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the customer with the m ost invoices \n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoice s \nFROM Invoices \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-----------------"}, {"role": "user", "content": " \n Find the customer with the most invoices \n"}, {"role": "assistan t", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invo ices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"r ole": "user", "content": " \n Find the customer with the most invoices \n"}, {"role": "assistant", "co ntent": "SELECT CustomerId, COUNT(InvoiceId) as NumOfInvoices\nFROM \"invoices\"\nGROUP BY CustomerId\nORDE R BY NumOfInvoices 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.Custo merId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Cust omerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbum s DESC\nLIMIT 1"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"r ole": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c \nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.Invoice Id\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n the customer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistan t", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n----------"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, Find the top 5 customers who bought the most albums in total quantity (across all invoice s):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"in voices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalA lbums DESC\nLIMIT 5\n-----"}, {"role": "user", "content": "which table stores custome r's orders"}] Ollama Response:

{'model': 'aya:latest', 'created\_at': '2024-06-14T11:34:31.47155993Z', 'message': {'role': 'assistant', 'co ntent': 'The "invoices" table in the database stores information about customer orders. Each row in the "in voices" table represents an order placed by a customer and contains details such as the invoice ID, custome r ID, invoice date, billing address, total amount, etc. The "invoices" table is linked to other tables, such as "customers," "invoice\_items," and "tracks," to provide a comprehensive view of the ordering process and associated data.'}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 98193343043, 'load\_duration': 672875, 'prompt eval count': 1962, 'prompt eval duration': 77828715000, 'eval count': 94, 'eval duration':

19645820000}

The "invoices" table in the database stores information about customer orders. Each row in the "invoices" table represents an order placed by a customer and contains details such as the invoice ID, customer ID, invoice date, billing address, total amount, etc. The "invoices" table is linked to other tables, such as "customers," "invoice\_items," and "tracks," to provide a comprehensive view of the ordering process and associated data.

The "invoices" table in the database stores information about customer orders. Each row in the "invoices" table represents an order placed by a customer and contains details such as the invoice ID, customer ID, invoice date, billing address, total amount, etc. The "invoices" table is linked to other tables, such as "customers," "invoice\_items," and "tracks," to provide a comprehensive view of the ordering process and associated data.

Couldn't run sql: Execution failed on sql 'The "invoices" table in the database stores information about c ustomer orders. Each row in the "invoices" table represents an order placed by a customer and contains deta ils such as the invoice ID, customer ID, invoice date, billing address, total amount, etc. The "invoices" t able is linked to other tables, such as "customers," "invoice\_items," and "tracks," to provide a comprehens ive view of the ordering process and associated data.': near "The": syntax error

In [19]: vn.ask(question="How many customers are there")

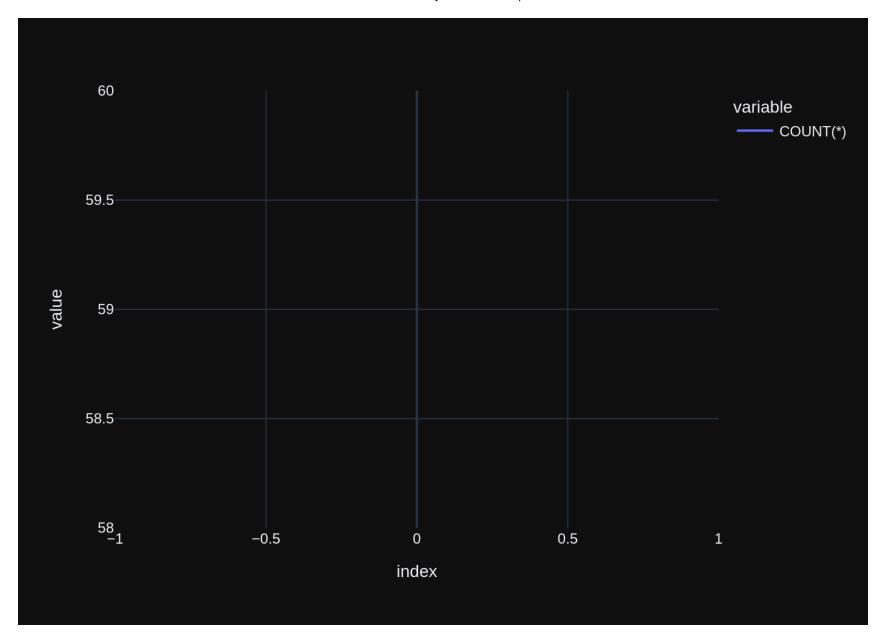
Number of requested results 10 is greater than number of elements in index 1, updating n results = 1

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery 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 "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR EMENT NOT NULL.\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n BillinaA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n rv NVARCHAR(40),\r\n **FOREIG** N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE INDEX IFK CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "customers"\r\n(\r CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL.\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n Country NVARCHAR(40),\r\n  $CHAR(40).\r\n$ State NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n one NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGE FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPD ATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE TABLE "invoic InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n e items"\r\n(\r\n InvoiceId INTEGER NO T NULL,\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Ouantity INTEGER FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON U NOT NULL,\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\t0N DELETE NO ACTION ON PDATE NO ACTION.\r\n UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON "invoice items" (InvoiceId)\n\nCREATE TAB AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT LE "albums"\r\n(\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n NULL,\r\n ArtistId INTEGER NOT NULL,\r\n \t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineTrackId ON "invoice items" EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r (TrackId)\n\nCREATE TABLE "employees"\r\n(\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(3 0), r nReportsTo INTEGER,\r\n BirthDate DATETIME.\r\n HireDate DATETIME.\r\n Address NVARCHAR City NVARCHAR(40),\r\n (70), r nState NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode N Email NVARCHAR(60).\r\n VARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n \nCREATE TABLE "playlists"\r\n(\r\n PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n  $VARCHAR(120)\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response$ Guidelines \n1. If the provided context is sufficient, please generate a valid SQL guery without any explan ations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a spec ific 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 in sufficient, 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 customer with the most invoices \n'}, {'role': 'assist ant', 'content': 'SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY Custom erId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n------'}, {'role': 'user', 'content': 'Ho w many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(\*) FROM "customers"'}, {'rol e': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(\*) A S TotalCustomers FROM "customers"'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spe

nt the most money overall. \n \n Hint: order total can be found on invoices table, calculation usin q invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SU M(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c. CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n------'}, {'role': 'user', 'content': ' Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerI d, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerI d\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n d the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on inv oices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'conte nt': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM "customers" c\nJOIN "invoices" i ON c.Custom erId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5'}, {'role': 'user', 'conte nt': ' \n Find the top 5 customers who spent the most money overall, \n \n Hint: order total c an be found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoice s" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer who bought the most albums in total quantity (across all inv oices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = i i.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': '\n Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.Customer Id, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Customer Id\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(\*) AS TotalCustomers\nFROM "customers" c \nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many custom ers are there'}] Ollama parameters: model=aya: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 \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN Billin CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n aAddress NVARCHAR(70).\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillinaCou ntry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n F0RE IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"customers \"\r\n(\r\n Customerid INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n FirstName NVARCHAR(40) NOT NU LL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70).\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(1 City NVARCHAR(40),\r\n 0), r nPhone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n Support

RepId INTEGER,\r\n FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREAT E TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n iceId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10.2) NOT NULL,\r\n Ouantity INTEGER NOT NULL.\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON D ELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\t0 N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON \"invoice items\" (InvoiceId)\n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES Title NVARCHAR(160) NOT NULL.\r\n ArtistId INTEGER NOT NULL.\r\n \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLine TrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE \"employees\"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NUL L.\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER.\r\n BirthDate DATETIME.\r\n HireDate DATETIM State NVARCHAR(40),\r\n E, r nAddress NVARCHAR(70).\r\n City NVARCHAR(40),\r\n Country NVARCHA  $R(40), \r\n$ PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVA  $RCHAR(60).\r\n$ FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlists\"\r\n(\r\n PlaylistId INTEGER PRIMARY KEY AUTOINCRE Name NVARCHAR(120) $\r\n\n\===Additional$  Context  $\n\n$ In the SQLite database invoice MENT NOT NULL,\r\n means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid S QL query without any explanations for the question. \n2. If the provided context is almost sufficient but r equires 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 guery 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 re levant 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 customer with the most invoice s \n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM In e": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(\*) F ROM \"customers\""}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "co ntent": "SELECT COUNT(\*) AS TotalCustomers FROM \"customers\""}, {"role": "user", "content": " \n the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoi ces table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "conten t": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.Custom {"role": "user", "content": " \n Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices \" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"rol e": "user", "content": " \n Find the top 5 customers who spent the most money overall, \n int: order total can be found on invoices table, calculation using invoice items detail table is unnecessar y \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM \"custom ers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most money ov

```
erall. \n
                    Hint: order total can be found on invoices table, calculation using invoice items deta
             \n
il table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalS
pent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDE
R BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n
                                                                       Find the customer who bought the mos
t albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.Customer
Id, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Custome
rId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums D
ESC\nLIMIT 1"}, {"role": "user", "content": " \n Get the total number of invoices for each customer
\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"cu
stomers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user",
"content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SEL
ECT c.Country, COUNT(*) AS TotalCustomers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomer
s DESC\nLIMIT 5"}, {"role": "user", "content": "How many customers are there"}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:35:45.803677191Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT COUNT(*) FROM "customers"'}, 'done reason': 'stop', 'done': True, 'total duration': 74278
728099, 'load duration': 1185480, 'prompt eval count': 1715, 'prompt eval duration': 71984476000, 'eval cou
nt': 9. 'eval duration': 1646219000}
SELECT COUNT(*) FROM "customers"
SELECT COUNT(*) FROM "customers"
   COUNT(*)
0
         59
Ollama parameters:
model=aya: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: 'How many customers are there'\n\nThe DataFrame was produced usin
q this query: SELECT COUNT(*) FROM \"customers\"\n\nThe following is information about the resulting panda
s DataFrame 'df': \nRunning df.dtypes gives:\n COUNT(*) int64\ndtype: object"}, {"role": "user", "conten
t": "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 o
nly Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:36:09.845967023Z', 'message': {'role': 'assistant', 'c
ontent': '```python\nimport plotly.express as px\n\n# Create a bar chart using Plotly Express\nfig = px.bar
(df, values=\'COUNT(*)\')\n\n# Add labels and title\nfig.update xaxes(title="")\nfig.update yaxes(title="Country")
unt")\nfig.update layout(title="Number of Customers")\n\n# Show the chart\nfig.show()\n```'}, 'done reaso
n': 'stop', 'done': True, 'total duration': 24022788475, 'load duration': 42487580, 'prompt eval count': 12
6, 'prompt eval duration': 4978402000, 'eval count': 92, 'eval duration': 18910877000}
```



```
Out[19]: ('SELECT COUNT(*) FROM "customers"',
              COUNT(*)
                    59,
          0
           Figure({
               'data': [{'hovertemplate': 'variable=COUNT(*)<br>index=%{x}<br>value=%{y}<extra></extra>',
                         'legendgroup': 'COUNT(*)',
                         'line': {'color': '#636efa', 'dash': 'solid'},
                         'marker': {'symbol': 'circle'},
                         'mode': 'lines',
                         'name': 'COUNT(*)',
                         'orientation': 'v',
                         'showlegend': True,
                         'type': 'scatter',
                         'x': array([0]),
                         'xaxis': 'x',
                         'y': array([59]),
                         'yaxis': 'y'}],
               'layout': {'legend': {'title': {'text': 'variable'}, 'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'index'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'value'}}}
          }))
```

#### English

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 fo rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR EMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n BillingA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n **FOREIG** rv NVARCHAR(40),\r\n N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n tName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n ddress NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n PostalCode NVARCHAR(10).\r\n Email NVARCHAR(60) NOT SupportRepId INTEGER,\r\n NULL,\r\n FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineI InvoiceId INTEGER NOT NULL.\r\n d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER N UnitPrice NUMERIC(10,2) NOT NULL,\r\n OT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (I nvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n **FOREIGN** KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE R(120)\r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "employe EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NU ReportsTo INTEGER,\r\n LL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n Βi Address NVARCHAR(70),\r\n S rthDate DATETIME,\r\n HireDate DATETIME,\r\n City NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n tate NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Phone NVARCHAR(2 FOREIGN KEY (ReportsTo) REFERENCES "employee Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n s" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n lbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION TEGER NOT NULL.\r\n PlavlistId INTEGER NOT NULL.\r\n ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist track"\r\n(\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n TrackId INTEGER NOT NULL.\r\n F0 REIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI N, r nON\r\n)\n\nCREATE TABLE sqlite sequence(name, seq)\n\nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIM ARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n peId INTEGER NOT NULL.\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFE RENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REF ERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeI d) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Addi tional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provide d context is sufficient, please generate a valid SQL guery without any explanations for the guestion. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular co lumn, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the qu

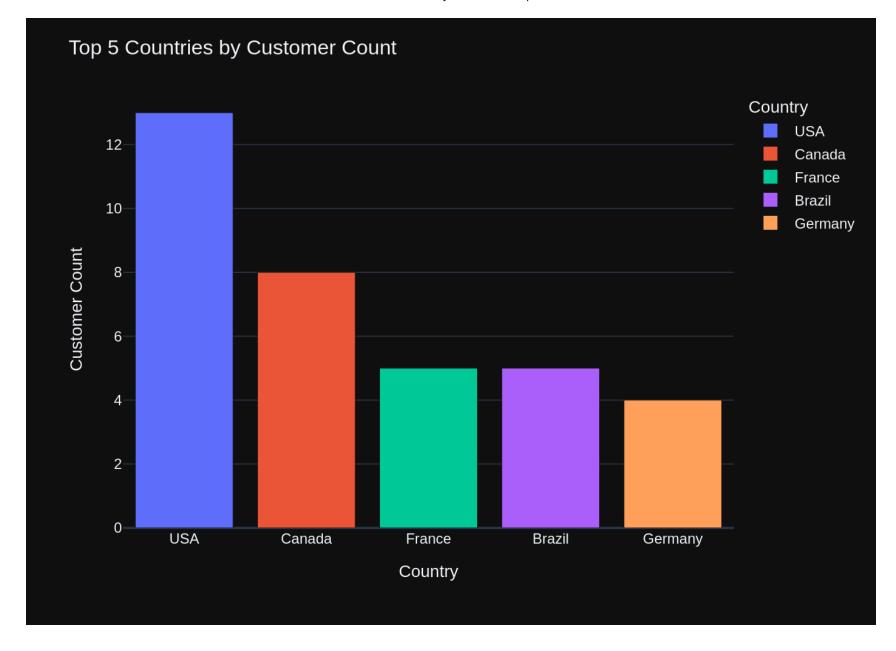
ery with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the guestion has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT c.Count ry, COUNT(\*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money overall, \n Hint: order total can be found on invoices table, calculation using invoice items detail table is un necessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "cu stomers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DE SC\nLIMIT 5\n-----'}, {'role': 'user', 'content': ' \n Find the top 5 customers wh o spent the most money overall, \n \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerI d, SUM(i.Total) AS TotalSpending\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGRO UP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n top 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n List all customers from Canada and their email addresses:\n'}, {'role': 'assistant', 'content': 'SELECT c.E mail, c.Country\nFROM "customers" c\nWHERE c.Country = \'Canada\''}, {'role': 'user', 'content': ' \n ind the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT Customer.Countr y, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM "customers" customer\nJOIN "invoices" invoice ON custome r.CustomerId = invoice.CustomerId\nGROUP BY Country'}, {'role': 'user', 'content': ' \n Hint: album qua ntity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total q uantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackI d) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Cus tomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n------'}, {'role': 'user', 'content': '\n Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(I nvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-----'}, {'role': 'user', 'content': '\n Hint: album quantity is found in invoi Find the top 5 customers who bought the most albums in total quantity (across all inv ce items. \n \n oices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = i i.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT (i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJ0IN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n top 5 countries that customers come from?\n'\}] Ollama parameters: model=aya:latest, options={}, keep alive=None

#### Prompt Content:

[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery 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 \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL.\r\n InvoiceDate DATETIME NOT NULL.\r\n Billin gAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCou ntry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n F0RE IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL, $\r\n$ Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) SupportRepId INTEGER,\r\n \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n ineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGE UnitPrice NUMERIC(10.2) NOT NULL.\r\n R NOT NULL.\r\n Ouantity INTEGER NOT NULL.\r\n FOREIGN KE Y (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION,\r\n OREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n \nCREATE TABLE \"media types\"\r\n(\r\n MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n ame NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE T EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n ABLE \"employees\"\r\n(\r\n LastName NVARC HAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n ReportsTo INT Title NVARCHAR(30),\r\n Address NVARCHAR(70).\r\n City NVARCHA EGER,\r\n BirthDate DATETIME,\r\n HireDate DATETIME.\r\n State NVARCHAR(40),\r\n Phone  $R(40), \r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n  $NVARCHAR(24).\r\n$ Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums \"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r ArtistId INTEGER NOT NULL.\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\t0 N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist track\"\r\n(\r\n PlavlistId INTEG CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistI ER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACT d. TrackId).\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\t0N DELETE NO ION ON UPDATE NO ACTION,\r\n ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite sequence(name, seq)\n\nCREATE TABLE \"tracks\"\r\n(\r TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Name NVARCHAR(200) NOT NULL,\r\n Id INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Milliseconds INTEGER NOT NULL.\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidel ines \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 st

ring in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the guery with a comment saying intermediate sql \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 guest ion has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"ro le": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "c ontent": "SELECT c.Country, COUNT(\*) AS TotalCustomers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "\n Find the top 5 customers who spent th Hint: order total can be found on invoices table, calculation using inv e most money overall. \n \n oice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.T otal) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.C ustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n-----"}, {"role": "user", "content": " Find the top 5 customers who spent the most money overall. \n \n Hint: order total can be fo und on invoices table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistan t", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most money overall, \n order total can be found on invoices table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIM IT 5"}, {"role": "user", "content": " \n List all customers from Canada and their email addresse s:\n"}, {"role": "assistant", "content": "SELECT c.Email, c.Country\nFROM \"customers\" c\nWHERE c.Country = 'Canada'"}, {"role": "user", "content": " \n Find the total number of invoices per country:\n"}, {"ro le": "assistant", "content": "SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM \"cus tomers\" customer\nJOIN \"invoices\" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Countr y"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assi stant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invo ice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n-------"}, {"role": "user", "content": " \n Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invo "user", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 cust omers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "conte nt": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJ0IN \"invoice items\" ii 0N i.InvoiceId = ii.InvoiceId\nGR0UP BY c.CustomerId \nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the customer with the mo st invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoice s\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER B Y TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n what are the top 5 countries that cus tomers come from?\n"}] Ollama Response: {'model': 'aya:latest', 'created at': '2024-06-14T11:37:36.246359462Z', 'message': {'role': 'assistant', 'c

```
ontent': 'SELECT Country, COUNT(CustomerId) AS CustomerCount\nFROM "customers"\nGROUP BY Country\nORDER BY
CustomerCount DESC\nLIMIT 5'}, 'done reason': 'stop', 'done': True, 'total duration': 86299283705, 'load du
ration': 750454, 'prompt eval count': 1989, 'prompt eval duration': 79197813000, 'eval count': 31, 'eval du
ration': 6289540000}
SELECT Country, COUNT(CustomerId) AS CustomerCount
FROM "customers"
GROUP BY Country
ORDER BY CustomerCount DESC
LIMIT 5
SELECT Country, COUNT(CustomerId) AS CustomerCount
FROM "customers"
GROUP BY Country
ORDER BY CustomerCount DESC
LIMIT 5
   Country CustomerCount
0
      USA
                       13
                        8
1
  Canada
2 France
                        5
3 Brazil
4 Germany
Ollama parameters:
model=aya: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 what are the top 5 countries that customers come from?
\n'\n\nThe DataFrame was produced using this query: SELECT Country, COUNT(CustomerId) AS CustomerCount\nFRO
M \"customers\"\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5\n\nThe following is information abo
ut the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Country
                                                                                     object\nCustomerCount
int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the r
esults of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value i
n the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- ju
st the code."}l
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:38:05.208271662Z', 'message': {'role': 'assistant', 'c
ontent': "```python\nimport plotly.express as px\n\n# Create a bar chart\nfiq = px.bar(df, x='Country', y
='CustomerCount', color='Country')\n\n# Add labels and title\nfig.update xaxes(title='Country')\nfig.update
yaxes(title='Customer Count')\nfiq.update layout(title='Top 5 Countries by Customer Count')\n\n# Show the
chart\nfig.show()\n```"}, 'done reason': 'stop', 'done': True, 'total duration': 28936353168, 'load duratio
n': 706025, 'prompt eval count': 185, 'prompt eval duration': 7681698000, 'eval count': 102, 'eval duratio
n': 21119977000}
```



```
Out[20]: ('SELECT Country, COUNT(CustomerId) AS CustomerCount\nFROM "customers"\nGROUP BY Country\nORDER BY Custome
          rCount DESC\nLIMIT 5',
             Country CustomerCount
          0
                 USA
                                  13
           1 Canada
                                   8
           2 France
                                   5
           3 Brazil
          4 Germany
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'Country=%{x}<br>CustomerCount=%{y}<extra></extra>',
                         'legendgroup': 'USA',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                         'name': 'USA',
                         'offsetgroup': 'USA',
                         'orientation': 'v',
                         'showlegend': True,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array(['USA'], dtype=object),
                         'xaxis': 'x',
                         'y': array([13]),
                         'yaxis': 'y'},
                        {'alignmentgroup': 'True',
                         'hovertemplate': 'Country=%{x}<br>CustomerCount=%{y}<extra></extra>',
                         'legendgroup': 'Canada',
                         'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
                         'name': 'Canada',
                         'offsetgroup': 'Canada',
                         'orientation': 'v',
                         'showlegend': True,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array(['Canada'], dtype=object),
                         'xaxis': 'x',
                         'y': array([8]),
                         'yaxis': 'y'},
                        {'alignmentgroup': 'True',
                         'hovertemplate': 'Country=%{x}<br>CustomerCount=%{y}<extra></extra>',
                         'legendgroup': 'France',
                         'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
                         'name': 'France',
```

```
'offsetgroup': 'France',
          'orientation': 'v',
          'showlegend': True,
          'textposition': 'auto',
          'type': 'bar',
          'x': array(['France'], dtype=object),
          'xaxis': 'x',
          'y': array([5]),
          'yaxis': 'y'},
        {'alignmentgroup': 'True',
          'hovertemplate': 'Country=%{x}<br>CustomerCount=%{y}<extra></extra>',
          'legendgroup': 'Brazil',
          'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
          'name': 'Brazil',
          'offsetgroup': 'Brazil',
          'orientation': 'v',
          'showlegend': True,
          'textposition': 'auto',
          'type': 'bar',
          'x': array(['Brazil'], dtype=object),
          'xaxis': 'x',
          'y': array([5]),
          'yaxis': 'y'},
         {'alignmentgroup': 'True',
          'hovertemplate': 'Country=%{x}<br>CustomerCount=%{y}<extra></extra>',
          'legendgroup': 'Germany',
          'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
          'name': 'Germany',
          'offsetgroup': 'Germany',
          'orientation': 'v',
          'showlegend': True,
          'textposition': 'auto',
          'type': 'bar',
          'x': array(['Germany'], dtype=object),
          'xaxis': 'x',
          'y': array([4]),
          'yaxis': 'y'}],
'layout': {'barmode': 'relative',
           'legend': {'title': {'text': 'Country'}, 'tracegroupgap': 0},
           'margin': {'t': 60},
           'template': '...',
           'title': {'text': 'Top 5 Countries by Customer Count'},
```

## German

```
In [21]: question = """
            aus welchen fünf Ländern kommen die meisten Kunden?
"""
            vn.ask(question=question)
```

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

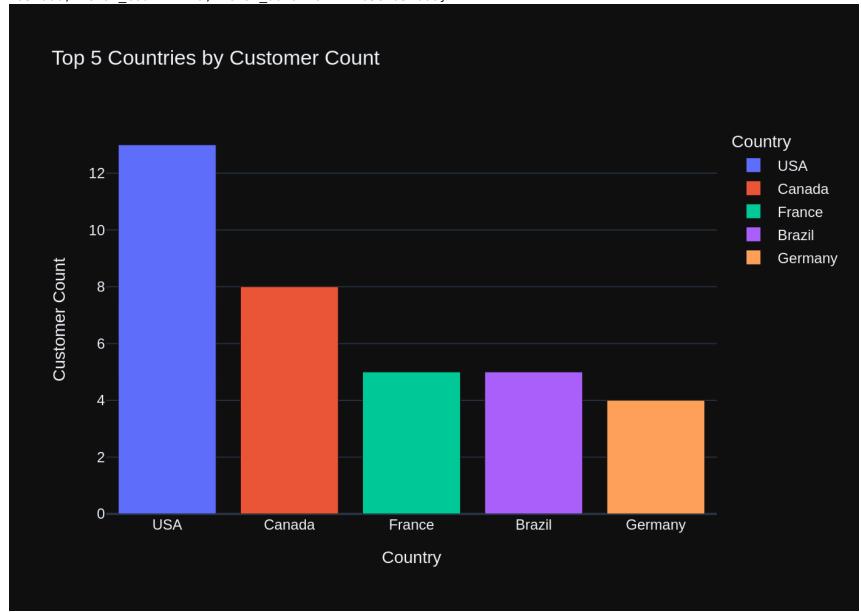
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions.  $\n==$ Tables  $\nCREATE$  TABLE sqlite stat1(tbl,idx,stat) $\nCREATE$  TABLE "customers" $\nCREATE$  TABLE sqlite stat1(tbl,idx,stat) $\nCREATE$  TABLE sqlite stat1(tbl,idx,stat1) $\nCREATE$ FirstName NVARCHAR(40) NOT NULL,\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n Country NVARCHAR(40),\r\n  $CHAR(40), \r\n$ State NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n one NVARCHAR(24),\r\n SupportRepId INTEGE FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\t0N DELETE NO ACTION ON UPD ATE NO ACTION\r\n)\n\nCREATE INDEX IFK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDEX IFK PlaylistTrack TrackId ON "playlist track" (TrackId)\n\nCREATE INDEX IFK CustomerSupportRepId ON "customers" (SupportRepI d)\n\nCREATE INDEX IFK TrackGenreId ON "tracks" (GenreId)\n\nCREATE TABLE "tracks"\r\n(\r\n TrackId INTE GER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Milliseconds I Composer NVARCHAR(220),\r\n NTEGER NOT NULL,\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (Album Id) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Genr eId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Med iaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCR EATE TABLE sqlite sequence(name, seq)\n\nCREATE INDEX IFK AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TAB EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR LE "employees"\r\n(\r\n (20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGE City NVARCHAR(4  $R_{i} r n$ BirthDate DATETIME.\r\n HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n State NVARCHAR(40),\r\n 0), r nCountry NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NV Email NVARCHAR(60),\r\n  $ARCHAR(24), \r\n$ Fax NVARCHAR(24),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n \nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is suff icient, please generate a valid SQL query without any explanations for the question. \n2. If the provided c ontext is almost sufficient but requires knowledge of a specific string in a particular column, please gene rate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a commen t saying intermediate sql \n3. If the provided context is insufficient, please explain why it can\'t be gen erated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered befor e, 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 c.Country, COUNT(\*) A S TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': what are the top 5 countries that customers come from?\n'}, {'role': 'assistan 'user', 'content': ' \n t', 'content': 'SELECT Country, COUNT(CustomerId) AS CustomerCount\nFROM "customers"\nGROUP BY Country\nORD ER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n List all customers from Canada a nd their email addresses:\n'\}, \{'role': 'assistant', 'content': 'SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent\nFROM Customers c\nJOIN Invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORD ER BY TotalSpent DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n List all customers from Canada and their email addresses:\n'}, {'role': 'assistant', 'content': 'SELECT c.Email, c.Country\nFROM "customers" c \nWHERE c.Country = \'Canada\''}, {'role': 'user', 'content': '\n Hint: album quantity is found in inv oice items, \n Find the top 5 customers who bought the most albums in total quantity (across all i nvoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFRO
M "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY Tota

lAlbums DESC\nLIMIT 5\n-----'}, {'role': 'user', 'content': '\n Find the top 5 mos t expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT Name, UnitPrice\nFR OM "tracks"\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the top 5 most e xpensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.U nitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5\n-----'}, {'role': 'use Find the top 5 customers who spent the most money overall, \n r', 'content': '\n er total can be found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n---------'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(\*) AS TotalCustomers FROM "customers"'}, {'role': 'user', 'content': ' \n t: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most album s in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUN T(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "i nvoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n aus welchen fünf Ländern kommen die meisten Kunden?\n'}] Ollama parameters: model=aya:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery 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 sqlite stat1(tbl,idx,stat)\n\nCREATE TABLE \"customers\"\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL.\r (\r\n Address NVARCHAR(70),\r\n LastName NVARCHAR(20) NOT NULL.\r\n Company NVARCHAR(80),\r\n v NVARCHAR(40).\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEG FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDEX IFK Playlist TrackTrackId ON \"playlist track\" (TrackId)\n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (Sup portRepId)\n\nCREATE INDEX IFK TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n EGER,\r\n Composer NVARCHAR(220),\r\n М illiseconds INTEGER NOT NULL,\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREI GN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n REIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite sequence(name, seq)\n\nCREATE INDEX IFK AlbumArtistId ON \"albums\" (Arti stId)\n\nCREATE TABLE \"employees\"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n

LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER,\r\n BirthDate DATETIME.\r\n HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n Country NVARCHAR(40),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n PostalCode NVARCHAR(1 Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n 0),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (Repo rtsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===A dditional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the prov ided context is sufficient, please generate a valid SQL query without any explanations for the question. \n 2. 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 w hy it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked a nd answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "conten t": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT c.C ountry, COUNT(\*) AS TotalCustomers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC \nLIMIT 5"}, {"role": "user", "content": " \n what are the top 5 countries that customers come from? \n"}, {"role": "assistant", "content": "SELECT Country, COUNT(CustomerId) AS CustomerCount\nFROM \"customer s\"\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": " \n ll customers from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SELECT c.Custome rId, c.Email, SUM(i.Total) AS TotalSpent\nFROM Customers c\nJOIN Invoices i ON c.CustomerId = i.CustomerId \nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 1"}, {"role": "user", "content": " \n l customers from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SELECT c.Email, c.Country\nFROM \"customers\" c\nWHERE c.Country = 'Canada'"}, {"role": "user", "content": " \n lbum quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii. TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROU P BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n------"}, {"role": "user", "conte Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "conte nt": "SELECT Name, UnitPrice\nFROM \"tracks\"\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "conten Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "conten t": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5\n-----------"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most mone v overall, \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS To  $talSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\n$ ORDER BY TotalSpent DESC\nLIMIT 5\n-----"}, {"role": "user", "content": "How many custo mers are there"}, {"role": "assistant", "content": "SELECT COUNT(\*) AS TotalCustomers FROM \"customers\""}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n op 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistan t", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoice s\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.C ustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n aus welchen f\u00fcnf L\u00e4ndern kommen die meisten Kunden?\n"}]

```
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:39:18.803521554Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT c.Country, COUNT(*) AS CustomerCount\nFROM "customers" c\nGROUP BY c.Country\nORDER BY Cus
tomerCount DESC\nLIMIT 5'}, 'done reason': 'stop', 'done': True, 'total_duration': 73491430715, 'load_durat
ion': 712886, 'prompt eval count': 1580, 'prompt eval duration': 65919171000, 'eval count': 33, 'eval durat
ion': 6877546000}
SELECT c.Country, COUNT(*) AS CustomerCount
FROM "customers" c
GROUP BY c.Country
ORDER BY CustomerCount DESC
LIMIT 5
SELECT c.Country, COUNT(*) AS CustomerCount
FROM "customers" c
GROUP BY c.Country
ORDER BY CustomerCount DESC
LIMIT 5
   Country CustomerCount
      USA
0
                       13
                        8
   Canada
1
2 France
                        5
3 Brazil
4 Germany
Ollama parameters:
model=aya: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 aus welchen f\u00fcnf L\u00e4ndern kommen die meisten Ku
nden?\n'\nThe DataFrame was produced using this query: SELECT c.Country, COUNT(*) AS CustomerCount\nFROM
\"customers\" c\nGROUP BY c.Country\nORDER BY CustomerCount DESC\nLIMIT 5\n\nThe following is information a
                                                                                       object\nCustomerCou
bout the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Country
       int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to char
t 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 explanation
s -- just the code."}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:39:43.694534215Z', 'message': {'role': 'assistant', 'c
ontent': '```python\nimport plotly.express as px\n\n# Create a bar chart\nfiq = px.bar(df, x="Country", y
="CustomerCount", color=\'Country\', title=\'Top 5 Countries by Customer Count\')\nfig.update layout(xaxis
title="Country", yaxis title="Customer Count")\nfig.show()\n```'}, 'done reason': 'stop', 'done': True, 'to
```

tal\_duration': 24870558289, 'load\_duration': 914418, 'prompt\_eval\_count': 185, 'prompt\_eval\_duration': 8153
957000, 'eval count': 79, 'eval duration': 16581957000}



```
Out[21]: ('SELECT c.Country, COUNT(*) AS CustomerCount\nFROM "customers" c\nGROUP BY c.Country\nORDER BY CustomerCo
         unt DESC\nLIMIT 5',
             Country CustomerCount
          0
                 USA
                                  13
           1 Canada
                                   8
           2 France
                                   5
           3 Brazil
          4 Germany
                                   4,
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'Country=%{x}<br>CustomerCount=%{y}<extra></extra>',
                         'legendgroup': 'USA',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                         'name': 'USA',
                         'offsetgroup': 'USA',
                         'orientation': 'v',
                         'showlegend': True,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array(['USA'], dtype=object),
                         'xaxis': 'x',
                         'y': array([13]),
                         'yaxis': 'y'},
                        {'alignmentgroup': 'True',
                         'hovertemplate': 'Country=%{x}<br>CustomerCount=%{y}<extra></extra>',
                         'legendgroup': 'Canada',
                         'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
                         'name': 'Canada',
                         'offsetgroup': 'Canada',
                         'orientation': 'v',
                         'showlegend': True,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array(['Canada'], dtype=object),
                         'xaxis': 'x',
                         'y': array([8]),
                         'yaxis': 'y'},
                        {'alignmentgroup': 'True',
                         'hovertemplate': 'Country=%{x}<br>CustomerCount=%{y}<extra></extra>',
                         'legendgroup': 'France',
                         'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
                         'name': 'France',
```

```
'offsetgroup': 'France',
          'orientation': 'v',
          'showlegend': True,
          'textposition': 'auto',
          'type': 'bar',
          'x': array(['France'], dtype=object),
          'xaxis': 'x',
          'y': array([5]),
          'yaxis': 'y'},
         {'alignmentgroup': 'True',
          'hovertemplate': 'Country=%{x}<br>CustomerCount=%{y}<extra></extra>',
          'legendgroup': 'Brazil',
          'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
          'name': 'Brazil',
          'offsetgroup': 'Brazil',
          'orientation': 'v',
          'showlegend': True,
          'textposition': 'auto',
          'type': 'bar',
          'x': array(['Brazil'], dtype=object),
          'xaxis': 'x',
          'y': array([5]),
          'yaxis': 'y'},
         {'alignmentgroup': 'True',
          'hovertemplate': 'Country=%{x}<br>CustomerCount=%{y}<extra></extra>',
          'legendgroup': 'Germany',
          'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
          'name': 'Germany',
          'offsetgroup': 'Germany',
          'orientation': 'v',
          'showlegend': True,
          'textposition': 'auto',
          'type': 'bar',
          'x': array(['Germany'], dtype=object),
          'xaxis': 'x',
          'y': array([4]),
          'yaxis': 'y'}],
'layout': {'barmode': 'relative',
           'legend': {'title': {'text': 'Country'}, 'tracegroupgap': 0},
           'template': '...',
           'title': {'text': 'Top 5 Countries by Customer Count'},
           'xaxis': {'anchor': 'y',
```

