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 = 'mistral'
```

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	15ac2fa5-0191-5f4b-95fe-f1cc0e4c1791-sql	\n Find the customer with the most invoi...	SELECT CustomerId, COUNT(InvoiceId) AS NumberO...	sql
5	17d893d5-1417-5ba3-a5ca-9f6ce15a727f-sql	\n Identify artists who have albums with...	SELECT a.ArtistId, a.Name AS ArtistName\nFROM ...	sql
6	27a11d7d-78c7-5027-ab98-36e4ee8f791c-sql	\n Find the top 5 customers who spent th...	SELECT c.CustomerId, SUM(i.Total) AS TotalSpen...	sql
7	3013d1b4-feb2-519d-bfb9-114500436e3d-sql	\n Find the customer with the most invoi...	SELECT c.CustomerId, COUNT(i.InvoiceId) AS Tot...	sql
8	32b99e7b-31ab-55d8-8431-fb010fa7af85-sql	\n Find the top 5 customers who spent th...	SELECT c.CustomerId, SUM(i.Total) AS TotalSpen...	sql
9	4033dd5b-7f55-5895-a91f-b03a091843cc-sql	\n List all invoices with a total exceedi...	SELECT *\nFROM "invoices" \nWHERE (Total > 10)...	sql
10	49e67df3-a604-51f8-ad01-b8f5a2043eac-sql	\n Get the total number of invoices for e...	SELECT c.CustomerId, COUNT(i.InvoiceId) AS Tot...	sql
11	4f130cef-6bfa-5e99-9455-9d6907e4f932-sql	\n Find all invoices since 2010 and the t...	SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmo...	sql
12	584873f8-1904-50f1-8f80-7ccf08059264-sql	\n List all customers from Canada and th...	SELECT c.Email, c.Country\nFROM "customers" c\...	sql
13	5b0b32a6-7f1d-544f-8c5b-9448cbc635ac-sql	\n List all genres and the number of trac...	SELECT g.Name, COUNT(t.GenreId) AS TotalTracks...	sql
14	6bed484b-9a80-57f4-ad89-5f775b5df252-sql	\n Get the average invoice total for each...	SELECT c.CustomerId, AVG(i.Total) AS AverageIn...	sql
15	6f22268c-5062-5f11-ba2d-8555f06b409d-sql	\n Find all tracks with a name containing...	SELECT * \nFROM "tracks" \nWHERE LOWER(Name) L...	sql
16	701ab26b-0be7-59be-ad5f-	\n Find the top 5 most	SELECT Name, UnitPrice\nFROM	sql

	id	question	content	training_data_type
	8890609364d7-sql	expensive tracks (...	"tracks"\nORDER B...	
17	70b4f686-c71b-5ee8-9458-6bbc776349bf-sql	\n Find all invoices since 2010 and the t...	SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmo...	sql
18	9a396a33-ecea-51a8-bd05-28f58a86eb86-sql	\n Hint: album quantity is found in invoi...	SELECT c.CustomerId, COUNT(ii.TrackId) AS Tota...	sql
19	9a9c970b-b94c-5f22-b54c-b86921a38b65-sql	\n Identify artists who have albums with...	SELECT a.ArtistId, a.Name AS ArtistName\nFROM ...	sql
20	9f03ddfc-84f0-509a-a8fd-7d0371ac0dde-sql	\n Get the average invoice total for each...	SELECT c.CustomerId, AVG(i.Total) AS AverageIn...	sql
21	a7185c88-7417-5b75-a52e-4eaef5f9deca-sql	\n List all albums and their correspondin...	SELECT a.Title, a.ArtistId, ar.Name AS ArtistN...	sql
22	aea89953-21b2-55d1-9dda-431ee6033c3d-sql	\n List all invoices with a total exceedi...	SELECT * \nFROM "invoices" \nWHERE Total > 10.00	sql
23	b3af5404-4691-599c-9782-f97c64708f87-sql	\n List all customers from Canada and th...	SELECT c.CustomerId, c.Email, SUM(i.Total) AS ...	sql
24	bb554ac0-f9e9-51b0-819f-e0e01db8650c-sql	\n Find the total number of invoices per ...	SELECT Customer.Country, COUNT(invoice.Invoice...	sql
25	c7de9fac-1104-5409-b17a-73b533b767d5-sql	\n List all employees and their reporting...	SELECT e1.*, CONCAT(e2.FirstName, ' ', e2.Last...	sql
26	d1d70c18-f5d9-5970-a32c-914deeca1087-sql	\n Find the customer who bought the most...	SELECT c.CustomerId, COUNT(ii.TrackId) AS Tota...	sql
27	d6a752e9-6f0c-5681-861f-7537fc183dc5-sql	\n Hint: album quantity is found in invoi...	SELECT i.CustomerId, COUNT(ii.TrackId) AS Tota...	sql
28	d8a2f948-dffa-5524-a5f9-174cc1a8da73-sql	Can you list all tables in the SQLite database...	SELECT name FROM sqlite_master WHERE type='table'	sql
29	d8a37163-5ce5-58cd-a316-ea5598d44d27-sql	what are the top 5 countries that customers co...	SELECT c.Country, COUNT(*) AS TotalCustomers\n...	sql
30	dd282d7c-a4ef-5e3a-87e0-cb45fac50808-sql	\n Find the total number of invoices per ...	SELECT i.BillingCountry, COUNT(*) AS TotalInvo...	sql
31	e3fce6ca-b370-5059-92ea-a9745a400b86-sql	\n Find the customer who bought the most...	SELECT i.CustomerId, COUNT(ii.TrackId) AS Tota...	sql
32	e7c4b3aa-664f-5f87-8b25-449a4482f3fd-sql	\n Get all playlists containing at least...	SELECT pt.PlaylistId, p.Name AS PlaylistName, ...	sql

	id	question	content	training_data_type
33	f33f8cb6-1b12-5ea7-8d9a-aef8166b9970-sql	\n Find the top 5 most expensive tracks (...	SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "t...	sql
34	f626b681-4d8f-563a-beee-1ea759baaa82-sql	\n List all genres and the number of trac...	SELECT g.Name, COUNT(t.GenreId) AS TotalTracks...	sql
35	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
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-	None	CREATE INDEX IFK_EmployeeReportsTo	ddl

	id	question	content	training_data_type
	ca8a80e3af3d-ddl		ON "employe...	
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
Add of existing embedding ID: d8a2f948-dffa-5524-a5f9-174cc1a8da73-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
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

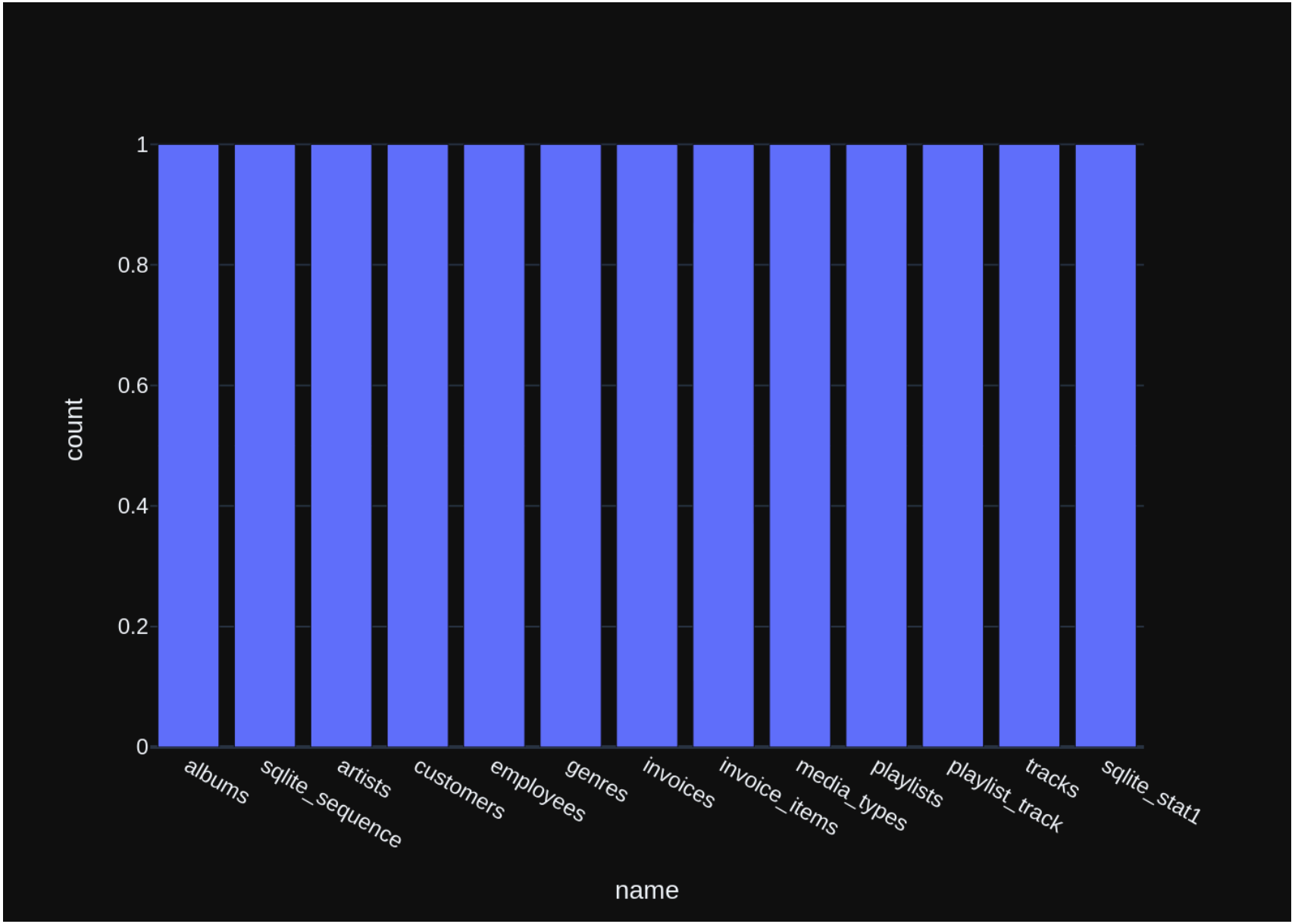
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE TABLE 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.CustomerId, c.Email,

SUM(i.Total) AS TotalSpent\nFROM Customers c\nJOIN Invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Cu
stomerId\nORDER BY TotalSpent DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   List all customers fro
m 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   List all genres and the
number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS T
otalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTrack
s DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the customer who bought the most albums in to
tal quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.T
rackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice
_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'ro
le': 'user', 'content': ' \n   List all genres and the number of tracks in each genre:\n'}, {'role': 'ass
istant', '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 r
eporting manager's name (if any):\n"}, {'role': 'assistant', 'content': 'SELECT e.FirstName, e.LastName, m
t.FirstName AS ManagerFirstName, mt.LastName AS ManagerLastName\nFROM "employees" e\nLEFT JOIN "employees"
mt ON e.ReportsTo = 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 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.InvoiceI
d = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content':
'Can you list all tables in the SQLite database catalog?'}]
Ollama parameters:
model=mistral: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.CustomerId, c.Email, SUM(i.Total) AS TotalSpent\nFROM Customers c\nJOIN Invoice
s i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 1"}, {"role": "u
ser", "content": " \n   List all customers from Canada and their email addresses:\n"}, {"role": "assista
nt", "content": "SELECT c.Email, c.Country\nFROM \"customers\" c\nWHERE c.Country = 'Canada'"}, {"role": "u
ser", "content": " \n   List all genres and the number of tracks in each genre:\n"}, {"role": "assistan
t", "content": "SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.G
enreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks 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\nGRO
UP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n   List all gen
res and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.G
enreId) 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 ManagerFirstName, mt.Last
Name AS ManagerLastName\nFROM \"employees\" e\nLEFT JOIN \"employees\" mt ON e.ReportsTo = mt.EmployeeId"},

```
{"role": "user", "content": "  \n    Hint: album quantity is found in invoice_items, \n    \n    Find the t
op 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistan
t", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoice
s\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.C
ustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "Can you list all tables in th
e SQLite database catalog?"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:12:22.618278646Z', 'message': {'role': 'assistan
t', 'content': " SELECT name FROM sqlite_master WHERE type='table'"}, 'done_reason': 'stop', 'done': True,
'total_duration': 65818536765, 'load_duration': 823506201, 'prompt_eval_count': 1898, 'prompt_eval_duratio
n': 62204714000, 'eval_count': 14, 'eval_duration': 2192397000}
 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=mistral: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 followin
g is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n name    object\ndt
ype: 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 data
frame, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the cod
e."}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:12:54.935966541Z', 'message': {'role': 'assistan
```

t', 'content': ' Here\'s a simple way to create a plotly figure if the DataFrame \'df\' has more than one r
ow, and an indicator chart if it only contains one row:\n\n```python\nimport plotly.express as px\nimport n
umpy as np\n\nif df.shape[0] > 1:\n   fig = px.bar(df, x="name")\nelse:\n   fig = go.Indicator(\n
value=np.random.uniform(0, 1),\n   domain={"x": [0, 1], "y": [0, 1]},\n   title={\'text\': df.ilo
c[0, 0]}\n   )\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 32290059343, 'load_duratio
n': 47695498, 'prompt_eval_count': 164, 'prompt_eval_duration': 6091584000, 'eval_count': 153, 'eval_durati
on': 26101590000}


count
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
name

Out[17]:
```
(" SELECT name FROM sqlite_master WHERE type='table'",
                name
0            albums
1   sqlite_sequence
2           artists
3         customers
4         employees
5            genres
6          invoices
7     invoice_items
8       media_types
9         playlists
10   playlist_track
11           tracks
12    sqlite_stat1,
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'name=%{x}<br>count=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'v',
              'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
              'x': array(['albums', 'sqlite_sequence', 'artists', 'customers', 'employees',
                          'genres', 'invoices', 'invoice_items', 'media_types', 'playlists',
                          'playlist_track', 'tracks', 'sqlite_stat1'], dtype=object),
              'xaxis': 'x',
              'y': array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]),
              'yaxis': 'y'}],
    'layout': {'barmode': 'relative',
               'legend': {'tracegroupgap': 0},
               'margin': {'t': 60},
               'template': '...',
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'name'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'count'}}}
})
```

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\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.CustomerI
d = 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', 'content': ' \n   Find the customer with the mo
st invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoice
s\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY To
talInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   Find the customer who bought the most al
bums 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\nJOI
N "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   Find the customer who
bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SEL
ECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.Invoice
Id = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n-------------------------'},
{'role': 'user', 'content': ' \n   Hint: album quantity is found in invoice_items, \n   \n   Find the t
op 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_ite
ms" 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'}, {'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   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', 'conten
t': "which table stores customer's orders"}]
Ollama parameters:
model=mistral: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\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 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   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 \"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 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 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   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": "which table stores customer's orders"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:14:25.090297461Z', 'message': {'role': 'assistant', 'content': ' The customer\'s orders are stored in the "invoices" table, but we need to join with "invoice_items" to get the album quantity information.'}, 'done_reason': 'stop', 'done': True, 'total_duration': 89730453866, 'load_duration': 665525, 'prompt_eval_count': 1995, 'prompt_eval_duration': 82472815000, 'eval_count': 36, 'eval_duration': 6370720000}
 The customer's orders are stored in the "invoices" table, but we need to join with "invoice_items" to get the album quantity information.

```
 The customer's orders are stored in the "invoices" table, but we need to join with "invoice_items" to get
the album quantity information.
Couldn't run sql:  Execution failed on sql ' The customer's orders are stored in the "invoices" table, but
we need to join with "invoice_items" to get the album quantity information.': 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': ' \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': '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 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\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 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 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'}, {'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': 'How many customers are there'}]
Ollama parameters:
model=mistral: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   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 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": " \n   Find the customer with the most invoice
s \n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM In
voices \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-------------------------"}, {"rol
e": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) F
ROM \"customers\""}, {"role": "user", "content": " \n   Find the top 5 customers who spent the most mone
y 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 To
talSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\n
ORDER BY TotalSpent DESC\nLIMIT 5\n-------------------------"}, {"role": "user", "content": " \n   Find
the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.In
voiceId) 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 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.

```
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    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.CustomerId"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"ro
le": "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.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": "How many customers are there"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:16:02.123916141Z', 'message': {'role': 'assistan
t', 'content': ' To get the total number of customers, you can use the following query:\n```\nSELECT COUNT
(*) AS TotalCustomers FROM "customers";\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 9699
3535688, 'load_duration': 866924, 'prompt_eval_count': 1990, 'prompt_eval_duration': 89452204000, 'eval_cou
nt': 37, 'eval_duration': 6841940000}
 To get the total number of customers, you can use the following query:
```
SELECT COUNT(*) AS TotalCustomers FROM "customers";
```
Output from LLM:  To get the total number of customers, you can use the following query:
```
SELECT COUNT(*) AS TotalCustomers FROM "customers";
```
Extracted SQL: SELECT COUNT(*) AS TotalCustomers FROM "customers"
SELECT COUNT(*) AS TotalCustomers FROM "customers"
   TotalCustomers
0              59
Ollama parameters:
model=mistral:latest,
options={},
keep_alive=None
Prompt Content:
```

[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: 'How many customers are there'\n\nThe DataFrame was produced using this query: SELECT COUNT(*) AS TotalCustomers FROM \"customers\"\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n TotalCustomers   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': 'mistral:latest', 'created_at': '2024-06-14T10:16:40.125113337Z', 'message': {'role': 'assistant', 'content': ' Here\'s the Python code to create a Plotly chart for your DataFrame `df`:\n\n```python\nimport plotly.graph_objects as go\n\nfig = go.Figure(data=go.Indicator(\n   mode="number+delta",\n   value=df["TotalCustomers"].item(),\n   delta={"reference": df["TotalCustomers"].iloc[0]},\n   domain={\'x\': [0, 1], \'y\': [0, 1]}\n))\n\nfig.write_image("customers.png")\n```\n\nThis code will create a Plotly figure with the total number of customers and the delta (if there are multiple records). The chart will be saved as "customers.png".'}, 'done_reason': 'stop', 'done': True, 'total_duration': 37975356259, 'load_duration': 43838922, 'prompt_eval_count': 137, 'prompt_eval_duration': 6681691000, 'eval_count': 166, 'eval_duration': 31200200000}


59

Out[19]:
```
('SELECT COUNT(*) AS TotalCustomers FROM "customers"',
   TotalCustomers
0              59,
Figure({
    'data': [{'delta': {'reference': 59},
              'domain': {'x': [0, 1], 'y': [0, 1]},
              'mode': 'number+delta',
              'type': 'indicator',
              'value': 59}],
    'layout': {'template': '...'}
}))
```

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.Count
ry, COUNT(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT
5'}, {'role': 'user', 'content': '  \n     Find the top 5 customers who spent the most money overall, \n
\n     Hint: order total can be found on invoices table, calculation using invoice_items detail table is un
necessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "cu
stomers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DE
SC\nLIMIT 5\n-------------------------'}, {'role': 'user', 'content': '  \n     Find the top 5 customers wh
o spent the most money overall, \n     \n     Hint: order total can be found on invoices table, calculation
using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerI
d, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP
BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n     List all custom
ers 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    Find the total n
umber of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT Customer.Country, COUNT(invoic
e.InvoiceId) AS InvoiceCount\nFROM "customers" customer\nJOIN "invoices" invoice ON customer.CustomerId = i
nvoice.CustomerId\nGROUP BY Country'}, {'role': 'user', 'content': '  \n    Hint: album quantity is found i
n 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     Find the cust
omer 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    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     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    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': 'what are the top 5 countries that
customers come from?'}]
Ollama parameters:
model=mistral: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   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 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 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\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   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   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   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   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   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": "what are the top 5 countries that customers come from?"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:18:33.018772124Z', 'message': {'role': 'assistant', 'content': ' To answer this question, we first need to find the unique countries for customers and then

count the number of customers per country. Here\'s an intermediate SQL query to obtain the unique countrie
s:\n\n   SELECT DISTINCT BillingCountry AS Country FROM "customers"\n\nAfter obtaining the list of distinc
t countries, you can create a subquery and use it to get the top 5 countries based on the number of custome
rs per country.\n\n   SELECT c.BillingCountry, COUNT(c.CustomerId) as NumberOfCustomers\n   FROM "custome
rs" c\n   GROUP BY c.BillingCountry\n   ORDER BY NumberOfCustomers DESC\n   LIMIT 5'}, 'done_reason': 's
top', 'done': True, 'total_duration': 112796294203, 'load_duration': 689530, 'prompt_eval_count': 1727, 'pr
ompt_eval_duration': 83272089000, 'eval_count': 152, 'eval_duration': 28551859000}
 To answer this question, we first need to find the unique countries for customers and then count the numbe
r of customers per country. Here's an intermediate SQL query to obtain the unique countries:

```
    SELECT DISTINCT BillingCountry AS Country FROM "customers"
```

After obtaining the list of distinct countries, you can create a subquery and use it to get the top 5 count
ries based on the number of customers per country.

```
    SELECT c.BillingCountry, COUNT(c.CustomerId) as NumberOfCustomers
    FROM "customers" c
    GROUP BY c.BillingCountry
    ORDER BY NumberOfCustomers DESC
    LIMIT 5
```
 To answer this question, we first need to find the unique countries for customers and then count the numbe
r of customers per country. Here's an intermediate SQL query to obtain the unique countries:

```
    SELECT DISTINCT BillingCountry AS Country FROM "customers"
```

After obtaining the list of distinct countries, you can create a subquery and use it to get the top 5 count
ries based on the number of customers per country.

```
    SELECT c.BillingCountry, COUNT(c.CustomerId) as NumberOfCustomers
    FROM "customers" c
    GROUP BY c.BillingCountry
    ORDER BY NumberOfCustomers DESC
    LIMIT 5
```
Couldn't run sql: Execution failed on sql ' To answer this question, we first need to find the unique coun
tries for customers and then count the number of customers per country. Here's an intermediate SQL query to
obtain the unique countries:

```
    SELECT DISTINCT BillingCountry AS Country FROM "customers"
```

After obtaining the list of distinct countries, you can create a subquery and use it to get the top 5 count
ries based on the number of customers per country.

```
SELECT c.BillingCountry, COUNT(c.CustomerId) as NumberOfCustomers
FROM "customers" c
GROUP BY c.BillingCountry
ORDER BY NumberOfCustomers DESC
LIMIT 5': near "To": 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   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   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   List all albums and their corresponding artist names \n'}]
Ollama parameters:
model=mistral: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

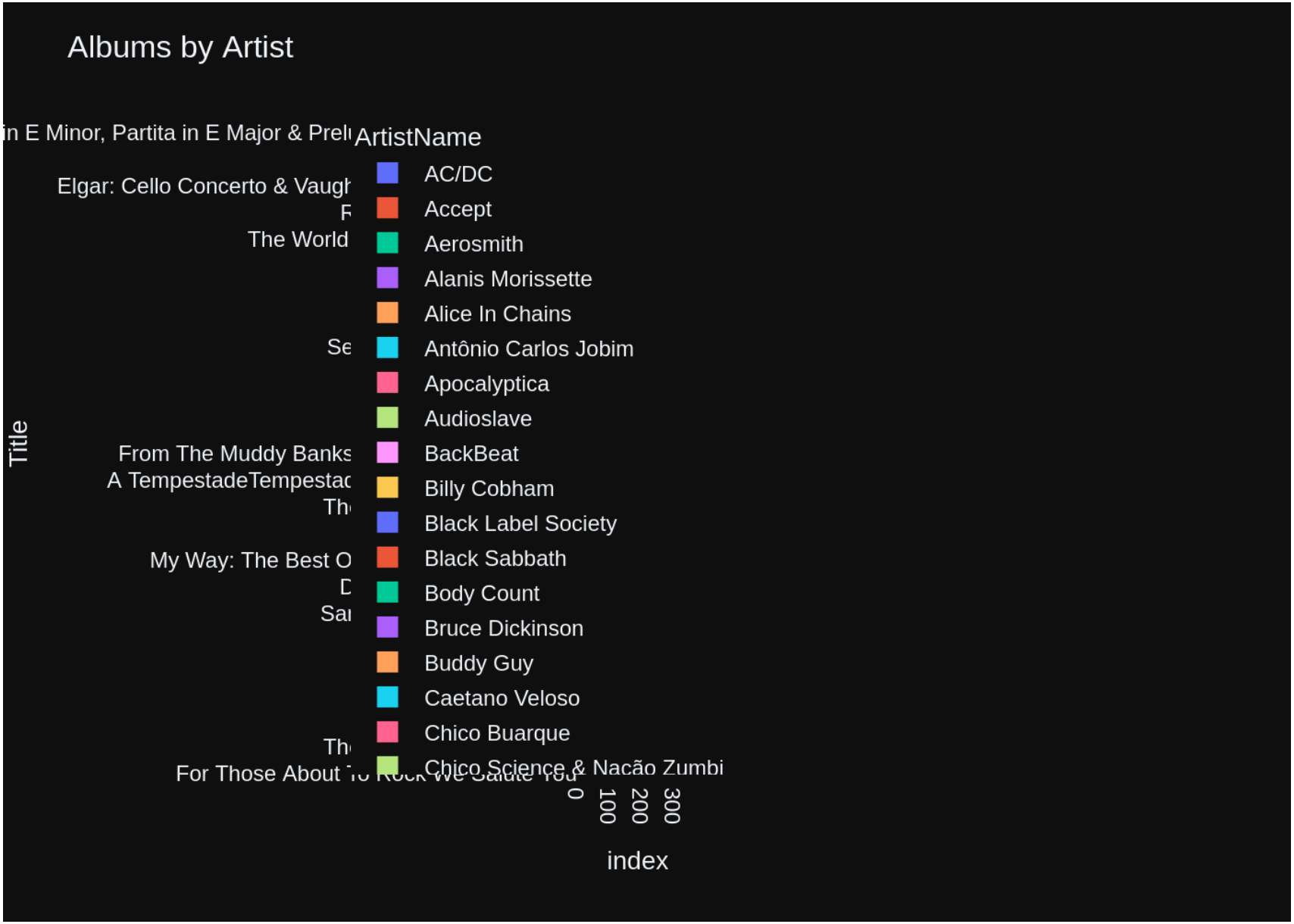
\"tracks\" (GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\n\nCREATE TAB
LE \"playlists\"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(12
0)\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n   GenreId INTEGER PRIMARY KEY AUTOINCREMENT 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    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 \"invo
ice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n--
------------------------"}, {"role": "user", "content": "  \n     Find the customer who bought the most albu
ms in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT i.CustomerId, CO
UNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceI
d\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n------------------------"}, {"role": "user",
"content": "  \n    List all albums and their corresponding artist names  \n"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:20:12.111215978Z', 'message': {'role': 'assistan
t', 'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = a
r.ArtistId\n------------------------'}, 'done_reason': 'stop', 'done': True, 'total_duration': 9905248944
0, 'load_duration': 856679, 'prompt_eval_count': 2024, 'prompt_eval_duration': 90308764000, 'eval_count': 4
2, 'eval_duration': 8095748000}
SELECT a.Title, ar.Name AS ArtistName
FROM "albums" a
JOIN "artists" ar ON a.ArtistId = ar.ArtistId
------------------------
SELECT a.Title, ar.Name AS ArtistName
FROM "albums" a
JOIN "artists" ar ON a.ArtistId = ar.ArtistId
------------------------
                                                 Title  \
0                For For Those About To Rock We Salute You
1                                     Balls to the Wall
2                                     Restless and Wild
3                                     Let There Be Rock
4                                              Big Ones
..                                                  ...
342                              Respighi:Pines of Rome
343  Schubert: The Late String Quartets & String Qu...
344                                 Monteverdi: L'Orfeo
345                               Mozart: Chamber Music
346  Koyaanisqatsi (Soundtrack from the Motion Pict...

                                            ArtistName
0                                                AC/DC
1                                               Accept
2                                               Accept
3                                                AC/DC
```

```
4                                          Aerosmith
..                                               ...
342                                  Eugene Ormandy
343                            Emerson String Quartet
344  C. Monteverdi, Nigel Rogers - Chiaroscuro; Lon...
345                                   Nash Ensemble
346                             Philip Glass Ensemble

[347 rows x 2 columns]
Ollama parameters:
model=mistral: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, ar.Name AS ArtistName\nFROM \"albums\"
a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId\n-------------------------\n\nThe following is informati
on about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Title         object\nArtistName
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 -- j
ust the code."}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:21:04.009201991Z', 'message': {'role': 'assistan
t', 'content': ' Here\'s a simple bar chart using Plotly Express for your DataFrame:\n\n```python\nimport p
lotly.express as px\n\nfig = px.bar(df, x=df.index, y=\'Title\', color=\'ArtistName\')\nfig.update_layout(t
itle="Albums by Artist")\nfig.show()\n```\n\nIn case there is only one value in the DataFrame:\n\n```python
\nimport plotly.graph_objects as go\n\nif len(df) == 1:\n    fig = go.Indicator(\n        domain={"x": [0,
1], "y": [0, 1]},\n        value=df.iloc[0][\'Title\'],\n        title={\'text\': df.iloc[0][\'ArtistName
\']}\n    )\nelse:\n    fig = px.bar(df, x=df.index, y=\'Title\', color=\'ArtistName\')\nfig.update_layout
(title="Albums by Artist")\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 51869
613443, 'load_duration': 793198, 'prompt_eval_count': 174, 'prompt_eval_duration': 7804172000, 'eval_coun
t': 236, 'eval_duration': 43969330000}
```


Albums by Artist
in E Minor, Partita in E Major & Preli
Elgar: Cello Concerto & Vaugh
F
The World
Se
From The Muddy Banks
A TempestadeTempestad
Th
My Way: The Best O
D
Sa
Th
For Those About
Title
ArtistName
AC/DC
Accept
Aerosmith
Alanis Morissette
Alice In Chains
Antônio Carlos Jobim
Apocalyptica
Audioslave
BackBeat
Billy Cobham
Black Label Society
Black Sabbath
Body Count
Bruce Dickinson
Buddy Guy
Caetano Veloso
Chico Buarque
Chico Science & Nação Zumbi
0
100
200
300
index

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

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

 [347 rows x 2 columns],
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'ArtistName=AC/DC<br>index=%{x}<br>Title=%{y}<extra></extra>',
               'legendgroup': 'AC/DC',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': 'AC/DC',
               'offsetgroup': 'AC/DC',
               'orientation': 'h',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([0, 3]),
               'xaxis': 'x',
```

```
 'y': array(['For Those About To Rock We Salute You', 'Let There Be Rock'],
           dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Accept<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Accept',
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': 'Accept',
 'offsetgroup': 'Accept',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([1, 2]),
 'xaxis': 'x',
 'y': array(['Balls to the Wall', 'Restless and Wild'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Aerosmith<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Aerosmith',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'Aerosmith',
 'offsetgroup': 'Aerosmith',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([4]),
 'xaxis': 'x',
 'y': array(['Big Ones'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Alanis Morissette<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Alanis Morissette',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Alanis Morissette',
 'offsetgroup': 'Alanis Morissette',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([5]),
```

```
                      'xaxis': 'x',
                      'y': array(['Jagged Little Pill'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=Alice In Chains<br>index=%{x}<br>Title=%{y}<extra></extra>',
                      'legendgroup': 'Alice In Chains',
                      'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
                      'name': 'Alice In Chains',
                      'offsetgroup': 'Alice In Chains',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([6]),
                      'xaxis': 'x',
                      'y': array(['Facelift'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=Antônio Carlos Jobim<br>index=%{x}<br>Title=%{y}<extra></extra
           >',
                      'legendgroup': 'Antônio Carlos Jobim',
                      'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
                      'name': 'Antônio Carlos Jobim',
                      'offsetgroup': 'Antônio Carlos Jobim',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([ 7, 33]),
                      'xaxis': 'x',
                      'y': array(['Warner 25 Anos', 'Chill: Brazil (Disc 2)'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=Apocalyptica<br>index=%{x}<br>Title=%{y}<extra></extra>',
                      'legendgroup': 'Apocalyptica',
                      'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
                      'name': 'Apocalyptica',
                      'offsetgroup': 'Apocalyptica',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
```

