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 = 'llama3'
```

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	127fd4bd-b9af-539d-9313-1d0234d073b7-sql	\n There are 3 tables: artists, albums and...	SELECT a.Name, COUNT(t.TrackId) AS TotalTracks...	sql
3	32b99e7b-31ab-55d8-8431-fb010fa7af85-sql	\n Find the top 5 customers who spent th...	SELECT c.CustomerId, SUM(i.Total) AS TotalSpen...	sql
4	d8a2f948-dffa-5524-a5f9-174cc1a8da73-sql	Can you list all tables in the SQLite database...	SELECT name FROM sqlite_master WHERE type='table'	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-	None	CREATE TABLE "employees"\r\n(\r\n	ddl

	id	question	content	training_data_type
	243a9b6a3320-ddl		EmployeeI...	
12	65df0648-bf05-5f75-9365-c21f54b2302d-ddl	None	CREATE TABLE "media_types"\r\n(\r\n MediaTy...	ddl
13	6b585176-e66d-5b23-8d86-ca8a80e3af3d-ddl	None	CREATE INDEX IFK_EmployeeReportsTo ON "employe...	ddl
14	868758b8-e018-55e7-8cc3-75c0e6d211c8-ddl	None	CREATE INDEX IFK_TrackAlbumId ON "tracks" (Alb...	ddl
15	9ea4613d-c1be-5a77-ada9-c54ee3f0cab7-ddl	None	CREATE INDEX IFK_TrackMediaTypeId ON "tracks" ...	ddl
16	a9c9a852-608d-5ef2-aede-26ba098d83d1-ddl	None	CREATE INDEX IFK_TrackGenreId ON "tracks" (Gen...	ddl
17	b42cc9e1-9219-5a42-9a06-de906f76239e-ddl	None	CREATE TABLE "tracks"\r\n(\r\n TrackId INTE...	ddl
18	c387b9d2-5ff4-5a07-8364-f5dab45bb2a9-ddl	None	CREATE TABLE "genres"\r\n(\r\n GenreId INTE...	ddl
19	d654f328-dc36-549e-84c3-06ee0db7e0f7-ddl	None	CREATE TABLE "playlist_track"\r\n(\r\n Play...	ddl
20	d93f0d68-023d-5afb-8121-ba346699d318-ddl	None	CREATE TABLE "customers"\r\n(\r\n CustomerI...	ddl
21	e5879308-329e-543f-a693-0c14e2f9972e-ddl	None	CREATE INDEX IFK_InvoiceLineTrackId ON "invoic...	ddl
22	ea84418b-1a28-59b4-a1f4-2fb674208adc-ddl	None	CREATE TABLE sqlite_sequence(name,seq)	ddl
0	9d2550eb-8e22-54cd-9fad-9e1be65ab03a-doc	None	In the SQLite database invoice means order	documentation

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

In [13]:
```
df_ddl
```

Out[13]:

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

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

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

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

Asking the AI

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

In [16]:
```
ts_start = time()
```

```
SELECT name FROM sqlite_master WHERE type = 'table';
```

In [17]:
```
vn.ask(question="Can you list all tables in the SQLite database catalog?")
```

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

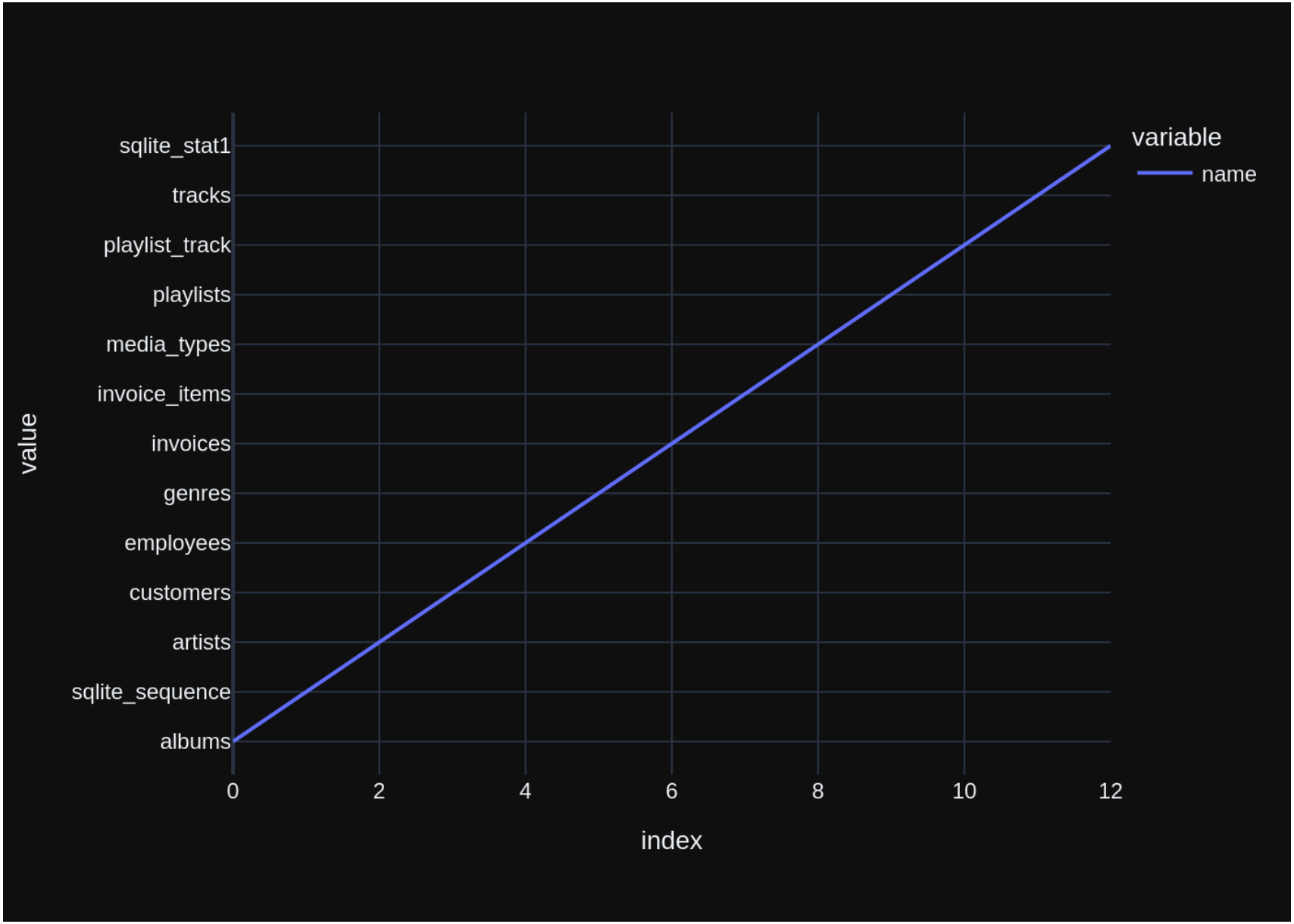
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE TABLE 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   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC

\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the top 5 customers who spent the most money 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 Tot
alSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant',
'content': 'SELECT COUNT(*) FROM "customers"'}, {'role': 'user', 'content': 'Can you list all tables in the
SQLite database catalog?'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE 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 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   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM \"customers\""}, {"role": "user", "content": "Can you list all tables in the SQLite database catalog?"}]

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