## Chinese

```
In [22]: question = """
顾客来自的前 5 个国家是哪些?
"""
vn.ask(question=question)
```

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

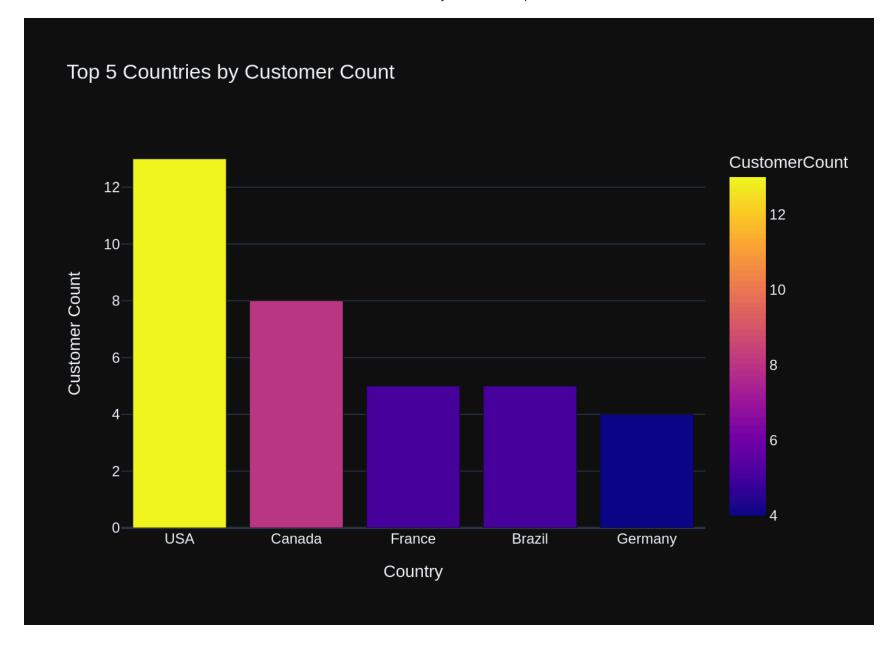
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL.\r\n State NVARCHAR(40),\r\n ompany NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n Phone NVARCHAR(24),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Fax NVARCHAR(24).\r \n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n FOREIGN KEY (SupportRepId) REFERENC ES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice i InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n InvoiceId INTEGER NOT N ULL.\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Ouantity INTEGER N FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UP OT NULL,\r\n DATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON U PDATE NO ACTION\r\n)\n\nCREATE TABLE "employees"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT N OT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL.\r\n Title NVAR BirthDate DATETIME,\r\n  $CHAR(30), \r\n$ ReportsTo INTEGER,\r\n HireDate DATETIME.\r\n Address NVA  $RCHAR(70).\r\n$ City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalC ode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n F0R EIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n \n)\n\nCREATE TABLE "invoices"\r\n(\r\n tomerId INTEGER NOT NULL.\r\n InvoiceDate DATETIME NOT NULL,\r\n BillingAddress NVARCHAR(70),\r\n BillingCountry NVARCHAR(40),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n Bill ingPostalCode NVARCHAR(10),\r\n
Total NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (CustomerId) REFERENCE S "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite sequ ence(name, seg)\n\nCREATE TABLE "albums"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL.\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist trac PlavlistId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL.\r\n CONSTRAINT PK Plavlis  $k"\r\n(\r\n$ tTrack PRIMARY KEY (PlaylistId, TrackId).\r\n FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Playlist FOREIGN KEY (TrackId) REFERENCES "tracks" (Trac Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n kId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON "invoice s" (CustomerId)\n\nCREATE INDEX IFK CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON "invoice items" (InvoiceId) $\n\n\n===Additional Context \n\nIn the SQLite database i$ nvoice means order\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 sufficien t but requires knowledge of a specific string in a particular column, please generate an intermediate SQL g uery 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 th e 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 aus welchen fünf Ländern kommen die meisten Kunden?\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(\*) AS CustomerCount\nFRO M "customers" c\nGROUP BY c.Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': ' what are the top 5 countries that customers come from?\n'}, {'role': 'assistant', 'content': 'SELECT

Country, COUNT(CustomerId) AS CustomerCount\nFROM "customers"\nGROUP BY Country\nORDER BY CustomerCount DES C\nLIMIT 5'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'rol e': 'assistant', 'content': 'SELECT c.Country, COUNT(\*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.C ountry\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(\*) AS TotalI nvoices\nFROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': ' \n Find the tota l number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(i.Inv oiceId) AS TotalInvoices\nFROM "invoices" i\nGROUP BY BillingCountry'}, {'role': 'user', 'content': ' \n Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT Customer.Count ry, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM "customers" customer\nJOIN "invoices" invoice ON custome r.CustomerId = invoice.CustomerId\nGROUP BY Country'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money overall. \n \n Hint: order total can be found on invoices table. calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Cust omerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n Find the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'co ntent': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.Custom erId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'conten t': ' \n Find the top 5 customers who spent the most money overall, \n \n Hint: order total ca n be found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'a ssistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n------------'}, {'role': 'user', 'content': ' \n List all customers from Canada and their email addresse s:\n'}, {'role': 'assistant', 'content': 'SELECT c.Email, c.Country\nFROM "customers" c\nWHERE c.Country = \'Canada\''}, {'role': 'user', 'content': ' \n 顾客来自的前 5 个国家是哪些?\n'}] Ollama parameters: model=ava: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 \"customers\"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTO INCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n State NVARCHAR(40), \r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24).\r SupportRepId INTEGER,\r\n Email NVARCHAR(60) NOT NULL,\r\n FOREIGN KEY (SupportRepId) REFERENC ES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoic InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n e items\"\r\n(\r\n InvoiceId INTEGER N OT NULL,\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Ouantity INTEGE FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION R NOT NULL,\r\n

ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTI ON ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"employees\"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Т BirthDate DATETIME.\r\n Ad itle NVARCHAR(30).\r\n ReportsTo INTEGER.\r\n HireDate DATETIME.\r\n dress NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Phone NVARCHAR(24),\r\n PostalCode NVARCHAR(10).\r\n Fax NVARCHAR(24),  $\r\n$ Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI ON\r\n)\n\nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n BillingAddress NVARCHAR(70).\r CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n BillingCountry NVARCHAR(40),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n Total NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (CustomerId) REFER BillingPostalCode NVARCHAR(10).\r\n ENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlit e sequence(name, seq)\n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT N ArtistId INTEGER NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"pl PlavlistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n avlist track\"\r\n(\r\n CONSTRAIN T PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlis ts\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustom erId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId) \n\nCREATE INDEX IFK InvoiceLineInvoiceId ON \"invoice items\" (InvoiceId)\n\n\n===Additional Context \n\nI n the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is suffici ent, please generate a valid SQL guery without any explanations for the guestion. \n2. If the provided cont ext is almost sufficient but requires knowledge of a specific string in a particular column, please generat e 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 generat ed. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, pl ease repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n en f\u00fcnf L\u00e4ndern kommen die meisten Kunden?\n"}, {"role": "assistant", "content": "SELECT c.Countr y, COUNT(\*) AS CustomerCount\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": " \n what are the top 5 countries that customers come from?\n"}, {"rol e": "assistant", "content": "SELECT Country, COUNT(CustomerId) AS CustomerCount\nFROM \"customers\"\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "what are the top 5 countri es that customers come from?"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(\*) AS TotalCustom ers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "c ontent": " \n Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SEL ECT i.BillingCountry, COUNT(\*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "content": " \n Find the total number of invoices per country:\n"}, {"role": "assistant", "cont ent": "SELECT BillingCountry, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY BillingCou ntry"}, {"role": "user", "content": " \n Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM \"customer s\" customer\nJOIN \"invoices\" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country"},

```
{"role": "user", "content": " \n Find the top 5 customers who spent the most money overall, \n
Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessa
ry \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM \"custo
mers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending
DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most money ov
          \n Hint: order total can be found on invoices table, calculation using invoice items deta
erall, \n
il table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalS
pent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDE
R BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n
                                                                     Find the top 5 customers who spent t
he most money overall, \n \n Hint: order total can be found on invoices table, calculation using in
voice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.
Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.
CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n------"}, {"role": "user", "content": "
      List all customers from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SEL
ECT c.Email, c.Country\nFROM \"customers\" c\nWHERE c.Country = 'Canada'"}, {"role": "user", "content": "
     \u987e\u5ba2\u6765\u81ea\u7684\u524d\ 5\ \u4e2a\u56fd\u5bb6\u662f\u54ea\u4e9b\uff1f\n"}
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:41:07.00523115Z', 'message': {'role': 'assistant', 'co
ntent': 'SELECT c.Country, COUNT(*) AS CustomerCount\nFROM "customers" c\nGROUP BY c.Country\nORDER BY Cust
omerCount DESC\nLIMIT 5;'}, 'done reason': 'stop', 'done': True, 'total duration': 83207475328, 'load durat
ion': 776415, 'prompt eval count': 1789, 'prompt eval duration': 75628625000, 'eval count': 34, 'eval durat
ion': 6881480000}
SELECT c.Country, COUNT(*) AS CustomerCount
FROM "customers" c
GROUP BY c.Country
ORDER BY CustomerCount DESC
LIMIT 5:
Output from LLM: SELECT c.Country, COUNT(*) AS CustomerCount
FROM "customers" c
GROUP BY c.Country
ORDER BY CustomerCount DESC
LIMIT 5:
Extracted SQL: SELECT c.Country, COUNT(*) AS CustomerCount
FROM "customers" c
GROUP BY c.Country
ORDER BY CustomerCount DESC
LIMIT 5
SELECT c.Country, COUNT(*) AS CustomerCount
FROM "customers" c
GROUP BY c.Country
ORDER BY CustomerCount DESC
LIMIT 5
```

```
Country CustomerCount
       USA
0
                       13
1
                        8
   Canada
                        5
2
  France
                        5
   Brazil
                        4
4 Germany
Ollama parameters:
model=aya: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 \u987e\u5ba2\u6765\u81ea\u7684\u524d 5 \u4e2a\u56fd\u5bb
6\u662f\u54ea\u4e9b\uff1f\n'\nThe DataFrame was produced using this guery: SELECT c.Country, COUNT(*) AS
CustomerCount\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY CustomerCount DESC\nLIMIT 5\n\nThe follow
ing is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Country
                         int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python
object\nCustomerCount
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 wi
th any explanations -- just the code."}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:41:38.822559028Z', 'message': {'role': 'assistant', 'c
ontent': '```python\nimport plotly.express as px\n\n# Create a bar chart\nfiq = px.bar(df, x="Country", y
="CustomerCount", color="CustomerCount",\n
                                                       title="Top 5 Countries by Customer Count")\n\n# Add
customizability to the figure\nfig.update xaxes(title="Country")\nfig.update yaxes(title="Customer Count")
\nig.update layout(barmode=\'group\', legend=dict(title="Country"))\n\n# Show the plot\nfig.show()\n\``'},
'done reason': 'stop', 'done': True, 'total duration': 31792382782, 'load duration': 693804, 'prompt eval c
ount': 188, 'prompt eval duration': 7249251000, 'eval count': 121, 'eval duration': 24406708000}
```



```
Out[22]: ('SELECT c.Country, COUNT(*) AS CustomerCount\nFROM "customers" c\nGROUP BY c.Country\nORDER BY CustomerCo
         unt DESC\nLIMIT 5',
            Country CustomerCount
          0
                USA
                               13
                                8
          1 Canada
          2 France
                                5
          3 Brazil
          4 Germany
                                4,
          Figure({
             'data': [{'alignmentgroup': 'True',
                       'hovertemplate': 'Country=%{x}<br/>customerCount=%{marker.color}<extra></extra>',
                       'legendgroup': '',
                       'marker': {'color': array([13, 8, 5, 5, 4]), 'coloraxis': 'coloraxis', 'pattern': {'sha
         pe': ''}},
                       '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': 'group',
                        'coloraxis': {'colorbar': {'title': {'text': 'CustomerCount'}},
                                     'colorscale': [[0.0, '#0d0887'], [0.1111111111111111,
                                                   '#46039f'], [0.22222222222222,
                                                   '#bd3786'], [0.555555555555556,
                                                   '#ed7953'], [0.7777777777778,
                                                   '#fb9f3a'], [0.88888888888888888,
                                                   '#fdca26'], [1.0, '#f0f921']]},
                        'legend': {'title': {'text': 'Country'}, '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': 'Customer Count'}}}
         }))
```

# More SQL questions

see sample-sql-queries-sqlite-chinook.ipynb

```
In [23]:    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 1, updating n results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery 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 "albums" (ArtistId)\n\nCREATE TABLE "alb Title NVARCHAR(160) NOT NUL AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n L.\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t \t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTvpeId INTEGER NOT NULL,\r\n GenreId INTEGER.\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT Bvtes INTEGER.\r\n UnitPrice NUMERIC(10.2) NOT NULL,\r\n NULL,\r\n FOREIGN KEY (AlbumId) REFERENC ES "albums" (Albumid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFEREN CES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) RE FERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX I FK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n ArtistId INTEGER PRIMARY KEY AU Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK TrackGenreId ON "tracks" (GenreId) TOINCREMENT NOT NULL,\r\n \n\nCREATE INDEX IFK PlaylistTrackTrackId ON "playlist track" (TrackId)\n\nCREATE TABLE "playlists"\r\n(\r PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "genres"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Name NVARCHAR(120)\r\n)\n\n CREATE INDEX IFK TrackMediaTypeId ON "tracks" (MediaTypeId)\n\n===Additional Context \n\nIn the SQLite da tabase invoice means order $\n$ ===Response Guidelines  $\n$ 1. 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 intermedi ate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Pleas e use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat th e answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n List all albums and the ir corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'conten List all albums and their corresponding artist names \n'\}, {'role': 'assistant', 'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId\n--------'}, {'role': 'user', 'content': ' \n There are 3 tables: artists, albums and track s, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n ind the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'content': 'SE LECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n There are 3 tables: artists, albums an d tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n n you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'conten t': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al. ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 1 0'}, {'role': 'user', 'content': ' \n Identify artists who have albums with tracks appearing in multip le genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artist s" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM "gen

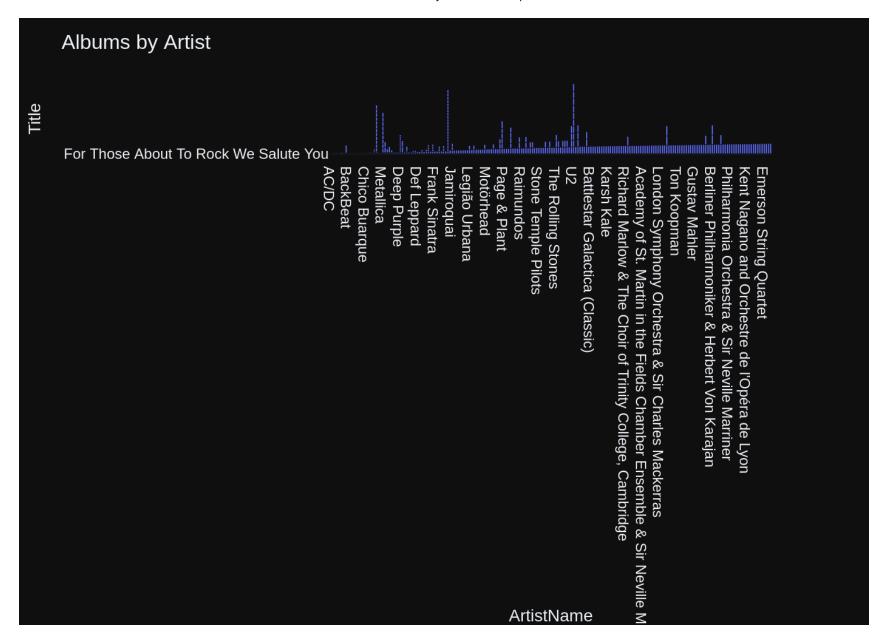
res" q2\n J0IN "tracks" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) >

1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'role': 'user', 'content': ' \n ists who have albums with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'S ELECT a.ArtistId, a.Name AS ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHE RE a.ArtistId IN (\n SELECT g2.GenreId\n FROM "genres" g2\n J0IN "tracks" t2 ON g2.GenreId = t2.GenreId \n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name'}, {'role': 'user', Hint: album quantity is found in invoice items, \n \n Find the top 5 customers wh 'content': ' \n o bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SE LECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY Tota lAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice item s, \n Find the top 5 customers who bought the most albums in total quantity (across all invoice s):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "inv oices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbum s DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer who bought the most albums in to tal quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.T rackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'ro le': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'c ontent': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice ite ms" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n List all albums and their corresponding artist names \n'}] Ollama parameters: model=aya:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery 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 \"albums\" (ArtistId)\n\nCREATE TABLE AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT \"albums\"\r\n(\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER P RIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n Medi aTypeId INTEGER NOT NULL.\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEG ER NOT NULL,\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreI d) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Med iaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n CREATE INDEX IFK TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"artists\"\r\n(\r\n GER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK TrackGenreId ON

\"tracks\" (GenreId)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON \"playlist track\" (TrackId)\n\nCREATE TAB PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LE \"playlists\"\r\n(\r\n 0)\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n ame NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided conte xt 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, p lease generate an intermediate SQL guery to find the distinct strings in that column. Prepend the guery wit h a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it ca n't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answe red before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.Artis tId"}, {"role": "user", "content": " \n List all albums and their corresponding artist names \n"}, {"r ole": "assistant", "content": "SELECT a.Title, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"artists\" a r ON a.ArtistId = ar.ArtistId\n------"}, {"role": "user", "content": " \n tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are Can you find the top 10 most popular artists based on the number of tracks\n"}, {"r linked by AlbumId.\n ole": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFR OM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.Album Id\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based on the number of trac ks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a. Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n Identify artists who hav e albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.Arti stId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.Art istId IN (\n SELECT q2.GenreId\n FROM \"genres\"  $q2\n JOIN \$  tracks\" t2 ON q2.GenreId = t2.GenreId\n G ROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName"}, {"role": "user", "content": " \n Identify artists who have albums with tracks appearing in multiple ge nres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM \"gen res\" q2\n JOIN \"tracks\" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreI d) > 1\n)\nGROUP BY a.ArtistId, a.Name"}, {"role": "user", "content": "\n Hint: album quantity is foun d in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (acro ss all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlb ums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers wh o bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SE LECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.In

```
voiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "cont
               Find the customer who bought the most albums in total quantity (across all invoices): \n"},
ent": " \n
{"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\"
c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.Invoic
eId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": "\n
t: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most album
s in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUN
T(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.Invo
iceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n
st all albums and their corresponding artist names \n"}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:42:58.573223775Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT a.Title, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.Ar
tistId'}, 'done reason': 'stop', 'done': True, 'total duration': 79649771391, 'load duration': 914325, 'pro
mpt eval count': 1794, 'prompt eval duration': 71656323000, 'eval count': 36, 'eval duration': 7294007000}
SELECT a.Title, ar.Name AS ArtistName
FROM "albums" a
JOIN "artists" ar ON a.ArtistId = ar.ArtistId
SELECT a.Title, ar.Name AS ArtistName
FROM "albums" a
JOIN "artists" ar ON a.ArtistId = ar.ArtistId
                                                Title \
0
                 For Those About To Rock We Salute You
1
                                     Balls to the Wall
2
                                     Restless and Wild
3
                                    Let There Be Rock
4
                                              Big Ones
. .
342
                                Respiahi: Pines of Rome
343
     Schubert: The Late String Quartets & String Qu...
                                  Monteverdi: L'Orfeo
344
345
                                Mozart: Chamber Music
346 Koyaanisqatsi (Soundtrack from the Motion Pict...
                                           ArtistName
0
                                                AC/DC
1
                                                Accept
2
                                               Accept
3
                                                AC/DC
4
                                             Aerosmith
342
                                        Eugene Ormandy
```

```
343
                                Emerson String Ouartet
    C. Monteverdi, Nigel Rogers - Chiaroscuro; Lon...
345
                                         Nash Ensemble
346
                                 Philip Glass Ensemble
[347 rows x 2 columns]
Ollama parameters:
model=aya:latest,
options={}.
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the guery
that answers the question the user asked: '\n List all albums and their corresponding artist names
\n'\nThe DataFrame was produced using this query: SELECT a.Title, ar.Name AS ArtistName\nFROM \"albums\"
a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId\n\nThe following is information about the resulting pan
das DataFrame 'df': \nRunning df.dtypes gives:\n Title
                                                               object\nArtistName
                                                                                     object\ndtvpe: objec
t"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataf
rame? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, us
e an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:43:20.206262559Z', 'message': {'role': 'assistant', 'c
ontent': "```python\nimport plotly.express as px\n\n# Create a bar chart\nfiq = px.bar(df, x='ArtistName',
y='Title')\n\n# Add title to the chart\nfig.update layout(title='Albums by Artist')\n\n# Show the chart\nfi
q.show()\n``"}, 'done reason': 'stop', 'done': True, 'total duration': 21607350787, 'load duration': 75087
5, 'prompt eval count': 184, 'prompt eval duration': 7294954000, 'eval count': 70, 'eval duration': 1417722
4000}
```



```
Out[23]: ('SELECT a.Title, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistI
          d',
                                                              Title \
           0
                             For Those About To Rock We Salute You
           1
                                                  Balls to the Wall
           2
                                                  Restless and Wild
           3
                                                  Let There Be Rock
           4
                                                           Big Ones
           . .
           342
                                            Respighi: Pines of Rome
           343
                Schubert: The Late String Quartets & String Qu...
                                               Monteverdi: L'Orfeo
           344
                                             Mozart: Chamber Music
           345
           346 Koyaanisgatsi (Soundtrack from the Motion Pict...
                                                         ArtistName
           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 \text{ rows } \times 2 \text{ columns}],
           Figure({
               'data': [{'alignmentgroup': 'True',
                          'hovertemplate': 'ArtistName=%{x}<br>Title=%{y}<extra></extra>',
                          'legendgroup': '',
                          'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                          'name': '',
                          'offsetgroup': '',
                          'orientation': 'v',
                          'showlegend': False,
                          'textposition': 'auto',
                          'type': 'bar',
                          'x': array(['AC/DC', 'Accept', 'Accept', ...,
                                      'C. Monteverdi, Nigel Rogers - Chiaroscuro; London Baroque; London Cornett & Sa
```

```
ckbu',
                                     'Nash Ensemble', 'Philip Glass Ensemble'], dtype=object),
                         'xaxis': 'x',
                         'y': array(['For Those About To Rock We Salute You', 'Balls to the Wall',
                                     'Restless and Wild', ..., "Monteverdi: L'Orfeo",
                                     'Mozart: Chamber Music',
                                     'Koyaanisqatsi (Soundtrack from the Motion Picture)'], dtype=object),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'title': {'text': 'Albums by Artist'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'ArtistName'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Title'}}}
          }))
         question = """
In [24]:
             Find all tracks with a name containing "What" (case-insensitive)
         0.00
         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 guery 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 "tracks" (GenreId)\n\nCREATE INDEX IFK Pl aylistTrackTrackId ON "playlist track" (TrackId)\n\nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMA RY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTvp eId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n Bytes INTEGER.\r\n UnitPrice NUMERIC(10.2) NOT NULL.\r\n FOREIGN KEY (AlbumId) REFE RENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REF ERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeI d) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IN DEX IFK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDEX IFK TrackMediaTypeId ON "tracks" (MediaTypeId)\n \nCREATE TABLE "playlist track"\r\n(\r\n PlaylistId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NUL CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) R EFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Tra ckid) REFERENCES "tracks" (Trackid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IF K InvoiceLineTrackId ON "invoice items" (TrackId)\n\nCREATE INDEX IFK AlbumArtistId ON "albums" (ArtistId) PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n \n\nCREATE TABLE "plavlists"\r\n(\r\n  $NVARCHAR(120)\r\n)\n\nCREATE TABLE "genres"\r\n(\r\n$ GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL Name  $NVARCHAR(120)\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order$ \n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL guery wi thout any explanations for the question. \n2. If the provided context is almost sufficient but requires kno wledge of a specific string in a particular column, please generate an intermediate SQL query to find the d istinct strings in that column. Prepend the query with a comment saying intermediate sql \n3. If the provid ed 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" (cas e-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT \* FROM tracks WHERE lower(Name) LIKE '%wha t%'"}, {'role': 'user', 'content': ' \n Find all tracks with a name containing "What" (case-insensitiv e)\n'}, {'role': 'assistant', 'content': 'SELECT \* \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%\''}, {'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 pt.PlaylistId, p.Name AS PlaylistName, SUM (t.Milliseconds) AS TotalDuration\nFROM "playlist track" pt\nJOIN "playlists" p ON pt.PlaylistId = p.Playli stId\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) > = 10'}, {'role': 'user', 'content': ' \n Get all playlists containing at least 10 tracks and the total duration of those tracks:\n'}, {'role': 'assistant', 'content': 'SELECT p.PlaylistId, SUM(t.Milliseconds) A S TotalDurationInMilliseconds\nFROM "playlists" p\nJOIN "playlist track" pt ON p.PlaylistId = pt.PlaylistId \nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROUP BY p.PlaylistId\nHAVING COUNT(pt.PlaylistId) >= 10\nORDE R BY TotalDurationInMilliseconds DESC'}, {'role': 'user', 'content': ' \n Identify artists who have al bums with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistI d, a.Name AS ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId I N (\n SELECT q2.GenreId\n FROM "genres" q2\n JOIN "tracks" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2. GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name'}, {'role': 'user', 'content': '

List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" q\nJOIN "tracks" t ON q.GenreId = t.GenreId\nGROUP B Y q.Name'}, {'role': 'user', 'content': ' \n List all genres and the number of tracks in each genr e:\n'}, {'role': 'assistant', 'content': 'SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" q\n JOIN "tracks" t ON q.GenreId = t.GenreId\nGROUP BY q.Name\nORDER BY TotalTracks DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assi stant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMI T 5'}, {'role': 'user', 'content': ' \n Find the top 5 most expensive tracks (based on unit price e):\n'}, {'role': 'assistant', 'content': 'SELECT Name, UnitPrice\nFROM "tracks"\nORDER BY UnitPrice DESC\n LIMIT 5'}, {'role': 'user', 'content': ' \n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT g2.GenreId\n FR OM "genres" q2\n JOIN "tracks" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.Gen reId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'role': 'user', 'content': '\n all tracks with a name containing "What" (case-insensitive)\n'}] Ollama parameters:

model=aya:latest,

options={},

keep alive=None

Prompt Content:

[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery 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 \"tracks\" (GenreId)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON \"playlist track\" (TrackId)\n\nCREATE TABLE \"tracks\"\r\n(\r\n PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n iaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220).\r\n Milliseconds INTE GER NOT NULL,\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreI d) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Med iaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n CREATE INDEX IFK TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDEX IFK TrackMediaTypeId ON \"tracks\" (M ediaTypeId)\n\nCREATE TABLE \"playlist track\"\r\n(\r\n PlavlistId INTEGER NOT NULL.\r\n TrackId INT CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n EGER NOT NULL.\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE INDEX IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE INDEX IFK AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n PlaylistId INTEGER PRIMARY KEY AUTOINCREMEN Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n T NOT NULL.\r\n GenreId INTEGER PRIMARY K EY AUTOINCREMENT NOT NULL,\r\n Name  $NVARCHAR(120)\r\n)\n\n===Additional Context \n\nIn the SQLite data$ base invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please gener ate a valid SQL query without any explanations for the question. \n2. If the provided context is almost suf ficient but requires knowledge of a specific string in a particular column, please generate an intermediate

SQL guery to find the distinct strings in that column. Prepend the guery with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please us e the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the an swer 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 \* FROM tracks WHERE lo wer(Name) LIKE '%what%'"}, {"role": "user", "content": " \n Find all tracks with a name containing \"Wh at\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT \* \nFROM \"tracks\" \nWHERE LOWER(Nam e) LIKE '%what%'"}, {"role": "user", "content": " \n Get all playlists containing at least 10 tracks a nd the total duration of those tracks:\n"}, {"role": "assistant", "content": "SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM \"playlist track\" pt\nJOIN \"playlists\" p ON pt.PlaylistId = p.PlaylistId\nJOIN \"tracks\" t ON pt.TrackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\n HAVING COUNT(pt.TrackId) >= 10"}, {"role": "user", "content": " \n Get all playlists containing at lea st 10 tracks and the total duration of those tracks:\n"}, {"role": "assistant", "content": "SELECT p.Playli stId, SUM(t.Milliseconds) AS TotalDurationInMilliseconds\nFROM \"playlists\" p\nJOIN \"playlist track\" pt ON p.PlaylistId = pt.PlaylistId\nJOIN \"tracks\" t ON pt.TrackId = t.TrackId\nGROUP BY p.PlaylistId\nHAVING COUNT(pt.PlaylistId) >= 10\nORDER BY TotalDurationInMilliseconds DESC"}, {"role": "user", "content": " \n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJ0IN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM \"genres\" q2\n JOIN \"tracks\" t2 ON q2.G enreId =  $t2.GenreId \setminus GROUP$  BY  $q2.GenreId \setminus HAVING$  COUNT( $q2.GenreId \setminus J \setminus GROUP$  BY a.ArtistId, a.Name"}, {"role": "user", "content": " \n List all genres and the number of tracks in each genre:\n"}, {"ro le": "assistant", "content": "SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres\" g\nJOIN \"tra cks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"role": "user", "content": " \n List all genres a nd the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreI d) AS TotalTracks\nFROM \"genres\" q\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the top 5 most expensive tracks (bas ed on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tra cks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the top 5 most exp ensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT Name, UnitPrice\nFROM \"tracks\"\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.A rtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a. ArtistId IN (\n SELECT q2.GenreId\n FROM \"genres\" q2\n JOIN \"tracks\" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName"}, {"role": "user", "content": " \n Find all tracks with a name containing \"What\" (case-insensitive) \n"}]

Add of existing embedding ID: 308edcd4-070f-52fe-b4c5-43177fb59bb1-sql Insert of existing embedding ID: 308edcd4-070f-52fe-b4c5-43177fb59bb1-sql

```
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:44:28.481142909Z', 'message': {'role': 'assistant', 'c
ontent': "SELECT * FROM tracks WHERE lower(Name) LIKE '%what%';"}, 'done reason': 'stop', 'done': True, 'to
tal duration': 68119427497, 'load duration': 743258, 'prompt eval count': 1578, 'prompt eval duration': 645
31291000, 'eval count': 15, 'eval duration': 2895087000}
SELECT * FROM tracks WHERE lower(Name) LIKE '%what%';
Output from LLM: SELECT * FROM tracks WHERE lower(Name) LIKE '%what%';
Extracted SQL: SELECT * FROM tracks WHERE lower(Name) LIKE '%what%'
SELECT * FROM tracks WHERE lower(Name) LIKE '%what%'
    TrackId
                                                      Name AlbumId \
                                                                   5
0
         26
                                             What It Takes
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
                                             What If I Do?
       1000
                                                                  80
7
       1039
                                                                  83
                                          What Now My Love
8
       1145
                                                                  89
                                               Whatsername
9
       1440
                        Whatever It Is, I Just Can't Stop
                                                                116
                                     Look What You've Done
10
       1469
                                                                119
       1470
                                                                119
11
                                         Get What You Need
12
       1628
                                                                133
                         What Is And What Should Never Be
                                                                146
13
       1778
             You're What's Happening (In The World Today)
14
       1823
                                                   So What
                                                                149
15
       2772
                      I Don't Know What To Do With Myself
                                                                223
16
       2884
                                                                231
                                             What Kate Did
       2893
                                                                230
17
                                 Whatever the Case May Be
               I Still Haven't Found What I'm Looking for
18
       2992
                                                                237
19
       3007
               I Still Haven't Found What I'm Looking For
                                                                238
20
       3258
                                                                255
                         Whatever Gets You Thru the Night
21
       3475
                                                                 322
                                      What Is It About Men
                                                                     Composer \
    MediaTypeId GenreId
0
                                      Steven Tyler, Joe Perry, Desmond Child
              1
                       1
              1
                       1
1
                                                    Audioslave/Chris Cornell
              1
                       2
2
                                                                  George Duke
3
              1
                       1
                                                     Jimmy Page/Robert Plant
              1
                       2
4
                                                                 Miles Davis
5
              1
                       1
                                       Mike Bordin, Billy Gould, Mike Patton
6
              1
                          Dave Grohl, Taylor Hawkins, Nate Mendel, Chris...
7
              1
                      12
                                   carl sigman/gilbert becaud/pierre leroyer
```

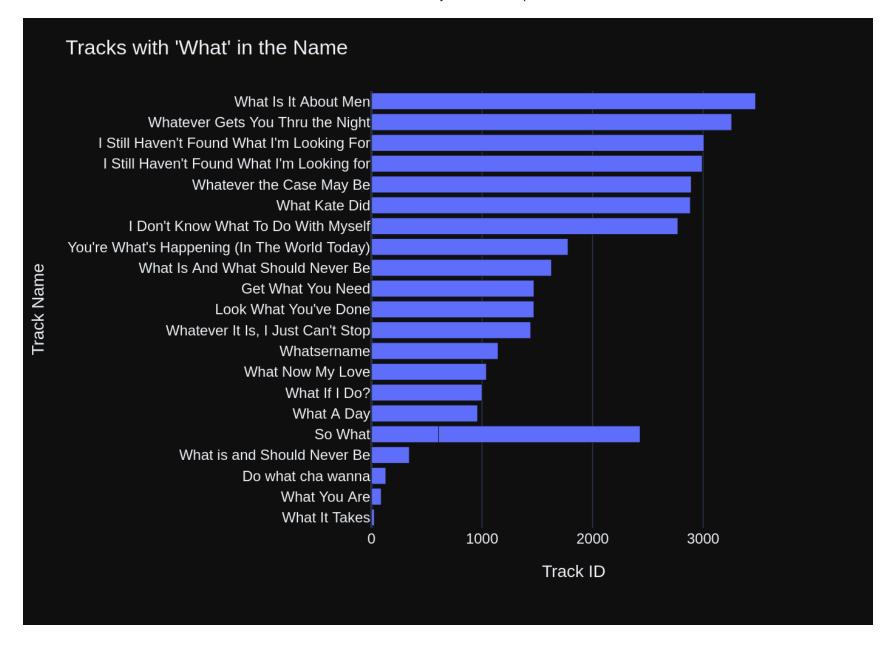
```
8
                                                                     Green Day
               1
                        4
9
               1
                        1
                                                              Jay Kay/Kay, Jay
10
               1
                        4
                                                                     N. Cester
               1
                        4
                                                C. Cester/C. Muncey/N. Cester
11
               1
                                                      Jimmy Page, Robert Plant
12
                        1
               1
                       14
                                        Allen Story/George Gordy/Robert Gordy
13
               1
                        3
14
                                                                  Culmer/Exalt
                        7
15
               1
                                                                           None
                       19
16
               3
                                                                           None
                       19
17
               3
                                                                           None
18
               1
                        1
                               Bono/Clayton, Adam/Mullen Jr., Larry/The Edge
19
               1
                        1
                                                                             U2
20
               2
                        9
                                                                           None
               2
                           Delroy "Chris" Cooper, Donovan Jackson, Earl C...
21
                       Bytes UnitPrice
    Milliseconds
0
                    10144730
                                    0.99
          310622
1
          249391
                     5988186
                                    0.99
2
                                    0.99
          274155
                     9018565
3
          260675
                                    0.99
                     8497116
          564009
                                    0.99
4
                    18360449
5
          158275
                     5203430
                                    0.99
6
          302994
                     9929799
                                    0.99
7
                                    0.99
          149995
                     4913383
8
          252316
                     8244843
                                    0.99
          247222
                                    0.99
9
                     8249453
10
                                    0.99
          230974
                     7517083
          247719
                                    0.99
11
                     8043765
          287973
                     9369385
                                    0.99
12
          142027
                     4631104
                                    0.99
13
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=aya: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-in sensitive)\n'\n\nThe DataFrame was produced using this query: SELECT \* FROM tracks WHERE lower(Name) LIKE '%what%'\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes giv es:\n TrackId int64\nName object\nAlbumId int64\nMediaTvpeId int64\nG enreId int64\nComposer obiect\nMilliseconds int64\nBvtes int64\nUnitPric float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to c hart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only o ne value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanat ions -- just the code."}]

### Ollama Response:

{'model': 'aya:latest', 'created\_at': '2024-06-14T11:44:55.790274122Z', 'message': {'role': 'assistant', 'c ontent': '```python\nimport plotly.express as px\n\n# Create a bar chart\nfig = px.bar(df, x="TrackId", y ="Name")\n\n# Update layout\nfig.update\_layout(\n title="Tracks with \'What\' in the Name",\n xaxis\_t itle="Track ID",\n yaxis\_title="Track Name",\n)\n\n# Render the figure\nfig.show()\n```'}, 'done\_reaso n': 'stop', 'done': True, 'total\_duration': 27282563735, 'load\_duration': 710633, 'prompt\_eval\_count': 218, 'prompt eval duration': 8497699000, 'eval count': 93, 'eval duration': 18654027000}



Out[24]:	("SE	LECT * FROM	tracks WHE	RE 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	W	hatever 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		ou're What'	s Happening (In The World Today)	146	
	14	1823		So What	149	
	15	2772	I D	on't Know What To Do With Myself	223	
	16	2884		What Kate Did	231	
	17	2893		Whatever the Case May Be	230	
	18	2992		ven't Found What I'm Looking for	237	
	19	3007		ven'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	
		M 1' T T				
	0	MediaTypeI		Characa Tallana Jan Bar		nposer \
	0		1 1	Steven Tyler, Joe Per	-	
	1		1 1	Audios	.ave/Chris Co	
	2		1 2	7.5	George	
	3		1 1	Jimmy	Page/Robert	
	4		1 2	Mika Dandin Dilly (	Miles	
	5		1 1	Mike Bordin, Billy (		
	6 7		1 1	Dave Grohl, Taylor Hawkins, Nate		
			1 12	carl sigman/gilbert beca	•	-
	8		1 4			en Day
	9 10		$egin{array}{cccc} 1 & & 1 \ 1 & & 4 \ \end{array}$		Jay Kay/Kay	, Jay Cester
	11		1 4 1 4	C. Cester/C.		
	12		1 1		Page, Robert	
	13		1 14	Allen Story/George G		
	14		1 3	Acten Story/deorge of	Culmer/	-
	15		1 7		cutille! /	None
	13		· /			MOLIC