```
 'x': array([8]),
 'xaxis': 'x',
 'y': array(['Plays Metallica By Four Cellos'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Audioslave<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Audioslave',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'Audioslave',
 'offsetgroup': 'Audioslave',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([  9,  10, 270]),
 'xaxis': 'x',
 'y': array(['Audioslave', 'Out Of Exile', 'Revelations'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=BackBeat<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'BackBeat',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'BackBeat',
 'offsetgroup': 'BackBeat',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([11]),
 'xaxis': 'x',
 'y': array(['BackBeat Soundtrack'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Billy Cobham<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Billy Cobham',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'Billy Cobham',
 'offsetgroup': 'Billy Cobham',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
```

```
                        'x': array([12]),
                        'xaxis': 'x',
                        'y': array(['The Best Of Billy Cobham'], dtype=object),
                        'yaxis': 'y'},
                       {'alignmentgroup': 'True',
                        'hovertemplate': 'ArtistName=Black Label Society<br>index=%{x}<br>Title=%{y}<extra></extra
            >',
                        'legendgroup': 'Black Label Society',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': 'Black Label Society',
                        'offsetgroup': 'Black Label Society',
                        'orientation': 'h',
                        'showlegend': True,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array([13, 14]),
                        'xaxis': 'x',
                        'y': array(['Alcohol Fueled Brewtality Live! [Disc 1]',
                                    'Alcohol Fueled Brewtality Live! [Disc 2]'], dtype=object),
                        'yaxis': 'y'},
                       {'alignmentgroup': 'True',
                        'hovertemplate': 'ArtistName=Black Sabbath<br>index=%{x}<br>Title=%{y}<extra></extra>',
                        'legendgroup': 'Black Sabbath',
                        'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
                        'name': 'Black Sabbath',
                        'offsetgroup': 'Black Sabbath',
                        'orientation': 'h',
                        'showlegend': True,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array([15, 16]),
                        'xaxis': 'x',
                        'y': array(['Black Sabbath', 'Black Sabbath Vol. 4 (Remaster)'], dtype=object),
                        'yaxis': 'y'},
                       {'alignmentgroup': 'True',
                        'hovertemplate': 'ArtistName=Body Count<br>index=%{x}<br>Title=%{y}<extra></extra>',
                        'legendgroup': 'Body Count',
                        'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
                        'name': 'Body Count',
                        'offsetgroup': 'Body Count',
                        'orientation': 'h',
                        'showlegend': True,
```

```
 'textposition': 'auto',
 'type': 'bar',
 'x': array([17]),
 'xaxis': 'x',
 'y': array(['Body Count'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Bruce Dickinson<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Bruce Dickinson',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Bruce Dickinson',
 'offsetgroup': 'Bruce Dickinson',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([18]),
 'xaxis': 'x',
 'y': array(['Chemical Wedding'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Buddy Guy<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Buddy Guy',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'Buddy Guy',
 'offsetgroup': 'Buddy Guy',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([19]),
 'xaxis': 'x',
 'y': array(['The Best Of Buddy Guy - The Millenium Collection'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Caetano Veloso<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Caetano Veloso',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'Caetano Veloso',
 'offsetgroup': 'Caetano Veloso',
 'orientation': 'h',
 'showlegend': True,
```

```
 'textposition': 'auto',
 'type': 'bar',
 'x': array([20, 21]),
 'xaxis': 'x',
 'y': array(['Prenda Minha', 'Sozinho Remix Ao Vivo'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Chico Buarque<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Chico Buarque',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'Chico Buarque',
 'offsetgroup': 'Chico Buarque',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([22]),
 'xaxis': 'x',
 'y': array(['Minha Historia'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Chico Science & Naç' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Chico Science & Nação Zumbi',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'Chico Science & Nação Zumbi',
 'offsetgroup': 'Chico Science & Nação Zumbi',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([23, 24]),
 'xaxis': 'x',
 'y': array(['Afrociberdelia', 'Da Lama Ao Caos'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Cidade Negra<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Cidade Negra',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'Cidade Negra',
 'offsetgroup': 'Cidade Negra',
 'orientation': 'h',
 'showlegend': True,
```

```
 'textposition': 'auto',
 'type': 'bar',
 'x': array([25, 26]),
 'xaxis': 'x',
 'y': array(['Acústico MTV [Live]', 'Cidade Negra - Hits'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Cláudio Zoli<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Cláudio Zoli',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'Cláudio Zoli',
 'offsetgroup': 'Cláudio Zoli',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([27]),
 'xaxis': 'x',
 'y': array(['Na Pista'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Various Artists<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Various Artists',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': 'Various Artists',
 'offsetgroup': 'Various Artists',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([28, 31, 44, 52]),
 'xaxis': 'x',
 'y': array(['Axé Bahia 2001', 'Carnaval 2001', 'Sambas De Enredo 2001',
             'Vozes do MPB'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Led Zeppelin<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Led Zeppelin',
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': 'Led Zeppelin',
 'offsetgroup': 'Led Zeppelin',
 'orientation': 'h',
```

```
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([ 29,  43, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137]),
 'xaxis': 'x',
 'y': array(['BBC Sessions [Disc 1] [Live]', 'Physical Graffiti [Disc 1]',
             'BBC Sessions [Disc 2] [Live]', 'Coda', 'Houses Of The Holy',
             'In Through The Out Door', 'IV', 'Led Zeppelin I', 'Led Zeppelin II',
             'Led Zeppelin III', 'Physical Graffiti [Disc 2]', 'Presence',
             'The Song Remains The Same (Disc 1)',
             'The Song Remains The Same (Disc 2)'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Frank Zappa & Capta' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Frank Zappa & Captain Beefheart',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'Frank Zappa & Captain Beefheart',
 'offsetgroup': 'Frank Zappa & Captain Beefheart',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([30]),
 'xaxis': 'x',
 'y': array(['Bongo Fury'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Marcos Valle<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Marcos Valle',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Marcos Valle',
 'offsetgroup': 'Marcos Valle',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([32]),
 'xaxis': 'x',
 'y': array(['Chill: Brazil (Disc 1)'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Metallica<br>index=%{x}<br>Title=%{y}<extra></extra>',
```

```
 'legendgroup': 'Metallica',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'Metallica',
 'offsetgroup': 'Metallica',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([ 34, 147, 148, 149, 150, 151, 152, 153, 154, 155]),
 'xaxis': 'x',
 'y': array(['Garage Inc. (Disc 1)', 'Black Album', 'Garage Inc. (Disc 2)',
             "Kill 'Em All", 'Load', 'Master Of Puppets', 'ReLoad',
             'Ride The Lightning', 'St. Anger', '...And Justice For All'],
            dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Queen<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Queen',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'Queen',
 'offsetgroup': 'Queen',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([ 35, 184, 185]),
 'xaxis': 'x',
 'y': array(['Greatest Hits II', 'Greatest Hits I', 'News Of The World'],
            dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Kiss<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Kiss',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'Kiss',
 'offsetgroup': 'Kiss',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([ 36, 125]),
 'xaxis': 'x',
```

```
 'y': array(['Greatest Kiss', 'Unplugged [Live]'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Spyro Gyra<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Spyro Gyra',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'Spyro Gyra',
 'offsetgroup': 'Spyro Gyra',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([ 37, 203]),
 'xaxis': 'x',
 'y': array(['Heart of the Night', 'Morning Dance'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Green Day<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Green Day',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'Green Day',
 'offsetgroup': 'Green Day',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([38, 88]),
 'xaxis': 'x',
 'y': array(['International Superhits', 'American Idiot'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=David Coverdale<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'David Coverdale',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'David Coverdale',
 'offsetgroup': 'David Coverdale',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([39]),
 'xaxis': 'x',
```

```
 'y': array(['Into The Light'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Gonzaguinha<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Gonzaguinha',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': 'Gonzaguinha',
 'offsetgroup': 'Gonzaguinha',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([40]),
 'xaxis': 'x',
 'y': array(['Meus Momentos'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Os Mutantes<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Os Mutantes',
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': 'Os Mutantes',
 'offsetgroup': 'Os Mutantes',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([41]),
 'xaxis': 'x',
 'y': array(['Minha História'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Deep Purple<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Deep Purple',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'Deep Purple',
 'offsetgroup': 'Deep Purple',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([42, 49, 57, 58, 59, 60, 61, 62, 63, 64, 65]),
 'xaxis': 'x',
```

```
 'y': array(['MK III The Final Concerts [Disc 1]', 'The Final Concerts (Disc 2)',
             'Come Taste The Band', 'Deep Purple In Rock', 'Fireball',
             "Knocking at Your Back Door: The Best Of Deep Purple in the 80's",
             'Machine Head', 'Purpendicular', 'Slaves And Masters', 'Stormbringer',
             'The Battle Rages On'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Santana<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Santana',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Santana',
 'offsetgroup': 'Santana',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([ 45, 196, 197]),
 'xaxis': 'x',
 'y': array(['Supernatural', 'Santana - As Years Go By', 'Santana Live'],
           dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Ed Motta<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Ed Motta',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'Ed Motta',
 'offsetgroup': 'Ed Motta',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([46]),
 'xaxis': 'x',
 'y': array(['The Best of Ed Motta'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Miles Davis<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Miles Davis',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'Miles Davis',
 'offsetgroup': 'Miles Davis',
 'orientation': 'h',
```

```
                        'showlegend': True,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array([ 47,  48, 156]),
                        'xaxis': 'x',
                        'y': array(['The Essential Miles Davis [Disc 1]',
                                    'The Essential Miles Davis [Disc 2]', 'Miles Ahead'], dtype=object),
                        'yaxis': 'y'},
                       {'alignmentgroup': 'True',
                        'hovertemplate': 'ArtistName=Gene Krupa<br>index=%{x}<br>Title=%{y}<extra></extra>',
                        'legendgroup': 'Gene Krupa',
                        'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
                        'name': 'Gene Krupa',
                        'offsetgroup': 'Gene Krupa',
                        'orientation': 'h',
                        'showlegend': True,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array([50]),
                        'xaxis': 'x',
                        'y': array(["Up An' Atom"], dtype=object),
                        'yaxis': 'y'},
                       {'alignmentgroup': 'True',
                        'hovertemplate': 'ArtistName=Toquinho & Vinícius<br>index=%{x}<br>Title=%{y}<extra></extra
    >',
                        'legendgroup': 'Toquinho & Vinícius',
                        'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
                        'name': 'Toquinho & Vinícius',
                        'offsetgroup': 'Toquinho & Vinícius',
                        'orientation': 'h',
                        'showlegend': True,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array([51]),
                        'xaxis': 'x',
                        'y': array(['Vinícius De Moraes - Sem Limite'], dtype=object),
                        'yaxis': 'y'},
                       {'alignmentgroup': 'True',
                        'hovertemplate': ('ArtistName=Creedence Clearwate' ... '}<br>Title=%{y}<extra></extra>'),
                        'legendgroup': 'Creedence Clearwater Revival',
                        'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
                        'name': 'Creedence Clearwater Revival',
```

```
 'offsetgroup': 'Creedence Clearwater Revival',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([53, 54]),
 'xaxis': 'x',
 'y': array(['Chronicle, Vol. 1', 'Chronicle, Vol. 2'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Cássia Eller<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Cássia Eller',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'Cássia Eller',
 'offsetgroup': 'Cássia Eller',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([55, 56]),
 'xaxis': 'x',
 'y': array(['Cássia Eller - Coleção Sem Limite [Disc 2]',
             'Cássia Eller - Sem Limite [Disc 1]'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Def Leppard<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Def Leppard',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': 'Def Leppard',
 'offsetgroup': 'Def Leppard',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([66]),
 'xaxis': 'x',
 'y': array(["Vault: Def Leppard's Greatest Hits"], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Dennis Chambers<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Dennis Chambers',
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
```

```
 'name': 'Dennis Chambers',
 'offsetgroup': 'Dennis Chambers',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([67]),
 'xaxis': 'x',
 'y': array(['Outbreak'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Djavan<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Djavan',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'Djavan',
 'offsetgroup': 'Djavan',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([68, 69]),
 'xaxis': 'x',
 'y': array(['Djavan Ao Vivo - Vol. 02', 'Djavan Ao Vivo - Vol. 1'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Elis Regina<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Elis Regina',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Elis Regina',
 'offsetgroup': 'Elis Regina',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([70]),
 'xaxis': 'x',
 'y': array(['Elis Regina-Minha História'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Eric Clapton<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Eric Clapton',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
```

```
 'name': 'Eric Clapton',
 'offsetgroup': 'Eric Clapton',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([71, 72]),
 'xaxis': 'x',
 'y': array(['The Cream Of Clapton', 'Unplugged'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Faith No More<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Faith No More',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'Faith No More',
 'offsetgroup': 'Faith No More',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([73, 74, 75, 76]),
 'xaxis': 'x',
 'y': array(['Album Of The Year', 'Angel Dust', 'King For A Day Fool For A Lifetime',
             'The Real Thing'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Falamansa<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Falamansa',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'Falamansa',
 'offsetgroup': 'Falamansa',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([77]),
 'xaxis': 'x',
 'y': array(['Deixa Entrar'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Foo Fighters<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Foo Fighters',
```

```
                    'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
                    'name': 'Foo Fighters',
                    'offsetgroup': 'Foo Fighters',
                    'orientation': 'h',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array([78, 79, 80, 81]),
                    'xaxis': 'x',
                    'y': array(['In Your Honor [Disc 1]', 'In Your Honor [Disc 2]', 'One By One',
                                'The Colour And The Shape'], dtype=object),
                    'yaxis': 'y'},
                   {'alignmentgroup': 'True',
                    'hovertemplate': 'ArtistName=Frank Sinatra<br>index=%{x}<br>Title=%{y}<extra></extra>',
                    'legendgroup': 'Frank Sinatra',
                    'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
                    'name': 'Frank Sinatra',
                    'offsetgroup': 'Frank Sinatra',
                    'orientation': 'h',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array([82]),
                    'xaxis': 'x',
                    'y': array(['My Way: The Best Of Frank Sinatra [Disc 1]'], dtype=object),
                    'yaxis': 'y'},
                   {'alignmentgroup': 'True',
                    'hovertemplate': 'ArtistName=Funk Como Le Gusta<br>index=%{x}<br>Title=%{y}<extra></extra
>',
                    'legendgroup': 'Funk Como Le Gusta',
                    'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
                    'name': 'Funk Como Le Gusta',
                    'offsetgroup': 'Funk Como Le Gusta',
                    'orientation': 'h',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array([83]),
                    'xaxis': 'x',
                    'y': array(['Roda De Funk'], dtype=object),
                    'yaxis': 'y'},
                   {'alignmentgroup': 'True',
```

```
 'hovertemplate': 'ArtistName=Gilberto Gil<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Gilberto Gil',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': 'Gilberto Gil',
 'offsetgroup': 'Gilberto Gil',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([84, 85, 86]),
 'xaxis': 'x',
 'y': array(['As Canções de Eu Tu Eles', 'Quanta Gente Veio Ver (Live)',
             'Quanta Gente Veio ver--Bônus De Carnaval'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Godsmack<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Godsmack',
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': 'Godsmack',
 'offsetgroup': 'Godsmack',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([87]),
 'xaxis': 'x',
 'y': array(['Faceless'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': "ArtistName=Guns N' Roses<br>index=%{x}<br>Title=%{y}<extra></extra>",
 'legendgroup': "Guns N' Roses",
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': "Guns N' Roses",
 'offsetgroup': "Guns N' Roses",
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([89, 90, 91]),
 'xaxis': 'x',
 'y': array(['Appetite for Destruction', 'Use Your Illusion I',
             'Use Your Illusion II'], dtype=object),
```

```
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Incognito<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Incognito',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Incognito',
 'offsetgroup': 'Incognito',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([92]),
 'xaxis': 'x',
 'y': array(['Blue Moods'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Iron Maiden<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Iron Maiden',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'Iron Maiden',
 'offsetgroup': 'Iron Maiden',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([ 93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 103, 104, 105, 106,
            107, 108, 109, 110, 111, 112, 113]),
 'xaxis': 'x',
 'y': array(['A Matter of Life and Death', 'A Real Dead One', 'A Real Live One',
            'Brave New World', 'Dance Of Death', 'Fear Of The Dark', 'Iron Maiden',
            'Killers', 'Live After Death', 'Live At Donington 1992 (Disc 1)',
            'Live At Donington 1992 (Disc 2)', 'No Prayer For The Dying',
            'Piece Of Mind', 'Powerslave', 'Rock In Rio [CD1]', 'Rock In Rio [CD2]',
            'Seventh Son of a Seventh Son', 'Somewhere in Time',
            'The Number of The Beast', 'The X Factor', 'Virtual XI'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=James Brown<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'James Brown',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'James Brown',
 'offsetgroup': 'James Brown',
```

```
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([114]),
 'xaxis': 'x',
 'y': array(['Sex Machine'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Jamiroquai<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Jamiroquai',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'Jamiroquai',
 'offsetgroup': 'Jamiroquai',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([115, 116, 117]),
 'xaxis': 'x',
 'y': array(['Emergency On Planet Earth', 'Synkronized',
             'The Return Of The Space Cowboy'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=JET<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'JET',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'JET',
 'offsetgroup': 'JET',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([118]),
 'xaxis': 'x',
 'y': array(['Get Born'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Jimi Hendrix<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Jimi Hendrix',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'Jimi Hendrix',
```

```
 'offsetgroup': 'Jimi Hendrix',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([119]),
 'xaxis': 'x',
 'y': array(['Are You Experienced?'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Joe Satriani<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Joe Satriani',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'Joe Satriani',
 'offsetgroup': 'Joe Satriani',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([120]),
 'xaxis': 'x',
 'y': array(['Surfing with the Alien (Remastered)'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Jorge Ben<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Jorge Ben',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': 'Jorge Ben',
 'offsetgroup': 'Jorge Ben',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([121]),
 'xaxis': 'x',
 'y': array(['Jorge Ben Jor 25 Anos'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Jota Quest<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Jota Quest',
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': 'Jota Quest',
```

```
 'offsetgroup': 'Jota Quest',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([122]),
 'xaxis': 'x',
 'y': array(['Jota Quest-1995'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=João Suplicy<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'João Suplicy',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'João Suplicy',
 'offsetgroup': 'João Suplicy',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([123]),
 'xaxis': 'x',
 'y': array(['Cafezinho'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Judas Priest<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Judas Priest',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Judas Priest',
 'offsetgroup': 'Judas Priest',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([124]),
 'xaxis': 'x',
 'y': array(['Living After Midnight'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Legião Urbana<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Legião Urbana',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'Legião Urbana',
```

```
 'offsetgroup': 'Legião Urbana',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([138, 139]),
 'xaxis': 'x',
 'y': array(['A TempestadeTempestade Ou O Livro Dos Dias', 'Mais Do Mesmo'],
           dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Lenny Kravitz<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Lenny Kravitz',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'Lenny Kravitz',
 'offsetgroup': 'Lenny Kravitz',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([140]),
 'xaxis': 'x',
 'y': array(['Greatest Hits'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Lulu Santos<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Lulu Santos',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'Lulu Santos',
 'offsetgroup': 'Lulu Santos',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([141, 142]),
 'xaxis': 'x',
 'y': array(['Lulu Santos - RCA 100 Anos De Música - Álbum 01',
            'Lulu Santos - RCA 100 Anos De Música - Álbum 02'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Marillion<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Marillion',
```

```
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'Marillion',
 'offsetgroup': 'Marillion',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([143]),
 'xaxis': 'x',
 'y': array(['Misplaced Childhood'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Marisa Monte<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Marisa Monte',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'Marisa Monte',
 'offsetgroup': 'Marisa Monte',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([144]),
 'xaxis': 'x',
 'y': array(['Barulhinho Bom'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Marvin Gaye<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Marvin Gaye',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'Marvin Gaye',
 'offsetgroup': 'Marvin Gaye',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([145]),
 'xaxis': 'x',
 'y': array(['Seek And Shall Find: More Of The Best (1963-1981)'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Men At Work<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Men At Work',
```

```
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': 'Men At Work',
 'offsetgroup': 'Men At Work',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([146]),
 'xaxis': 'x',
 'y': array(['The Best Of Men At Work'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Milton Nascimento<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Milton Nascimento',
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': 'Milton Nascimento',
 'offsetgroup': 'Milton Nascimento',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([157, 158]),
 'xaxis': 'x',
 'y': array(['Milton Nascimento Ao Vivo', 'Minas'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Motörhead<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Motörhead',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'Motörhead',
 'offsetgroup': 'Motörhead',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([159]),
 'xaxis': 'x',
 'y': array(['Ace Of Spades'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Mônica Marianno<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Mônica Marianno',
```

```
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Mônica Marianno',
 'offsetgroup': 'Mônica Marianno',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([160]),
 'xaxis': 'x',
 'y': array(['Demorou...'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Mötley Crüe<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Mötley Crüe',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'Mötley Crüe',
 'offsetgroup': 'Mötley Crüe',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([161]),
 'xaxis': 'x',
 'y': array(['Motley Crue Greatest Hits'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Nirvana<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Nirvana',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'Nirvana',
 'offsetgroup': 'Nirvana',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([162, 163]),
 'xaxis': 'x',
 'y': array(['From The Muddy Banks Of The Wishkah [Live]', 'Nevermind'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=O Terço<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'O Terço',
```

```
                      'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
                      'name': 'O Terço',
                      'offsetgroup': 'O Terço',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([164]),
                      'xaxis': 'x',
                      'y': array(['Compositores'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=Olodum<br>index=%{x}<br>Title=%{y}<extra></extra>',
                      'legendgroup': 'Olodum',
                      'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
                      'name': 'Olodum',
                      'offsetgroup': 'Olodum',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([165]),
                      'xaxis': 'x',
                      'y': array(['Olodum'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=Os Paralamas Do Sucesso<br>index=%{x}<br>Title=%{y}<extra></ex
tra>',
                      'legendgroup': 'Os Paralamas Do Sucesso',
                      'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
                      'name': 'Os Paralamas Do Sucesso',
                      'offsetgroup': 'Os Paralamas Do Sucesso',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([166, 167, 168]),
                      'xaxis': 'x',
                      'y': array(['Acústico MTV', 'Arquivo II', 'Arquivo Os Paralamas Do Sucesso'],
                                 dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
```

```
 'hovertemplate': 'ArtistName=Ozzy Osbourne<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Ozzy Osbourne',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'Ozzy Osbourne',
 'offsetgroup': 'Ozzy Osbourne',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([169, 170, 171, 172, 173, 255]),
 'xaxis': 'x',
 'y': array(['Bark at the Moon (Remastered)', 'Blizzard of Ozz',
             'Diary of a Madman (Remastered)', 'No More Tears (Remastered)',
             'Tribute', 'Speak of the Devil'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Page & Plant<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Page & Plant',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': 'Page & Plant',
 'offsetgroup': 'Page & Plant',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([174]),
 'xaxis': 'x',
 'y': array(['Walking Into Clarksdale'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Passengers<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Passengers',
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': 'Passengers',
 'offsetgroup': 'Passengers',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([175]),
 'xaxis': 'x',
 'y': array(['Original Soundtracks 1'], dtype=object),
```

```
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': "ArtistName=Paul D'Ianno<br>index=%{x}<br>Title=%{y}<extra></extra>",
 'legendgroup': "Paul D'Ianno",
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': "Paul D'Ianno",
 'offsetgroup': "Paul D'Ianno",
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([176]),
 'xaxis': 'x',
 'y': array(['The Beast Live'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Pearl Jam<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Pearl Jam',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Pearl Jam',
 'offsetgroup': 'Pearl Jam',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([177, 178, 179, 180, 181]),
 'xaxis': 'x',
 'y': array(['Live On Two Legs [Live]', 'Pearl Jam', 'Riot Act', 'Ten', 'Vs.'],
           dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Pink Floyd<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Pink Floyd',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'Pink Floyd',
 'offsetgroup': 'Pink Floyd',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([182]),
 'xaxis': 'x',
```

```
                      'y': array(['Dark Side Of The Moon'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=Planet Hemp<br>index=%{x}<br>Title=%{y}<extra></extra>',
                      'legendgroup': 'Planet Hemp',
                      'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
                      'name': 'Planet Hemp',
                      'offsetgroup': 'Planet Hemp',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([183]),
                      'xaxis': 'x',
                      'y': array(['Os Cães Ladram Mas A Caravana Não Pára'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=R.E.M. Feat. Kate Pearson<br>index=%{x}<br>Title=%{y}<extra></
extra>',
                      'legendgroup': 'R.E.M. Feat. Kate Pearson',
                      'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
                      'name': 'R.E.M. Feat. Kate Pearson',
                      'offsetgroup': 'R.E.M. Feat. Kate Pearson',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([186]),
                      'xaxis': 'x',
                      'y': array(['Out Of Time'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=R.E.M.<br>index=%{x}<br>Title=%{y}<extra></extra>',
                      'legendgroup': 'R.E.M.',
                      'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
                      'name': 'R.E.M.',
                      'offsetgroup': 'R.E.M.',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([187, 188, 189]),
```

```
                      'xaxis': 'x',
                      'y': array(['Green', 'New Adventures In Hi-Fi', 'The Best Of R.E.M.: The IRS Years'],
                                 dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=Raimundos<br>index=%{x}<br>Title=%{y}<extra></extra>',
                      'legendgroup': 'Raimundos',
                      'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
                      'name': 'Raimundos',
                      'offsetgroup': 'Raimundos',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([190]),
                      'xaxis': 'x',
                      'y': array(['Cesta Básica'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=Raul Seixas<br>index=%{x}<br>Title=%{y}<extra></extra>',
                      'legendgroup': 'Raul Seixas',
                      'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
                      'name': 'Raul Seixas',
                      'offsetgroup': 'Raul Seixas',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([191]),
                      'xaxis': 'x',
                      'y': array(['Raul Seixas'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=Red Hot Chili Peppers<br>index=%{x}<br>Title=%{y}<extra></extr
a>',
                      'legendgroup': 'Red Hot Chili Peppers',
                      'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                      'name': 'Red Hot Chili Peppers',
                      'offsetgroup': 'Red Hot Chili Peppers',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
```

```
 'type': 'bar',
 'x': array([192, 193, 194]),
 'xaxis': 'x',
 'y': array(['Blood Sugar Sex Magik', 'By The Way', 'Californication'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Rush<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Rush',
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': 'Rush',
 'offsetgroup': 'Rush',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([195]),
 'xaxis': 'x',
 'y': array(['Retrospective I (1974-1980)'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Skank<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Skank',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'Skank',
 'offsetgroup': 'Skank',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([198, 199]),
 'xaxis': 'x',
 'y': array(['Maquinarama', 'O Samba Poconé'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Smashing Pumpkins<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Smashing Pumpkins',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Smashing Pumpkins',
 'offsetgroup': 'Smashing Pumpkins',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
```

```
                     'type': 'bar',
                     'x': array([200, 201]),
                     'xaxis': 'x',
                     'y': array(['Judas 0: B-Sides and Rarities', 'Rotten Apples: Greatest Hits'],
                                dtype=object),
                     'yaxis': 'y'},
                    {'alignmentgroup': 'True',
                     'hovertemplate': 'ArtistName=Soundgarden<br>index=%{x}<br>Title=%{y}<extra></extra>',
                     'legendgroup': 'Soundgarden',
                     'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
                     'name': 'Soundgarden',
                     'offsetgroup': 'Soundgarden',
                     'orientation': 'h',
                     'showlegend': True,
                     'textposition': 'auto',
                     'type': 'bar',
                     'x': array([202]),
                     'xaxis': 'x',
                     'y': array(['A-Sides'], dtype=object),
                     'yaxis': 'y'},
                    {'alignmentgroup': 'True',
                     'hovertemplate': ('ArtistName=Stevie Ray Vaughan ' ... '}<br>Title=%{y}<extra></extra>'),
                     'legendgroup': 'Stevie Ray Vaughan & Double Trouble',
                     'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
                     'name': 'Stevie Ray Vaughan & Double Trouble',
                     'offsetgroup': 'Stevie Ray Vaughan & Double Trouble',
                     'orientation': 'h',
                     'showlegend': True,
                     'textposition': 'auto',
                     'type': 'bar',
                     'x': array([204]),
                     'xaxis': 'x',
                     'y': array(['In Step'], dtype=object),
                     'yaxis': 'y'},
                    {'alignmentgroup': 'True',
                     'hovertemplate': 'ArtistName=Stone Temple Pilots<br>index=%{x}<br>Title=%{y}<extra></extra
            >',
                     'legendgroup': 'Stone Temple Pilots',
                     'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
                     'name': 'Stone Temple Pilots',
                     'offsetgroup': 'Stone Temple Pilots',
                     'orientation': 'h',
```

```
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([205]),
 'xaxis': 'x',
 'y': array(['Core'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=System Of A Down<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'System Of A Down',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'System Of A Down',
 'offsetgroup': 'System Of A Down',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([206]),
 'xaxis': 'x',
 'y': array(['Mezmerize'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Terry Bozzio, Tony ' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Terry Bozzio, Tony Levin & Steve Stevens',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'Terry Bozzio, Tony Levin & Steve Stevens',
 'offsetgroup': 'Terry Bozzio, Tony Levin & Steve Stevens',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([207]),
 'xaxis': 'x',
 'y': array(['[1997] Black Light Syndrome'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=The Black Crowes<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'The Black Crowes',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'The Black Crowes',
 'offsetgroup': 'The Black Crowes',
 'orientation': 'h',
```

```
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([208, 209]),
                      'xaxis': 'x',
                      'y': array(['Live [Disc 1]', 'Live [Disc 2]'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=The Clash<br>index=%{x}<br>Title=%{y}<extra></extra>',
                      'legendgroup': 'The Clash',
                      'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                      'name': 'The Clash',
                      'offsetgroup': 'The Clash',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([210]),
                      'xaxis': 'x',
                      'y': array(['The Singles'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=The Cult<br>index=%{x}<br>Title=%{y}<extra></extra>',
                      'legendgroup': 'The Cult',
                      'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
                      'name': 'The Cult',
                      'offsetgroup': 'The Cult',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([211, 212]),
                      'xaxis': 'x',
                      'y': array(['Beyond Good And Evil',
                                  'Pure Cult: The Best Of The Cult (For Rockers, Ravers, Lovers & Sinners) [U
K]'],
                                 dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=The Doors<br>index=%{x}<br>Title=%{y}<extra></extra>',
                      'legendgroup': 'The Doors',
                      'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
```

```
                      'name': 'The Doors',
                      'offsetgroup': 'The Doors',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([213]),
                      'xaxis': 'x',
                      'y': array(['The Doors'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=The Police<br>index=%{x}<br>Title=%{y}<extra></extra>',
                      'legendgroup': 'The Police',
                      'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
                      'name': 'The Police',
                      'offsetgroup': 'The Police',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([214]),
                      'xaxis': 'x',
                      'y': array(['The Police Greatest Hits'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=The Rolling Stones<br>index=%{x}<br>Title=%{y}<extra></extra
           >',
                      'legendgroup': 'The Rolling Stones',
                      'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
                      'name': 'The Rolling Stones',
                      'offsetgroup': 'The Rolling Stones',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([215, 216, 217]),
                      'xaxis': 'x',
                      'y': array(['Hot Rocks, 1964-1971 (Disc 1)', 'No Security', 'Voodoo Lounge'],
                                 dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=The Tea Party<br>index=%{x}<br>Title=%{y}<extra></extra>',
```

```
 'legendgroup': 'The Tea Party',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'The Tea Party',
 'offsetgroup': 'The Tea Party',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([218, 219]),
 'xaxis': 'x',
 'y': array(['Tangents', 'Transmission'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=The Who<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'The Who',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'The Who',
 'offsetgroup': 'The Who',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([220]),
 'xaxis': 'x',
 'y': array(['My Generation - The Very Best Of The Who'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Tim Maia<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Tim Maia',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'Tim Maia',
 'offsetgroup': 'Tim Maia',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([221, 222]),
 'xaxis': 'x',
 'y': array(['Serie Sem Limite (Disc 1)', 'Serie Sem Limite (Disc 2)'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Titãs<br>index=%{x}<br>Title=%{y}<extra></extra>',
```

```
                      'legendgroup': 'Titãs',
                      'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
                      'name': 'Titãs',
                      'offsetgroup': 'Titãs',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([223, 224]),
                      'xaxis': 'x',
                      'y': array(['Acústico', 'Volume Dois'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=Battlestar Galactica<br>index=%{x}<br>Title=%{y}<extra></extra
>',
                      'legendgroup': 'Battlestar Galactica',
                      'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
                      'name': 'Battlestar Galactica',
                      'offsetgroup': 'Battlestar Galactica',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([225, 226]),
                      'xaxis': 'x',
                      'y': array(['Battlestar Galactica: The Story So Far',
                                  'Battlestar Galactica, Season 3'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=Heroes<br>index=%{x}<br>Title=%{y}<extra></extra>',
                      'legendgroup': 'Heroes',
                      'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                      'name': 'Heroes',
                      'offsetgroup': 'Heroes',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([227]),
                      'xaxis': 'x',
                      'y': array(['Heroes, Season 1'], dtype=object),
                      'yaxis': 'y'},
```

