Generating SQL for SQLite using Ollama, ChromaDB

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

Which LLM do you want to use?

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

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

- Vanna Hosted Vector DB (Recommended)
  Use Vanna.AIs 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 = 'phi3'
```

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.
    """
    # Use regular expressions to match and replace square brackets
    pattern = r"\[([^\]]+)]"  # 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	0658ba3d-98ff-51f4-9006-a24f87045858-sql	How many customers are there	SELECT COUNT(*) FROM "customers"	sql
2	0e1a2b7b-d65e-53de-b839-edb7afcf4ab1-sql	\n Hint: album quantity is found in invoi...	SELECT i.CustomerId, COUNT(ii.TrackId) AS Tota...	sql
3	127fd4bd-b9af-539d-9313-1d0234d073b7-sql	\n There are 3 tables: artists, albums and...	SELECT a.Name, COUNT(t.TrackId) AS TotalTracks...	sql
4	17d893d5-1417-5ba3-a5ca-9f6ce15a727f-sql	\n Identify artists who have albums with...	SELECT a.ArtistId, a.Name AS ArtistName\nFROM ...	sql
5	3013d1b4-feb2-519d-bfb9-114500436e3d-sql	\n Find the customer with the most invoi...	SELECT c.CustomerId, COUNT(i.InvoiceId) AS Tot...	sql
6	32b99e7b-31ab-55d8-8431-fb010fa7af85-sql	\n Find the top 5 customers who spent th...	SELECT c.CustomerId, SUM(i.Total) AS TotalSpen...	sql
7	49e67df3-a604-51f8-ad01-b8f5a2043eac-sql	\n Get the total number of invoices for e...	SELECT c.CustomerId, COUNT(i.InvoiceId) AS Tot...	sql
8	584873f8-1904-50f1-8f80-7ccf08059264-sql	\n List all customers from Canada and th...	SELECT c.Email, c.Country\nFROM "customers" c\...	sql
9	6bed484b-9a80-57f4-ad89-5f775b5df252-sql	\n Get the average invoice total for each...	SELECT c.CustomerId, AVG(i.Total) AS AverageIn...	sql
10	6f22268c-5062-5f11-ba2d-8555f06b409d-sql	\n Find all tracks with a name containing...	SELECT * \nFROM "tracks" \nWHERE LOWER(Name) L...	sql
11	70b4f686-c71b-5ee8-9458-6bbc776349bf-sql	\n Find all invoices since 2010 and the t...	SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmo...	sql
12	9a396a33-ecea-51a8-bd05-28f58a86eb86-sql	\n Hint: album quantity is found in invoi...	SELECT c.CustomerId, COUNT(ii.TrackId) AS Tota...	sql
13	9a9c970b-b94c-5f22-b54c-b86921a38b65-sql	\n Identify artists who have albums with...	SELECT a.ArtistId, a.Name AS ArtistName\nFROM ...	sql
14	a7185c88-7417-5b75-a52e-4eaef5f9deca-sql	\n List all albums and their correspondin...	SELECT a.Title, a.ArtistId, ar.Name AS ArtistN...	sql
15	aea89953-21b2-55d1-9dda-431ee6033c3d-sql	\n List all invoices with a total exceedi...	SELECT * \nFROM "invoices" \nWHERE Total > 10.00	sql
16	d1d70c18-f5d9-5970-a32c-	\n Find the customer who	SELECT c.CustomerId, COUNT(ii.TrackId)	sql

	id	question	content	training_data_type
	914deeca1087-sql	bought the most...	AS Tota...	
17	d8a2f948-dffa-5524-a5f9-174cc1a8da73-sql	Can you list all tables in the SQLite database...	SELECT name FROM sqlite_master WHERE type='table'	sql
18	d8a37163-5ce5-58cd-a316-ea5598d44d27-sql	what are the top 5 countries that customers co...	SELECT c.Country, COUNT(*) AS TotalCustomers\n...	sql
19	dd282d7c-a4ef-5e3a-87e0-cb45fac50808-sql	\n Find the total number of invoices per ...	SELECT i.BillingCountry, COUNT(*) AS TotalInvo...	sql
20	e7c4b3aa-664f-5f87-8b25-449a4482f3fd-sql	\n Get all playlists containing at least...	SELECT pt.PlaylistId, p.Name AS PlaylistName, ...	sql
21	f33f8cb6-1b12-5ea7-8d9a-aef8166b9970-sql	\n Find the top 5 most expensive tracks (...	SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "t...	sql
22	f626b681-4d8f-563a-beee-1ea759baaa82-sql	\n List all genres and the number of trac...	SELECT g.Name, COUNT(t.GenreId) AS TotalTracks...	sql
23	fd25ebba-4066-5a0f-8613-7b1c2ace0339-sql	\n List all employees and their reporting...	SELECT e.FirstName, e.LastName, mt.FirstName A...	sql
0	039f9d54-59f7-5f29-8c04-14dbc3e95671-ddl	None	CREATE TABLE "artists"\r\n(\r\n ArtistId IN...	ddl
1	0db84e3d-ef41-563c-803e-21c1b985dc19-ddl	None	CREATE TABLE "invoices"\r\n(\r\n InvoiceId ...	ddl
2	10cba811-ddba-5042-9e90-d764dfcd1629-ddl	None	CREATE INDEX IFK_InvoiceCustomerId ON "invoice...	ddl
3	2c711317-b93d-5f60-a728-cb1c6fcbc040-ddl	None	CREATE INDEX IFK_CustomerSupportRepId ON "cust...	ddl
4	37319c81-65f7-50ee-956b-795de244bee5-ddl	None	CREATE TABLE sqlite_stat1(tbl,idx,stat)	ddl
5	40bd77cd-e1de-5872-8693-624117ff413c-ddl	None	CREATE INDEX IFK_InvoiceLineInvoiceId ON "invo...	ddl
6	41130543-7164-562a-90a7-0fd0a409c154-ddl	None	CREATE TABLE "albums"\r\n(\r\n AlbumId INTE...	ddl
7	458debc8-8082-5450-a17a-66028bd55ace-ddl	None	CREATE TABLE "playlists"\r\n(\r\n PlaylistI...	ddl
8	4815f3fd-925b-53ce-9dfa-0e4285d5abd3-ddl	None	CREATE TABLE "invoice_items"\r\n(\r\n Invoi...	ddl

	id	question	content	training_data_type
9	48d484e9-984c-58ff-b391-75521c69d486-ddl	None	CREATE INDEX IFK_PlaylistTrackTrackId ON "play...	ddl
10	551e1120-a6ee-554f-8b8a-ccf4f22d3636-ddl	None	CREATE INDEX IFK_AlbumArtistId ON "albums" (Ar...	ddl
11	5ff4911e-45c1-5a59-9566-243a9b6a3320-ddl	None	CREATE TABLE "employees"\r\n(\r\n EmployeeI...	ddl
12	65df0648-bf05-5f75-9365-c21f54b2302d-ddl	None	CREATE TABLE "media_types"\r\n(\r\n MediaTy...	ddl
13	6b585176-e66d-5b23-8d86-ca8a80e3af3d-ddl	None	CREATE INDEX IFK_EmployeeReportsTo ON "employe...	ddl
14	868758b8-e018-55e7-8cc3-75c0e6d211c8-ddl	None	CREATE INDEX IFK_TrackAlbumId ON "tracks" (Alb...	ddl
15	9ea4613d-c1be-5a77-ada9-c54ee3f0cab7-ddl	None	CREATE INDEX IFK_TrackMediaTypeId ON "tracks" ...	ddl
16	a9c9a852-608d-5ef2-aede-26ba098d83d1-ddl	None	CREATE INDEX IFK_TrackGenreId ON "tracks" (Gen...	ddl
17	b42cc9e1-9219-5a42-9a06-de906f76239e-ddl	None	CREATE TABLE "tracks"\r\n(\r\n TrackId INTE...	ddl
18	c387b9d2-5ff4-5a07-8364-f5dab45bb2a9-ddl	None	CREATE TABLE "genres"\r\n(\r\n GenreId INTE...	ddl
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 CustomerI...	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

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():
        ddl = strip_brackets(ddl)
        vn.train(ddl=ddl)
```

In [15]:
```
if False:
    # Sometimes you may want to add documentation about your business terminology or definitions.
    vn.train(documentation="In the SQLite database invoice means order")
```

Asking the AI

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

In [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-a5f9-174cc1a8da73-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-a5f9-174cc1a8da73-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-1ea759baaa82-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-a5f9-174cc1a8da73-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-1ea759baaa82-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
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE "playlists"\r\n(\r\n   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\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   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 "media_types"\r\n(\r\n   MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "artists"\r\n(\r\n   ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(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   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) 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 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\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\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 sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'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 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   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', 'content': ' \n   List all customers from Canada and their email addresses:\n'}, {'role': 'assistant', 'content': 'SELECT c.Email, c.Country\nFR

OM "customers" c\nWHERE c.Country = \'Canada\'\'}, {'role': 'user', 'content': ' \n   Find the customer w
ho 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.Customer
Id = i.CustomerId\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY T
otalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   List all genres and the number of tracks i
n 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'}, {'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, mt.FirstName AS ManagerFirstName, mt.LastName AS ManagerLastName\nFROM "employe
es" e\nLEFT JOIN "employees" mt ON e.ReportsTo = mt.EmployeeId'}, {'role': 'user', 'content': ' \n   Hin
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 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.UnitPr
ice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   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 TotalDuration\nFROM "playlist_track" pt\nJOIN "playlists" p ON pt.P
laylistId = p.PlaylistId\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING
COUNT(pt.TrackId) >= 10'}, {'role': 'user', 'content': 'Can you list all tables in the SQLite database cata
log?'}]
Ollama parameters:
model=phi3: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 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
\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n   GenreId INTEGER PRIMARY KEY AUTOINCREM
ENT NOT NULL,\r\n   Name NVARCHAR(120)\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   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) 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   Name NVARC
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\tON 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
\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   Arti
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\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 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": " \n   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.Name, COUNT(t.TrackId) AS TotalTracks\n
FROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.Alb
umId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n   List all
albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.Arti
stId, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"role":
"user", "content": " \n   List all customers from Canada and their email addresses:\n"}, {"role": "assis
tant", "content": "SELECT c.Email, c.Country\nFROM \"customers\" c\nWHERE c.Country = 'Canada'"}, {"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.Invoice
Id = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content":
" \n   List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SEL
ECT 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   List all employees and their reporting manager's
name (if any):\n"}, {"role": "assistant", "content": "SELECT e.FirstName, e.LastName, mt.FirstName AS Manag
erFirstName, mt.LastName AS ManagerLastName\nFROM \"employees\" e\nLEFT JOIN \"employees\" mt ON e.ReportsT
o = mt.EmployeeId"}, {"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 invoice
s):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"cu
stomers\" 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   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   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 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\nHAVING COUNT(pt.TrackId) >= 10"}, {"role": "user", "content": "Can you list all tables in the SQLite database catalog?"}]

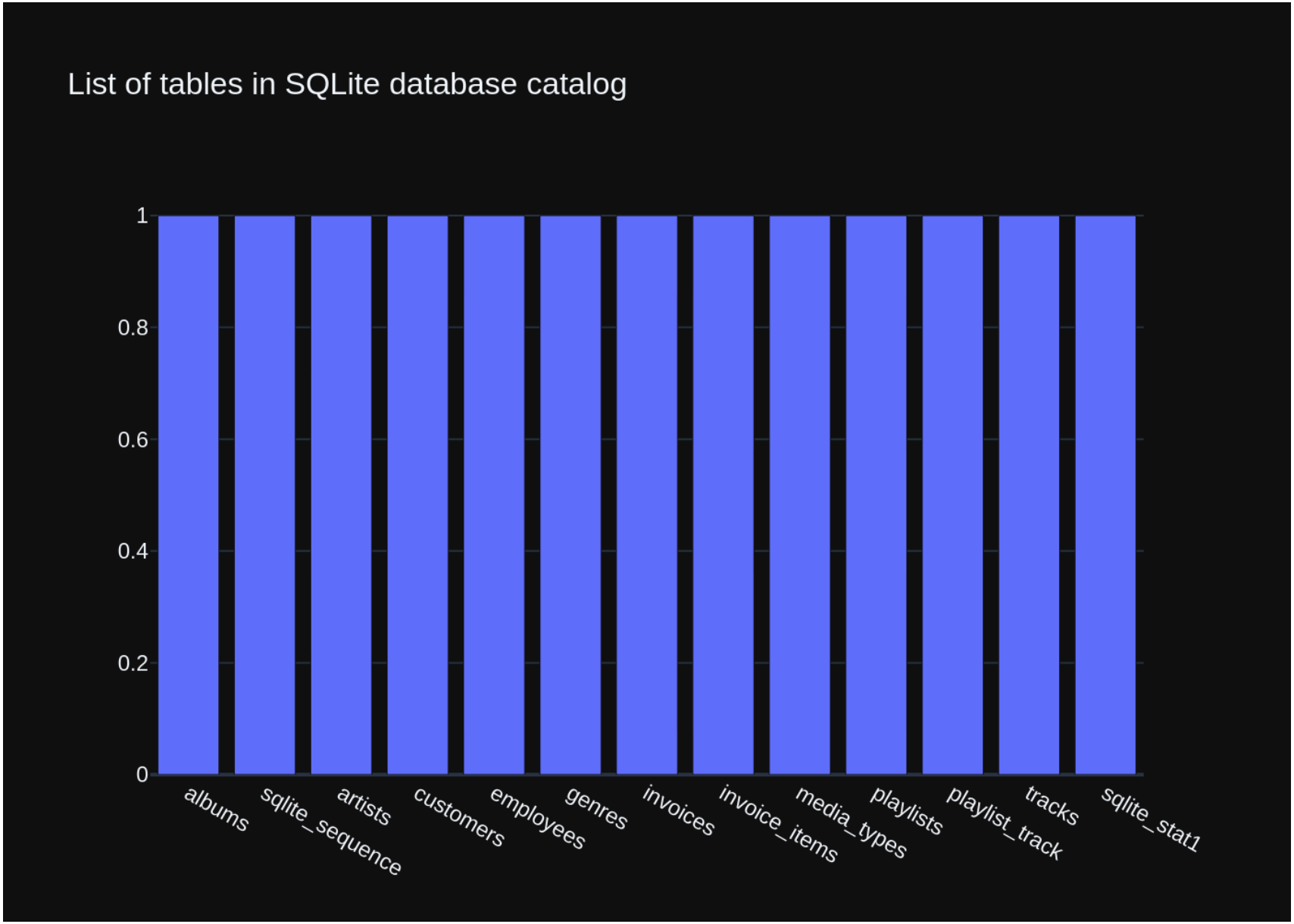
Add of existing embedding ID: d8a2f948-dffa-5524-a5f9-174cc1a8da73-sql
Insert of existing embedding ID: d8a2f948-dffa-5524-a5f9-174cc1a8da73-sql

```
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:37:12.019981085Z', 'message': {'role': 'assistant',
'content': "SELECT name FROM sqlite_master WHERE type='table';\n\nThis SQL query selects and returns the na
mes of all tables in the SQLite database catalog by checking for entries in the `sqlite_master` table where
the `type` column has a value of 'table'. The `sqlite_master` is a special table that holds information abo
ut all other tables, indexes, triggers, views, etc. within the SQLite database. This query gives us a list
of all such objects without any duplicates or extra columns since it directly targets the relevant entries
in this internal catalog table."}, 'done_reason': 'stop', 'done': True, 'total_duration': 56245347270, 'loa
d_duration': 1081768687, 'prompt_eval_count': 1952, 'prompt_eval_duration': 40838681000, 'eval_count': 115,
'eval_duration': 13722406000}
SELECT name FROM sqlite_master WHERE type='table';

This SQL query selects and returns the names of all tables in the SQLite database catalog by checking for e
ntries in the `sqlite_master` table where the `type` column has a value of 'table'. The `sqlite_master` is
a special table that holds information about all other tables, indexes, triggers, views, etc. within the SQ
Lite database. This query gives us a list of all such objects without any duplicates or extra columns since
it directly targets the relevant entries in this internal catalog table.
Output from LLM: SELECT name FROM sqlite_master WHERE type='table';

This SQL query selects and returns the names of all tables in the SQLite database catalog by checking for e
ntries in the `sqlite_master` table where the `type` column has a value of 'table'. The `sqlite_master` is
a special table that holds information about all other tables, indexes, triggers, views, etc. within the SQ
Lite database. This query gives us a list of all such objects without any duplicates or extra columns since
it directly targets the relevant entries in this internal catalog table.
Extracted SQL: SELECT name FROM sqlite_master WHERE type='table'
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
Ollama parameters:
model=phi3: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 query: SELECT name FROM sqlite_master WHERE type='table'\n\nThe following
is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n name    object\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': 'phi3:latest', 'created_at': '2024-06-14T02:37:26.583025276Z', 'message': {'role': 'assistant',
'content': "import plotly.graph_objs as go\n\ntrace = go.Bar(x=list(df['name']), y=[1]*len(df['name']))\n\n
data = [trace]\n\nlayout = go.Layout(title='List of tables in SQLite database catalog')\n\nfig = go.Figure
(data=data, layout=layout)\n\nfig.show()"}, 'done_reason': 'stop', 'done': True, 'total_duration': 14534845
712, 'load_duration': 45348429, 'prompt_eval_count': 162, 'prompt_eval_duration': 4290449000, 'eval_count':
86, 'eval_duration': 10154040000}
```


List of tables in SQLite database catalog
1
0.8
0.6
0.4
0.2
0
albums
sqlite_sequence
artists
customers
employees
genres
invoices
invoice_items
media_types
playlists
playlist_track
tracks
sqlite_stat1

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': [{'type': 'bar',
              'x': [albums, sqlite_sequence, artists, customers, employees,
                    genres, invoices, invoice_items, media_types, playlists,
                    playlist_track, tracks, sqlite_stat1],
              'y': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]}],
    'layout': {'template': '...', 'title': {'text': 'List of tables in SQLite database catalog'}}
}))
```

In [18]:
```
vn.ask(question="which table stores customer's orders")
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "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   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) 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_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   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) 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 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 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 "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   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),\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   PlaylistId INTEGER PRIMARY 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   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_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 "media_types"\r\n(\r\n   MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\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 query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Find the top 5 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   Find the customer with the mos
t 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 Tot
alInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   Find the customer who bought the most alb
ums 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\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 invoice_items, \n   \n   Find
the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assi
stant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoic
es" i ON c.CustomerId = i.CustomerId\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.Cus
tomerId\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 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.InvoiceId = ii.InvoiceId\nGROUP BY i.Custom
erId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Get the total number of in
voices 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.CustomerId\nGROUP BY c.CustomerI
d'}, {'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': 'assi
stant', '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 5'}, {'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 To
talCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   List all customers from Canada and thei
r email addresses:\n'}, {'role': 'assistant', 'content': 'SELECT c.Email, c.Country\nFROM "customers" c\nWH
ERE c.Country = \'Canada\''}, {'role': 'user', 'content': ' \n   Get the average invoice total for each c
ustomer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nF
ROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'use
r', 'content': "which table stores customer's orders"}]
Ollama parameters:
model=phi3: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   BillingCou
ntry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FORE