```
3
                      19
16
                                                                         None
              3
17
                       19
                                                                          None
18
              1
                       1
                               Bono/Clayton, Adam/Mullen Jr., Larry/The Edge
              1
                       1
19
                                                                           U2
              2
20
                       9
                                                                          None
              2
                          Delroy "Chris" Cooper, Donovan Jackson, Earl C...
21
                      Bytes UnitPrice
    Milliseconds
          310622
0
                   10144730
                                   0.99
                    5988186
                                   0.99
1
          249391
2
                    9018565
                                   0.99
          274155
3
          260675
                    8497116
                                   0.99
4
          564009
                   18360449
                                   0.99
5
                                   0.99
          158275
                    5203430
6
                    9929799
                                   0.99
          302994
7
                                   0.99
          149995
                    4913383
8
                                   0.99
          252316
                    8244843
9
          247222
                    8249453
                                   0.99
10
          230974
                    7517083
                                   0.99
                                   0.99
11
          247719
                    8043765
12
          287973
                                   0.99
                    9369385
          142027
                                   0.99
13
                    4631104
          189152
                                   0.99
14
                    6162894
                    7251478
                                   0.99
15
          221387
16
         2610250
                  484583988
                                   1.99
17
         2616410
                  183867185
                                   1.99
18
          353567
                   11542247
                                   0.99
          280764
19
                    9306737
                                   0.99
20
          215084
                                   0.99
                    3499018
          209573
21
                    3426106
                                   0.99
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'TrackId=%{x}<br>Name=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'h',
               'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
              'x': array([ 26, 88, 130, 342, 607, 960, 1000, 1039, 1145, 1440, 1469, 1470,
```

```
1628, 1778, 1823, 2772, 2884, 2893, 2992, 3007, 3258, 3475]),
                         'xaxis': 'x'.
                         'y': 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),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'title': {'text': "Tracks with 'What' in the Name"},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Track ID'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Track Name'}}}
          }))
         question = """
In [25]:
             Get the total number of invoices for each customer
         0.00
         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 guery 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 "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR EMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n BillingA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n **FOREIG** rv NVARCHAR(40),\r\n N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON "invoice items" (InvoiceId)\n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KE InvoiceId INTEGER NOT NULL,\r\n Y AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n Price NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERE NCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLin eTrackId ON "invoice items" (TrackId)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KE Y AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NUL  $L.\r\n$ Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHA  $R(40), \r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n  $VARCHAR(24).\r\n$ FOREIGN KEY (SupportR epId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IN DEX IFK CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "employees"\r\n(\r\n Id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVA ReportsTo INTEGER,\r\n RCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n BirthDate DATETIME.\r\n State NVARCHAR(40), \r\n HireDate DATETIME,\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24).\r FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DEL Email NVARCHAR(60),\r\n ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON "employees" (ReportsTo)\n\n CREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n (200) NOT NULL.\r\n AlbumId INTEGER.\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL,\r\n Bvtes INTEGER.\r\n UnitPrice NUMER FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION IC(10,2) NOT NULL,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n ON UPDATE NO ACTION.\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DEL ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL que ry without any explanations for the question. \n2. If the provided context is almost sufficient but require s knowledge of a specific string in a particular column, please generate an intermediate SQL guery to find the distinct strings in that column. Prepend the guery with a comment saying intermediate sql \n3. If the p rovided 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 Get the total number of invoices for each cu stomer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFRO M "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'use

r', 'content': ' \n Find the total number of invoices per country:\n'}, {'role': 'assistant', 'conten

t': 'SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM "customers" customer\nJOIN "in voices" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country'}, {'role': 'user', 'conten Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.Cus tomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Cus tomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': '\n Find the customer with the most invoices \n'\}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(I nvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-----'}, {'role': 'user', 'content': ' \n Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(i.InvoiceId) AS TotalInvoices \nFROM "invoices" i\nGROUP BY BillingCountry'}, {'role': 'user', 'content': ' \n Find the customer wit h the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(InvoiceId) as NumOfInv oices\nFROM "invoices"\nGROUP BY CustomerId\nORDER BY NumOfInvoices DESC\nLIMIT 1'}, {'role': 'user', 'cont Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELE CT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY AverageInvoiceTotal DESC\nLIMIT 10'}, {'role': 'user', 'con Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SEL ECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJ0IN "invoices" i ON c.CustomerI d = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n Find the total number of in voices per country:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(\*) AS TotalInvoic es\nFROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': ' \n Get the average in voice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Custo merId\nORDER BY AverageInvoiceTotal DESC'}, {'role': 'user', 'content': ' \n Get the total number of in voices for each customer\n'}l Ollama parameters: model=aya: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 \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL.\r\n InvoiceDate DATETIME NOT NULL.\r\n Billin gAddress NVARCHAR(70).\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillinaCou ntrv NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n F0RE IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK InvoiceLineInv oiceId ON \"invoice items\" (InvoiceId)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL,\r UnitPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Tr

ackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE \"customers\"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCH FirstName NVARCHAR(40) NOT NULL,\r\n Company NVARCHAR(80),\r\n AR(20) NOT NULL,\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(2 4),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n FOREI GN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"employees EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NUL L.\r\n FirstName NVARCHAR(20) NOT NULL.\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER.\r\n thDate DATETIME.\r\n HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n St Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r ate NVARCHAR(40),\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON \"employees\" (ReportsTo)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NUL Milliseconds INTEGER NOT NULL.\r\n L.\r\n GenreId INTEGER.\r\n Composer NVARCHAR(220),\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (A es INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n lbumid) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"me dia types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is su fficient, 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 gen erate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comme nt saying intermediate sql \n3. If the provided context is insufficient, please explain why it can't be gen erated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered befor e, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Customer Id\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n Find the total number of invoices per cou ntry:\n"}, {"role": "assistant", "content": "SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCo unt\nFROM \"customers\" customer\nJOIN \"invoices\" invoice ON customer.CustomerId = invoice.CustomerId\nGR OUP BY Country", {"role": "user", "content": "\n Find the customer with the most invoices \n"}, {"ro le": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\n LIMIT 1"}, {"role": "user", "content": " \n Find the customer with the most invoices \n"}, {"role": "a ssistant", "content": "SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY C ustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n------"}, {"role": "user". "conten Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT B illingCountry, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY BillingCountry"}, {"rol e": "user", "content": " \n Find the customer with the most invoices \n"}, {"role": "assistant", "cont

ent": "SELECT CustomerId, COUNT(InvoiceId) as NumOfInvoices\nFROM \"invoices\"\nGROUP BY CustomerId\nORDER
BY NumOfInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n Get the average invoice total for ea
ch customer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTota
l\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER B
Y AverageInvoiceTotal DESC\nLIMIT 10"}, {"role": "user", "content": " \n Get the average invoice total
for each customer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoi
ceTotal\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"},
{"role": "user", "content": " \n Find the total number of invoices per country:\n"}, {"role": "assistan
t", "content": "SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.Billing
Country"}, {"role": "user", "content": " \n Get the average invoice total for each customer:\n"}, {"rol
e": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c
\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY AverageInvoiceTotal D
ESC"}, {"role": "user", "content": " \n Get the total number of invoices for each customer\n"}]

Add of existing embedding ID: 49e67df3-a604-51f8-ad01-b8f5a2043eac-sql Insert of existing embedding ID: 49e67df3-a604-51f8-ad01-b8f5a2043eac-sql

```
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:46:17.453305849Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i 0
N c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, 'done reason': 'stop', 'done': True, 'total duratio
n': 81545834261, 'load duration': 720171, 'prompt eval count': 1775, 'prompt eval duration': 71066412000,
'eval count': 48, 'eval duration': 9788172000}
SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
    CustomerId TotalInvoices
0
             1
                            7
             2
1
2
             3
                            7
3
                            7
             4
             5
                            7
4
5
                            7
             6
             7
                            7
6
7
             8
                            7
                            7
8
             9
9
            10
                            7
10
            11
                            7
11
            12
12
            13
13
                            7
            14
                            7
14
            15
15
                            7
            16
16
            17
17
            18
                            7
18
            19
19
            20
                            7
20
            21
                            7
21
            22
22
            23
                            7
23
                            7
            24
                            7
24
            25
25
                            7
            26
26
            27
```

27	28	7				
28	29	7				
29	30					
30	31	7 7 7 7 7				
31	32	7				
32	33	7				
33	34	7				
34	35	7				
35	36	7				
36	37	7				
37	38	7				
38	39	7				
39	40	7				
40	41	7				
41	42	7				
42	43	7				
43	44	7 7 7 7 7 7 7 7 7				
44	45	7				
45	46	7				
46	47	7				
47	48	7				
48	49	7				
49	50	7				
50	51	7				
51	52	7				
52	53	7 7 7 7 7 7				
53	54	7				
54	55	7				
55	56	7				
56	57	7 7 6				
57	58	7				
58	59	6				
Ollama	parameters:					
model=aya:latest,						
ontions={}						

options={},

keep alive=None

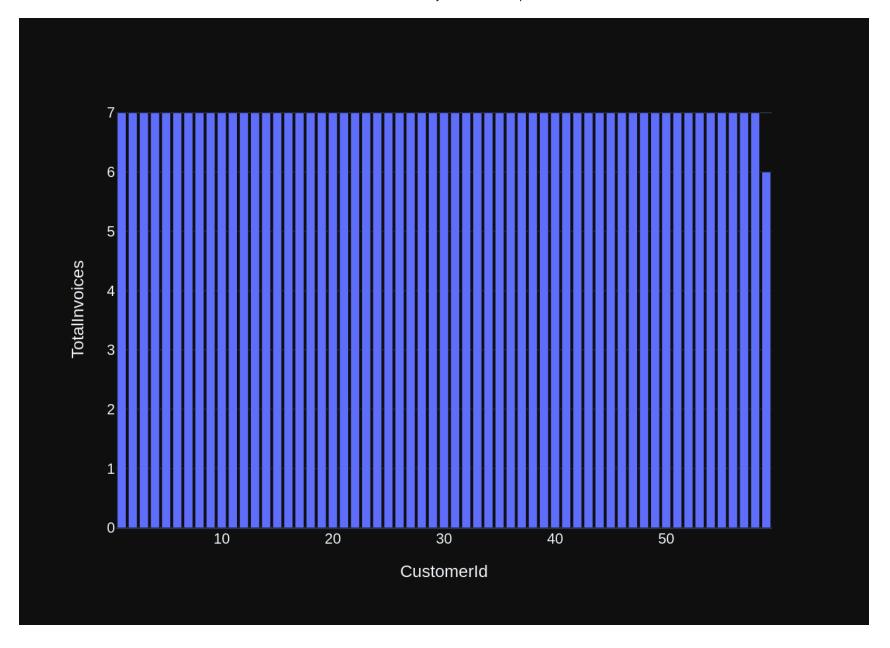
Prompt Content:

[{"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 total number of invoices for each customer\n'\n \nThe DataFrame was produced using this query: SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFR OM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\n\nThe follow ing is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId

int64\nTotalInvoices int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python pl otly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If t here is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer wit h any explanations -- just the code."}]

Ollama Response:

{'model': 'aya:latest', 'created\_at': '2024-06-14T11:46:37.105427371Z', 'message': {'role': 'assistant', 'c ontent': "```python\nimport plotly.express as  $px\n\m$  Create a bar chart with the data\nfig =  $px.bar(df, x = CustomerId', y='TotalInvoices')\n\m$  Display the chart\nfig.show()\n```"}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 19624266060, 'load\_duration': 751486, 'prompt\_eval\_count': 205, 'prompt\_eval\_duration': 8433048000, 'eval\_count': 54, 'eval\_duration': 11053824000}



Out[25]: ('SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.Cus tomerId = i.CustomerId\nGROUP BY c.CustomerId',

tome		comeria/ugkoup B
	CustomerId	TotalInvoices
0	1	7
1	2	7
2	3	7
3	4	7
4	5	7
5	6	7
6	7	7
7	8	7
8	9	7
9	10	7
10	11	7
11	12	7
12	13	7
13	14	7
14	15	7
15	16	7
16	17	7
17	18	7
18	19	7
19	20	7
20	21	7
21	22	7
22	23	7
23	24	7
24	25	7
25	26	7
26	27	7
27	28	7
28	29	7
29	30	7
30	31	7
31	32	7
32	33	7
33	34	7
34	35	7
35	36	7
36	37	7
37	38	7
38	39	7

```
7
39
          40
                        7
40
          41
                        7
41
          42
                        7
42
          43
                        7
43
          44
                        7
          45
44
                        7
45
          46
                        7
46
          47
                        7
47
          48
48
          49
                        7
                        7
          50
49
                        7
50
          51
          52
                        7
51
52
          53
                        7
                        7
53
          54
                        7
54
          55
55
          56
                        7
          57
                        7
56
                        7
57
          58
                        6,
58
          59
Figure({
   'data': [{'alignmentgroup': 'True',
            'hovertemplate': 'CustomerId=%{x}<br>TotalInvoices=%{y}<extra></extra>',
            'legendgroup': '',
            'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
            'name': '',
            'offsetgroup': '',
            'orientation': 'v',
            'showlegend': False,
            'textposition': 'auto',
            'type': 'bar',
            'x': array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
                      19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
                      37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
                      55, 56, 57, 58, 59]),
            'xaxis': 'x',
            7, 7, 7, 7, 7, 7, 7, 7, 7, 6]),
            'yaxis': 'y'}],
   'layout': {'barmode': 'relative',
             'legend': {'tracegroupgap': 0},
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery 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 "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR CustomerId INTEGER NOT NULL,\r\n EMENT NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n BillinaA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillinaCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n rv NVARCHAR(40),\r\n **FOREIG** N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n UnitPrice NUMERIC(10.2) NOT NULL.\r FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\t Quantity INTEGER NOT NULL,\r\n ON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t \t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (Custome rId)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON "invoice items" (InvoiceId)\n\nCREATE INDEX IFK InvoiceLin eTrackId ON "invoice items" (TrackId)\n\nCREATE TABLE "employees"\r\n(\r\n EmployeeId INTEGER PRIMARY KE Y AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NUL L.\r\n Title NVARCHAR(30).\r\n ReportsTo INTEGER.\r\n BirthDate DATETIME.\r\n HireDate DATETIM E, r nAddress NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHA  $R(40), \r\n$ PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n  $RCHAR(60).\r\n$ FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION O N UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMEN T NOT NULL.\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n  $NVARCHAR(80).\r\n$ Address NVARCHAR(70),\r\n Coun Fax NVARCHAR(24),\r\n trv NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(24),\r\n SupportRepId INTEGER,\r\n FOREIGN KEY (SupportRepId) REFERENCES "em Email NVARCHAR(60) NOT NULL,\r\n ployees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r Title NVARCHAR(160) NOT NULL,\r\n Albumid INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n stId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\CREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREM ENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "med ia types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeRe portsTo ON "employees" (ReportsTo)\n\n\===Additional Context \n\nIn the SQLite database invoice means orde r\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query w ithout any explanations for the question. \n2. If the provided context is almost sufficient but requires kn owledge 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 t able(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it wa s given before. \n'}, {'role': 'user', 'content': ' \n Find the total number of invoices per countr

y:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "invoices" i\nGROUP BY BillingCountry'}, {'role': 'user', 'content': ' \n Find the total number of invo ices per country:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices \nFROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': ' \n Find the total numbe r of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT Customer.Country, COUNT(invoice.In voiceId) AS InvoiceCount\nFROM "customers" customer\nJOIN "invoices" invoice ON customer.CustomerId = invoi ce.CustomerId\nGROUP BY Country'}, {'role': 'user', 'content': ' \n Get the total number of invoices fo r each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvo ices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'rol e': 'user', 'content': ' \n Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.Invoic eDate > \'2010-01-01\'\nGROUP BY i.InvoiceId'}, {'role': 'user', 'content': ' \n Find all invoices sinc e 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM(i.To tal) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'role': 'user', 'content': ' \n Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\n ORDER BY NumberOfInvoices DESC\nLIMIT 1\n------'}, {'role': 'user', 'content': '\n List all invoices with a total exceeding \$10:\n'}, {'role': 'assistant', 'content': 'SELECT \*\nFROM "invoic es" \nWHERE (Total > 10)\nGROUP BY InvoiceId'}, {'role': 'user', 'content': ' \n Find the customer wit h the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS Tota lInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORD ER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceId, SUM(i.Total) AS Invoi ceAmount\nFROM "invoices" i\nWHERE strftime(\'%Y\', i.InvoiceDate) >= \'2010\'\nGROUP BY i.InvoiceId\nORDER BY InvoiceAmount DESC'}, {'role': 'user', 'content': ' \n Find the total number of invoices per countr y:\n'}] Ollama parameters: model=aya: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 \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n Billin gAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40).\r\n BillinaCou ntrv NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n F0RE IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL L,\r\n InvoiceId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL.\r\n UnitPrice NUMERIC(10.2) NO Ouantity INTEGER NOT NULL.\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceI T NULL,\r\n d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (Tra

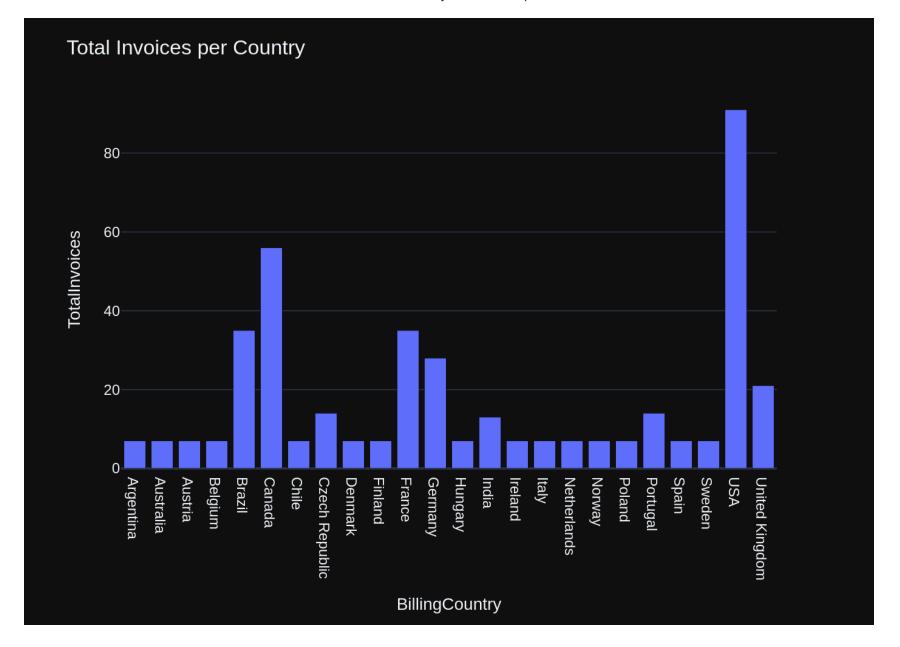
ckid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invoi ces\" (CustomerId)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON \"invoice items\" (InvoiceId)\n\nCREATE INDE X IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE \"employees\"\r\n(\r\n EmployeeI d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVAR CHAR(20) NOT NULL.\r\n Title NVARCHAR(30).\r\n ReportsTo INTEGER,\r\n BirthDate DATETIME.\r\n HireDate DATETIME,\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Fax NVARCHAR(24),\r PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON D Email NVARCHAR(60).\r\n ELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n CustomerId INTEGER PRIMA RY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT N ULL,\r\n City NVARCHAR(40),\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n State NVARC Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n  $HAR(40).\r\n$ Phone NVARCHAR(24),\r\n  $NVARCHAR(24).\r\n$ Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER.\r\n FOREIGN KEY (Support RepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n Albumid INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL.\r\n ArtistId INTEGER NOT NULL.\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistI d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTE GER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL.\r\n AlbumId INTEGER.\r\n Milliseconds I MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n NTEGER NOT NULL,\r\n Bvtes INTEGER.\r\n UnitPrice NUMERIC(10.2) NOT NULL,\r\n FOREIGN KEY (Album Id) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Ge nreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)  $\n\n\$  IFK EmployeeReportsTo ON \"employees\" (ReportsTo)\n\n\===Additional Context \n\nIn the S QLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, pl ease 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 in termediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying i ntermediate sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n 4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please r epeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT BillingCountry, COUNT(i.Invoi ceId) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY BillingCountry"}, {"role": "user", "content": " \n Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT i.BillingCount ry, COUNT(\*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "content Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT C ustomer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM \"customers\" customer\nJOIN \"invoices\" i nvoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country"}, {"role": "user", "content": " \n Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.Customer Id, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Cust omerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceId, SUM(i.Total) AS TotalAmou

```
nt\nFROM \"invoices\" i\nWHERE i.InvoiceDate > '2010-01-01'\nGROUP BY i.InvoiceId"}, {"role": "user", "cont
              Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "cont
ent": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate >= '2010
-01-01'\nGROUP BY i.InvoiceDate"}, {"role": "user", "content": " \n
                                                                        Find the customer with the most in
voices \n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFR
OM Invoices \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n------"},
{"role": "user", "content": " \n List all invoices with a total exceeding $10:\n"}, {"role": "assistan
t", "content": "SELECT *\nFROM \"invoices\" \nWHERE (Total > 10)\nGROUP BY InvoiceId"}, {"role": "user", "c
                  Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT
c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerI
d = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "conten
            Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "conten
t": "SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount\nFROM \"invoices\" i\nWHERE strftime('%Y', i.Invoice
Date) >= '2010'\nGROUP BY i.InvoiceId\nORDER BY InvoiceAmount DESC"}, {"role": "user", "content": " \n
Find the total number of invoices per country:\n"}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:47:58.789365595Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT BillingCountry, COUNT(InvoiceId) AS TotalInvoices\nFROM Invoices\nGROUP BY BillingCountr
y'}, 'done reason': 'stop', 'done': True, 'total duration': 81588051375, 'load duration': 761760, 'prompt e
val count': 1841, 'prompt eval duration': 76227879000, 'eval count': 23, 'eval duration': 4650469000}
SELECT BillingCountry, COUNT(InvoiceId) AS TotalInvoices
FROM Invoices
GROUP BY BillingCountry
SELECT BillingCountry, COUNT(InvoiceId) AS TotalInvoices
FROM Invoices
GROUP BY BillingCountry
   BillingCountry TotalInvoices
                               7
0
         Argentina
                               7
1
        Australia
                               7
2
           Austria
                               7
3
           Belgium
4
            Brazil
                              35
5
           Canada
                              56
6
            Chile
                               7
7
    Czech Republic
                              14
8
           Denmark
                               7
                               7
9
           Finland
                              35
10
           France
                              28
11
           Germany
12
           Hungary
                               7
13
             India
                              13
14
           Ireland
                               7
```

```
7
15
             Italy
                                 7
16
       Netherlands
17
                                 7
            Norway
                                 7
            Poland
18
19
          Portugal
                                14
20
                                 7
             Spain
                                 7
21
            Sweden
22
               USA
                                91
                                21
23 United Kinadom
Ollama parameters:
model=aya: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 total number of invoices per country:\n'\n\nThe DataFrame was produced using this query: SELECT BillingCountry, COUNT(InvoiceId) AS TotalInvoices\nFROM Invoices\nGROUP BY BillingCountry\n\nThe following is information about the resulting pandas DataFrame 'df':\nRunning df.dtypes gives:\n BillingCountry object\nTotalInvoices int64\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 Indicato r. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:

{'model': 'aya:latest', 'created\_at': '2024-06-14T11:48:21.564785636Z', 'message': {'role': 'assistant', 'c ontent': "```python\nimport plotly.express as px\n\n# Create a bar chart\nfig = px.bar(df, x='BillingCountr y', y='TotalInvoices')\n\n# Add title to the chart\nfig.update\_layout(title='Total Invoices per Country')\n\n# Show the chart\nfig.show()\n```"}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 22749722772, 'load\_duration': 682980, 'prompt\_eval\_count': 175, 'prompt\_eval\_duration': 7253147000, 'eval\_count': 74, 'e val\_duration': 15365016000}



```
Out[26]: ('SELECT BillingCountry, COUNT(InvoiceId) AS TotalInvoices\nFROM Invoices\nGROUP BY BillingCountry',
               BillingCountry TotalInvoices
           0
                    Argentina
           1
                                           7
                    Australia
           2
                                           7
                      Austria
           3
                                           7
                      Belgium
           4
                       Brazil
                                           35
           5
                       Canada
                                           56
           6
                        Chile
                                           7
           7
                                           14
               Czech Republic
                                           7
           8
                      Denmark
           9
                                           7
                      Finland
                                           35
           10
                       France
                                           28
           11
                      Germany
           12
                                           7
                      Hungary
           13
                        India
                                           13
           14
                      Ireland
                                           7
           15
                                           7
                        Italy
                                           7
           16
                  Netherlands
           17
                                           7
                       Norway
                                           7
           18
                       Poland
           19
                     Portugal
                                           14
                                           7
           20
                        Spain
                                           7
           21
                       Sweden
           22
                          USA
                                           91
           23 United Kingdom
                                          21,
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'BillingCountry=%{x}<br>TotalInvoices=%{y}<extra></extra>',
                         'legendgroup': '',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                         'name': '',
                         'offsetgroup': '',
                         'orientation': 'v',
                         'showlegend': False,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array(['Argentina', 'Australia', 'Austria', 'Belgium', 'Brazil', 'Canada',
                                      'Chile', 'Czech Republic', 'Denmark', 'Finland', 'France', 'Germany',
                                     'Hungary', 'India', 'Ireland', 'Italy', 'Netherlands', 'Norway',
                                     'Poland', 'Portugal', 'Spain', 'Sweden', 'USA', 'United Kingdom'],
                                    dtype=object),
```

```
'xaxis': 'x',
                         'y': array([ 7,  7,  7,  7,  35,  56,  7,  14,  7,  7,  35,  28,  7,  13,  7,  7,  7,
                                      7, 14, 7, 7, 91, 21]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'title': {'text': 'Total Invoices per Country'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'BillingCountry'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalInvoices'}}}
          }))
         question = """
In [27]:
             List all invoices with a total exceeding $10:
         0.00
         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 guery 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 items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER NOT NULL.\r\n InvoiceId INTEGER NOT NULL,\r\n ice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERE NCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLin eInvoiceId ON "invoice items" (InvoiceId)\n\nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n BillingState NVARCHAR(40),\r\n BillingAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10).\r\n ingCountry NVARCHAR(40),\r\n Total NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO N\r\n)\n\nCREATE INDEX IFK InvoiceLineTrackId ON "invoice items" (TrackId)\n\nCREATE INDEX IFK InvoiceCusto merId ON "invoices" (CustomerId)\n\nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCRE MENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" Bvtes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "med ia types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeRe portsTo ON "employees" (ReportsTo)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KEY A FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n UTOINCREMENT NOT NULL,\r\n Address NVARCHAR(70),\r\n Company NVARCHAR(80),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Fax NVARCHAR(24),\r PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n FOREIGN KEY (SupportRepId) REFERENC ES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "employee  $s"\r\n(\r\n$ EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NUL ReportsTo INTEGER,\r\n L.\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n Bir thDate DATETIME,\r\n HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n St ate NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employees" (E mploveeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON "customers" (SupportRepId) $\n\n===Additional$  Context  $\n\n$  the SQLite database invoice means order $\n\n===$ Response Guidelines \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 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 invoices with a total exceeding \$10:\n'}, {'rol e': 'assistant', 'content': 'SELECT \* \nFROM "invoices" \nWHERE Total > 10.00'}, {'role': 'user', 'conten List all invoices with a total exceeding \$10:\n'}, {'role': 'assistant', 'content': 'SELECT \*

FROM "invoices" WHERE Total > 10'}, {'role': 'user', 'content': ' \n List all invoices with a total exc eeding \$10:\n'}, {'role': 'assistant', 'content': 'SELECT \*\nFROM "invoices" \nWHERE (Total > 10)\nGROUP BY InvoiceId'}, {'role': 'user', 'content': ' \n Find all invoices since 2010 and the total amount invoice d:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate > \'2010-01-01\'\nGROUP BY i.InvoiceId'}, {'role': 'user', 'content': ' \n all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.Invo iceDate, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i. InvoiceDate'}, {'role': 'user', 'content': ' \n Find all invoices since 2010 and the total amount invoi ced:\n'\}, {'role': 'assistant', 'content': 'SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount\nFROM "invoic es" i\nWHERE strftime(\'%Y\', i.InvoiceDate) >= \'2010\'\nGROUP BY i.InvoiceId\nORDER BY InvoiceAmount DES C'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money overall, \n Hint: order total can be found on invoices table, calculation using invoice items detail table is un necessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "cu stomers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DE SC\nLIMIT 5\n-----'}, {'role': 'user', 'content': ' \n Find the top 5 customers wh o spent the most money overall, \n \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerI d, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n customers who spent the most money overall, \n \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Cust omerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most al bums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, C OUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\n GROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n-----'}, {'role': 'user', 'c ontent': '\n List all invoices with a total exceeding \$10:\n'}] Ollama parameters: model=aya:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery 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 items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY K EY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n tPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCE S \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) RE FERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK Invo iceLineInvoiceId ON \"invoice items\" (InvoiceId)\n\nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGE R PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT

BillingAddress NVARCHAR(70).\r\n BillingCity NVARCHAR(40),\r\n NULL,\r\n BillingState NVARCHAR(4 0), r nBillingCountry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10).\r\n Total NUMERIC(10.2) NOT NULL,\r\n FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE IN DEX IFK InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEG ER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds I NTEGER NOT NULL,\r\n Bvtes INTEGER,\r\n UnitPrice NUMERIC(10.2) NOT NULL.\r\n FOREIGN KEY (Album Id) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (Ge nreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE INDEX IFK EmployeeReportsTo ON \"employees\" (ReportsTo)\n\nCREATE TABLE \"customers\"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL.\r\n Name NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR  $(40), \r\n$ State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone  $NVARCHAR(24).\r\n$ Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL.\r\n SupportRepId INTEGER.\r FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDAT E NO ACTION\r\n)\n\nCREATE TABLE \"employees\"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHA  $R(30), \r\n$ ReportsTo INTEGER,\r\n BirthDate DATETIME.\r\n HireDate DATETIME.\r\n Address NVARCH City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode  $AR(70).\r\n$  $NVARCHAR(10), \r\n$ Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60).\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficien t, please generate a valid SQL guery without any explanations for the question. \n2. If the provided contex t 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 say ing intermediate sql \n3. If the provided context is insufficient, please explain why it can't be generate d. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, ple ase repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n nvoices with a total exceeding \$10:\n"}, {"role": "assistant", "content": "SELECT \* \nFROM \"invoices\" \nW HERE Total > 10.00"}, {"role": "user", "content": " \n List all invoices with a total exceeding \$1 0:\n"}, {"role": "assistant", "content": "SELECT \* FROM \"invoices\" WHERE Total > 10"}, {"role": "user", "content": " \n List all invoices with a total exceeding \$10:\n"}, {"role": "assistant", "content": "SE LECT \*\nFROM \"invoices\" \nWHERE (Total > 10)\nGROUP BY InvoiceId"}, {"role": "user", "content": " \n Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate > '2010-01-01'\nGROUP BY i.InvoiceId"}, {"role": "user", "content": " \n Find all invoices since 2010 and the total amount invoi ced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"invoi ces\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"role": "user", "content": " \n Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT

i.InvoiceId, SUM(i.Total) AS InvoiceAmount\nFROM \"invoices\" i\nWHERE strftime('%Y', i.InvoiceDate) >= '20 10'\nGROUP BY i.InvoiceId\nORDER BY InvoiceAmount DESC"}, {"role": "user", "content": " \n \n Hint: order total can be found on invoices ta p 5 customers who spent the most money overall, \n ble, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SE LECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n-------"}, {"rol e": "user", "content": " \n Find the top 5 customers who spent the most money overall, \n int: order total can be found on invoices table, calculation using invoice items detail table is unnecessar y \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers \" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\n LIMIT 5"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most money overall, Hint: order total can be found on invoices table, calculation using invoice items detail tabl \n e is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpending \nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5"}, {"role": "user", "content": " \n Hint: album quantity is found in invoic Find the top 5 customers who bought the most albums in total quantity (across all invo e items. \n \n ices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY To talAlbums DESC\nLIMIT 5\n-----"}, {"role": "user", "content": " \n List all invoice s with a total exceeding \$10:\n"}]

Add of existing embedding ID: 6f4242b8-c8d7-5d80-877e-00b6407ae9ad-sql Insert of existing embedding ID: 6f4242b8-c8d7-5d80-877e-00b6407ae9ad-sql

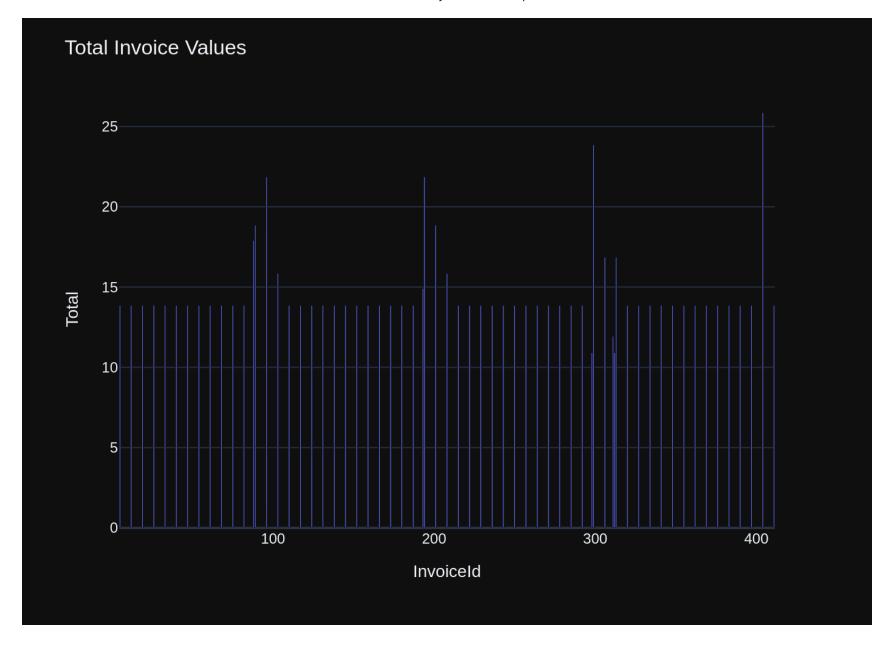
```
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:49:45.901654854Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT * FROM "invoices" WHERE Total > 10'}, 'done reason': 'stop', 'done': True, 'total duratio
n': 84240199001, 'load_duration': 748869, 'prompt_eval_count': 1933, 'prompt eval duration': 80805917000,
'eval count': 14, 'eval duration': 2736744000}
SELECT * FROM "invoices" WHERE Total > 10
SELECT * FROM "invoices" WHERE Total > 10
    InvoiceId CustomerId
                                   InvoiceDate
                                                           BillingAddress \
                       23 2009-01-11 00:00:00
                                                          69 Salem Street
0
                       2 2009-02-11 00:00:00
1
           12
                                                  Theodor-Heuss-Straße 34
2
           19
                       40 2009-03-14 00:00:00
                                                          8. Rue Hanovre
3
           26
                       19 2009-04-14 00:00:00
                                                          1 Infinite Loop
           33
                       57 2009-05-15 00:00:00
4
                                                          Calle Lira, 198
          . . .
59
          383
                       10 2013-08-12 00:00:00 Rua Dr. Falcão Filho, 155
60
          390
                       48 2013-09-12 00:00:00
                                                    Lijnbaansgracht 120bg
61
          397
                       27 2013-10-13 00:00:00
                                                          1033 N Park Ave
62
          404
                       6 2013-11-13 00:00:00
                                                            Rilská 3174/6
                       44 2013-12-14 00:00:00
63
          411
                                                          Porthaninkatu 9
   BillingCity BillingState BillingCountry BillingPostalCode Total
0
                                        USA
                                                         2113 13.86
        Boston
                         MA
1
                                                        70174 13.86
     Stuttgart
                       None
                                    Germany
2
                                                        75002 13.86
         Paris
                       None
                                     France
     Cupertino
3
                         CA
                                        USA
                                                        95014 13.86
                                      Chile
                                                         None 13.86
4
     Santiago
                       None
                        . . .
                                        . . .
                                                          . . .
                                                                 . . .
59
     São Paulo
                         SP
                                     Brazil
                                                    01007-010 13.86
60
     Amsterdam
                         ۷V
                                Netherlands
                                                         1016 13.86
                                                        85719 13.86
61
        Tucson
                         ΑZ
                                        USA
62
                                                        14300 25.86
        Prague
                       None Czech Republic
      Helsinki
                                                        00530 13.86
63
                       None
                                    Finland
[64 rows x 9 columns]
```

Ollama parameters: model=aya: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 \* FROM \"invoices\" WHERE Total > 10\n\nThe following is in Ollama Response:

formation about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n InvoiceId in t64\nCustomerId int64\nInvoiceDate object\nBillingAddress object\nBillingCity object\nBillingState object\nBillingCountry object\nBillingPostalCode object\nTotal float64\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 -- j ust the code."}]

{'model': 'aya:latest', 'created\_at': '2024-06-14T11:50:09.976149558Z', 'message': {'role': 'assistant', 'c ontent': "```python\nimport pandas as pd\nimport plotly.express as px\n\n# Assuming your DataFrame is named 'df' and you want to plot the 'Total' column\nfig = px.bar(df, x='InvoiceId', y='Total')\nfig.update\_layout (title='Total Invoice Values')\nfig.show()\n```"}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 2 4046059090, 'load\_duration': 759433, 'prompt\_eval\_count': 210, 'prompt\_eval\_duration': 8561281000, 'eval\_count': 75, 'eval duration': 15352376000}