```
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Lost<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Lost',
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': 'Lost',
 'offsetgroup': 'Lost',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([228, 229, 230, 260]),
 'xaxis': 'x',
 'y': array(['Lost, Season 3', 'Lost, Season 1', 'Lost, Season 2', 'LOST, Season 4'],
           dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=U2<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'U2',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'U2',
 'offsetgroup': 'U2',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([231, 232, 233, 234, 235, 236, 237, 238, 239, 254]),
 'xaxis': 'x',
 'y': array(['Achtung Baby', "All That You Can't Leave Behind", 'B-Sides 1980-1990',
             'How To Dismantle An Atomic Bomb', 'Pop', 'Rattle And Hum',
             'The Best Of 1980-1990', 'War', 'Zooropa',
             'Instant Karma: The Amnesty International Campaign to Save Darfur'],
            dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=UB40<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'UB40',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'UB40',
 'offsetgroup': 'UB40',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
```

```
                     'type': 'bar',
                     'x': array([240]),
                     'xaxis': 'x',
                     'y': array(['UB40 The Best Of - Volume Two [UK]'], dtype=object),
                     'yaxis': 'y'},
                    {'alignmentgroup': 'True',
                     'hovertemplate': 'ArtistName=Van Halen<br>index=%{x}<br>Title=%{y}<extra></extra>',
                     'legendgroup': 'Van Halen',
                     'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
                     'name': 'Van Halen',
                     'offsetgroup': 'Van Halen',
                     'orientation': 'h',
                     'showlegend': True,
                     'textposition': 'auto',
                     'type': 'bar',
                     'x': array([241, 242, 243, 244]),
                     'xaxis': 'x',
                     'y': array(['Diver Down', 'The Best Of Van Halen, Vol. I', 'Van Halen',
                                 'Van Halen III'], dtype=object),
                     'yaxis': 'y'},
                    {'alignmentgroup': 'True',
                     'hovertemplate': 'ArtistName=Velvet Revolver<br>index=%{x}<br>Title=%{y}<extra></extra>',
                     'legendgroup': 'Velvet Revolver',
                     'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
                     'name': 'Velvet Revolver',
                     'offsetgroup': 'Velvet Revolver',
                     'orientation': 'h',
                     'showlegend': True,
                     'textposition': 'auto',
                     'type': 'bar',
                     'x': array([245]),
                     'xaxis': 'x',
                     'y': array(['Contraband'], dtype=object),
                     'yaxis': 'y'},
                    {'alignmentgroup': 'True',
                     'hovertemplate': 'ArtistName=Vinícius De Moraes<br>index=%{x}<br>Title=%{y}<extra></extra
           >',
                     'legendgroup': 'Vinícius De Moraes',
                     'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
                     'name': 'Vinícius De Moraes',
                     'offsetgroup': 'Vinícius De Moraes',
                     'orientation': 'h',
```

```
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([246]),
 'xaxis': 'x',
 'y': array(['Vinicius De Moraes'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Zeca Pagodinho<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Zeca Pagodinho',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'Zeca Pagodinho',
 'offsetgroup': 'Zeca Pagodinho',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([247]),
 'xaxis': 'x',
 'y': array(['Ao Vivo [IMPORT]'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=The Office<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'The Office',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'The Office',
 'offsetgroup': 'The Office',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([248, 249, 250]),
 'xaxis': 'x',
 'y': array(['The Office, Season 1', 'The Office, Season 2', 'The Office, Season 3'],
           dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Dread Zeppelin<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Dread Zeppelin',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'Dread Zeppelin',
 'offsetgroup': 'Dread Zeppelin',
```

```
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([251]),
 'xaxis': 'x',
 'y': array(['Un-Led-Ed'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Battlestar Galactic' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Battlestar Galactica (Classic)',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': 'Battlestar Galactica (Classic)',
 'offsetgroup': 'Battlestar Galactica (Classic)',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([252]),
 'xaxis': 'x',
 'y': array(['Battlestar Galactica (Classic), Season 1'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Aquaman<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Aquaman',
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': 'Aquaman',
 'offsetgroup': 'Aquaman',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([253]),
 'xaxis': 'x',
 'y': array(['Aquaman'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Scorpions<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Scorpions',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'Scorpions',
 'offsetgroup': 'Scorpions',
```

```
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([256]),
 'xaxis': 'x',
 'y': array(['20th Century Masters - The Millennium Collection: The Best of Scorpions'],
           dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=House Of Pain<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'House Of Pain',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'House Of Pain',
 'offsetgroup': 'House Of Pain',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([257]),
 'xaxis': 'x',
 'y': array(['House of Pain'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=O Rappa<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'O Rappa',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'O Rappa',
 'offsetgroup': 'O Rappa',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([258]),
 'xaxis': 'x',
 'y': array(['Radio Brasil (O Som da Jovem Vanguarda) - Seleccao de Henrique Amaro'],
           dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Cake<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Cake',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
```

```
                        'name': 'Cake',
                        'offsetgroup': 'Cake',
                        'orientation': 'h',
                        'showlegend': True,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array([259]),
                        'xaxis': 'x',
                        'y': array(['Cake: B-Sides and Rarities'], dtype=object),
                        'yaxis': 'y'},
                       {'alignmentgroup': 'True',
                        'hovertemplate': 'ArtistName=Aisha Duo<br>index=%{x}<br>Title=%{y}<extra></extra>',
                        'legendgroup': 'Aisha Duo',
                        'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
                        'name': 'Aisha Duo',
                        'offsetgroup': 'Aisha Duo',
                        'orientation': 'h',
                        'showlegend': True,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array([261]),
                        'xaxis': 'x',
                        'y': array(['Quiet Songs'], dtype=object),
                        'yaxis': 'y'},
                       {'alignmentgroup': 'True',
                        'hovertemplate': 'ArtistName=Habib Koité and Bamada<br>index=%{x}<br>Title=%{y}<extra></ext
ra>',
                        'legendgroup': 'Habib Koité and Bamada',
                        'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
                        'name': 'Habib Koité and Bamada',
                        'offsetgroup': 'Habib Koité and Bamada',
                        'orientation': 'h',
                        'showlegend': True,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array([262]),
                        'xaxis': 'x',
                        'y': array(['Muso Ko'], dtype=object),
                        'yaxis': 'y'},
                       {'alignmentgroup': 'True',
                        'hovertemplate': 'ArtistName=Karsh Kale<br>index=%{x}<br>Title=%{y}<extra></extra>',
                        'legendgroup': 'Karsh Kale',
```

```
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'Karsh Kale',
 'offsetgroup': 'Karsh Kale',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([263]),
 'xaxis': 'x',
 'y': array(['Realize'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=The Posies<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'The Posies',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'The Posies',
 'offsetgroup': 'The Posies',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([264]),
 'xaxis': 'x',
 'y': array(['Every Kind of Light'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Luciana Souza/Romer' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Luciana Souza/Romero Lubambo',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': 'Luciana Souza/Romero Lubambo',
 'offsetgroup': 'Luciana Souza/Romero Lubambo',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([265]),
 'xaxis': 'x',
 'y': array(['Duos II'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Aaron Goldberg<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Aaron Goldberg',
```

```
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': 'Aaron Goldberg',
 'offsetgroup': 'Aaron Goldberg',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([266]),
 'xaxis': 'x',
 'y': array(['Worlds'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Nicolaus Esterhazy ' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Nicolaus Esterhazy Sinfonia',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'Nicolaus Esterhazy Sinfonia',
 'offsetgroup': 'Nicolaus Esterhazy Sinfonia',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([267]),
 'xaxis': 'x',
 'y': array(['The Best of Beethoven'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Temple of the Dog<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Temple of the Dog',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Temple of the Dog',
 'offsetgroup': 'Temple of the Dog',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([268]),
 'xaxis': 'x',
 'y': array(['Temple of the Dog'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Chris Cornell<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Chris Cornell',
```

```
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'Chris Cornell',
 'offsetgroup': 'Chris Cornell',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([269]),
 'xaxis': 'x',
 'y': array(['Carry On'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Alberto Turco & Nov' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Alberto Turco & Nova Schola Gregoriana',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'Alberto Turco & Nova Schola Gregoriana',
 'offsetgroup': 'Alberto Turco & Nova Schola Gregoriana',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([271]),
 'xaxis': 'x',
 'y': array(['Adorate Deum: Gregorian Chant from the Proper of the Mass'],
           dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Richard Marlow & Th' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Richard Marlow & The Choir of Trinity College, Cambridge',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'Richard Marlow & The Choir of Trinity College, Cambridge',
 'offsetgroup': 'Richard Marlow & The Choir of Trinity College, Cambridge',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([272]),
 'xaxis': 'x',
 'y': array(['Allegri: Miserere'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=English Concert & T' ... '}<br>Title=%{y}<extra></extra>'),
```

```
                  'legendgroup': 'English Concert & Trevor Pinnock',
                  'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
                  'name': 'English Concert & Trevor Pinnock',
                  'offsetgroup': 'English Concert & Trevor Pinnock',
                  'orientation': 'h',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array([273, 314]),
                  'xaxis': 'x',
                  'y': array(['Pachelbel: Canon & Gigue',
                              'Handel: Music for the Royal Fireworks (Original Version 1749)'],
                             dtype=object),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': ('ArtistName=Anne-Sophie Mutter,' ... '}<br>Title=%{y}<extra></extra>'),
                  'legendgroup': 'Anne-Sophie Mutter, Herbert Von Karajan & Wiener Philharmoniker',
                  'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
                  'name': 'Anne-Sophie Mutter, Herbert Von Karajan & Wiener Philharmoniker',
                  'offsetgroup': 'Anne-Sophie Mutter, Herbert Von Karajan & Wiener Philharmoniker',
                  'orientation': 'h',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array([274]),
                  'xaxis': 'x',
                  'y': array(['Vivaldi: The Four Seasons'], dtype=object),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': ('ArtistName=Hilary Hahn, Jeffre' ... '}<br>Title=%{y}<extra></extra>'),
                  'legendgroup': 'Hilary Hahn, Jeffrey Kahane, Los Angeles Chamber Orchestra & Margaret Batje
r',
                  'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
                  'name': 'Hilary Hahn, Jeffrey Kahane, Los Angeles Chamber Orchestra & Margaret Batjer',
                  'offsetgroup': 'Hilary Hahn, Jeffrey Kahane, Los Angeles Chamber Orchestra & Margaret Batje
r',
                  'orientation': 'h',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array([275]),
                  'xaxis': 'x',
```

```
                      'y': array(['Bach: Violin Concertos'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=Wilhelm Kempff<br>index=%{x}<br>Title=%{y}<extra></extra>',
                      'legendgroup': 'Wilhelm Kempff',
                      'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                      'name': 'Wilhelm Kempff',
                      'offsetgroup': 'Wilhelm Kempff',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([276]),
                      'xaxis': 'x',
                      'y': array(['Bach: Goldberg Variations'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=Yo-Yo Ma<br>index=%{x}<br>Title=%{y}<extra></extra>',
                      'legendgroup': 'Yo-Yo Ma',
                      'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
                      'name': 'Yo-Yo Ma',
                      'offsetgroup': 'Yo-Yo Ma',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([277]),
                      'xaxis': 'x',
                      'y': array(['Bach: The Cello Suites'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=Scholars Baroque Ensemble<br>index=%{x}<br>Title=%{y}<extra></
extra>',
                      'legendgroup': 'Scholars Baroque Ensemble',
                      'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
                      'name': 'Scholars Baroque Ensemble',
                      'offsetgroup': 'Scholars Baroque Ensemble',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([278]),
```

```
                      'xaxis': 'x',
                      'y': array(['Handel: The Messiah (Highlights)'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': ('ArtistName=Academy of St. Mart' ... '}<br>Title=%{y}<extra></extra>'),
                      'legendgroup': 'Academy of St. Martin in the Fields & Sir Neville Marriner',
                      'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
                      'name': 'Academy of St. Martin in the Fields & Sir Neville Marriner',
                      'offsetgroup': 'Academy of St. Martin in the Fields & Sir Neville Marriner',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([279]),
                      'xaxis': 'x',
                      'y': array(['The World of Classical Favourites'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': ('ArtistName=Academy of St. Mart' ... '}<br>Title=%{y}<extra></extra>'),
                      'legendgroup': 'Academy of St. Martin in the Fields Chamber Ensemble & Sir Neville Marrine
r',
                      'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
                      'name': 'Academy of St. Martin in the Fields Chamber Ensemble & Sir Neville Marriner',
                      'offsetgroup': 'Academy of St. Martin in the Fields Chamber Ensemble & Sir Neville Marrine
r',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([280]),
                      'xaxis': 'x',
                      'y': array(['Sir Neville Marriner: A Celebration'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': ('ArtistName=Berliner Philharmon' ... '}<br>Title=%{y}<extra></extra>'),
                      'legendgroup': 'Berliner Philharmoniker, Claudio Abbado & Sabine Meyer',
                      'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
                      'name': 'Berliner Philharmoniker, Claudio Abbado & Sabine Meyer',
                      'offsetgroup': 'Berliner Philharmoniker, Claudio Abbado & Sabine Meyer',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
```

```
 'type': 'bar',
 'x': array([281]),
 'xaxis': 'x',
 'y': array(['Mozart: Wind Concertos'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Royal Philharmonic ' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Royal Philharmonic Orchestra & Sir Thomas Beecham',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'Royal Philharmonic Orchestra & Sir Thomas Beecham',
 'offsetgroup': 'Royal Philharmonic Orchestra & Sir Thomas Beecham',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([282]),
 'xaxis': 'x',
 'y': array(['Haydn: Symphonies 99 - 104'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Orchestre Révolutio' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Orchestre Révolutionnaire et Romantique & John Eliot Gardiner',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'Orchestre Révolutionnaire et Romantique & John Eliot Gardiner',
 'offsetgroup': 'Orchestre Révolutionnaire et Romantique & John Eliot Gardiner',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([283]),
 'xaxis': 'x',
 'y': array(['Beethoven: Symhonies Nos. 5 & 6'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Britten Sinfonia, I' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Britten Sinfonia, Ivor Bolton & Lesley Garrett',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'Britten Sinfonia, Ivor Bolton & Lesley Garrett',
 'offsetgroup': 'Britten Sinfonia, Ivor Bolton & Lesley Garrett',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
```

```
 'type': 'bar',
 'x': array([284]),
 'xaxis': 'x',
 'y': array(['A Soprano Inspired'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Chicago Symphony Ch' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Chicago Symphony Chorus, Chicago Symphony Orchestra & Sir Georg Solti',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'Chicago Symphony Chorus, Chicago Symphony Orchestra & Sir Georg Solti',
 'offsetgroup': 'Chicago Symphony Chorus, Chicago Symphony Orchestra & Sir Georg Solti',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([285]),
 'xaxis': 'x',
 'y': array(['Great Opera Choruses'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Sir Georg Solti & W' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Sir Georg Solti & Wiener Philharmoniker',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': 'Sir Georg Solti & Wiener Philharmoniker',
 'offsetgroup': 'Sir Georg Solti & Wiener Philharmoniker',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([286]),
 'xaxis': 'x',
 'y': array(['Wagner: Favourite Overtures'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Academy of St. Mart' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': ('Academy of St. Martin in the F' ... 'ville Marriner & Sylvia McNair'),
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': ('Academy of St. Martin in the F' ... 'ville Marriner & Sylvia McNair'),
 'offsetgroup': ('Academy of St. Martin in the F' ... 'ville Marriner & Sylvia McNair'),
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
```

```
 'type': 'bar',
 'x': array([287]),
 'xaxis': 'x',
 'y': array(['Fauré: Requiem, Ravel: Pavane & Others'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=London Symphony Orc' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'London Symphony Orchestra & Sir Charles Mackerras',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'London Symphony Orchestra & Sir Charles Mackerras',
 'offsetgroup': 'London Symphony Orchestra & Sir Charles Mackerras',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([288]),
 'xaxis': 'x',
 'y': array(['Tchaikovsky: The Nutcracker'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Barry Wordsworth & ' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Barry Wordsworth & BBC Concert Orchestra',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Barry Wordsworth & BBC Concert Orchestra',
 'offsetgroup': 'Barry Wordsworth & BBC Concert Orchestra',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([289]),
 'xaxis': 'x',
 'y': array(['The Last Night of the Proms'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Herbert Von Karajan' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Herbert Von Karajan, Mirella Freni & Wiener Philharmoniker',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'Herbert Von Karajan, Mirella Freni & Wiener Philharmoniker',
 'offsetgroup': 'Herbert Von Karajan, Mirella Freni & Wiener Philharmoniker',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
```

```
 'type': 'bar',
 'x': array([290]),
 'xaxis': 'x',
 'y': array(['Puccini: Madama Butterfly - Highlights'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Eugene Ormandy<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Eugene Ormandy',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'Eugene Ormandy',
 'offsetgroup': 'Eugene Ormandy',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([291, 310, 342]),
 'xaxis': 'x',
 'y': array(['Holst: The Planets, Op. 32 & Vaughan Williams: Fantasies',
             'Strauss: Waltzes', 'Respighi:Pines of Rome'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Luciano Pavarotti<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Luciano Pavarotti',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'Luciano Pavarotti',
 'offsetgroup': 'Luciano Pavarotti',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([292]),
 'xaxis': 'x',
 'y': array(["Pavarotti's Opera Made Easy"], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Leonard Bernstein &' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Leonard Bernstein & New York Philharmonic',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'Leonard Bernstein & New York Philharmonic',
 'offsetgroup': 'Leonard Bernstein & New York Philharmonic',
 'orientation': 'h',
 'showlegend': True,
```

```
                     'textposition': 'auto',
                     'type': 'bar',
                     'x': array([293]),
                     'xaxis': 'x',
                     'y': array(["Great Performances - Barber's Adagio and Other Romantic Favorites for String
s"],
                                dtype=object),
                     'yaxis': 'y'},
                    {'alignmentgroup': 'True',
                     'hovertemplate': ('ArtistName=Boston Symphony Orc' ... '}<br>Title=%{y}<extra></extra>'),
                     'legendgroup': 'Boston Symphony Orchestra & Seiji Ozawa',
                     'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
                     'name': 'Boston Symphony Orchestra & Seiji Ozawa',
                     'offsetgroup': 'Boston Symphony Orchestra & Seiji Ozawa',
                     'orientation': 'h',
                     'showlegend': True,
                     'textposition': 'auto',
                     'type': 'bar',
                     'x': array([294]),
                     'xaxis': 'x',
                     'y': array(['Carmina Burana'], dtype=object),
                     'yaxis': 'y'},
                    {'alignmentgroup': 'True',
                     'hovertemplate': ('ArtistName=Aaron Copland & Lon' ... '}<br>Title=%{y}<extra></extra>'),
                     'legendgroup': 'Aaron Copland & London Symphony Orchestra',
                     'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
                     'name': 'Aaron Copland & London Symphony Orchestra',
                     'offsetgroup': 'Aaron Copland & London Symphony Orchestra',
                     'orientation': 'h',
                     'showlegend': True,
                     'textposition': 'auto',
                     'type': 'bar',
                     'x': array([295]),
                     'xaxis': 'x',
                     'y': array(['A Copland Celebration, Vol. I'], dtype=object),
                     'yaxis': 'y'},
                    {'alignmentgroup': 'True',
                     'hovertemplate': 'ArtistName=Ton Koopman<br>index=%{x}<br>Title=%{y}<extra></extra>',
                     'legendgroup': 'Ton Koopman',
                     'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                     'name': 'Ton Koopman',
                     'offsetgroup': 'Ton Koopman',
```

```
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([296]),
 'xaxis': 'x',
 'y': array(['Bach: Toccata & Fugue in D Minor'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Sergei Prokofiev & ' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Sergei Prokofiev & Yuri Temirkanov',
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': 'Sergei Prokofiev & Yuri Temirkanov',
 'offsetgroup': 'Sergei Prokofiev & Yuri Temirkanov',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([297]),
 'xaxis': 'x',
 'y': array(['Prokofiev: Symphony No.1'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Chicago Symphony Or' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Chicago Symphony Orchestra & Fritz Reiner',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'Chicago Symphony Orchestra & Fritz Reiner',
 'offsetgroup': 'Chicago Symphony Orchestra & Fritz Reiner',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([298]),
 'xaxis': 'x',
 'y': array(['Scheherazade'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Orchestra of The Ag' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Orchestra of The Age of Enlightenment',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Orchestra of The Age of Enlightenment',
 'offsetgroup': 'Orchestra of The Age of Enlightenment',
```

```
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([299]),
 'xaxis': 'x',
 'y': array(['Bach: The Brandenburg Concertos'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Emanuel Ax, Eugene ' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Emanuel Ax, Eugene Ormandy & Philadelphia Orchestra',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'Emanuel Ax, Eugene Ormandy & Philadelphia Orchestra',
 'offsetgroup': 'Emanuel Ax, Eugene Ormandy & Philadelphia Orchestra',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([300]),
 'xaxis': 'x',
 'y': array(['Chopin: Piano Concertos Nos. 1 & 2'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=James Levine<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'James Levine',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'James Levine',
 'offsetgroup': 'James Levine',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([301]),
 'xaxis': 'x',
 'y': array(['Mascagni: Cavalleria Rusticana'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Berliner Philharmon' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Berliner Philharmoniker & Hans Rosbaud',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'Berliner Philharmoniker & Hans Rosbaud',
 'offsetgroup': 'Berliner Philharmoniker & Hans Rosbaud',
```

```
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([302]),
 'xaxis': 'x',
 'y': array(['Sibelius: Finlandia'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Maurizio Pollini<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Maurizio Pollini',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'Maurizio Pollini',
 'offsetgroup': 'Maurizio Pollini',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([303]),
 'xaxis': 'x',
 'y': array(['Beethoven Piano Sonatas: Moonlight & Pastorale'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Gustav Mahler<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Gustav Mahler',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'Gustav Mahler',
 'offsetgroup': 'Gustav Mahler',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([304]),
 'xaxis': 'x',
 'y': array(['Great Recordings of the Century - Mahler: Das Lied von der Erde'],
           dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Felix Schmidt, Lond' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Felix Schmidt, London Symphony Orchestra & Rafael Frühbeck de Burgos',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'Felix Schmidt, London Symphony Orchestra & Rafael Frühbeck de Burgos',
```

```
                     'offsetgroup': 'Felix Schmidt, London Symphony Orchestra & Rafael Frühbeck de Burgos',
                     'orientation': 'h',
                     'showlegend': True,
                     'textposition': 'auto',
                     'type': 'bar',
                     'x': array([305]),
                     'xaxis': 'x',
                     'y': array(['Elgar: Cello Concerto & Vaughan Williams: Fantasias'], dtype=object),
                     'yaxis': 'y'},
                    {'alignmentgroup': 'True',
                     'hovertemplate': ('ArtistName=Edo de Waart & San ' ... '}<br>Title=%{y}<extra></extra>'),
                     'legendgroup': 'Edo de Waart & San Francisco Symphony',
                     'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                     'name': 'Edo de Waart & San Francisco Symphony',
                     'offsetgroup': 'Edo de Waart & San Francisco Symphony',
                     'orientation': 'h',
                     'showlegend': True,
                     'textposition': 'auto',
                     'type': 'bar',
                     'x': array([306]),
                     'xaxis': 'x',
                     'y': array(['Adams, John: The Chairman Dances'], dtype=object),
                     'yaxis': 'y'},
                    {'alignmentgroup': 'True',
                     'hovertemplate': ('ArtistName=Antal Doráti & Lond' ... '}<br>Title=%{y}<extra></extra>'),
                     'legendgroup': 'Antal Doráti & London Symphony Orchestra',
                     'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
                     'name': 'Antal Doráti & London Symphony Orchestra',
                     'offsetgroup': 'Antal Doráti & London Symphony Orchestra',
                     'orientation': 'h',
                     'showlegend': True,
                     'textposition': 'auto',
                     'type': 'bar',
                     'x': array([307]),
                     'xaxis': 'x',
                     'y': array(["Tchaikovsky: 1812 Festival Overture, Op.49, Capriccio Italien & Beethoven: Wel
lington's Victory"],
                                dtype=object),
                     'yaxis': 'y'},
                    {'alignmentgroup': 'True',
                     'hovertemplate': ('ArtistName=Choir Of Westminste' ... '}<br>Title=%{y}<extra></extra>'),
                     'legendgroup': 'Choir Of Westminster Abbey & Simon Preston',
```

```
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'Choir Of Westminster Abbey & Simon Preston',
 'offsetgroup': 'Choir Of Westminster Abbey & Simon Preston',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([308]),
 'xaxis': 'x',
 'y': array(['Palestrina: Missa Papae Marcelli & Allegri: Miserere'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Michael Tilson Thom' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Michael Tilson Thomas & San Francisco Symphony',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Michael Tilson Thomas & San Francisco Symphony',
 'offsetgroup': 'Michael Tilson Thomas & San Francisco Symphony',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([309, 311]),
 'xaxis': 'x',
 'y': array(['Prokofiev: Romeo & Juliet', 'Berlioz: Symphonie Fantastique'],
           dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Chor der Wiener Sta' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Chor der Wiener Staatsoper, Herbert Von Karajan & Wiener Philharmoniker',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'Chor der Wiener Staatsoper, Herbert Von Karajan & Wiener Philharmoniker',
 'offsetgroup': 'Chor der Wiener Staatsoper, Herbert Von Karajan & Wiener Philharmoniker',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([312]),
 'xaxis': 'x',
 'y': array(['Bizet: Carmen Highlights'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': "ArtistName=The King's Singers<br>index=%{x}<br>Title=%{y}<extra></extra
```

```
>",
                 'legendgroup': "The King's Singers",
                 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
                 'name': "The King's Singers",
                 'offsetgroup': "The King's Singers",
                 'orientation': 'h',
                 'showlegend': True,
                 'textposition': 'auto',
                 'type': 'bar',
                 'x': array([313]),
                 'xaxis': 'x',
                 'y': array(['English Renaissance'], dtype=object),
                 'yaxis': 'y'},
                {'alignmentgroup': 'True',
                 'hovertemplate': ('ArtistName=Berliner Philharmon' ... '}<br>Title=%{y}<extra></extra>'),
                 'legendgroup': 'Berliner Philharmoniker & Herbert Von Karajan',
                 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
                 'name': 'Berliner Philharmoniker & Herbert Von Karajan',
                 'offsetgroup': 'Berliner Philharmoniker & Herbert Von Karajan',
                 'orientation': 'h',
                 'showlegend': True,
                 'textposition': 'auto',
                 'type': 'bar',
                 'x': array([315, 319, 335]),
                 'xaxis': 'x',
                 'y': array(['Grieg: Peer Gynt Suites & Sibelius: Pelléas et Mélisande',
                             'Mozart: Symphonies Nos. 40 & 41',
                             'Prokofiev: Symphony No.5 & Stravinksy: Le Sacre Du Printemps'],
                            dtype=object),
                 'yaxis': 'y'},
                {'alignmentgroup': 'True',
                 'hovertemplate': ('ArtistName=Sir Georg Solti, Su' ... '}<br>Title=%{y}<extra></extra>'),
                 'legendgroup': 'Sir Georg Solti, Sumi Jo & Wiener Philharmoniker',
                 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
                 'name': 'Sir Georg Solti, Sumi Jo & Wiener Philharmoniker',
                 'offsetgroup': 'Sir Georg Solti, Sumi Jo & Wiener Philharmoniker',
                 'orientation': 'h',
                 'showlegend': True,
                 'textposition': 'auto',
                 'type': 'bar',
                 'x': array([316]),
                 'xaxis': 'x',
```

```
                         'y': array(['Mozart Gala: Famous Arias'], dtype=object),
                         'yaxis': 'y'},
                        {'alignmentgroup': 'True',
                         'hovertemplate': "ArtistName=Christopher O'Riley<br>index=%{x}<br>Title=%{y}<extra></extra
>",
                         'legendgroup': "Christopher O'Riley",
                         'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
                         'name': "Christopher O'Riley",
                         'offsetgroup': "Christopher O'Riley",
                         'orientation': 'h',
                         'showlegend': True,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array([317]),
                         'xaxis': 'x',
                         'y': array(['SCRIABIN: Vers la flamme'], dtype=object),
                         'yaxis': 'y'},
                        {'alignmentgroup': 'True',
                         'hovertemplate': 'ArtistName=Fretwork<br>index=%{x}<br>Title=%{y}<extra></extra>',
                         'legendgroup': 'Fretwork',
                         'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
                         'name': 'Fretwork',
                         'offsetgroup': 'Fretwork',
                         'orientation': 'h',
                         'showlegend': True,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array([318]),
                         'xaxis': 'x',
                         'y': array(['Armada: Music from the Courts of England and Spain'], dtype=object),
                         'yaxis': 'y'},
                        {'alignmentgroup': 'True',
                         'hovertemplate': 'ArtistName=Amy Winehouse<br>index=%{x}<br>Title=%{y}<extra></extra>',
                         'legendgroup': 'Amy Winehouse',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                         'name': 'Amy Winehouse',
                         'offsetgroup': 'Amy Winehouse',
                         'orientation': 'h',
                         'showlegend': True,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array([320, 321]),
```

```
 'xaxis': 'x',
 'y': array(['Back to Black', 'Frank'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Calexico<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Calexico',
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': 'Calexico',
 'offsetgroup': 'Calexico',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([322]),
 'xaxis': 'x',
 'y': array(['Carried to Dust (Bonus Track Version)'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Otto Klemperer & Ph' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Otto Klemperer & Philharmonia Orchestra',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'Otto Klemperer & Philharmonia Orchestra',
 'offsetgroup': 'Otto Klemperer & Philharmonia Orchestra',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([323]),
 'xaxis': 'x',
 'y': array(["Beethoven: Symphony No. 6 'Pastoral' Etc."], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'ArtistName=Yehudi Menuhin<br>index=%{x}<br>Title=%{y}<extra></extra>',
 'legendgroup': 'Yehudi Menuhin',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Yehudi Menuhin',
 'offsetgroup': 'Yehudi Menuhin',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([324]),
```

```
 'xaxis': 'x',
 'y': array(['Bartok: Violin & Viola Concertos'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Philharmonia Orches' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Philharmonia Orchestra & Sir Neville Marriner',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'Philharmonia Orchestra & Sir Neville Marriner',
 'offsetgroup': 'Philharmonia Orchestra & Sir Neville Marriner',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([325]),
 'xaxis': 'x',
 'y': array(["Mendelssohn: A Midsummer Night's Dream"], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Academy of St. Mart' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Academy of St. Martin in the Fields, Sir Neville Marriner & Thurston Dart',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'Academy of St. Martin in the Fields, Sir Neville Marriner & Thurston Dart',
 'offsetgroup': 'Academy of St. Martin in the Fields, Sir Neville Marriner & Thurston Dart',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([326]),
 'xaxis': 'x',
 'y': array(['Bach: Orchestral Suites Nos. 1 - 4'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Les Arts Florissant' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Les Arts Florissants & William Christie',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'Les Arts Florissants & William Christie',
 'offsetgroup': 'Les Arts Florissants & William Christie',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([327]),
```

```
 'xaxis': 'x',
 'y': array(['Charpentier: Divertissements, Airs & Concerts'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=The 12 Cellists of ' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'The 12 Cellists of The Berlin Philharmonic',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'The 12 Cellists of The Berlin Philharmonic',
 'offsetgroup': 'The 12 Cellists of The Berlin Philharmonic',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([328]),
 'xaxis': 'x',
 'y': array(['South American Getaway'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Adrian Leaper & Dor' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Adrian Leaper & Doreen de Feis',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'Adrian Leaper & Doreen de Feis',
 'offsetgroup': 'Adrian Leaper & Doreen de Feis',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([329]),
 'xaxis': 'x',
 'y': array(['Górecki: Symphony No. 3'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Roger Norrington, L' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': 'Roger Norrington, London Classical Players',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'Roger Norrington, London Classical Players',
 'offsetgroup': 'Roger Norrington, London Classical Players',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([330]),
```