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
T NULL,\r\n   Quantity 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 TABLE \"customers\"\r\n(\r\n   Custom
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   Phone NV
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
0),\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 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   PlaylistId 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   Title NV
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
(\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\" (TrackI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"media_types\"\r\n(\r\n   MediaTy
peId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\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 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   Fin
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"}, {"role": "user", "conten
t": " \n   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.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 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 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 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 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 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.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"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 \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"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 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"}, {"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 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   Get the average invoice 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.CustomerId"}, {"role": "user", "content": "which table stores customer's orders"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:38:28.157625127Z', 'message': {'role': 'assistant', 'content': 'The "invoices" table stores the customer\'s orders, as it represents each individual order made by a customer in the SQLite database. Each row within this table corresponds to an invoice and contains information about that specific order, including details related to the customer who placed the order (CustomerId).'}, 'done_reason': 'stop', 'done': True, 'total_duration': 61200302233, 'load_duration': 821925, 'prompt_eval_count': 1983, 'prompt_eval_duration': 51448853000, 'eval_count': 62, 'eval_duration': 8970229000}
The "invoices" table stores the customer's orders, as it represents each individual order made by a customer in the SQLite database. Each row within this table corresponds to an invoice and contains information about that specific order, including details related to the customer who placed the order (CustomerId).
The "invoices" table stores the customer's orders, as it represents each individual order made by a customer in the SQLite database. Each row within this table corresponds to an invoice and contains information about that specific order, including details related to the customer who placed the order (CustomerId).
Couldn't run sql: Execution failed on sql 'The "invoices" table stores the customer's orders, as it represents each individual order made by a customer in the SQLite database. Each row within this table correspond

```
s to an invoice and contains information about that specific order, including details related to the custom
er who placed the order (CustomerId).': 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 query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "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   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) 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_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 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 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 INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\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   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) REFERENCES "tracks" (TrackId) \r\n\t\tON 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   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_InvoiceLineTrackId 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 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\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 "playlists"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\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 query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM "customers"'}, {'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': '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 deta

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'}, {'role': 'user', 'content': ' \n   Find the customer who bought the most alb
ums 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\nJOIN
"invoice_items" ii ON i.InvoiceId = ii.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.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" 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': 'SELECT c.Country, COUNT
(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'r
ole': '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\nGROUP BY c.CustomerId\n
ORDER 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 albums in total quantity (across al
l invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAl
bums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDE
R BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Hint: album quantity is found in inv
oice_items, \n   \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'}, {'role': 'user', 'content': ' \n   Get the average invoice total for each custom
er:\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'}, {'role': 'user',
'content': 'How many customers are there'}]
Ollama parameters:
model=phi3: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   BillingCou
ntry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FORE
IGN 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\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

City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(1
0),\r\n   Phone 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\tON 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   Invo
iceId 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) 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\tO
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
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_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
E,\r\n   Address 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   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
MENT NOT NULL,\r\n   Name NVARCHAR(120)\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 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 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 question has been asked and answered before, please repeat the answer exactly
as it was given before. \n"}, {"role": "user", "content": "How many customers are there"}, {"role": "assist
ant", "content": "SELECT COUNT(*) FROM \"customers\""}, {"role": "user", "content": " \n   Find the cust
omer 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.Cust
omerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"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"}, {"role": "assistant", "content": "SELECT c.Cus
tomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Customer
Id\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 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 1"}, {"role": "user", "content": " \n   Get the total number
of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceI
d) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.C
ustomerId"}, {"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.Country\nORDER BY TotalCustomers 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 c.CustomerId, COUNT(ii.TrackId) A
S TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice_ite
ms\" 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   \n   Find the top 5
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_
items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"ro
le": "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", "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"}, {"role":
"user", "content": " \n   Get the average invoice total for each customer:\n"}, {"role": "assistant", "co
ntent": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJOIN \"invoices\"
i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": "How many customers
are there"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:39:37.840976915Z', 'message': {'role': 'assistant',
'content': 'SELECT COUNT(*) as TotalCustomers FROM "customers"\nThis SQL query will return the total number
of unique customers present in the "customers" table. The COUNT(*) function is an aggregate function that r
eturns the number of rows present in a given table, in this case, the \'customers\' table. By enclosing it
with SELECT and appending as TotalCustomers to label the output column, we get our desired count result.'},
'done_reason': 'stop', 'done': True, 'total_duration': 69638430002, 'load_duration': 887236, 'prompt_eval_c
ount': 1931, 'prompt_eval_duration': 54492266000, 'eval_count': 92, 'eval_duration': 14322154000}
SELECT COUNT(*) as TotalCustomers FROM "customers"
This SQL query will return the total number of unique customers present in the "customers" table. The COUNT
(*) function is an aggregate function that returns the number of rows present in a given table, in this cas
e, the 'customers' table. By enclosing it with SELECT and appending as TotalCustomers to label the output c
olumn, we get our desired count result.
SELECT COUNT(*) as TotalCustomers FROM "customers"
This SQL query will return the total number of unique customers present in the "customers" table. The COUNT
(*) function is an aggregate function that returns the number of rows present in a given table, in this cas
e, the 'customers' table. By enclosing it with SELECT and appending as TotalCustomers to label the output c
olumn, we get our desired count result.
Couldn't run sql: Execution failed on sql 'SELECT COUNT(*) as TotalCustomers FROM "customers"
This SQL query will return the total number of unique customers present in the "customers" table. The COUNT
(*) function is an aggregate function that returns the number of rows present in a given table, in this cas
e, the 'customers' table. By enclosing it with SELECT and appending as TotalCustomers to label the output c
olumn, we get our desired count result.': near "SQL": syntax error

In [ ]:

In [20]:
```
vn.ask(question="what are the top 5 countries that customers come from?")
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "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   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) 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 "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 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 "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   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) 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   Name NVARCHAR(120)\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 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(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\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_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 sqlite_sequence(name,seq)\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   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\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 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 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': '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   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.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.Email, c.Country\nFROM "customers" c\nWHERE c.Country = \'Canada\''}, {'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\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\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices 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 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.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the total number 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   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 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'}, {'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': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM "customers"'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}]
Ollama parameters:
model=phi3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE \"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   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   FORE
IGN 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
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
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 \"invoice_items\"\r\n(\r\n   InvoiceL
ineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGE
R NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KE
Y (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   F
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   N
ame NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE T
ABLE \"employees\"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARC
HAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INT
EGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHA
R(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),\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
\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n   PlaylistId INTEG
ER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistI
d, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACT
ION 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 sqlite_sequence(name,seq)\n\nCREATE TABLE \"tracks\"\r\n(\r
\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   Album
Id 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\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 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 quest
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 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.Email, c.Country\nFROM \"customers\" c\nWHERE c.Country = 'Canada'"}, {"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\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\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices 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 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.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find the total number 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   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 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"}, {"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": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM \"customers\""}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:41:07.754537316Z', 'message': {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(i.*) AS CustomerCount\nFROM "invoices" i\nGROUP BY i.BillingCountry\nORDER BY CustomerCount DESC\nLIMIT 5\n- Response: The above SQL query will return the top 5 countries that customers come from based on the number of invoices associated with each country. It groups all invoices by their billing country and counts the number of records (invoices) for each group, then orders them in descending order by this count to get the top countries first.'}, 'done_reason': 'stop', 'done': True, 'total_duration': 89870392143, 'load_duration': 910649, 'prompt_eval_count': 2029, 'prompt_eval_duration': 65539874000, 'eval_count': 118, 'eval_duration': 23436033000}
SELECT i.BillingCountry, COUNT(i.*) AS CustomerCount
FROM "invoices" i
GROUP BY i.BillingCountry

```
ORDER BY CustomerCount DESC
LIMIT 5
- Response: The above SQL query will return the top 5 countries that customers come from based on the numbe
r of invoices associated with each country. It groups all invoices by their billing country and counts the
number of records (invoices) for each group, then orders them in descending order by this count to get the
top countries first.
SELECT i.BillingCountry, COUNT(i.*) AS CustomerCount
FROM "invoices" i
GROUP BY i.BillingCountry
ORDER BY CustomerCount DESC
LIMIT 5
- Response: The above SQL query will return the top 5 countries that customers come from based on the numbe
r of invoices associated with each country. It groups all invoices by their billing country and counts the
number of records (invoices) for each group, then orders them in descending order by this count to get the
top countries first.
Couldn't run sql:  Execution failed on sql 'SELECT i.BillingCountry, COUNT(i.*) AS CustomerCount
FROM "invoices" i
GROUP BY i.BillingCountry
ORDER BY CustomerCount DESC
LIMIT 5
- Response: The above SQL query will return the top 5 countries that customers come from based on the numbe
r of invoices associated with each country. It groups all invoices by their billing country and counts the
number of records (invoices) for each group, then orders them in descending order by this count to get the
top countries first.': near "*": syntax error
```

More SQL questions

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

In [21]:
```
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 query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_AlbumArtistId ON "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   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 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 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_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n   ArtistId INTEGER 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 TABLE "playlists"\r\n(\r\n   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\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 context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId'}, {'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 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   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 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   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 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'}, {'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 invo

ices):\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 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 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 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.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 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 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.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'}, {'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   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 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\nHAVING COUNT(pt.TrackId) >= 10'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}]
Ollama parameters:
model=phi3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_AlbumArtistId ON \"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   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 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 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_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"artists\"\r\n(\r\n   ArtistId INTEGER 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

LE \"playlists\"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(12
0)\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n   GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   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 query to find the distinct strings in that column. Prepend the query 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": "
\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.Artis
tId"}, {"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 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 g2.GenreId\n FROM \"genres\" g2\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 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 g2.GenreId\n FROM \"genres\" g2\n JOIN \"tracks\" t2 ON g2.GenreId = t2.GenreId\n G
ROUP BY g2.GenreId\n HAVING COUNT(g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name"}, {"role": "user", "con
tent": " \n   Hint: album quantity is found in invoice_items, \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
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"}, {"role": "user", "content": " \n   Find the customer who bought the most albums i
n 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 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 albums in total quantity (across all invoices):\n"}, {"role": "assi
stant", "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   List all genres and the number of tracks in each genre:\n"}, {"ro
le": "assistant", "content": "SELECT g.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   Get all playlist
s containing at least 10 tracks and the total duration of those tracks:\n"}, {"role": "assistant", "conten
t": "SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM \"playlist_tr
ack\" 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", "content": " \n   Li
st all albums and their corresponding artist names \n"}]

Add of existing embedding ID: a7185c88-7417-5b75-a52e-4eaef5f9deca-sql
Insert of existing embedding ID: a7185c88-7417-5b75-a52e-4eaef5f9deca-sql

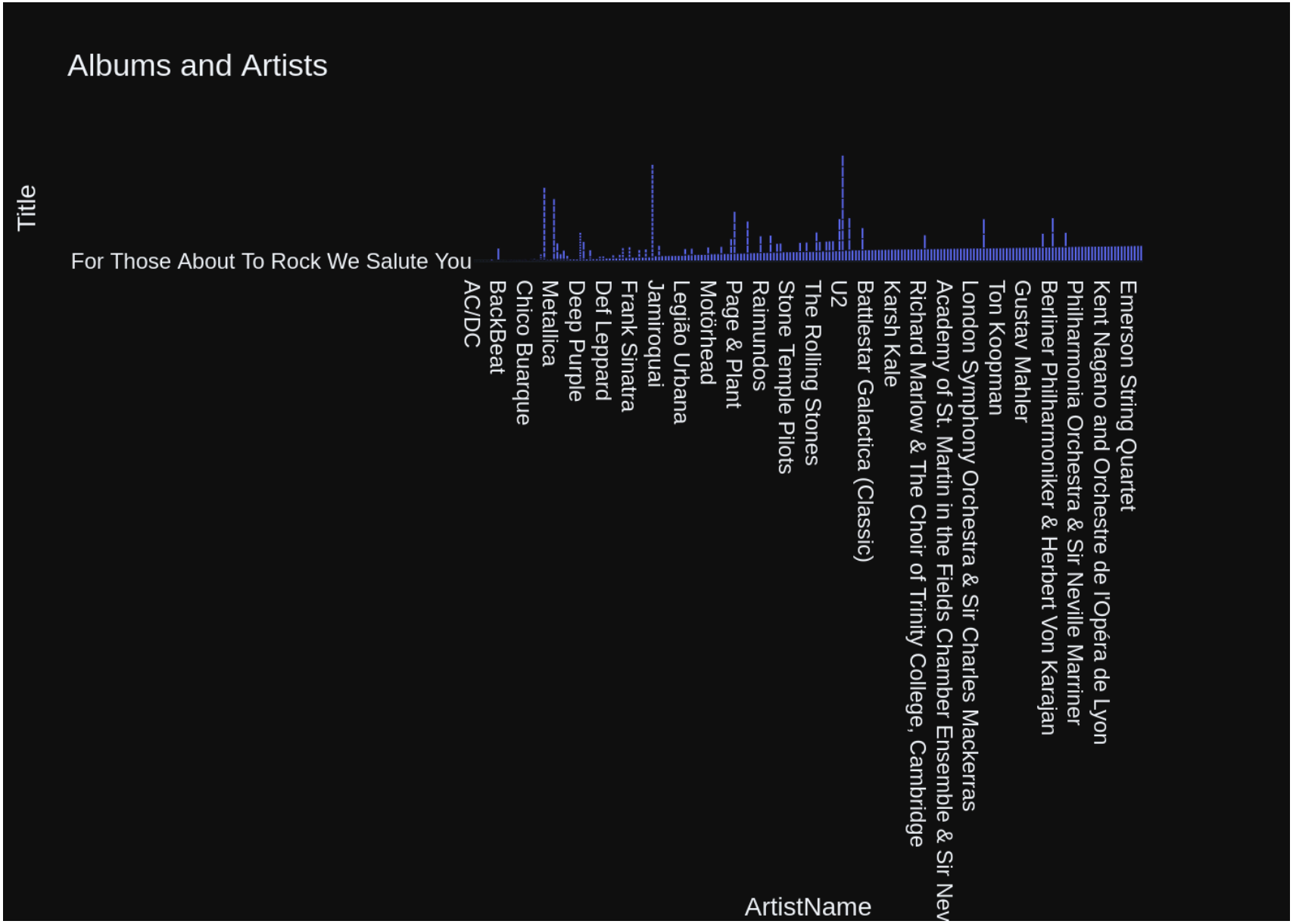
```
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:42:40.659318525Z', 'message': {'role': 'assistant',
'content': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.Arti
stId = ar.ArtistId;\n\nThis SQL statement retrieves the title of each album along with the associated artis
t\'s ID and name from two tables: \'albums\' and \'artists\'. The JOIN operation links records in both tabl
es based on their shared \'ArtistId\' field, ensuring that for every album listed, there is an accompanying
artist record.'}, 'done_reason': 'stop', 'done': True, 'total_duration': 92861642143, 'load_duration': 1610
308, 'prompt_eval_count': 1989, 'prompt_eval_duration': 70868665000, 'eval_count': 118, 'eval_duration': 21
408066000}
SELECT a.Title, a.ArtistId, ar.Name AS ArtistName
FROM "albums" a
JOIN "artists" ar ON a.ArtistId = ar.ArtistId;

This SQL statement retrieves the title of each album along with the associated artist's ID and name from tw
o tables: 'albums' and 'artists'. The JOIN operation links records in both tables based on their shared 'Ar
tistId' field, ensuring that for every album listed, there is an accompanying artist record.
Output from LLM: SELECT a.Title, a.ArtistId, ar.Name AS ArtistName
FROM "albums" a
JOIN "artists" ar ON a.ArtistId = ar.ArtistId;

This SQL statement retrieves the title of each album along with the associated artist's ID and name from tw
o tables: 'albums' and 'artists'. The JOIN operation links records in both tables based on their shared 'Ar
tistId' field, ensuring that for every album listed, there is an accompanying artist record.
Extracted SQL: SELECT a.Title, a.ArtistId, ar.Name AS ArtistName
FROM "albums" a
JOIN "artists" ar ON a.ArtistId = ar.ArtistId
SELECT a.Title, a.ArtistId, ar.Name AS ArtistName
FROM "albums" a
JOIN "artists" ar ON a.ArtistId = ar.ArtistId
                                                 Title  ArtistId  \
0                For For Those About To Rock We Salute You         1
1                                     Balls to the Wall         2
2                                     Restless and Wild         2
3                                     Let There Be Rock         1
4                                              Big Ones         3
..                                                  ...       ...
342                           Respighi:Pines of Rome       226
343  Schubert: The Late String Quartets & String Qu...       272
344                              Monteverdi: L'Orfeo       273
345                           Mozart: Chamber Music       274
346  Koyaanisqatsi (Soundtrack from the Motion Pict...       275
```

```
                                          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 rows x 3 columns]
Ollama parameters:
model=phi3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all albums and their corresponding artist names
\n'\n\nThe DataFrame was produced using this query: SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM
\"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId\n\nThe following is information about the res
ulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Title         object\nArtistId       int64\nArti
stName    object\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to
chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only
one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explana
tions -- just the code."}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:43:02.303652975Z', 'message': {'role': 'assistant',
'content': "import plotly.express as px\n\nfig = px.bar(df, x='ArtistName', y='Title', title='Albums and Ar
tists')\nfig.show()\n\nif df.empty:\n    fig = px.pie(values=[1], labels=['Single Value'], title='Only Sing
le Value')\n    fig.show()"}, 'done_reason': 'stop', 'done': True, 'total_duration': 21617686566, 'load_dur
ation': 43626818, 'prompt_eval_count': 210, 'prompt_eval_duration': 7014937000, 'eval_count': 82, 'eval_dur
ation': 14513956000}
```


Albums and Artists
Title
For Those About To Rock We Salute You
AC/DC
BackBeat
Chico Buarque
Metallica
Deep Purple
Def Leppard
Frank Sinatra
Jamiroquai
Legião Urbana
Motörhead
Page & Plant
Raimundos
Stone Temple Pilots
The Rolling Stones
U2
Battlestar Galactica (Classic)
Karsh Kale
Richard Marlow & The Choir of Trinity College, Cambridge
Academy of St. Martin in the Fields Chamber Ensemble & Sir
London Symphony Orchestra & Sir Charles Mackerras
Ton Koopman
Gustav Mahler
Berliner Philharmoniker & Herbert Von Karajan
Philharmonia Orchestra & Sir Neville Marriner
Kent Nagano and Orchestre de l'Opéra de Lyon
Emerson String Quartet
ArtistName

Out[21]:
```
('SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = a
r.ArtistId',
                                                 Title  ArtistId  \
0                For For Those About To Rock We Salute You         1
1                                     Balls to the Wall         2
2                                     Restless and Wild         2
3                                     Let There Be Rock         1
4                                               Big Ones         3
..                                                   ...       ...
342                             Respighi:Pines of Rome       226
343  Schubert: The Late String Quartets & String Qu...       272
344                               Monteverdi: L'Orfeo       273
345                             Mozart: Chamber Music       274
346  Koyaanisqatsi (Soundtrack from the Motion Pict...       275

                                           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 rows x 3 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},
                'template': '...',
                'title': {'text': 'Albums and Artists'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'ArtistName'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Title'}}}
 }))
```

In [22]:
```
question = """
    Find all tracks with a name containing "What" (case-insensitive)
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\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   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_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 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 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 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\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 sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\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 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\nHAVING COUNT(pt.TrackId) >= 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 "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'}, {'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'}, {'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   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 "arti

sts" 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.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   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\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n   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\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId'}, {'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\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}]
Ollama parameters:
model=phi3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\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   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_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 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 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 AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n   GenreId INTEGER PRIMARY K

EY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\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 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 * \nFROM \"tracks\" \n
WHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": " \n   Get all playlists containing at le
ast 10 tracks and the total duration of those tracks:\n"}, {"role": "assistant", "content": "SELECT pt.Play
listId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM \"playlist_track\" pt\nJOIN \"pl
aylists\" p ON pt.PlaylistId = p.PlaylistId\nJOIN \"tracks\" t ON pt.TrackId = t.TrackId\nGROUP BY pt.Playl
istId, p.Name\nHAVING COUNT(pt.TrackId) >= 10"}, {"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 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"}, {"role": "user", "co
ntent": " \n   List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "conten
t": "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   Find the top 5 most expensive tracks (base
d on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"trac
ks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Identify artists who ha
ve albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.Art
istId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.Ar
tistId 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   There are 3 tables: artists, albums and tracks, where albums and artist
s are linked by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most popula
r artists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.Track
Id) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t O
N al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "conten
t": " \n   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.Invoic
eId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n   List
all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.
ArtistId, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"ro
le": "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", "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    Find all tracks with a name containing \"What\" (case-insensitive)\n"}]
Add of existing embedding ID: 6f22268c-5062-5f11-ba2d-8555f06b409d-sql
Insert of existing embedding ID: 6f22268c-5062-5f11-ba2d-8555f06b409d-sql
```

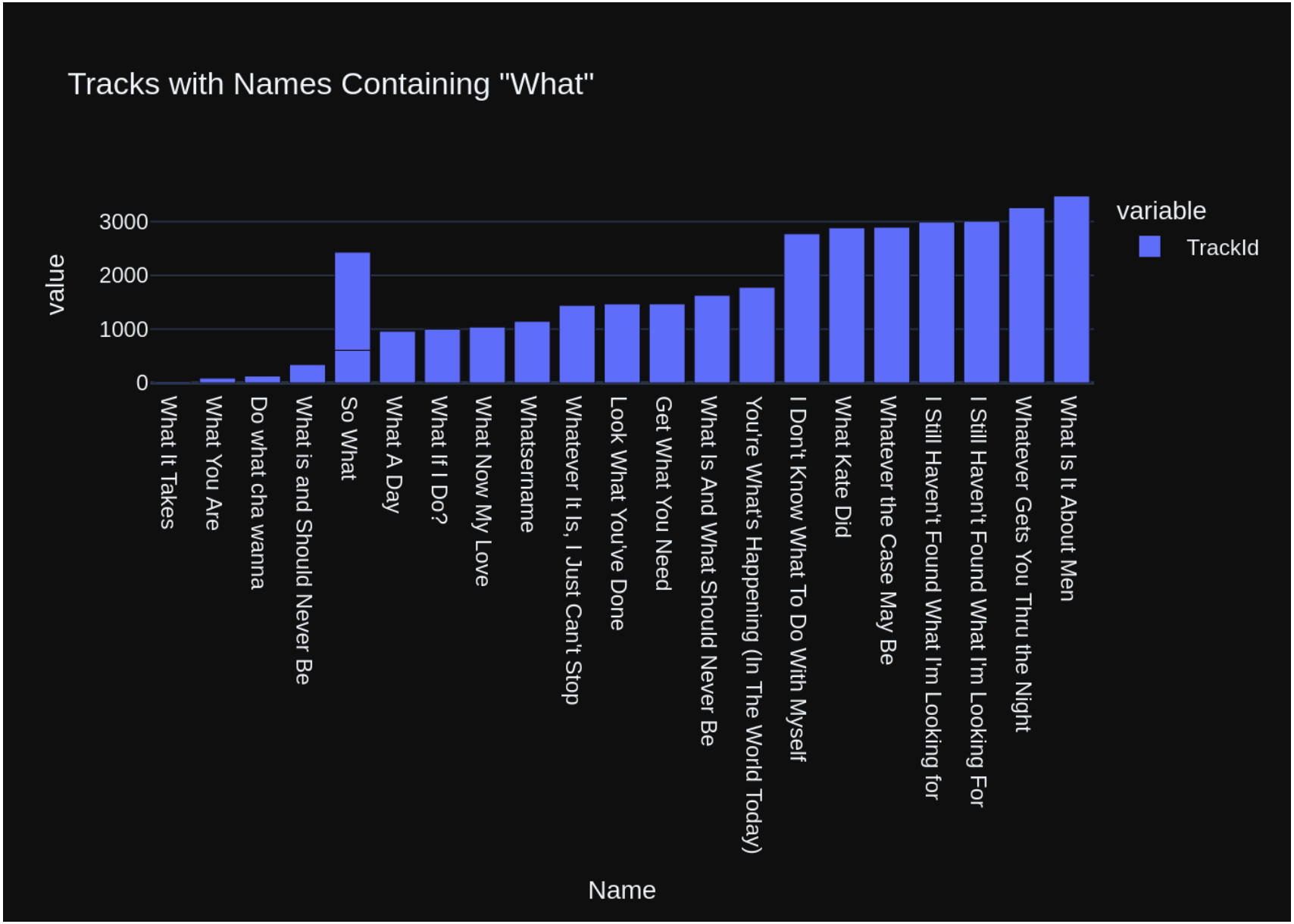
```
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:44:05.427951064Z', 'message': {'role': 'assistant',
'content': 'SELECT * \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%\''}, 'done_reason': 'stop', 'done':
True, 'total_duration': 62969990811, 'load_duration': 656682, 'prompt_eval_count': 1938, 'prompt_eval_durat
ion': 58767036000, 'eval_count': 23, 'eval_duration': 3618252000}
SELECT *
FROM "tracks"
WHERE LOWER(Name) LIKE '%what%'
SELECT *
FROM "tracks"
WHERE LOWER(Name) LIKE '%what%'
    TrackId                                        Name  AlbumId  \
0        26                           What It Takes        5
1        88                            What You Are       10
2       130                       Do what cha wanna       13
3       342             What is and Should Never Be       30
4       607                                So What       48
5       960                             What A Day       76
6      1000                           What If I Do?       80
7      1039                        What Now My Love       83
8      1145                             Whatsername       89
9      1440       Whatever It Is, I Just Can't Stop      116
10     1469                   Look What You've Done      119
11     1470                       Get What You Need      119
12     1628        What Is And What Should Never Be      133
13     1778  You're What's Happening (In The World Today)      146
14     1823                                So What      149
15     2772     I Don't Know What To Do With Myself      223
16     2884                           What Kate Did      231
17     2893                 Whatever the Case May Be      230
18     2992    I Still Haven't Found What I'm Looking for      237
19     3007    I Still Haven't Found What I'm Looking For      238
20     3258         Whatever Gets You Thru the Night      255
21     3475                       What Is It About Men      322

    MediaTypeId  GenreId                                        Composer  \
0             1        1            Steven Tyler, Joe Perry, Desmond Child
1             1        1                        Audioslave/Chris Cornell
2             1        2                                     George Duke
3             1        1                      Jimmy Page/Robert Plant
4             1        2                                     Miles Davis
5             1        1             Mike Bordin, Billy Gould, Mike Patton
```

```
6             1        1  Dave Grohl, Taylor Hawkins, Nate Mendel, Chris...
7             1       12          carl sigman/gilbert becaud/pierre leroyer
8             1        4                                          Green Day
9             1        1                                    Jay Kay/Kay, Jay
10            1        4                                          N. Cester
11            1        4                       C. Cester/C. Muncey/N. Cester
12            1        1                            Jimmy Page, Robert Plant
13            1       14               Allen Story/George Gordy/Robert Gordy
14            1        3                                       Culmer/Exalt
15            1        7                                               None
16            3       19                                               None
17            3       19                                               None
18            1        1      Bono/Clayton, Adam/Mullen Jr., Larry/The Edge
19            1        1                                                 U2
20            2        9                                               None
21            2        9  Delroy "Chris" Cooper, Donovan Jackson, Earl C...