```
Out[27]: ('SELECT * FROM "invoices" WHERE Total > 10',
                                               InvoiceDate
               InvoiceId CustomerId
                                                                        BillingAddress \
                       5
                                   23 2009-01-11 00:00:00
           0
                                                                       69 Salem Street
           1
                      12
                                    2 2009-02-11 00:00:00
                                                               Theodor-Heuss-Straße 34
           2
                      19
                                   40 2009-03-14 00:00:00
                                                                        8, Rue Hanovre
           3
                      26
                                   19 2009-04-14 00:00:00
                                                                       1 Infinite Loop
           4
                      33
                                   57 2009-05-15 00:00:00
                                                                       Calle Lira, 198
                      . . .
                                  . . .
                     383
           59
                                   10
                                      2013-08-12 00:00:00
                                                             Rua Dr. Falcão Filho, 155
           60
                     390
                                   48 2013-09-12 00:00:00
                                                                 Lijnbaansgracht 120bg
           61
                     397
                                   27 2013-10-13 00:00:00
                                                                       1033 N Park Ave
           62
                     404
                                    6 2013-11-13 00:00:00
                                                                         Rilská 3174/6
           63
                                   44 2013-12-14 00:00:00
                     411
                                                                       Porthaninkatu 9
              BillingCity BillingState BillingCountry BillingPostalCode Total
           0
                   Boston
                                     MA
                                                    USA
                                                                      2113 13.86
           1
                Stuttgart
                                   None
                                                Germany
                                                                     70174 13.86
           2
                    Paris
                                   None
                                                 France
                                                                     75002 13.86
           3
                Cupertino
                                     CA
                                                     USA
                                                                     95014 13.86
           4
                 Santiago
                                   None
                                                   Chile
                                                                      None 13.86
                       . . .
                                    . . .
                                                     . . .
                                                                       . . .
                                                                               . . .
           . .
           59
                São Paulo
                                     SP
                                                 Brazil
                                                                 01007-010 13.86
           60
                Amsterdam
                                     ۷V
                                            Netherlands
                                                                      1016 13.86
                                                     USA
                                                                     85719 13.86
           61
                   Tucson
                                     ΑZ
           62
                                                                     14300 25.86
                   Prague
                                   None
                                         Czech Republic
           63
                 Helsinki
                                   None
                                                Finland
                                                                     00530 13.86
           [64 \text{ rows } \times 9 \text{ 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([ 5, 12, 19, 26, 33, 40, 47, 54, 61, 68, 75, 82, 88, 89,
                                       96, 103, 110, 117, 124, 131, 138, 145, 152, 159, 166, 173, 180, 187,
                                      193, 194, 201, 208, 215, 222, 229, 236, 243, 250, 257, 264, 271, 278,
```

```
285, 292, 298, 299, 306, 311, 312, 313, 320, 327, 334, 341, 348, 355,
                                     362, 369, 376, 383, 390, 397, 404, 411]),
                         'xaxis': 'x'.
                         'y': array([13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                                     13.86, 13.86, 17.91, 18.86, 21.86, 15.86, 13.86, 13.86, 13.86, 13.86,
                                     13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 14.91, 21.86,
                                     18.86, 15.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                                     13.86, 13.86, 13.86, 13.86, 10.91, 23.86, 16.86, 11.94, 10.91, 16.86,
                                     13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                                     13.86, 13.86, 25.86, 13.86]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'title': {'text': 'Total Invoice Values'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceId'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total'}}}
          }))
         question = """
In [28]:
             Find all invoices since 2010 and the total amount invoiced:
         0.00
         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 guery 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 "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR CustomerId INTEGER NOT NULL,\r\n EMENT NOT NULL.\r\n InvoiceDate DATETIME NOT NULL,\r\n BillinaA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n rv NVARCHAR(40),\r\n **FOREIG** N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n UnitPrice NUMERIC(10.2) NOT NULL.\r FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\t Quantity INTEGER NOT NULL,\r\n ON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t \t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON "invoice items" (InvoiceId)\n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK InvoiceLin eTrackId ON "invoice items" (TrackId)\n\nCREATE TABLE "employees"\r\n(\r\n EmployeeId INTEGER PRIMARY KE Y AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NUL L.\r\n Title NVARCHAR(30).\r\n ReportsTo INTEGER.\r\n BirthDate DATETIME.\r\n HireDate DATETIM E, r nAddress NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHA  $R(40), \r\n$ PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION O  $RCHAR(60).\r\n$ N UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMEN T NOT NULL.\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company City NVARCHAR(40),\r\n Address NVARCHAR(70),\r\n  $NVARCHAR(80).\r\n$ State NVARCHAR(40),\r\n Coun PostalCode NVARCHAR(10),\r\n trv NVARCHAR(40),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n SupportRepId INTEGER,\r\n Email NVARCHAR(60) NOT NULL,\r\n FOREIGN KEY (SupportRepId) REFERENCES "em ployees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n(\r TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n GenreId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n Id INTEGER.\r\n Composer NVARCHAR(220),\r Milliseconds INTEGER NOT NULL,\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (Albumid) REFERENCES "albums" (Albumid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO A CTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist trac PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n  $k"\r\n(\r\n$ CONSTRAINT PK Playlis tTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Playlist Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (Trac kId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite data base invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please gener ate a valid SQL query without any explanations for the question. \n2. If the provided context is almost suf ficient 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

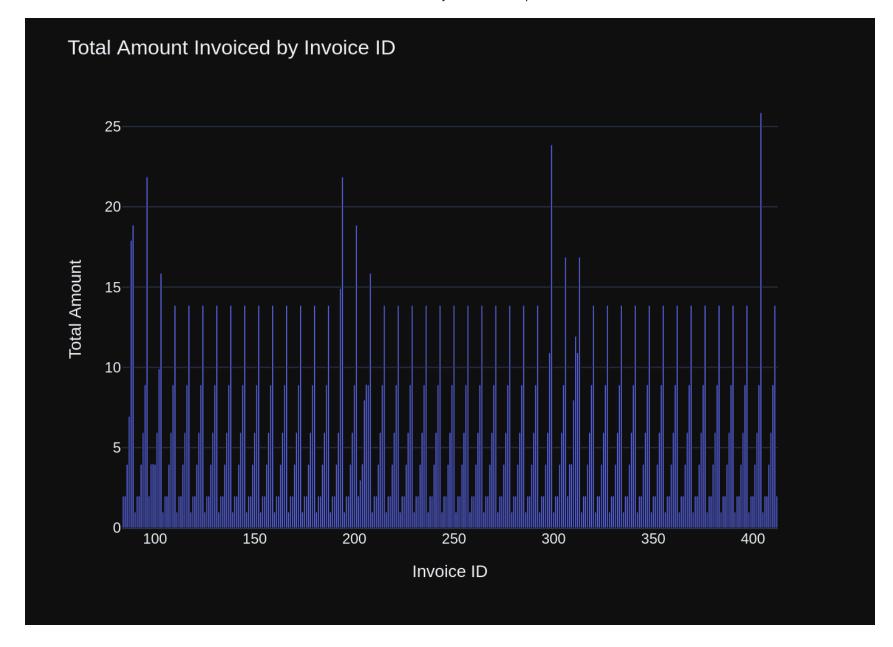
\_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please u se the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the a

nswer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n Find all invoices since 20 10 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceId, SUM(i.Total) A S TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate > \'2010-01-01\'\nGROUP BY i.InvoiceId'}, {'role': 'u ser', 'content': ' \n Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assist ant', 'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'role': 'user', 'content': ' \n Find all invoices since 2 010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount\nFROM "invoices" i\nWHERE strftime(\'%Y\', i.InvoiceDate) >= \'2010\'\nGROUP BY i.InvoiceI d\nORDER BY InvoiceAmount DESC'}, {'role': 'user', 'content': '\n List all invoices with a total excee ding \$10:\n'}, {'role': 'assistant', 'content': 'SELECT \*\nFROM "invoices" \nWHERE (Total > 10)\nGROUP BY I nvoiceId'}, {'role': 'user', 'content': ' \n List all invoices with a total exceeding \$10:\n'}, {'rol e': 'assistant', 'content': 'SELECT \* FROM "invoices" WHERE Total > 10'}, {'role': 'user', 'content': '\n Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry v. COUNT(i.InvoiceId) AS TotalInvoices\nFROM "invoices" i\nGROUP BY BillingCountry'}, {'role': 'user', 'con List all invoices with a total exceeding \$10:\n'}, {'role': 'assistant', 'content': 'SELECT \* \nFROM "invoices" \nWHERE Total > 10.00'}, {'role': 'user', 'content': ' \n Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(InvoiceId) AS Tot alInvoices\nFROM Invoices\nGROUP BY BillingCountry'}, {'role': 'user', 'content': ' \n Find the total n umber of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': ' \n Find th e total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT Customer.Country, COU NT(invoice.InvoiceId) AS InvoiceCount\nFROM "customers" customer\nJOIN "invoices" invoice ON customer.Custo merId = invoice.CustomerId\nGROUP BY Country'}, {'role': 'user', 'content': ' \n Find all invoices sinc e 2010 and the total amount invoiced:\n'}] Ollama parameters: model=ava: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 \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n Billin gAddress NVARCHAR(70).\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40).\r\n BillinaCou ntry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n F0RE IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL InvoiceId INTEGER NOT NULL,\r\n
TrackId INTEGER NOT NULL,\r\n L.\r\n UnitPrice NUMERIC(10,2) NO Ouantity INTEGER NOT NULL.\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceI T NULL,\r\n d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (Tra

ckid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON \"in voice items\" (InvoiceId)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDE X IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE \"employees\"\r\n(\r\n EmployeeI d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVAR CHAR(20) NOT NULL,\r\n Title NVARCHAR(30).\r\n ReportsTo INTEGER.\r\n BirthDate DATETIME.\r\n HireDate DATETIME,\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON D Email NVARCHAR(60),\r\n ELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n CustomerId INTEGER PRIMA RY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT N Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n ULL,\r\n State NVARC  $HAR(40).\r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n  $NVARCHAR(24).\r\n$ Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n FOREIGN KEY (Support RepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL.\r\n GenreId INTEGER,\r\n Comp Milliseconds INTEGER NOT NULL.\r\n oser NVARCHAR(220),\r\n Bvtes INTEGER.\r\n UnitPrice NUMERIC(1 FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON 0.2) NOT NULL,\r\n UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\t0N DELETE NO ACTION FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON D ON UPDATE NO ACTION,\r\n ELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\CREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n AUTOINCREMENT NOT NULL,\r\n EIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE \"playlist track\"\r\n(\r\n PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT N CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n ULL.\r\n FOREIGN KEY (PlavlistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Addi tional Context \n\nIn the SOLite database invoice means order\n\n===Response Guidelines \n1. If the provide d 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 co lumn, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the qu ery 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 all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": " \n "SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate > '2010-01-01'\n GROUP BY i.InvoiceId"}, {"role": "user", "content": " \n Find all invoices since 2010 and the total amo unt invoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFRO M \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"role": "user", "content Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "conten t": " \n t": "SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount\nFROM \"invoices\" i\nWHERE strftime('%Y', i.Invoice Date) >= '2010'\nGROUP BY i.InvoiceId\nORDER BY InvoiceAmount DESC"}, {"role": "user", "content": " \n

```
List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT *\nFROM \"invoi
ces\" \nWHERE (Total > 10)\nGROUP BY InvoiceId"}, {"role": "user", "content": " \n List all invoices wi
th a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM \"invoices\" WHERE Total > 1
0"}, {"role": "user", "content": " \n Find the total number of invoices per country:\n"}, {"role": "ass
istant", "content": "SELECT BillingCountry, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"invoices\" i\nGROUP
BY BillingCountry"}, {"role": "user", "content": " \n List all invoices with a total exceeding $1
0:\n"}, {"role": "assistant", "content": "SELECT * \nFROM \"invoices\" \nWHERE Total > 10.00"}, {"role": "u
ser". "content": " \n Find the total number of invoices per country:\n"}. {"role": "assistant". "conten
t": "SELECT BillingCountry, COUNT(InvoiceId) AS TotalInvoices\nFROM Invoices\nGROUP BY BillingCountry"},
{"role": "user", "content": " \n Find the total number of invoices per country:\n"}, {"role": "assistan
t", "content": "SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.Billing
Country"}, {"role": "user", "content": " \n Find the total number of invoices per country:\n"}, {"rol
e": "assistant", "content": "SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM \"cust
omers\" customer\nJOIN \"invoices\" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Countr
y"}, {"role": "user", "content": " \n Find all invoices since 2010 and the total amount invoiced:\n"}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:51:37.5025166Z', 'message': {'role': 'assistant', 'con
tent': 'SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-
01-01\'\nGROUP BY i.InvoiceId'}, 'done reason': 'stop', 'done': True, 'total duration': 87431054480, 'load
duration': 831166, 'prompt eval count': 1862, 'prompt eval duration': 76397222000, 'eval count': 50, 'eval
duration': 10350319000}
SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount
FROM "invoices" i
WHERE i.InvoiceDate >= '2010-01-01'
GROUP BY i.InvoiceId
SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount
FROM "invoices" i
WHERE i.InvoiceDate >= '2010-01-01'
GROUP BY i.InvoiceId
     InvoiceId TotalAmount
                      1.98
0
            84
            85
                      1.98
1
2
                      3.96
            86
3
            87
                      6.94
4
            88
                      17.91
                       . . .
           . . .
324
           408
                      3.96
325
           409
                      5.94
326
                      8.91
           410
327
           411
                     13.86
328
           412
                      1.99
```

```
[329 rows x 2 columns]
Ollama parameters:
model=aya: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 invoices since 2010 and the total amount invoic
ed:\n'\nThe DataFrame was produced using this query: SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\nFRO
M \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.InvoiceId\n\nThe following is informatio
n about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n InvoiceId
                                                                                          int64\nTotalAmou
      float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to cha
nt
rt 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 explanation
s -- just the code."}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:52:05.65230383Z', 'message': {'role': 'assistant', 'co
ntent': "```python\nimport plotly.express as px\n\n# Create a bar chart\nfiq = px.bar(df, x='InvoiceId', y
='TotalAmount')\n\n# Update layout\nfig.update layout(\n title='Total Amount Invoiced by Invoice ID',\n
xaxis title='Invoice ID',\n yaxis title='Total Amount'\n)\n\n# Show the plot\nfiq.show()\n``"}, 'done r
eason': 'stop', 'done': True, 'total duration': 28122681637, 'load duration': 747459, 'prompt eval count':
212, 'prompt eval duration': 8750979000, 'eval count': 93, 'eval duration': 19238768000}
```



```
Out[28]: ('SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01
          \'\nGROUP BY i.InvoiceId',
                InvoiceId TotalAmount
           0
                       84
                                  1.98
           1
                       85
                                  1.98
           2
                                  3.96
                       86
           3
                       87
                                  6.94
           4
                       88
                                 17.91
                      . . .
                                   . . .
           324
                      408
                                  3.96
           325
                      409
                                  5.94
           326
                      410
                                  8.91
           327
                      411
                                 13.86
           328
                      412
                                  1.99
           [329 rows \times 2 columns],
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'InvoiceId=%{x}<br>TotalAmount=%{y}<extra></extra>',
                          'legendgroup': '',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                          'name': '',
                         'offsetgroup': '',
                         'orientation': 'v',
                         'showlegend': False,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array([ 84, 85, 86, ..., 410, 411, 412]),
                         'xaxis': 'x',
                         'y': array([ 1.98, 1.98, 3.96, ..., 8.91, 13.86, 1.99]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'title': {'text': 'Total Amount Invoiced by Invoice ID'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Invoice ID'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total Amount'}}}
           }))
         question = """
In [29]:
             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 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 fo rmat instructions. \n===Tables \nCREATE INDEX IFK EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE TA EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n BLE "employees"\r\n(\r\n LastName NVARCHAR (20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGE BirthDate DATETIME,\r\n  $R_{i} r n$ HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n City NVARCHAR(4  $0), r\n$ State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NV  $ARCHAR(24), \r\n$ Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n  $n(\r\n$ FirstName NVARCHAR(40) NOT NUL  $L,\r\n$ LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n PostalCode NVARCHAR(1 City NVARCHAR(40),\r\n State NVARCHAR(40).\r\n Country NVARCHAR(40),\r\n Fax NVARCHAR(24),\r\n 0),\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\t0N DELETE NO A RepId INTEGER.\r\n CTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON "customers" (SupportRepId)\n\nCR InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n EATE TABLE "invoices"\r\n(\r\n CustomerId I NTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n BillingAddress NVARCHAR(70).\r\n Billing BillingState NVARCHAR(40),\r\n BillingCountry NVARCHAR(40),\r\n City NVARCHAR(40),\r\n BillingPost FOREIGN KEY (CustomerId) REFERENCES "cust alCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n omers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice items"\r InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r  $\n(\r\n$ TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Ouantity INTEGER NOT NUL  $L.\r\n$ FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE N FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "artists"\r\n(\r\n ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMEN T NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NU Composer NVARCHAR(220),\r\n LL,\r\n GenreId INTEGER,\r\n Milliseconds INTEGER NOT NULL.\r\n tes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" (Al FOREIGN KEY (GenreId) REFERENCES "genres" (G bumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n enreId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Title NVARCHAR(160) NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\t0N DELETE NO stId INTEGER NOT NULL,\r\n ACTION ON UPDATE NO ACTION\r\n\\nCREATE TABLE sglite stat1(tbl,idx,stat)\\n\\n===Additional Context \\n\nI n the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is suffici ent, please generate a valid SQL query without any explanations for the question. \n2. If the provided cont ext is almost sufficient but requires knowledge of a specific string in a particular column, please generat e 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 genera ted. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, p lease repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': " \n

employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': 'SELECT e.Firs tName, e.LastName, mt.FirstName AS ManagerFirstName, mt.LastName AS ManagerLastName\nFROM "employees" e\nLE FT JOIN "employees" mt ON e.ReportsTo = mt.EmployeeId'}, {'role': 'user', 'content': " \n List all empl oyees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': 'SELECT e1.\*, CONC AT(e2.FirstName, \' \', e2.LastName) AS ManagerName\nFROM "employees" e1\nLEFT JOIN "employees" e2 ON e1.Re portsTo = e2.EmployeeId\nORDER BY e1.EmployeeId'}, {'role': 'user', 'content': " \n List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': "SELECT e.FirstName, e.L astName, CASE WHEN e.ReportsTo IS NULL THEN 'None' ELSE (\n SELECT e2.FirstName | | ', ' || e2.LastName\n WHERE e2.EmployeeId = e.ReportsTo\n) END AS ManagerName\nFROM employees e"}, {'rol FROM employees e2\n e': 'user', 'content': ' \n Find the top 5 customers who spent the most money overall, \n int: order total can be found on invoices table, calculation using invoice items detail table is unnecessar y \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n List all customers from Canada and their email addresses:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent\nFROM Customers c\nJOIN Invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money overall, \n Hint: order total can be found on invoices table, calculation using invoice items detail table is un necessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM "customers" c\nJ0IN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpend ing DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most mone Hint: order total can be found on invoices table, calculation using invoice items v overall, \n detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS To talSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDE R BY TotalSpent DESC\nLIMIT 5\n-----'}, {'role': 'user', 'content': '\n customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.Invoic eId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Cus tomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Find the customer wi th the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(InvoiceId) as NumOfIn voices\nFROM "invoices"\nGROUP BY CustomerId\nORDER BY NumOfInvoices DESC\nLIMIT 1'}, {'role': 'user', 'con tent': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT c.Country. COUNT(\*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC \nLIMIT 5'}, {'role': 'user', 'content': " \n List all employees and their reporting manager's name (if any):\n"}] Ollama parameters: model=aya: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 EmployeeReportsTo ON \"employees\" (ReportsTo)\n\nCREATE

LastName NVAR TABLE \"employees\"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n CHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo IN TEGER,\r\n BirthDate DATETIME,\r\n HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n City NVARCH Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r\n  $AR(40), \r\n$ State NVARCHAR(40),\r\n Phon e NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENC ES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"custom ers\"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40).\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(1 Fax NVARCHAR(24),\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO RepId INTEGER,\r\n ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Custom erId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n BillingAddress NVARCHAR(70).\r\n illingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCountry NVARCHAR(40),\r\n Billi ngPostalCode NVARCHAR(10).\r\n Total NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice i InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n tems\"\r\n(\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Ouantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\t0N DELETE NO ACTION UPDATE NO ACTION.\r\n ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"artists\"\r\n(\r\n ArtistId INTEGER PRIMARY KEY AUTOINCREMENT Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY NOT NULL,\r\n MediaTypeId INT AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL.\r\n AlbumId INTEGER.\r\n EGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NUL Bvtes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES L.\r\n FOREIGN KEY (GenreId) REFERENC \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n ES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) R EFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABL E \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Title NVARCHAR(160) NO T NULL,\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite stat1(tbl,idx,stat)\n\n\n===Add itional Context \n\nIn the SOLite database invoice means order\n\n===Response Guidelines \n1. If the provid ed 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 co lumn, please generate an intermediate SQL guery to find the distinct strings in that column. Prepend the qu ery 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": List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "conten " \n t": "SELECT e.FirstName, e.LastName, mt.FirstName AS ManagerFirstName, mt.LastName AS ManagerLastName\nFROM \"employees\" e\nLEFT JOIN \"employees\" mt ON e.ReportsTo = mt.EmployeeId"}, {"role": "user", "content": "

List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT e1.\*, CONCAT(e2.FirstName, ' ', e2.LastName) AS ManagerName\nFROM \"employees\" e1\nLEFT JOIN \"emp loyees\" e2 ON e1.ReportsTo = e2.EmployeeId\nORDER BY e1.EmployeeId"}, {"role": "user", "content": " \n List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELE CT e.FirstName, e.LastName, CASE WHEN e.ReportsTo IS NULL THEN 'None' ELSE (\n SELECT e2.FirstName || '. FROM employees e2\n WHERE e2.EmployeeId = e.ReportsTo\n) END AS ManagerName\nFROM ' || e2.LastName\n employees e"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most money over Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpen t\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER B Y TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all customers from Canada and thei r email addresses:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, c.Email, SUM(i.Total) AS Tot alSpent\nFROM Customers c\nJOIN Invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 1"}, {"role": "user", "content": " \n Find the top 5 customers who spent the mo \n Hint: order total can be found on invoices table, calculation using invoice st money overall. \n items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Tota l) AS TotalSpending\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.C ustomerId\nORDER BY TotalSpending DESC\nLIMIT 5"}, {"role": "user", "content": " \n tomers who spent the most money overall, \n \n Hint: order total can be found on invoices table, ca lculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c. CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Custo merId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n------"}, {"role": "use Find the customer with the most invoices \n"}, {"role": "assistant", "content": "S ELECT c.CustomerId. COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.Cus tomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "co Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT C ntent": " \n ustomerId, COUNT(InvoiceId) as NumOfInvoices\nFROM \"invoices\"\nGROUP BY CustomerId\nORDER BY NumOfInvoice s DESC\nLIMIT 1"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(\*) AS TotalCustomers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all employ ees and their reporting manager's name (if any):\n"}]

Add of existing embedding ID: c7de9fac-1104-5409-b17a-73b533b767d5-sql Insert of existing embedding ID: c7de9fac-1104-5409-b17a-73b533b767d5-sql

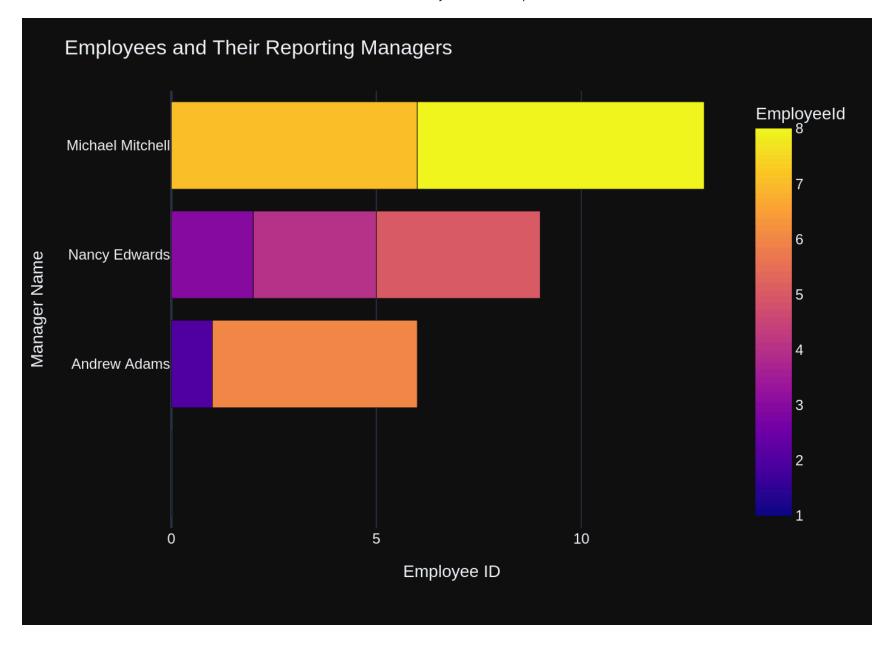
```
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:53:42.283101278Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT e1.*, CONCAT(e2.FirstName, \' \', e2.LastName) AS ManagerName\nFROM "employees" e1\nLEFT J
OIN "employees" e2 ON e1.ReportsTo = e2.EmployeeId\nORDER BY e1.EmployeeId'}, 'done reason': 'stop', 'don
e': True, 'total duration': 96520943358, 'load duration': 880957, 'prompt eval count': 1980, 'prompt eval d
uration': 83432847000, 'eval count': 59, 'eval duration': 12384361000}
SELECT e1.*, CONCAT(e2.FirstName, ' ', e2.LastName) AS ManagerName
FROM "employees" el
LEFT JOIN "employees" e2 ON e1.ReportsTo = e2.EmployeeId
ORDER BY e1.EmployeeId
SELECT e1.*, CONCAT(e2.FirstName, ' ', e2.LastName) AS ManagerName
FROM "employees" el
LEFT JOIN "employees" e2 ON e1.ReportsTo = e2.EmployeeId
ORDER BY e1.EmployeeId
  EmployeeId LastName FirstName
                                               Title ReportsTo \
                                     General Manager
                 Adams
0
           1
                          Andrew
                                                            NaN
1
           2
              Edwards
                                                            1.0
                           Nancy
                                        Sales Manager
2
           3 Peacock
                                                            2.0
                            Jane Sales Support Agent
3
                                                            2.0
                  Park Margaret Sales Support Agent
                           Steve Sales Support Agent
4
             Johnson
                                                            2.0
5
           6 Mitchell
                       Michael
                                           IT Manager
                                                            1.0
6
           7
                          Robert
                                            IT Staff
                                                            6.0
                  Kina
7
           8 Callahan
                                            IT Staff
                                                            6.0
                         Laura
            BirthDate
                                  HireDate
                                                               Address \
                                                   11120 Jasper Ave NW
0 1962-02-18 00:00:00 2002-08-14 00:00:00
                                                          825 8 Ave SW
1 1958-12-08 00:00:00 2002-05-01 00:00:00
2 1973-08-29 00:00:00 2002-04-01 00:00:00
                                                         1111 6 Ave SW
3 1947-09-19 00:00:00 2003-05-03 00:00:00
                                                      683 10 Street SW
4 1965-03-03 00:00:00 2003-10-17 00:00:00
                                                          7727B 41 Ave
                                                   5827 Bowness Road NW
5 1973-07-01 00:00:00 2003-10-17 00:00:00
6 1970-05-29 00:00:00 2004-01-02 00:00:00 590 Columbia Boulevard West
7 1968-01-09 00:00:00 2004-03-04 00:00:00
                                                           923 7 ST NW
        City State Country PostalCode
                                                   Phone
                                                                       Fax \
0
     Edmonton
                AB Canada
                              T5K 2N1 +1 (780) 428-9482 +1 (780) 428-3457
1
     Calgary
                AB Canada
                              T2P 2T3 +1 (403) 262-3443 +1 (403) 262-3322
2
                AB Canada
                             T2P 5M5 +1 (403) 262-3443 +1 (403) 262-6712
     Calgary
3
                AB Canada
     Calgary
                              T2P 5G3 +1 (403) 263-4423 +1 (403) 263-4289
4
     Calgary
                AB Canada
                              T3B 1Y7
                                      1 (780) 836-9987 1 (780) 836-9543
5
     Calgary
                AB Canada
                              T3B 0C5 +1 (403) 246-9887 +1 (403) 246-9899
6 Lethbridge
                AB Canada
                              T1K 5N8 +1 (403) 456-9986 +1 (403) 456-8485
```

7 Lethbridge

AB Canada

T1H 1Y8 +1 (403) 467-3351 +1 (403) 467-8772

```
Email
                                ManagerName
0
     andrew@chinookcorp.com
1
      nancv@chinookcorp.com
                               Andrew Adams
2
                              Nancy Edwards
      jane@chinookcorp.com
  margaret@chinookcorp.com
                              Nancy Edwards
4
      steve@chinookcorp.com
                              Nancy Edwards
5
   michael@chinookcorp.com
                               Andrew Adams
     robert@chinookcorp.com Michael Mitchell
7
     laura@chinookcorp.com Michael Mitchell
Ollama parameters:
model=aya: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 (i
f any):\n'\nThe DataFrame was produced using this guery: SELECT e1.*, CONCAT(e2.FirstName, ' ', e2.LastNa
me) AS ManagerName\nFROM \"employees\" e1\nLEFT JOIN \"employees\" e2 ON e1.ReportsTo = e2.EmployeeId\nORDE
R BY e1.EmployeeId\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.
                                                     object\nFirstName
                               int64\nLastName
                                                                            object\nTitle
dtypes gives:\n EmployeeId
                                                                                                    obj
                   float64\nBirthDate
                                           object\nHireDate
ect\nReportsTo
                                                                  obiect\nAddress
                                                                                         object\nCity
                                                                     object\nPhone
                                                                                            obiect\nFax
object\nState
                       object\nCountry
                                              obiect\nPostalCode
                                              object\ndtype: object"}, {"role": "user", "content": "Can y
obiect\nEmail
                       object\nManagerName
ou generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas da
taframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Pytho
n code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:54:37.943937815Z', 'message': {'role': 'assistant', 'c
ontent': '```python\nimport plotly.express as px\n\n# Assuming \'EmployeeId\' is the index and first column
\' column for ordering\n# fig = px.bar(df, x=df[\ReportsTo\'], y="ManagerName", color="EmployeeId")\n\nfi
                    title="Employees and Their Reporting Managers",\n
                                                                       xaxis title="Employee ID",\n
yaxis title="Manager Name"\n)\n\n# If you want to show only employees without a manager (if any)\n# filter
values = df[\'ManagerName'].dropna().tolist()\n# fig.add trace(px.bar(filter values, x=range(len(filter values))))
lues)), marker color="blue").data[0])\n\nfig.show()\n```'}, 'done reason': 'stop', 'done': True, 'total dur
ation': 55633883231, 'load duration': 43550316, 'prompt eval count': 284, 'prompt eval duration': 120087960
00, 'eval count': 207, 'eval duration': 43525964000}
```



```
('SELECT e1.*, CONCAT(e2.FirstName, \' \', e2.LastName) AS ManagerName\nFROM "employees" e1\nLEFT JOIN "em
Out[29]:
         ployees" e2 ON e1.ReportsTo = e2.EmployeeId\nORDER BY e1.EmployeeId',
              EmployeeId LastName FirstName
                                                           Title ReportsTo \
          0
                                                  General Manager
                       1
                            Adams
                                      Andrew
                                                                         NaN
          1
                      2
                          Edwards
                                                    Sales Manager
                                                                         1.0
                                      Nancy
          2
                       3
                           Peacock
                                                                         2.0
                                       Jane Sales Support Agent
          3
                             Park Margaret Sales Support Agent
                       4
                                                                         2.0
          4
                                      Steve Sales Support Agent
                       5
                          Johnson
                                                                         2.0
          5
                         Mitchell
                                    Michael
                                                       IT Manager
                                                                         1.0
          6
                       7
                                                                         6.0
                             King
                                      Robert
                                                         IT Staff
          7
                                                         IT Staff
                                                                         6.0
                         Callahan
                                       Laura
                        BirthDate
                                              HireDate
                                                                            Address \
                                                                11120 Jasper Ave NW
             1962-02-18 00:00:00 2002-08-14 00:00:00
          1 1958-12-08 00:00:00
                                  2002-05-01 00:00:00
                                                                       825 8 Ave SW
          2 1973-08-29 00:00:00 2002-04-01 00:00:00
                                                                      1111 6 Ave SW
          3 1947-09-19 00:00:00 2003-05-03 00:00:00
                                                                   683 10 Street SW
            1965-03-03 00:00:00 2003-10-17 00:00:00
                                                                       7727B 41 Ave
          5 1973-07-01 00:00:00 2003-10-17 00:00:00
                                                               5827 Bowness Road NW
            1970-05-29 00:00:00 2004-01-02 00:00:00
                                                        590 Columbia Boulevard West
          7 1968-01-09 00:00:00 2004-03-04 00:00:00
                                                                        923 7 ST NW
                   City State Country PostalCode
                                                                                    Fax \
                                                               Phone
          0
                           AB Canada
                                          T5K 2N1 +1 (780) 428-9482 +1 (780) 428-3457
               Edmonton
          1
                           AB Canada
                                          T2P 2T3 +1 (403) 262-3443
                                                                     +1 (403) 262-3322
                Calgary
          2
                                          T2P 5M5 +1 (403) 262-3443 +1 (403) 262-6712
                Calgary
                           AB Canada
          3
                Calgary
                           AB Canada
                                          T2P 5G3 +1 (403) 263-4423 +1 (403) 263-4289
          4
                                                   1 (780) 836-9987
                Calgary
                           AB Canada
                                          T3B 1Y7
                                                                      1 (780) 836-9543
          5
                                         T3B 0C5 +1 (403) 246-9887
                           AB Canada
                Calgary
                                                                     +1 (403) 246-9899
          6
             Lethbridge
                               Canada
                                          T1K 5N8 +1 (403) 456-9986
                           AB
                                                                     +1 (403) 456-8485
             Lethbridge
                           AB Canada
                                          T1H 1Y8 +1 (403) 467-3351 +1 (403) 467-8772
                                            ManagerName
                                 Email
          0
               andrew@chinookcorp.com
          1
                                           Andrew Adams
                nancy@chinookcorp.com
          2
                                          Nancy Edwards
                  jane@chinookcorp.com
          3
             margaret@chinookcorp.com
                                          Nancy Edwards
          4
                 steve@chinookcorp.com
                                          Nancy Edwards
          5
                                           Andrew Adams
              michael@chinookcorp.com
          6
                                       Michael Mitchell
               robert@chinookcorp.com
          7
                laura@chinookcorp.com
                                       Michael Mitchell ,
          Figure({
```