```
 'xaxis': 'x',
 'y': array(['Purcell: The Fairy Queen'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ("ArtistName=Charles Dutoit & L'" ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': "Charles Dutoit & L'Orchestre Symphonique de Montréal",
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': "Charles Dutoit & L'Orchestre Symphonique de Montréal",
 'offsetgroup': "Charles Dutoit & L'Orchestre Symphonique de Montréal",
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([331]),
 'xaxis': 'x',
 'y': array(['The Ultimate Relexation Album'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Equale Brass Ensemb' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': ('Equale Brass Ensemble, John El' ... 'Monteverdi Orchestra and Choir'),
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': ('Equale Brass Ensemble, John El' ... 'Monteverdi Orchestra and Choir'),
 'offsetgroup': ('Equale Brass Ensemble, John El' ... 'Monteverdi Orchestra and Choir'),
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([332]),
 'xaxis': 'x',
 'y': array(['Purcell: Music for the Queen Mary'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': ('ArtistName=Kent Nagano and Orc' ... '}<br>Title=%{y}<extra></extra>'),
 'legendgroup': "Kent Nagano and Orchestre de l'Opéra de Lyon",
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': "Kent Nagano and Orchestre de l'Opéra de Lyon",
 'offsetgroup': "Kent Nagano and Orchestre de l'Opéra de Lyon",
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([333]),
```

```
                     'xaxis': 'x',
                     'y': array(['Weill: The Seven Deadly Sins'], dtype=object),
                     'yaxis': 'y'},
                    {'alignmentgroup': 'True',
                     'hovertemplate': 'ArtistName=Julian Bream<br>index=%{x}<br>Title=%{y}<extra></extra>',
                     'legendgroup': 'Julian Bream',
                     'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
                     'name': 'Julian Bream',
                     'offsetgroup': 'Julian Bream',
                     'orientation': 'h',
                     'showlegend': True,
                     'textposition': 'auto',
                     'type': 'bar',
                     'x': array([334]),
                     'xaxis': 'x',
                     'y': array(['J.S. Bach: Chaconne, Suite in E Minor, Partita in E Major & Prelude, Fugue and
Allegro'],
                                dtype=object),
                     'yaxis': 'y'},
                    {'alignmentgroup': 'True',
                     'hovertemplate': 'ArtistName=Martin Roscoe<br>index=%{x}<br>Title=%{y}<extra></extra>',
                     'legendgroup': 'Martin Roscoe',
                     'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
                     'name': 'Martin Roscoe',
                     'offsetgroup': 'Martin Roscoe',
                     'orientation': 'h',
                     'showlegend': True,
                     'textposition': 'auto',
                     'type': 'bar',
                     'x': array([336]),
                     'xaxis': 'x',
                     'y': array(['Szymanowski: Piano Works, Vol. 1'], dtype=object),
                     'yaxis': 'y'},
                    {'alignmentgroup': 'True',
                     'hovertemplate': ('ArtistName=Göteborgs Symfonike' ... '}<br>Title=%{y}<extra></extra>'),
                     'legendgroup': 'Göteborgs Symfoniker & Neeme Järvi',
                     'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
                     'name': 'Göteborgs Symfoniker & Neeme Järvi',
                     'offsetgroup': 'Göteborgs Symfoniker & Neeme Järvi',
                     'orientation': 'h',
                     'showlegend': True,
                     'textposition': 'auto',
```

```
                        'type': 'bar',
                        'x': array([337]),
                        'xaxis': 'x',
                        'y': array(['Nielsen: The Six Symphonies'], dtype=object),
                        'yaxis': 'y'},
                       {'alignmentgroup': 'True',
                        'hovertemplate': 'ArtistName=Itzhak Perlman<br>index=%{x}<br>Title=%{y}<extra></extra>',
                        'legendgroup': 'Itzhak Perlman',
                        'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
                        'name': 'Itzhak Perlman',
                        'offsetgroup': 'Itzhak Perlman',
                        'orientation': 'h',
                        'showlegend': True,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array([338]),
                        'xaxis': 'x',
                        'y': array(["Great Recordings of the Century: Paganini's 24 Caprices"], dtype=object),
                        'yaxis': 'y'},
                       {'alignmentgroup': 'True',
                        'hovertemplate': 'ArtistName=Michele Campanella<br>index=%{x}<br>Title=%{y}<extra></extra
                    >',
                        'legendgroup': 'Michele Campanella',
                        'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
                        'name': 'Michele Campanella',
                        'offsetgroup': 'Michele Campanella',
                        'orientation': 'h',
                        'showlegend': True,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array([339]),
                        'xaxis': 'x',
                        'y': array(["Liszt - 12 Études D'Execution Transcendante"], dtype=object),
                        'yaxis': 'y'},
                       {'alignmentgroup': 'True',
                        'hovertemplate': 'ArtistName=Gerald Moore<br>index=%{x}<br>Title=%{y}<extra></extra>',
                        'legendgroup': 'Gerald Moore',
                        'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
                        'name': 'Gerald Moore',
                        'offsetgroup': 'Gerald Moore',
                        'orientation': 'h',
                        'showlegend': True,
```

```
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([340]),
                      'xaxis': 'x',
                      'y': array(['Great Recordings of the Century - Shubert: Schwanengesang, 4 Lieder'],
                                 dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': ('ArtistName=Mela Tenenbaum, Pro' ... '}<br>Title=%{y}<extra></extra>'),
                      'legendgroup': 'Mela Tenenbaum, Pro Musica Prague & Richard Kapp',
                      'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
                      'name': 'Mela Tenenbaum, Pro Musica Prague & Richard Kapp',
                      'offsetgroup': 'Mela Tenenbaum, Pro Musica Prague & Richard Kapp',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([341]),
                      'xaxis': 'x',
                      'y': array(['Locatelli: Concertos for Violin, Strings and Continuo, Vol. 3'],
                                 dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=Emerson String Quartet<br>index=%{x}<br>Title=%{y}<extra></ext
ra>',
                      'legendgroup': 'Emerson String Quartet',
                      'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                      'name': 'Emerson String Quartet',
                      'offsetgroup': 'Emerson String Quartet',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([343]),
                      'xaxis': 'x',
                      'y': array(["Schubert: The Late String Quartets & String Quintet (3 CD's)"],
                                 dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': ('ArtistName=C. Monteverdi, Nige' ... '}<br>Title=%{y}<extra></extra>'),
                      'legendgroup': ('C. Monteverdi, Nigel Rogers - ' ... 'roque; London Cornett & Sackbu'),
                      'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
```

```
                      'name': ('C. Monteverdi, Nigel Rogers - ' ... 'roque; London Cornett & Sackbu'),
                      'offsetgroup': ('C. Monteverdi, Nigel Rogers - ' ... 'roque; London Cornett & Sackbu'),
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([344]),
                      'xaxis': 'x',
                      'y': array(["Monteverdi: L'Orfeo"], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=Nash Ensemble<br>index=%{x}<br>Title=%{y}<extra></extra>',
                      'legendgroup': 'Nash Ensemble',
                      'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
                      'name': 'Nash Ensemble',
                      'offsetgroup': 'Nash Ensemble',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([345]),
                      'xaxis': 'x',
                      'y': array(['Mozart: Chamber Music'], dtype=object),
                      'yaxis': 'y'},
                     {'alignmentgroup': 'True',
                      'hovertemplate': 'ArtistName=Philip Glass Ensemble<br>index=%{x}<br>Title=%{y}<extra></extr
a>',
                      'legendgroup': 'Philip Glass Ensemble',
                      'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
                      'name': 'Philip Glass Ensemble',
                      'offsetgroup': 'Philip Glass Ensemble',
                      'orientation': 'h',
                      'showlegend': True,
                      'textposition': 'auto',
                      'type': 'bar',
                      'x': array([346]),
                      'xaxis': 'x',
                      'y': array(['Koyaanisqatsi (Soundtrack from the Motion Picture)'], dtype=object),
                      'yaxis': 'y'}],
          'layout': {'barmode': 'relative',
                     'legend': {'title': {'text': 'ArtistName'}, 'tracegroupgap': 0},
                     'margin': {'t': 60},
```

```
                'template': '...',
                'title': {'text': 'Albums by Artist'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'index'}},
                '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   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   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', 'co
ntent': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'co
ntent': 'SELECT Name, UnitPrice\nFROM "tracks"\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'conte
nt': ' \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.A
rtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT g2.GenreId\n FROM "genres" g2\n JOIN "tracks" t2 O
N 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 t
racks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n   Can y
ou 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.Arti
stId\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 (acros
s all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlb
ums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice_items" ii ON i.Inv
oiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'conte
nt': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}]
Ollama parameters:
model=mistral: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 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   Med
iaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTE
GER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId)
REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreI
d) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (Med
iaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n
CREATE INDEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON \"tracks\" (M
ediaTypeId)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INT
EGER 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 AUTOINCREMEN
T 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   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\nORDER BY TotalTracks DESC\nLIMIT
5"}, {"role": "user", "content": " \n   Find the top 5 most expensive tracks (based on unit price):\n"},
{"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.Unit
Price DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find the top 5 most expensive tracks (based on
unit price):\n"}, {"role": "assistant", "content": "SELECT Name, UnitPrice\nFROM \"tracks\"\nORDER BY UnitP
rice DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Identify artists who have albums with tracks ap
pearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistN
ame\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT g
2.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", "conte
nt": " \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 the c
ustomer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "co
ntent": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i O
N c.CustomerId = i.CustomerId\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.Customer
Id\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n   Find all tracks with a name
containing \"What\" (case-insensitive)\n"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:22:31.070897089Z', 'message': {'role': 'assistan

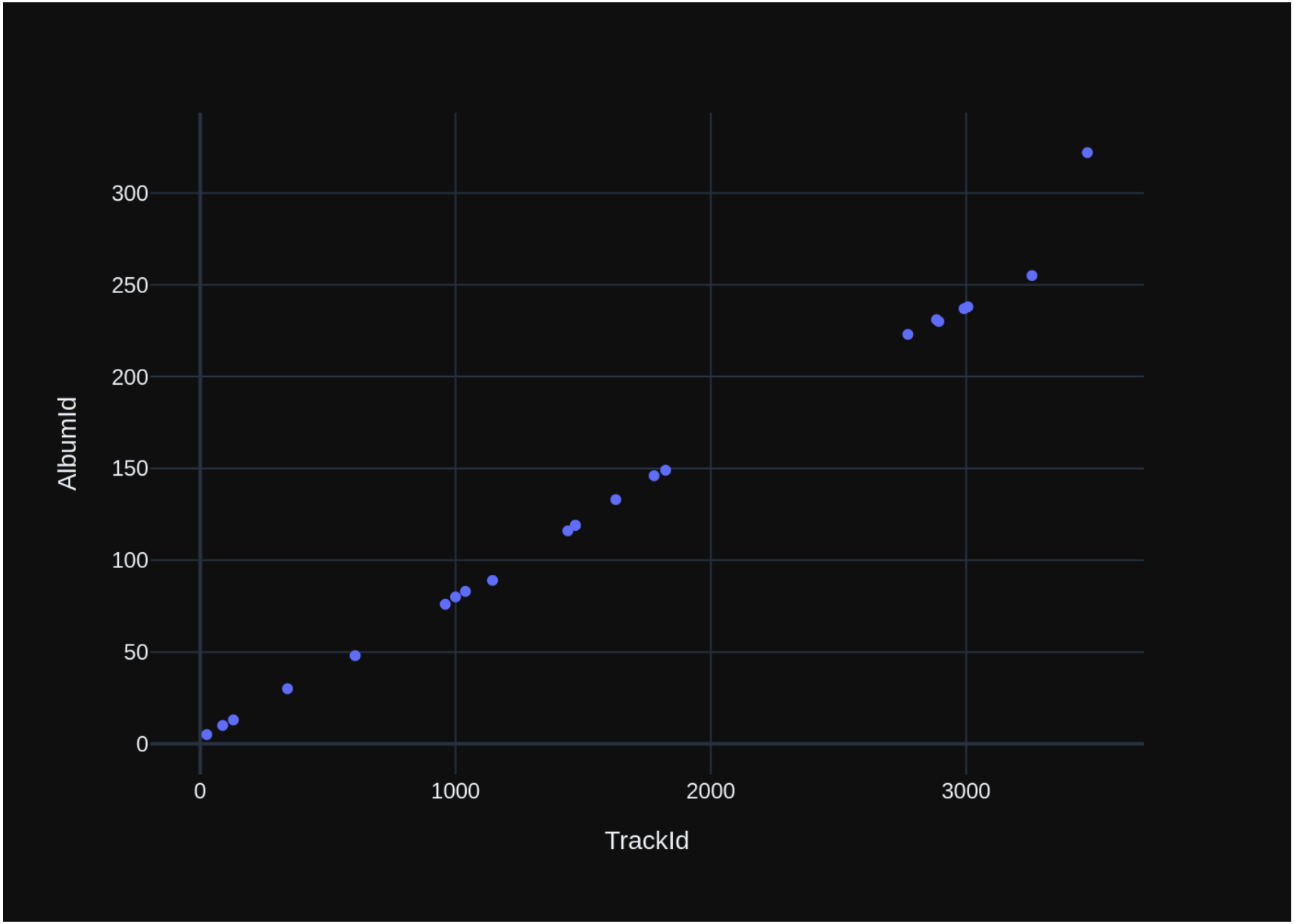
```
t', 'content': "SELECT * FROM tracks WHERE lower(Name) LIKE '%what%'"}, 'done_reason': 'stop', 'done': Tru
e, 'total_duration': 86368255151, 'load_duration': 1052620, 'prompt_eval_count': 1835, 'prompt_eval_duratio
n': 82882638000, 'eval_count': 16, 'eval_duration': 2837423000}
SELECT * FROM tracks 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=mistral:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Find all tracks with a name containing \"What\" (case-in
sensitive)\n'\n\nThe DataFrame was produced using this query: SELECT * FROM tracks WHERE lower(Name) LIKE
```

'%what%'\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n TrackId   int64\nName   object\nAlbumId   int64\nMediaTypeId   int64\nGenreId   int64\nComposer   object\nMilliseconds   int64\nBytes   int64\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 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': 'mistral:latest', 'created_at': '2024-06-14T10:22:56.63424517Z', 'message': {'role': 'assistant', 'content': ' ```python\nimport plotly.express as px\n\nif df.shape[0] > 1:\n   fig = px.bar("Name", "Milliseconds", data=df)\nelse:\n   fig = px.bar(px.data.treatment(), orientation=\'h\')\nfig.update_layout(title="Tracks Containing \'What\'")\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 25538149369, 'load_duration': 708027, 'prompt_eval_count': 229, 'prompt_eval_duration': 9240488000, 'eval_count': 85, 'eval_duration': 16203132000}


AlbumId
300
250
200
150
100
50
0
0
1000
2000
3000
TrackId

Out[22]: ("SELECT * FROM tracks WHERE 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': [{'hovertemplate': 'TrackId=%{x}<br>AlbumId=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'symbol': 'circle'},
              'mode': 'markers',
              'name': '',
              'orientation': 'v',
              'showlegend': False,
              'type': 'scatter',
              'x': array([  26,   88,  130,  342,  607,  960, 1000, 1039, 1145, 1440, 1469, 1470,
                         1628, 1778, 1823, 2772, 2884, 2893, 2992, 3007, 3258, 3475]),
              'xaxis': 'x',
```

```
                'y': array([  5,  10,  13,  30,  48,  76,  80,  83,  89, 116, 119, 119, 133, 146,
                           149, 223, 231, 230, 237, 238, 255, 322]),
                'yaxis': 'y'}],
     'layout': {'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'TrackId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'AlbumId'}}}
 }))
```

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 total number of invoices per country:\n'}, {'role': 'assistant', 'conten
t': 'SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM "customers" customer\nJOIN "in
voices" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country'}, {'role': 'user', 'conten
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.Cus
tomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n
Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(I
nvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT
1\n-------------------------'}, {'role': 'user', 'content': ' \n   Get the average invoice total for each
customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\n
FROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'us
er', '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   Get the average invoice total for each customer:\n'}, {'role': 'assist
ant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoi
ces" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY AverageInvoiceTotal DESC'}, {'role':
'user', 'content': ' \n   Find the customer who bought the most albums in total quantity (across all inv
oices): \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 TotalA
lbums DESC\nLIMIT 1\n-------------------------'}, {'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   Find the top 5 customers who spent the mo
st money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice
_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Tota
l) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Customer
Id\nORDER BY TotalSpent DESC\nLIMIT 5\n-------------------------'}, {'role': 'user', 'content': ' \n   Ge
t the total number of invoices for each customer\n'}]
Ollama parameters:
model=mistral: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 total number of invoices per cou
ntry:\n"}, {"role": "assistant", "content": "SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCo
unt\nFROM \"customers\" customer\nJOIN \"invoices\" invoice ON customer.CustomerId = invoice.CustomerId\nGR
OUP BY Country"}, {"role": "user", "content": " \n   Find the customer with the most invoices \n"}, {"ro
le": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\"
c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\n
LIMIT 1"}, {"role": "user", "content": " \n   Find the customer with the most invoices \n"}, {"role": "a
ssistant", "content": "SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY C

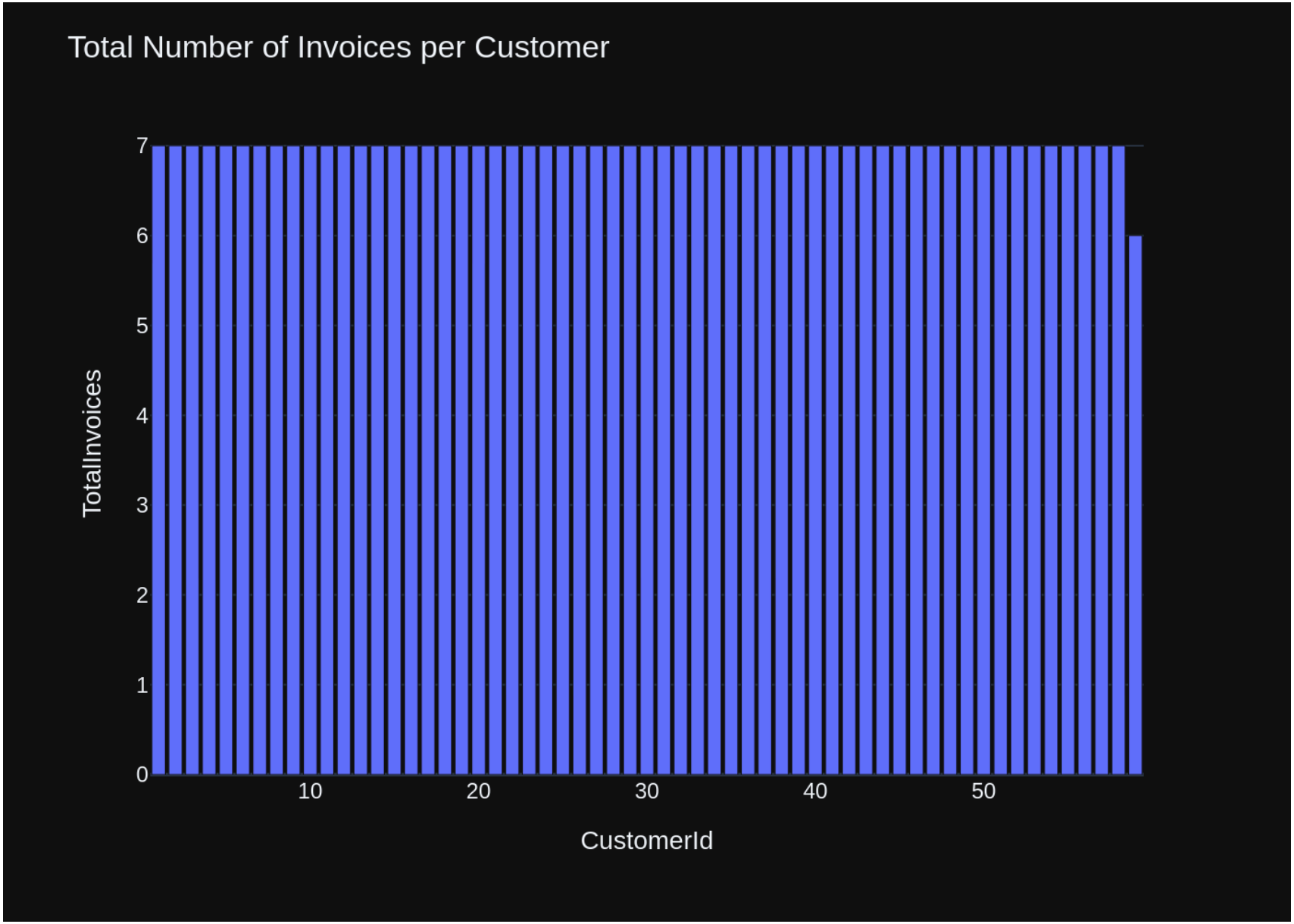
ustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-------------------------"}, {"role": "user", "conten
t": " \n   Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT
c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerI
d = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   Find the total number of in
voices per country:\n"}, {"role": "assistant", "content": "SELECT i.BillingCountry, COUNT(*) AS TotalInvoic
es\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "content": " \n   Get the average
invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) A
S AverageInvoiceTotal\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY
c.CustomerId\nORDER BY AverageInvoiceTotal DESC"}, {"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 quanti
ty (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 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\nLIM
IT 5\n-------------------------"}, {"role": "user", "content": " \n   Get the total number of invoices fo
r each customer\n"}]

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

```
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:24:39.467759452Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoice
s" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, 'done_reason': 'stop', 'done': True, 'total_d
uration': 102744231979, 'load_duration': 768015, 'prompt_eval_count': 2026, 'prompt_eval_duration': 9185489
1000, 'eval_count': 54, 'eval_duration': 10126001000}
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=mistral: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 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': 'mistral:latest', 'created_at': '2024-06-14T10:25:27.942287567Z', 'message': {'role': 'assistant', 'content': ' Here\'s the Python code to create a bar chart using Plotly Express:\n\n```python\nimport plotly.express as px\n\nfig = px.bar(df, x="CustomerId", y="TotalInvoices")\nfig.update_layout(title=\'Total Number of Invoices per Customer\')\nfig.show()\n```\n\nIf there\'s only one value in the dataframe:\n\n```python\nimport plotly.graph_objects as go\n\nfig = go.Figure(data=go.Indicator(\n   mode="gauge+number+delta",\n   value=df.iloc[0][\'TotalInvoices\'],\n   domain={"x": [0, 1], "y": [0, .6]},\n   title={\'text\': \'Total Invoices\'},\n   delta={\'reference\': 0}\n))\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 48446810527, 'load_duration': 42170001, 'prompt_eval_count': 218, 'prompt_eval_duration': 9978690000, 'eval_count': 202, 'eval_duration': 38380861000}


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

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

```
                    'margin': {'t': 60},
                    'template': '...',
                    'title': {'text': 'Total Number of Invoices per Customer'},
                    'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                    'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalInvoices'}}}
 }))
```

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
```

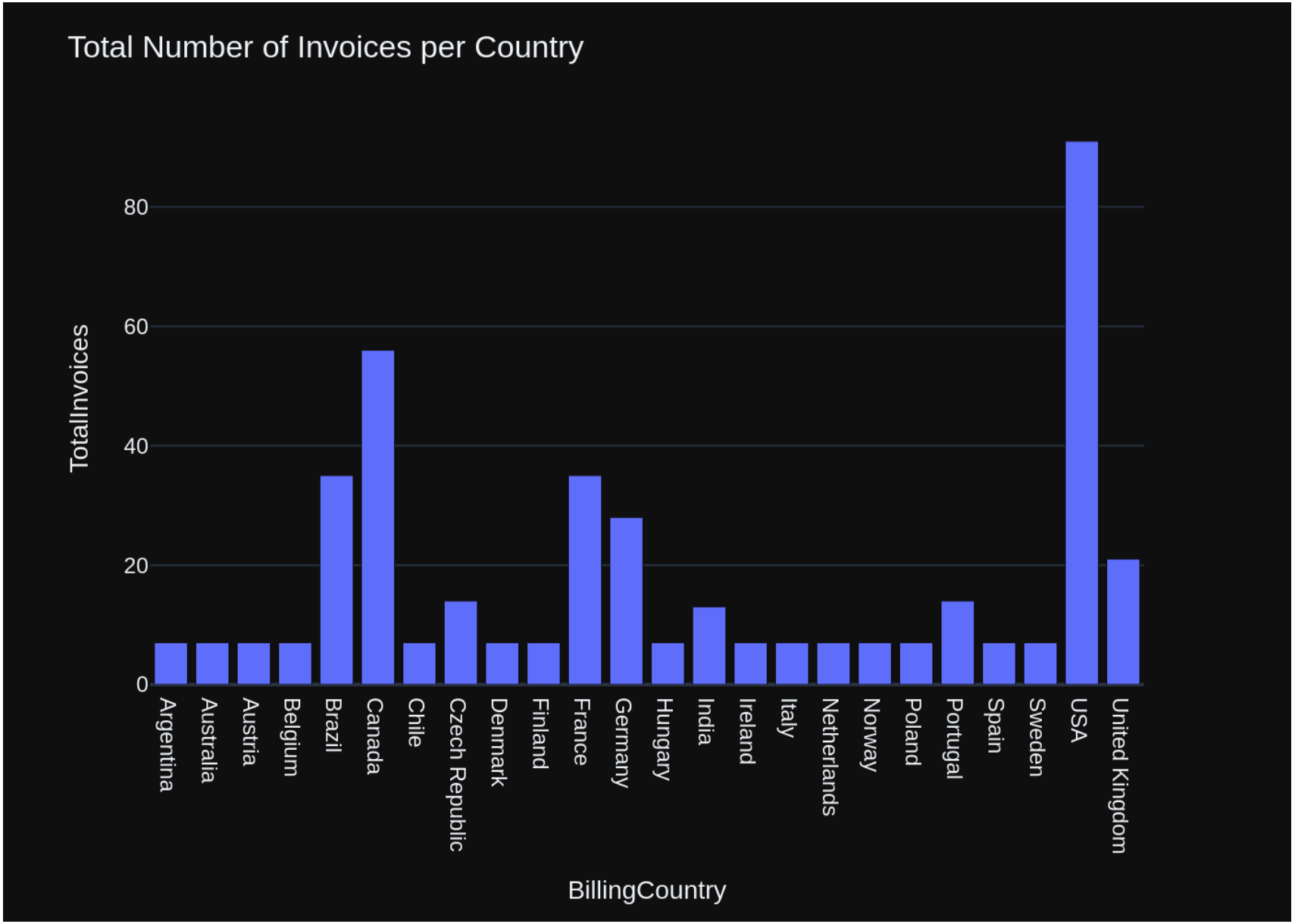
```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER 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_InvoiceCustomerId ON "invoices" (Custome
rId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceLin
eTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE "employees"\r\n(\r\n    EmployeeId INTEGER PRIMARY KE
Y AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NUL
L,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIM
E,\r\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 O
N UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company
NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Coun
try 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 "em
ployees" (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    Arti
stId 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 AUTOINCREM
ENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT
NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n
Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums"
(AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres"
(GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "med
ia_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeRe
portsTo ON "employees" (ReportsTo)\n\n\n===Additional Context \n\nIn the SQLite database invoice means orde
r\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query w
ithout any explanations for the question. \n2. If the provided context is almost sufficient but requires kn
owledge of a specific string in a particular column, please generate an intermediate SQL query to find the
distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provi
ded context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant t
able(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it wa
s given before. \n'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per countr
```

y:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM "invoice
s" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': ' \n   Find the total number of invoices p
er country:\n'}, {'role': 'assistant', 'content': 'SELECT Customer.Country, COUNT(invoice.InvoiceId) AS Inv
oiceCount\nFROM "customers" customer\nJOIN "invoices" invoice ON customer.CustomerId = invoice.CustomerId\n
GROUP 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 "cus
tomers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'con
tent': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'con
tent': '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 i
nvoices \n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nF
ROM Invoices \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-------------------------'},
{'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistan
t', 'content': 'SELECT *\nFROM "invoices" \nWHERE (Total > 10)\nGROUP BY InvoiceId'}, {'role': 'user', 'con
tent': ' \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'}, {'role': 'assistant', 'content': 'SE
LECT 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   Hint:
album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums i
n total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(i
i.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\n
GROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'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   Find the total num
ber of invoices per country:\n'}]
Ollama parameters:
model=mistral: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_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   Find
the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT Customer.Country, C
OUNT(invoice.InvoiceId) AS InvoiceCount\nFROM \"customers\" customer\nJOIN \"invoices\" invoice ON custome
r.CustomerId = invoice.CustomerId\nGROUP BY Country"}, {"role": "user", "content": " \n   Get the total n
umber of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.Inv
oiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP B
Y c.CustomerId"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amount in

```
voiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"in
voices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"role": "user", "content": "
\n     Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT CustomerId,
COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC
\nLIMIT 1\n-------------------------"}, {"role": "user", "content": "  \n    List all invoices with a total
exceeding $10:\n"}, {"role": "assistant", "content": "SELECT *\nFROM \"invoices\" \nWHERE (Total > 10)\nGRO
UP BY InvoiceId"}, {"role": "user", "content": "  \n     Find the customer with the most invoices \n"}, {"r
ole": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\"
c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\n
LIMIT 1"}, {"role": "user", "content": "  \n    Find all invoices since 2010 and the total amount invoice
d:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount\nFROM \"invoice
s\" i\nWHERE strftime('%Y', i.InvoiceDate) >= '2010'\nGROUP BY i.InvoiceId\nORDER BY InvoiceAmount DESC"},
{"role": "user", "content": "  \n    Hint: album quantity is found in invoice_items, \n    \n    Find the t
op 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistan
t", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"in
voice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT
5"}, {"role": "user", "content": "  \n    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 \"invo
ice_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"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:27:05.664604384Z', 'message': {'role': 'assistan
t', 'content': 'SELECT BillingCountry, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "invoices" i\nGROUP BY Bil
lingCountry'}, 'done_reason': 'stop', 'done': True, 'total_duration': 97599009451, 'load_duration': 947696,
'prompt_eval_count': 2030, 'prompt_eval_duration': 90511294000, 'eval_count': 34, 'eval_duration': 62987220
00}
SELECT BillingCountry, COUNT(i.InvoiceId) AS TotalInvoices
FROM "invoices" i
GROUP BY BillingCountry
SELECT BillingCountry, COUNT(i.InvoiceId) AS TotalInvoices
FROM "invoices" i
GROUP BY BillingCountry
    BillingCountry  TotalInvoices
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=mistral:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Find the total number of invoices per country:\n'\n\nThe
DataFrame was produced using this query: SELECT BillingCountry, COUNT(i.InvoiceId) AS TotalInvoices\nFROM
\"invoices\" i\nGROUP BY BillingCountry\n\nThe following is information about the resulting pandas DataFram
e 'df': \nRunning df.dtypes gives:\n BillingCountry    object\nTotalInvoices      int64\ndtype: object"},
{"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe?
Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an I
ndicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:27:40.266220629Z', 'message': {'role': 'assistan
t', 'content': ' ```python\nimport plotly.express as px\n\nfig = px.bar(df, x=\'BillingCountry\', y=\'Total
Invoices\')\nfig.update_layout(title="Total Number of Invoices per Country")\n\nif len(df) == 1:\n    fig =
px.scatter(df, x=[df.index[0]], y=\'TotalInvoices\', mode=\'markers+text\', marker=dict(symbol=\'star\', si
ze=20), textposition=\'top center\')\n    fig.update_layout(title="Single Invoice Count: {}".format(df.iloc
[0,0]))\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 34580530273, 'load_duration': 139571
2, 'prompt_eval_count': 197, 'prompt_eval_duration': 8368913000, 'eval_count': 146, 'eval_duration': 261218
52000}
```


Total Number of Invoices per Country
TotalInvoices
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
BillingCountry

Out[24]: ('SELECT BillingCountry, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "invoices" i\nGROUP BY BillingCountry',

	BillingCountry	TotalInvoices
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': 'BillingCountry=%{x}<br>TotalInvoices=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'v',
              'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
              'x': array(['Argentina', 'Australia', 'Austria', 'Belgium', 'Brazil', 'Canada',
                          'Chile', 'Czech Republic', 'Denmark', 'Finland', 'France', 'Germany',
                          'Hungary', 'India', 'Ireland', 'Italy', 'Netherlands', 'Norway',
                          'Poland', 'Portugal', 'Spain', 'Sweden', 'USA', 'United Kingdom'],
                         dtype=object),
```

```
                'xaxis': 'x',
                'y': array([ 7,  7,  7,  7, 35, 56,  7, 14,  7,  7, 35, 28,  7, 13,  7,  7,  7,  7,
                             7, 14,  7,  7, 91, 21]),
                'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'title': {'text': 'Total Number of Invoices per Country'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'BillingCountry'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalInvoices'}}}
 }))
```