    Milliseconds      Bytes  UnitPrice
0         310622   10144730       0.99
1         249391    5988186       0.99
2         274155    9018565       0.99
3         260675    8497116       0.99
4         564009   18360449       0.99
5         158275    5203430       0.99
6         302994    9929799       0.99
7         149995    4913383       0.99
8         252316    8244843       0.99
9         247222    8249453       0.99
10        230974    7517083       0.99
11        247719    8043765       0.99
12        287973    9369385       0.99
13        142027    4631104       0.99
14        189152    6162894       0.99
15        221387    7251478       0.99
16       2610250  484583988       1.99
17       2616410  183867185       1.99
18        353567   11542247       0.99
19        280764    9306737       0.99
20        215084    3499018       0.99
21        209573    3426106       0.99
Ollama parameters:
model=phi3: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 * \nFROM \"tracks\" \nWHERE LOWER(Nam
e) LIKE '%what%'\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dt
ypes gives:\n TrackId           int64\nName             object\nAlbumId           int64\nMediaTypeId
int64\nGenreId           int64\nComposer         object\nMilliseconds      int64\nBytes             int64\n
UnitPrice       float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly c
ode to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there i
s 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': 'phi3:latest', 'created_at': '2024-06-14T02:44:24.043801939Z', 'message': {'role': 'assistant',
'content': '```python\nimport plotly.express as px\n\n# Assuming that df contains at least a column named
\'Name\' to visualize based on the query result\nfig = px.bar(df, x=\'Name\', y=[\'TrackId\'], title=\'Trac
ks with Names Containing "What"\')\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duratio
n': 18586993197, 'load_duration': 44760805, 'prompt_eval_count': 238, 'prompt_eval_duration': 7004063000,
'eval_count': 74, 'eval_duration': 11493020000}
```


Tracks with Names Containing "What"
value
3000
2000
1000
0
variable
TrackId
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)
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
Name

Out[22]: ('SELECT * \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%\'',

	TrackId	Name	AlbumId	MediaTypeId	GenreId	Composer
0	26	What It Takes	5	1	1	Steven Tyler, Joe Perry, Desmond Child
1	88	What You Are	10	1	1	Audioslave/Chris Cornell
2	130	Do what cha wanna	13	1	2	George Duke
3	342	What is and Should Never Be	30	1	1	Jimmy Page/Robert Plant
4	607	So What	48	1	2	Miles Davis
5	960	What A Day	76	1	1	Mike Bordin, Billy Gould, Mike Patton
6	1000	What If I Do?	80	1	1	Dave Grohl, Taylor Hawkins, Nate Mendel, Chris...
7	1039	What Now My Love	83	1	12	carl sigman/gilbert becaud/pierre leroyer
8	1145	Whatsername	89	1	4	Green Day
9	1440	Whatever It Is, I Just Can't Stop	116	1	1	Jay Kay/Kay, Jay
10	1469	Look What You've Done	119	1	4	N. Cester
11	1470	Get What You Need	119	1	4	C. Cester/C. Muncey/N. Cester
12	1628	What Is And What Should Never Be	133	1	1	Jimmy Page, Robert Plant
13	1778	You're What's Happening (In The World Today)	146	1	14	Allen Story/George Gordy/Robert Gordy
14	1823	So What	149	1	3	Culmer/Exalt
15	2772	I Don't Know What To Do With Myself	223	1	7	None
16	2884	What Kate Did	231			
17	2893	Whatever the Case May Be	230			
18	2992	I Still Haven't Found What I'm Looking for	237			
19	3007	I Still Haven't Found What I'm Looking For	238			
20	3258	Whatever Gets You Thru the Night	255			
21	3475	What Is It About Men	322			

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

    Milliseconds      Bytes  UnitPrice
0         310622   10144730       0.99
1         249391    5988186       0.99
2         274155    9018565       0.99
3         260675    8497116       0.99
4         564009   18360449       0.99
5         158275    5203430       0.99
6         302994    9929799       0.99
7         149995    4913383       0.99
8         252316    8244843       0.99
9         247222    8249453       0.99
10        230974    7517083       0.99
11        247719    8043765       0.99
12        287973    9369385       0.99
13        142027    4631104       0.99
14        189152    6162894       0.99
15        221387    7251478       0.99
16       2610250  484583988       1.99
17       2616410  183867185       1.99
18        353567   11542247       0.99
19        280764    9306737       0.99
20        215084    3499018       0.99
21        209573    3426106       0.99  ,
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'variable=TrackId<br>Name=%{x}<br>value=%{y}<extra></extra>',
              'legendgroup': 'TrackId',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': 'TrackId',
              'offsetgroup': 'TrackId',
              'orientation': 'v',
              'showlegend': True,
              'textposition': 'auto',
              'type': 'bar',
              'x': array(['What It Takes', 'What You Are', 'Do what cha wanna',
```

```
                                'What is and Should Never Be', 'So What', 'What A Day', 'What If I Do?',
                                'What Now My Love', 'Whatsername', "Whatever It Is, I Just Can't Stop",
                                "Look What You've Done", 'Get What You Need',
                                'What Is And What Should Never Be',
                                "You're What's Happening (In The World Today)", 'So What',
                                "I Don't Know What To Do With Myself", 'What Kate Did',
                                'Whatever the Case May Be',
                                "I Still Haven't Found What I'm Looking for",
                                "I Still Haven't Found What I'm Looking For",
                                'Whatever Gets You Thru the Night', 'What Is It About Men'],
                               dtype=object),
                  'xaxis': 'x',
                  'y': array([  26,   88,  130,  342,  607,  960, 1000, 1039, 1145, 1440, 1469, 1470,
                              1628, 1778, 1823, 2772, 2884, 2893, 2992, 3007, 3258, 3475]),
                  'yaxis': 'y'}],
      'layout': {'barmode': 'relative',
                 'legend': {'title': {'text': 'variable'}, 'tracegroupgap': 0},
                 'template': '...',
                 'title': {'text': 'Tracks with Names Containing "What"'},
                 'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                 'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'value'}}}
 }))
```

In [23]:

```
question = """
    Get the total number of invoices for each customer
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "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   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) 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_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 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) 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 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 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\n   FOREIGN 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 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(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_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 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\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 sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Get 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.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'use

r', 'content': ' \n   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.Custome
rId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'conten
t': ' \n   Get the average invoice 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.CustomerId'}, {'role': 'user', 'content': ' \n   Find the total number of invoic
es per country:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\n
FROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': ' \n   Find the customer who
bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SEL
ECT 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 TotalA
lbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amou
nt invoiced:\n'}, {'role': 'assistant', '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 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': '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'}, {'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 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.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
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 "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   Get the total number of invoices for each customer
\n'}]
Ollama parameters:
model=phi3: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   BillingCou

ntry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FORE
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
\n   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
\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_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
L,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Byt
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\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   Get
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 customer with the most invoices
\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\nORDER BY TotalInvoi
ces DESC\nLIMIT 1"}, {"role": "user", "content": " \n   Get the average invoice total for each custome
r:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM

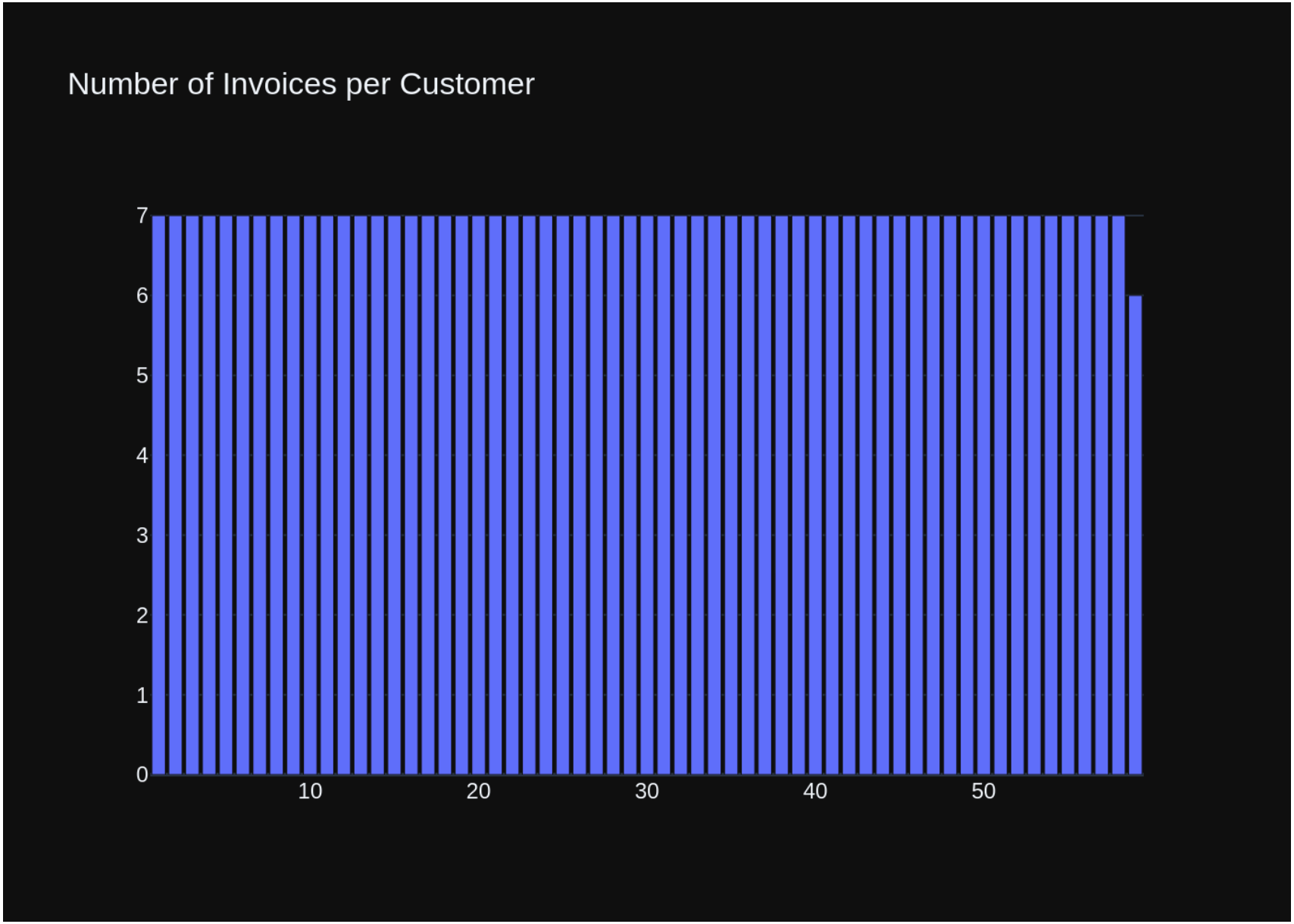
\"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "use
r", "content": " \n   Find the total number of invoices per country:\n"}, {"role": "assistant", "conten
t": "SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"},
{"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.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user",
"content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant",
"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 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"}, {"role": "user", "content": " \n   Hint: album quan
tity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total qu
antity (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_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   \n   Find the top 5
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_
items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"ro
le": "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", "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"}, {"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': 'phi3:latest', 'created_at': '2024-06-14T02:45:33.534043463Z', 'message': {'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;'}, 'done_reason': 'stop', 'done': True, 'total_durat
ion': 69377181955, 'load_duration': 734671, 'prompt_eval_count': 1952, 'prompt_eval_duration': 59862022000,
'eval_count': 54, 'eval_duration': 8625404000}
SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId;
Output from LLM: SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId;
Extracted SQL: 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
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
39          40              7
40          41              7
41          42              7
42          43              7
43          44              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
53          54              7
54          55              7
55          56              7
56          57              7
57          58              7
58          59              6
Ollama parameters:
model=phi3: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 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': 'phi3:latest', 'created_at': '2024-06-14T02:45:51.301740474Z', 'message': {'role': 'assistant',
'content': "import plotly.graph_objects as go\n\nfig = go.Figure()\nfig.add_trace(go.Bar(x=df['CustomerI
d'], y=df['TotalInvoices']))\nfig.update_layout(title='Number of Invoices per Customer')\nfig.show()"}, 'do
ne_reason': 'stop', 'done': True, 'total_duration': 17741480952, 'load_duration': 41389443, 'prompt_eval_co
unt': 217, 'prompt_eval_duration': 6784682000, 'eval_count': 68, 'eval_duration': 10870475000}
```


Number of Invoices per Customer
7
6
5
4
3
2
1
0
10
20
30
40
50

Out[23]: ('SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.Cus
tomerId = i.CustomerId\nGROUP BY c.CustomerId',

	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

```
39          40              7
40          41              7
41          42              7
42          43              7
43          44              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
53          54              7
54          55              7
55          56              7
56          57              7
57          58              7
58          59              6,
Figure({
    'data': [{'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]),
              'y': array([7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
                          7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
                          7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 6])}],
    'layout': {'template': '...', 'title': {'text': 'Number of Invoices per Customer'}}
}))
```

In [24]:

```
question = """
    Find the total number of invoices per country:
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "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   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) 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_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   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) REFERENCES "tracks" (TrackId) \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 INDEX IFK_InvoiceLineTrackId 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 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\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 "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 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 "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 "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   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_EmployeeReportsTo ON "employees" (ReportsTo)\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 sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per countr

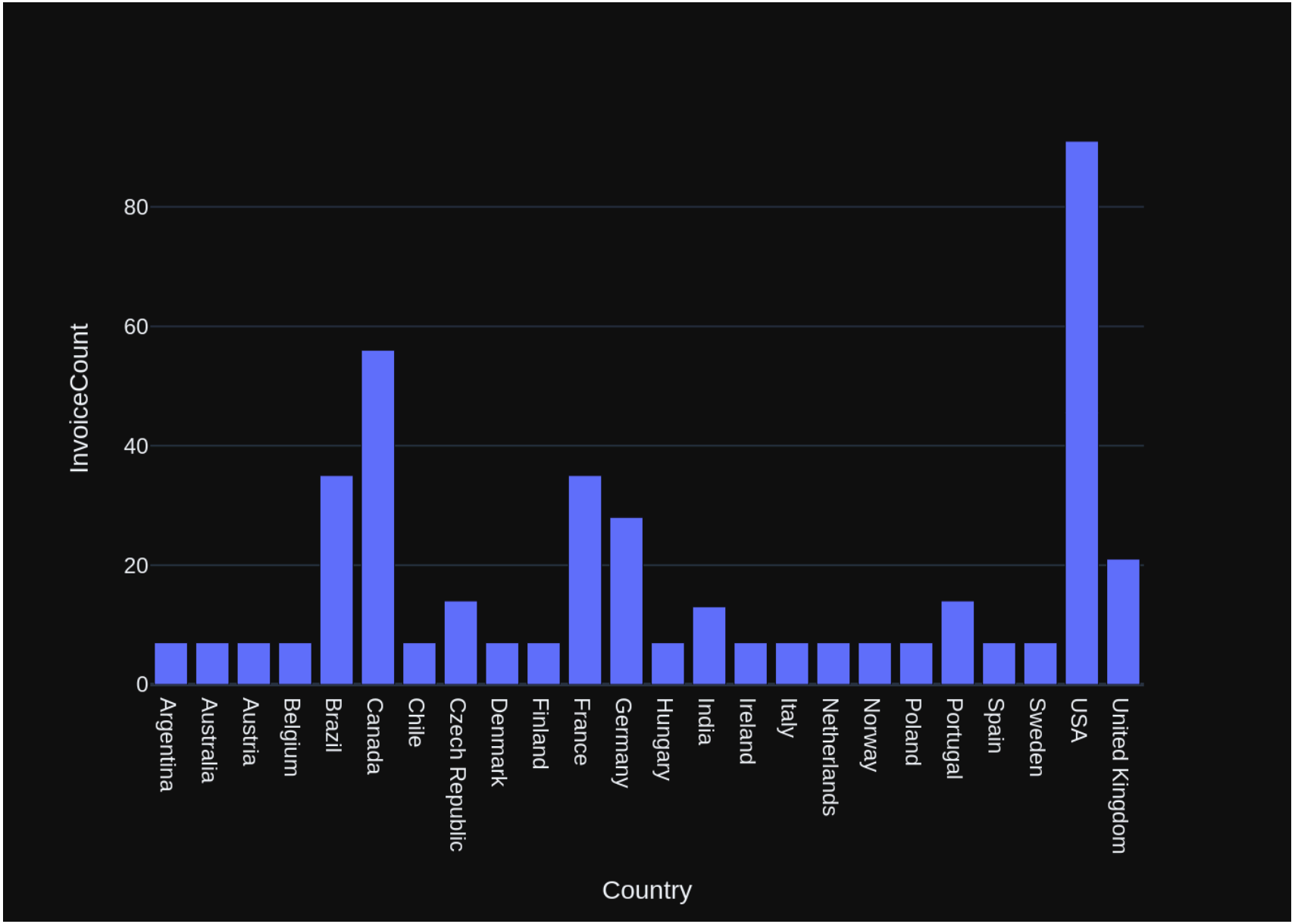
y:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', '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 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': '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 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   \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\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM "invoices" \nWHERE Total > 10.00'}, {'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\nGROUP BY c.CustomerId\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, 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 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\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per countr
y:\n'}]
Ollama parameters:
model=phi3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE \"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   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   FORE
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
T NULL,\r\n   Quantity 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_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   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 D
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   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARC
HAR(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 (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
MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds I
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 (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\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 i.BillingCountry, COUNT(*) AS
TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "content": " \n   Get t
he total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, C
OUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerI

d\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "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 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": "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 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   \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\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM \"invoices\" \nWHERE Total > 10.00"}, {"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\nGROUP BY c.CustomerId\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, 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 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\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:47:03.00009704Z', 'message': {'role': 'assistant', 'content': 'SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM "customers" customer\nJOIN "invoices" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country;'}, 'done_reason': 'stop', 'done': True, 'total_duration': 71592818186, 'load_duration': 647105, 'prompt_eval_count': 1987, 'prompt_eval_duration': 61680612000, 'eval_count': 52, 'eval_duration': 8996605000}
SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount
FROM "customers" customer
JOIN "invoices" invoice ON customer.CustomerId = invoice.CustomerId
GROUP BY Country;
Output from LLM: SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount
FROM "customers" customer
JOIN "invoices" invoice ON customer.CustomerId = invoice.CustomerId

```
GROUP BY Country;
Extracted SQL: SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount
FROM "customers" customer
JOIN "invoices" invoice ON customer.CustomerId = invoice.CustomerId
GROUP BY Country
SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount
FROM "customers" customer
JOIN "invoices" invoice ON customer.CustomerId = invoice.CustomerId
GROUP BY Country
           Country  InvoiceCount
0        Argentina             7
1        Australia             7
2          Austria             7
3          Belgium             7
4           Brazil            35
5           Canada            56
6            Chile             7
7   Czech Republic            14
8          Denmark             7
9          Finland             7
10          France            35
11         Germany            28
12         Hungary             7
13           India            13
14         Ireland             7
15           Italy             7
16     Netherlands             7
17          Norway             7
18          Poland             7
19        Portugal            14
20           Spain             7
21          Sweden             7
22             USA            91
23  United Kingdom            21
Ollama parameters:
model=phi3: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 Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount
```

\nFROM \"customers\" customer\nJOIN \"invoices\" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Country   object\nInvoiceCount   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 Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:47:22.873100357Z', 'message': {'role': 'assistant', 'content': "import plotly.express as px\n\nfig = px.bar(df, x='Country', y='InvoiceCount')\nfig.show()\n\nif len(df) == 1:\n   fig = px.pie(df, values=['InvoiceCount'], names=['Country'])\n   fig.show()"}, 'done_reason': 'stop', 'done': True, 'total_duration': 19845616463, 'load_duration': 46145478, 'prompt_eval_count': 211, 'prompt_eval_duration': 6955079000, 'eval_count': 75, 'eval_duration': 12799524000}


InvoiceCount
0
20
40
60
80
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
Country

Out[24]: ('SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM "customers" customer\nJOIN "invoices" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country',

	Country	InvoiceCount
0	Argentina	7
1	Australia	7
2	Austria	7
3	Belgium	7
4	Brazil	35
5	Canada	56
6	Chile	7
7	Czech Republic	14
8	Denmark	7
9	Finland	7
10	France	35
11	Germany	28
12	Hungary	7
13	India	13
14	Ireland	7
15	Italy	7
16	Netherlands	7
17	Norway	7
18	Poland	7
19	Portugal	14
20	Spain	7
21	Sweden	7
22	USA	91
23	United Kingdom	21,

```
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'Country=%{x}<br>InvoiceCount=%{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,
                              7, 14,  7,  7, 91, 21]),
                 'yaxis': 'y'}],
      'layout': {'barmode': 'relative',
                 'legend': {'tracegroupgap': 0},
                 'margin': {'t': 60},
                 'template': '...',
                 'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Country'}},
                 'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceCount'}}}
  }))
```

In [25]:
```
question = """
    List all invoices with a total exceeding $10:
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoice_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   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) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId 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   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) 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 INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\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   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_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   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 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 "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   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),\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\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 sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM "invoices" \nWHERE Total > 10.00'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'conten

t': '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 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.Cust
omerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Hint: album quantity is f
ound in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (a
cross all invoices):\n'}, {'role': '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.CustomerI
d\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 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.Invoic
eId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'conten
t': ' \n   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 i.CustomerId, COUNT(ii.T
rackId) 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   Get the total numb
er of invoices for each customer\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'}, {'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 total number of invoices per country:\n'}, {'role': 'assistan
t', 'content': 'SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.BillingCo
untry'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}]
Ollama parameters:
model=phi3: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
EY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   Uni
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
NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(4
0),\r\n   BillingCountry 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   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 (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   Last
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
\n   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
AR(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),\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\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 query 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   List all i
nvoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM \"invoices\" \nW
HERE Total > 10.00"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amoun
t invoiced:\n"}, {"role": "assistant", "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 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": "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"}, {"role": "use
r", "content": " \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customer