```
'data': [{'alignmentgroup': 'True',
                       'hovertemplate': 'index=%{x}<br>ManagerName=%{y}<br>EmployeeId=%{marker.color}<extra></extr
         a>',
                       'leaendaroup': ''.
                       'marker': {'color': array([1, 2, 3, 4, 5, 6, 7, 8]), 'coloraxis': 'coloraxis', 'pattern':
         {'shape': ''}},
                       'name': '',
                       'offsetgroup': '',
                       'orientation': 'h'.
                       'showlegend': False,
                       'textposition': 'auto',
                       'type': 'bar',
                       'x': array([0, 1, 2, 3, 4, 5, 6, 7]),
                       'xaxis': 'x',
                       'y': array([' ', 'Andrew Adams', 'Nancy Edwards', 'Nancy Edwards', 'Nancy Edwards',
                                  'Andrew Adams', 'Michael Mitchell', 'Michael Mitchell'], dtype=object),
                       'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                        'coloraxis': {'colorbar': {'title': {'text': 'EmployeeId'}},
                                     'colorscale': [[0.0, '#0d0887'], [0.111111111111111,
                                                    '#46039f'], [0.22222222222222,
                                                    '#bd3786'], [0.555555555555556,
                                                    '#ed7953'], [0.7777777777778,
                                                    '#fb9f3a'], [0.888888888888888,
                                                    '#fdca26'], [1.0, '#f0f921']]},
                        'legend': {'tracegroupgap': 0},
                        'margin': {'t': 60},
                        'template': '...',
                        'title': {'text': 'Employees and Their Reporting Managers'},
                        'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Employee ID'}},
                        'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Manager Name'}}}
          }))
        question = """
In [30]:
            Get the average invoice total for each customer:
         0.00
        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 fo rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR EMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n BillingA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n **FOREIG** rv NVARCHAR(40).\r\n N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON "invoice items" (InvoiceId)\n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KE InvoiceId INTEGER NOT NULL,\r\n Y AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES Price NUMERIC(10,2) NOT NULL,\r\n "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERE NCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLin eTrackId ON "invoice items" (TrackId)\n\nCREATE TABLE sqlite stat1(tbl,idx,stat)\n\nCREATE INDEX IFK Custom erSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRI MARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL.\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n  $CHAR(40), \r\n$ PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24).\r\n Email NVARCHAR(60) NOT NULL,\r\n  $\times NVARCHAR(24).\r\n$ SupportRepId INTEGER,\r\n FOREIGN KEY (Suppo rtRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE TABLE "employees"\r\n(\r\n LastName NVARCHAR(20) NOT NULL,\r\n NTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER.\r\n BirthDate DATETIME.\r\n R(20) NOT NULL,\r\n Hir State NVARCHAR(40),\r\n eDate DATETIME,\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n Co untry NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\t0N DELETE NO Email NVARCHAR(60),\r\n ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n \n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query with out any explanations for the question. \n2. If the provided context is almost sufficient but requires knowl edge of a specific string in a particular column, please generate an intermediate SQL guery to find the dis tinct strings in that column. Prepend the guery 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 gi ven before. \n'}, {'role': 'user', 'content': ' \n Get the average invoice total for each custome r:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "c ustomers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'c Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'S ELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i ON c.Custome rId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY AverageInvoiceTotal DESC\nLIMIT 10'}, {'role': 'user', Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'content': ' \n 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJ0IN "invoices" i ON c.Custo merId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY AverageInvoiceTotal DESC'}, {'role': 'user', 'conten

t': ' \n Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELE CT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoices table, calculation u sing invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n------'}, {'role': 'user', 'content': ' Find the top 5 customers who spent the most money overall. \n \n Hint: order total can be fo und on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistan t', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money overall, \n Hint: order t otal can be found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'ro le': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the total number of invoices per country:\n'}, {'role': 'assistan t', 'content': 'SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM "customers" custome r\nJOIN "invoices" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country'}, {'role': 'use Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistan r'. 'content': ' \n t', 'content': 'SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate > \'2010-01-01\'\nGROUP BY i.InvoiceId'}, {'role': 'user', 'content': ' \n Find the customer with the mo st invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoice s\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY To talInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': '\n Get the average invoice total for each cu stomer:\n'}] Ollama parameters: model=aya: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 \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL.\r\n InvoiceDate DATETIME NOT NULL.\r\n Billin gAddress NVARCHAR(70).\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillinaCou BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n ntrv NVARCHAR(40).\r\n F0RE IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK InvoiceLineInv oiceId ON \"invoice items\" (InvoiceId)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL,\r UnitPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Tr

ackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE sqlite stat1(tbl,idx,stat)\n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"customers\"\r\n(\r\n FirstName NVARCHAR(40) NOT NULL,\r\n stomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastNa me NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(4  $0), \r\n$ State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n  $ARCHAR(24), \r\n$ Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO A CTION\r\n)\n\CREATE INDEX IFK EmployeeReportsTo ON \"employees\" (ReportsTo)\n\nCREATE TABLE \"employees EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NUL FirstName NVARCHAR(20) NOT NULL,\r\n L.\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER,\r\n Bir HireDate DATETIME.\r\n thDate DATETIME.\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40).\r\n St PostalCode NVARCHAR(10).\r\n ate NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Phone NVARCHAR(24),\r Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQL ite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, plea se generate a valid SOL guery without any explanations for the guestion. \n2. If the provided context is al most sufficient but requires knowledge of a specific string in a particular column, please generate an inte rmediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying int ermediate 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 repe at the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n Get the average in voice total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.C ustomerId"}, {"role": "user", "content": " \n Get the average invoice total for each customer:\n"}, {"r ole": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY AverageInvoiceTotal DESC\nLIMIT 10"}, {"role": "user", "content": " \n Get the average invoice total for each custome r:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY Averag eInvoiceTotal DESC"}, {"role": "user", "content": " \n Get the total number of invoices for each custom er\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "use Find the top 5 customers who spent the most money overall, \n r", "content": " \n er total can be found on invoices table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJ0 IN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5 \n-----"}, {"role": "user", "content": " \n Find the top 5 customers who spent the Hint: order total can be found on invoices table, calculation using invoi most money overall, \n \n ce items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Tot al) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Cus tomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the top 5 customer

```
s who spent the most money overall, \n \n Hint: order total can be found on invoices table, calcula
tion using invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.Custo
merId, SUM(i.Total) AS TotalSpending\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Custome
rId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5"}, {"role": "user", "content": " \n
nd the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT Customer.Countr
y, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM \"customers\" customer\nJOIN \"invoices\" invoice ON cust
omer.CustomerId = invoice.CustomerId\nGROUP BY Country"}, {"role": "user", "content": " \n Find all inv
oices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceId,
SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate > '2010-01-01'\nGROUP BY i.InvoiceI
d"}, {"role": "user", "content": " \n Find the customer with the most invoices \n"}, {"role": "assista
nt", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"inv
oices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"},
{"role": "user", "content": " \n Get the average invoice total for each customer:\n"}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:56:05.887708575Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJ0IN "invoices" i
ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId;'}, 'done reason': 'stop', 'done': True, 'total durat
ion': 87823527216, 'load duration': 600799, 'prompt eval count': 1783, 'prompt eval duration': 76828344000,
'eval count': 50, 'eval duration': 10392615000}
SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId:
Output from LLM: SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId:
Extracted SQL: SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
   CustomerId AverageInvoiceTotal
0
             1
                          5.660000
1
             2
                          5.374286
2
             3
                          5.660000
3
             4
                          5.660000
4
             5
                          5.802857
5
                          7.088571
```

6 7	7 8	6.088571 5.374286
8	9	5.374286
9	10	5.374286
10	11	5.374286
11	12	5.374286
12	13	5.374286
13	14	5.374286
14	15	5.517143
15	16	5.374286
16	17	5.660000
17	18	5.374286
18	19	5.517143
19	20	5.660000
20	21	5.374286
21	22	5.660000
22	23	5.374286
23	24	6.231429
24	25	6.088571
25	26	6.802857
26	27	5.374286
27	28	6.231429
28	29	5.374286
29	30	5.374286
30	31	5.374286
31	32	5.374286
32	33	5.374286
33	34	5.660000
34	35	5.374286
35	36	5.374286
36	37	6.231429
37	38	5.374286
38	39	5.517143
39	40	5.517143
40	41	5.374286
41	42	5.660000
42	43	5.802857
43	44	5.945714
44	45	6.517143
45	46	6.517143
46	47	5.374286
47	48	5.802857

48	49	5.374286
49	50	5.374286
50	51	5.517143
51	52	5.374286
52	53	5.374286
53	54	5.374286
54	55	5.374286
55	56	5.374286
56	57	6.660000
57	58	5.517143
58	59	6.106667
0llama	parameters:	

Ollama parameters:

model=aya: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'\nT he DataFrame was produced using this guery: SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM  $\colored{Continuous}$  c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\n\nThe followin g is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId int64\nAverageInvoiceTotal float64\ndtype: object"}, {"role": "user", "content": "Can you generate the P ython plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'd f'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not an swer with any explanations -- just the code."}]

## Ollama Response:

{'model': 'aya:latest', 'created at': '2024-06-14T11:56:33.118437311Z', 'message': {'role': 'assistant', 'c ontent': "```python\nimport pandas as pd\nimport plotly.express as px\n\n# Assuming the dataframe is named 'df' and contains a column called 'CustomerId' and a column called 'AverageInvoiceTotal'\nfig = px.scatter (df, x='CustomerId', y='AverageInvoiceTotal')\nfig.update layout(title='Average Invoice Total by Customer') \nfig.show()\n```"}, 'done reason': 'stop', 'done': True, 'total duration': 27203051491, 'load duration': 4 2228539, 'prompt eval count': 204, 'prompt eval duration': 8836657000, 'eval count': 87, 'eval duration': 1 8269933000}



ustome	erld = i.Cu	stomerId\nGROUP BY c.
C	ustomerId	AverageInvoiceTotal
0	1	5.660000
1	2	5.374286
2	3	5.660000
3	4	5.660000
4	5	5.802857
5	6	7.088571
6	7	6.088571
7	8	5.374286
8	9	5.374286
9	10	5.374286
10	11	5.374286
11	12	5.374286
12	13	5.374286
13	14	5.374286
14	15	5.517143
15	16	5.374286
16	17	5.660000
17	18	5.374286
18	19	5.517143
19	20	5.660000
20	21	5.374286
21	22	5.660000
22	23	5.374286
23	24	6.231429
24	25	6.088571
25	26	6.802857
26	27	5.374286
27	28	6.231429
28	29	5.374286
29	30	5.374286
30	31	5.374286
31	32	5.374286
32	33	5.374286
33	34	5.660000
34	35	5.374286
35	36	5.374286
36	37	6.231429
37	38	5.374286
38	39	5.517143

```
5.517143
39
            40
40
            41
                           5.374286
41
            42
                           5.660000
42
            43
                           5.802857
43
            44
                           5.945714
            45
44
                           6.517143
45
            46
                           6.517143
46
            47
                           5.374286
47
            48
                           5.802857
48
            49
                           5.374286
            50
49
                           5.374286
50
            51
                           5.517143
51
            52
                           5.374286
52
            53
                           5.374286
53
            54
                           5.374286
54
            55
                           5.374286
55
            56
                           5.374286
            57
56
                           6.660000
57
            58
                           5.517143
58
            59
                           6.106667,
Figure({
    'data': [{'hovertemplate': 'CustomerId=%{x}<br>AverageInvoiceTotal=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'symbol': 'circle'},
              'mode': 'markers'.
              'name': '',
              'orientation': 'v',
              'showlegend': False,
              'type': 'scatter',
              'x': array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
                          19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
                           37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
                           55, 56, 57, 58, 591).
              'xaxis': 'x',
              'y': array([5.66
                                     , 5.37428571, 5.66
                                                              , 5.66
                                                                          , 5.80285714, 7.08857143,
                           6.08857143, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571,
                           5.37428571, 5.37428571, 5.51714286, 5.37428571, 5.66
                           5.51714286, 5.66
                                                 , 5.37428571, 5.66
                                                                         , 5.37428571, 6.23142857,
                           6.08857143, 6.80285714, 5.37428571, 6.23142857, 5.37428571, 5.37428571,
                           5.37428571, 5.37428571, 5.37428571, 5.66
                                                                          , 5.37428571, 5.37428571,
                           6.23142857, 5.37428571, 5.51714286, 5.51714286, 5.37428571, 5.66
                           5.80285714, 5.94571429, 6.51714286, 6.51714286, 5.37428571, 5.80285714,
```

```
5.37428571, 5.37428571, 5.51714286, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.3742
```

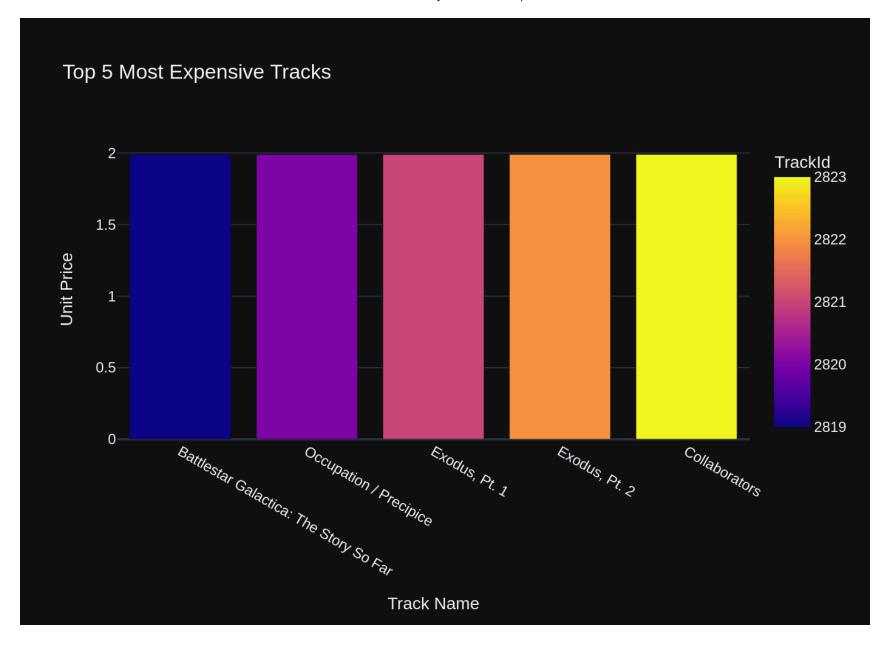
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMEN Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n T NOT NULL,\r\n MediaTypeId INTEGER NOT NU LL,\r\n GenreId INTEGER.\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n tes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" (Al bumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (G enreId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\CREATE INDEX IFK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDEX IFK TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK PlaylistTr ackTrackId ON "playlist track" (TrackId)\n\nCREATE INDEX IFK InvoiceLineTrackId ON "invoice items" (TrackI d)\n\nCREATE INDEX IFK TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n d INTEGER NOT NULL.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n REIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r \n)\n\nCREATE TABLE "playlist track"\r\n(\r\n PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlavlistI d) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDE X IFK AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "albums"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n EIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n \n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL guery without any explanations for the gues tion. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a pa rticular column, please generate an intermediate SQL query to find the distinct strings in that column. Pre pend 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 guestion has be en asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'use r', 'content': ' \n Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistan t', 'content': 'SELECT Name, UnitPrice\nFROM "tracks"\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5\n--------'}, {'role': 'user', 'content': '\n Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n There are 3 tables: artists, a lbums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumI d.\n Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistan t', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJ0IN "albums" al ON a.Art istId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC \nLIMIT 10'}, {'role': 'user', 'content': '\n There are 3 tables: artists, albums and tracks, where alb

ums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based on the number of tracks\n'\}, {'role': 'assistant', 'content': 'SELECT a.Artis tId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistI d = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTrac ks DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, Find the top 5 customers who bought the most albums in total quantity (across all invoice s):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "cus tomers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.In voiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most al bums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, C OUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\n GROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n-----'}, {'role': 'user', 'c Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELE CT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceI d = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': Find the top 5 customers who spent the most money overall, \n \n found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assist ant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n------------'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n ind the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJ0 IN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMI T 5'}, {'role': 'user', 'content': ' \n Find the top 5 most expensive tracks (based on unit pric e):\n'}] Ollama parameters: model=aya:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery 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 \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREM ENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres \" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK Trac kAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDEX IFK TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX I

FK PlaylistTrackTrackId ON \"playlist track\" (TrackId)\n\nCREATE INDEX IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE INDEX IFK TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE TABLE \"invoi InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n ce items\"\r\n(\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Ouantity INTEG FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\t0N DELETE NO ACTIO ER NOT NULL.\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO AC N ON UPDATE NO ACTION,\r\n TION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist track\"\r\n(\r\n PlaylistId INTEGER NOT NUL TrackId INTEGER NOT NULL.\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackI L.\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UP d), r nFOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"albums AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL.\r  $\"\r\n(\r\n$ FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\t0 ArtistId INTEGER NOT NULL.\r\n N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice m eans order $\n$ ===Response Guidelines  $\n$ 1. If the provided context is sufficient, please generate a valid SQ L query without any explanations for the question. \n2. If the provided context is almost sufficient but re quires 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 re levant table(s). \n5. If the guestion 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 most expensive tracks (ba sed on unit price):\n"}, {"role": "assistant", "content": "SELECT Name, UnitPrice\nFROM \"tracks\"\nORDER B Y UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the top 5 most expensive tracks (bas ed on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tra cks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5\n------"}, {"role": "user", "content": " \n Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", There are 3 tables: artists, albums and tracks, where albums and artists are linked by A rtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based o n the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTrack s\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t. AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": "\n are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and track s are linked by AlbumId,\n Can you find the top 10 most popular artists based on the number of tracks \n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalT racks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bough t the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c. CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY Tota lAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice item

```
s, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoice
s):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"in
voices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalA
lbums DESC\nLIMIT 5\n------}, {"role": "user", "content": "\n Hint: album quantity
is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantit
y (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS T
otalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Custom
erId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n
                                                                               Find the top 5 customers
who spent the most money overall, \n \n Hint: order total can be found on invoices table, calculati
on using invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.Custome
rId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\n
GROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n------"}, {"role": "user", "co
                Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who b
ought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELEC
T i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON
i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user",
"content": " \n
                 Find the top 5 most expensive tracks (based on unit price):\n"}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:58:03.47402083Z', 'message': {'role': 'assistant', 'co
ntent': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nWHERE t.UnitPrice = (\n
                                                                                         SELECT MAX(Uni
            FROM "tracks"\n)\nLIMIT 5'}, 'done reason': 'stop', 'done': True, 'total duration': 902602673
tPrice)\n
86, 'load duration': 724111, 'prompt eval count': 1834, 'prompt eval duration': 79384834000, 'eval count':
49, 'eval duration': 10270028000}
SELECT t.TrackId, t.Name, t.UnitPrice
FROM "tracks" t
WHERE t.UnitPrice = (
   SELECT MAX(UnitPrice)
   FROM "tracks"
LIMIT 5
SELECT t.TrackId, t.Name, t.UnitPrice
FROM "tracks" t
WHERE t.UnitPrice = (
   SELECT MAX(UnitPrice)
   FROM "tracks"
)
LIMIT 5
  TrackId
                                            Name UnitPrice
0
     2819 Battlestar Galactica: The Story So Far
                                                       1.99
                           Occupation / Precipice
1
     2820
                                                       1.99
2
     2821
                                    Exodus, Pt. 1
                                                       1.99
3
      2822
                                    Exodus, Pt. 2
                                                       1.99
```

```
4
      2823
                                     Collaborators
                                                        1.99
Ollama parameters:
model=aya: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 most expensive tracks (based on unit pric
e):\n'\nThe DataFrame was produced using this query: SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"track
                               SELECT MAX(UnitPrice)\n FROM \"tracks\"\n)\nLIMIT 5\n\nThe following
s\" t\nWHERE t.UnitPrice = (\n
is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n TrackId
                                                                                                     int64
                                    float64\ndtype: object"}, {"role": "user", "content": "Can you generat
                obiect\nUnitPrice
\nName
e the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe ca
lled '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': 'aya:latest', 'created at': '2024-06-14T11:58:29.881413717Z', 'message': {'role': 'assistant', 'c
ontent': "```python\nimport plotly.express as px\n\n# Create a bar chart\nfiq = px.bar(df, x='Name', y='Uni
tPrice', color='TrackId', title='Top 5 Most Expensive Tracks')\nfig.update xaxes(title='Track Name')\nfig.u
pdate yaxes(title='Unit Price')\nfig.show()\n```"}, 'done reason': 'stop', 'done': True, 'total duration':
26380112094, 'load duration': 44296480, 'prompt eval count': 214, 'prompt eval duration': 8889221000, 'eval
count': 84, 'eval duration': 17394527000}
```



```
Out[31]: ('SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nWHERE t.UnitPrice = (\n
                                                                                         SELECT MAX(UnitPric
                FROM "tracks"\n)\nLIMIT 5',
         e)\n
            TrackId
                                                     Name UnitPrice
          0
               2819 Battlestar Galactica: The Story So Far
                                                                1.99
          1
               2820
                                    Occupation / Precipice
                                                                1.99
          2
               2821
                                             Exodus, Pt. 1
                                                                1.99
          3
               2822
                                             Exodus, Pt. 2
                                                                1.99
          4
               2823
                                             Collaborators
                                                                1.99,
          Figure({
              'data': [{'alignmentgroup': 'True',
                       'hovertemplate': 'Name=%{x}<br>UnitPrice=%{y}<br>TrackId=%{marker.color}<extra></extra>',
                       'legendgroup': '',
                       'marker': {'color': array([2819, 2820, 2821, 2822, 2823]),
                                  'coloraxis': 'coloraxis',
                                  'pattern': {'shape': ''}},
                       'name': '',
                       'offsetgroup': '',
                       'orientation': 'v',
                       'showlegend': False,
                       'textposition': 'auto',
                       'type': 'bar',
                       'x': array(['Battlestar Galactica: The Story So Far', 'Occupation / Precipice',
                                   'Exodus, Pt. 1', 'Exodus, Pt. 2', 'Collaborators'], dtype=object),
                       'xaxis': 'x',
                       'y': array([1.99, 1.99, 1.99, 1.99, 1.99]),
                       'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                        'coloraxis': {'colorbar': {'title': {'text': 'TrackId'}},
                                     'colorscale': [[0.0, '#0d0887'], [0.111111111111111,
                                                    '#46039f'], [0.22222222222222,
                                                    '#bd3786'], [0.555555555555556,
                                                    '#ed7953'], [0.7777777777778,
                                                    '#fb9f3a'], [0.88888888888888888,
                                                    '#fdca26'], [1.0, '#f0f921']]},
                        'legend': {'tracegroupgap': 0},
                        'template': '...',
                        'title': {'text': 'Top 5 Most Expensive Tracks'},
                        'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Track Name'}},
```

```
'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Unit Price'}}}
}))

In [32]: 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 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 fo rmat instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMEN Name NVARCHAR(200) NOT NULL,\r\n T NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NU LL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL,\r\n tes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" (Al bumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (G enreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK TrackGenreId ON "tracks" (GenreId)\n\nCREATE TABLE "genres"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NU LL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON "playlist track" (TrackId)\n \nCREATE INDEX IFK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "playlists"\r\n(\r\n TEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK TrackMediaType Id ON "tracks" (MediaTypeId)\n\nCREATE TABLE "playlist track"\r\n(\r\n PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL.\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n REIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI ON\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n le NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "ar tists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\cREATE INDEX IFK AlbumArtistId ON "albums" (ArtistId)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Respons e Guidelines \n1. If the provided context is sufficient, please generate a valid SQL guery without any expl anations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a sp ecific string in a particular column, please generate an intermediate SQL query to find the distinct string s 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. I f 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'}, {'r ole': 'assistant', 'content': 'SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "track s" t ON q.GenreId = t.GenreId\nGROUP BY q.Name\nORDER BY TotalTracks DESC\nLIMIT 5'}, {'role': 'user', 'con List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'conten t': 'SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" q\nJOIN "tracks" t ON q.GenreId = t.Genr eId\nGROUP BY q.Name'}, {'role': 'user', 'content': '\n There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a. ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nORDER BY To talTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': '\n There are 3 tables: artists, albums and tra cks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'content': 'S ELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.Artist Id\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'ro le': 'user', 'content': ' \n Identify artists who have albums with tracks appearing in multiple genre s:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artists" a\nJO IN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM "genres" q2

 $\n$  JOIN "tracks" t2 ON q2.GenreId = t2.GenreId\n GROUP BY g2.GenreId\n HAVING COUNT(g2.GenreId) > 1\n)\n GROUP BY a.ArtistId, a.Name'}, {'role': 'user', 'content': ' \n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name A S ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELE CT g2.GenreId\n FROM "genres" g2\n JOIN "tracks" t2 ON g2.GenreId = t2.GenreId\n GROUP BY g2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'role': 'user', 'conte Get all playlists containing at least 10 tracks and the total duration of those tracks:\n'}, {'role': 'assistant', 'content': 'SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS Tota lDuration\nFROM "playlist track" pt\nJOIN "playlists" p ON pt.PlaylistId = p.PlaylistId\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10'}, {'role': 'user', Get all playlists containing at least 10 tracks and the total duration of those track s:\n'}, {'role': 'assistant', 'content': 'SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDurationInMillis econds\nFROM "playlists" p\nJOIN "playlist track" pt ON p.PlaylistId = pt.PlaylistId\nJOIN "tracks" t ON p t.TrackId = t.TrackId\nGROUP BY p.PlaylistId\nHAVING COUNT(pt.PlaylistId) >= 10\nORDER BY TotalDurationInMi lliseconds DESC'}, {'role': 'user', 'content': ' \n Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT Name, UnitPrice\nFROM "tracks"\nORDER BY UnitPrice DE SC\nLIMIT 5'}, {'role': 'user', 'content': '\n Find the top 5 most expensive tracks (based on unit pri ce):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER B Y t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n List all genres and the number of track s in each genre:\n'}] Ollama parameters: model=aya:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery 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 \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREM ENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n Bvtes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres \" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK Trac kGenreId ON \"tracks\" (GenreId)\n\nCREATE TABLE \"genres\"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINC REMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON \"playlist trac k\" (TrackId)\n\nCREATE INDEX IFK TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"playlists\"\r\n(\r Name NVARCHAR(120)\r\n)\n\nCREATE INDEX PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n IFK TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE TABLE \"playlist track\"\r\n(\r\n

INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY (Playl FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO istId. TrackId).\r\n ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOI NCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL.\r\n KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCR EATE INDEX IFK AlbumArtistId ON \"albums\" (ArtistId)\ $n\n\n==Additional$  Context  $\n\n$  the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SOL query without any explanations for the question. \n2. If the provided context is almost sufficien t but requires knowledge of a specific string in a particular column, please generate an intermediate SQL g uery to find the distinct strings in that column. Prepend the guery with a comment saying intermediate sql  $\n$ 3. If the provided context is insufficient, please explain why it can't be generated.  $\n$ 4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer e xactly 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.GenreId) AS TotalTracks \nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks DESC \nLIMIT 5"}, {"role": "user", "content": " \n List all genres and the number of tracks in each genr e:\n"}, {"role": "assistant", "content": "SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres\" q \nJOIN \"tracks\" t ON q.GenreId = t.GenreId\nGROUP BY q.Name"}, {"role": "user", "content": " \n are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and track s are linked by AlbumId.\n Can you find the top 10 most popular artists based on the number of tracks \n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalT racks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "conten t": " \n There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistI d, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFRO M \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumI d\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n rtists who have albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistI d\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM \"genres\" q2\n JOIN \"tracks\" t2 ON q2.GenreId = t 2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name"}, {"rol e": "user", "content": " \n Identify artists who have albums with tracks appearing in multiple genre s:\n\n\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\n JOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT g2.GenreId\n FROM \"genres \" g2\n JOIN \"tracks\" t2 ON g2.GenreId = t2.GenreId\n GROUP BY g2.GenreId\n HAVING COUNT(g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName"}, {"role": "user", "content": "\n lists containing at least 10 tracks and the total duration of those tracks:\n"}, {"role": "assistant", "con tent": "SELECT pt.PlaylistId. p.Name AS PlaylistName. SUM(t.Milliseconds) AS TotalDuration\nFROM \"playlist track\" pt\nJOIN \"plavlists\" p ON pt.PlavlistId = p.PlavlistId\nJOIN \"tracks\" t ON pt.TrackId = t.Trac kId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10"}, {"role": "user", "content": " \n

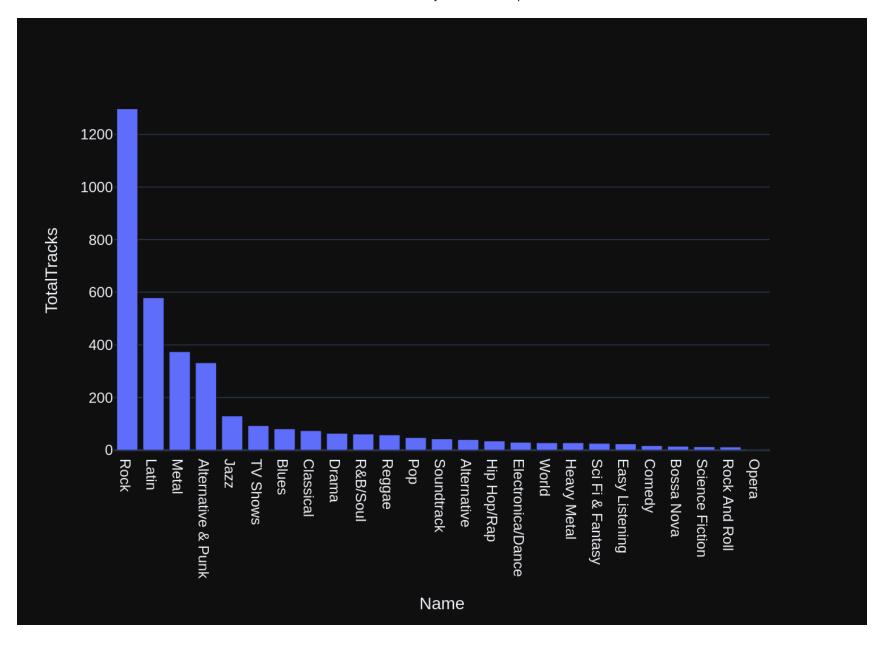
```
Get all playlists containing at least 10 tracks and the total duration of those tracks:\n"}, {"role": "assi
stant", "content": "SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDurationInMilliseconds\nFROM \"playlis
ts\" p\nJOIN \"playlist track\" pt ON p.PlaylistId = pt.PlaylistId\nJOIN \"tracks\" t ON pt.TrackId = t.Tra
ckId\nGROUP BY p.PlaylistId\nHAVING COUNT(pt.PlaylistId) >= 10\nORDER BY TotalDurationInMilliseconds DES
C"}, {"role": "user", "content": " \n Find the top 5 most expensive tracks (based on unit price):\n"},
{"role": "assistant", "content": "SELECT Name, UnitPrice\nFROM \"tracks\"\nORDER BY UnitPrice DESC\nLIMIT
5"}, {"role": "user", "content": " \n Find the top 5 most expensive tracks (based on unit price):\n"},
{"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.Unit
Price DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all genres and the number of tracks in eac
h genre:\n"}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T11:59:58.747106766Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId =
t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks DESC'}, 'done reason': 'stop', 'done': True, 'total durati
on': 88756658443, 'load duration': 1182865, 'prompt eval count': 1820, 'prompt eval duration': 77331191000,
'eval count': 52, 'eval duration': 10716801000}
SELECT g.Name, COUNT(t.GenreId) AS TotalTracks
FROM "genres" g
JOIN "tracks" t ON g.GenreId = t.GenreId
GROUP BY a.Name
ORDER BY TotalTracks DESC
SELECT g.Name, COUNT(t.GenreId) AS TotalTracks
FROM "genres" a
JOIN "tracks" t ON g.GenreId = t.GenreId
GROUP BY g.Name
ORDER BY TotalTracks DESC
                 Name TotalTracks
0
                              1297
                 Rock
1
                               579
                 Latin
2
                Metal
                                374
3
   Alternative & Punk
                                332
4
                  Jazz
                               130
5
             TV Shows
                                93
6
                 Blues
                                81
7
             Classical
                                74
8
                 Drama
                                 64
9
             R&B/Soul
                                 61
10
                                 58
                Reggae
11
                   Pop
                                 48
12
            Soundtrack
                                 43
13
           Alternative
                                 40
14
                                 35
           Hip Hop/Rap
```