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   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   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 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 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\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 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 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   List all invoices with a total exceeding $10:\n'}]
Ollama parameters:
model=mistral:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE \"invoice_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) 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   List all invoices with a total exceeding $1
0:\n"}, {"role": "assistant", "content": "SELECT *\nFROM \"invoices\" \nWHERE (Total > 10)\nGROUP BY Invoic
eId"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amount invoice
d:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"invoice
s\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"role": "user", "content": " \n
Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT

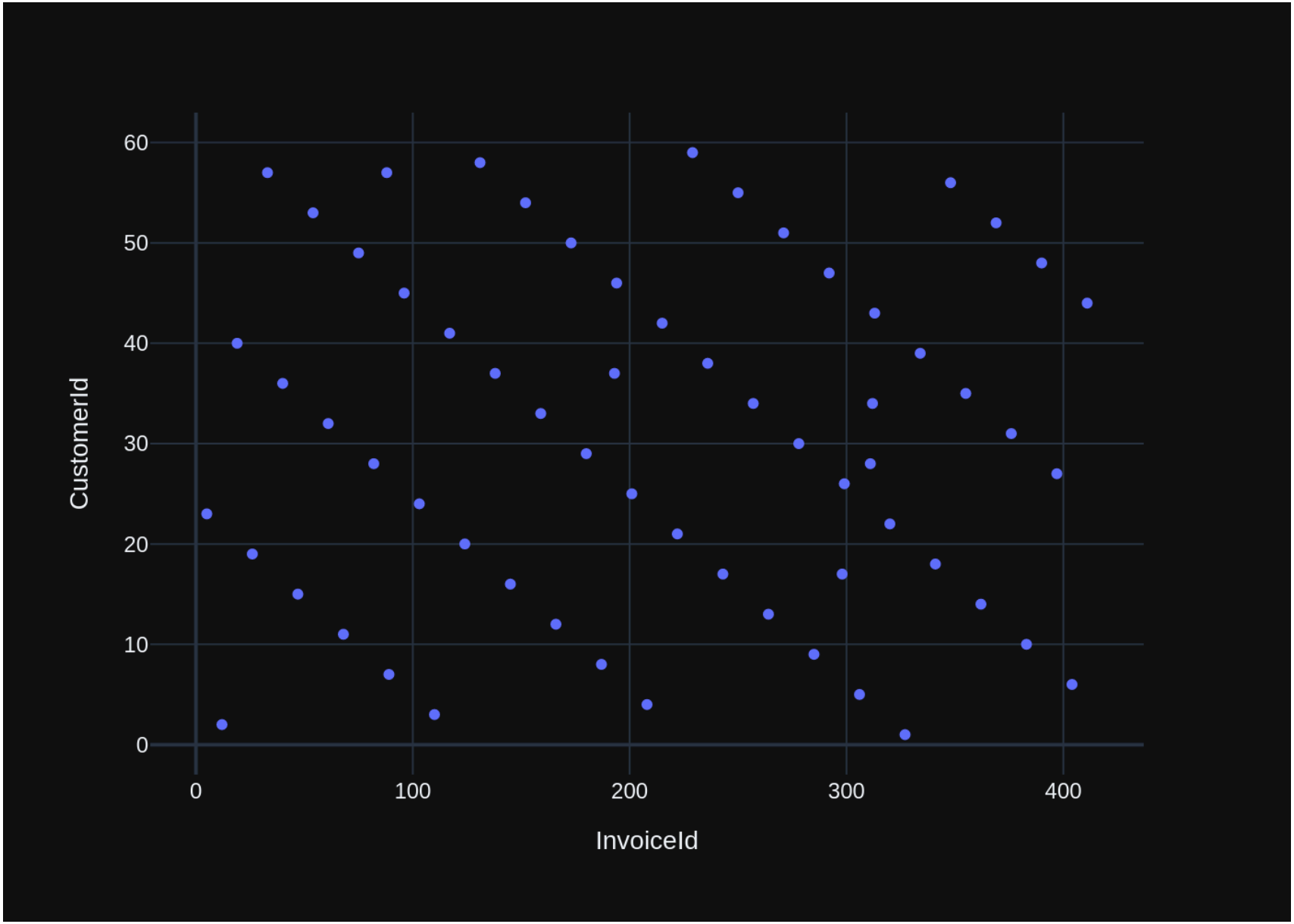
```
i.InvoiceId, SUM(i.Total) AS InvoiceAmount\nFROM \"invoices\" i\nWHERE strftime('%Y', i.InvoiceDate) >= '20
10'\nGROUP BY i.InvoiceId\nORDER BY InvoiceAmount DESC"}, {"role": "user", "content": "  \n     Find the to
p 5 customers who spent the most money overall, \n     \n     Hint: order total can 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\n-------------------------"}, {"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\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\n-------------------------"}, {"role": "user", "content": "  \n    Hint: album quantity is found in invoi
ce_items, \n    \n    Find the top 5 customers who bought the most albums in total quantity (across all inv
oices):\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\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 c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFRO
M \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice_items\" ii ON i.Invoi
ceId = 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.Invoic
eId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": "  \n    List
all invoices with a total exceeding $10:\n"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:29:09.537193706Z', 'message': {'role': 'assistan
t', 'content': 'SELECT * FROM "invoices" WHERE Total > 10;'}, 'done_reason': 'stop', 'done': True, 'total_d
uration': 89170121983, 'load_duration': 924298, 'prompt_eval_count': 1957, 'prompt_eval_duration': 85390901
000, 'eval_count': 16, 'eval_duration': 2839663000}
SELECT * FROM "invoices" WHERE Total > 10;
Output from LLM: SELECT * FROM "invoices" WHERE Total > 10;
Extracted SQL: SELECT * FROM "invoices" WHERE Total > 10
SELECT * FROM "invoices" WHERE Total > 10
    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=mistral:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all invoices with a total exceeding $10:\n'\n\nThe
DataFrame was produced using this query: SELECT * FROM \"invoices\" WHERE Total > 10\n\nThe following is in
formation about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n InvoiceId              in
t64\nCustomerId             int64\nInvoiceDate           object\nBillingAddress        object\nBillingCity
object\nBillingState          object\nBillingCountry        object\nBillingPostalCode     object\nTotal
float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the
results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value
in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- j
ust the code."}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:29:33.818677879Z', 'message': {'role': 'assistan
t', 'content': ' ```python\nimport plotly.express as px\n\nif df.shape[0] > 1:\n    fig = px.bar("InvoiceI
d", "Total", data=df)\nelse:\n    fig = px.Indicator(data_frame=df, mode="gauge+number", value_prefix="Tota
l: $")\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 24256061150, 'load_duration': 690208,
```

'prompt_eval_count': 226, 'prompt_eval_duration': 9760335000, 'eval_count': 78, 'eval_duration': 14399832000}



Out[25]:
```
('SELECT * FROM "invoices" WHERE Total > 10',
    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': [{'hovertemplate': 'InvoiceId=%{x}<br>CustomerId=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'symbol': 'circle'},
              'mode': 'markers',
              'name': '',
              'orientation': 'v',
              'showlegend': False,
              'type': 'scatter',
              '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([23,  2, 40, 19, 57, 36, 15, 53, 32, 11, 49, 28, 57,  7, 45, 24,  3, 41,
                            20, 58, 37, 16, 54, 33, 12, 50, 29,  8, 37, 46, 25,  4, 42, 21, 59, 38,
                            17, 55, 34, 13, 51, 30,  9, 47, 17, 26,  5, 28, 34, 43, 22,  1, 39, 18,
                            56, 35, 14, 52, 31, 10, 48, 27,  6, 44]),
                'yaxis': 'y'}],
     'layout': {'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': 'CustomerId'}}}
 }))
```

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   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 strfti
me(\'%Y\', i.InvoiceDate) >= \'2010\'\nGROUP BY i.InvoiceId\nORDER BY InvoiceAmount DESC'}, {'role': 'use
r', '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 * FROM "invoice
s" WHERE Total > 10'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per countr
y:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(i.InvoiceId) AS TotalInvoices\nFROM
"invoices" i\nGROUP BY BillingCountry'}, {'role': 'user', 'content': ' \n   List all invoices with a tota
l 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': 'assistan
t', 'content': 'SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.BillingCo
untry'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role':
'assistant', 'content': 'SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM "customer
s" customer\nJOIN "invoices" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country'}, {'rol
e': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assistan
t', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoice
s" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n   Find th
e 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 all invoices since 2010 and the total amount invoi
ced:\n'}]
Ollama parameters:
model=mistral: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

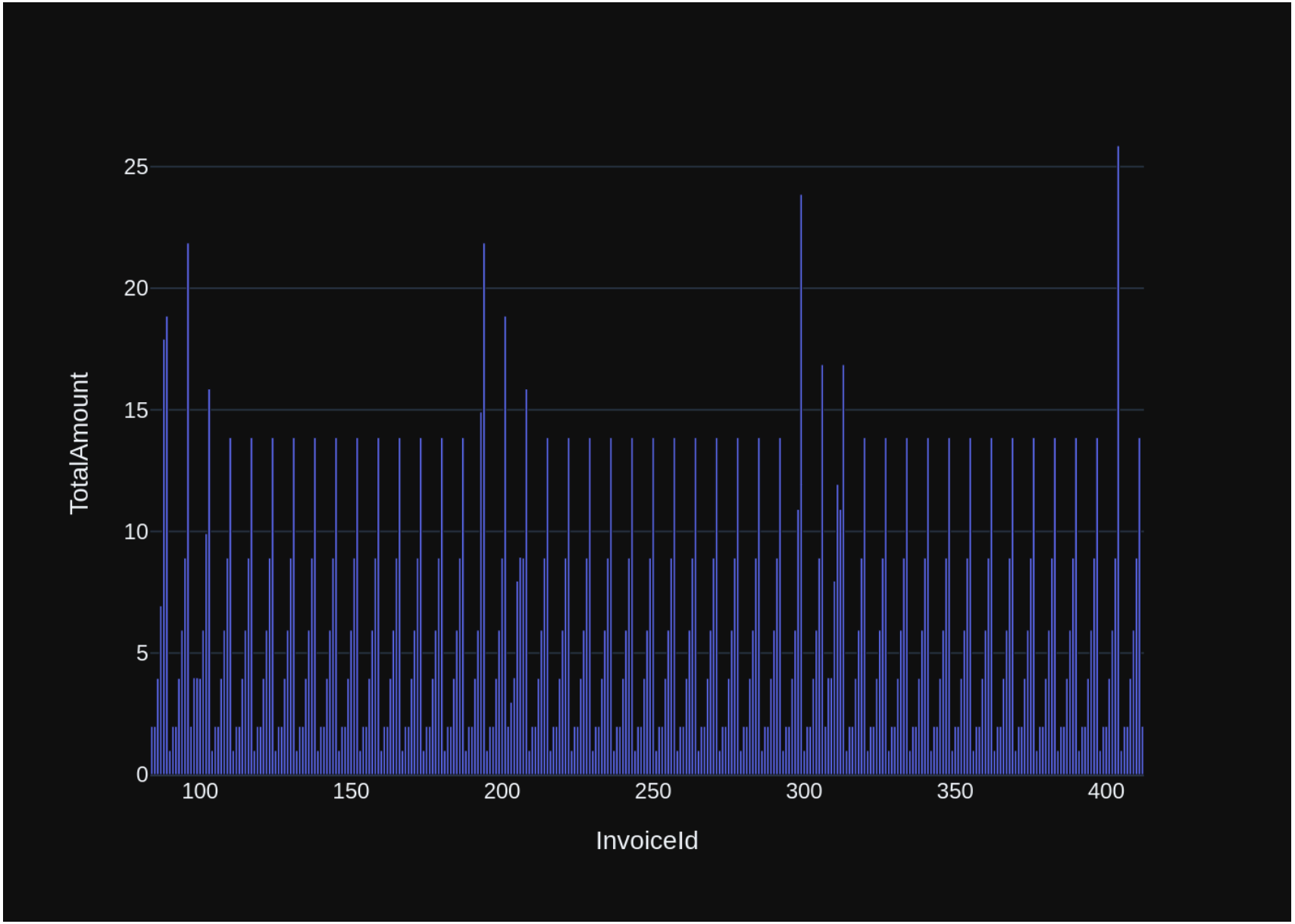
T NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tracks\" (Tra
ckId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId 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   Find all invoices since 2010 and the tot
al amount invoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmoun
t\nFROM \"invoices\" i\nWHERE strftime('%Y', i.InvoiceDate) >= '2010'\nGROUP BY i.InvoiceId\nORDER BY Invoi
ceAmount DESC"}, {"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": "assistan
t", "content": "SELECT * FROM \"invoices\" WHERE Total > 10"}, {"role": "user", "content": " \n   Find th
e total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT BillingCountry, COUNT
(i.InvoiceId) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY BillingCountry"}, {"role": "user", "content":
" \n   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 invo
ices per country:\n"}, {"role": "assistant", "content": "SELECT i.BillingCountry, COUNT(*) AS TotalInvoices
\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "content": " \n   Find the total num
ber 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 invoic
es for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS Tota
lInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerI
d"}, {"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.Custome
rId\nORDER BY TotalAlbums DESC\nLIMIT 1\n-------------------------"}, {"role": "user", "content": " \n
Find all invoices since 2010 and the total amount invoiced:\n"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:31:15.149185057Z', 'message': {'role': 'assistan
t', 'content': 'SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >
\'2010-01-01\'\nGROUP BY i.InvoiceId'}, 'done_reason': 'stop', 'done': True, 'total_duration': 10120290811
0, 'load_duration': 805028, 'prompt_eval_count': 2020, 'prompt_eval_duration': 90125709000, 'eval_count': 5
5, 'eval_duration': 10336698000}
```
SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount
FROM "invoices" i
WHERE i.InvoiceDate > '2010-01-01'
GROUP BY i.InvoiceId
```
SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount
FROM "invoices" i
WHERE i.InvoiceDate > '2010-01-01'
GROUP BY i.InvoiceId

	InvoiceId	TotalAmount
0	84	1.98
1	85	1.98
2	86	3.96
3	87	6.94
4	88	17.91
..	...	...
324	408	3.96
325	409	5.94

```
326        410         8.91
327        411        13.86
328        412         1.99

[329 rows x 2 columns]
Ollama parameters:
model=mistral: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 TotalAmount\nFRO
M \"invoices\" i\nWHERE i.InvoiceDate > '2010-01-01'\nGROUP BY i.InvoiceId\n\nThe following is information
about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n InvoiceId        int64\nTotalAmount
float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the
results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value
in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- j
ust the code."}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:31:51.99002513Z', 'message': {'role': 'assistant',
'content': ' Here\'s the Plotly code for plotting the DataFrame `df`:\n\n```python\nimport plotly.express a
s px\n\nfig = px.bar(df, x=\'InvoiceId\', y=\'TotalAmount\')\nfig.show()\n```\n\nFor a single value in the
dataframe:\n\n```python\nimport plotly.graph_objects as go\n\nfig = go.Indicator(\n    value = df.iloc[0]
[\'TotalAmount\'],\n    title = {\'text\': \'Total Amount\'},\n    mode = "gauge+number+delta"\n)\nfig.show
()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 36815150627, 'load_duration': 42752516,
'prompt_eval_count': 226, 'prompt_eval_duration': 10176457000, 'eval_count': 142, 'eval_duration': 26550951
000}
```


25
20
15
10
5
0
TotalAmount
100
150
200
250
300
350
400
InvoiceId

Out[26]:
```
('SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate > \'2010-01-01
\'\nGROUP BY i.InvoiceId',
      InvoiceId  TotalAmount
 0           84         1.98
 1           85         1.98
 2           86         3.96
 3           87         6.94
 4           88        17.91
 ..         ...          ...
 324        408         3.96
 325        409         5.94
 326        410         8.91
 327        411        13.86
 328        412         1.99

 [329 rows x 2 columns],
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'InvoiceId=%{x}<br>TotalAmount=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([ 84,  85,  86, ..., 410, 411, 412]),
               'xaxis': 'x',
               'y': array([ 1.98,  1.98,  3.96, ...,  8.91, 13.86,  1.99]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalAmount'}}}
 }))
```

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

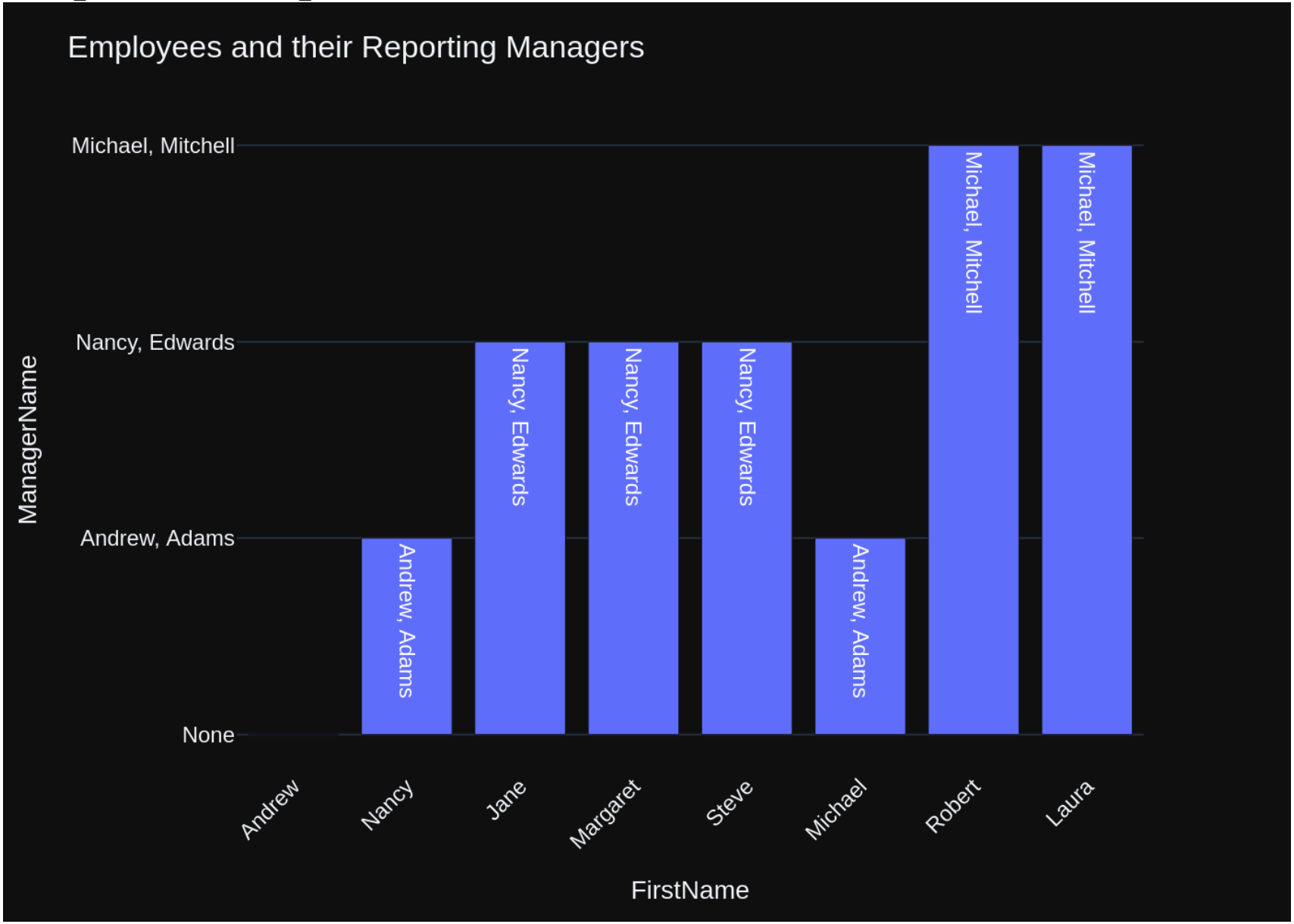
employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': 'SELECT e.Firs
tName, e.LastName, mt.FirstName AS ManagerFirstName, mt.LastName AS ManagerLastName\nFROM "employees" e\nLE
FT JOIN "employees" mt ON e.ReportsTo = mt.EmployeeId'}, {'role': 'user', 'content': " \n   List all empl
oyees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': 'SELECT e1.*, CONC
AT(e2.FirstName, \' \', e2.LastName) AS ManagerName\nFROM "employees" e1\nLEFT JOIN "employees" e2 ON e1.Re
portsTo = e2.EmployeeId\nORDER BY e1.EmployeeId'}, {'role': 'user', 'content': ' \n   Find the top 5 cus
tomers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, ca
lculation using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.
CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerI
d\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   List a
ll customers from Canada and their email addresses:\n'}, {'role': 'assistant', 'content': 'SELECT c.Custome
rId, c.Email, SUM(i.Total) AS TotalSpent\nFROM Customers c\nJOIN Invoices i ON c.CustomerId = i.CustomerId
\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   Find th
e top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoice
s table, calculation using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content':
'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId =
i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n-------------------------'}, {'rol
e': 'user', 'content': ' \n   Find the customer with the most invoices \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\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'use
r', '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 TotalCustom
ers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   List all customers from Canada and their email a
ddresses:\n'}, {'role': 'assistant', 'content': 'SELECT c.Email, c.Country\nFROM "customers" c\nWHERE c.Cou
ntry = \'Canada\''}, {'role': 'user', 'content': ' \n   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': '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   List all employe
es and their reporting manager's name (if any):\n"}]
Ollama parameters:
model=mistral: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   List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "content":
"SELECT e1.*, CONCAT(e2.FirstName, ' ', e2.LastName) AS ManagerName\nFROM \"employees\" e1\nLEFT JOIN \"emp
loyees\" e2 ON e1.ReportsTo = e2.EmployeeId\nORDER BY e1.EmployeeId"}, {"role": "user", "content": " \n

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   List all customers from Canada and their email addresses:\n"}, {"role": "assistant", "conte
nt": "SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent\nFROM Customers c\nJOIN Invoices i ON c.Cust
omerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 1"}, {"role": "user", "conten
t": " \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total ca
n be found on invoices table, calculation using invoice_items detail table is unnecessary \n"}, {"role": "a
ssistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoic
es\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n----------
---------------"}, {"role": "user", "content": " \n   Find the customer with the most invoices \n"}, {"r
ole": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\"
c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\n
LIMIT 1"}, {"role": "user", "content": "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.Cou
ntry\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n   List all customers fro
m 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   Find the customer with
the most invoices \n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(InvoiceId) AS NumberOfIn
voices \nFROM Invoices \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n------------------
-----"}, {"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   List all employees and their reporting manager's name (if any):\n"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:33:36.360366798Z', 'message': {'role': 'assistan
t', 'content': "SELECT e.FirstName, e.LastName, CASE WHEN e.ReportsTo IS NULL THEN 'None' ELSE (\n   SELEC
T e2.FirstName || ', ' || e2.LastName\n   FROM employees e2\n   WHERE e2.EmployeeId = e.ReportsTo\n) END
AS ManagerName\nFROM employees e;"}, 'done_reason': 'stop', 'done': True, 'total_duration': 104263194101,
'load_duration': 733873, 'prompt_eval_count': 1950, 'prompt_eval_duration': 88411791000, 'eval_count': 81,
'eval_duration': 15065826000}
```
SELECT e.FirstName, e.LastName, CASE WHEN e.ReportsTo IS NULL THEN 'None' ELSE (
    SELECT e2.FirstName || ', ' || e2.LastName
    FROM employees e2
    WHERE e2.EmployeeId = e.ReportsTo
) END AS ManagerName
FROM employees e;
```
Output from LLM: SELECT e.FirstName, e.LastName, CASE WHEN e.ReportsTo IS NULL THEN 'None' ELSE (
    SELECT e2.FirstName || ', ' || e2.LastName
    FROM employees e2
    WHERE e2.EmployeeId = e.ReportsTo

```
) END AS ManagerName
FROM employees e;
Extracted SQL: SELECT e.FirstName, e.LastName, CASE WHEN e.ReportsTo IS NULL THEN 'None' ELSE (
    SELECT e2.FirstName || ', ' || e2.LastName
    FROM employees e2
    WHERE e2.EmployeeId = e.ReportsTo
) END AS ManagerName
FROM employees e
SELECT e.FirstName, e.LastName, CASE WHEN e.ReportsTo IS NULL THEN 'None' ELSE (
    SELECT e2.FirstName || ', ' || e2.LastName
    FROM employees e2
    WHERE e2.EmployeeId = e.ReportsTo
) END AS ManagerName
FROM employees e
  FirstName  LastName        ManagerName
0    Andrew     Adams               None
1     Nancy   Edwards      Andrew, Adams
2      Jane   Peacock     Nancy, Edwards
3  Margaret      Park     Nancy, Edwards
4     Steve   Johnson     Nancy, Edwards
5   Michael  Mitchell      Andrew, Adams
6    Robert      King  Michael, Mitchell
7     Laura  Callahan  Michael, Mitchell
Ollama parameters:
model=mistral: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 e.FirstName, e.LastName, CASE WHEN e.Repo
rtsTo IS NULL THEN 'None' ELSE (\n    SELECT e2.FirstName || ', ' || e2.LastName\n    FROM employees e2\n
WHERE e2.EmployeeId = e.ReportsTo\n) END AS ManagerName\nFROM employees e\n\nThe following is information a
bout the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n FirstName      object\nLastName
object\nManagerName    object\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': 'mistral:latest', 'created_at': '2024-06-14T10:34:08.227426139Z', 'message': {'role': 'assistan
t', 'content': ' ```python\nimport plotly.express as px\n\nfig = px.bar(df, x=\'FirstName\', y=\'ManagerNam
e\', text_auto=True)\nfig.update_layout(title="Employees and their Reporting Managers")\nfig.update_xaxes(t
```

ickangle=-45)\nif df.shape[0] == 1:\n   fig = px.scatter(df, x=\'FirstName\', y=None, hover_data={\'Manage
rName\': df[\'ManagerName\'].iloc[0]})\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_dura
tion': 31841891174, 'load_duration': 653942, 'prompt_eval_count': 246, 'prompt_eval_duration': 9513597000,
'eval_count': 129, 'eval_duration': 22238359000}



Out[27]:
```
("SELECT e.FirstName, e.LastName, CASE WHEN e.ReportsTo IS NULL THEN 'None' ELSE (\n    SELECT e2.FirstNam
e || ', ' || e2.LastName\n    FROM employees e2\n    WHERE e2.EmployeeId = e.ReportsTo\n) END AS ManagerNa
me\nFROM employees e",
   FirstName  LastName        ManagerName
 0    Andrew     Adams               None
 1     Nancy   Edwards      Andrew, Adams
 2      Jane   Peacock     Nancy, Edwards
 3  Margaret      Park     Nancy, Edwards
 4     Steve   Johnson     Nancy, Edwards
 5   Michael  Mitchell      Andrew, Adams
 6    Robert      King  Michael, Mitchell
 7     Laura  Callahan  Michael, Mitchell,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'FirstName=%{x}<br>ManagerName=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'texttemplate': '%{y}',
               'type': 'bar',
               'x': array(['Andrew', 'Nancy', 'Jane', 'Margaret', 'Steve', 'Michael', 'Robert',
                           'Laura'], dtype=object),
               'xaxis': 'x',
               'y': array(['None', 'Andrew, Adams', 'Nancy, Edwards', 'Nancy, Edwards',
                           'Nancy, Edwards', 'Andrew, Adams', 'Michael, Mitchell',
                           'Michael, Mitchell'], dtype=object),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'title': {'text': 'Employees and their Reporting Managers'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'tickangle': -45, 'title': {'text': 'FirstN
ame'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'ManagerName'}}}
 }))
```

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 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\nORDER BY AverageInvoiceTotal DESC'}, {'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 s

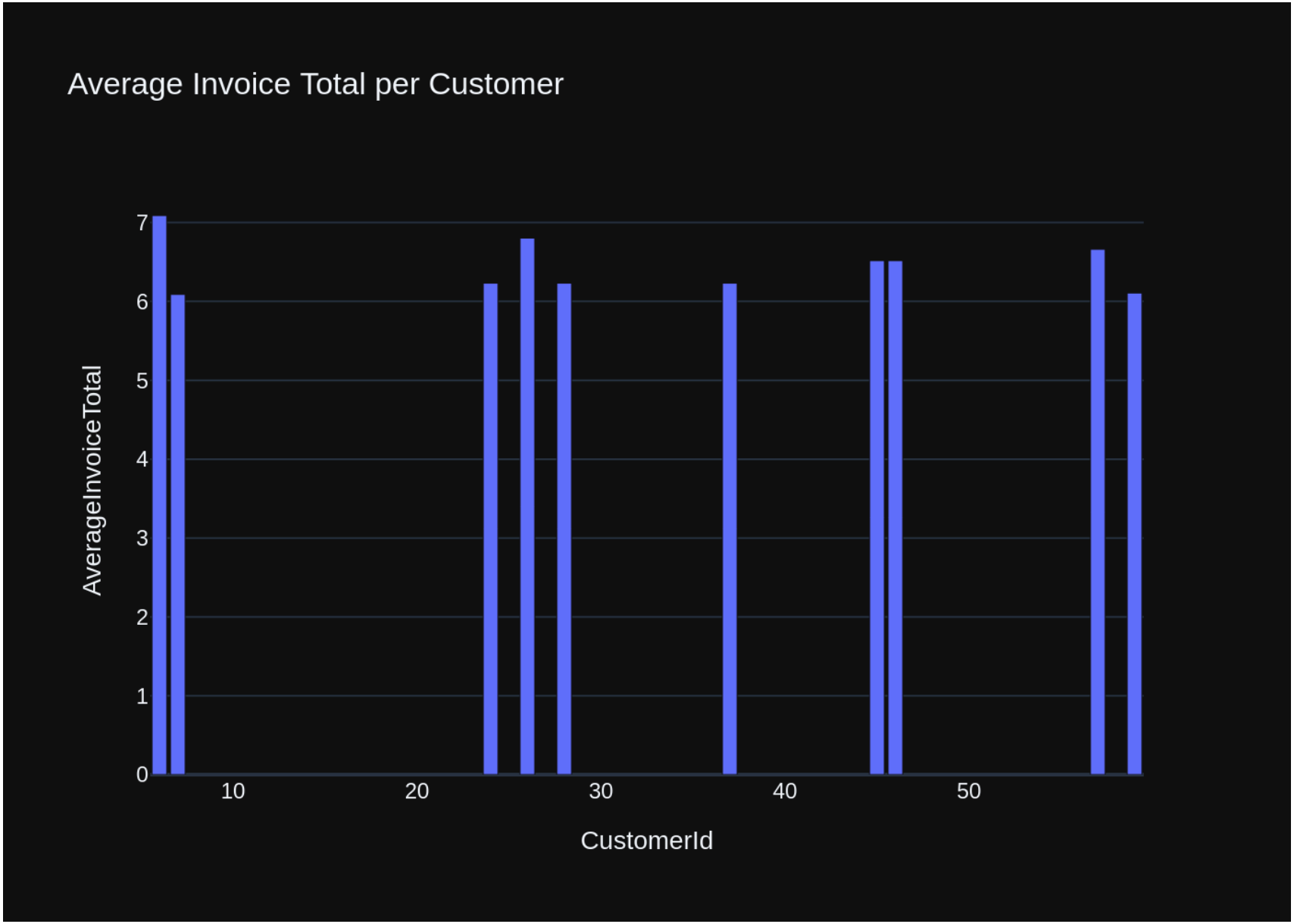
pent the most money overall, \n \n Hint: order total can be found on invoices table, calculation us
ing invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId,
SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY
c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n------------------------'}, {'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': 'user',
'content': ' \n Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'S
ELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM "customers" customer\nJOIN "invoice
s" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country'}, {'role': 'user', 'content': '
\n Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SE
LECT i.InvoiceId, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate > \'2010-01-01\'\nGRO
UP BY i.InvoiceId'}, {'role': 'user', 'content': ' \n Find the customer with the most invoices \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\nORDER BY TotalInvoices DESC
\nLIMIT 1'}, {'role': 'user', 'content': ' \n Find all invoices since 2010 and the total amount invoice
d:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM "invoice
s" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'role': 'user', 'content': ' \n
Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountr
y, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "invoices" i\nGROUP BY BillingCountry'}, {'role': 'user', 'con
tent': ' \n Get the average invoice total for each customer:\n'}]
Ollama parameters:
model=mistral: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 average invoice total for each customer:\n"}, {"r
ole": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\"
c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY AverageInvoiceTotal
DESC"}, {"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 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\n---------------
----------"}, {"role": "user", "content": " \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"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpen
t\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER B
Y TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find the total number of invoices per co
untry:\n"}, {"role": "assistant", "content": "SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceC
ount\nFROM \"customers\" customer\nJOIN \"invoices\" invoice ON customer.CustomerId = invoice.CustomerId\nG
ROUP BY Country"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amount i
nvoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\nFROM \"inv
oices\" i\nWHERE i.InvoiceDate > '2010-01-01'\nGROUP BY i.InvoiceId"}, {"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\nG
ROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n   Find all
invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceD
ate, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.Invo
iceDate"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"rol
e": "assistant", "content": "SELECT BillingCountry, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"invoices\"
i\nGROUP BY BillingCountry"}, {"role": "user", "content": " \n   Get the average invoice total for each c
ustomer:\n"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:35:41.909273042Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoice
s" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY AverageInvoiceTotal DESC\nLIMIT 10;<!-
- Added semicolon at the end -->'}, 'done_reason': 'stop', 'done': True, 'total_duration': 93568642005, 'lo
ad_duration': 670865, 'prompt_eval_count': 1933, 'prompt_eval_duration': 79126158000, 'eval_count': 77, 'ev
al_duration': 13816166000}
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
LIMIT 10;<!-- Added semicolon at the end -->
Output from LLM: SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY AverageInvoiceTotal DESC
LIMIT 10;<!-- Added semicolon at the end -->
Extracted SQL: SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY AverageInvoiceTotal DESC
LIMIT 10
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
LIMIT 10