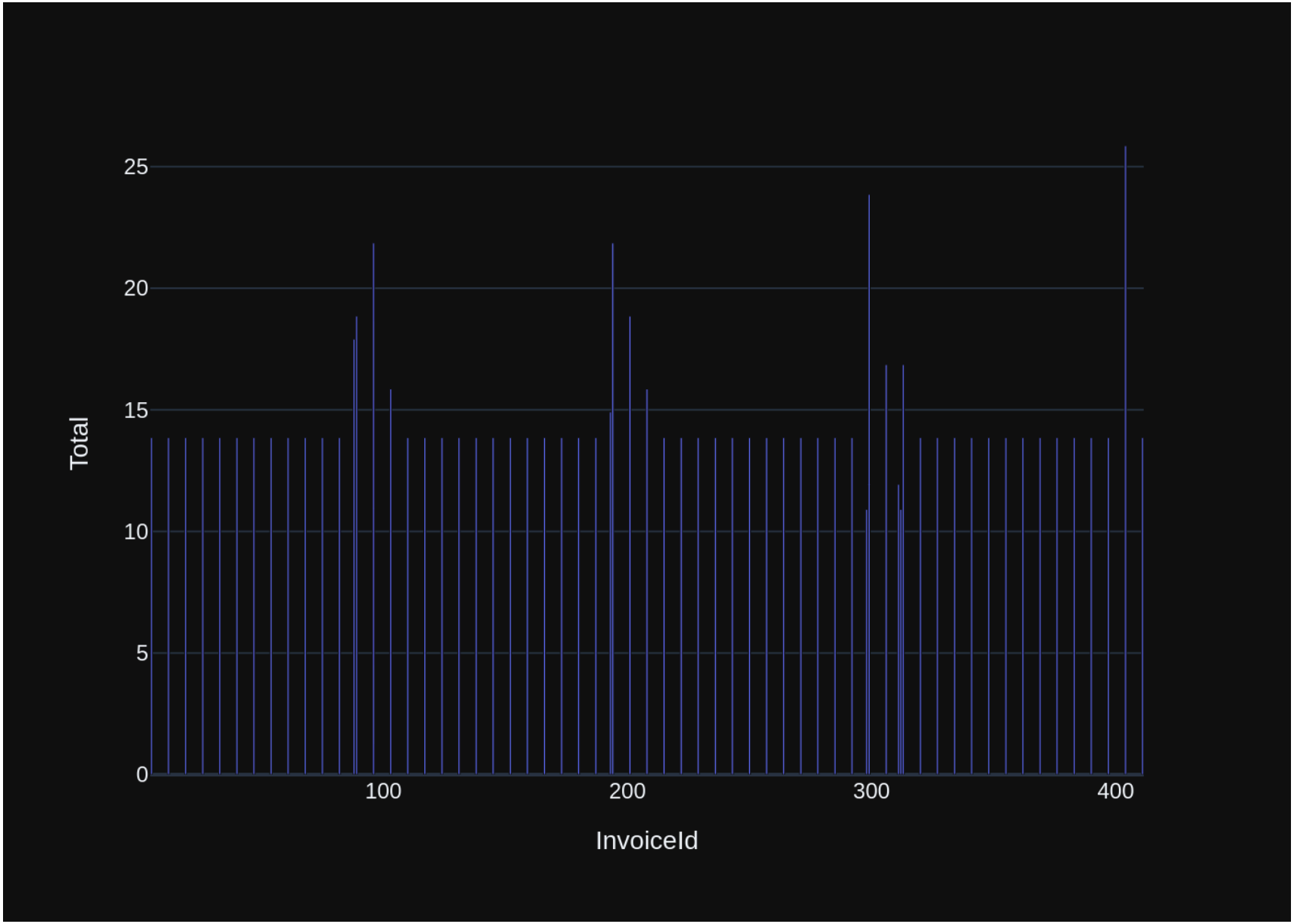
s who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Hint: album quantity is found in invoice_items, \n   \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\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 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 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 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 5"}, {"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 \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"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": "user", "content": " \n   Find the total number 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   List all invoices with a total exceeding $10:\n"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:48:29.210491467Z', 'message': {'role': 'assistant', 'content': 'SELECT *\nFROM "invoices" \nWHERE (Total > 10)\nGROUP BY InvoiceId;'}, 'done_reason': 'stop', 'done': True, 'total_duration': 66235136309, 'load_duration': 682874, 'prompt_eval_count': 1966, 'prompt_eval_duration': 60972001000, 'eval_count': 27, 'eval_duration': 4441250000}
SELECT *
FROM "invoices"
WHERE (Total > 10)
GROUP BY InvoiceId;
Output from LLM: SELECT *
FROM "invoices"
WHERE (Total > 10)
GROUP BY InvoiceId;
Extracted SQL: SELECT *
FROM "invoices"
WHERE (Total > 10)
GROUP BY InvoiceId
SELECT *

```
FROM "invoices"
WHERE (Total > 10)
GROUP BY InvoiceId
    InvoiceId  CustomerId          InvoiceDate             BillingAddress  \
0           5          23  2009-01-11 00:00:00            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
..        ...         ...                  ...                        ...
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
63        411          44  2013-12-14 00:00:00            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           VV     Netherlands              1016  13.86
61      Tucson           AZ             USA             85719  13.86
62      Prague         None  Czech Republic             14300  25.86
63    Helsinki         None         Finland             00530  13.86

[64 rows x 9 columns]
Ollama parameters:
model=phi3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all invoices with a total exceeding $10:\n'\n\nThe
DataFrame was produced using this query: SELECT *\nFROM \"invoices\" \nWHERE (Total > 10)\nGROUP BY Invoice
Id\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n
InvoiceId              int64\nCustomerId             int64\nInvoiceDate          object\nBillingAddress
object\nBillingCity          object\nBillingState          object\nBillingCountry        object\nBillingPo
stalCode     object\nTotal               float64\ndtype: object"}, {"role": "user", "content": "Can you ge
```

```
nerate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas datafra
me called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python cod
e. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:48:47.303291729Z', 'message': {'role': 'assistant',
'content': "import plotly.express as px\n\nfig = px.bar(df, x='InvoiceId', y='Total')\nfig.show()\n\nif df.
shape[0] == 1:\n    fig = px.indicator(values=[df['Total']])\n    fig.show()"}, 'done_reason': 'stop', 'don
e': True, 'total_duration': 18065039154, 'load_duration': 47353642, 'prompt_eval_count': 236, 'prompt_eval_
duration': 6798174000, 'eval_count': 71, 'eval_duration': 11124594000}
```


25
20
15
10
5
0
Total
100
200
300
400
InvoiceId

Out[25]:
```
('SELECT *\nFROM "invoices" \nWHERE (Total > 10)\nGROUP BY InvoiceId',
    InvoiceId  CustomerId          InvoiceDate             BillingAddress  \
0           5          23  2009-01-11 00:00:00            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
..        ...         ...                  ...                        ...
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
63        411          44  2013-12-14 00:00:00            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           VV     Netherlands              1016  13.86
61      Tucson           AZ             USA             85719  13.86
62      Prague         None  Czech Republic             14300  25.86
63    Helsinki         None         Finland             00530  13.86

[64 rows x 9 columns],
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'InvoiceId=%{x}<br>Total=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'v',
              'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
              'x': array([  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, 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, 13.86, 25.86, 13.86]),
                'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total'}}}
})
```

In [26]:
```
question = """
    Find all invoices since 2010 and the total amount invoiced:
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "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   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) 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_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   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) REFERENCES "tracks" (TrackId) \r\n\t\tON 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_InvoiceLineTrackId 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 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\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 "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 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 "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   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"\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_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\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 sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate

_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please 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.InvoiceDate, SUM(i.Total)
AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'rol
e': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'c
ontent': 'SELECT *\nFROM "invoices" \nWHERE (Total > 10)\nGROUP BY InvoiceId'}, {'role': 'user', 'content':
' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFR
OM "invoices" \nWHERE Total > 10.00'}, {'role': 'user', 'content': ' \n   Find the total number of invoic
es per country:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\n
FROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': ' \n   Find the total number
of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT Customer.Country, COUNT(invoice.Invo
iceId) AS InvoiceCount\nFROM "customers" customer\nJOIN "invoices" invoice ON customer.CustomerId = invoic
e.CustomerId\nGROUP BY Country'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for
each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoic
es\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 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 customer who bought the most albums in total quantity
(across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS To
talAlbums\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'}, {'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.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', 'co
ntent': ' \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': 'user', 'content': '
\n   Find all invoices since 2010 and the total amount invoiced:\n'}]
Ollama parameters:
model=phi3: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   BillingCou

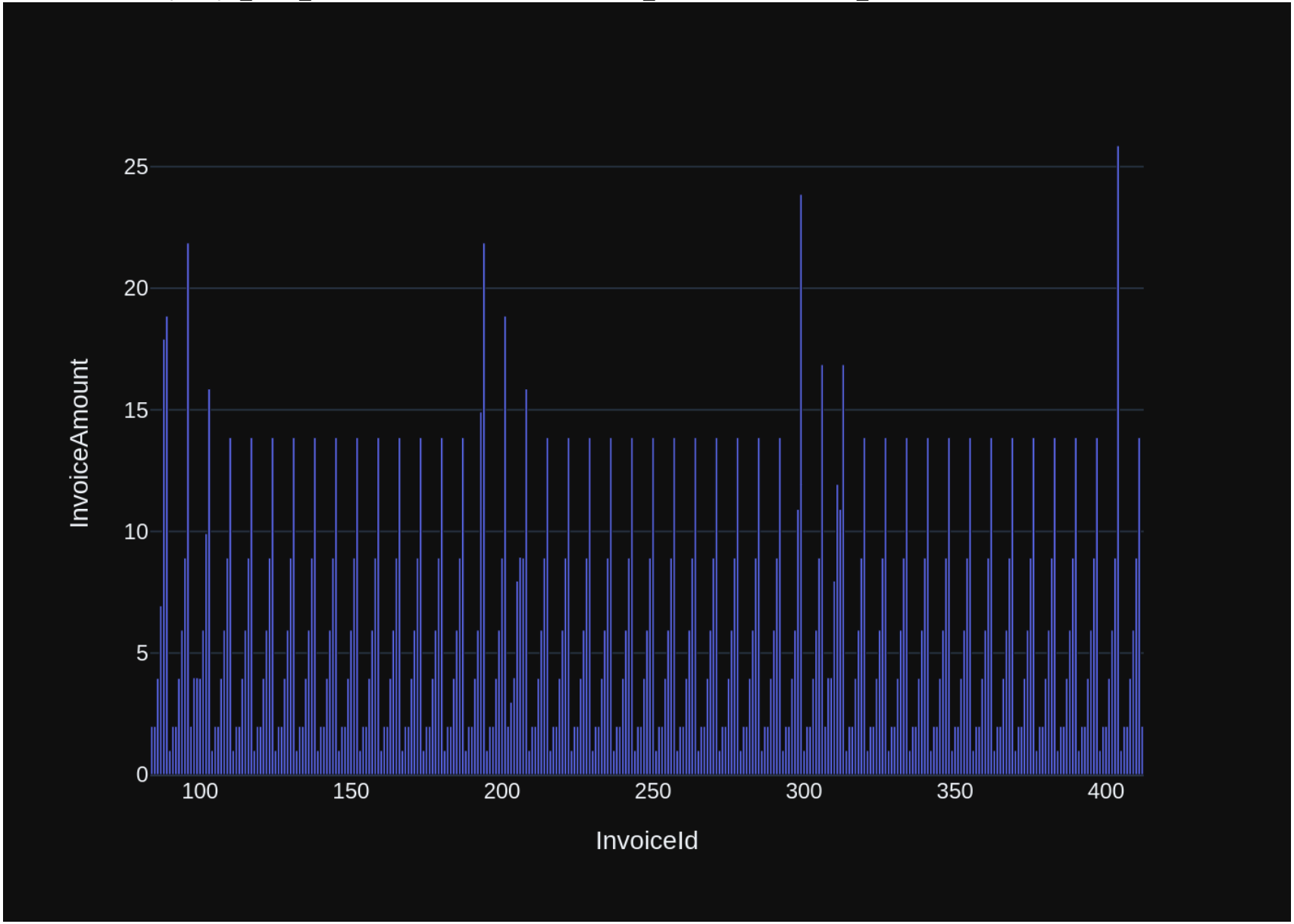
ntry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FORE
IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NO
T NULL,\r\n   Quantity 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 \"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
\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON D
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   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARC
HAR(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 (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
oser NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\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\tON DELETE NO ACTION
ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON D
ELETE 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   FOR
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
ULL,\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\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 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":
" \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content":
"SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-0

1'\nGROUP BY i.InvoiceDate"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT *\nFROM \"invoices\" \nWHERE (Total > 10)\nGROUP BY InvoiceId"}, {"role": "user", "content": " \n   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 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", "content": "SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM \"customers\" customer\nJOIN \"invoices\" invoice 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.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 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 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\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 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.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": "user", "content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:50:00.137712Z', 'message': {'role': 'assistant', 'content': 'SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount\nFROM "invoices" i\nWHERE strftime(\'%Y\', i.InvoiceDate) >= \'2010\'\nGROUP BY i.InvoiceId\nORDER BY InvoiceAmount DESC;'}, 'done_reason': 'stop', 'done': True, 'total_duration': 72733332693, 'load_duration': 712770, 'prompt_eval_count': 2018, 'prompt_eval_duration': 60910143000, 'eval_count': 64, 'eval_duration': 10840683000}
SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount
FROM "invoices" i
WHERE strftime('%Y', i.InvoiceDate) >= '2010'
GROUP BY i.InvoiceId
ORDER BY InvoiceAmount DESC;
Output from LLM: SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount
FROM "invoices" i
WHERE strftime('%Y', i.InvoiceDate) >= '2010'

```
GROUP BY i.InvoiceId
ORDER BY InvoiceAmount DESC;
Extracted SQL: SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount
FROM "invoices" i
WHERE strftime('%Y', i.InvoiceDate) >= '2010'
GROUP BY i.InvoiceId
ORDER BY InvoiceAmount DESC
SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount
FROM "invoices" i
WHERE strftime('%Y', i.InvoiceDate) >= '2010'
GROUP BY i.InvoiceId
ORDER BY InvoiceAmount DESC
     InvoiceId  InvoiceAmount
0          404          25.86
1          299          23.86
2           96          21.86
3          194          21.86
4           89          18.86
..         ...            ...
324        377           0.99
325        384           0.99
326        391           0.99
327        398           0.99
328        405           0.99

[329 rows x 2 columns]
Ollama parameters:
model=phi3: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'\n\nThe DataFrame was produced using this query: SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount\nF
ROM \"invoices\" i\nWHERE strftime('%Y', i.InvoiceDate) >= '2010'\nGROUP BY i.InvoiceId\nORDER BY InvoiceAm
ount DESC\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gi
ves:\n InvoiceId          int64\nInvoiceAmount    float64\ndtype: object"}, {"role": "user", "content": "Ca
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': 'phi3:latest', 'created_at': '2024-06-14T02:50:17.824092557Z', 'message': {'role': 'assistant',
```

'content': "import plotly.express as px\n\nfig = px.bar(df, x='InvoiceId', y='InvoiceAmount')\nfig.show()\n\nif df.shape[0] == 1:\n   fig = px.pie(df, values='InvoiceAmount', names=['Total'])\n   fig.show()"}, 'done_reason': 'stop', 'done': True, 'total_duration': 17659710113, 'load_duration': 41233884, 'prompt_eval_count': 235, 'prompt_eval_duration': 6683685000, 'eval_count': 78, 'eval_duration': 10885899000}



Out[26]:
```
('SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount\nFROM "invoices" i\nWHERE strftime(\'%Y\', i.InvoiceDa
te) >= \'2010\'\nGROUP BY i.InvoiceId\nORDER BY InvoiceAmount DESC',
      InvoiceId  InvoiceAmount
 0          404          25.86
 1          299          23.86
 2           96          21.86
 3          194          21.86
 4           89          18.86
 ..         ...            ...
 324        377           0.99
 325        384           0.99
 326        391           0.99
 327        398           0.99
 328        405           0.99

 [329 rows x 2 columns],
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'InvoiceId=%{x}<br>InvoiceAmount=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([404, 299,  96, ..., 391, 398, 405]),
               'xaxis': 'x',
               'y': array([25.86, 23.86, 21.86, ...,  0.99,  0.99,  0.99]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceAmount'}}}
 }))
```

In [27]:
```
question = """
    List all employees and their reporting manager's name (if any):
"""
```

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

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\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   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),\r\n   FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \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   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 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 INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\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   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) 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_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   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) 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\n   Name NVARCHAR(120)\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   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"\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 sqlite_stat1(tbl,idx,stat)\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 sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': " \n   List all

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   Find the top
5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices tabl
e, calculation using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELE
CT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Cust
omerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   F
ind 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': 'user', 'content': 'what are the top 5 co
untries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(*) AS TotalC
ustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'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   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'conten
t': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.C
ustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n   Find all invoices s
ince 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM
(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDat
e'}, {'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\nWHE
RE strftime(\'%Y\', i.InvoiceDate) >= \'2010\'\nGROUP BY i.InvoiceId\nORDER BY InvoiceAmount DESC'}, {'rol
e': 'user', 'content': ' \n   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\nF
ROM "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'}, {'role': 'user', 'content': '
\n   Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.Cust
omerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Cust
omerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': " \n   List all employees and their reportin
g manager's name (if any):\n"}]
Ollama parameters:
model=phi3: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
TABLE \"employees\"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVAR
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
AR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\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
0),\r\n   Phone 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\tON DELETE NO
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   B
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
tems\"\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   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) 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\n   Name NVARCHAR(120)\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 INT
EGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NUL
L,\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) REFERENC
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 SQLite 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 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":
" \n   List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "conten
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": "
\n   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": "use

r", "content": " \n   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
ntent": "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 DE
SC\nLIMIT 5"}, {"role": "user", "content": " \n   List all customers from Canada and their email address
es:\n"}, {"role": "assistant", "content": "SELECT c.Email, c.Country\nFROM \"customers\" c\nWHERE c.Country
= 'Canada'"}, {"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 \"customer
s\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "conte
nt": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "conte
nt": "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 2010 and the
total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAm
ount\nFROM \"invoices\" i\nWHERE strftime('%Y', i.InvoiceDate) >= '2010'\nGROUP BY i.InvoiceId\nORDER BY In
voiceAmount DESC"}, {"role": "user", "content": " \n   Find the customer who bought the most albums in t
otal 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 \"i
nvoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT
1"}, {"role": "user", "content": " \n   Get the average invoice total for each customer:\n"}, {"role": "a
ssistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJOIN
\"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n
List all employees and their reporting manager's name (if any):\n"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:51:28.465161491Z', 'message': {'role': 'assistant',
'content': 'SELECT e1.*, CONCAT(e2.FirstName, \' \', e2.LastName) AS ManagerName\nFROM "employees" e1\nLEFT
JOIN "employees" e2 ON e1.ReportsTo = e2.EmployeeId\nORDER BY e1.EmployeeId;'}, 'done_reason': 'stop', 'don
e': True, 'total_duration': 70526310908, 'load_duration': 784876, 'prompt_eval_count': 2001, 'prompt_eval_d
uration': 59372945000, 'eval_count': 66, 'eval_duration': 10341982000}
SELECT e1.*, CONCAT(e2.FirstName, ' ', e2.LastName) AS ManagerName
FROM "employees" e1
LEFT JOIN "employees" e2 ON e1.ReportsTo = e2.EmployeeId
ORDER BY e1.EmployeeId;
Output from LLM: SELECT e1.*, CONCAT(e2.FirstName, ' ', e2.LastName) AS ManagerName
FROM "employees" e1
LEFT JOIN "employees" e2 ON e1.ReportsTo = e2.EmployeeId
ORDER BY e1.EmployeeId;
Extracted SQL: SELECT e1.*, CONCAT(e2.FirstName, ' ', e2.LastName) AS ManagerName
FROM "employees" e1
LEFT JOIN "employees" e2 ON e1.ReportsTo = e2.EmployeeId
ORDER BY e1.EmployeeId
SELECT e1.*, CONCAT(e2.FirstName, ' ', e2.LastName) AS ManagerName

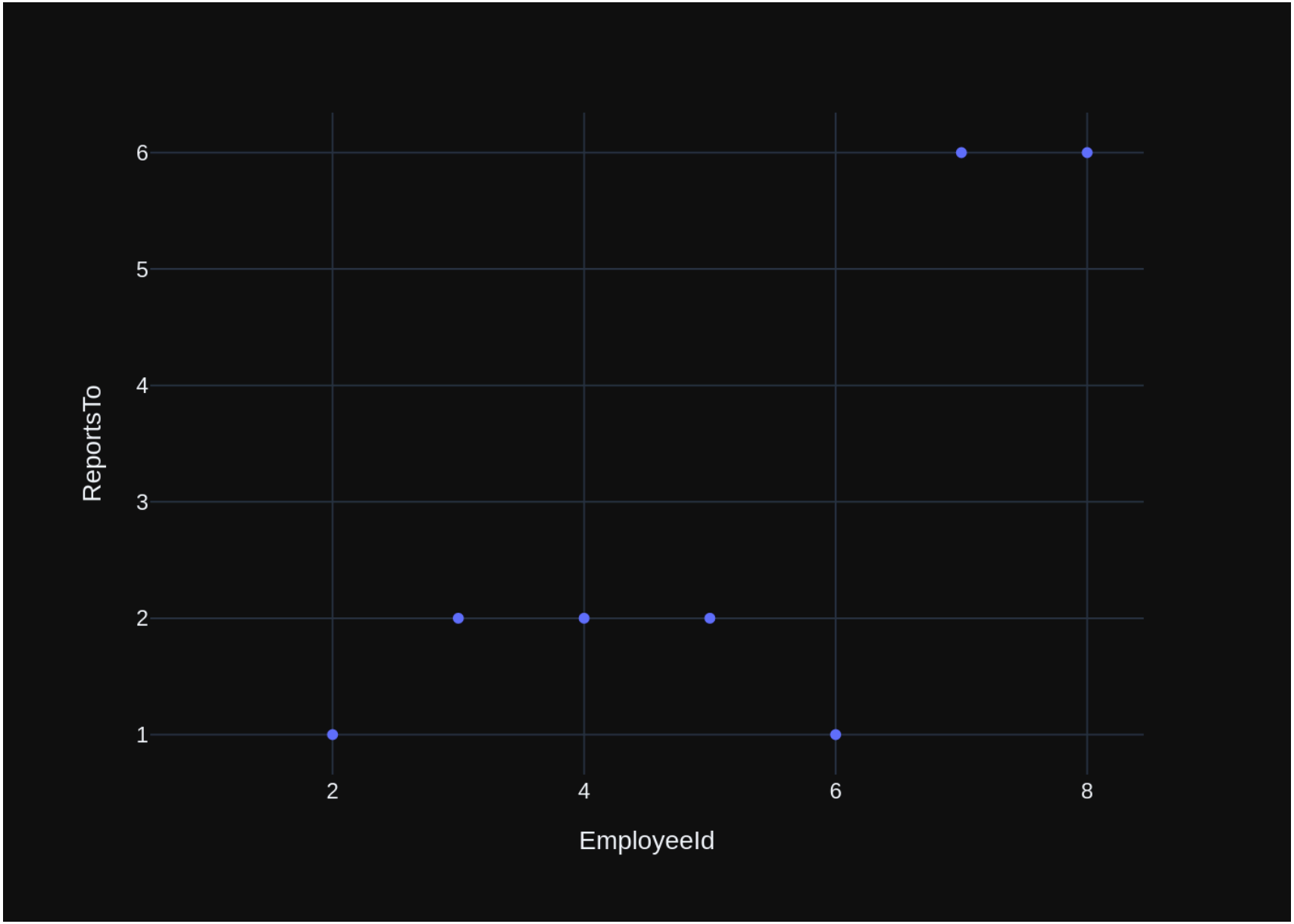
```
FROM "employees" e1
LEFT JOIN "employees" e2 ON e1.ReportsTo = e2.EmployeeId
ORDER BY e1.EmployeeId
   EmployeeId  LastName FirstName                Title  ReportsTo  \
0           1     Adams    Andrew      General Manager        NaN
1           2   Edwards     Nancy        Sales Manager        1.0
2           3   Peacock      Jane  Sales Support Agent        2.0
3           4      Park  Margaret  Sales Support Agent        2.0
4           5   Johnson     Steve  Sales Support Agent        2.0
5           6  Mitchell   Michael           IT Manager        1.0
6           7      King    Robert             IT Staff        6.0
7           8  Callahan     Laura             IT Staff        6.0

             BirthDate             HireDate                     Address  \
0  1962-02-18 00:00:00  2002-08-14 00:00:00          11120 Jasper Ave NW
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
4  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
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     Calgary    AB  Canada    T2P 5M5  +1 (403) 262-3443  +1 (403) 262-6712
3     Calgary    AB  Canada    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     nancy@chinookcorp.com      Andrew Adams
2      jane@chinookcorp.com     Nancy Edwards
3  margaret@chinookcorp.com     Nancy Edwards
4     steve@chinookcorp.com     Nancy Edwards
5   michael@chinookcorp.com      Andrew Adams
6    robert@chinookcorp.com  Michael Mitchell
7     laura@chinookcorp.com  Michael Mitchell
```