```
15
     Electronica/Dance
                                 30
16
                 World
                                 28
17
                                 28
           Heavy Metal
18
      Sci Fi & Fantasy
                                 26
19
        Easy Listening
                                 24
20
                                 17
                Comedy
21
                                 15
            Bossa Nova
22
                                 13
       Science Fiction
23
         Rock And Roll
                                 12
24
                 Opera
                                  1
Ollama parameters:
model=ava: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 genr e:\n'\n\nThe DataFrame was produced using this query: SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks DESC\n\nTh e following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name object\nTotalTracks int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python pl otly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If t here 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': 'aya:latest', 'created\_at': '2024-06-14T12:00:29.165866209Z', 'message': {'role': 'assistant', 'c ontent': "```python\nimport plotly.express as px\n\n# Assuming your DataFrame is named 'df' and it has a co lumn named 'Name' for genre names and a column named 'TotalTracks' for the number of tracks in each genre\n fig = px.bar(df, x='Name', y='TotalTracks', color\_discrete\_sequence=px.colors.sequential['viridis'])\nfig.u pdate\_layout(title='Genres with Number of Tracks')\nfig.show()\n``"}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 30393344924, 'load\_duration': 690637, 'prompt\_eval\_count': 206, 'prompt\_eval\_duration': 8534022000, 'eval count': 106, 'eval duration': 21729391000}



```
Out[32]: ('SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.Genr
          eId\nGROUP BY q.Name\nORDER BY TotalTracks DESC',
                             Name TotalTracks
           0
                             Rock
                                           1297
           1
                            Latin
                                            579
           2
                            Metal
                                            374
               Alternative & Punk
                                            332
           4
                                            130
                             Jazz
           5
                         TV Shows
                                             93
           6
                            Blues
                                             81
           7
                        Classical
                                             74
           8
                            Drama
                                             64
           9
                         R&B/Soul
                                             61
           10
                                             58
                           Reggae
                              Pop
           11
                                             48
           12
                       Soundtrack
                                             43
           13
                      Alternative
                                             40
           14
                                             35
                      Hip Hop/Rap
           15
                Electronica/Dance
                                             30
           16
                            World
                                             28
           17
                      Heavy Metal
                                             28
           18
                 Sci Fi & Fantasy
                                             26
                   Easy Listening
           19
                                             24
           20
                                             17
                           Comedy
           21
                       Bossa Nova
                                             15
                  Science Fiction
           22
                                             13
           23
                    Rock And Roll
                                             12
           24
                            Opera
                                              1,
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'Name=%{x}<br>TotalTracks=%{y}<extra></extra>',
                         'legendgroup': '',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                         'name': '',
                         'offsetgroup': '',
                         'orientation': 'v',
                         'showlegend': False,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array(['Rock', 'Latin', 'Metal', 'Alternative & Punk', 'Jazz', 'TV Shows',
                                      'Blues', 'Classical', 'Drama', 'R&B/Soul', 'Reggae', 'Pop',
                                      'Soundtrack', 'Alternative', 'Hip Hop/Rap', 'Electronica/Dance',
```

```
'World', 'Heavy Metal', 'Sci Fi & Fantasy', 'Easy Listening', 'Comedy',
                                   'Bossa Nova', 'Science Fiction', 'Rock And Roll', 'Opera'], dtype=object),
                        'xaxis': 'x',
                        'y': array([1297, 579, 374, 332, 130, 93, 81, 74, 64, 61,
                                                                                                     48.
                                     43, 40, 35, 30, 28, 28, 26, 24, 17, 15, 13,
                                                                                                     12.
                                      1]),
                        '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': 'TotalTracks'}}}
          }))
         question = """
In [33]:
            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 1, updating n results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery 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 "tracks" (GenreId)\n\nCREATE TABLE "track TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n  $s"\r\n(\r\n$ AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL.\r\n GenreId INTEGER.\r\n Composer NVARCHAR(22 Bytes INTEGER,\r\n Milliseconds INTEGER NOT NULL.\r\n 0), r nUnitPrice NUMERIC(10,2) NOT NUL  $L,\r\n$ FOREIGN KEY (Albumid) REFERENCES "albums" (Albumid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT  $0N,\r\n$ FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON "playlist track" (TrackId)\n\nCREATE INDE X IFK TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE INDEX IFK TrackAlbumId ON "tracks" (AlbumId)\n\n CREATE TABLE "genres"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR  $(120)\r\n)\n\n\CREATE TABLE "albums"\r\n(\r\n$ AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist trac  $k"\r\n(\r\n$ PlavlistId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL.\r\n CONSTRAINT PK Playlis tTrack PRIMARY KEY (PlaylistId, TrackId).\r\n FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Playlist Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (Trac kId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK AlbumArtistId ON "albums" (Ar tistId)\n\nCREATE TABLE "playlists"\r\n(\r\n PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\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 guery to find the distinct st rings in that column. Prepend the guery 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). \n 5. If the question has been asked and answered before, please repeat the answer exactly as it was given bef ore. \n'}, {'role': 'user', 'content': ' \n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FR OM "genres" q2\n JOIN "tracks" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.Gen reId) > 1\n)\nGROUP BY a.ArtistId, a.Name'}, {'role': 'user', 'content': ' \n There are 3 tables: artist s, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by Albu Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assista nt', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a \nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Artis tId, a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': '\n artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked b v AlbumId.\n Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'a ssistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJ0IN "albums" al 0 N a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTrack s DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n Identify artists who have albums with tracks appe

aring in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistNam e\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.Genre Id\n FROM "genres" g2\n JOIN "tracks" t2 ON g2.GenreId = t2.GenreId\n GROUP BY g2.GenreId\n HAVING COUN T(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'role': 'user', 'content': '\n List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT q.Nam e, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.N ame'}, {'role': 'user', 'content': ' \n List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" q\nJOIN "tr acks" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'con tent': 'SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" q\nJOIN "tracks" t ON q.GenreId = t.GenreId\nGROUP BY q.Name\nORDER BY TotalTracks DESC'}, {'role': 'user', 'content': '\n lists containing at least 10 tracks and the total duration of those tracks:\n'}, {'role': 'assistant', 'con tent': 'SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDurationInMilliseconds\nFROM "playlists" p\nJOIN "playlist track" pt ON p.PlaylistId = pt.PlaylistId\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROUP BY p. PlaylistId\nHAVING COUNT(pt.PlaylistId) >= 10\nORDER BY TotalDurationInMilliseconds DESC'}, {'role': 'use Get all playlists containing at least 10 tracks and the total duration of those tr r', 'content': '\n acks:\n'}, {'role': 'assistant', 'content': 'SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Millisecon ds) AS TotalDuration\nFROM "playlist track" pt\nJOIN "playlists" p ON pt.PlaylistId = p.PlaylistId\nJOIN "t racks" t ON pt.TrackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10'}, {'rol e': 'user', 'content': ' \n Find all tracks with a name containing "What" (case-insensitive)\n'}, {'rol e': 'assistant', 'content': "SELECT \* FROM tracks WHERE lower(Name) LIKE '%what%'"}, {'role': 'user', 'cont ent': ' \n Get all genres that do not have any tracks associated with them:\n'}] Ollama parameters: model=aya:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery 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 \"tracks\" (GenreId)\n\nCREATE TABLE \"tr acks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Name NVARCHAR(200) NOT NUL L.\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER.\r\n

Milliseconds INTEGER NOT NULL,\r\n  $ARCHAR(220).\r\n$ Bytes INTEGER,\r\n UnitPrice NUMERIC(10.2) N OT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDA NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE N TE NO ACTION.\r\n O ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON \"playlist track\" (TrackId) \n\nCREATE INDEX IFK TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE INDEX IFK TrackAlbumId ON \"tra cks\" (AlbumId)\n\nCREATE TABLE \"genres\"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL Name NVARCHAR(120) $\r\n)\n\n$ CREATE TABLE  $\"albums\"\r\n(\r\n$ L.\r\n AlbumId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN K EY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCRE PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r ATE TABLE \"playlist track\"\r\n(\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlavlistId) REFER ENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Track Id) REFERENCES \"tracks\" (Trackid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IF K AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n PlavlistId INTEGER PRIMA RY KEY AUTOINCREMENT NOT NULL,\r\n Name  $NVARCHAR(120)\r\n)\n\n==Additional Context \n\nIn the SQLite$ database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please q enerate 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 intermedi ate SQL query to find the distinct strings in that column. Prepend the guery with a comment saying intermed iate sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Pleas e use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat th e answer exactly as it was given before. \n"}, {"role": "user", "content": " \n Identify artists who h ave albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.Ar tistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.A rtistId IN (\n SELECT g2.GenreId\n FROM \"genres\" g2\n JOIN \"tracks\" t2 ON g2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name"}, {"role": "user", "co There are 3 tables: artists, albums and tracks, where albums and artists are linked by Arti stId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based on t he number of tracks\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName, COUNT ks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"rol e": "user", "content": " \n There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular arti sts based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.Al bumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " Identify artists who have albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assist ant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.Arti stId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT g2.GenreId\n FROM \"genres\" g2\n J0IN \"tracks\" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName"}, {"role": "user", "content": " \n List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"qenres\" q\nJOIN \"tracks\" t ON q.GenreId = t.GenreId\nGROUP BY q.Name"}, {"role": "user", "content": " List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS TotalTracks $\nFROM \genres\ggnJOIN \genreId = t.GenreId = t.G$ UP BY g.Name\nORDER BY TotalTracks DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all genres an d the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreI d) AS TotalTracks\nFROM \"qenres\" q\nJOIN \"tracks\" t ON q.GenreId = t.GenreId\nGROUP BY q.Name\nORDER BY TotalTracks DESC"}, {"role": "user", "content": " \n Get all playlists containing at least 10 track s and the total duration of those tracks:\n"}, {"role": "assistant", "content": "SELECT p.PlaylistId, SUM

```
(t.Milliseconds) AS TotalDurationInMilliseconds\nFROM \"playlists\" p\nJOIN \"playlist track\" pt ON p.Play
listId = pt.PlaylistId\nJ0IN \"tracks\" t ON pt.TrackId = t.TrackId\nGROUP BY p.PlaylistId\nHAVING COUNT(p
t.PlaylistId) >= 10\nORDER BY TotalDurationInMilliseconds DESC"}, {"role": "user", "content": " \n
all playlists containing at least 10 tracks and the total duration of those tracks:\n"}, {"role": "assistan
t", "content": "SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM
\"playlist track\" pt\nJOIN \"playlists\" p ON pt.PlaylistId = p.PlaylistId\nJOIN \"tracks\" t ON pt.TrackI
d = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10"}, {"role": "user", "content
             Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant",
"content": "SELECT * FROM tracks WHERE lower(Name) LIKE '%what%'"}, {"role": "user", "content": " \n
t all genres that do not have any tracks associated with them:\n"}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T12:01:57.859757089Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT g.Name AS GenreName, t.Name AS TrackName\nFROM "genres" g\nLEFT JOIN "tracks" t ON g.Genr
eId = t.GenreId\nWHERE t.TrackId IS NULL\nORDER BY g.Name;'}, 'done reason': 'stop', 'done': True, 'total d
uration': 88528908631, 'load duration': 668771, 'prompt eval count': 1833, 'prompt eval duration': 76626764
000, 'eval count': 54, 'eval duration': 11202291000}
SELECT g.Name AS GenreName, t.Name AS TrackName
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
ORDER BY q.Name;
Output from LLM: SELECT q.Name AS GenreName, t.Name AS TrackName
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
ORDER BY q.Name;
Extracted SQL: SELECT q.Name AS GenreName, t.Name AS TrackName
FROM "genres" q
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
ORDER BY g.Name
SELECT g.Name AS GenreName, t.Name AS TrackName
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
ORDER BY g.Name
Empty DataFrame
Columns: [GenreName, TrackName]
Index: []
Ollama parameters:
model=aya:latest,
options={},
```

keep alive=None Prompt Content:

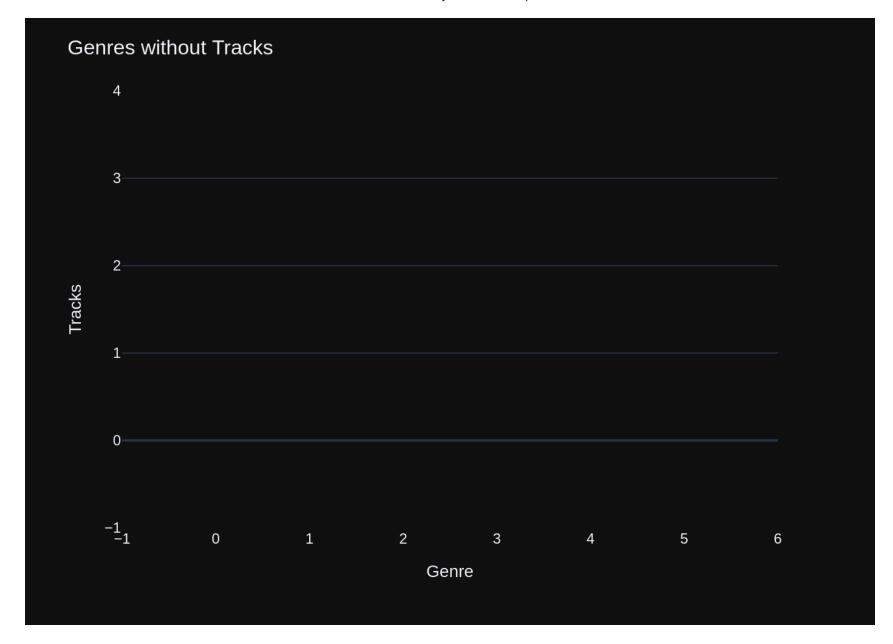
Ollama Response:

[{"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 wi th them:\n'\nThe DataFrame was produced using this query: SELECT q.Name AS GenreName, t.Name AS TrackName \nFROM \"genres\" g\nLEFT JOIN \"tracks\" t ON g.GenreId = t.GenreId\nWHERE t.TrackId IS NULL\nORDER BY g. Name\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes give object\ndtype: object"}, {"role": "user", "content": "Can you genera s:\n GenreName obiect\nTrackName te the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe c alled 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. D o not answer with any explanations -- just the code."}]

{'model': 'aya:latest', 'created at': '2024-06-14T12:02:26.09308949Z', 'message': {'role': 'assistant', 'co ntent': "```python\nimport plotly.express as px\n\n# Create a bar chart using plotly express\nfig = px.bar (df, x='GenreName', y='TrackName')\n\n# Update layout\nfig.update layout(\n title='Genres without Track

xaxis title='Genre',\n yaxis title='Tracks'\n)\n\n# Display the plot\nfiq.show()\n```"}, 'done s',\n reason': 'stop', 'done': True, 'total duration': 28230978356, 'load duration': 953095, 'prompt eval count':

208, 'prompt eval duration': 8968379000, 'eval count': 92, 'eval duration': 19125884000}



```
Out[33]: ('SELECT q.Name AS GenreName, t.Name AS TrackName\nFROM "genres" g\nLEFT JOIN "tracks" t ON q.GenreId =
          t.GenreId\nWHERE t.TrackId IS NULL\nORDER BY g.Name',
           Empty DataFrame
           Columns: [GenreName, TrackName]
           Index: [],
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'GenreName=%{x}<br/>br>TrackName=%{y}<extra></extra>',
                         'legendgroup': '',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                         'name': '',
                         'offsetgroup': '',
                         'orientation': 'v',
                         'showlegend': False,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array([], dtype=object),
                         'xaxis': 'x',
                         'y': array([], dtype=object),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'title': {'text': 'Genres without Tracks'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Genre'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Tracks'}}}
          }))
In [34]:
         question = """
             List all customers who have not placed any orders:
         0.00
         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 fo rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR EMENT NOT NULL.\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n BillingA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n **FOREIG** rv NVARCHAR(40),\r\n N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Firs tName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n ddress NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT SupportRepId INTEGER,\r\n NULL,\r\n FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineI InvoiceId INTEGER NOT NULL,\r\n d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER N UnitPrice NUMERIC(10,2) NOT NULL,\r\n OT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (I nvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n **FOREIGN** KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n LastName NVARCH TABLE "employees"\r\n(\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTE AR(20) NOT NULL,\r\n GER,\r\n BirthDate DATETIME,\r\n HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n City NVARCHAR (40), r nState NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone  $NVARCHAR(24), \r\n$ Fax NVARCHAR(24),  $\r\n$ Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist tra PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n CONSTRAINT PK Plavli stTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Playlis tId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (Tra ckid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n EGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE N O ACTION\r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "playl PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n ists"\r\n(\r\n Name NVARCHAR(120)\r\n)\n\n CREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Name NVARCHAR (200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL.\r\n GenreId INTEGER,\r\n Milliseconds INTEGER NOT NULL,\r\n Composer NVARCHAR(220),\r\n Bytes INTEGER.\r\n UnitPrice NUMER IC(10,2) NOT NULL,\r\n FOREIGN KEY (Albumid) REFERENCES "albums" (Albumid) \r\n\t\tON DELETE NO ACTION FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DEL ON UPDATE NO ACTION.\r\n ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\n \n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If th e provided context is sufficient, please generate a valid SQL query without any explanations for the questi on. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a part icular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepe

nd the query with a comment saying intermediate sql \n3. If the provided context is insufficient, please ex plain 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', Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoic es" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums D ESC\nLIMIT 1\n-----'}, {'role': 'user', 'content': ' \n Find the top 5 customers w ho spent the most money overall, \n \n Hint: order total can be found on invoices table, calculation n using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.Customer Id, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJ0IN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n------'}, {'role': 'user', 'conten Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c \nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\n GROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': '\n Hint: alb um quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in t otal quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.T rackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'ro le': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'c ontent': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n------------'}, {'role': 'user', 'content': ' \n Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(InvoiceId) as NumOfInvoices\nFROM "invoices"\nGROUP BY Cu stomerId\nORDER BY NumOfInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Find the customer w ith the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS To talInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\n0 RDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Find the top 5 customers who s pent the most money overall, \n \n Hint: order total can be found on invoices table, calculation us ing invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(InvoiceId) AS Numbe rOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-----------'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money overa Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpen ding\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n List all customers who have not placed anv orders:\n'}] Ollama parameters:

model=aya: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 \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL.\r\n InvoiceDate DATETIME NOT NULL.\r\n Billin gAddress NVARCHAR(70).\r\n BillinaCou BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40).\r\n ntry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n FirstName NVARCHAR(40) NOT NULL,\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Phone NVARCHAR(24),\r\n PostalCode NVARCHAR(10),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) NULL,\r\n SupportRepId INTEGER,\r\n \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n ineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n TrackId INTEGE Quantity INTEGER NOT NULL,\r\n R NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KE Y (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n OREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n \nCREATE TABLE \"employees\"\r\n(\r\n Last Name NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n Rep BirthDate DATETIME,\r\n ortsTo INTEGER.\r\n HireDate DATETIME,\r\n Address NVARCHAR(70),\r\n Ci tv NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),  $\r\n$ Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsT o) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TAB PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n LE \"playlist track\"\r\n(\r\n NSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) RE FERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL.\r FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\t0 ArtistId INTEGER NOT NULL.\r\n N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (Supp PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL ortRepId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n L.\r\n TrackId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL,\r\n NOT NULL,\r\n GenreId INTEGER,\r\n Bytes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres \" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK Invo

iceCustomerId ON \"invoices\" (CustomerId)\n\n===Additional Context \n\nIn the SQLite database invoice me ans order\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 requ ires knowledge of a specific string in a particular column, please generate an intermediate SQL query to fi nd the distinct strings in that column. Prepend the query with a comment saying intermediate sgl \n3. If th e provided context is insufficient, please explain why it can't be generated. \n4. Please use the most rele vant 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 customer who bought the most albu ms in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT i.CustomerId, CO UNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceI d\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n------"}, {"role": "user", "content": " \n \n Find the top 5 customers who spent the most money overall, \n Hint: order t otal can be found on invoices table, calculation using invoice items detail table is unnecessary \n"}, {"ro le": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n-------"}, {"role": "user", "content": " \n Find the customer who bought the most albu ms in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, CO UNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJ OIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nL IMIT 1"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGRO UP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n quantity is found in invoice items. \n \n Find the top 5 customers who bought the most albums in tota l quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.Trac kId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n------"}, {"role": "user", "content": Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT CustomerI d, COUNT(InvoiceId) as NumOfInvoices\nFROM \"invoices\"\nGROUP BY CustomerId\nORDER BY NumOfInvoices DESC\n LIMIT 1"}, {"role": "user", "content": " \n Find the customer with the most invoices \n"}, {"role": "a ssistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most money overall, \n Hint: order total can be found on invoices table, calculation using invoice items detail table is un necessarv \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"c ustomers\" c\nJ0IN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpen t DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the customer with the most invoices \n"}, {"r ole": "assistant", "content": "SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGR OUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n------"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most money overall, \n \n otal can be found on invoices table, calculation using invoice items detail table is unnecessary \n"}, {"ro

```
le": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM \"customers\" c\nJOI
        N \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT
        5"}, {"role": "user", "content": " \n List all customers who have not placed any orders:\n"}]
        Ollama Response:
        {'model': 'aya:latest', 'created at': '2024-06-14T12:03:57.91348811Z', 'message': {'role': 'assistant', 'co
        ntent': 'SELECT CustomerId, FirstName, LastName\nFROM "customers"\nLEFT JOIN "invoices" ON "customers".Cust
        omerId = "invoices".CustomerId\nWHERE InvoiceId IS NULL'}, 'done reason': 'stop', 'done': True, 'total dura
        tion': 91730200472, 'load duration': 850451, 'prompt eval count': 1943, 'prompt eval duration': 8239876200
        0, 'eval count': 41, 'eval duration': 8454704000}
        SELECT CustomerId, FirstName, LastName
        FROM "customers"
        LEFT JOIN "invoices" ON "customers".CustomerId = "invoices".CustomerId
        WHERE InvoiceId IS NULL
        SELECT CustomerId, FirstName, LastName
        FROM "customers"
        LEFT JOIN "invoices" ON "customers".CustomerId = "invoices".CustomerId
        WHERE InvoiceId IS NULL
        Couldn't run sql: Execution failed on sql 'SELECT CustomerId, FirstName, LastName
        FROM "customers"
        LEFT JOIN "invoices" ON "customers".CustomerId = "invoices".CustomerId
        WHERE InvoiceId IS NULL': ambiguous column name: CustomerId
In [35]: question = """
             There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums
             Can you find the top 10 most popular artists based on the number of tracks
```

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

vn.ask(question=question)

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery 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 "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMEN T NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NU LL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n tes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" (Al bumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (G enreId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n stId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "artists"\r\n(\r\n ArtistId INTEGER PRIMARY KEY AUTOINCR EMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK AlbumArtistId ON "albums" (ArtistId)\n\n CREATE INDEX IFK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "playlists"\r\n(\r\n GER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name  $NVARCHAR(120)\r\n)\n\nCREATE TABLE "genres"\r\n(\r\n$ GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "playlis t track"\r\n(\r\n PlavlistId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL.\r\n CONSTRAINT PK P laylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Pl FOREIGN KEY (TrackId) REFERENCES "tracks" aylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK TrackGenreId ON "tracks"  $(GenreId)\n\nCREATE\ INDEX\ IFK\ PlaylistTrackTrackId\ ON\ "playlist\ track"\ (TrackId)\n\n===Additional\ Context$ \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is su fficient, please generate a valid SQL guery without any explanations for the guestion. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please gen erate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comme nt saying intermediate sql \n3. If the provided context is insufficient, please explain why it can\'t be ge nerated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered befor e, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': '\n are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and track s are linked by AlbumId.\n Can you find the top 10 most popular artists based on the number of tracks \n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalT racks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.Al bumId\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': '\n There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based on the number of trac ks\n'}, {'role': 'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a \nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name \nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': '\n Identify artists who have al bums with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistI d, a.Name AS ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId I N (\n SELECT q2.GenreId\n FROM "genres" q2\n JOIN "tracks" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2. GenreId\n HAVING COUNT(g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'role': 'us

er', 'content': ' \n Identify artists who have albums with tracks appearing in multiple genres:\n\n

\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artists" a\nJOIN "al bums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM "genres" q2\n JOI N "tracks" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP B Y a.ArtistId, a.Name'}, {'role': 'user', 'content': '\n Hint: album quantity is found in invoice item s, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoice s):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "cus tomers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.In voiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most al bums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, C OUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\n GROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n-----'}, {'role': 'user', 'c Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELE CT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceI d = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELE CT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.Artist Id'}, {'role': 'user', 'content': '\n Hint: album quantity is found in invoice items, \n \n the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assi stant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "i nvoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n List all genres and the number of tracks in each genre:\n'}, {'ro le': 'assistant', 'content': 'SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "track s" t ON q.GenreId = t.GenreId\nGROUP BY q.Name\nORDER BY TotalTracks DESC\nLIMIT 5'}, {'role': 'user', 'con tent': '\n There are 3 tables: artists, albums and tracks, where albums and artists are linked by Artis tId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based on th e number of tracks\n'\l Ollama parameters: model=aya: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 TrackId INTEGER PRIMARY KEY AUTOINCREM rmat instructions. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n AlbumId INTEGER,\r\n ENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres \" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES

\"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Title NVARCHAR(160) NOT NULL.\r \"\r\n(\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\t0 N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"artists\"\r\n(\r\n ArtistId INTEGER PRIMAR Y KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK AlbumArtistId ON \"albums PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n \n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120) $\r\n$ ) PlavlistId INTEGER NOT NULL.\r\n \nCREATE TABLE \"playlist track\"\r\n(\r\n TrackId INTEGER NOT NUL CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) R EFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (T rackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDE X IFK TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON \"playlist track\"  $(TrackId)_n\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelin$ es \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations f or the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific str ing 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 insufficie nt, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the guest ion has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"ro le": "user", "content": " \n There are 3 tables: artists, albums and tracks, where albums and artists ar e linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular ar tists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS Art istName, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistI d\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTracks DESC\nLIM IT 10"}, {"role": "user", "content": " \n There are 3 tables: artists, albums and tracks, where albums a Can you find the top 10 mo nd artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n st popular artists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUN T(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tra cks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n Identify artists who have albums with tracks appearing in multiple genres: $\n\n\$ {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"album s\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM \"genres\" q2\n JOI N \"tracks\" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName"}, {"role": "user", "content": " \n Identify artists who hav e albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.Arti stId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.Art istId IN (\n SELECT q2.GenreId\n FROM \"genres\"  $q2\n JOIN \$  tracks\" t2 ON q2.GenreId = t2.GenreId\n G ROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name"}, {"role": "user", "con Hint: album quantity is found in invoice items, \n tent": " \n \n Find the top 5 customers who bo ught the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId =

i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY Tota lAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "\n Hint: album quantity is found in invoice item Find the top 5 customers who bought the most albums in total quantity (across all invoice s):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"in voices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalA is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantit y (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS T otalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Custom erId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "conten Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bough t the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i. CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.Inv oiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "conte List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres\" q\nJOIN \"tracks\" t ON q.GenreId = t.Genr eId\nGROUP BY q.Name\nORDER BY TotalTracks DESC\nLIMIT 5"}, {"role": "user", "content": " \n There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are Can you find the top 10 most popular artists based on the number of tracks\n"}] linked by AlbumId.\n

Add of existing embedding ID: 7d92306a-d153-5752-97bf-7bf120cd496a-sql Insert of existing embedding ID: 7d92306a-d153-5752-97bf-7bf120cd496a-sql

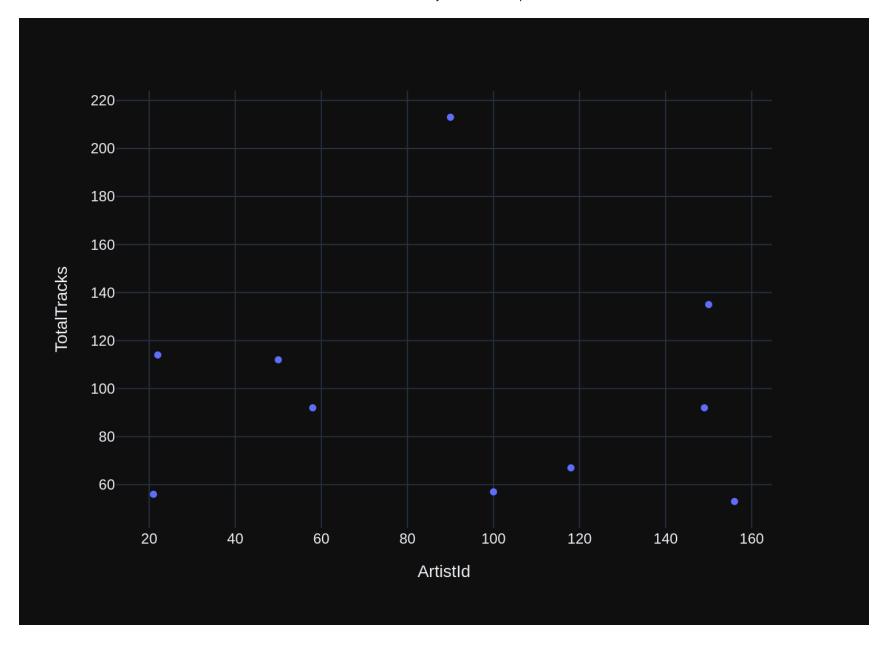
```
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T12:05:35.758673381Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN
"albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.
Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, 'done reason': 'stop', 'done': True, 'total duration': 9780276
1370, 'load duration': 780469, 'prompt eval count': 1909, 'prompt eval duration': 79639662000, 'eval coun
t': 84, 'eval duration': 17514399000}
SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks
FROM "artists" a
JOIN "albums" al ON a.ArtistId = al.ArtistId
JOIN "tracks" t ON al.AlbumId = t.AlbumId
GROUP BY a.ArtistId, a.Name
ORDER BY TotalTracks DESC
LIMIT 10
SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks
FROM "artists" a
JOIN "albums" al ON a.ArtistId = al.ArtistId
JOIN "tracks" t ON al.AlbumId = t.AlbumId
GROUP BY a.ArtistId, a.Name
ORDER BY TotalTracks DESC
LIMIT 10
                 ArtistName TotalTracks
   ArtistId
                 Iron Maiden
0
         90
                                      213
                                      135
1
        150
                          U2
               Led Zeppelin
2
         22
                                      114
3
         50
                   Metallica
                                      112
                Deep Purple
4
         58
                                       92
5
        149
                        Lost
                                       92
6
        118
                   Pearl Jam
                                       67
7
        100
               Lennv Kravitz
                                       57
        21 Various Artists
                                       56
        156
                 The Office
                                       53
Ollama parameters:
model=aya: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 There are 3 tables: artists, albums and tracks, where alb
ums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n
                                                                                     Can you find the top
10 most popular artists based on the number of tracks\n'\n\nThe DataFrame was produced using this query: SE
```

LECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\"

al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nO RDER BY TotalTracks DESC\nLIMIT 10\n\nThe following is information about the resulting pandas DataFrame 'd f': \nRunning df.dtypes gives:\n ArtistId int64\nArtistName object\nTotalTracks int64\ndtyp e: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of t he dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataf rame, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the cod e."}]

## Ollama Response:

{'model': 'aya:latest', 'created\_at': '2024-06-14T12:06:06.424434876Z', 'message': {'role': 'assistant', 'c ontent': "```python\nimport pandas as pd\nimport plotly.express as px\n\n# Assuming the 'df' DataFrame has columns ['ArtistId', 'ArtistName', 'TotalTracks']\nfig = px.bar(df, x='ArtistName', y='TotalTracks', color ='ArtistId', title='Top 10 Most Popular Artists')\nfig.update\_xaxes(rotation=45)\nfig.show()\n```"}, 'done\_ reason': 'stop', 'done': True, 'total\_duration': 30640339181, 'load\_duration': 41478552, 'prompt\_eval\_coun t': 264, 'prompt eval duration': 11145363000, 'eval count': 93, 'eval duration': 19354548000}



```
Out[35]: ('SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "album
          s" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name
          \nORDER BY TotalTracks DESC\nLIMIT 10'.
                             ArtistName TotalTracks
              ArtistId
           0
                    90
                            Iron Maiden
                                                 213
           1
                                     IJ2
                                                 135
                   150
           2
                    22
                           Led Zeppelin
                                                 114
           3
                    50
                              Metallica
                                                 112
                            Deep Purple
           4
                    58
                                                  92
           5
                                                  92
                   149
                                   Lost
           6
                                                  67
                   118
                              Pearl Jam
           7
                   100
                          Lenny Kravitz
                                                  57
           8
                    21 Various Artists
                                                  56
                   156
                             The Office
                                                  53,
           Figure({
               'data': [{'hovertemplate': 'ArtistId=%{x}<br>TotalTracks=%{y}<extra></extra>',
                         'legendgroup': '',
                         'marker': {'color': '#636efa', 'symbol': 'circle'},
                         'mode': 'markers',
                         'name': '',
                         'orientation': 'v',
                         'showlegend': False,
                         'type': 'scatter',
                         'x': array([ 90, 150, 22, 50, 58, 149, 118, 100, 21, 156]),
                         'xaxis': 'x'.
                         'y': array([213, 135, 114, 112, 92, 92, 67, 57, 56, 53]),
                         'yaxis': 'y'}],
               'layout': {'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'ArtistId'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalTracks'}}}
          }))
         question = """
In [36]:
              List all customers from Canada and their email addresses:
          0.00
         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 fo rmat instructions. \n===Tables \nCREATE INDEX IFK CustomerSupportRepId ON "customers" (SupportRepId)\n\nCRE CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n ATE TABLE "customers"\r\n(\r\n FirstName N VARCHAR(40) NOT NULL.\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address City NVARCHAR(40),\r\n  $NVARCHAR(70).\r\n$ State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Post alCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NUL L.\r\n SupportRepId INTEGER,\r\n FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n \t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER P RIMARY KEY AUTOINCREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NU LL,\r\n BillingAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(4 Total NUMERIC(10.2) 0), r nBillingCountry NVARCHAR(40).\r\n BillingPostalCode NVARCHAR(10).\r\n NOT NULL,\r\n FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION O N UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE TABLE "e EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n mplovees"\r\n(\r\n LastName NVARCHAR(20) NOT NULL.\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30).\r\n ReportsTo INTEGER.\r\n BirthDate DATETIME,\r\n HireDate DATETIME,\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(2 4), r nFax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employee s" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice items"\r\n(\r InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Ouantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI ON\r\n)\n\nCREATE TABLE sqlite sequence(name,seq)\n\nCREATE TABLE "playlist track"\r\n(\r\n TrackId INTEGER NOT NULL,\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY (Playli NTEGER NOT NULL,\r\n FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO AC stId, TrackId),\r\n TION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO A CTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE T AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n ABLE "albums"\r\n(\r\n Title NVARCHAR(160) N FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) OT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\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 sufficien t but requires knowledge of a specific string in a particular column, please generate an intermediate SQL g uery 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 th e 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 customers from Canada and their email addresses:\n'}, {'role': 'assistant', 'content': 'SELECT c.Email, c.Country\nFROM "customer s" c\nWHERE c.Country = \'Canada\''}, {'role': 'user', 'content': ' \n List all customers from Canada and their email addresses:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.Email, SUM(i.Tota

l) AS TotalSpent\nFROM Customers c\nJOIN Invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\n ORDER BY TotalSpent DESC\nLIMIT 1'}, {'role': 'user', 'content': '\n\ what are the top 5 countries that

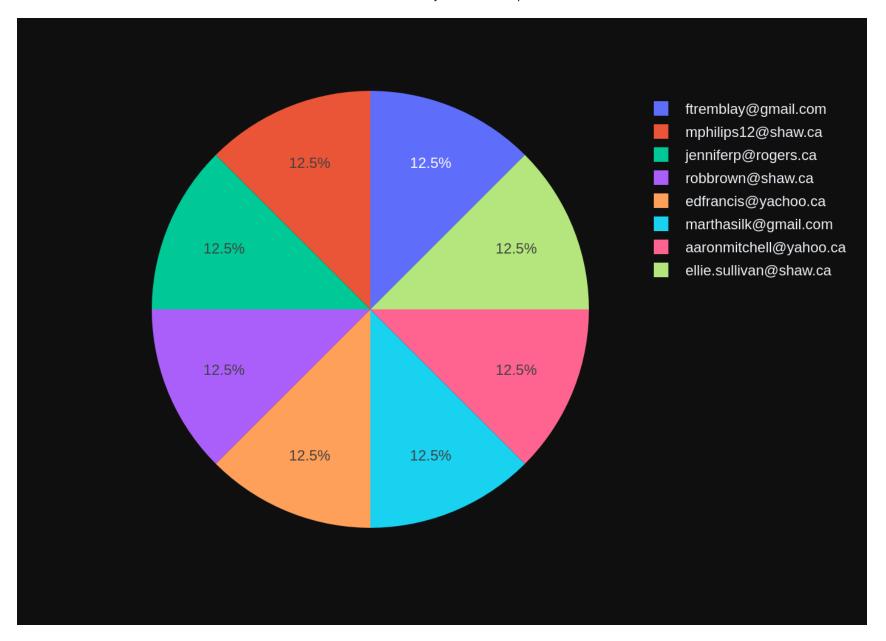
customers come from?\n'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(CustomerId) AS CustomerCo unt\nFROM "customers"\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'conten t': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT c.C ountry, COUNT(\*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nL IMIT 5'}, {'role': 'user', 'content': ' \n 顾客来自的前 5 个国家是哪些?\n'}, {'role': 'assistant', 'conten t': 'SELECT c.Country, COUNT(\*) AS CustomerCount\nFROM "customers" c\nGROUP BY c.Country\nORDER BY Customer Count DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n aus welchen fünf Ländern kommen die meisten Kun den?\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(\*) AS CustomerCount\nFROM "customers" c \nGROUP BY c.Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n e customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.Invo iceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.C ustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(InvoiceId) as NumOf Invoices\nFROM "invoices"\nGROUP BY CustomerId\nORDER BY NumOfInvoices DESC\nLIMIT 1'}, {'role': 'user', 'c Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT ontent': '\n CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER BY NumberOfIn voices DESC\nLIMIT 1\n-----'}, {'role': 'user', 'content': ' \n er of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT Customer.Country, COUNT(invoice.I nvoiceId) AS InvoiceCount\nFROM "customers" customer\nJOIN "invoices" invoice ON customer.CustomerId = invo ice.CustomerId\nGROUP BY Country'}, {'role': 'user', 'content': '\n List all customers from Canada an d their email addresses:\n'\l Ollama parameters: model=aya:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery 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 CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nC CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n REATE TABLE \"customers\"\r\n(\r\n FirstNa me NVARCHAR(40) NOT NULL.\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Addr City NVARCHAR(40),\r\n ess NVARCHAR(70),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INT EGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n CustomerId INTEGER NOT NULL.\r\n InvoiceDate DATETIME NOT NULL,\r\n BillingAddress NVARCHAR(70).\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR (40),\r\n BillingCountry NVARCHAR(40).\r\n BillingPostalCode NVARCHAR(10).\r\n Total NUMERIC(10.2) FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION NOT NULL,\r\n ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE TABLE

\"employees\"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(2 0) NOT NULL.\r\n FirstName NVARCHAR(20) NOT NULL.\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGE BirthDate DATETIME,\r\n  $R.\r\n$ HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n City NVARCHAR(4 State NVARCHAR(40),\r\n 0), r nCountry NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NV  $ARCHAR(24), \r\n$ Fax NVARCHAR(24).\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice i InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n tems\"\r\n(\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10.2) NOT NULL.\r\n Ouantity INTEGER FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON NOT NULL.\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION UPDATE NO ACTION,\r\n ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sglite sequence(name,seg)\n\nCREATE TABLE \"playlist track\"\r\n  $(\r\n$ PlavlistId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL.\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlavlistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackI d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON \"employee s\" (ReportsTo)\n\nCREATE TABLE \"albums\"\r\n(\r\n Albumid INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL Title NVARCHAR(160) NOT NULL.\r\n ArtistId INTEGER NOT NULL.\r\n FOREIGN KEY (ArtistId) R EFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Co ntext \n\nIn the SQLite database invoice means order\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 pro vided context is almost sufficient but requires knowledge of a specific string in a particular column, plea se 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 b e generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered b efore, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n List all customers from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SELECT c.E mail, c.Country\nFROM \"customers\" c\nWHERE c.Country = 'Canada'"}, {"role": "user", "content": " \n List all customers from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SELECT c.C ustomerId, c.Email, SUM(i.Total) AS TotalSpent\nFROM Customers c\nJOIN Invoices i ON c.CustomerId = i.Custo merId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 1"}, {"role": "user", "content": " \n t are the top 5 countries that customers come from?\n"}, {"role": "assistant", "content": "SELECT Country, COUNT(CustomerId) AS CustomerCount\nFROM \"customers\"\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMI T 5"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "ass istant", "content": "SELECT c.Country, COUNT(\*) AS TotalCustomers\nFROM \"customers\" c\nGROUP BY c.Country \nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "\n\u987e\u5ba2\u6765\u81ea\u7 684\u524d 5 \u4e2a\u56fd\u5bb6\u662f\u54ea\u4e9b\uff1f\n"}, {"role": "assistant", "content": "SELECT c.Coun try, COUNT(\*) AS CustomerCount\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY CustomerCount DESC\nLIMI T 5"}, {"role": "user", "content": " \n aus welchen f\u00fcnf L\u00e4ndern kommen die meisten Kunden? \n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(\*) AS CustomerCount\nFROM \"customers\" c\n GROUP BY c.Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": " \n customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.Invoic eId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY

```
c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n Find the custom
er with the most invoices \n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(InvoiceId) as Nu
mOfInvoices\nFROM \"invoices\"\nGROUP BY CustomerId\nORDER BY NumOfInvoices DESC\nLIMIT 1"}, {"role": "use
                        Find the customer with the most invoices \n"}, {"role": "assistant", "content": "S
r", "content": " \n
ELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER BY Numb
erOfInvoices DESC\nLIMIT 1\n------"}, {"role": "user", "content": " \n Find the tota
l number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT Customer.Country, COUNT(inv
oice.InvoiceId) AS InvoiceCount\nFROM \"customers\" customer\nJOIN \"invoices\" invoice ON customer.Custome
rId = invoice.CustomerId\nGROUP BY Country"}, {"role": "user", "content": " \n List all customers from
Canada and their email addresses:\n"}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T12:07:23.528312344Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT c.Email, c.Country\nFROM "customers" c\nWHERE c.Country = \'Canada\''}, 'done reason': 's
top', 'done': True, 'total duration': 76986025324, 'load duration': 667861, 'prompt eval count': 1655, 'pro
mpt eval duration': 71284784000, 'eval count': 25, 'eval duration': 5058967000}
SELECT c.Email, c.Country
FROM "customers" c
WHERE c.Country = 'Canada'
SELECT c.Email, c.Country
FROM "customers" c
WHERE c.Country = 'Canada'
                   Email Country
0
      ftremblay@gmail.com Canada
      mphilips12@shaw.ca Canada
1
2
      jenniferp@rogers.ca Canada
3
         robbrown@shaw.ca Canada
      edfrancis@yachoo.ca Canada
     marthasilk@gmail.com Canada
6 aaronmitchell@yahoo.ca Canada
7 ellie.sullivan@shaw.ca Canada
Ollama parameters:
model=aya: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 addresse
s:\n'\nThe DataFrame was produced using this query: SELECT c.Email, c.Country\nFROM \"customers\" c\nWHE
RE c.Country = 'Canada'\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunnin
                               object\nCountrv
                                                 object\ndtype: object"}, {"role": "user", "content": "Ca
q df.dtvpes gives:\n Email
n 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 Pyt
```

hon code. Do not answer with any explanations -- just the code."}]
Ollama Response:

{'model': 'aya:latest', 'created\_at': '2024-06-14T12:07:59.762002397Z', 'message': {'role': 'assistant', 'c ontent': "```python\nimport plotly.express as px\n\n# Create a bar chart using the 'df' dataframe and custo mize it to suit the request\nfig = px.bar(df, x='Email', y='Country', color='Country')\n\n# Update layout f or better visualization\nfig.update\_layout(\n title='Canadian Customers',\n xaxis\_title='Customer Email',\n yaxis\_title='Country',\n barmode='stack', # For stacked bar chart\n color\_discrete\_sequence =px.colors.sequential.Plotly\n)\n\n# Show the plot\nfig.show()\n``"}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 36208571147, 'load\_duration': 44219294, 'prompt\_eval\_count': 176, 'prompt\_eval\_duration': 7567635000, 'eval count': 132, 'eval duration': 28502004000}



```
Out[36]: ('SELECT c.Email, c.Country\nFROM "customers" c\nWHERE c.Country = \'Canada\'',
                               Email Country
                ftremblay@gmail.com Canada
          0
           1
                 mphilips12@shaw.ca Canada
           2
                jenniferp@rogers.ca Canada
           3
                    robbrown@shaw.ca Canada
                edfrancis@yachoo.ca Canada
               marthasilk@gmail.com Canada
             aaronmitchell@yahoo.ca Canada
          7 ellie.sullivan@shaw.ca Canada,
          Figure({
               'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
                         'hovertemplate': 'Email=%{label}<extra></extra>',
                         'labels': array(['ftremblay@qmail.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),
                         'legendgroup': '',
                         'name': '',
                         'showlegend': True,
                         'type': 'pie'}],
               'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'template': '...'}
          }))
         question = """
In [37]:
              Find the customer with the most invoices
         0.00
         vn.ask(question=question)
```

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

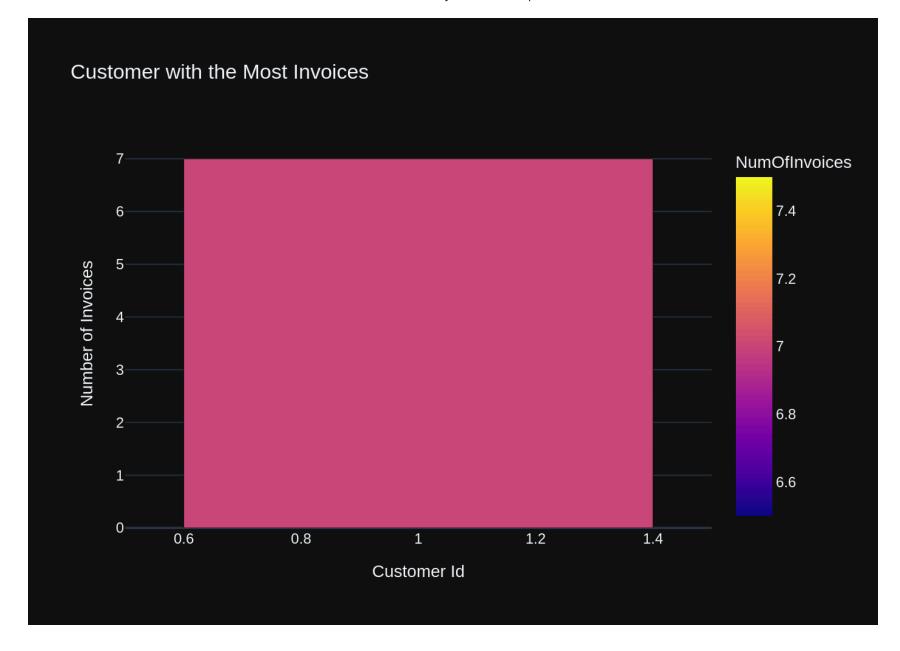
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR EMENT NOT NULL.\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n BillinaA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n **FOREIG** rv NVARCHAR(40),\r\n N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON "invoice items" (InvoiceId)\n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KE Y AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES Price NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERE NCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLin eTrackId ON "invoice items" (TrackId)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KE Y AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NUL  $L.\r\n$ Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHA  $R(40), \r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n  $VARCHAR(24).\r\n$ FOREIGN KEY (SupportR epId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IN DEX IFK CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "employees"\r\n(\r\n Id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVA LastName NVARCHAR(20) NOT NULL,\r\n ReportsTo INTEGER,\r\n RCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n BirthDate DATETIME.\r\n Address NVARCHAR(70),\r\n State NVARCHAR(40),\r\n HireDate DATETIME,\r\n City NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Country NVARCHAR(40),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24).\r FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DEL Email NVARCHAR(60),\r\n ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON "employees" (ReportsTo)\n\n CREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR GenreId INTEGER,\r\n (200) NOT NULL.\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL.\r\n Milliseconds INTEGER NOT NULL,\r\n Composer NVARCHAR(220),\r\n Bvtes INTEGER.\r\n UnitPrice NUMER FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\t0N DELETE NO ACTION IC(10,2) NOT NULL,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DEL ON UPDATE NO ACTION.\r\n ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL que ry without any explanations for the question. \n2. If the provided context is almost sufficient but require s 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 guery with a comment saying intermediate sql \n3. If the p rovided 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.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "cus tomers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices

DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Find the customer with the most invoices \n'}, {'rol e': 'assistant', 'content': 'SELECT CustomerId, COUNT(InvoiceId) as NumOfInvoices\nFROM "invoices"\nGROUP B Y CustomerId\nORDER BY NumOfInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': '\n Find the custom er with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(InvoiceId) AS Nu mberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-----------'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money ov Hint: order total can be found on invoices table, calculation using invoice items deta erall, \n il table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalS pent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n-----'}, {'role': 'user', 'content': ' \n customers who spent the most money overall, \n \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Custome rId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoi ces table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'conten t': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM "customers" c\nJOIN "invoices" i ON c.Custome rId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5'}, {'role': 'user', 'conten Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c \nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\n GROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': '\n customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'c ontent': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n------------'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "i nvoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quanti ty is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quan tity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) A S TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Custome rId\nORDER BY TotalAlbums DESC\nLIMIT 5\n------'}, {'role': 'user', 'content': '\n Find the customer with the most invoices \n'\] Ollama parameters: model=aya:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery 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 \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL.\r\n InvoiceDate DATETIME NOT NULL.\r\n Billin aAddress NVARCHAR(70).\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40).\r\n BillinaCou ntry NVARCHAR(40),\r\n Total NUMERIC(10,2) NOT NULL,\r\n BillingPostalCode NVARCHAR(10).\r\n F0RE IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK InvoiceLineInv oiceId ON \"invoice items\" (InvoiceId)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n InvoiceId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL.\r UnitPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Tr ackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE \"customers\"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCH AR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(2 4),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER.\r\n FOREI GN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"employees  $\"\r\n(\r\n$ EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NUL L.\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER,\r\n Bir thDate DATETIME.\r\n HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n St ate NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON \"employees\" (ReportsTo)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT AlbumId INTEGER,\r\n NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n MediaTypeId INTEGER NOT NUL Composer NVARCHAR(220),\r\n L.\r\n GenreId INTEGER,\r\n Milliseconds INTEGER NOT NULL,\r\n es INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (A lbumid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"me dia types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SOLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is su fficient, 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 gen erate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comme nt saying intermediate sql \n3. If the provided context is insufficient, please explain why it can't be gen erated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered befor e, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n d the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i. InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROU P BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n ustomer with the most invoices \n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(InvoiceId)

as NumOfInvoices\nFROM \"invoices\"\nGROUP BY CustomerId\nORDER BY NumOfInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n Find the customer with the most invoices \n"}, {"role": "assistant", "content t": "SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER B Y NumberOfInvoices DESC\nLIMIT 1\n------"}, {"role": "user", "content": " \n he top 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoic es table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "conten t": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.Custom {"role": "user", "content": " \n Find the top 5 customers who spent the most money overall, \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessa ry \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customer s\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC \nLIMIT 5"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most money overal Hint: order total can be found on invoices table, calculation using invoice items detail t able is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpend ing\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the customer who bought the mo st albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.Custome rId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Custom erId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n Find the customer who bought the most albums in tota l quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.Tra ckId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP B Y i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n------"}, {"role": "user", "conten Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bough t the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c. CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJ0IN \"invoice\_items\" ii 0N i.InvoiceId = ii.InvoiceId\nGR0UP BY c.CustomerId\nORDER BY Tota lAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice item Find the top 5 customers who bought the most albums in total quantity (across all invoice s):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"in voices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalA lbums DESC\nLIMIT 5\n-----"}, {"role": "user", "content": " \n Find the customer w ith the most invoices \n"}] Ollama Response: {'model': 'aya:latest', 'created at': '2024-06-14T12:09:39.473541408Z', 'message': {'role': 'assistant', 'c ontent': 'SELECT CustomerId, COUNT(InvoiceId) as NumOfInvoices\nFROM "invoices\\nGROUP BY CustomerId\\nORDE R BY NumOfInvoices DESC\nLIMIT 1'}, 'done reason': 'stop', 'done': True, 'total duration': 99603221108, 'lo ad duration': 836831, 'prompt eval count': 1997, 'prompt eval duration': 90547117000, 'eval count': 39, 'ev al duration': 8319781000} SELECT CustomerId, COUNT(InvoiceId) as NumOfInvoices FROM "invoices"

```
GROUP BY CustomerId
ORDER BY NumOfInvoices DESC
LIMIT 1
SELECT CustomerId, COUNT(InvoiceId) as NumOfInvoices
FROM "invoices"
GROUP BY CustomerId
ORDER BY NumOfInvoices DESC
LIMIT 1
   CustomerId NumOfInvoices
           1
Ollama parameters:
model=aya: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 Dat
aFrame was produced using this query: SELECT CustomerId, COUNT(InvoiceId) as NumOfInvoices\nFROM \"invoice
s\"\nGROUP BY CustomerId\nORDER BY NumOfInvoices DESC\nLIMIT 1\n\nThe following is information about the re
sulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId
                                                                            int64\nNumOfInvoices
                                                                                                    int64
\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the result
s 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."}l
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T12:10:13.133751625Z', 'message': {'role': 'assistant', 'c
ontent': "```python\nimport pandas as pd\nimport plotly.express as px\n\n# Assuming your DataFrame is named
'df' and it has two columns: 'CustomerId' and 'NumOfInvoices'\nfig = px.bar(df, x='CustomerId', y='NumOfInv
oices', color='NumOfInvoices',\n
                                             title='Customer with the Most Invoices')\nfig.update layout(x
axis title='Customer Id', yaxis title='Number of Invoices')\nfig.show()\n```"}, 'done reason': 'stop', 'don
e': True, 'total duration': 33633313758, 'load duration': 41227286, 'prompt eval count': 193, 'prompt eval
duration': 8891968000, 'eval count': 114, 'eval duration': 24645906000}
```



```
Out[37]: ('SELECT CustomerId, COUNT(InvoiceId) as NumOfInvoices\nFROM "invoices"\nGROUP BY CustomerId\nORDER BY Nu
         mOfInvoices DESC\nLIMIT 1',
            CustomerId NumOfInvoices
          0
                     1
                                   7,
          Figure({
              'data': [{'alignmentgroup': 'True',
                       'hovertemplate': 'CustomerId=%{x}<br>NumOfInvoices=%{marker.color}<extra></extra>',
                       'legendgroup': '',
                       'marker': {'color': array([7]), 'coloraxis': 'coloraxis', 'pattern': {'shape': ''}},
                       'name': '',
                       'offsetgroup': ''
                       'orientation': 'v',
                       'showlegend': False,
                       'textposition': 'auto',
                       'type': 'bar',
                       'x': array([1]),
                       'xaxis': 'x',
                       'y': array([7]),
                       'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                        'coloraxis': {'colorbar': {'title': {'text': 'NumOfInvoices'}},
                                     'colorscale': [[0.0, '#0d0887'], [0.111111111111111,
                                                   '#46039f'], [0.22222222222222,
                                                   '#bd3786'], [0.555555555555556,
                                                   '#ed7953'], [0.77777777777778,
                                                   '#fb9f3a'], [0.888888888888888,
                                                   '#fdca26'], [1.0, '#f0f921']]},
                        'legend': {'tracegroupgap': 0},
                        'template': '...',
                        'title': {'text': 'Customer with the Most Invoices'},
                        'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Customer Id'}},
                        'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Number of Invoices'}}}
         }))
In [ ]:
```

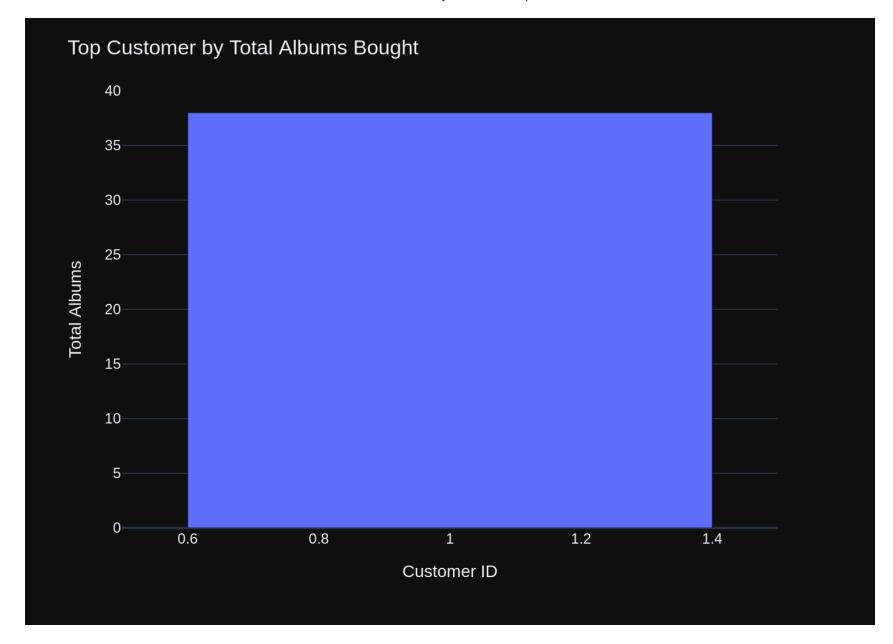
#### Advanced SQL questions

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery 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 "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMEN AlbumId INTEGER.\r\n T NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n MediaTypeId INTEGER NOT NU LL,\r\n GenreId INTEGER.\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n tes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" (Al bumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (G enreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice item InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n  $s"\r\n(\r\n$ InvoiceId INTEGER NOT NUL L, r nTrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Ouantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDAT FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDA E NO ACTION.\r\n TE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n Albumid INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL ArtistId INTEGER NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n FOREIGN KEY (ArtistId) R EFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK Alb umArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTO INCREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n ingAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillinaC ountry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10).\r\n Total NUMERIC(10.2) NOT NULL.\r\n REIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE INDEX IFK InvoiceLineTrackId ON "invoice items" (TrackId)\n\nCREATE INDEX IFK InvoiceLineIn voiceId ON "invoice items" (InvoiceId)\n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\n CREATE INDEX IFK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name  $NVARCHAR(120)\r\n)\n\n===Additional Context \n\nIn the SQ$ Lite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, ple ase generate a valid SQL query without any explanations for the question. \n2. If the provided context is a lmost sufficient but requires knowledge of a specific string in a particular column, please generate an int ermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying in termediate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n 4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please r epeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': '\n mer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'conten t': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.Cust omerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': '\n Hint: album quantity is found in invoi Find the top 5 customers who bought the most albums in total quantity (across all inv oices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = i i.InvoiceId\nGROUP BY c.CustomerId\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'}, {'role': 'assis tant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n-----------'}, {'role': 'user', 'content': '\n Hint: album quantity is found in invoice items, Find the top 5 customers who bought the most albums in total quantity (across all invoice s):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFRO M "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY Tota lAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice item Find the top 5 customers who bought the most albums in total quantity (across all invoice s):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "inv oices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbum s DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'r ole': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJ0 IN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMI T 5\n-----'}, {'role': 'user', 'content': '\n Find the top 5 customers who spent Hint: order total can be found on invoices table, calculation using i the most money overall, \n \n nvoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM (i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.C ustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n-----'}, {'role': 'user', 'content': ' Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerI d, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerI d\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n d the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on inv oices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'conte nt': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJ0IN "invoices" i ON c.CustomerI d = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' Find the top 5 customers who spent the most money overall, \n\\n\ Hint: order total can be fo und on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistan t', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM "customers" c\nJOIN "invoices" i 0 N c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5'}, {'role': 'use r', 'content': ' \n Find the customer who bought the most albums in total quantity (across all invoice s): \n'}] Ollama parameters: model=aya: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 \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREM ENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT Composer NVARCHAR(220),\r\n NULL,\r\n GenreId INTEGER,\r\n Milliseconds INTEGER NOT NULL.\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\"

(AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres \" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoic InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n e items\"\r\n(\r\n InvoiceId INTEGER N OT NULL.\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Ouantity INTEGE FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION R NOT NULL.\r\n ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\t0N DELETE NO ACTI ON ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMEN Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL.\r\n tistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IN DEX IFK AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER P RIMARY KEY AUTOINCREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NU LL.\r\n BillingAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(4 BillingPostalCode NVARCHAR(10),\r\n 0),\r\n BillingCountry NVARCHAR(40).\r\n Total NUMERIC(10.2) FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION NOT NULL,\r\n ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE IN DEX IFK InvoiceLineInvoiceId ON \"invoice items\" (InvoiceId)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"in voices\" (CustomerId)\n\nCREATE INDEX IFK TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"artists ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name  $NVARCHAR(120)\r\n)\n\n==$ Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the pro vided 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 particul ar column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend t he guery with a comment saying intermediate sql \n3. If the provided context is insufficient, please explai n why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been aske d and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "con Find the customer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"custo mers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = i i.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most al bums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, C OUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\n JOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\n LIMIT 5"}, {"role": "user", "content": " \n Find the customer who bought the most albums in total quan tity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Cu stomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n------"}, {"role": "user", "content": " Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the m ost albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.Custome rId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "

\n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the m ost albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.Custome rId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.I nvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most al bums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, C OUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.Invoice Find the top 5 customers who spent the most money overall, \n er total can be found on invoices table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJ0 IN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5 \n-----"}, {"role": "user", "content": " \n Find the customer with the most invoic es \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalI nvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most Hint: order total can be found on invoices table, calculation using invoice it monev overall, \n \n ems detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) A S TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Customer Id\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoices table, calculation u sing invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGR OUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5"}, {"role": "user", "content": "\n customer who bought the most albums in total quantity (across all invoices): \n"}] Ollama Response: {'model': 'aya:latest', 'created at': '2024-06-14T12:11:54.460004919Z', 'message': {'role': 'assistant', 'c ontent': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, 'done reason': 'stop', 'done': True, 'total duration': 101216919787, 'load duration': 712294, 'prompt eval count': 1927, 'prompt eval duration': 87957431000, 'eval count': 59, 'eval duration': 12656171000} SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums FROM "invoices" i JOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId GROUP BY i.CustomerId ORDER BY TotalAlbums DESC LIMIT 1 SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums FROM "invoices" i JOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId GROUP BY i.CustomerId ORDER BY TotalAlbums DESC

```
LIMIT 1
   CustomerId TotalAlbums
            1
                        38
Ollama parameters:
model=aya: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
uantity (across all invoices): \n'\n\nThe DataFrame was produced using this query: SELECT i.CustomerId, COU
NT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId
\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n\nThe following is information about the resul
ting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId
                                                                       int64\nTotalAlbums
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': 'aya:latest', 'created at': '2024-06-14T12:12:23.109286184Z', 'message': {'role': 'assistant', 'c
ontent': '```python\nimport plotly.express as px\n\n# Create a bar chart\nfig = px.bar(df, x="CustomerId",
y="TotalAlbums")\n\n# Add title and labels\nfig.update layout(\n title="Top Customer by Total Albums Bou
           xaxis title="Customer ID",\n yaxis title="Total Albums"\n)\n\n# Show the chart\nfiq.show()\n`
aht",\n
``'}, 'done reason': 'stop', 'done': True, 'total duration': 28621525296, 'load duration': 41795270, 'promp
t eval count': 224, 'prompt eval duration': 8958782000, 'eval count': 95, 'eval duration': 19566274000}
```



```
Out[38]: ('SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.I
          nvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1',
              CustomerId TotalAlbums
           0
                       1
                                   38,
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'CustomerId=%{x}<br>TotalAlbums=%{y}<extra></extra>',
                         'legendgroup': '',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                         'name': '',
                         'offsetgroup': '',
                         'orientation': 'v',
                         'showlegend': False,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array([1]),
                         'xaxis': 'x',
                         'y': array([38]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'title': {'text': 'Top Customer by Total Albums Bought'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Customer ID'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total Albums'}}}
           }))
         question = """
In [391:
             Hint: album quantity is found in invoice items,
             Find the top 5 customers who bought the most albums in total quantity (across all invoices):
         0.00
         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 guery 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 items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY InvoiceId INTEGER NOT NULL,\r\n
TrackId INTEGER NOT NULL,\r\n AUTOINCREMENT NOT NULL,\r\n ice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERE NCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n bumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL.\r\n GenreId INTEGER,\r\n Composer NVARCHAR(22 0), r nMilliseconds INTEGER NOT NULL,\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NUL FOREIGN KEY (Albumid) REFERENCES "albums" (Albumid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI L, r nFOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT  $0N,\r\n$ FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON  $ION, \r\n$ UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NU Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK Al bumArtistId ON "albums" (ArtistId)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON "invoice items" (InvoiceId) \n\nCREATE INDEX IFK InvoiceLineTrackId ON "invoice items" (TrackId)\n\nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDa te DATETIME NOT NULL,\r\n BillingAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillinaS tate NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n BillingCountry NVARCHAR(40),\r\n Total FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DEL NUMERIC(10,2) NOT NULL,\r\n ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\n CREATE INDEX IFK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name  $NVARCHAR(120)\r\n)\n\n===Additional Context \n\nIn the SQ$ Lite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, ple ase generate a valid SQL query without any explanations for the question. \n2. If the provided context is a lmost sufficient but requires knowledge of a specific string in a particular column, please generate an int ermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying in termediate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n 4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please r epeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n ntity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total q uantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackI d) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice item s" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n Find the top 5 cust omers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'conte nt': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'us Hint: album quantity is found in invoice items, \n \n Find the top 5 custome er', 'content': ' \n rs who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'conten

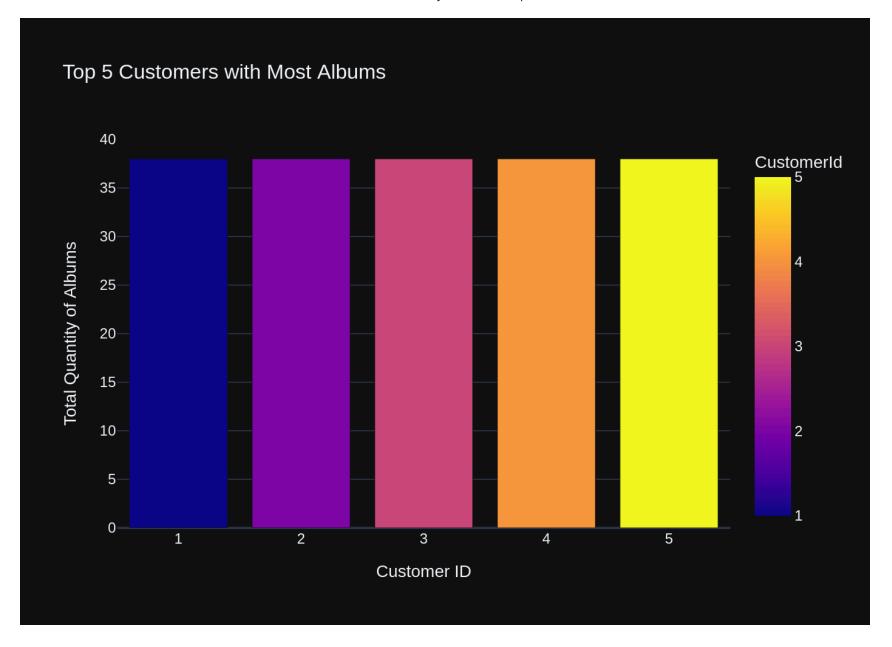
t': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n-----------'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n ind the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "in voice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer who bought the most albums in total quantity (acros s all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlb ums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.Inv oiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'conte nt': ' \n Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i \nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\n LIMIT 1'}, {'role': 'user', 'content': ' \n Find the customer who bought the most albums in total guan tity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Custom erId\nORDER BY TotalAlbums DESC\nLIMIT 1\n------'}, {'role': 'user', 'content': '\n Find the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'co ntent': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.Custom erId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n------'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money overall, \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessa ry \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customer s" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLI MIT 5'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money overall, Hint: order total can be found on invoices table, calculation using invoice items detail tabl e is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpending \nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY Tot alSpending DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice i \n Find the top 5 customers who bought the most albums in total quantity (across all invoice tems, \n s):\n'}l Ollama parameters: model=aya: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 items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY K InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n EY AUTOINCREMENT NOT NULL,\r\n tPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCE

S \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) RE FERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks \"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n GenreId INTEGER,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\r\n Composer NVARCHAR(22 Bytes INTEGER,\r\n  $0), \r\n$ Milliseconds INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NUL FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO AC  $L,\r\n$ TION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTI ACTION,\r\n ON ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMEN Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n tistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IN DEX IFK AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON \"invoice items \" (InvoiceId)\n\nCREATE INDEX IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE \"invo InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n CustomerId INTEGER NOT NUL BillingAddress NVARCHAR(70),\r\n L,\r\n InvoiceDate DATETIME NOT NULL,\r\n BillingCity NVARCHAR(4 0), r nBillingState NVARCHAR(40),\r\n BillingCountry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR FOREIGN KEY (CustomerId) REFERENCES \"customers\" (Custo (10), r nTotal NUMERIC(10,2) NOT NULL,\r\n merId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invo ices\" (CustomerId)\n\nCREATE INDEX IFK TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"artists\"\r ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\n===Addi  $\n(\r\n$ tional Context \n\nIn the SOLite database invoice means order\n\n===Response Guidelines \n1. If the provide d 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 co lumn, please generate an intermediate SQL guery to find the distinct strings in that column. Prepend the qu ery 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": Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought th e most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.Cust omerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Cus tomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbu ms DESC\nLIMIT 5"}, {"role": "user", "content": "\n Hint: album quantity is found in invoice items, \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"r ole": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices \" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"r ole": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\n JOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\n LIMIT 5\n-----"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across al l invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\n

FROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\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"}, {"role": "assistant", "content": "SELECT c.CustomerI d, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Customer Id\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DE SC\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 i.CustomerId, COUNT(ii.Track Id) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n-----------------"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most money overall, Hint: order total can be found on invoices table, calculation using invoice items detail table e is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nF ROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY To talSpent DESC\nLIMIT 5\n-----"}, {"role": "user", "content": " \n Find the top 5 c ustomers who spent the most money overall, \n \n Hint: order total can be found on invoices table. calculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Cus tomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": "\n Find the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "co ntent": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5"}, {"role": "use r", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customer s who bought the most albums in total quantity (across all invoices):\n"}] Ollama Response: {'model': 'aya:latest', 'created at': '2024-06-14T12:14:02.627686998Z', 'message': {'role': 'assistant', 'c ontent': 'SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId \nORDER BY TotalAlbums DESC\nLIMIT 5'}, 'done reason': 'stop', 'done': True, 'total duration': 99430760525, 'load duration': 1769703, 'prompt eval count': 1952, 'prompt eval duration': 82824362000, 'eval count': 77, 'eval duration': 15991516000} SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbums FROM "customers" c JOIN "invoices" i ON c.CustomerId = i.CustomerId JOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId GROUP BY c.CustomerId ORDER BY TotalAlbums DESC LIMIT 5 SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbums

```
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
JOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5
   CustomerId TotalAlbums
0
            1
                        38
            2
                        38
1
2
            3
                        38
3
            4
                        38
            5
                        38
Ollama parameters:
model=aya: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 Hint: album quantity is found in invoice items, \n
Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'\nThe Data
Frame was produced using this query: SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbums\nFROM \"customer
s\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.
InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n\nThe following is information about
the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId
                                                                                int64\nTotalAlbums
4\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the resul
ts of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in th
e dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just t
he code."}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T12:14:33.031287618Z', 'message': {'role': 'assistant', 'c
ontent': "```python\nimport plotly.express as px\n\n# Assuming 'CustomerId' and 'TotalAlbums' are columns i
n the 'df' dataframe\nfiq = px.bar(df, x='CustomerId', y='TotalAlbums', color='CustomerId',\n
itle='Top 5 Customers with Most Albums')\nfiq.update layout(xaxis title='Customer ID', yaxis title='Total Q
uantity of Albums')\n```"}, 'done reason': 'stop', 'done': True, 'total duration': 30375856025, 'load durat
ion': 42170593, 'prompt eval count': 256, 'prompt eval duration': 10663963000, 'eval count': 94, 'eval dura
```

tion': 19613313000}



```
Out[39]: ('SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.Custo
         merId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER
         BY TotalAlbums DESC\nLIMIT 5',
             CustomerId TotalAlbums
          0
                     1
                     2
          1
                                38
          2
                     3
                                38
          3
                     4
                                38
                     5
          4
                                38,
          Figure({
              'data': [{'alignmentgroup': 'True',
                       'hovertemplate': 'CustomerId=%{marker.color}<br>TotalAlbums=%{y}<extra></extra>',
                       'legendgroup': '',
                       'marker': {'color': array([1, 2, 3, 4, 5]), 'coloraxis': 'coloraxis', 'pattern': {'shape':
         ''}},
                       'name': '',
                       'offsetgroup': '',
                       'orientation': 'v',
                       'showlegend': False,
                       'textposition': 'auto',
                       'type': 'bar',
                       'x': array([1, 2, 3, 4, 5]),
                       'xaxis': 'x',
                       'y': array([38, 38, 38, 38, 38]),
                       'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                        'coloraxis': {'colorbar': {'title': {'text': 'CustomerId'}},
                                     'colorscale': [[0.0, '#0d0887'], [0.1111111111111111,
                                                   '#46039f'], [0.22222222222222,
                                                   '#bd3786'], [0.555555555555556,
                                                   '#ed7953'], [0.7777777777778,
                                                   '#fb9f3a'], [0.88888888888888888,
                                                   '#fdca26'], [1.0, '#f0f921']]},
                        'legend': {'tracegroupgap': 0},
                        'template': '...',
                        'title': {'text': 'Top 5 Customers with Most Albums'},
                        'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Customer ID'}},
                        'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total Quantity of Album
```

```
s'}}}
          }))
         SELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums
         FROM Customers c
         JOIN invoices i ON c.CustomerId = i.CustomerId
         JOIN invoice items il ON i.InvoiceId = il.InvoiceId
         GROUP BY c.CustomerId
         ORDER BY TotalAlbums DESC
         LIMIT 5
         question = """
In [40]:
              Find the top 5 customers who spent the most money overall,
              Hint: order total can be found on invoices table, calculation using invoice items detail table is unne
         0.00
         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 guery 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 "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR InvoiceDate DATETIME NOT NULL.\r\n EMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n BillinaA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n rv NVARCHAR(40),\r\n **FOREIG** N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n UnitPrice NUMERIC(10.2) NOT NULL.\r FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\t Quantity INTEGER NOT NULL,\r\n ON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t \t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON "invoice items" (InvoiceId)\n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK InvoiceLin eTrackId ON "invoice items" (TrackId)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KE Y AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NUL L.\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70).\r\n City NVARCHAR(40).\r\n State NVARCHA  $R(40), \r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER.\r\n  $VARCHAR(24).\r\n$ FOREIGN KEY (SupportR epId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TA BLE "employees"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR (20) NOT NULL.\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGE Address NVARCHAR(70),\r\n City NVARCHAR(4  $R.\r\n$ BirthDate DATETIME.\r\n HireDate DATETIME.\r\n State NVARCHAR(40),\r\n Phone NV  $0), \r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n  $ARCHAR(24).\r\n$ Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n (\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n bumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(22 Bytes INTEGER,\r\n  $0), \r\n$ Milliseconds INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NUL FOREIGN KEY (Albumid) REFERENCES "albums" (Albumid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI  $L,\r\n$ FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT  $0N,\r\n$ FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON  $ION, \r\n$ UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist track"\r\n(\r\n PlaylistId INTEGER NOT NULL.\r\n Tra CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n ckId INTEGER NOT NULL.\r\n FOREI GN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r \n  $\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON "employees" (ReportsTo)\n\n\n===Additional Context <math>\n\n$  the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, p lease generate a valid SQL guery 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 in termediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying i ntermediate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n 4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please r

epeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoices table. calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Custome rId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n------'}, {'role': 'use r', 'content': ' \n Find the top 5 customers who spent the most money overall, \n \n Hint: ord er total can be found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money overall, \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessa ry \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM "custom ers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DES C\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album qua ntity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total q uantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackI d) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice item s" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n Find the top 5 cust omers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'conte nt': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n----------------'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n F ind the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJ0 IN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMI T 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n d the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'as sistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoi ce items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'us er', 'content': ' \n Find the customer who bought the most albums in total quantity (across all invoic es): \n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "i nvoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlb ums DESC\nLIMIT 1\n-----'}, {'role': 'user', 'content': ' \n Find the top 5 custom ers who spent the most money overall, \n \n Hint: order total can be found on invoices table, calcu

lation using invoice\_items detail table is unnecessary \n'}]
Ollama parameters:
model=aya:latest,
options={},
keep\_alive=None
Prompt Content:

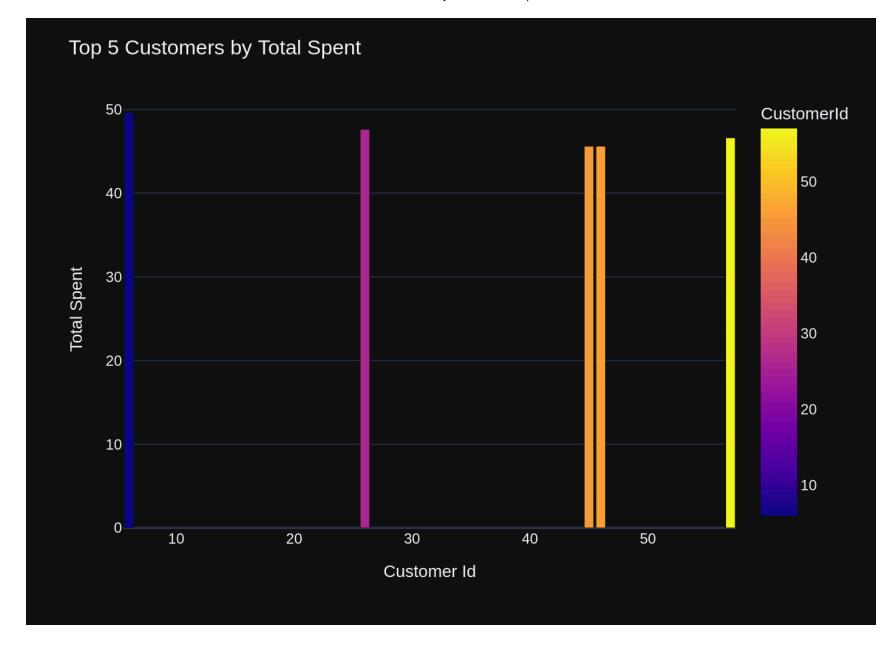
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n gAddress NVARCHAR(70),\r\n BillingState NVARCHAR(40),\r\n BillingCity NVARCHAR(40),\r\n BillinaCou ntry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n F0RE IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL InvoiceId INTEGER NOT NULL,\r\n
TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NO T NULL,\r\n Ouantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceI d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (Tra ckId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON \"in voice items\" (InvoiceId)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDE X IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE \"customers\"\r\n(\r\n d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVAR CHAR(20) NOT NULL, $\r\n$ Address NVARCHAR(70),\r\n Company NVARCHAR(80),\r\n City NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Phone NVARCHAR(2 Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n GN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE TABLE \"employees\"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER,\r\n BirthDate DATETIME,\r\n HireDate DATETIME,\r\n Address NVARCHAR(70),\r\n PostalCode NVARCHAR(1 City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Fax NVARCHAR(24),\r\n 0), r nPhone NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (Repo rtsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Milliseconds INTEGER NOT NULL,\r\n oser NVARCHAR(220),\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(1 0,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\t0N DELETE NO ACTION FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON D ON UPDATE NO ACTION.\r\n ELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist track\"\r\n(\r\n PlavlistId INTEGER CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, Tr NOT NULL.\r\n TrackId INTEGER NOT NULL,\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION O ackId),\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTIO N UPDATE NO ACTION.\r\n 

ional Context \n\nIn the SOLite database invoice means order\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 colum n, 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 guestion has been asked and ans wered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " Find the top 5 customers who spent the most money overall, \n \n Hint: order total can be fo und on invoices table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistan t", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n------------"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most money overal Hint: order total can be found on invoices table, calculation using invoice items detail t able is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent \nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the top 5 customers who spent the mo st money overall, \n \n Hint: order total can be found on invoices table, calculation using invoice \_items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Tota l) AS TotalSpending\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.C ustomerId\nORDER BY TotalSpending DESC\nLIMIT 5"}, {"role": "user", "content": " \n Hint: album quantit y is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quant ity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbums\nFROM \"customers\" c\nJ0IN \"invoices\" i ON c.CustomerId = i.CustomerId\nJ0IN \"invoice it ems\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"rol e": "user", "content": " \n Hint: album quantity is found in invoice items, \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "c ontent": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.Custome rId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "\n Hint: album quantity is fou Find the top 5 customers who bought the most albums in total quantity (acr nd in invoice items, \n \n oss all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAl bums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\n ORDER BY TotalAlbums DESC\nLIMIT 5\n------"}, {"role": "user", "content": " \n album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums i n total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(i i.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.Invoice Id\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums i n total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(i i.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGR OUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n ustomer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.Invoice

```
Id) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.
CustomerId\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 i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON
i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n-------------
----"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most money overall,
             Hint: order total can be found on invoices table, calculation using invoice items detail tabl
e is unnecessary \n"}]
Ollama Response:
{'model': 'aya:latest', 'created_at': '2024-06-14T12:16:17.152796179Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT c.CustomerId, SUM(ii.UnitPrice * ii.Quantity) AS TotalSpent\nFROM "customers" c\nJOIN "in
voices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY
c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, 'done reason': 'stop', 'done': True, 'total duration': 1
04006532246, 'load duration': 657165, 'prompt eval count': 1990, 'prompt eval duration': 85764614000, 'eval
count': 82, 'eval duration': 17405277000}
SELECT c.CustomerId, SUM(ii.UnitPrice * ii.Quantity) AS TotalSpent
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
JOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 5
SELECT c.CustomerId, SUM(ii.UnitPrice * ii.Quantity) AS TotalSpent
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
JOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 5
   CustomerId TotalSpent
0
           6
                   49.62
                   47.62
1
           26
2
           57
                   46.62
3
           45
                   45.62
           46
                   45.62
Ollama parameters:
model=aya: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 customers who spent the most money overa
```

ll, \n \n Hint: order total can be found on invoices table, calculation using invoice\_items detail table is unnecessary \n'\n\nThe DataFrame was produced using this query: SELECT c.CustomerId, SUM(ii.UnitPr ice \* ii.Quantity) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId \nJOIN \"invoice\_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC \nLIMIT 5\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gi ves:\n CustomerId int64\nTotalSpent float64\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 dataf rame called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python c ode. Do not answer with any explanations -- just the code."}]

{'model': 'aya:latest', 'created\_at': '2024-06-14T12:16:51.100336731Z', 'message': {'role': 'assistant', 'c ontent': "```python\nimport plotly.express as px\n\n# Create a bar chart\nfig = px.bar(df, x='CustomerId', y='TotalSpent', color='CustomerId')\n\n# Update layout\nfig.update\_layout(\n title='Top 5 Customers by T otal Spent',\n xaxis\_title='Customer Id',\n yaxis\_title='Total Spent',\n showlegend=True\n)\n\n# R ender the figure\nfig.show()\n```"}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 33919966171, 'l oad\_duration': 41270151, 'prompt\_eval\_count': 268, 'prompt\_eval\_duration': 11436326000, 'eval\_count': 106, 'eval duration': 22397497000}



```
Out[40]: ('SELECT c.CustomerId, SUM(ii.UnitPrice * ii.Quantity) AS TotalSpent\nFROM "customers" c\nJOIN "invoices"
         i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.Custom
         erId\nORDER BY TotalSpent DESC\nLIMIT 5',
             CustomerId TotalSpent
          0
                     6
                             49.62
                    26
          1
                             47.62
          2
                    57
                             46.62
          3
                    45
                             45.62
                    46
                             45.62,
          Figure({
              'data': [{'alignmentgroup': 'True',
                       'hovertemplate': 'CustomerId=%{marker.color}<br>TotalSpent=%{y}<extra></extra>',
                       'legendgroup': '',
                       'marker': {'color': array([ 6, 26, 57, 45, 46]), 'coloraxis': 'coloraxis', 'pattern': {'sha
         pe': ''}},
                       'name': '',
                       'offsetgroup': '',
                       'orientation': 'v',
                       'showlegend': False.
                       'textposition': 'auto',
                       'type': 'bar',
                       'x': array([ 6, 26, 57, 45, 46]),
                       'xaxis': 'x',
                       'y': array([49.62, 47.62, 46.62, 45.62, 45.62]),
                       'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                        'coloraxis': {'colorbar': {'title': {'text': 'CustomerId'}},
                                     'colorscale': [[0.0, '#0d0887'], [0.1111111111111111,
                                                   '#46039f'], [0.22222222222222.
                                                    '#bd3786'], [0.55555555555556,
                                                    '#ed7953'], [0.7777777777778,
                                                    '#fb9f3a'], [0.88888888888888888,
                                                   '#fdca26'], [1.0, '#f0f921']]},
                        'legend': {'tracegroupgap': 0},
                        'margin': {'t': 60},
                        'showlegend': True,
                        'template': '...',
                        'title': {'text': 'Top 5 Customers by Total Spent'},
                        'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Customer Id'}},
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery 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 PlaylistTrackTrackId ON "playlist track" (TrackId)\n\nCRE ATE TABLE "plavlists"\r\n(\r\n PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCH  $AR(120)\r\n)\n\nCREATE TABLE "playlist track"\r\n(\r\n$ PlaylistId INTEGER NOT NULL.\r\n TrackId INTE CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n GER NOT NULL,\r\n FOREIGN KEY (P laylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n F0RE IGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCRE ATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(20 0) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL,\r\n Bytes INTEGER,\r\n UnitPrice NUMER IC(10,2) NOT NULL,\r\n FOREIGN KEY (Albumid) REFERENCES "albums" (Albumid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DEL ON UPDATE NO ACTION,\r\n ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK TrackGenreId ON "tracks" (GenreId)\n\nCREATE IND EX IFK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDEX IFK TrackMediaTypeId ON "tracks" (MediaTypeId)\n \nCREATE INDEX IFK AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "albums"\r\n(\r\n R PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NU FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "genres"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\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). \n 5. If the question has been asked and answered before, please repeat the answer exactly as it was given bef ore. \n'}, {'role': 'user', 'content': ' \n Get all playlists containing at least 10 tracks and the to tal duration of those tracks:\n'}, {'role': 'assistant', 'content': 'SELECT p.PlaylistId, SUM(t.Millisecond s) AS TotalDurationInMilliseconds\nFROM "playlists" p\nJOIN "playlist track" pt ON p.PlaylistId = pt.Playli stId\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROUP BY p.PlaylistId\nHAVING COUNT(pt.PlaylistId) >= 10\n ORDER BY TotalDurationInMilliseconds DESC'}, {'role': 'user', 'content': ' \n Get all playlists contai ning at least 10 tracks and the total duration of those tracks:\n'}, {'role': 'assistant', 'content': 'SELE CT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM "playlist track" pt\nJ OIN "playlists" p ON pt.PlaylistId = p.PlaylistId\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROUP BY pt.P laylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10'}, {'role': 'user', 'content': ' \n List all genres a nd the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreI d) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.Name'}, {'role': 'user', 'content': ' \n List all genres and the number of tracks in each genre:\n'}, {'role': 'assistan t', 'content': 'SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" q\nJOIN "tracks" t ON q.Genre Id = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Nam e, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" q\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY q.Name\nORDER BY TotalTracks DESC'}, {'role': 'user', 'content': ' \n There are 3 tables: artists, album s and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'cont ent': 'SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "al bums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Nam e\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumI Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistan t', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.Art istId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC \nLIMIT 10'}, {'role': 'user', 'content': ' \n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFRO M "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM "genres" g2\n JOIN "tracks" t2 ON g2.GenreId = t2.GenreId\n GROUP BY g2.GenreId\n HAVING COUNT(g2.G enreId) > 1\n)\nGROUP BY a.ArtistId, a.Name'}, {'role': 'user', 'content': ' \n Identify artists who h ave albums with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.Ar tistId, a.Name AS ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.Artis tId IN (\n SELECT g2.GenreId\n FROM "genres" g2\n J0IN "tracks" t2 ON g2.GenreId = t2.GenreId\n GROUP B Y q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'rol e': 'user', 'content': ' \n Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nF ROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY To talAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': '\n Get all playlists containing at least 10 t racks and the total duration of those tracks:\n'}]

Ollama parameters:

model=aya:latest,

options={},

keep\_alive=None

Prompt Content:

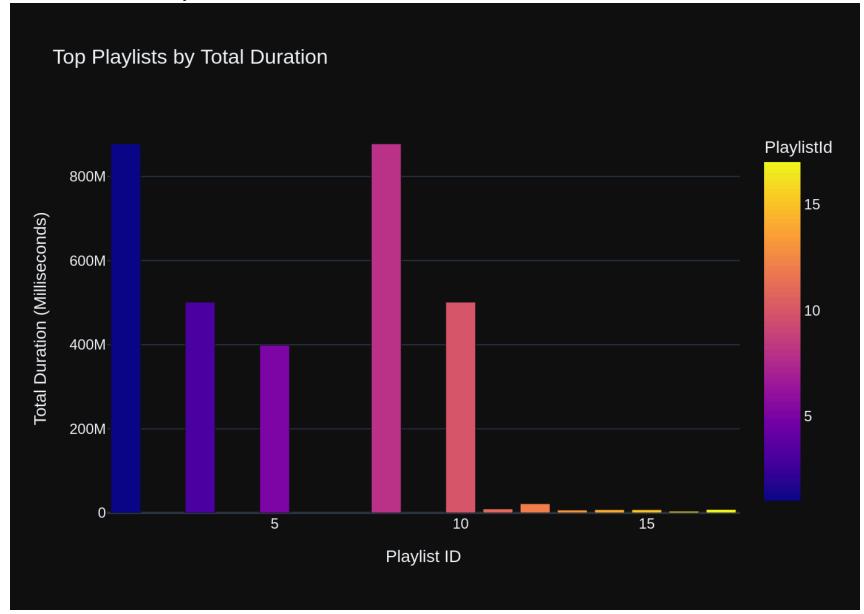
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE INDEX IFK PlaylistTrackTrackId ON \"playlist track\" (TrackId)\n\nC PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n REATE TABLE \"playlists\"\r\n(\r\n Name NV ARCHAR(120)\r\n)\n\nCREATE TABLE \"playlist track\"\r\n(\r\n PlaylistId INTEGER NOT NULL,\r\n TrackI CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n d INTEGER NOT NULL,\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Name NVA RCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGE Composer NVARCHAR(220),\r\n  $R.\r\n$ Milliseconds INTEGER NOT NULL,\r\n Bytes INTEGER,\r\n UnitP rice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\t0N DELET

FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\t0N DE E NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeI LETE NO ACTION ON UPDATE NO ACTION.\r\n d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK TrackGenreId ON \"tracks\" (Gen reId)\n\nCREATE INDEX IFK TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDEX IFK TrackMediaTypeId ON \"tr acks\" (MediaTypeId)\n\nCREATE INDEX IFK AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"albums \"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r ArtistId INTEGER NOT NULL.\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\t0 N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name  $NVARCHAR(120)\r\n)\n\n==Additional Context \n\nIn the SQLite dat$ abase invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please gene rate a valid SQL query without any explanations for the question. \n2. If the provided context is almost su fficient but requires knowledge of a specific string in a particular column, please generate an intermediat e 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 Get all playlists contai ning at least 10 tracks and the total duration of those tracks:\n"}, {"role": "assistant", "content": "SELE CT p.PlaylistId, SUM(t.Milliseconds) AS TotalDurationInMilliseconds\nFROM \"playlists\" p\nJOIN \"playlist track\" pt ON p.PlaylistId = pt.PlaylistId\nJOIN \"tracks\" t ON pt.TrackId = t.TrackId\nGROUP BY p.Playlis tId\nHAVING COUNT(pt.PlaylistId) >= 10\nORDER BY TotalDurationInMilliseconds DESC"}, {"role": "user", "cont ent": " \n Get all playlists containing at least 10 tracks and the total duration of those track s:\n"}, {"role": "assistant", "content": "SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM \"playlist track\" pt\nJOIN \"playlists\" p ON pt.PlaylistId = p.PlaylistId\nJOIN  $\t^{"tracks"} t ON pt.TrackId = t.TrackId \nGROUP BY pt.PlaylistId, p.Name \nHAVING COUNT(pt.TrackId) >= 10"},$ {"role": "user", "content": " \n List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"qenres\" q\nJOIN \"tracks\" t ON q.GenreId = t.GenreId\nGROUP BY q.Name"}, {"role": "user", "content": " \n List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS T otalTracks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalT racks DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all genres and the number of tracks in eac h genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"gen res\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks DESC"}, {"rol e": "user", "content": " \n There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular arti sts based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS Artis tName, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId \nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTracks DESC\nLIMI T 10"}, {"role": "user", "content": " \n There are 3 tables: artists, albums and tracks, where albums an d artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 mos t popular artists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT (t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"trac ks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user",

```
"content": " \n
                    Identify artists who have albums with tracks appearing in multiple genres:\n\n\n"},
{"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"album
s\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM \"genres\" q2\n JOI
N \"tracks\" t2 ON g2.GenreId = t2.GenreId\n GROUP BY g2.GenreId\n HAVING COUNT(g2.GenreId) > 1\n)\nGROUP
BY a.ArtistId, a.Name"}, {"role": "user", "content": " \n Identify artists who have albums with tracks
appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS Artis
tName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT
q2.GenreId\n FROM \"genres\" q2\n JOIN \"tracks\" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n
HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName"}, {"role": "user", "conte
              Find the customer who bought the most albums in total quantity (across all invoices): \n"},
{"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\"
i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DES
C\nLIMIT 1"}, {"role": "user", "content": " \n Get all playlists containing at least 10 tracks and the
total duration of those tracks:\n"}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T12:18:33.580723434Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDurationInMilliseconds\nFROM "playlists" p\nJOI
N "playlist track" pt ON p.PlaylistId = pt.PlaylistId\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROUP B
Y p.PlaylistId\nHAVING COUNT(pt.PlaylistId) >= 10\nORDER BY TotalDurationInMilliseconds DESC;'}, 'done reas
on': 'stop', 'done': True, 'total duration': 102379992467, 'load duration': 717212, 'prompt eval count': 18
86, 'prompt eval duration': 82153533000, 'eval count': 93, 'eval duration': 19608469000}
SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDurationInMilliseconds
FROM "plavlists" p
JOIN "playlist track" pt ON p.PlaylistId = pt.PlaylistId
JOIN "tracks" t ON pt.TrackId = t.TrackId
GROUP BY p.PlaylistId
HAVING COUNT(pt.PlaylistId) >= 10
ORDER BY TotalDurationInMilliseconds DESC:
Output from LLM: SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDurationInMilliseconds
FROM "playlists" p
JOIN "playlist track" pt ON p.PlaylistId = pt.PlaylistId
JOIN "tracks" t ON pt.TrackId = t.TrackId
GROUP BY p.PlavlistId
HAVING COUNT(pt.PlaylistId) >= 10
ORDER BY TotalDurationInMilliseconds DESC:
Extracted SQL: SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDurationInMilliseconds
FROM "playlists" p
JOIN "playlist track" pt ON p.PlaylistId = pt.PlaylistId
JOIN "tracks" t ON pt.TrackId = t.TrackId
GROUP BY p.PlaylistId
HAVING COUNT(pt.PlaylistId) >= 10
ORDER BY TotalDurationInMilliseconds DESC
```

```
SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDurationInMilliseconds
FROM "playlists" p
JOIN "playlist track" pt ON p.PlaylistId = pt.PlaylistId
JOIN "tracks" t ON pt.TrackId = t.TrackId
GROUP BY p.PlaylistId
HAVING COUNT(pt.PlavlistId) >= 10
ORDER BY TotalDurationInMilliseconds DESC
    PlaylistId TotalDurationInMilliseconds
0
             1
                                 877683083
             8
1
                                 877683083
2
             3
                                 501094957
3
            10
                                 501094957
             5
4
                                 398705153
5
            12
                                  21770592
6
            11
                                   9486559
7
            17
                                   8206312
8
            14
                                   7575051
9
            15
                                   7439811
10
            13
                                   6755730
11
            16
                                   4122018
Ollama parameters:
model=aya: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.PlaylistId, SUM
(t.Milliseconds) AS TotalDurationInMilliseconds\nFROM \"playlists\" p\nJOIN \"playlist track\" pt ON p.Pl
aylistId = pt.PlaylistId\nJOIN \"tracks\" t ON pt.TrackId = t.TrackId\nGROUP BY p.PlaylistId\nHAVING COUNT
(pt.PlaylistId) >= 10\nORDER BY TotalDurationInMilliseconds DESC\n\nThe following is information about the
resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n PlaylistId
                                                                                            int64\nTotalDu
                      int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python pl
rationInMilliseconds
otly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If t
here is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer wit
h any explanations -- just the code."}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T12:19:08.88830933Z', 'message': {'role': 'assistant', 'co
ntent': "```python\nimport pandas as pd\nimport plotly.express as px\n\n# Assuming the 'df' DataFrame has c
olumns 'PlaylistId' and 'TotalDurationInMilliseconds'\nfig = px.bar(df, x='PlaylistId', y='TotalDurationInM
illiseconds', color='PlaylistId',\n
                                              title='Top Playlists by Total Duration')\nfig.update xaxes
(title='Playlist ID')\nfig.update yaxes(title='Total Duration (Milliseconds)')\nfig.update layout(barmode
```

='stack')\nfig.show()\n```"}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 35279205269, 'load\_duration': 44466320, 'prompt\_eval\_count': 260, 'prompt\_eval\_duration': 10508878000, 'eval\_count': 122, 'eval\_duration': 24680742000}



```
Out[41]: ('SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDurationInMilliseconds\nFROM "playlists" p\nJOIN "pla
         ylist track" pt ON p.PlaylistId = pt.PlaylistId\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROUP BY p.Pl
         aylistId\nHAVING COUNT(pt.PlaylistId) >= 10\nORDER BY TotalDurationInMilliseconds DESC',
              PlaylistId TotalDurationInMilliseconds
                      1
                                           877683083
                       8
          1
                                           877683083
          2
                       3
                                           501094957
          3
                      10
                                           501094957
          4
                      5
                                           398705153
          5
                      12
                                           21770592
          6
                      11
                                            9486559
          7
                      17
                                             8206312
          8
                      14
                                             7575051
          9
                      15
                                             7439811
                      13
          10
                                             6755730
          11
                      16
                                             4122018,
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'PlaylistId=%{marker.color}<br/>br>TotalDurationInMilliseconds=%{y}<extra></ex
         tra>',
                        'legendaroup': ''.
                        'marker': {'color': array([ 1, 8, 3, 10, 5, 12, 11, 17, 14, 15, 13, 16]),
                                  'coloraxis': 'coloraxis',
                                  'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showledend': False,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array([ 1, 8, 3, 10, 5, 12, 11, 17, 14, 15, 13, 16]),
                        'xaxis': 'x',
                        'y': array([877683083, 877683083, 501094957, 501094957, 398705153, 21770592,
                                               8206312, 7575051, 7439811,
                                     9486559.
                                                                               6755730,
                                                                                          4122018]),
                        'yaxis': 'y'}],
              'lavout': {'barmode': 'stack',
                        'coloraxis': {'colorbar': {'title': {'text': 'PlaylistId'}},
                                      'colorscale': [[0.0, '#0d0887'], [0.111111111111111,
                                                     '#46039f'l. [0.22222222222222.
                                                     '#bd3786'], [0.55555555555556,
```

```
'#ed7953'], [0.7777777777778,
                                                    '#fb9f3a'], [0.888888888888888,
                                                    '#fdca26'], [1.0, '#f0f921']]},
                        'legend': {'tracegroupgap': 0},
                        'template': '...',
                        'title': {'text': 'Top Playlists by Total Duration'},
                        'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Playlist ID'}},
                        'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total Duration (Millisec
         onds)'}}}
          }))
        question = """
In [42]:
             Identify artists who have albums with tracks appearing in multiple genres:
         0.00
        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 fo rmat instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMEN Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n T NOT NULL,\r\n MediaTypeId INTEGER NOT NU LL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n tes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" (Al bumid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (G enreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK AlbumArtistI d ON "albums" (ArtistId)\n\nCREATE INDEX IFK TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK TrackAl bumId ON "tracks" (AlbumId)\n\nCREATE TABLE "albums"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (Arti stid) REFERENCES "artists" (Artistid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE TABLE "genres"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON "pla ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT ylist track" (TrackId)\n\nCREATE TABLE "artists"\r\n(\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "playlist track"\r\n(\r\n NULL,\r\n PlavlistId INTEGER NOT N CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackI ULL,\r\n TrackId INTEGER NOT NULL,\r\n d),\r\n FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDA TE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPD ATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explan ations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a spec ific 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 in sufficient, 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 Identify artists who have albums with tracks appearing in multi ple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artis ts" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT g2.GenreId\n FROM "ge nres" q2\n J0IN "tracks" t2 0N q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name'}, {'role': 'user', 'content': ' \n Identify artists who have albu ms with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT g2.GenreId\n FROM "genres" g2\n JOIN "tracks" t2 ON g2.GenreId = t2.GenreId\n GROUP BY g2.Ge nreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'role': 'use r', 'content': ' \n There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists bas ed on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "trac ks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'rol e': 'user', 'content': ' \n There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular arti sts based on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': '\n ist all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Titl e, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId'}, {'r ole': 'user', 'content': ' \n List all albums and their corresponding artist names \n'}, {'role': 'ass istant', 'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.Artis tId = ar.ArtistId'}, {'role': 'user', 'content': ' \n List all albums and their corresponding artist na mes \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId\n-----'}, {'role': 'user', 'content': '\n List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT q.Nam e, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.N ame'}, {'role': 'user', 'content': ' \n List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" q\nJOIN "tr acks" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'con tent': 'SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks DESC'}, {'role': 'user', 'content': '\n Identify art ists who have albums with tracks appearing in multiple genres:\n\n\n'}] Ollama parameters: model=aya:latest, options={},

keep alive=None

Prompt Content:

[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery 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 \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREM AlbumId INTEGER.\r\n ENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres \" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK Albu mArtistId ON \"albums\" (ArtistId)\n\nCREATE INDEX IFK TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX IFK TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL.\r\n EIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE INDEX IFK TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE TABLE \"genres\"\r\n(\r\n nreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK Playli stTrackTrackId ON \"playlist track\" (TrackId)\n\nCREATE TABLE \"artists\"\r\n(\r\n ArtistId INTEGER PRI Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"playlist track\"\r\n(\r\n MARY KEY AUTOINCREMENT NOT NULL,\r\n

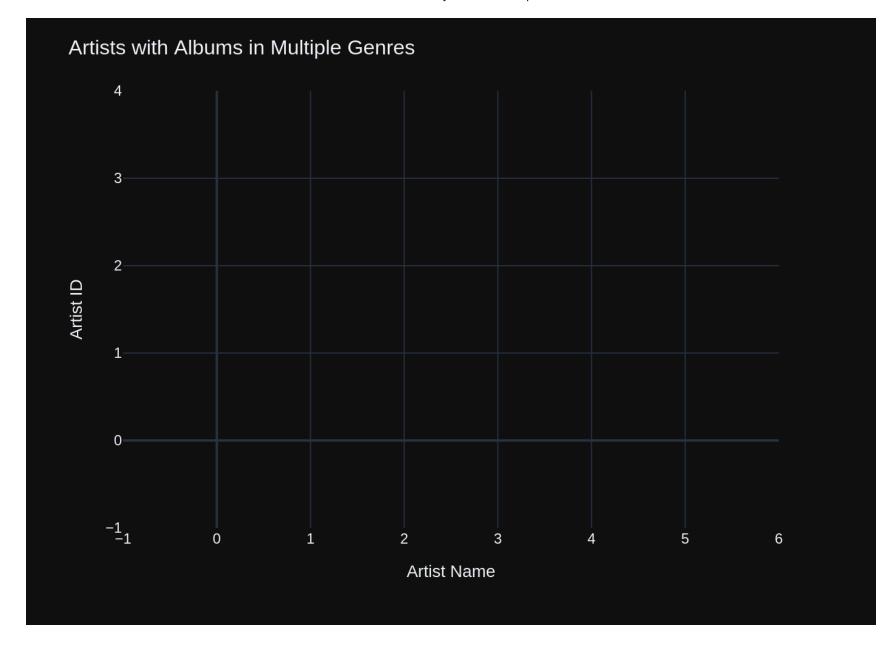
PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n CONSTRAINT PK PlaylistTrack PRIMARY FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\t0 KEY (PlavlistId, TrackId),\r\n N DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t \t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the SOLite database invoic e means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL guery 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 t o find the distinct strings in that column. Prepend the guery with a comment saying intermediate sql \n3. I f 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 exactl y as it was given before. \n"}, {"role": "user", "content": " \n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name A S ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM \"genres\" q2\n JOIN \"tracks\" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.Gen reId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name"}, {"role": "user", "content": "\n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM \"genres\" q2\n JOIN \"tracks\" t2 ON q2.G enreId = t2.GenreId\n GROUP BY g2.GenreId\n HAVING COUNT(g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName"}, {"role": "user", "content": " \n There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n the top 10 most popular artists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al O N a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": "\n There are 3 tables: artists, albums an d tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n n you find the top 10 most popular artists based on the number of tracks\n"}, {"role": "assistant", "conten t": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n List all albums and their corresponding artist names \n"}, {"ro le": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n List all albums and thei r corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, ar.Name AS ArtistName \nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Titl e, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId\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.GenreId) AS TotalTracks\nFROM \"genres\" g \nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"role": "user", "content": " \n all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, CO UNT(t.GenreId) AS TotalTracks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Na me\nORDER BY TotalTracks DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all genres and the numb

```
er of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS Total
Tracks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTra
cks DESC"}, {"role": "user", "content": " \n Identify artists who have albums with tracks appearing in
multiple genres:\n\n\n"}]
Ollama Response:
{'model': 'aya:latest', 'created at': '2024-06-14T12:20:48.292270826Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.
ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM "genres" q2\n JOIN "tracks" t2 ON q2.GenreI
d = t2.GenreId\n GROUP BY g2.GenreId\n HAVING COUNT(DISTINCT g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.N
ame ORDER BY ArtistName'}, 'done reason': 'stop', 'done': True, 'total duration': 99309355580, 'load durati
on': 695941, 'prompt eval count': 1713, 'prompt eval duration': 72941843000, 'eval count': 122, 'eval durat
ion': 25767649000}
SELECT a.ArtistId, a.Name AS ArtistName
FROM "artists" a
JOIN "albums" al ON a.ArtistId = al.ArtistId
WHERE a.ArtistId IN (
 SELECT g2.GenreId
  FROM "genres" q2
  JOIN "tracks" t2 ON g2.GenreId = t2.GenreId
  GROUP BY q2.GenreId
  HAVING COUNT(DISTINCT g2.GenreId) > 1
GROUP BY a.ArtistId, a.Name ORDER BY ArtistName
SELECT a.ArtistId, a.Name AS ArtistName
FROM "artists" a
JOIN "albums" al ON a.ArtistId = al.ArtistId
WHERE a.ArtistId IN (
  SELECT q2.GenreId
  FROM "genres" q2
  JOIN "tracks" t2 ON g2.GenreId = t2.GenreId
  GROUP BY q2.GenreId
  HAVING COUNT(DISTINCT g2.GenreId) > 1
GROUP BY a.ArtistId, a.Name ORDER BY ArtistName
Empty DataFrame
Columns: [ArtistId, ArtistName]
Index: []
Ollama parameters:
model=aya: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 Identify artists who have albums with tracks appearing in multiple genres:\n\n\n'\n\nThe DataFrame was produced using this query: SELECT a.ArtistId, a.Name AS Art istName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SEL ECT g2.GenreId\n FROM \"genres\" g2\n JOIN \"tracks\" t2 ON g2.GenreId = t2.GenreId\n GROUP BY g2.GenreId\n HAVING COUNT(DISTINCT g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName\n\nThe fol lowing is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n ArtistId object\nArtistName object\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plo tly 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': 'aya:latest', 'created\_at': '2024-06-14T12:21:22.563619697Z', 'message': {'role': 'assistant', 'c ontent': '```python\nimport plotly.express as px\n\n# Create a bar chart\nfig = px.bar(df, x="ArtistName", y="ArtistId", color="ArtistName")\n\n# Add labels and title\nfig.update\_xaxes(title="Artist Name")\nfig.update\_yaxes(title="Artist ID")\nfig.update\_layout(title="Artists with Albums in Multiple Genres")\n\n# Show t he chart\nfig.show()\n```'}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 34268888635, 'load\_duration': 43028180, 'prompt\_eval\_count': 279, 'prompt\_eval\_duration': 12069928000, 'eval\_count': 105, 'eval\_duration': 22102998000}

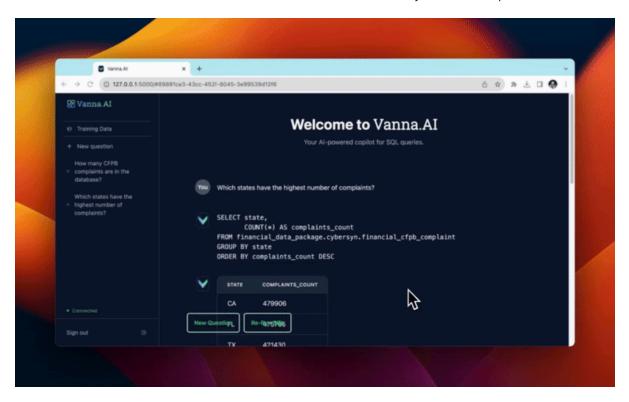


```
Out[42]: ('SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistI
          d\nWHERE a.ArtistId IN (\n SELECT g2.GenreId\n FROM "genres" g2\n JOIN "tracks" t2 ON g2.GenreId = t
          2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(DISTINCT q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name
          ORDER BY ArtistName',
           Empty DataFrame
          Columns: [ArtistId, ArtistName]
           Index: [],
           Figure({
               'data': [],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'title': {'text': 'Artists with Albums in Multiple Genres'},
                          'xaxis': {'anchor': 'y',
                                    'categoryarray': [],
                                    'categoryorder': 'array',
                                    'domain': [0.0, 1.0],
                                    'title': {'text': 'Artist Name'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Artist ID'}}}
          }))
```

### Check completion time

```
In []:
In [43]: ts_stop = time()
    elapsed_time = ts_stop - ts_start
    print(f"test running on '{hostname}' with '{model_name}' LLM took : {elapsed_time:.2f} sec")
    test running on 'ducklover1' with 'aya' LLM took : 2988.49 sec
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