	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
Ollama parameters:
model=mistral: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\nLIMIT 10\n\nThe following is information about the resulting pandas DataFrame 'df': \nR
unning 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 the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an I
ndicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:36:19.731297599Z', 'message': {'role': 'assistan
t', 'content': ' ```python\nimport plotly.express as px\n\nfig = px.bar(df, x="CustomerId", y="AverageInvoi
ceTotal", title="Average Invoice Total per Customer")\nfig.update_layout(showlegend=False)\n\nif len(df) ==
1:\n    fig = go.Indicator(\n        domain = {\'x\': [0, 1], \'y\': [0, 1]},\n        value = df.iloc[0]
[\'AverageInvoiceTotal\'],\n        title = {\'text\': "Total Average Invoice", \'font\': {\'size\': 2
4}},\n        mode = "gauge+number+delta"\n    )\n```'}, 'done_reason': 'stop', 'done': True, 'total_durati
on': 37796820169, 'load_duration': 612961, 'prompt_eval_count': 230, 'prompt_eval_duration': 9594532000, 'e
val_count': 157, 'eval_duration': 28113256000}
```


Average Invoice Total per Customer
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\nLIMIT 10',
    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,
 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]),
               'xaxis': 'x',
               'y': array([7.08857143, 6.80285714, 6.66      , 6.51714286, 6.51714286, 6.23142857,
                           6.23142857, 6.23142857, 6.10666667, 6.08857143]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'showlegend': False,
                'template': '...',
                'title': {'text': 'Average Invoice Total per Customer'},
                '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
```

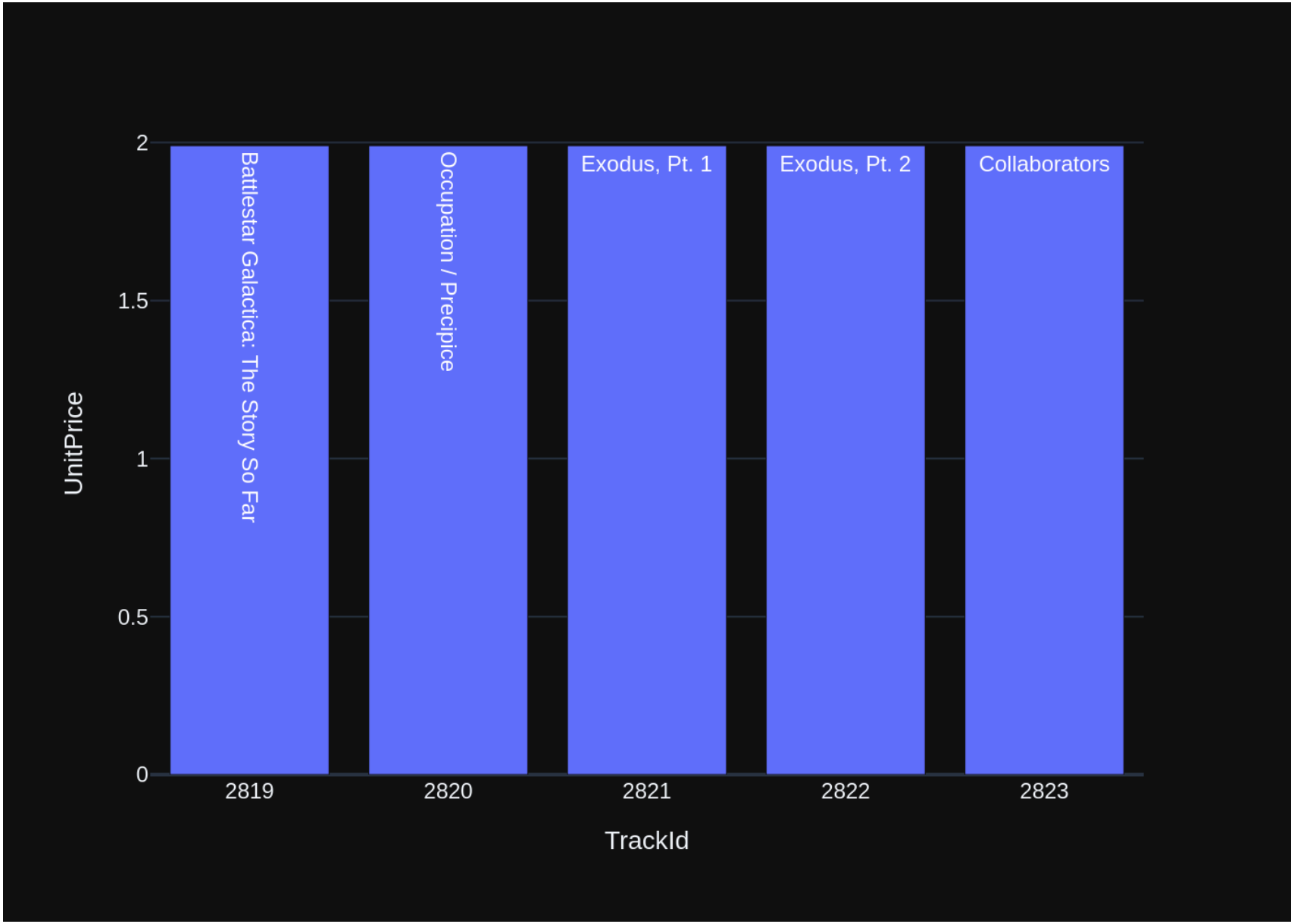
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY 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 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   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\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 "in
voice_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_i
tems, \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   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\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 tota
l quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.Invo
iceLineId) 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.Custome
rId\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 "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 1'}, {'role': 'user', 'content': ' \n   Find the top 5 most expensi
ve tracks (based on unit price):\n'}]
Ollama parameters:
model=mistral: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
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 Name, UnitPrice\nFROM \"tracks\"\nORDER B
Y UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find the top 5 most expensive tracks (bas
ed on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tra
cks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n   There are 3 tables: arti
sts, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by Al
bumId,\n   Can you find the top 10 most popular artists based on the number of tracks\n"}, {"role": "assis
tant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al O
N a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTra
cks 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\n-------------------------"}, {"role":
"user", "content": " \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 cust
omers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "conte
nt": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii
ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "use

```
r", "content": "  \n     Find the top 5 customers who spent the most money overall, \n     \n     Hint: ord
er total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n"},
{"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJO
IN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5
\n-------------------------"}, {"role": "user", "content": "  \n    Hint: album quantity is found in invoic
e_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 i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\n
FROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER
BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "  \n     Find the top 5 customers who spent th
e most money overall, \n     \n     Hint: order total can be found on invoices table, calculation using inv
oice_items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.T
otal) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.C
ustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"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.Customer
Id = 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 top 5 most expensive tracks (bas
ed on unit price):\n"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:37:52.507909631Z', 'message': {'role': 'assistan
t', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5
\n-------------------------'}, 'done_reason': 'stop', 'done': True, 'total_duration': 92679958139, 'load_du
ration': 741181, 'prompt_eval_count': 2031, 'prompt_eval_duration': 85191035000, 'eval_count': 38, 'eval_du
ration': 6804064000}
SELECT t.TrackId, t.Name, t.UnitPrice
FROM "tracks" t
ORDER BY t.UnitPrice DESC
LIMIT 5
-------------------------
SELECT t.TrackId, t.Name, t.UnitPrice
FROM "tracks" t
ORDER BY t.UnitPrice DESC
LIMIT 5
-------------------------
   TrackId                                  Name  UnitPrice
0     2819  Battlestar Galactica: The Story So Far       1.99
1     2820                 Occupation / Precipice       1.99
2     2821                         Exodus, Pt. 1       1.99
3     2822                         Exodus, Pt. 2       1.99
4     2823                         Collaborators       1.99
Ollama parameters:
model=mistral: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 t.TrackId, t.Name, t.UnitPrice\nFROM \"tracks
\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5\n------------------------\n\nThe following is information about t
he resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n TrackId        int64\nName          object
\nUnitPrice    float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly co
de 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 ex
planations -- just the code."}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:38:24.763602894Z', 'message': {'role': 'assistan
t', 'content': ' ```python\nimport plotly.express as px\n\nif df.shape[0] > 1:\n    fig = px.bar(df, x="Tra
ckId", y="UnitPrice", text="Name")\nelse:\n    fig = go.Indicator(\n        value=df.iloc[0][\'UnitPrice
\'],\n        title={\n            \'text\': df.iloc[0]["Name"],\n            \'font\': {\'size\': 18}\n
},\n        domain={\'x\': [0, 1]}\n    )\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_d
uration': 32230596297, 'load_duration': 48595513, 'prompt_eval_count': 207, 'prompt_eval_duration': 8785821
000, 'eval_count': 130, 'eval_duration': 23305055000}
```


2
1.5
1
0.5
0
UnitPrice
Battlestar Galactica: The Story So Far
Occupation / Precipice
Exodus, Pt. 1
Exodus, Pt. 2
Collaborators
2819
2820
2821
2822
2823
TrackId

Out[29]:
```
('SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5\n----------
----------',
    TrackId                                  Name  UnitPrice
 0     2819  Battlestar Galactica: The Story So Far       1.99
 1     2820                 Occupation / Precipice       1.99
 2     2821                          Exodus, Pt. 1       1.99
 3     2822                          Exodus, Pt. 2       1.99
 4     2823                          Collaborators       1.99,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'TrackId=%{x}<br>UnitPrice=%{y}<br>Name=%{text}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'text': array(['Battlestar Galactica: The Story So Far', 'Occupation / Precipice',
                              'Exodus, Pt. 1', 'Exodus, Pt. 2', 'Collaborators'], dtype=object),
               'textposition': 'auto',
               'type': 'bar',
               'x': array([2819, 2820, 2821, 2822, 2823]),
               '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': 'TrackId'}},
                '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\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   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 Artis
tName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT g2.G
enreId\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'}, {'rol
e': 'assistant', 'content': 'SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDura
tion\nFROM "playlist_track" pt\nJOIN "playlists" p ON pt.PlaylistId = p.PlaylistId\nJOIN "tracks" t ON pt.T
rackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10'}, {'role': 'user', 'con
tent': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'con
tent': 'SELECT Name, UnitPrice\nFROM "tracks"\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'conten
t': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'conten
t': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role':
'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assi
stant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMI
T 5\n-------------------------'}, {'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 c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFRO
M "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 genres and the number of tracks in each genre:\n'}]
Ollama parameters:
model=mistral: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

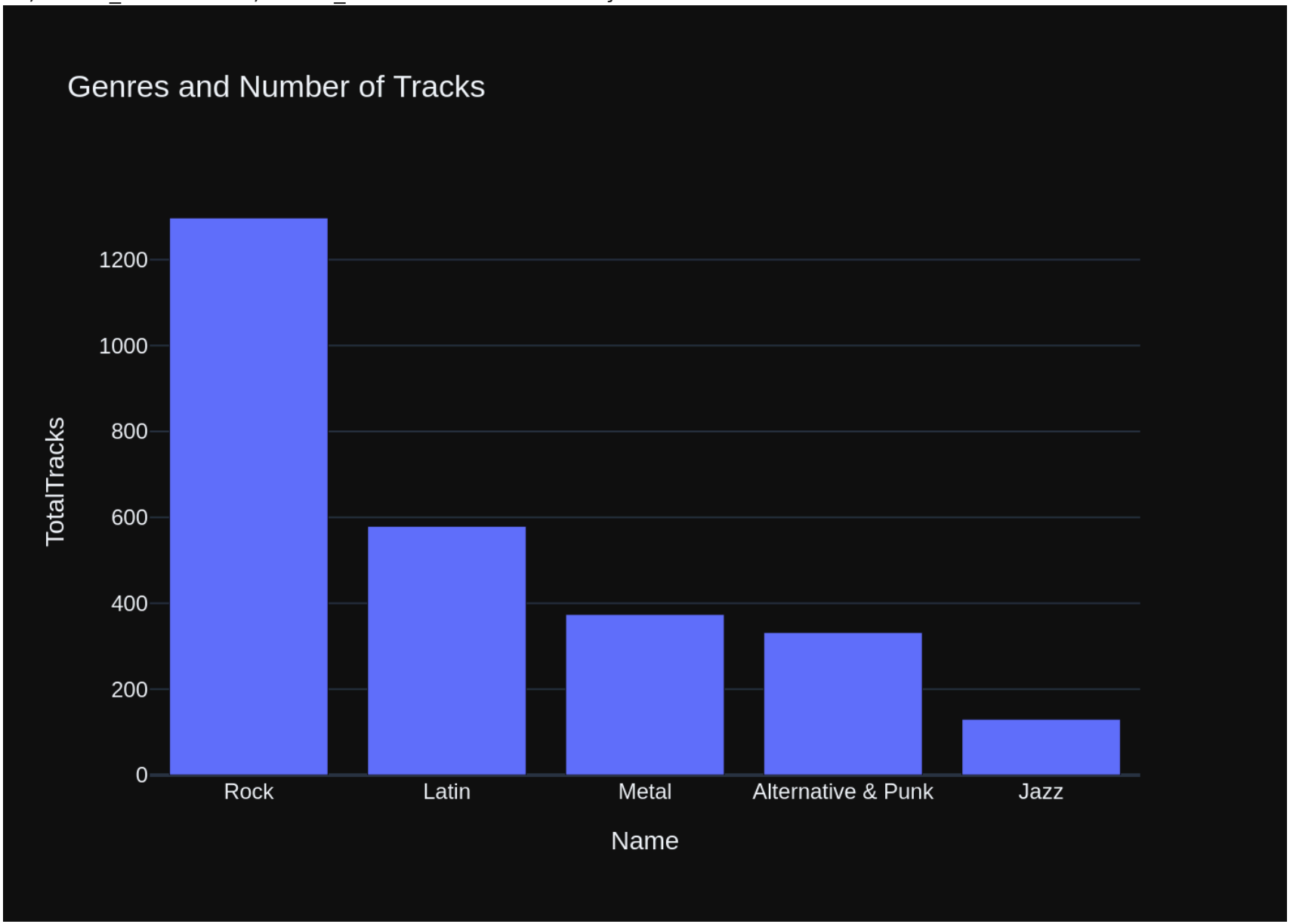
NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOI
NCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN
KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCR
EATE 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 sufficien
t but requires knowledge of a specific string in a particular column, please generate an intermediate SQL q
uery to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql
\n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the
most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer e
xactly as it was given before. \n"}, {"role": "user", "content": " \n   List all genres and the number of
tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS TotalTracks
\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks DESC
\nLIMIT 5"}, {"role": "user", "content": " \n   List all genres and the number of tracks in each genr
e:\n"}, {"role": "assistant", "content": "SELECT 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 track
s are linked by AlbumId,\n   Can you find the top 10 most popular artists based on the number of tracks
\n"}, {"role": "assistant", "content": "SELECT a.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.N
ame\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.Artist
Id, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.Artis
tId IN (\n SELECT g2.GenreId\n FROM \"genres\" g2\n 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"}, {"role": "user", "conte
nt": " \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 ORDER BY ArtistName"}, {"role": "user", "content": " \n   Get all playlists containing at l
east 10 tracks and the total duration of those tracks:\n"}, {"role": "assistant", "content": "SELECT pt.Pla
ylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM \"playlist_track\" pt\nJOIN \"p
laylists\" p ON pt.PlaylistId = p.PlaylistId\nJOIN \"tracks\" t ON pt.TrackId = t.TrackId\nGROUP BY pt.Play
listId, 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 ex
pensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.Un
itPrice\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 t.Trac
kId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5\n-------------------------
"}, {"role": "user", "content": " \n   Hint: album quantity is found in invoice_items, \n   \n   Find t
he top 5 customers 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 \"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 genres and the number of tracks in each genre:\n"}]

Insert of existing embedding ID: 5b0b32a6-7f1d-544f-8c5b-9448cbc635ac-sql
Add of existing embedding ID: 5b0b32a6-7f1d-544f-8c5b-9448cbc635ac-sql

```
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:39:57.814908671Z', 'message': {'role': 'assistan
t', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.Genre
Id = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks DESC\nLIMIT 5'}, 'done_reason': 'stop', 'done': True,
'total_duration': 92964135809, 'load_duration': 687316, 'prompt_eval_count': 1939, 'prompt_eval_duration':
80924181000, 'eval_count': 64, 'eval_duration': 11406013000}
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=mistral:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all genres and the number of tracks in each genr
e:\n'\n\nThe DataFrame was produced using this query: SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM
\"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks DESC\nLIMIT
5\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n N
ame           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': 'mistral:latest', 'created_at': '2024-06-14T10:40:33.473727323Z', 'message': {'role': 'assistan
t', 'content': ' Here\'s the Python code to create a bar chart using Plotly Express:\n\n```python\nimport p
lotly.express as px\n\nfig = px.bar(df, x="Name", y="TotalTracks", title=\'Genres and Number of Tracks\')\n
```

fig.show()\n```\n\nIf the dataframe \'df\' only contains one value:\n\n```python\nimport plotly.graph_objects as go\n\nfig = go.Indicator(\n   value=df[\'TotalTracks\'].iloc[0],\n   title={\'text\': df[\'Name\'].iloc[0]},\n   mode="number+delta",\n)\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 35633880455, 'load_duration': 41806016, 'prompt_eval_count': 226, 'prompt_eval_duration': 8816799000, 'eval_count': 154, 'eval_duration': 26684462000}



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},
                'template': '...',
                'title': {'text': 'Genres and Number of Tracks'},
                '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
```

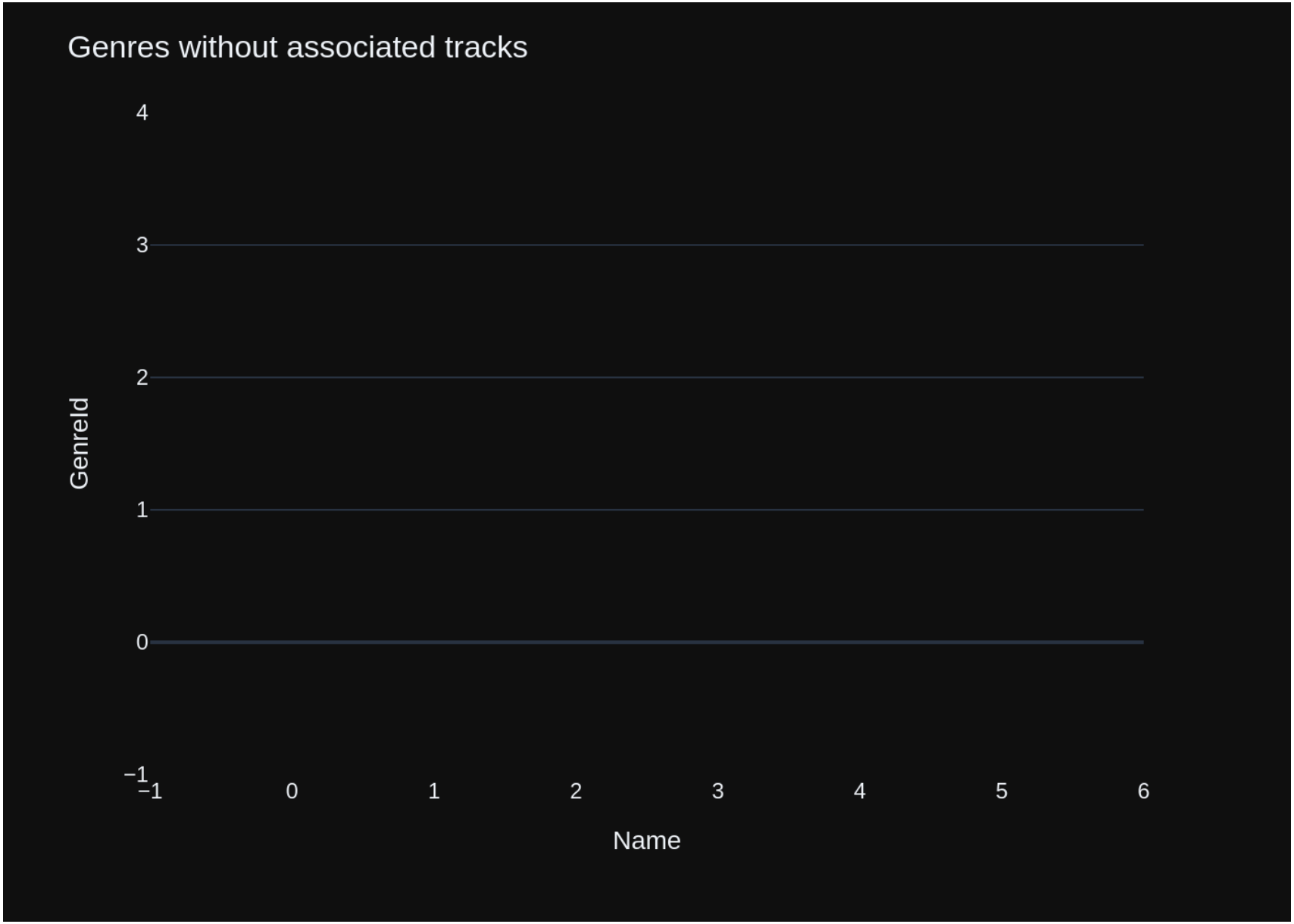
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE 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 * FROM tracks WHERE lower(Name) LIKE '%what%'"}, {'role': 'user', 'conten
t': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'c
ontent': 'SELECT * \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%\''}, {'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   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 genres that do not have any tracks associated wit
h them:\n'}]
Ollama parameters:
model=mistral: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 * FROM t
racks WHERE lower(Name) LIKE '%what%'"}, {"role": "user", "content": " \n   Find all tracks with a name c
ontaining \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * \nFROM \"tracks\" \nW
HERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": " \n   List all albums and their correspond
ing 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
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 genres that do not have any tracks associated with them:\n"}]
Ollama Response:

{'model': 'mistral:latest', 'created_at': '2024-06-14T10:42:35.283069288Z', 'message': {'role': 'assistan
t', 'content': 'SELECT g.GenreId, g.Name\nFROM "genres" g\nLEFT JOIN "tracks" t ON g.GenreId = t.GenreId\nW
HERE t.TrackId IS NULL\nGROUP BY g.GenreId, g.Name\nHAVING COUNT(t.TrackId) = 0;\nSELECT COUNT(*) FROM (SEL
ECT GenreId FROM genres) as temp_genres\nLEFT JOIN "tracks" t ON temp_genres.GenreId = t.GenreId\nWHERE t.T
rackId IS NULL GROUP BY temp_genres.GenreId HAVING COUNT(t.TrackId) = 0;\nBoth queries are equivalent and r
eturn the same result.\nSELECT g.GenreId, g.Name\nFROM "genres" g\nLEFT JOIN "tracks" t ON g.GenreId = t.Ge
nreId\nWHERE t.TrackId IS NULL GROUP BY g.GenreId HAVING COUNT(t.TrackId) = 0;\nThis query is more concise
and should be preferred.'}, 'done_reason': 'stop', 'done': True, 'total_duration': 121716024758, 'load_dura
tion': 709670, 'prompt_eval_count': 1850, 'prompt_eval_duration': 75602217000, 'eval_count': 253, 'eval_dur
ation': 45472297000}
SELECT g.GenreId, g.Name
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
GROUP BY g.GenreId, g.Name
HAVING COUNT(t.TrackId) = 0;
SELECT COUNT(*) FROM (SELECT GenreId FROM genres) as temp_genres
LEFT JOIN "tracks" t ON temp_genres.GenreId = t.GenreId
WHERE t.TrackId IS NULL GROUP BY temp_genres.GenreId HAVING COUNT(t.TrackId) = 0;
Both queries are equivalent and return the same result.
SELECT g.GenreId, g.Name
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL GROUP BY g.GenreId HAVING COUNT(t.TrackId) = 0;
This query is more concise and should be preferred.
Output from LLM: SELECT g.GenreId, g.Name
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
GROUP BY g.GenreId, g.Name
HAVING COUNT(t.TrackId) = 0;
SELECT COUNT(*) FROM (SELECT GenreId FROM genres) as temp_genres
LEFT JOIN "tracks" t ON temp_genres.GenreId = t.GenreId
WHERE t.TrackId IS NULL GROUP BY temp_genres.GenreId HAVING COUNT(t.TrackId) = 0;
Both queries are equivalent and return the same result.
SELECT g.GenreId, g.Name
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL GROUP BY g.GenreId HAVING COUNT(t.TrackId) = 0;
This query is more concise and should be preferred.
Extracted SQL: SELECT g.GenreId, g.Name
FROM "genres" g

```
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
GROUP BY g.GenreId, g.Name
HAVING COUNT(t.TrackId) = 0
SELECT g.GenreId, g.Name
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
GROUP BY g.GenreId, g.Name
HAVING COUNT(t.TrackId) = 0
Empty DataFrame
Columns: [GenreId, Name]
Index: []
Ollama parameters:
model=mistral: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.GenreId, g.Name\nFROM \"genres\" g\nLE
FT JOIN \"tracks\" t ON g.GenreId = t.GenreId\nWHERE t.TrackId IS NULL\nGROUP BY g.GenreId, g.Name\nHAVING
COUNT(t.TrackId) = 0\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning d
f.dtypes gives:\n GenreId    object\nName       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': 'mistral:latest', 'created_at': '2024-06-14T10:43:16.91046839Z', 'message': {'role': 'assistant',
'content': ' Here\'s some example Plotly code to create a bar chart for the given DataFrame `df`:\n\n```pyt
hon\nimport plotly.express as px\n\nfig = px.bar(df, x=\'Name\', y=\'GenreId\')\nfig.update_layout(title="G
enres without associated tracks")\nfig.show()\n```\n\nIn case there is only one value in the DataFrame, use
an Indicator instead:\n\n```python\nif len(df) == 1:\n    fig = px.indicator(df, title=\'No associated trac
ks\')\nelse:\n    fig = px.bar(df, x=\'Name\', y=\'GenreId\')\nfig.update_layout(title="Genres without asso
ciated tracks")\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 41624988024, 'lo
ad_duration': 44363850, 'prompt_eval_count': 236, 'prompt_eval_duration': 9806808000, 'eval_count': 177, 'e
val_duration': 31682966000}
```


Genres without associated tracks
GenreId
4
3
2
1
0
−1
−1
0
1
2
3
4
5
6
Name

Out[31]:
```
('SELECT g.GenreId, g.Name\nFROM "genres" g\nLEFT JOIN "tracks" t ON g.GenreId = t.GenreId\nWHERE t.TrackI
d IS NULL\nGROUP BY g.GenreId, g.Name\nHAVING COUNT(t.TrackId) = 0',
 Empty DataFrame
 Columns: [GenreId, Name]
 Index: [],
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'Name=%{x}<br>GenreId=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([], dtype=object),
               'xaxis': 'x',
               'y': array([], dtype=object),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'title': {'text': 'Genres without associated tracks'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'GenreId'}}}
 }))
```

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 i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoic
es" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums D
ESC\nLIMIT 1\n-------------------------'}, {'role': 'user', 'content': '  \n     Find the top 5 customers w
ho spent the most money overall, \n     \n     Hint: order total can be found on invoices table, calculatio
n using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.Customer
Id, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP
BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n-------------------------'}, {'role': 'user', '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 c.CustomerId, COUNT(ii.T
rackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice
_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'ro
le': 'user', 'content': '  \n    Hint: album quantity is found in invoice_items, \n    \n    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\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 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 customer with the most invoices \n'}, {'role': 'ass
istant', 'content': 'SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY Cus
tomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-------------------------'}, {'role': 'user', 'content':
'  \n     List all customers from Canada and their email addresses:\n'}, {'role': 'assistant', 'content':
'SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent\nFROM Customers c\nJOIN Invoices i ON c.CustomerI
d = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent 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 m
ost albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.Custome
rId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = i
i.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n
List all customers who have not placed any orders:\n'}]
Ollama parameters:
```

```
model=mistral: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 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 i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n-------------------------"}, {"role": "user", "content": " \n   Find the top 5 customers who spent the most money overall, \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\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, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n-------------------------"}, {"role": "user", "content": " \n   Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT 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   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   List all customers from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent\nFROM Customers c\nJOIN Invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 1"}, {"role": "user", "content": " \n   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.Cu
stomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    List all customers who
have not placed any orders:\n"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:44:53.147712582Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.CustomerId, c.FirstName, c.LastName\nFROM "customers" c\nLEFT JOIN "invoices" i ON
c.CustomerId = i.CustomerId\nWHERE i.InvoiceId IS NULL;\n------------------------'}, 'done_reason': 'sto
p', 'done': True, 'total_duration': 96148993120, 'load_duration': 1319152, 'prompt_eval_count': 2016, 'prom
pt_eval_duration': 84893497000, 'eval_count': 57, 'eval_duration': 10324947000}
SELECT c.CustomerId, c.FirstName, c.LastName
FROM "customers" c
LEFT JOIN "invoices" i ON c.CustomerId = i.CustomerId
WHERE i.InvoiceId IS NULL;
------------------------
Output from LLM: SELECT c.CustomerId, c.FirstName, c.LastName
FROM "customers" c
LEFT JOIN "invoices" i ON c.CustomerId = i.CustomerId
WHERE i.InvoiceId IS NULL;
------------------------
Extracted SQL: SELECT c.CustomerId, c.FirstName, c.LastName
FROM "customers" c
LEFT JOIN "invoices" i ON c.CustomerId = i.CustomerId
WHERE i.InvoiceId IS NULL
SELECT c.CustomerId, c.FirstName, c.LastName
FROM "customers" c
LEFT JOIN "invoices" i ON c.CustomerId = i.CustomerId
WHERE i.InvoiceId IS NULL
Empty DataFrame
Columns: [CustomerId, FirstName, LastName]
Index: []
Ollama parameters:
model=mistral:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all customers who have not placed any orders:\n'\n
\nThe DataFrame was produced using this query: SELECT c.CustomerId, c.FirstName, c.LastName\nFROM \"custome
rs\" c\nLEFT JOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nWHERE i.InvoiceId IS NULL\n\nThe following
is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId    object
\nFirstName     object\nLastName      object\ndtype: object"}, {"role": "user", "content": "Can you generat
```

e 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': 'mistral:latest', 'created_at': '2024-06-14T10:45:38.532354562Z', 'message': {'role': 'assistant', 'content': ' ```python\nimport plotly.express as px\n\nfig = px.bar("CustomerId", "FirstName", "LastName", data=df)\nfig.update_layout(title="Customers without Orders")\nfig.write_image("no_orders.png")\n```\n\nIf there is only one value in the dataframe:\n\n```python\nimport plotly.graph_objects as go\n\nfig = go.Figure(data=go.Indicator(\n   mode="gauge",\n   value=df.shape[0],\n   domain={"x": [0, 1], "y": [0, 1]},\n   title={"text": "Customers without Orders", "font": {"size": 24}},\n))\nfig.write_image("no_orders.png")\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 45382366006, 'load_duration': 995283, 'prompt_eval_count': 215, 'prompt_eval_duration': 9416432000, 'eval_count': 189, 'eval_duration': 35833220000}

Out[32]:
```
('SELECT c.CustomerId, c.FirstName, c.LastName\nFROM "customers" c\nLEFT JOIN "invoices" i ON c.CustomerId
= i.CustomerId\nWHERE i.InvoiceId IS NULL',
 Empty DataFrame
 Columns: [CustomerId, FirstName, LastName]
 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