```
Ollama parameters:
model=phi3: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'\n\nThe DataFrame was produced using this query: 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.
dtypes gives:\n EmployeeId       int64\nLastName        object\nFirstName       object\nTitle           obj
ect\nReportsTo      float64\nBirthDate       object\nHireDate        object\nAddress         object\nCity
object\nState           object\nCountry         object\nPostalCode      object\nPhone           object\nFax
object\nEmail           object\nManagerName     object\ndtype: object"}, {"role": "user", "content": "Can y
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': 'phi3:latest', 'created_at': '2024-06-14T02:51:54.197047431Z', 'message': {'role': 'assistant',
'content': "import plotly.graph_objects as go\n\nfig = go.Figure()\n\nfor column in df.columns:\n    if df
[column].nunique() > 1:\n        fig.add_trace(go.Bar(x=df['EmployeeId'], y=df[column]))\n    else:\n
fig.add_indicator(value=df[column][0])\n\nfig.update_layout(title='Employees and Their Managers')\nfig.show
()"}, 'done_reason': 'stop', 'done': True, 'total_duration': 25704966494, 'load_duration': 636586, 'prompt_
eval_count': 296, 'prompt_eval_duration': 8603090000, 'eval_count': 111, 'eval_duration': 17011493000}
```


6
5
4
3
2
1
ReportsTo
2
4
6
8
EmployeeId

Out[27]:
```
('SELECT e1.*, CONCAT(e2.FirstName, \' \', e2.LastName) AS ManagerName\nFROM "employees" e1\nLEFT JOIN "em
ployees" e2 ON e1.ReportsTo = e2.EmployeeId\nORDER BY e1.EmployeeId',
    EmployeeId  LastName FirstName                Title  ReportsTo  \
 0           1     Adams    Andrew      General Manager        NaN
 1           2   Edwards     Nancy        Sales Manager        1.0
 2           3   Peacock      Jane  Sales Support Agent        2.0
 3           4      Park  Margaret  Sales Support Agent        2.0
 4           5   Johnson     Steve  Sales Support Agent        2.0
 5           6  Mitchell   Michael           IT Manager        1.0
 6           7      King    Robert             IT Staff        6.0
 7           8  Callahan     Laura             IT Staff        6.0

              BirthDate             HireDate                     Address  \
 0  1962-02-18 00:00:00  2002-08-14 00:00:00          11120 Jasper Ave NW
 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
 4  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
 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     Calgary    AB  Canada    T2P 5M5  +1 (403) 262-3443  +1 (403) 262-6712
 3     Calgary    AB  Canada    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     nancy@chinookcorp.com      Andrew Adams
 2      jane@chinookcorp.com     Nancy Edwards
 3  margaret@chinookcorp.com     Nancy Edwards
 4     steve@chinookcorp.com     Nancy Edwards
 5   michael@chinookcorp.com      Andrew Adams
 6    robert@chinookcorp.com  Michael Mitchell
 7     laura@chinookcorp.com  Michael Mitchell  ,
 Figure({
```

```
        'data': [{'hovertemplate': 'EmployeeId=%{x}<br>ReportsTo=%{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]),
                  'xaxis': 'x',
                  'y': array([nan,  1.,  2.,  2.,  2.,  1.,  6.,  6.]),
                  'yaxis': 'y'}],
        'layout': {'legend': {'tracegroupgap': 0},
                   'margin': {'t': 60},
                   'template': '...',
                   'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'EmployeeId'}},
                   'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'ReportsTo'}}}
    }))
```

In [28]:

```
question = """
    Get the average invoice total for each customer:
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "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   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) 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_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 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) 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 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   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 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 INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\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   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),\r\n   FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \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 query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Get the average invoice 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.CustomerId'}, {'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 "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 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\nG

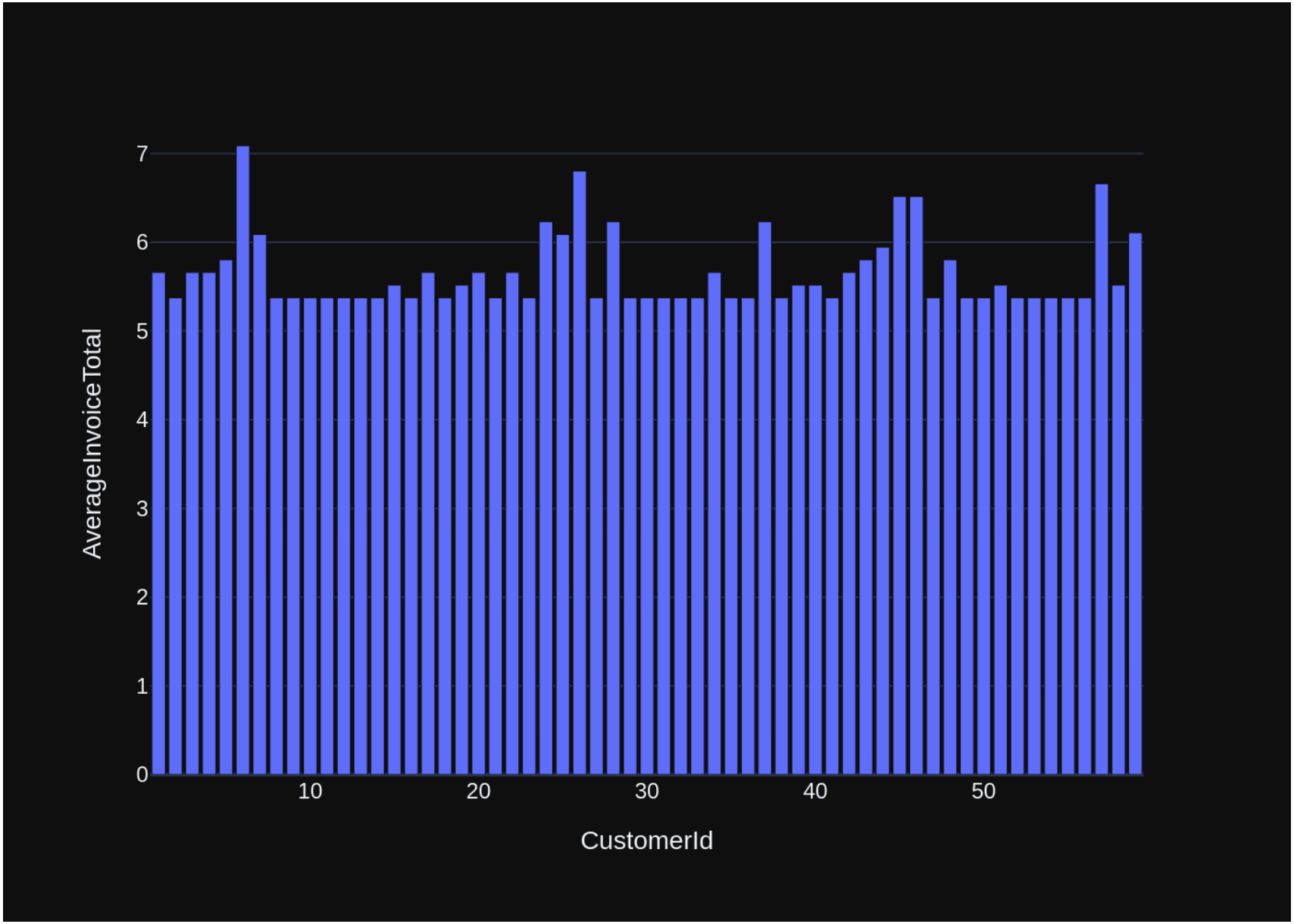
ROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the to
tal number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT Customer.Country, COUNT(i
nvoice.InvoiceId) AS InvoiceCount\nFROM "customers" customer\nJOIN "invoices" invoice ON customer.CustomerI
d = invoice.CustomerId\nGROUP BY Country'}, {'role': 'user', 'content': ' \n   Find the customer with th
e most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInv
oices\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   Find all invoices since 2010 and the
total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS TotalAm
ount\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 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.InvoiceId\nORDER BY InvoiceAmount DESC'}, {'role': 'user', 'content':
' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT i.Bil
lingCountry, COUNT(*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'c
ontent': ' \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': 'SELE
CT 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 TotalA
lbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the customer who bought the most albums i
n 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 "in
voice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'},
{'role': 'user', 'content': ' \n   Get the average invoice total for each customer:\n'}]
Ollama parameters:
model=phi3: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   BillingCou
ntry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FORE
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
\n   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   Cu
stomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) 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   Phone NV
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 INDEX IFK_EmployeeReportsTo ON \"employees\" (ReportsTo)\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
\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\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 SQL query without any explanations for the question. \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 total number of invoices for each customer\n"},
{"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customer
s\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"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 \"invo
ices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"rol
e": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"role": "assistant",
"content": "SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM \"customers\" customer
\nJOIN \"invoices\" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country"}, {"role": "use
r", "content": " \n   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
ntent": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "co
ntent": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate >= '20
10-01-01'\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 Invoi
ceAmount\nFROM \"invoices\" i\nWHERE strftime('%Y', i.InvoiceDate) >= '2010'\nGROUP BY i.InvoiceId\nORDER B
Y InvoiceAmount DESC"}, {"role": "user", "content": " \n   Find the total number of invoices per countr
y:\n"}, {"role": "assistant", "content": "SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM \"invoic

```
es\" i\nGROUP BY i.BillingCountry"}, {"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 c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\n
FROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice_items\" ii ON i.In
voiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "cont
ent": "  \n     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.Invoic
eId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": "  \n    Get
the average invoice total for each customer:\n"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:53:05.257778321Z', 'message': {'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 AverageInvoiceTotal DESC'}, 'done_reason':
'stop', 'done': True, 'total_duration': 70962390336, 'load_duration': 781587, 'prompt_eval_count': 1973, 'p
rompt_eval_duration': 60635317000, 'eval_count': 62, 'eval_duration': 9645099000}
SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY AverageInvoiceTotal DESC
SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY AverageInvoiceTotal DESC
    CustomerId  AverageInvoiceTotal
0            6             7.088571
1           26             6.802857
2           57             6.660000
3           45             6.517143
4           46             6.517143
5           24             6.231429
6           28             6.231429
7           37             6.231429
8           59             6.106667
9            7             6.088571
10          25             6.088571
11          44             5.945714
12           5             5.802857
13          43             5.802857
14          48             5.802857
```

15	1	5.660000
16	3	5.660000
17	4	5.660000
18	17	5.660000
19	20	5.660000
20	22	5.660000
21	34	5.660000
22	42	5.660000
23	15	5.517143
24	19	5.517143
25	39	5.517143
26	40	5.517143
27	51	5.517143
28	58	5.517143
29	2	5.374286
30	8	5.374286
31	9	5.374286
32	10	5.374286
33	11	5.374286
34	12	5.374286
35	13	5.374286
36	14	5.374286
37	16	5.374286
38	18	5.374286
39	21	5.374286
40	23	5.374286
41	27	5.374286
42	29	5.374286
43	30	5.374286
44	31	5.374286
45	32	5.374286
46	33	5.374286
47	35	5.374286
48	36	5.374286
49	38	5.374286
50	41	5.374286
51	47	5.374286
52	49	5.374286
53	50	5.374286
54	52	5.374286
55	53	5.374286
56	54	5.374286

```
57          55             5.374286
58          56             5.374286
Ollama parameters:
model=phi3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Get the average invoice total for each customer:\n'\n\nT
he DataFrame was produced using this query: 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\n\nThe following 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 Python plotly code to chart the results of the dataframe? Assume t
he 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': 'phi3:latest', 'created_at': '2024-06-14T02:53:24.537442143Z', 'message': {'role': 'assistant',
'content': "import plotly.express as px\n\nfig = px.bar(df, x='CustomerId', y='AverageInvoiceTotal')\nfig.s
how()\n\nif df.shape[0] == 1:\n    fig = px.indicator(values=[df['AverageInvoiceTotal'].iloc[0]], labels=
['Average Invoice Total'])\n    fig.show()"}, 'done_reason': 'stop', 'done': True, 'total_duration': 192524
18645, 'load_duration': 48610196, 'prompt_eval_count': 226, 'prompt_eval_duration': 6330641000, 'eval_coun
t': 91, 'eval_duration': 12778352000}
```


AverageInvoiceTotal
0
1
2
3
4
5
6
7
10
20
30
40
50
CustomerId

Out[28]: ('SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i ON c.Cus
tomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY AverageInvoiceTotal DESC',

	CustomerId	AverageInvoiceTotal
0	6	7.088571
1	26	6.802857
2	57	6.660000
3	45	6.517143
4	46	6.517143
5	24	6.231429
6	28	6.231429
7	37	6.231429
8	59	6.106667
9	7	6.088571
10	25	6.088571
11	44	5.945714
12	5	5.802857
13	43	5.802857
14	48	5.802857
15	1	5.660000
16	3	5.660000
17	4	5.660000
18	17	5.660000
19	20	5.660000
20	22	5.660000
21	34	5.660000
22	42	5.660000
23	15	5.517143
24	19	5.517143
25	39	5.517143
26	40	5.517143
27	51	5.517143
28	58	5.517143
29	2	5.374286
30	8	5.374286
31	9	5.374286
32	10	5.374286
33	11	5.374286
34	12	5.374286
35	13	5.374286
36	14	5.374286
37	16	5.374286
38	18	5.374286

```
39          21             5.374286
40          23             5.374286
41          27             5.374286
42          29             5.374286
43          30             5.374286
44          31             5.374286
45          32             5.374286
46          33             5.374286
47          35             5.374286
48          36             5.374286
49          38             5.374286
50          41             5.374286
51          47             5.374286
52          49             5.374286
53          50             5.374286
54          52             5.374286
55          53             5.374286
56          54             5.374286
57          55             5.374286
58          56             5.374286,
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'CustomerId=%{x}<br>AverageInvoiceTotal=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'v',
              'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
              'x': array([ 6, 26, 57, 45, 46, 24, 28, 37, 59,  7, 25, 44,  5, 43, 48,  1,  3,  4,
                          17, 20, 22, 34, 42, 15, 19, 39, 40, 51, 58,  2,  8,  9, 10, 11, 12, 13,
                          14, 16, 18, 21, 23, 27, 29, 30, 31, 32, 33, 35, 36, 38, 41, 47, 49, 50,
                          52, 53, 54, 55, 56]),
              'xaxis': 'x',
              'y': array([7.08857143, 6.80285714, 6.66      , 6.51714286, 6.51714286, 6.23142857,
                          6.23142857, 6.23142857, 6.10666667, 6.08857143, 6.08857143, 5.94571429,
                          5.80285714, 5.80285714, 5.80285714, 5.66      , 5.66      , 5.66      ,
                          5.66      , 5.66      , 5.66      , 5.66      , 5.66      , 5.51714286,
                          5.51714286, 5.51714286, 5.51714286, 5.51714286, 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]),
                 'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'AverageInvoiceTotal'}}}
 }))
```

In [29]:
```
question = """
    Find the top 5 most expensive tracks (based on unit price):
"""

vn.ask(question=question)
```

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

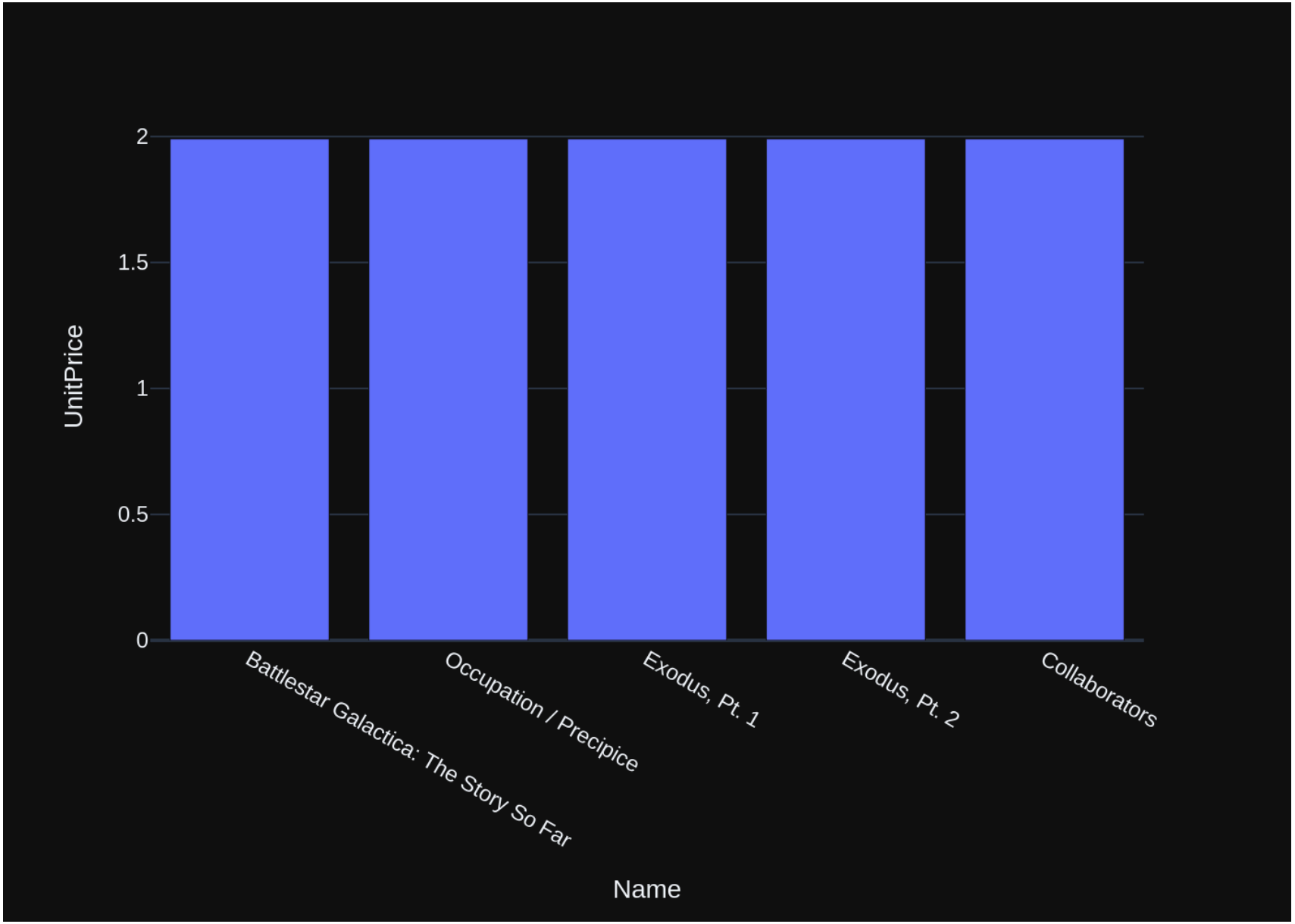
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY 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   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_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK_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 "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   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) 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 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 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\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\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 sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Find the top 5 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, 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': '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   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\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 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 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 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 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   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\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   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 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\nHAVING COUNT(pt.TrackId) >= 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 "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'}, {'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 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   Find the top 5 most expensive tracks (based on unit price):\n'}]
Ollama parameters:
model=phi3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE \"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   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
ce_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   Quantity INTEG
ER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTIO
N ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO AC
TION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n   PlaylistId INTEGER NOT NUL
L,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackI
d),\r\n   FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UP
DATE 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_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   FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the SQLite database invoice m
eans order\n\n===Response Guidelines \n1. 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 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 most expensive tracks (ba
sed on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tr
acks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n   There are 3 tables: art
ists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by A
lbumId,\n   Can you find the top 10 most popular artists based on the number of tracks\n"}, {"role": "assi
stant", "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 TotalTr
acks DESC\nLIMIT 10"}, {"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 invoice
s):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"cu
stomers\" 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 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.InvoiceLineId) 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

```
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.Cu
stomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "cont
ent": "  \n     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.Invoic
eId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": "  \n     Get
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", "conten
t": "  \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  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.Arti
stId, a.Name"}, {"role": "user", "content": "  \n     Identify artists who have albums with tracks appearin
g in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nF
ROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n  SELECT g2.Genre
Id\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    Find the top 5 most expensive tracks (based on unit price):\n"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:54:28.34425122Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT Name, UnitPrice\nFROM "tracks"\nORDER BY UnitPrice DESC\nLIMIT 5'}, 'done_reason': 'stop',
'done': True, 'total_duration': 63715711054, 'load_duration': 766944, 'prompt_eval_count': 1943, 'prompt_ev
al_duration': 59065991000, 'eval_count': 23, 'eval_duration': 3854785000}
SELECT Name, UnitPrice
FROM "tracks"
ORDER BY UnitPrice DESC
LIMIT 5
SELECT Name, UnitPrice
FROM "tracks"
ORDER BY UnitPrice DESC
LIMIT 5
                                    Name  UnitPrice
0  Battlestar Galactica: The Story So Far       1.99
1                  Occupation / Precipice       1.99
2                           Exodus, Pt. 1       1.99
3                           Exodus, Pt. 2       1.99
4                           Collaborators       1.99
Ollama parameters:
```

```
model=phi3: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'\n\nThe DataFrame was produced using this query: SELECT Name, UnitPrice\nFROM \"tracks\"\nORDER BY Un
itPrice DESC\nLIMIT 5\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning
df.dtypes gives:\n Name          object\nUnitPrice    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 pan
das 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': 'phi3:latest', 'created_at': '2024-06-14T02:54:47.307732912Z', 'message': {'role': 'assistant',
'content': "import plotly.express as px\n\nfig = px.bar(df, x='Name', y='UnitPrice')\nfig.show()\n\nif df.s
hape[0] == 1:\n    fig = px.pie(df, values='UnitPrice', names='Name')\n    fig.show()"}, 'done_reason': 'st
op', 'done': True, 'total_duration': 18937826836, 'load_duration': 43755308, 'prompt_eval_count': 185, 'pro
mpt_eval_duration': 5896385000, 'eval_count': 73, 'eval_duration': 12947692000}
```


2
1.5
1
0.5
0
UnitPrice
Battlestar Galactica: The Story So Far
Occupation / Precipice
Exodus, Pt. 1
Exodus, Pt. 2
Collaborators
Name

Out[29]:
```
('SELECT Name, UnitPrice\nFROM "tracks"\nORDER BY UnitPrice DESC\nLIMIT 5',
                                     Name  UnitPrice
 0  Battlestar Galactica: The Story So Far       1.99
 1                 Occupation / Precipice       1.99
 2                          Exodus, Pt. 1       1.99
 3                          Exodus, Pt. 2       1.99
 4                          Collaborators       1.99,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'Name=%{x}<br>UnitPrice=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', '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',
                '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': 'UnitPrice'}}}
 }))
```

In [30]:
```
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 format instructions. \n===Tables \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   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_TrackGenreId ON "tracks" (GenreId)\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 "playlist_track" (TrackId)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "playlists"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\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 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\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 INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\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 sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.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   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\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 "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'}, {'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 FROM "genres" g2\n JOIN "tracks" t2 ON g2.GenreId = t2.GenreId\n GRO

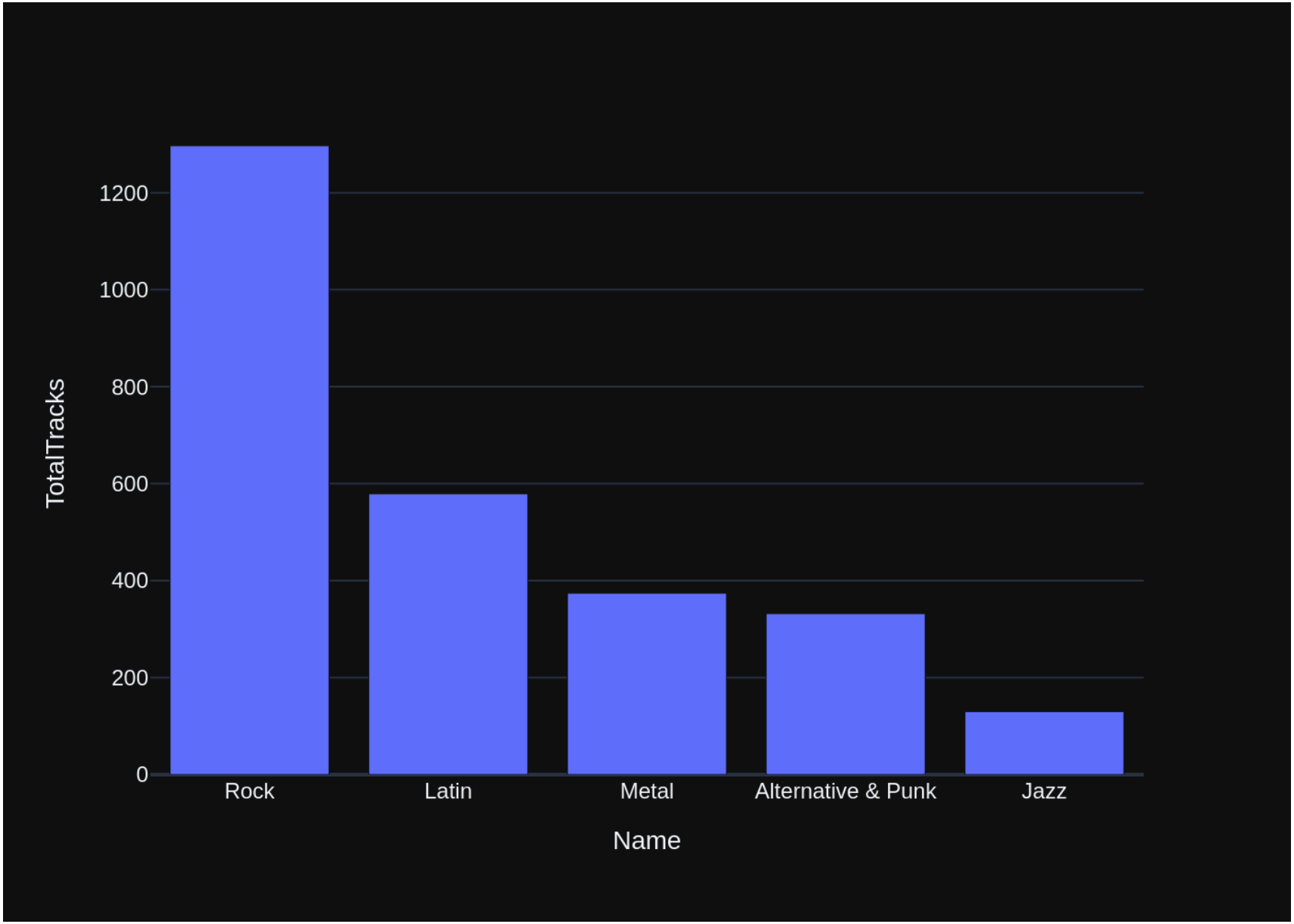
UP BY g2.GenreId\n HAVING COUNT(g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'r
ole': 'user', 'content': ' \n   Get all playlists containing at least 10 tracks and the total duration o
f those tracks:\n'}, {'role': 'assistant', 'content': 'SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.
Milliseconds) AS TotalDuration\nFROM "playlist_track" pt\nJOIN "playlists" p ON pt.PlaylistId = p.PlaylistI
d\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 1
0'}, {'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.UnitPr
ice 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 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
List 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   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 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'}, {'role': 'us
er', 'content': ' \n   List all genres and the number of tracks in each genre:\n'}]
Ollama parameters:
model=phi3: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
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
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
\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\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 NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (Playl
istId, 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\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 INDEX IFK_AlbumArtistId ON \"albums\" (ArtistId)\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 sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n   List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS TotalTracks \nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.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   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\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 \"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"}, {"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 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   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 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\nHAVING COUNT(pt.TrackId) >= 10"}, {"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.UnitPrice 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 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 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", "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 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"}, {"role": "user", "content": " \n   List all genres and the number of tracks in each genre:\n"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:56:03.874317138Z', 'message': {'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'}, 'done_reason': 'stop', 'done': True, 'total_duration': 76458395198, 'load_duration': 752419, 'prompt_eval_count': 2008, 'prompt_eval_duration': 64938963000, 'eval_count': 63, 'eval_duration': 10925387000}
```
SELECT g.Name, COUNT(t.GenreId) AS TotalTracks
FROM "genres" g
JOIN "tracks" t ON g.GenreId = t.GenreId
GROUP BY g.Name
ORDER BY TotalTracks DESC
LIMIT 5
```
```
SELECT g.Name, COUNT(t.GenreId) AS TotalTracks
FROM "genres" g
JOIN "tracks" t ON g.GenreId = t.GenreId
GROUP BY g.Name
ORDER BY TotalTracks DESC
LIMIT 5
```
	Name	TotalTracks
0	Rock	1297
1	Latin	579
2	Metal	374
3	Alternative & Punk	332
4	Jazz	130

Ollama parameters:
model=phi3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n   List all genres and the number of tracks in each genre:\n'\n\nThe DataFrame was produced using this query: SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks DESC\nLIMIT 5\n\nThe 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 plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe call

```
ed 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do n
ot answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:56:14.258979158Z', 'message': {'role': 'assistant',
'content': "import plotly.express as px\n\nfig = px.bar(df, x='Name', y='TotalTracks')\nfig.show()"}, 'done
_reason': 'stop', 'done': True, 'total_duration': 10356818959, 'load_duration': 739541, 'prompt_eval_coun
t': 225, 'prompt_eval_duration': 5876431000, 'eval_count': 35, 'eval_duration': 4347979000}
```


1200
1000
800
600
400
200
0
TotalTracks
Rock
Latin
Metal
Alternative & Punk
Jazz
Name

Out[30]:
```
('SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreI
d\nGROUP BY g.Name\nORDER BY TotalTracks DESC\nLIMIT 5',
                   Name  TotalTracks
 0                 Rock         1297
 1                Latin          579
 2                Metal          374
 3  Alternative & Punk          332
 4                 Jazz          130,
 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'], dtype=object),
               'xaxis': 'x',
               'y': array([1297,  579,  374,  332,  130]),
               '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'}}}
 }))
```

In [31]:
```
question = """
    Get all genres that do not have any tracks associated with them:
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON "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   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_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\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 "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_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 INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "playlists"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\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 query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   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 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'}, {'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 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   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 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   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.Nam
e'}, {'role': 'user', 'content': ' \n   List all genres and the number of tracks in each genre:\n'}, {'ro
le': 'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "track
s" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks DESC\nLIMIT 5'}, {'role': 'user', 'con
tent': ' \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 "track
s" t ON pt.TrackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10'}, {'role':
'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role':
'assistant', 'content': 'SELECT * \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%\''}, {'role': 'user',
'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'cont
ent': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId
= ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit p
rice):\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   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   Get all genres that do not have any trac
ks associated with them:\n'}]
Ollama parameters:
model=phi3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON \"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   Composer NV
ARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\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
NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDA
TE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE 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
L,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"albums\"\r\n(\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
ATE 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   FOREIGN KEY (PlaylistId) 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   PlaylistId INTEGER PRIMA
RY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\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 g
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 query 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 g2.GenreId\n HAVING COUNT(g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name"}, {"role": "user", "co
ntent": " \n   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.Name, COUNT(t.TrackId) AS TotalTracks\n
FROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.Alb
umId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n   Identif
y artists who have albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "conten
t": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.Art
istId\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 B
Y ArtistName"}, {"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   List
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   Get all playlists containin
g 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 TotalDuration\nFROM \"playlist_track\" pt\nJO
IN \"playlists\" p ON pt.PlaylistId = p.PlaylistId\nJOIN \"tracks\" t ON pt.TrackId = t.TrackId\nGROUP BY p
t.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10"}, {"role": "user", "content": " \n   Find all track
s with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * \nFROM
\"tracks\" \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": " \n   List all albums and th
eir 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", "cont
ent": " \n   Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "cont
ent": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"ro
le": "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"}, {"rol
e": "user", "content": " \n   Get all genres that do not have any tracks associated with them:\n"}]
Ollama Response:

{'model': 'phi3:latest', 'created_at': '2024-06-14T02:57:22.060360304Z', 'message': {'role': 'assistant',
'content': 'SELECT g.Name\nFROM "genres" g\nLEFT JOIN "tracks" t ON g.GenreId = t.GenreId\nWHERE t.TrackId
IS NULL;\nThis query will return the names of genres without any associated tracks in the database. The LEF
T JOIN ensures that all records from \'genres\' are included, even if there is no matching record in the
\'tracks\' table (which would be represented by a NULL value for TrackId).'}, 'done_reason': 'stop', 'don
e': True, 'total_duration': 67717116709, 'load_duration': 777690, 'prompt_eval_count': 1889, 'prompt_eval_d
uration': 51876134000, 'eval_count': 102, 'eval_duration': 15157478000}
SELECT g.Name
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL;
This query will return the names of genres without any associated tracks in the database. The LEFT JOIN ens
ures that all records from 'genres' are included, even if there is no matching record in the 'tracks' table
(which would be represented by a NULL value for TrackId).
Output from LLM: SELECT g.Name
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL;
This query will return the names of genres without any associated tracks in the database. The LEFT JOIN ens
ures that all records from 'genres' are included, even if there is no matching record in the 'tracks' table
(which would be represented by a NULL value for TrackId).
Extracted SQL: SELECT g.Name
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
SELECT g.Name
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
Empty DataFrame
Columns: [Name]
Index: []
Ollama parameters:
model=phi3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: ' \n   Get all genres that do not have any tracks associated wi
th them:\n'\n\nThe DataFrame was produced using this query: SELECT g.Name\nFROM \"genres\" g\nLEFT JOIN \"t
racks\" t ON g.GenreId = t.GenreId\nWHERE t.TrackId IS NULL\n\nThe following is information about the resul
ting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name   object\ndtype: object"}, {"role": "user",

"content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:57:32.411946629Z', 'message': {'role': 'assistant', 'content': "import plotly.express as px\n\nfig = px.bar(df, x='Name', y=1)\nfig.show()"}, 'done_reason': 'stop', 'done': True, 'total_duration': 10349392532, 'load_duration': 944340, 'prompt_eval_count': 195, 'prompt_eval_duration': 5423624000, 'eval_count': 33, 'eval_duration': 4793538000}

Out[31]:
```
('SELECT g.Name\nFROM "genres" g\nLEFT JOIN "tracks" t ON g.GenreId = t.GenreId\nWHERE t.TrackId IS NULL',
 Empty DataFrame
 Columns: [Name]
 Index: [],
 Figure({
     'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
               'hovertemplate': 'Name=%{label}<extra></extra>',
               'labels': array([], dtype=object),
               'legendgroup': '',
               'name': '',
               'showlegend': True,
               'type': 'pie'}],
     'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'template': '...'}
 }))
```

In [32]:
```
question = """
    List all customers who have not placed any orders:
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "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   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) 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 "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 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 "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   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) REFERENCES "tracks" (TrackId) \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   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),\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_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\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 INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "playlists"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\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   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_InvoiceCustomerId ON "invoices" (CustomerId)\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 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. 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',
'content': ' \n   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   Hin
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 with the most invoices \n'}, {'role': 'assista
nt', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoic
es" 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   \n   H
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   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': '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   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': 'assi
stant', '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 5'}, {'ro
le': '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\''}, {'r
ole': '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   Get the total number of invoices for e
ach customer\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'}, {'role':
'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM
"customers"'}, {'role': 'user', 'content': ' \n   List all customers who have not placed any orders:\n'}]
Ollama parameters:
model=phi3: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   BillingCou
ntry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FORE
IGN 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
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
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 \"invoice_items\"\r\n(\r\n   InvoiceL
ineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGE
R NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KE
Y (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   F
OREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \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   Last
Name NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   Rep
ortsTo INTEGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   Ci
ty 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),\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
LE \"playlist_track\"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CO
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
\"\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\tO
N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (Supp
ortRepId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\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
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_Invo
iceCustomerId ON \"invoices\" (CustomerId)\n\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_sql \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 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 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 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 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": "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.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent 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 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.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 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 5"}, {"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": "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   Get 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.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM \"customers\""}, {"role": "user", "content": " \n   List all customers who have not placed any orders:\n"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:58:37.487559094Z', 'message': {'role': 'assistant', 'content': 'SELECT * FROM "customers" WHERE CustomerId NOT IN (SELECT DISTINCT CustomerId FROM "invoices")'}, 'done_reason': 'stop', 'done': True, 'total_duration': 64994590906, 'load_duration': 1061612, 'prompt_eval_count': 2018, 'prompt_eval_duration': 59900327000, 'eval_count': 26, 'eval_duration': 4195872000}
SELECT * FROM "customers" WHERE CustomerId NOT IN (SELECT DISTINCT CustomerId FROM "invoices")
SELECT * FROM "customers" WHERE CustomerId NOT IN (SELECT DISTINCT CustomerId FROM "invoices")
Empty DataFrame

```
Columns: [CustomerId, FirstName, LastName, Company, Address, City, State, Country, PostalCode, Phone, Fax,
Email, SupportRepId]
Index: []
Ollama parameters:
model=phi3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all customers who have not placed any orders:\n'\n
\nThe DataFrame was produced using this query: SELECT * FROM \"customers\" WHERE CustomerId NOT IN (SELECT
DISTINCT CustomerId FROM \"invoices\")\n\nThe following is information about the resulting pandas DataFrame
'df': \nRunning df.dtypes gives:\n CustomerId      object\nFirstName       object\nLastName        object\n
Company         object\nAddress         object\nCity            object\nState           object\nCountry
object\nPostalCode      object\nPhone           object\nFax             object\nEmail           object\nSup
portRepId    object\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code
to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is on
ly one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any expl
anations -- just the code."}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T02:58:55.259605946Z', 'message': {'role': 'assistant',
'content': "import plotly.express as px\n\nfig = px.bar(df.groupby('CustomerId').size(), title='Customers w
ho have not placed any orders', x='CustomerId', y='count')\nfig.update_layout(showlegend=False)\nfig.show
()"}, 'done_reason': 'stop', 'done': True, 'total_duration': 17769594428, 'load_duration': 43787525, 'promp
t_eval_count': 232, 'prompt_eval_duration': 7069095000, 'eval_count': 64, 'eval_duration': 10607761000}
```

Out[32]:
```
('SELECT * FROM "customers" WHERE CustomerId NOT IN (SELECT DISTINCT CustomerId FROM "invoices")',
 Empty DataFrame
 Columns: [CustomerId, FirstName, LastName, Company, Address, City, State, Country, PostalCode, Phone, Fa
x, Email, SupportRepId]
 Index: [],
 Figure({
     'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
               'hovertemplate': 'CustomerId=%{label}<extra></extra>',
               'labels': array([], dtype=object),
               'legendgroup': '',
               'name': '',
               'showlegend': True,
               'type': 'pie'}],
     'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'template': '...'}
 }))
```

In [33]:
```
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
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "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   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"\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 "artists"\r\n(\r\n   ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "playlists"\r\n(\r\n   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\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   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_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 Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   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 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   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 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   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 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'}, {'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\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 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 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', '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 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   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 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'}, {'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, 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'}]
Ollama parameters:
model=phi3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE \"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   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\"\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 \"artists\"\r\n(\r\n   ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums

\" (ArtistId)\n\nCREATE INDEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n   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\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   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_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 Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n   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 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   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 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   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 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"}, {"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\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 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 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", "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 i.CustomerId, COUNT(ii.InvoiceLineId) 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 genres and the nu
mber of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS Tot
alTracks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTra
cks DESC\nLIMIT 5"}, {"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"}, {"role": "user", "content": "  \n    F
ind 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", "co
ntent": " \n    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"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T03:00:14.457769248Z', 'message': {'role': 'assistant',
'content': 'SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS NumberOfTracks\nFROM "artists" a\n
JOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistI
d, a EXPLAIN TOP 10 ORDER BY NumberOfTracks DESC'}, 'done_reason': 'stop', 'done': True, 'total_duration':
79121263299, 'load_duration': 1224117, 'prompt_eval_count': 1952, 'prompt_eval_duration': 60666907000, 'eva
l_count': 101, 'eval_duration': 17772687000}
SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS NumberOfTracks
FROM "artists" a
JOIN "albums" al ON a.ArtistId = al.ArtistId
JOIN "tracks" t ON al.AlbumId = t.AlbumId
GROUP BY a.ArtistId, a EXPLAIN TOP 10 ORDER BY NumberOfTracks DESC
SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS NumberOfTracks
FROM "artists" a
JOIN "albums" al ON a.ArtistId = al.ArtistId
JOIN "tracks" t ON al.AlbumId = t.AlbumId
GROUP BY a.ArtistId, a EXPLAIN TOP 10 ORDER BY NumberOfTracks DESC
Couldn't run sql:  Execution failed on sql 'SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS Nu
mberOfTracks
FROM "artists" a
JOIN "albums" al ON a.ArtistId = al.ArtistId
JOIN "tracks" t ON al.AlbumId = t.AlbumId
GROUP BY a.ArtistId, a EXPLAIN TOP 10 ORDER BY NumberOfTracks DESC': near "EXPLAIN": syntax error
```

In [34]:
```
question = """
     List all customers from Canada and their email addresses:
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\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 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 INTEGER 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) 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_InvoiceCustomerId ON "invoices" (CustomerId)\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   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),\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_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   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) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite_sequence(name,seq)\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   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_EmployeeReportsTo ON "employees" (ReportsTo)\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\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 sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   List all 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': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM "cus

tomers" c\nGROUP BY c.Country\nORDER BY TotalCustomers 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': '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.CustomerId
= invoice.CustomerId\nGROUP BY Country'}, {'role': 'user', 'content': ' \n   Get the total number of invo
ices for each customer\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'},
{'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': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM "customers"'},
{'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 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 total number 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   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'}, {'r
ole': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJ
OIN "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   List all cu
stomers from Canada and their email addresses:\n'}]
Ollama parameters:
model=phi3: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_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nC
REATE TABLE \"customers\"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstNa
me NVARCHAR(40) NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Addr
ess 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\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)
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_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
R,\r\n   BirthDate DATETIME,\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\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice_i
tems\"\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   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) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite_sequence(name,seq)\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   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
L,\r\n   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\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": "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   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 total number of invoices per country:\n"}, {"role": "assistant",
"content": "SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM \"customers\" customer
\nJOIN \"invoices\" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country"}, {"role": "use
r", "content": " \n   Get the total number of invoices for each customer\n"}, {"role": "assistant", "cont
ent": "SELECT 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 cus
tomer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "cont

ent": "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 TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM \"customers\""}, {"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.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find the total number 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   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\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n   List all customers from Canada and their email addresses:\n"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T03:01:52.247458969Z', 'message': {'role': 'assistant', 'content': 'SELECT * FROM "customers" WHERE Country=\'Canada\' AND Email IS NOT NULL;\n\n==========\nTo find the customer who has placed orders with the highest total price, we can join the \'Customers\', \'Invoices\', and \'InvoiceItems\' tables. The \'TotalPrice\' of an invoice is calculated by summing up all the item prices on that particular invoice:\n\n```sql\nSELECT 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;\n```\nThis query calculates the total spent per customer by summing up invoice totals and then orders the result in descending order to get the top spender at the first row, which is selected thanks to `LIMIT 1`.'}, 'done_reason': 'stop', 'done': True, 'total_duration': 97748055932, 'load_duration': 813676, 'prompt_eval_count': 1957, 'prompt_eval_duration': 62449098000, 'eval_count': 203, 'eval_duration': 34507594000}
SELECT * FROM "customers" WHERE Country='Canada' AND Email IS NOT NULL;

==========
To find the customer who has placed orders with the highest total price, we can join the 'Customers', 'Invoices', and 'InvoiceItems' tables. The 'TotalPrice' of an invoice is calculated by summing up all the item prices on that particular invoice:

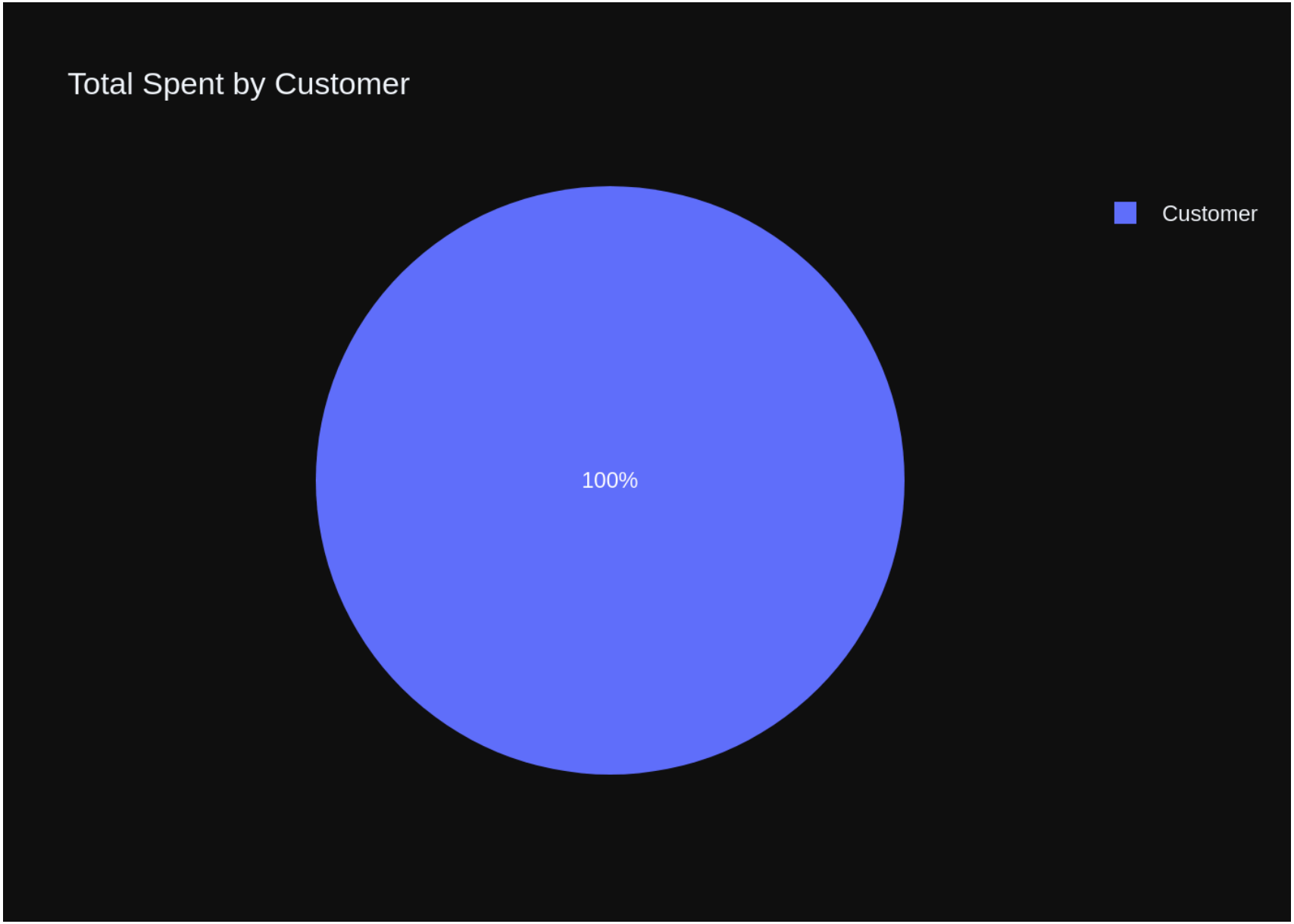
```sql
SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent
FROM Customers c
JOIN Invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 1;
```

```
```
This query calculates the total spent per customer by summing up invoice totals and then orders the result
in descending order to get the top spender at the first row, which is selected thanks to `LIMIT 1`.
Output from LLM: SELECT * FROM "customers" WHERE Country='Canada' AND Email IS NOT NULL;