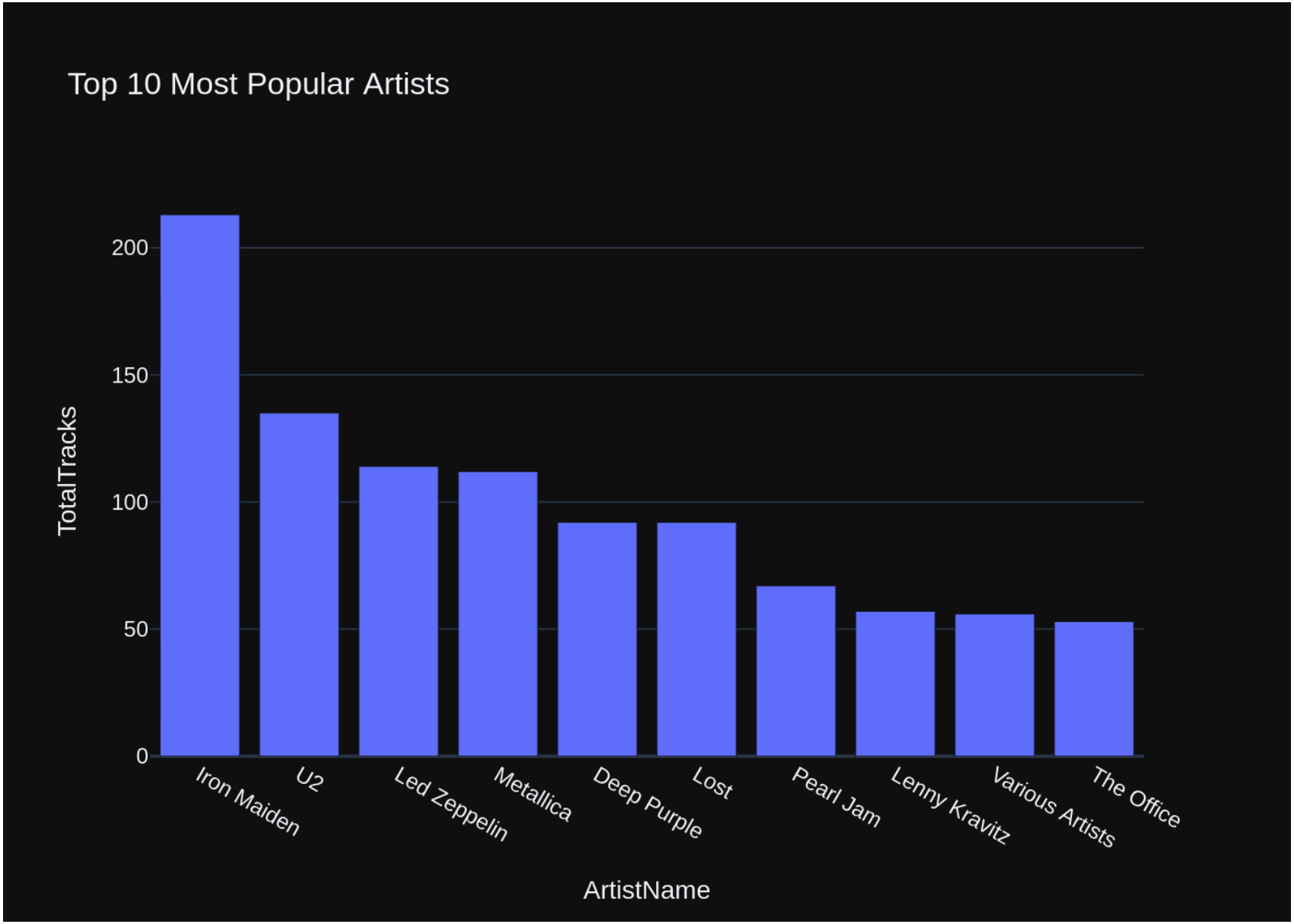
```
Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role':
'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "i
nvoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY
c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Hint: album quanti
ty is found in invoice_items, \n    \n    Find the top 5 customers who bought the most albums in total quan
tity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) A
S TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Custome
rId\nORDER BY TotalAlbums DESC\nLIMIT 5\n-------------------------'}, {'role': 'user', 'content': '  \n
Hint: album quantity is found in invoice_items, \n    \n    Find the top 5 customers who bought the most al
bums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, C
OUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\n
GROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'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': 'us
er', 'content': '  \n    Hint: album quantity is found in invoice_items, \n    \n    Find the top 5 custome
rs who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'conten
t': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" i
i ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'use
r', '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, CO
UNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.Nam
e'}, {'role': 'user', 'content': ' \n    There are 3 tables: artists, albums and tracks, where albums and a
rtists are linked by ArtistId, albums and tracks are linked by AlbumId,\n    Can you find the top 10 most p
opular artists based on the number of tracks\n'}]
Ollama parameters:
model=mistral: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 \"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 \"artists\"\r\n(\r\n   ArtistId INTEGER PRIMAR
Y KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums
\" (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 NUL
L,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) R
EFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (T
rackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDE
X IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_track\"
(TrackId)\n\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelin
es \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations f
or the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific str
ing in a particular column, please generate an intermediate SQL query to find the distinct strings in that
column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficie
nt, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the quest
ion has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"ro
le": "user", "content": " \n   There are 3 tables: artists, albums and tracks, where albums and artists ar
e linked by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most popular ar
tists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId)
AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON a
l.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": "ass
istant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.A
rtistId = 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.ArtistI
d, a.Name ORDER BY ArtistName"}, {"role": "user", "content": " \n   Identify artists who have albums wit
h 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.Gen
reId\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 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.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 invoi
ce_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all inv
oices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM

\"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY To
talAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n   List all albums and their corresponding art
ist 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 i
n total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(i
i.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.Invoice
Id\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n   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   List all genres and the numb
er of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS Total
Tracks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"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"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:47:28.451336848Z', 'message': {'role': 'assistan
t', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a
\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Artis
tId, a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, 'done_reason': 'stop', 'done': True, 'total_duration':
109840887765, 'load_duration': 875821, 'prompt_eval_count': 1986, 'prompt_eval_duration': 90448067000, 'eva
l_count': 99, 'eval_duration': 18707716000}
```
SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks
FROM "artists" a
JOIN "albums" al ON a.ArtistId = al.ArtistId
JOIN "tracks" t ON al.AlbumId = t.AlbumId
GROUP BY a.ArtistId, a.Name
ORDER BY TotalTracks DESC
LIMIT 10
```
```
SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks
FROM "artists" a
JOIN "albums" al ON a.ArtistId = al.ArtistId
JOIN "tracks" t ON al.AlbumId = t.AlbumId
GROUP BY a.ArtistId, a.Name
ORDER BY TotalTracks DESC
LIMIT 10
```

	ArtistId	ArtistName	TotalTracks
0	90	Iron Maiden	213
1	150	U2	135
2	22	Led Zeppelin	114
3	50	Metallica	112

```
4        58      Deep Purple           92
5       149             Lost           92
6       118        Pearl Jam           67
7       100    Lenny Kravitz           57
8        21  Various Artists           56
9       156       The Office           53
Ollama parameters:
model=mistral:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: ' \n    There are 3 tables: artists, albums and tracks, where alb
ums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n    Can you find the top
10 most popular artists based on the number of tracks\n'\n\nThe DataFrame was produced using this query: SE
LECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\"
al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nO
RDER BY TotalTracks DESC\nLIMIT 10\n\nThe following is information about the resulting pandas DataFrame 'd
f': \nRunning df.dtypes gives:\n ArtistId        int64\nArtistName     object\nTotalTracks     int64\ndtyp
e: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of t
he dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataf
rame, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the cod
e."}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:48:20.103148549Z', 'message': {'role': 'assistan
t', 'content': " Here's the Python code to create a bar chart using Plotly Express:\n\n```python\nimport pl
otly.express as px\n\nfig = px.bar(df, x='ArtistName', y='TotalTracks', title='Top 10 Most Popular Artist
s')\nfig.show()\n```\n\nIn case there is only one value in the dataframe, use an Indicator:\n\n```python\ni
mport plotly.graph_objects as go\n\nif df.shape[0] == 1:\n    fig = go.Indicator(\n        value=df.iloc[0]
['TotalTracks'],\n        domain={'x': [0, 1], 'y': [0, 1]}\n    )\nelse:\n    fig = px.bar(df, x='ArtistNa
me', y='TotalTracks', title='Top 10 Most Popular Artists')\nfig.show()\n```"}, 'done_reason': 'stop', 'don
e': True, 'total_duration': 51624259926, 'load_duration': 42692311, 'prompt_eval_count': 307, 'prompt_eval_
duration': 13667685000, 'eval_count': 209, 'eval_duration': 37862685000}
```


Top 10 Most Popular Artists
TotalTracks
200
150
100
50
0
Iron Maiden
U2
Led Zeppelin
Metallica
Deep Purple
Lost
Pearl Jam
Lenny Kravitz
Various Artists
The Office
ArtistName

Out[33]:
```
('SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "album
s" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name
\nORDER BY TotalTracks DESC\nLIMIT 10',
    ArtistId       ArtistName  TotalTracks
 0        90      Iron Maiden          213
 1       150               U2          135
 2        22     Led Zeppelin          114
 3        50        Metallica          112
 4        58      Deep Purple           92
 5       149             Lost           92
 6       118        Pearl Jam           67
 7       100    Lenny Kravitz           57
 8        21  Various Artists           56
 9       156       The Office           53,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'ArtistName=%{x}<br>TotalTracks=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Iron Maiden', 'U2', 'Led Zeppelin', 'Metallica', 'Deep Purple', 'Lost',
                           'Pearl Jam', 'Lenny Kravitz', 'Various Artists', 'The Office'],
                          dtype=object),
               'xaxis': 'x',
               'y': array([213, 135, 114, 112,  92,  92,  67,  57,  56,  53]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Top 10 Most Popular Artists'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'ArtistName'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalTracks'}}}
 }))
```

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': ' \n   List all customers from Canada and their email addresses:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.Email, SUM(i.Tota

```
l) AS TotalSpent\nFROM Customers c\nJOIN Invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\n
ORDER BY TotalSpent DESC\nLIMIT 1'}, {'role': 'user', 'content': 'what are the top 5 countries that custome
rs come from?'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM "cust
omers" 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 cus
tomer 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    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 Countr
y'}, {'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 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 "invoice
s" i ON c.CustomerId = i.CustomerId\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.Cust
omerId\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 customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assis
tant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_
items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n-------
-----------------'}, {'role': 'user', 'content': '  \n     List all customers from Canada and their email
addresses:\n'}]
Ollama parameters:
model=mistral: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)
```

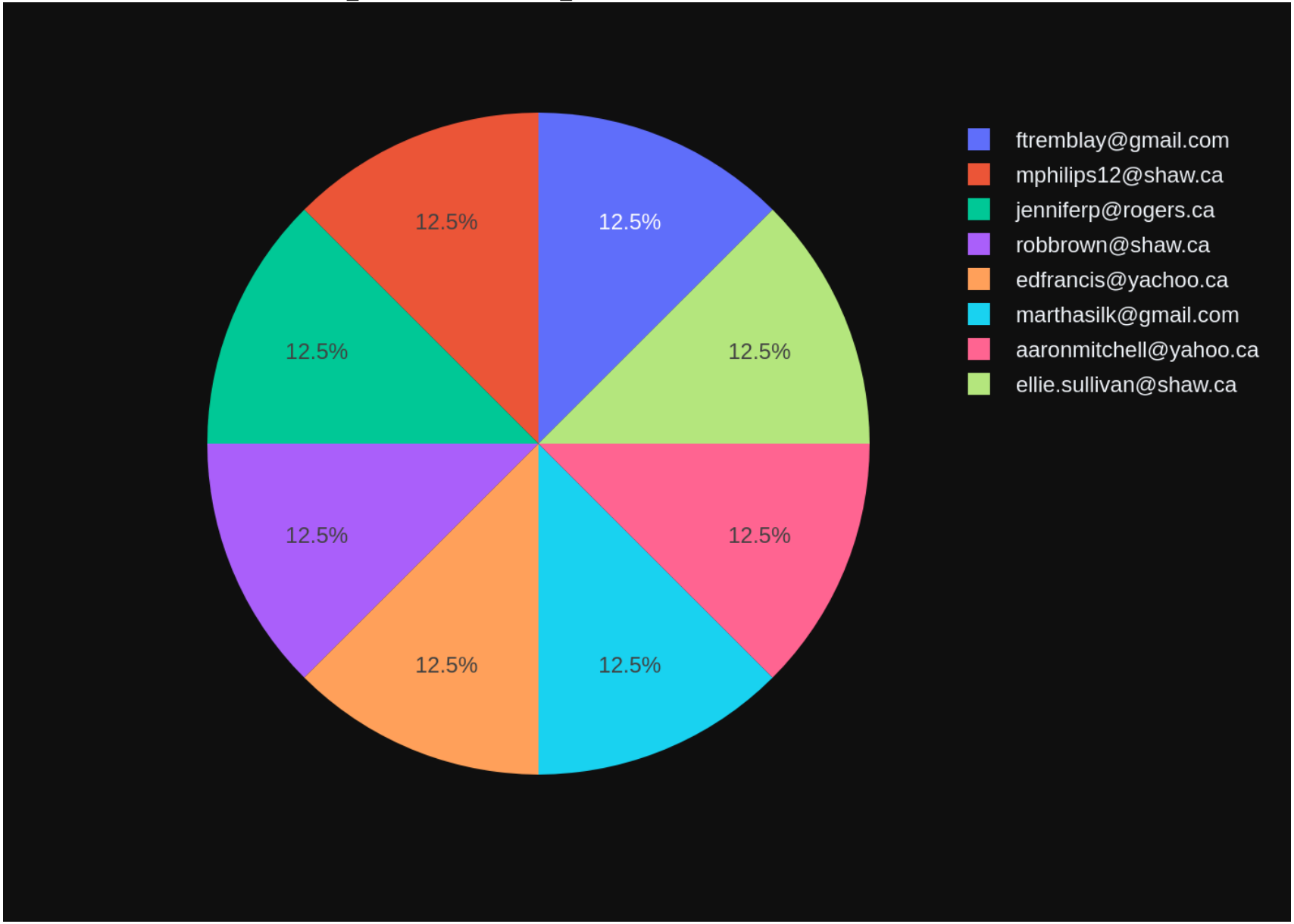
NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE TABLE
\"employees\"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   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": " \n
List all customers from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SELECT c.C
ustomerId, c.Email, SUM(i.Total) AS TotalSpent\nFROM Customers c\nJOIN Invoices i ON c.CustomerId = i.Custo
merId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 1"}, {"role": "user", "content": "what are th
e 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"}, {"rol
e": "user", "content": " \n   Find the customer with the most invoices \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\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "us
er", "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 Nu
mberOfInvoices DESC\nLIMIT 1\n-------------------------"}, {"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.Custo

merId = invoice.CustomerId\nGROUP BY Country"}, {"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.Cust
omerId"}, {"role": "user", "content": " \n   Find the customer who bought the most albums in total quant
ity (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 1"}, {"rol
e": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) F
ROM \"customers\""}, {"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(i
i.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGR
OUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n-------------------------"}, {"role": "user", "con
tent": " \n   List all customers from Canada and their email addresses:\n"}]

Insert of existing embedding ID: 584873f8-1904-50f1-8f80-7ccf08059264-sql
Add of existing embedding ID: 584873f8-1904-50f1-8f80-7ccf08059264-sql

```
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:49:50.501798484Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.Email, c.Country\nFROM "customers" c\nWHERE c.Country = \'Canada\';'}, 'done_reaso
n': 'stop', 'done': True, 'total_duration': 90298836967, 'load_duration': 838721, 'prompt_eval_count': 200
2, 'prompt_eval_duration': 84931851000, 'eval_count': 27, 'eval_duration': 4780482000}
SELECT c.Email, c.Country
FROM "customers" c
WHERE c.Country = 'Canada';
Output from LLM: SELECT c.Email, c.Country
FROM "customers" c
WHERE c.Country = 'Canada';
Extracted SQL: SELECT c.Email, c.Country
FROM "customers" c
WHERE c.Country = 'Canada'
SELECT c.Email, c.Country
FROM "customers" c
WHERE c.Country = 'Canada'
                    Email Country
0     ftremblay@gmail.com  Canada
1      mphilips12@shaw.ca  Canada
2     jenniferp@rogers.ca  Canada
3        robbrown@shaw.ca  Canada
4     edfrancis@yachoo.ca  Canada
5    marthasilk@gmail.com  Canada
6  aaronmitchell@yahoo.ca  Canada
7  ellie.sullivan@shaw.ca  Canada
Ollama parameters:
model=mistral: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.Email, c.Country\nFROM \"customers\" c\nWHER
E c.Country = 'Canada'\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning
df.dtypes gives:\n Email      object\nCountry    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 d
ataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Pyth
on code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:50:14.776488365Z', 'message': {'role': 'assistan
t', 'content': ' ```python\nimport plotly.express as px\n\nfig = px.bar("Country", "Email", data=df)\nfig.u
```

pdate_layout(title="Customers from Canada and their Email Addresses")\nif len(df) == 1:\n   fig = px.scatter(x=\'Country\', y=\'Email\', data=df, marker=dict(symbol=\'star\'))\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 24247590274, 'load_duration': 44645878, 'prompt_eval_count': 182, 'prompt_eval_duration': 7932752000, 'eval_count': 89, 'eval_duration': 16220417000}



Out[34]:
```
('SELECT c.Email, c.Country\nFROM "customers" c\nWHERE c.Country = \'Canada\'',
                     Email Country
0     ftremblay@gmail.com  Canada
1      mphilips12@shaw.ca  Canada
2     jenniferp@rogers.ca  Canada
3        robbrown@shaw.ca  Canada
4     edfrancis@yachoo.ca  Canada
5    marthasilk@gmail.com  Canada
6  aaronmitchell@yahoo.ca  Canada
7  ellie.sullivan@shaw.ca  Canada,
Figure({
    'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
              'hovertemplate': 'Email=%{label}<extra></extra>',
              'labels': array(['ftremblay@gmail.com', 'mphilips12@shaw.ca', 'jenniferp@rogers.ca',
                               'robbrown@shaw.ca', 'edfrancis@yachoo.ca', 'marthasilk@gmail.com',
                               'aaronmitchell@yahoo.ca', 'ellie.sullivan@shaw.ca'], dtype=object),
              'legendgroup': '',
              'name': '',
              'showlegend': True,
              'type': 'pie'}],
    'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'template': '...'}
}))
```

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

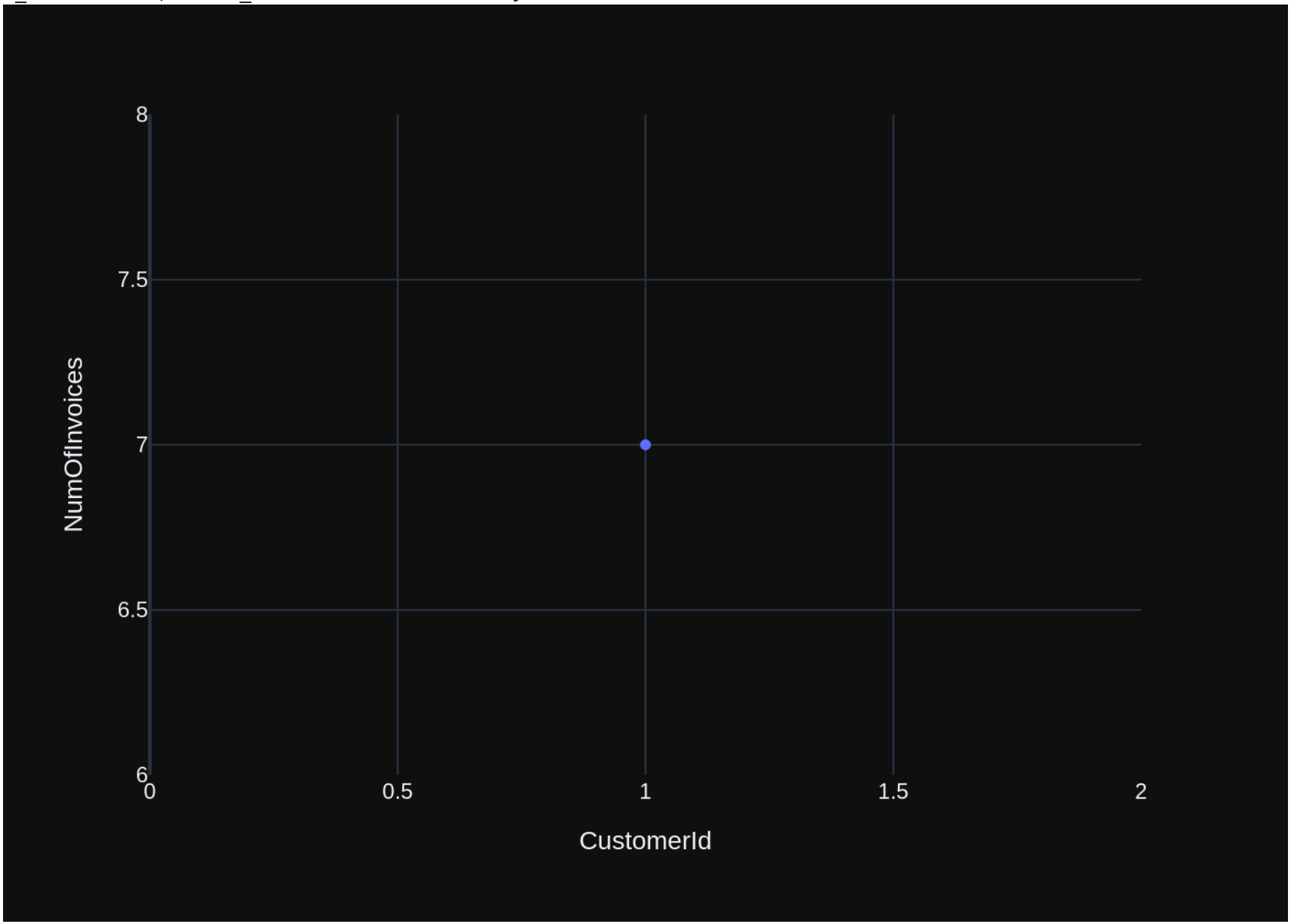
DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   Find the customer with the most invoices \n'}, {'rol
e': 'assistant', 'content': 'SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROU
P BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-----------------------'}, {'role': 'user', 'co
ntent': ' \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order tota
l can be found on invoices table, calculation using invoice_items detail table is unnecessary \n'}, {'rol
e': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "inv
oices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n--------
----------------'}, {'role': 'user', 'content': ' \n   Find the top 5 customers who spent the most mone
y 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 To
talSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDE
R BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the customer who bought the mos
t albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.Customer
Id, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId
\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\n
LIMIT 1'}, {'role': 'user', 'content': ' \n   Find the customer who bought the most albums in total quan
tity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId)
AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Custom
erId\nORDER BY TotalAlbums DESC\nLIMIT 1\n-------------------------'}, {'role': 'user', 'content': ' \n
Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most 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\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': '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\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 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   Find the customer with the most invoices \n'}]
Ollama parameters:
model=mistral: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 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   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\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 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 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   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"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:51:48.848167537Z', 'message': {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(InvoiceId) as NumOfInvoices\nFROM "invoices"\nGROUP BY CustomerId\nORDER BY NumOfInvoices DESC\nLIMIT 1;'}, 'done_reason': 'stop', 'done': True, 'total_duration': 93962804971, 'load_duration': 766747, 'prompt_eval_count': 1942, 'prompt_eval_duration': 84850661000, 'eval_count': 45, 'eval_duration': 8229349000}

```
SELECT CustomerId, COUNT(InvoiceId) as NumOfInvoices
FROM "invoices"
GROUP BY CustomerId
ORDER BY NumOfInvoices DESC
LIMIT 1;
Output from LLM: SELECT CustomerId, COUNT(InvoiceId) as NumOfInvoices
FROM "invoices"
GROUP BY CustomerId
ORDER BY NumOfInvoices DESC
LIMIT 1;
Extracted SQL: SELECT CustomerId, COUNT(InvoiceId) as NumOfInvoices
FROM "invoices"
GROUP BY CustomerId
ORDER BY NumOfInvoices DESC
LIMIT 1
SELECT CustomerId, COUNT(InvoiceId) as NumOfInvoices
FROM "invoices"
GROUP BY CustomerId
ORDER BY NumOfInvoices DESC
LIMIT 1
   CustomerId  NumOfInvoices
0           1              7
Ollama parameters:
model=mistral:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n     Find the customer with the most invoices \n'\n\nThe Dat
aFrame was produced using this query: SELECT CustomerId, COUNT(InvoiceId) as NumOfInvoices\nFROM \"invoices
\"\nGROUP BY CustomerId\nORDER BY NumOfInvoices DESC\nLIMIT 1\n\nThe following is information about the res
ulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId       int64\nNumOfInvoices    int64\n
dtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results
of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the d
ataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the
code."}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:52:21.141039004Z', 'message': {'role': 'assistan
t', 'content': ' ```python\nimport plotly.express as px\n\nfig = px.bar("CustomerId", "NumOfInvoices", data
=df)\nfig.update_layout(title="Customers with Most Invoices")\nif len(df) == 1:\n    fig = px.scatter(x=[df
["CustomerId"].iloc[0]], y=[df["NumOfInvoices"].iloc[0]], mode=\'markers+lines\', marker=dict(color=\'blue
\'), name="Indicated Customer")\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration':
```

32268875400, 'load_duration': 41388255, 'prompt_eval_count': 208, 'prompt_eval_duration': 9138845000, 'eval_count': 125, 'eval_duration': 23038774000}



Out[35]:
```
('SELECT CustomerId, COUNT(InvoiceId) as NumOfInvoices\nFROM "invoices"\nGROUP BY CustomerId\nORDER BY Num
OfInvoices DESC\nLIMIT 1',
    CustomerId  NumOfInvoices
 0           1             7,
 Figure({
     'data': [{'hovertemplate': 'CustomerId=%{x}<br>NumOfInvoices=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'symbol': 'circle'},
               'mode': 'markers',
               'name': '',
               'orientation': 'v',
               'showlegend': False,
               'type': 'scatter',
               'x': array([1]),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'}],
     'layout': {'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': 'NumOfInvoices'}}}
 }))
```

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   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   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 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   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\nJO
IN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMI
T 5\n-------------------------'}, {'role': 'user', 'content': ' \n   Find the top 5 customers who spent
the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using i
nvoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM
(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.C
ustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n-------------------------'}, {'role': 'user', 'content': '
\n   Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerI
d, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerI
d\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   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.CustomerI
d = 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', 'content': ' \n   Find the customer who bought
the most albums in total quantity (across all invoices): \n'}]
Ollama parameters:
model=mistral: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   Find the customer who bought the most albums in total quan
tity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId)
AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Cu
stomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n-------------------------"}, {"role": "user", "content": "
\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.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId
= ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "
\n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the m
ost albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.Custome

rId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.I
nvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n
Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most al
bums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, C
OUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.Invoice
Id\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n-------------------------"}, {"role": "use
r", "content": " \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: ord
er total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n"},
{"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJO
IN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5
\n-------------------------"}, {"role": "user", "content": " \n   Find the customer with the most invoic
es \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM
\"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalI
nvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n   Find the top 5 customers who spent the most
money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_it
ems detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) A
S TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Customer
Id\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find the customer with the m
ost invoices \n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoice
s \nFROM Invoices \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-------------------------
"}, {"role": "user", "content": " \n   Find the customer who bought the most albums in total quantity (a
cross all invoices): \n"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:54:01.202224356Z', 'message': {'role': 'assistan
t', 'content': 'SELECT CustomerId, SUM(Quantity) as TotalAlbums\nFROM Invoice_Items\nGROUP BY CustomerId\nO
RDER BY TotalAlbums DESC\nLIMIT 1;\n-------------------------'}, 'done_reason': 'stop', 'done': True, 'tota
l_duration': 99931304657, 'load_duration': 822343, 'prompt_eval_count': 2030, 'prompt_eval_duration': 91132
511000, 'eval_count': 44, 'eval_duration': 8122667000}
```
SELECT CustomerId, SUM(Quantity) as TotalAlbums
FROM Invoice_Items
GROUP BY CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1;
-------------------------
Output from LLM: SELECT CustomerId, SUM(Quantity) as TotalAlbums
FROM Invoice_Items
GROUP BY CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1;
-------------------------
Extracted SQL: SELECT CustomerId, SUM(Quantity) as TotalAlbums
FROM Invoice_Items
```