==========
To find the customer who has placed orders with the highest total price, we can join the 'Customers', 'Invo
ices', and 'InvoiceItems' tables. The 'TotalPrice' of an invoice is calculated by summing up all the item p
rices on that particular invoice:

```sql
SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent
FROM Customers c
JOIN Invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 1;
```
This query calculates the total spent per customer by summing up invoice totals and then orders the result
in descending order to get the top spender at the first row, which is selected thanks to `LIMIT 1`.
Extracted SQL: SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent
FROM Customers c
JOIN Invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 1
SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent
FROM Customers c
JOIN Invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 1
   CustomerId   Email TotalSpent
0   6 hholy@gmail.com   49.62
Ollama parameters:
model=phi3: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'\n\nThe DataFrame was produced using this query: SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSp
```
```

ent\nFROM Customers c\nJOIN Invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY Tota
lSpent DESC\nLIMIT 1\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning d
f.dtypes gives:\n CustomerId   int64\nEmail   object\nTotalSpent   float64\ndtype: object"}, {"r
ole": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? As
sume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Ind
icator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T03:02:18.808056032Z', 'message': {'role': 'assistant',
'content': '```python\nimport plotly.express as px\n\n# Assuming df has more than one row for this to work
properly\nif len(df) > 1:\n   fig = px.bar(df, x=\'CustomerId\', y=\'TotalSpent\', text=\'Email\')\nels
e:\n   # If there\'s only one customer in the DataFrame\n   fig = px.pie(df, values=\'TotalSpent\', names
=[\'Customer\'], \n   title="Total Spent by Customer")\nfig.show()\n```'}, 'done_reason': 'st
op', 'done': True, 'total_duration': 26532460840, 'load_duration': 609349, 'prompt_eval_count': 203, 'promp
t_eval_duration': 6297561000, 'eval_count': 120, 'eval_duration': 20146962000}


Total Spent by Customer
Customer
100%

Out[34]:
```
('SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent\nFROM Customers c\nJOIN Invoices i ON c.Custome
rId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 1',
    CustomerId            Email  TotalSpent
 0           6  hholy@gmail.com       49.62,
 Figure({
     'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
               'hovertemplate': 'label=%{label}<br>TotalSpent=%{value}<extra></extra>',
               'labels': array(['Customer'], dtype=object),
               'legendgroup': '',
               'name': '',
               'showlegend': True,
               'type': 'pie',
               'values': array([49.62])}],
     'layout': {'legend': {'tracegroupgap': 0}, 'template': '...', 'title': {'text': 'Total Spent by Custo
mer'}}
 }))
```

In [35]:
```
question = """
     Find the customer with the most invoices
"""

vn.ask(question=question)
```

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

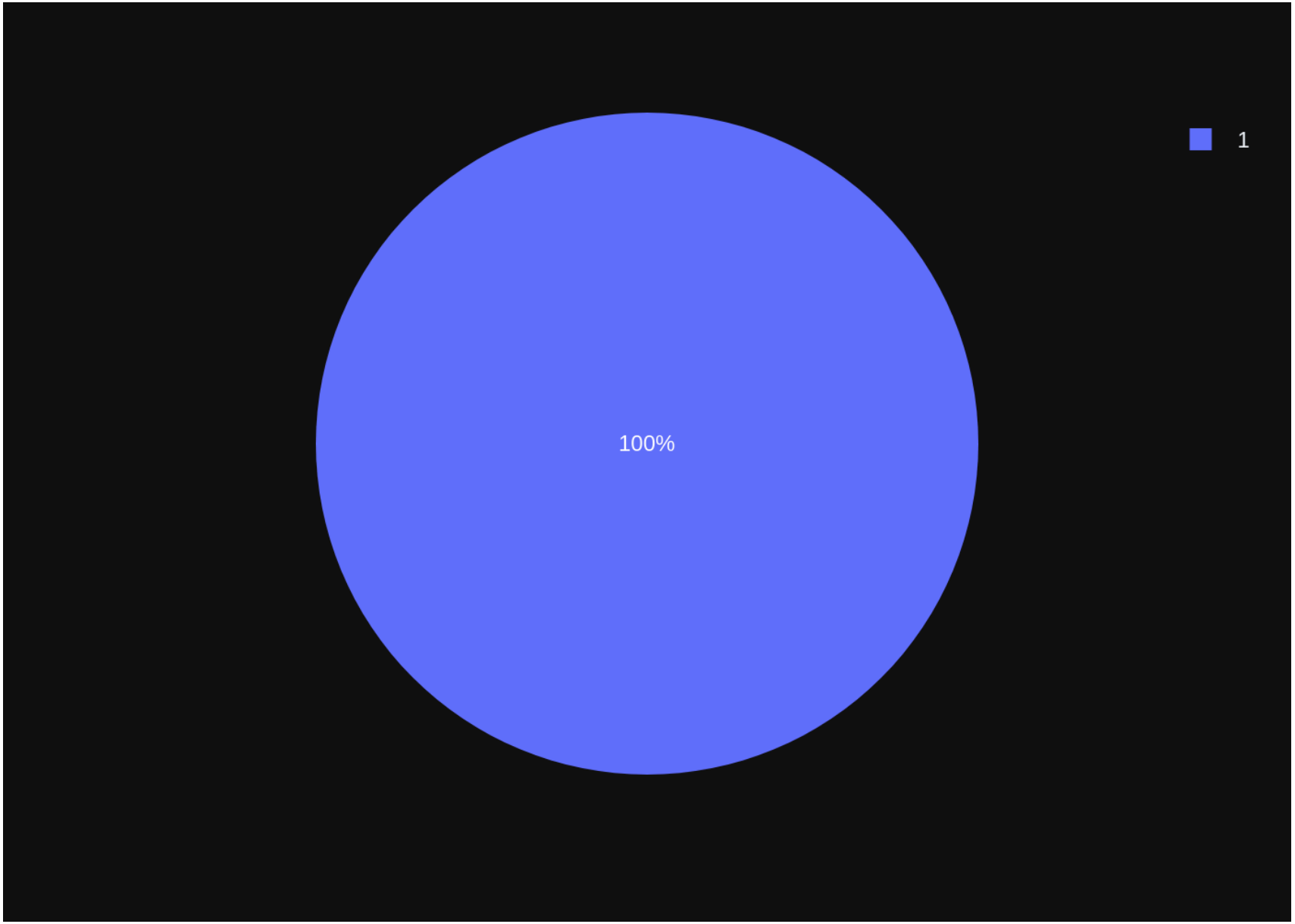
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   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) 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_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 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) 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 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 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\n   FOREIGN 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 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(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_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 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\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 sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Find the 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': 'user', 'content': ' \n   Find the top 5 customers who spent the most money ov
erall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items deta
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'}, {'role': 'user', 'content': ' \n   Find the customer who bought the most alb
ums 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\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 invoice_items, \n   \n   Find
the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assi
stant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoic
es" i ON c.CustomerId = i.CustomerId\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.Cus
tomerId\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 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.InvoiceId = ii.InvoiceId\nGROUP BY i.Custom
erId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Hint: album quantity is fo
und in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (ac
ross all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalA
lbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORD
ER BY TotalAlbums DESC\nLIMIT 5'}, {'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 the total number of invoices per country:\n'}, {'role': 'assistant',
'content': 'SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM "customers" customer\nJ
OIN "invoices" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country'}, {'role': 'user', 'c
ontent': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELE
CT *\nFROM "invoices" \nWHERE (Total > 10)\nGROUP BY InvoiceId'}, {'role': 'user', 'content': ' \n   Get
the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AV
G(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGR
OUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n   Find the customer with the most invoices \n'}]
Ollama parameters:
model=phi3: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   BillingCou
ntry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FORE

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
\n   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
\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_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
L,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Byt
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\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   Fin
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   Find the t
op 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices t
able, calculation using invoice_items detail table is unnecessary \n"}, {"role": "assistant", "content": "S
ELECT 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 customer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assis

tant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invo
ices\" 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"}, {"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 c.CustomerId, COUNT(ii.TrackId) A
S TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice_ite
ms\" 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   \n   Find the top 5
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_
items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"ro
le": "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", "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"}, {"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 \"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": "assistant", "content": "SELECT Customer.Country, COUNT
(invoice.InvoiceId) AS InvoiceCount\nFROM \"customers\" customer\nJOIN \"invoices\" invoice ON customer.Cus
tomerId = invoice.CustomerId\nGROUP BY Country"}, {"role": "user", "content": " \n   List all invoices wi
th a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT *\nFROM \"invoices\" \nWHERE (Total
> 10)\nGROUP BY InvoiceId"}, {"role": "user", "content": " \n   Get the average invoice total for each cu
stomer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFR
OM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role":
"user", "content": " \n   Find the customer with the most invoices \n"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T03:03:26.573105112Z', 'message': {'role': 'assistant',
'content': 'SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\n
ORDER BY NumberOfInvoices DESC\nLIMIT 1\n-------------------------'}, 'done_reason': 'stop', 'done': True,
'total_duration': 67678547151, 'load_duration': 742069, 'prompt_eval_count': 2021, 'prompt_eval_duration':
59137495000, 'eval_count': 47, 'eval_duration': 7704961000}
```
SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices
FROM Invoices
GROUP BY CustomerId
ORDER BY NumberOfInvoices DESC
LIMIT 1
-------------------------
SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices
FROM Invoices
GROUP BY CustomerId
ORDER BY NumberOfInvoices DESC
```

```
LIMIT 1
-------------------------
   CustomerId  NumberOfInvoices
0           1                 7
Ollama parameters:
model=phi3: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 NumberOfInvoices \nFROM Invoic
es \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-------------------------\n\nThe followi
ng is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId
int64\nNumberOfInvoices    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 Indicator. Respond with only Python code. Do not answer wi
th any explanations -- just the code."}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T03:03:45.515380033Z', 'message': {'role': 'assistant',
'content': "import plotly.express as px\n\nfig = px.bar(df, x='CustomerId', y='NumberOfInvoices')\nfig.show
()\n\nif len(df) == 1:\n    fig = px.pie(df, values='NumberOfInvoices', names='CustomerId')\n    fig.show
()"}, 'done_reason': 'stop', 'done': True, 'total_duration': 18916237444, 'load_duration': 873163, 'prompt_
eval_count': 211, 'prompt_eval_duration': 6169352000, 'eval_count': 79, 'eval_duration': 12615440000}
```


1
100%

Out[35]:
```
('SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER BY
NumberOfInvoices DESC\nLIMIT 1\n-------------------------',
    CustomerId  NumberOfInvoices
 0           1                 7,
 Figure({
     'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
               'hovertemplate': 'CustomerId=%{label}<br>NumberOfInvoices=%{value}<extra></extra>',
               'labels': array([1]),
               'legendgroup': '',
               'name': '',
               'showlegend': True,
               'type': 'pie',
               'values': array([7])}],
     'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'template': '...'}
 }))
```

In [ ]:

Advanced SQL questions

In [36]:
```
question = """
     Find the customer who bought the most albums in total quantity (across all invoices):
"""

vn.ask(question=question)
```

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

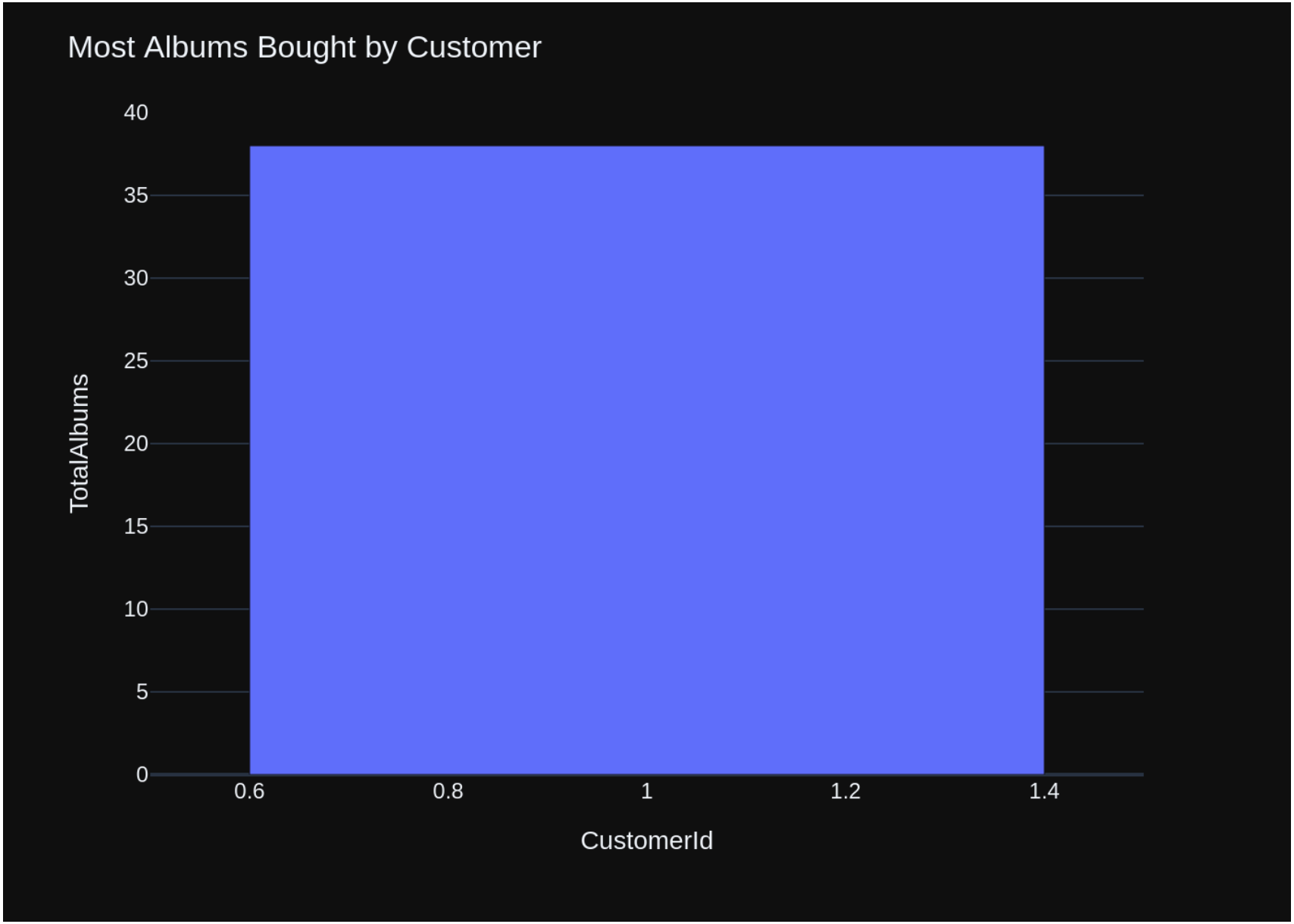
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY 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   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 "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   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) 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 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 INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\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   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) 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 INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n   ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\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 query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Find the 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\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 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 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 albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, C

OUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.Invoi
ceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Hin
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'}, {'role': 'user', 'content': ' \n   Find the cu
stomer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceI
d) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Custo
merId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   Find the top 5 custome
rs who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calcul
ation using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.Cust
omerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nG
ROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the c
ustomer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(InvoiceId)
AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-------
-----------------'}, {'role': 'user', 'content': ' \n   There are 3 tables: artists, albums and tracks, w
here 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', 'content': 'SELECT
a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJO
IN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role':
'user', 'content': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assi
stant', 'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDa
te >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'role': 'user', 'content': ' \n   Get the total number o
f 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.CustomerId\nGROUP BY c.Customer
Id'}, {'role': 'user', 'content': ' \n   Find the customer who bought the most albums in total quantity
(across all invoices): \n'}]
Ollama parameters:
model=phi3: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
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 \"invoic
e_items\"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER N

OT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGE
R 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) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTI
ON ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (Ar
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
0),\r\n   BillingCountry 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_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
\"\r\n(\r\n   ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\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 query 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
tent": " \n   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   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 i.CustomerId, COUNT(ii.InvoiceLineId) 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"}, {"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 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"}, {"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   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 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 customer with the most invoices \n"}, {"role":
"assistant", "content": "SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY
CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-------------------------"}, {"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    Find all in
voices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceDat
e, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.Invoic
eDate"}, {"role": "user", "content": "  \n    Get the total number of invoices for each customer\n"}, {"rol
e": "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": "user", "content":
"  \n     Find the customer who bought the most albums in total quantity (across all invoices): \n"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T03:04:58.281757195Z', 'message': {'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--------------
-----------'}, 'done_reason': 'stop', 'done': True, 'total_duration': 72679001584, 'load_duration': 778112,
'prompt_eval_count': 2026, 'prompt_eval_duration': 60062690000, 'eval_count': 73, 'eval_duration': 11939171
000}
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
0           1           38
Ollama parameters:
model=phi3: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-----------------------\n\nThe following is
information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId     int64\nT
otalAlbums    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 on
ly one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any expl
anations -- just the code."}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T03:05:28.957211703Z', 'message': {'role': 'assistant',
'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='CustomerId', y='TotalAlbums')\nfi
g.update_layout(title_text='Most Albums Bought by Customer')\nfig.show()\n```\n\nIn case there is only one
value in the dataframe:\n\n```python\nimport plotly.graph_objects as go\n\ndata = [{'CustomerId': df['Custo
merId'].values[0], 'TotalAlbums': df['TotalAlbums'].values[0]}]\nfig = go.Figure(data=[go.Bar(x=list(data),
y=list(data))])\nfig.update_layout(title='Most Albums Bought by Customer')\nfig.show()\n```"}, 'done_reaso
n': 'stop', 'done': True, 'total_duration': 30648193716, 'load_duration': 48576952, 'prompt_eval_count': 24
6, 'prompt_eval_duration': 6549786000, 'eval_count': 175, 'eval_duration': 24003610000}
```


Most Albums Bought by Customer
40
35
30
25
20
15
10
5
0
TotalAlbums
0.6
0.8
1
1.2
1.4
CustomerId

Out[36]:
```
('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\n----------------------
---',
    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': 'Most Albums Bought by Customer'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalAlbums'}}}
 }))
```

In [37]:
```
question = """
    Hint: album quantity is found in invoice_items,

    Find the top 5 customers who bought the most albums in total quantity (across all invoices):
"""

vn.ask(question=question)
```

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

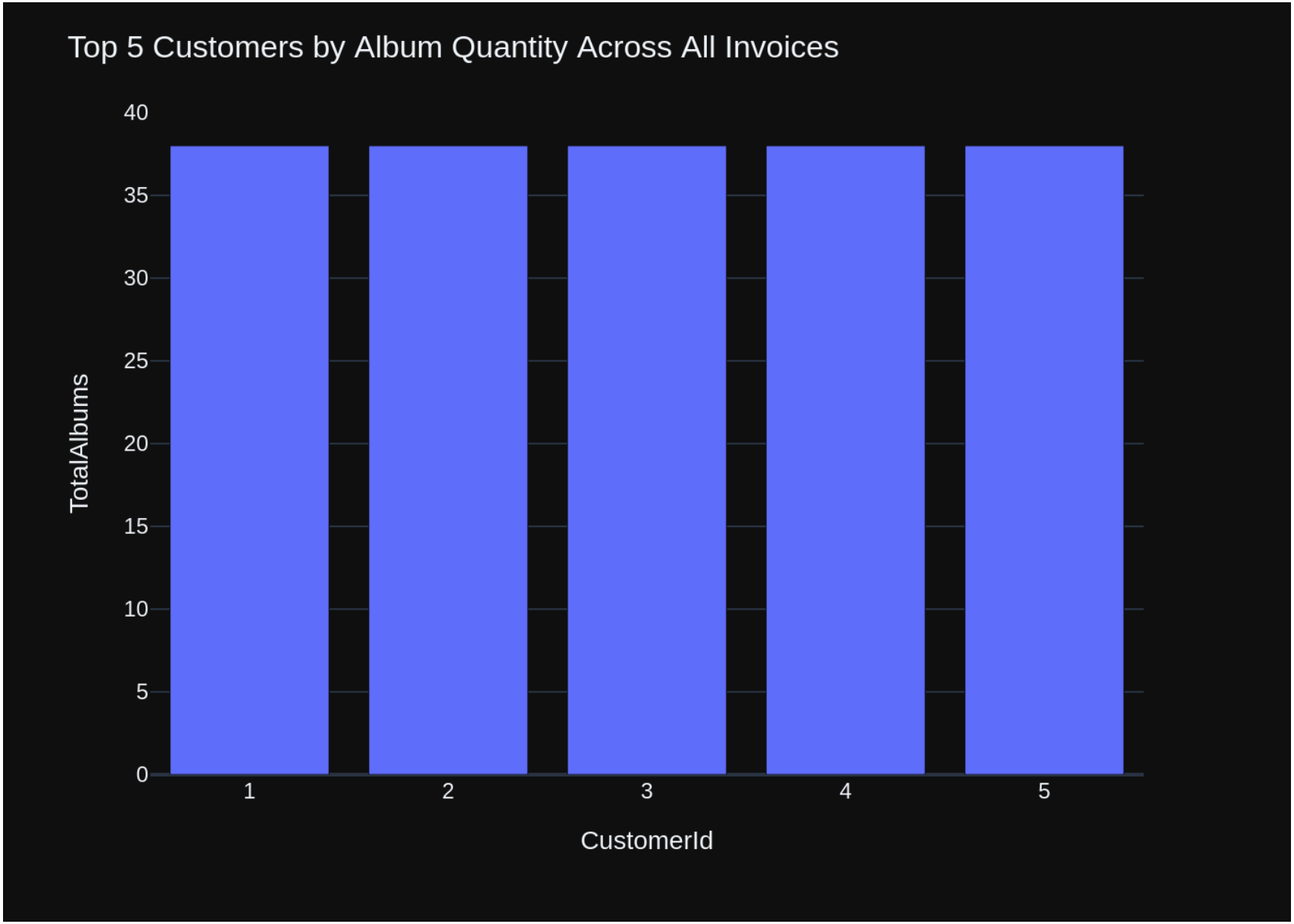
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId 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   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) 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 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 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 INDEX 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 "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   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) 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_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n   ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\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 query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   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\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 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.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 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'}, {'role': 'user',
'content': ' \n   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   Fi
nd the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assista
nt', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_it
ems" 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, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items de
tail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS Tota
lSpent\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   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 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   Find the customer with the most invoices \n'}, {'role': 'assista
nt', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoic
es" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'rol
e': 'user', 'content': ' \n   Find the customer with the most invoices \n'}, {'role': 'assistant', 'cont
ent': 'SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER
BY NumberOfInvoices DESC\nLIMIT 1\n-------------------------'}, {'role': 'user', 'content': ' \n   Find t
he top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.Track
Id, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content':
' \n   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'}]
Ollama parameters:
model=phi3: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
EY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   Uni
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
AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(22

0),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NUL
L,\r\n   FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO AC
TION,\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 ACTI
ON ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (Ar
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
ices\"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NUL
L,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(4
0),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR
(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES \"customers\" (Custo
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
\n(\r\n   ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\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 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":
" \n   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
\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
\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"}, {"role": "user", "content": " \n   Find the customer who bought the most albums in total quan
tity (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_it
ems\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"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 B

```
Y TotalAlbums DESC\nLIMIT 1\n-------------------------"}, {"role": "user", "content": "  \n     Find the to
p 5 customers who spent the most money overall, \n     \n     Hint: order total can be found on invoices ta
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"}, {"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     Find the customer with t
he most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalIn
voices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nOR
DER 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 NumberOfInvoices \n
FROM Invoices \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-------------------------"},
{"role": "user", "content": "  \n    Find the top 5 most expensive tracks (based on unit price):\n"}, {"rol
e": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.UnitPrice
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 albums in total quantity (across all invoices):\n"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T03:06:37.29021214Z', '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 5\n----------------
--------'}, 'done_reason': 'stop', 'done': True, 'total_duration': 68235322985, 'load_duration': 709720, 'p
rompt_eval_count': 2032, 'prompt_eval_duration': 56226237000, 'eval_count': 73, 'eval_duration': 1132549200
0}
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 5
-------------------------
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 5
-------------------------
   CustomerId  TotalAlbums
0           1           38
```

```
1           2           38
2           3           38
3           4           38
4           5           38
Ollama parameters:
model=phi3: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    \n
Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'\n\nThe Data
Frame was produced using this query: 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------------------------\n\nThe following is information about the resulting pandas DataFra
me 'df': \nRunning df.dtypes gives:\n CustomerId     int64\nTotalAlbums    int64\ndtype: object"}, {"role":
"user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume t
he 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': 'phi3:latest', 'created_at': '2024-06-14T03:06:58.152142826Z', 'message': {'role': 'assistant',
'content': "```python\nimport plotly.express as px\n\n# Assuming df has been preprocessed to have a single
row if needed for indicator charting\nfig = px.bar(df, x='CustomerId', y='TotalAlbums')\nfig.update_layout
(title='Top 5 Customers by Album Quantity Across All Invoices')\nfig.show()\n```"}, 'done_reason': 'stop',
'done': True, 'total_duration': 20834225016, 'load_duration': 708813, 'prompt_eval_count': 266, 'prompt_eva
l_duration': 7568197000, 'eval_count': 87, 'eval_duration': 13177411000}
```


Top 5 Customers by Album Quantity Across All Invoices
TotalAlbums
40
35
30
25
20
15
10
5
0
1
2
3
4
5
CustomerId

Out[37]:
```
('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 5\n----------------------
---',
    CustomerId  TotalAlbums
 0           1           38
 1           2           38
 2           3           38
 3           4           38
 4           5           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, 2, 3, 4, 5]),
               'xaxis': 'x',
               'y': array([38, 38, 38, 38, 38]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'title': {'text': 'Top 5 Customers by Album Quantity Across All Invoices'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalAlbums'}}}
 }))
```

```sql
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
```

In [38]:
```
question = """
     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
"""

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
ry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\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
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) REFERENCES "invoices" (InvoiceId) \r\n\t\t
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 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    Fax N
VARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\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
R,\r\n    BirthDate DATETIME,\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\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    Al
bumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(22
0),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice 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
ON,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\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 "playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    Tra
ckId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREI
GN 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_EmployeeReportsTo ON "employees" (ReportsTo)\n\n\n===Additional Context \n\nIn the
SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, p
lease 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
```

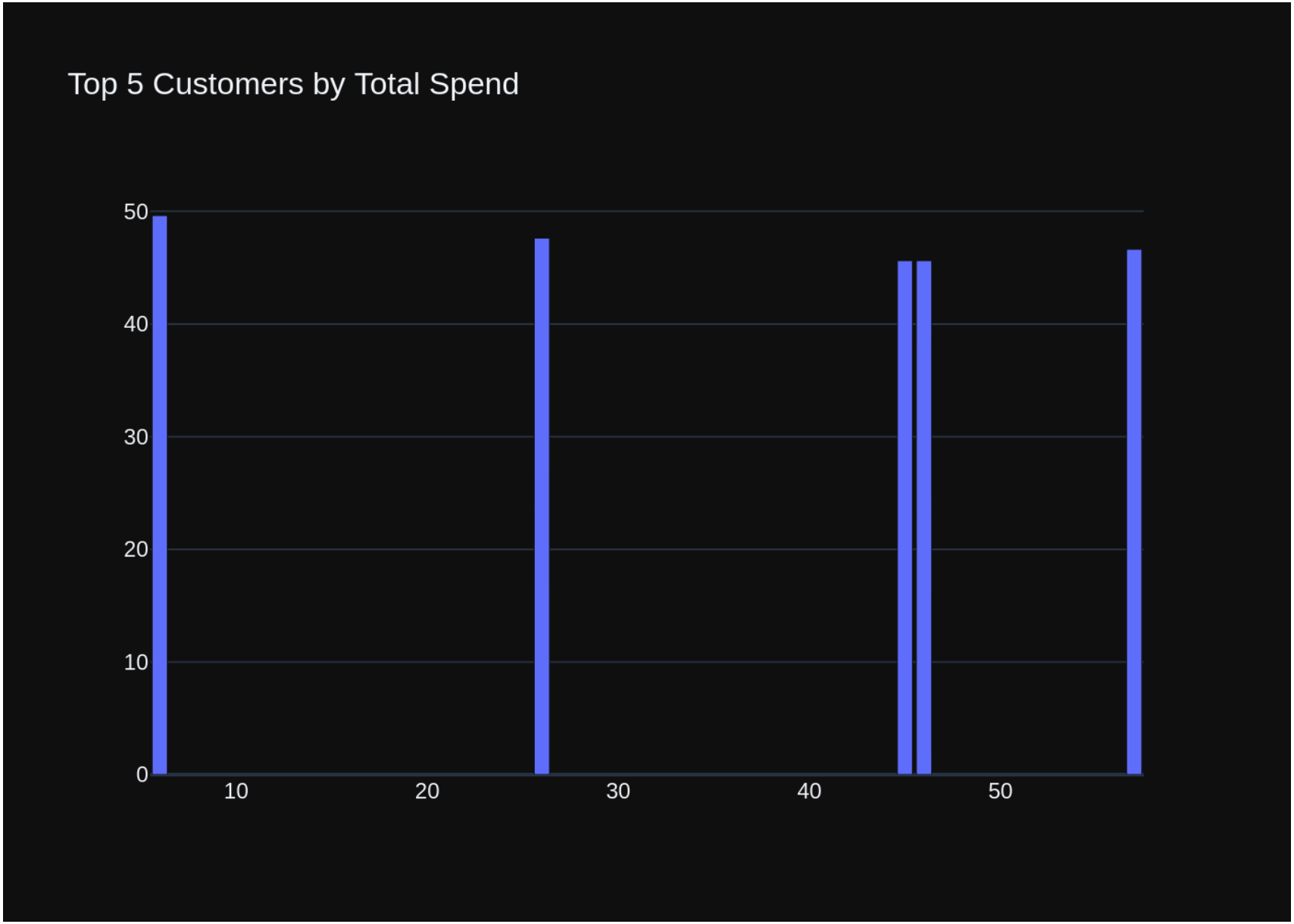
epeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': '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   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\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 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 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 all invoices):\n'}, {'role': '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   \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\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': 'user', 'content': ' \n   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 "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 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\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'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\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-------------------------'}, {'role': 'user', 'content': ' \n   Get the average invoice 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.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 using invoice_items detail table is unnecessary \n'}]
Ollama parameters:
model=phi3: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    BillingCou
ntry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FORE
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
T NULL,\r\n    Quantity 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    CustomerI
d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVAR
CHAR(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 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
City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(1
0),\r\n    Phone NVARCHAR(24),\r\n    Fax 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    Comp
oser NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\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\tON DELETE NO ACTION
ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON D
ELETE 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 NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, Tr
ackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION O
N UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTIO
N ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON \"employees\" (ReportsTo)\n\n\n===Addit
ional 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 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 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 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 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\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 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 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 all invoices):\n"}, {"role": "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 \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\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": "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 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\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"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\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-------------------------"}, {"role": "user", "content": " \n Get the average invoice 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.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 using invoice_items detail table is unnecessary \n"}]

```
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T03:08:05.28787172Z', 'message': {'role': 'assistant', 'c
ontent': '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------------------------
'}, 'done_reason': 'stop', 'done': True, 'total_duration': 67048540282, 'load_duration': 743624, 'prompt_ev
al_count': 1975, 'prompt_eval_duration': 56116255000, 'eval_count': 66, 'eval_duration': 9997177000}
SELECT c.CustomerId, SUM(i.Total) AS TotalSpent
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 5
------------------------
SELECT c.CustomerId, SUM(i.Total) AS TotalSpent
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 5
------------------------
   CustomerId  TotalSpent
0           6       49.62
1          26       47.62
2          57       46.62
3          45       45.62
4          46       45.62
Ollama parameters:
model=phi3: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(i.Total)
AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Custome
rId\nORDER BY TotalSpent DESC\nLIMIT 5\n------------------------\n\nThe following is information about the
resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId      int64\nTotalSpent    float64
\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."}]
```

```
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T03:08:23.096401435Z', 'message': {'role': 'assistant',
'content': "import plotly.graph_objs as go\n\ntrace = go.Bar(x=df['CustomerId'], y=df['TotalSpent'])\ndata
= [trace]\nlayout = go.Layout(title='Top 5 Customers by Total Spend')\nfig = go.Figure(data=data, layout=la
yout)\nfig.show()"}, 'done_reason': 'stop', 'done': True, 'total_duration': 17788626696, 'load_duration': 6
34606, 'prompt_eval_count': 261, 'prompt_eval_duration': 6887309000, 'eval_count': 80, 'eval_duration': 108
05448000}
```


Top 5 Customers by Total Spend
50
40
30
20
10
0
10
20
30
40
50

Out[38]:
```
('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-------------------------',
    CustomerId  TotalSpent
 0           6       49.62
 1          26       47.62
 2          57       46.62
 3          45       45.62
 4          46       45.62,
 Figure({
     'data': [{'type': 'bar', 'x': array([ 6, 26, 57, 45, 46]), 'y': array([49.62, 47.62, 46.62, 45.62, 4
5.62])}],
     'layout': {'template': '...', 'title': {'text': 'Top 5 Customers by Total Spend'}}
 }))
```

In [39]:
```
question = """
     Get all playlists containing at least 10 tracks and the total duration of those tracks:
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE TABLE "playlists"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\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   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 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_TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX 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   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 "genres"\r\n(\r\n   GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\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 query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   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 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\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 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   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   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 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   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 g2.GenreId\n FROM "gen
res" 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'}, {'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.N
ame 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.GenreId
\n HAVING COUNT(g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'role': 'user', 'c
ontent': ' \n   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   Hin
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   Hin
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'}, {'role': 'user', 'content': ' \n   Get all pla
ylists containing at least 10 tracks and the total duration of those tracks:\n'}]
Ollama parameters:
model=phi3: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
REATE TABLE \"playlists\"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NV
ARCHAR(120)\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackI
d 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 \"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
R,\r\n   Composer NVARCHAR(220),\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\tON DELET

E NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DE
LETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeI
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
\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tO
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\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 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\nHAVING COUNT(pt.TrackId) >= 10"}, {"role": "user", "content": " \n   List all g
enres 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.Nam
e"}, {"role": "user", "content": " \n   List all genres and the number of tracks in each genre:\n"}, {"ro
le": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres\" g\nJOIN \"tra
cks\" t ON g.GenreId = t.GenreId\nGROUP BY g.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 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   Iden
tify artists who have albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "cont
ent": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.A
rtistId\nWHERE a.ArtistId IN (\n SELECT g2.GenreId\n FROM \"genres\" g2\n JOIN \"tracks\" t2 ON g2.Genre
Id = 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 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 g2.GenreId\n FROM \"gen
res\" g2\n JOIN \"tracks\" t2 ON g2.GenreId = t2.GenreId\n GROUP BY g2.GenreId\n HAVING COUNT(g2.GenreI
d) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName"}, {"role": "user", "content": " \n   Find th
e 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\nGROUP BY c.Custo

```
merId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": "  \n    Hint: album quantity is f
ound in invoice_items, \n    \n    Find the top 5 customers who bought the most albums in total quantity (a
cross all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS Total
Albums\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": "use
r", "content": "  \n     Find the customer 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 \"i
nvoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY Total
Albums DESC\nLIMIT 1\n-------------------------"}, {"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 i.CustomerId, COUNT(ii.TrackId) AS
TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Custo
merId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "  \n     Get all playlists contai
ning at least 10 tracks and the total duration of those tracks:\n"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T03:09:33.778384012Z', 'message': {'role': 'assistant',
'content': 'SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDuration\nFROM "playlist_track" ptt JOIN "trac
ks" t ON ptt.TrackId = t.TrackId\nGROUP BY p.PlaylistId\nHAVING COUNT(ptt.TrackId) >= 10\nORDER BY TotalDur
ation DESC\nLIMIT 5'}, 'done_reason': 'stop', 'done': True, 'total_duration': 70569147454, 'load_duration':
681478, 'prompt_eval_count': 2020, 'prompt_eval_duration': 56950140000, 'eval_count': 80, 'eval_duration':
12939123000}
SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDuration
FROM "playlist_track" ptt JOIN "tracks" t ON ptt.TrackId = t.TrackId
GROUP BY p.PlaylistId
HAVING COUNT(ptt.TrackId) >= 10
ORDER BY TotalDuration DESC
LIMIT 5
SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDuration
FROM "playlist_track" ptt JOIN "tracks" t ON ptt.TrackId = t.TrackId
GROUP BY p.PlaylistId
HAVING COUNT(ptt.TrackId) >= 10
ORDER BY TotalDuration DESC
LIMIT 5
Couldn't run sql:  Execution failed on sql 'SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDuration
FROM "playlist_track" ptt JOIN "tracks" t ON ptt.TrackId = t.TrackId
GROUP BY p.PlaylistId
HAVING COUNT(ptt.TrackId) >= 10
ORDER BY TotalDuration DESC
LIMIT 5': no such column: p.PlaylistId
```

In [40]:
```
question = """
     Identify artists who have albums with tracks appearing in multiple genres:
```

```
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "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   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_AlbumArtistId 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   FOREIGN 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   GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE TABLE "artists"\r\n(\r\n   ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\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   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\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 sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   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 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'}, {'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 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   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 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   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.Ar

tistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreId) 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': '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   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\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 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 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.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 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 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 5\n-------------------------'}, {'role': 'user', 'content': ' \n   Identify artists who have albums with tracks appearing in multiple genres:\n\n\n'}]
Ollama parameters:
model=phi3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE \"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   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_AlbumArtistId 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   FOREIGN 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   GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\n\nCREATE TABLE \"artists\"\r\n(\r\n   ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\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   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\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 sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n   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 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"}, {"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 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\nORDER 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   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\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"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", "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"}, {"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   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\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 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 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.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 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 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 5\n------------------------"}, {"role": "user", "content": " \n   Identify artists who have albums with tracks appearing in multiple genres:\n\n\n"}]
Ollama Response:
{'model': 'phi3:latest', 'created_at': '2024-06-14T03:10:59.204861487Z', 'message': {'role': 'assistant', 'content': 'SELECT DISTINCT a.ArtistId, a.Name AS ArtistName\nFROM "artists" a\nWHERE EXISTS (\n SELECT g2.GenreId\n FROM "genres" g2\n JOIN "tracks" t ON g2.GenreId = t.GenreId\n WHERE a.ArtistId = (\n   SELECT al.ArtistId\n   FROM "albums" al\n   WHERE al.AlbumId IN (SELECT t.AlbumId FROM "tracks" t)\n   AND EXISTS (\n   SELECT 1\n   FROM "genres" g\n   WHERE t.GenreId = g.GenreId\n   AND COUNT(g.GenreId) > 1\n   )\n)\n)\nORDER BY ArtistName;'}, 'done_reason': 'stop', 'done': True, 'total_duration': 85379663186, 'load_duration': 838263, 'prompt_eval_count': 1894, 'prompt_eval_duration': 561252500000, 'eval_count': 178, 'eval_duration': 28564817000}

```
SELECT DISTINCT a.ArtistId, a.Name AS ArtistName
FROM "artists" a
WHERE EXISTS (
  SELECT g2.GenreId
  FROM "genres" g2
  JOIN "tracks" t ON g2.GenreId = t.GenreId
  WHERE a.ArtistId = (
    SELECT al.ArtistId
    FROM "albums" al
    WHERE al.AlbumId IN (SELECT t.AlbumId FROM "tracks" t)
      AND EXISTS (
        SELECT 1
        FROM "genres" g
        WHERE t.GenreId = g.GenreId
          AND COUNT(g.GenreId) > 1
    )
  )
)
ORDER BY ArtistName;
```

Output from LLM: SELECT DISTINCT a.ArtistId, a.Name AS ArtistName
FROM "artists" a

```
WHERE EXISTS (
  SELECT g2.GenreId
  FROM "genres" g2
  JOIN "tracks" t ON g2.GenreId = t.GenreId
  WHERE a.ArtistId = (
    SELECT al.ArtistId
    FROM "albums" al
    WHERE al.AlbumId IN (SELECT t.AlbumId FROM "tracks" t)
      AND EXISTS (
        SELECT 1
        FROM "genres" g
        WHERE t.GenreId = g.GenreId
          AND COUNT(g.GenreId) > 1
    )
  )
)
ORDER BY ArtistName;
Extracted SQL: SELECT DISTINCT a.ArtistId, a.Name AS ArtistName
FROM "artists" a
WHERE EXISTS (
  SELECT g2.GenreId
  FROM "genres" g2
  JOIN "tracks" t ON g2.GenreId = t.GenreId
  WHERE a.ArtistId = (
    SELECT al.ArtistId
    FROM "albums" al
    WHERE al.AlbumId IN (SELECT t.AlbumId FROM "tracks" t)
      AND EXISTS (
        SELECT 1
        FROM "genres" g
        WHERE t.GenreId = g.GenreId
          AND COUNT(g.GenreId) > 1
    )
  )
)
ORDER BY ArtistName
SELECT DISTINCT a.ArtistId, a.Name AS ArtistName
FROM "artists" a
WHERE EXISTS (
  SELECT g2.GenreId
  FROM "genres" g2
  JOIN "tracks" t ON g2.GenreId = t.GenreId
```

```
  WHERE a.ArtistId = (
    SELECT al.ArtistId
    FROM "albums" al
    WHERE al.AlbumId IN (SELECT t.AlbumId FROM "tracks" t)
      AND EXISTS (
        SELECT 1
        FROM "genres" g
        WHERE t.GenreId = g.GenreId
          AND COUNT(g.GenreId) > 1
    )
  )
)
ORDER BY ArtistName
Couldn't run sql:  Execution failed on sql 'SELECT DISTINCT a.ArtistId, a.Name AS ArtistName
FROM "artists" a
WHERE EXISTS (
  SELECT g2.GenreId
  FROM "genres" g2
  JOIN "tracks" t ON g2.GenreId = t.GenreId
  WHERE a.ArtistId = (
    SELECT al.ArtistId
    FROM "albums" al
    WHERE al.AlbumId IN (SELECT t.AlbumId FROM "tracks" t)
      AND EXISTS (
        SELECT 1
        FROM "genres" g
        WHERE t.GenreId = g.GenreId
          AND COUNT(g.GenreId) > 1
    )
  )
)
ORDER BY ArtistName': misuse of aggregate function COUNT()
```

Check completion time

In [ ]:

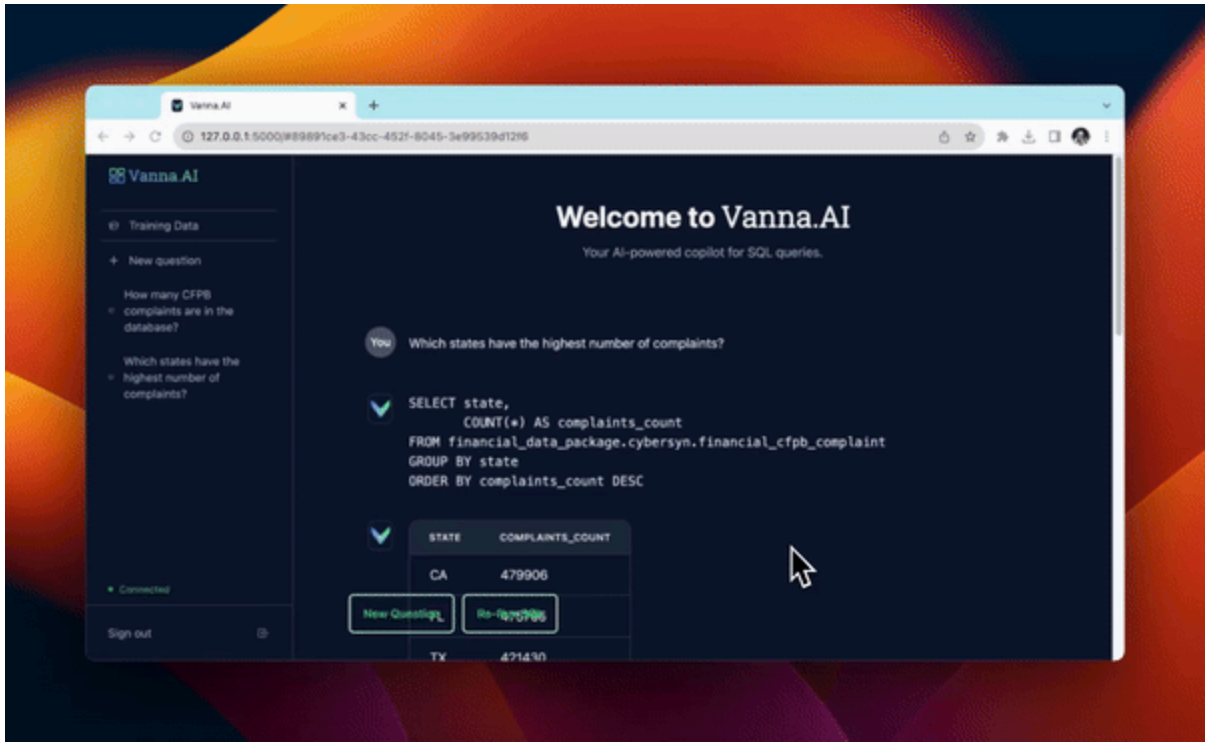
In [41]:
```
ts_stop = time()
```

```
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 'phi3' LLM took : 2083.60 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