```
GROUP BY CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1
SELECT CustomerId, SUM(Quantity) as TotalAlbums
FROM Invoice_Items
GROUP BY CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1
Couldn't run sql:  Execution failed on sql 'SELECT CustomerId, SUM(Quantity) as TotalAlbums
FROM Invoice_Items
GROUP BY CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1': no such column: CustomerId
```

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\n----------
-----'}, {'role': 'user', 'content': ' \n   Hint: album quantity is found in invoice_items, \n   \n   F
ind the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role':
'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "in
voice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'},
{'role': 'user', 'content': ' \n   Find the customer who bought the most albums in total quantity (acros
s all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlb
ums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice_items" ii ON i.Inv
oiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'conte
nt': ' \n   Find the customer who bought the most albums in total quantity (across all invoices): \n'},
{'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i
\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\n
LIMIT 1\n-------------------------'}, {'role': 'user', 'content': ' \n   Find the top 5 customers who sp
ent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation usi
ng invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, S
UM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY
c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n-------------------------'}, {'role': 'user', 'content': '
\n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be fo
und on invoices table, calculation using invoice_items detail table is unnecessary \n'}, {'role': 'assistan
t', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON
c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user',
'content': ' \n   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.AlbumI
d\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n   There are 3
tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are
linked by AlbumId,\n   Can you find the top 10 most popular artists based on the number of tracks\n'}, {'r
ole': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFR
OM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGR
OUP BY a.ArtistId, a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'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'}]
Ollama parameters:
model=mistral: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

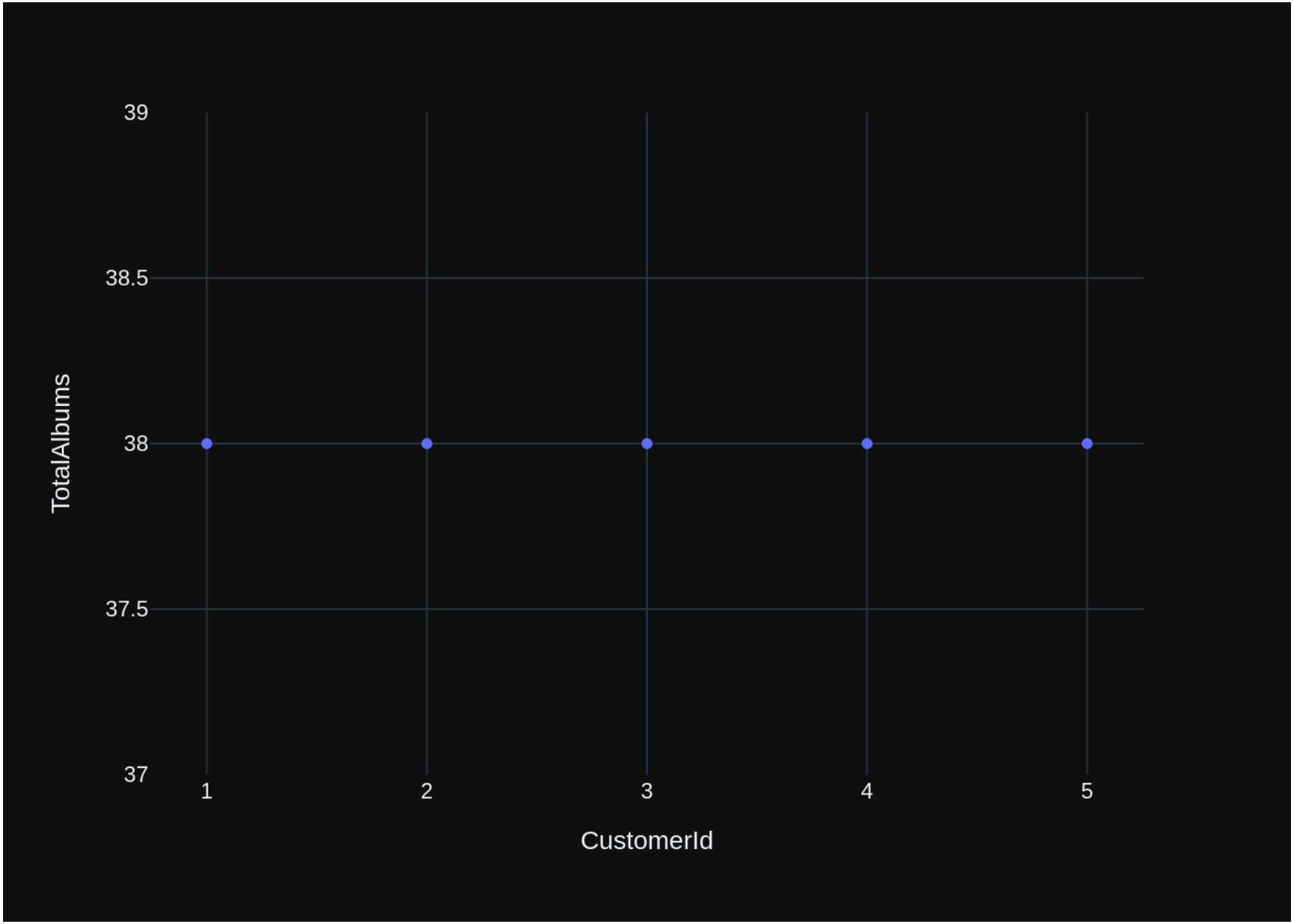
EY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   Uni
tPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCE
S \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (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\n-------------------------"}, {"role": "user", "content": " \n   Hint: album quantity is found in

invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across al
l invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\n
FROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER
BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find the customer who bought the most
albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerI
d, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Customer
Id\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DE
SC\nLIMIT 1"}, {"role": "user", "content": " \n   Find the customer who bought the most albums in total
quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.Track
Id) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY
i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\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": "assist
ant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\"
i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n---------------
----------"}, {"role": "user", "content": " \n   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"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpen
t\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER B
Y TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n   There are 3 tables: artists, albums and t
racks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n   Can y
ou 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 1
0"}, {"role": "user", "content": " \n   There are 3 tables: artists, albums and tracks, where albums and a
rtists are linked by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most p
opular artists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Na
me AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = a
l.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTracks
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"}]

Insert of existing embedding ID: 0e1a2b7b-d65e-53de-b839-edb7afcf4ab1-sql
Add of existing embedding ID: 0e1a2b7b-d65e-53de-b839-edb7afcf4ab1-sql

```
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:55:35.177955387Z', 'message': {'role': 'assistan
t', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_ite
ms" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, 'done_re
ason': 'stop', 'done': True, 'total_duration': 93927241601, 'load_duration': 1010606, 'prompt_eval_count':
1893, 'prompt_eval_duration': 80529455000, 'eval_count': 70, 'eval_duration': 12561624000}
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=mistral: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\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtype
s gives:\n CustomerId     int64\nTotalAlbums    int64\ndtype: object"}, {"role": "user", "content": "Can yo
u generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dat
aframe 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': 'mistral:latest', 'created_at': '2024-06-14T10:56:24.641848622Z', 'message': {'role': 'assistan
t', 'content': ' Here\'s the Python code to create a bar chart using Plotly Express:\n\n```python\nimport p
```

lotly.express as px\n\nfig = px.bar("CustomerId", "TotalAlbums", data=df, title="Top 5 Customers who bought the most albums")\nfig.show()\n```\n\nIf there is only one value in the DataFrame:\n\n```python\nimport plotly.graph_objects as go\n\nif df.shape[0] == 1:\n   fig = go.Indicator(\n   value=df.iloc[0, 1],\n   title={\'text\': "Total Albums", \'font\': {\'size\': 24}},\n   mode="number+delta",\n   delta={\'reference\': 0}\n   )\nelse:\n   fig = px.bar("CustomerId", "TotalAlbums", data=df, title="Top 5 Customers who bought the most albums")\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 49436291132, 'load_duration': 1264153, 'prompt_eval_count': 237, 'prompt_eval_duration': 9849558000, 'eval_count': 220, 'eval_duration': 39453779000}


39
38.5
38
37.5
37
TotalAlbums
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',
    CustomerId  TotalAlbums
 0           1           38
 1           2           38
 2           3           38
 3           4           38
 4           5           38,
 Figure({
     'data': [{'hovertemplate': 'CustomerId=%{x}<br>TotalAlbums=%{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]),
               'xaxis': 'x',
               'y': array([38, 38, 38, 38, 38]),
               'yaxis': 'y'}],
     'layout': {'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': '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 unn
"""
```

```
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 "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 "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 "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\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 r

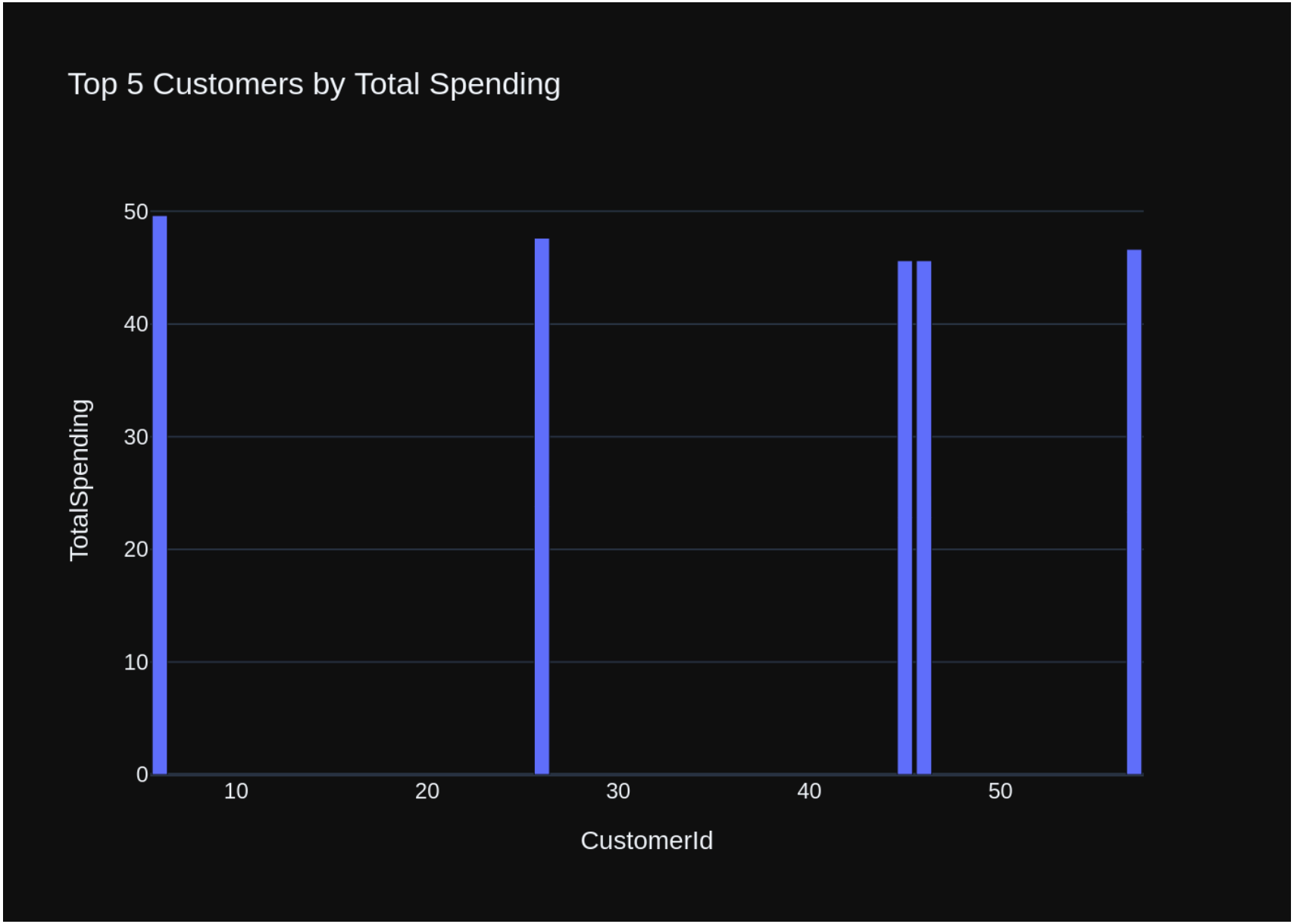
epeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': '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\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 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\nORDER BY AverageInvoiceTotal DESC\nLIMIT 10'}, {'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=mistral: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 ans
wered 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 fo
und on invoices table, calculation using invoice_items detail table is unnecessary \n"}, {"role": "assistan
t", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i
ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n----------------
--------"}, {"role": "user", "content": " \n   Find the top 5 customers who spent the most money overal
l, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail t
able is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent
\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY
TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Hint: album quantity is found in invoice_i
tems, \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\n------------------------"}, {"role":
"user", "content": " \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 cust
omers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "conte
nt": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_item
s\" 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 cust
omers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "conte
nt": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii
ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"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 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 \"invoi
ces\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbu
ms 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": "SEL
ECT 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
TotalAlbums DESC\nLIMIT 1"}, {"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\nORDER BY Ave
rageInvoiceTotal DESC\nLIMIT 10"}, {"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 i
nvoice_items detail table is unnecessary \n"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:58:01.510893523Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM "customers" c\nJOIN "invoices" i O
N c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5'}, 'done_reaso
n': 'stop', 'done': True, 'total_duration': 96776735486, 'load_duration': 890396, 'prompt_eval_count': 199
7, 'prompt_eval_duration': 84220486000, 'eval_count': 63, 'eval_duration': 11567398000}
SELECT c.CustomerId, SUM(i.Total) AS TotalSpending
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpending DESC
LIMIT 5
SELECT c.CustomerId, SUM(i.Total) AS TotalSpending
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpending DESC
LIMIT 5
   CustomerId  TotalSpending
0           6          49.62
1          26          47.62
2          57          46.62
3          45          45.62
4          46          45.62
Ollama parameters:
model=mistral: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 TotalSpending\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Cust
omerId\nORDER BY TotalSpending DESC\nLIMIT 5\n\nThe following is information about the resulting pandas Dat
aFrame 'df': \nRunning df.dtypes gives:\n CustomerId         int64\nTotalSpending    float64\ndtype: objec
t"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataf
rame? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, us
```

e an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T10:58:32.889925312Z', 'message': {'role': 'assistan
t', 'content': ' ```python\nimport plotly.express as px\n\nfig = px.bar(df, x=\'CustomerId\', y=\'TotalSpen
ding\', title="Top 5 Customers by Total Spending")\nfig.update_layout(showlegend=False)\nif len(df) == 1:\n
fig = px.scatter(x=[df[\'CustomerId\'].iloc[0]], y=[df[\'TotalSpending\'].iloc[0]], mode=\'markers\')\n``
`'}, 'done_reason': 'stop', 'done': True, 'total_duration': 31353759554, 'load_duration': 777529, 'prompt_e
val_count': 258, 'prompt_eval_duration': 11208394000, 'eval_count': 109, 'eval_duration': 20014988000}


Top 5 Customers by Total Spending
TotalSpending
50
40
30
20
10
0
10
20
30
40
50
CustomerId

Out[38]:
```
('SELECT c.CustomerId, SUM(i.Total) AS TotalSpending\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerI
d = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpending DESC\nLIMIT 5',
    CustomerId  TotalSpending
 0           6          49.62
 1          26          47.62
 2          57          46.62
 3          45          45.62
 4          46          45.62,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'CustomerId=%{x}<br>TotalSpending=%{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]),
               'xaxis': 'x',
               'y': array([49.62, 47.62, 46.62, 45.62, 45.62]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'showlegend': False,
                'template': '...',
                'title': {'text': 'Top 5 Customers by Total Spending'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalSpending'}}}
 }))
```

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.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n   There are 3 tables: artists,

```
albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumI
d,\n    Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistan
t', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.Art
istId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC
\nLIMIT 10'}, {'role': 'user', 'content': '  \n     Identify artists who have albums with tracks appearing
in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFRO
M "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n  SELECT g2.GenreId\n
FROM "genres" g2\n  JOIN "tracks" t2 ON g2.GenreId = t2.GenreId\n  GROUP BY g2.GenreId\n  HAVING COUNT(g2.G
enreId) > 1\n)\nGROUP BY a.ArtistId, a.Name'}, {'role': 'user', 'content': '  \n     Identify artists who h
ave albums with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.Ar
tistId, a.Name AS ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.Artis
tId IN (\n  SELECT g2.GenreId\n  FROM "genres" g2\n  JOIN "tracks" t2 ON g2.GenreId = t2.GenreId\n  GROUP B
Y g2.GenreId\n  HAVING COUNT(g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'rol
e': 'user', 'content': '  \n     Find the customer who bought the most albums in total quantity (across all
invoices): \n'}, {'role': 'assistant', 'content': 'SELECT 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    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 c.Custome
rId, 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\n
LIMIT 5'}, {'role': 'user', 'content': '  \n    Find all tracks with a name containing "What" (case-insensi
tive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE lower(Name) LIKE '%what%'"}, {'rol
e': 'user', 'content': '  \n     Get all playlists containing at least 10 tracks and the total duration of
those tracks:\n'}]
Ollama parameters:
model=mistral: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.ArtistId, a.Name AS ArtistName, COUN
T(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tra
cks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"ro
le": "user", "content": " \n   There are 3 tables: artists, albums and tracks, where albums and artists ar
e linked by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most popular ar
tists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId)
AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON a
l.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": "ass
istant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.A
rtistId = 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.ArtistI
d, a.Name"}, {"role": "user", "content": " \n   Identify artists who have albums with tracks appearing i
n 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

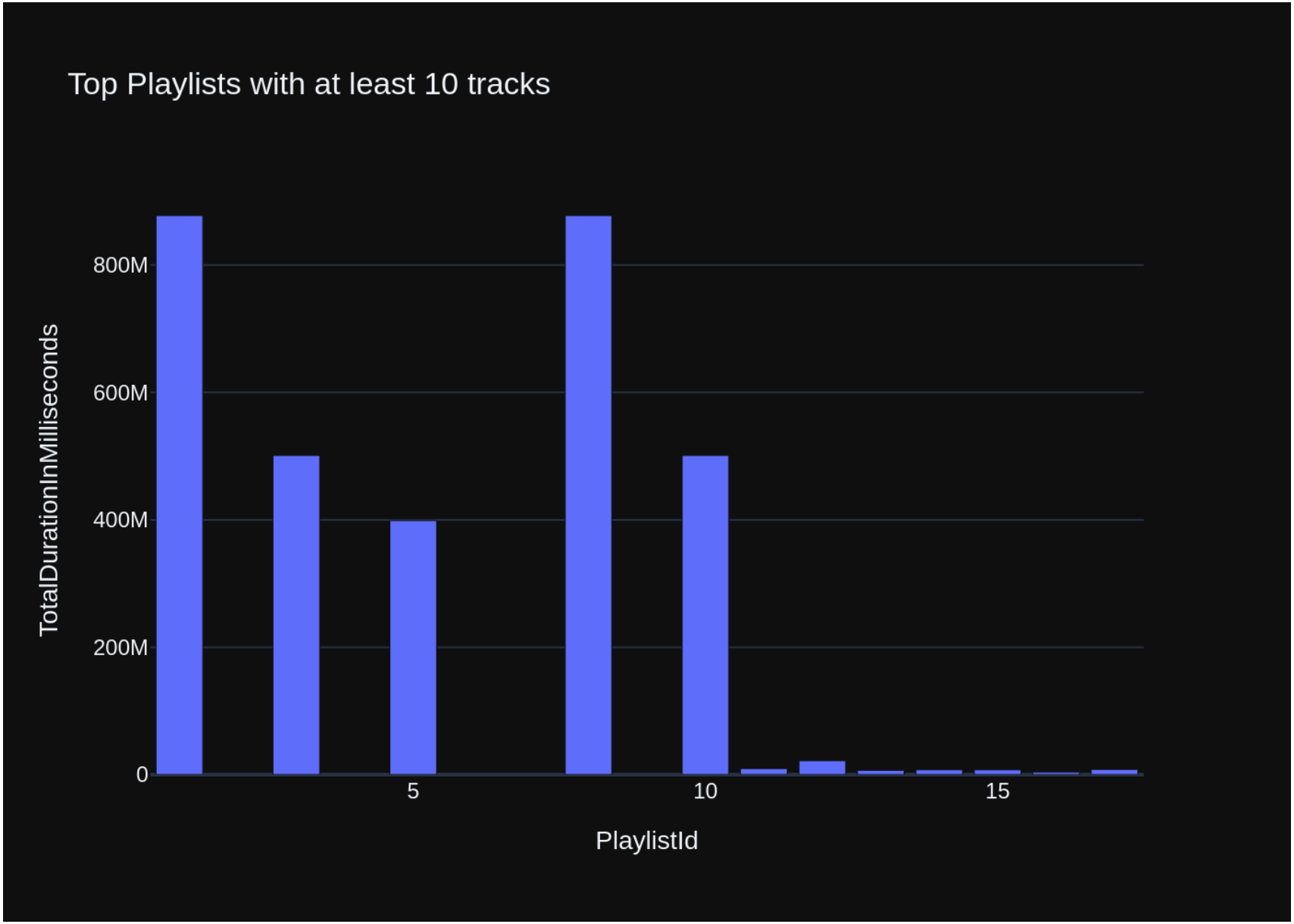
FROM \"genres\" g2\n JOIN \"tracks\" t2 ON g2.GenreId = t2.GenreId\n GROUP BY g2.GenreId\n HAVING COUNT
(g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName"}, {"role": "user", "content": " \n
Find the customer who bought the most albums in total quantity (across all invoices): \n"}, {"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   Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"r
ole": "assistant", "content": "SELECT * FROM tracks WHERE lower(Name) LIKE '%what%'"}, {"role": "user", "co
ntent": " \n   Get all playlists containing at least 10 tracks and the total duration of those track
s:\n"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T11:00:39.385466833Z', 'message': {'role': 'assistan
t', 'content': 'SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDurationInMilliseconds\nFROM "playlists" p
\nJOIN "playlist_track" pt ON p.PlaylistId = pt.PlaylistId\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROU
P BY p.PlaylistId\nHAVING COUNT(pt.PlaylistId) >= 10\nORDER BY TotalDurationInMilliseconds DESC;\n\nFor Pos
tgreSQL:\nSELECT p.playlist_id, SUM(t.milliseconds) AS totalduration\nFROM playlists p\nJOIN playlist_track
pt ON p.playlist_id = pt.playlist_id\nJOIN tracks t ON pt.track_id = t.track_id\nGROUP BY p.playlist_id\nHA
VING COUNT(pt.playlist_id) >= 10\nORDER BY totalduration DESC;'}, 'done_reason': 'stop', 'done': True, 'tot
al_duration': 126411747342, 'load_duration': 864224, 'prompt_eval_count': 1969, 'prompt_eval_duration': 867
64503000, 'eval_count': 211, 'eval_duration': 38962126000}
SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDurationInMilliseconds
FROM "playlists" p
JOIN "playlist_track" pt ON p.PlaylistId = pt.PlaylistId
JOIN "tracks" t ON pt.TrackId = t.TrackId
GROUP BY p.PlaylistId
HAVING COUNT(pt.PlaylistId) >= 10
ORDER BY TotalDurationInMilliseconds DESC;

For PostgreSQL:
SELECT p.playlist_id, SUM(t.milliseconds) AS totalduration
FROM playlists p
JOIN playlist_track pt ON p.playlist_id = pt.playlist_id
JOIN tracks t ON pt.track_id = t.track_id
GROUP BY p.playlist_id
HAVING COUNT(pt.playlist_id) >= 10
ORDER BY totalduration DESC;
Output from LLM: SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDurationInMilliseconds
FROM "playlists" p

```
JOIN "playlist_track" pt ON p.PlaylistId = pt.PlaylistId
JOIN "tracks" t ON pt.TrackId = t.TrackId
GROUP BY p.PlaylistId
HAVING COUNT(pt.PlaylistId) >= 10
ORDER BY TotalDurationInMilliseconds DESC;

For PostgreSQL:
SELECT p.playlist_id, SUM(t.milliseconds) AS totalduration
FROM playlists p
JOIN playlist_track pt ON p.playlist_id = pt.playlist_id
JOIN tracks t ON pt.track_id = t.track_id
GROUP BY p.playlist_id
HAVING COUNT(pt.playlist_id) >= 10
ORDER BY totalduration DESC;
Extracted SQL: SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDurationInMilliseconds
FROM "playlists" p
JOIN "playlist_track" pt ON p.PlaylistId = pt.PlaylistId
JOIN "tracks" t ON pt.TrackId = t.TrackId
GROUP BY p.PlaylistId
HAVING COUNT(pt.PlaylistId) >= 10
ORDER BY TotalDurationInMilliseconds DESC
SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDurationInMilliseconds
FROM "playlists" p
JOIN "playlist_track" pt ON p.PlaylistId = pt.PlaylistId
JOIN "tracks" t ON pt.TrackId = t.TrackId
GROUP BY p.PlaylistId
HAVING COUNT(pt.PlaylistId) >= 10
ORDER BY TotalDurationInMilliseconds DESC
    PlaylistId  TotalDurationInMilliseconds
0            1                    877683083
1            8                    877683083
2            3                    501094957
3           10                    501094957
4            5                    398705153
5           12                     21770592
6           11                      9486559
7           17                      8206312
8           14                      7575051
9           15                      7439811
10          13                      6755730
11          16                      4122018
Ollama parameters:
```

```
model=mistral:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n     Get all playlists containing at least 10 tracks and the
total duration of those tracks:\n'\n\nThe DataFrame was produced using this query: SELECT p.PlaylistId, SUM
(t.Milliseconds) AS TotalDurationInMilliseconds\nFROM \"playlists\" p\nJOIN \"playlist_track\" pt ON p.Play
listId = pt.PlaylistId\nJOIN \"tracks\" t ON pt.TrackId = t.TrackId\nGROUP BY p.PlaylistId\nHAVING COUNT(p
t.PlaylistId) >= 10\nORDER BY TotalDurationInMilliseconds DESC\n\nThe following is information about the re
sulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n PlaylistId                     int64\nTotalDura
tionInMilliseconds    int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plot
ly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If the
re 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': 'mistral:latest', 'created_at': '2024-06-14T11:01:42.211248221Z', 'message': {'role': 'assistan
t', 'content': ' Here\'s the Python code to create a bar chart using Plotly:\n\n```python\nimport plotly.ex
press as px\n\nfig = px.bar(df, x=\'PlaylistId\', y=\'TotalDurationInMilliseconds\', title="Top Playlists w
ith at least 10 tracks")\nfig.show()\n```\n\nFor a single value:\n\n```python\nimport plotly.graph_objects
as go\n\nif df.shape[0] == 1:\n    fig = go.Indicator(\n        domain={"x": [0, 1], "y": [0, 1]},\n
value=df.iloc[0][\'TotalDurationInMilliseconds\'] / (60 * 60 * 24),\n        mode="gauge+number",\n
title={\'text\': "Total Duration (Days)"},\n        gauge={\n            \'axis\': {\'range\': [None, df.il
oc[0][\'TotalDurationInMilliseconds\'] / (60 * 60 * 24)]},\n            \'bar\': {\'color\': "lightblu
e"},\n            \'steps\': [{\'range\': [0, 0.5], \'color\': "red"}, {\'range\': [0.5, 1], \'color\': "gr
een"}]\n        }\n    )\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 6280019
4159, 'load_duration': 42574142, 'prompt_eval_count': 279, 'prompt_eval_duration': 10756527000, 'eval_coun
t': 303, 'eval_duration': 51950994000}
```


Top Playlists with at least 10 tracks
TotalDurationInMilliseconds
800M
600M
400M
200M
0
5
10
15
PlaylistId

Out[39]:
```
('SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDurationInMilliseconds\nFROM "playlists" p\nJOIN "playl
ist_track" pt ON p.PlaylistId = pt.PlaylistId\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROUP BY p.Playl
istId\nHAVING COUNT(pt.PlaylistId) >= 10\nORDER BY TotalDurationInMilliseconds DESC',
     PlaylistId  TotalDurationInMilliseconds
 0            1                    877683083
 1            8                    877683083
 2            3                    501094957
 3           10                    501094957
 4            5                    398705153
 5           12                     21770592
 6           11                      9486559
 7           17                      8206312
 8           14                      7575051
 9           15                      7439811
 10          13                      6755730
 11          16                      4122018,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'PlaylistId=%{x}<br>TotalDurationInMilliseconds=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([ 1,  8,  3, 10,  5, 12, 11, 17, 14, 15, 13, 16]),
               'xaxis': 'x',
               'y': array([877683083, 877683083, 501094957, 501094957, 398705153,  21770592,
                           9486559,   8206312,   7575051,   7439811,   6755730,   4122018]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Top Playlists with at least 10 tracks'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'PlaylistId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalDurationInMilliseco
nds'}}}
 }))
```

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.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n   There are 3 tables: artists, albums 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 albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId\n-------------------------'}, {'role': 'user', 'content': ' \n   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   Identify artists who have albums with tracks appearing in multiple genres:\n\n\n'}]
Ollama parameters:
model=mistral: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.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistId, a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n   There are 3 tables: artists, albums 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 albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId\n-------------------------"}, {"role": "user", "content": " \n   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 TotalTra

cks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks D
ESC\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 c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c
\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.Invoice
Id\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Hint:
album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums i
n total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(i
i.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGR
OUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Identify a
rtists who have albums with tracks appearing in multiple genres:\n\n\n"}]
Ollama Response:
{'model': 'mistral:latest', 'created_at': '2024-06-14T11:02:58.617205873Z', 'message': {'role': 'assistan
t', '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.Genre
Id = t2.GenreId\n GROUP BY g2.GenreId\n HAVING COUNT(g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDE
R BY ArtistName\n\nThis query will find artists who have albums that belong to multiple genres. The subquer
y in the WHERE clause groups by genre and selects only those genres which contain more than one track. The
outer query then joins these genres with the artists\' table and groups the results by artist name.'}, 'don
e_reason': 'stop', 'done': True, 'total_duration': 76273678538, 'load_duration': 1347112, 'prompt_eval_coun
t': 1024, 'prompt_eval_duration': 39959845000, 'eval_count': 206, 'eval_duration': 35726896000}
```
SELECT a.ArtistId, a.Name AS ArtistName
FROM "artists" a
JOIN "albums" al ON a.ArtistId = al.ArtistId
WHERE a.ArtistId IN (
  SELECT g2.GenreId
  FROM "genres" g2
  JOIN "tracks" t2 ON g2.GenreId = t2.GenreId
  GROUP BY g2.GenreId
  HAVING COUNT(g2.GenreId) > 1
)
GROUP BY a.ArtistId, a.Name ORDER BY ArtistName
```

This query will find artists who have albums that belong to multiple genres. The subquery in the WHERE clau
se groups by genre and selects only those genres which contain more than one track. The outer query then jo
ins these genres with the artists' table and groups the results by artist name.

```
SELECT a.ArtistId, a.Name AS ArtistName
FROM "artists" a
JOIN "albums" al ON a.ArtistId = al.ArtistId
WHERE a.ArtistId IN (
  SELECT g2.GenreId
  FROM "genres" g2
```

```
  JOIN "tracks" t2 ON g2.GenreId = t2.GenreId
  GROUP BY g2.GenreId
  HAVING COUNT(g2.GenreId) > 1
)
GROUP BY a.ArtistId, a.Name ORDER BY ArtistName

This query will find artists who have albums that belong to multiple genres. The subquery in the WHERE clau
se groups by genre and selects only those genres which contain more than one track. The outer query then jo
ins these genres with the artists' table and groups the results by artist name.
Couldn't run sql:  Execution failed on sql 'SELECT a.ArtistId, a.Name AS ArtistName
FROM "artists" a
JOIN "albums" al ON a.ArtistId = al.ArtistId
WHERE a.ArtistId IN (
  SELECT g2.GenreId
  FROM "genres" g2
  JOIN "tracks" t2 ON g2.GenreId = t2.GenreId
  GROUP BY g2.GenreId
  HAVING COUNT(g2.GenreId) > 1
)
GROUP BY a.ArtistId, a.Name ORDER BY ArtistName

This query will find artists who have albums that belong to multiple genres. The subquery in the WHERE clau
se groups by genre and selects only those genres which contain more than one track. The outer query then jo
ins these genres with the artists' table and groups the results by artist name.': near "This": syntax error
```

Check completion time

In [ ]:

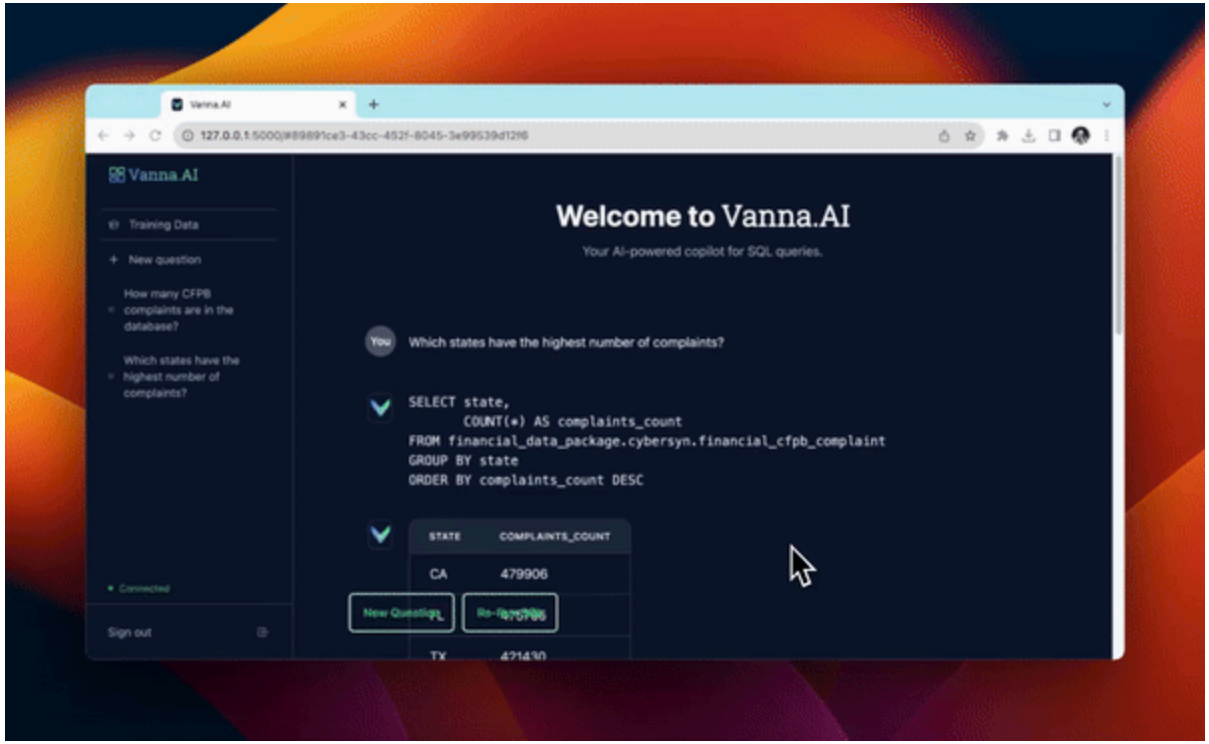
In [41]:
```
ts_stop = time()

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 'mistral' LLM took : 3101.98 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