```
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:41:27.367782099Z', 'message': {'role': 'assistant',
'content': "SELECT name FROM sqlite_master WHERE type='table'"}, 'done_reason': 'stop', 'done': True, 'tota
l_duration': 39403111660, 'load_duration': 1716996616, 'prompt_eval_count': 1127, 'prompt_eval_duration': 3
5404057000, 'eval_count': 11, 'eval_duration': 1900754000}
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=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: 'Can you list all tables in the SQLite database catalog?'\n\nThe
DataFrame was produced using this query: SELECT name FROM sqlite_master WHERE type='table'\n\nThe following
is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n name    object\ndtyp
e: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of t
he dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataf
rame, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the cod
e."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:41:37.741811699Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nfig = px.bar(df, x='name', y='')\nfig.show()\n```"}, 'done_re
ason': 'stop', 'done': True, 'total_duration': 10352841522, 'load_duration': 669138, 'prompt_eval_count': 1
50, 'prompt_eval_duration': 5482320000, 'eval_count': 27, 'eval_duration': 4728513000}
```


variable
name
value
sqlite_stat1
tracks
playlist_track
playlists
media_types
invoice_items
invoices
genres
employees
customers
artists
sqlite_sequence
albums
0
2
4
6
8
10
12
index

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': [{'hovertemplate': 'variable=name<br>index=%{x}<br>value=%{y}<extra></extra>',
              'legendgroup': 'name',
              'line': {'color': '#636efa', 'dash': 'solid'},
              'marker': {'symbol': 'circle'},
              'mode': 'lines',
              'name': 'name',
              'orientation': 'v',
              'showlegend': True,
              'type': 'scatter',
              'x': array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12]),
              'xaxis': 'x',
              'y': array(['albums', 'sqlite_sequence', 'artists', 'customers', 'employees',
                          'genres', 'invoices', 'invoice_items', 'media_types', 'playlists',
                          'playlist_track', 'tracks', 'sqlite_stat1'], dtype=object),
              'yaxis': 'y'}],
    'layout': {'legend': {'title': {'text': 'variable'}, 'tracegroupgap': 0},
               'margin': {'t': 60},
               'template': '...',
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'index'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'value'}}}
})
```

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

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "employees"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE "playlists"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE TABLE "albums"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "media_types"\r\n(\r\n   MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total)

AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId
\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   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.InvoiceLineId) AS Total
Albums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nOR
DER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role':
'assistant', 'content': 'SELECT COUNT(*) FROM "customers"'}, {'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, w
here albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find
the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT
a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJO
IN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role':
'user', 'content': "which table stores customer's orders"}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"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 PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE TABLE \"albums\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId)\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"media_types\"\r\n(\r\n   MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n   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": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM \"customers\""}, {"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": " \nThere 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": "which table stores customer's orders"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:42:30.287415659Z', 'message': {'role': 'assistant', 'content': 'The "invoices" table stores customers\' orders.'}, 'done_reason': 'stop', 'done': True, 'total_duration': 52110789124, 'load_duration': 836052, 'prompt_eval_count': 1314, 'prompt_eval_duration': 49717291000, 'eval_count': 11, 'eval_duration': 1947400000}
The "invoices" table stores customers' orders.
The "invoices" table stores customers' orders.

```
Couldn't run sql:  Execution failed on sql 'The "invoices" table stores customers' orders.': near "The": sy
ntax error
```

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

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

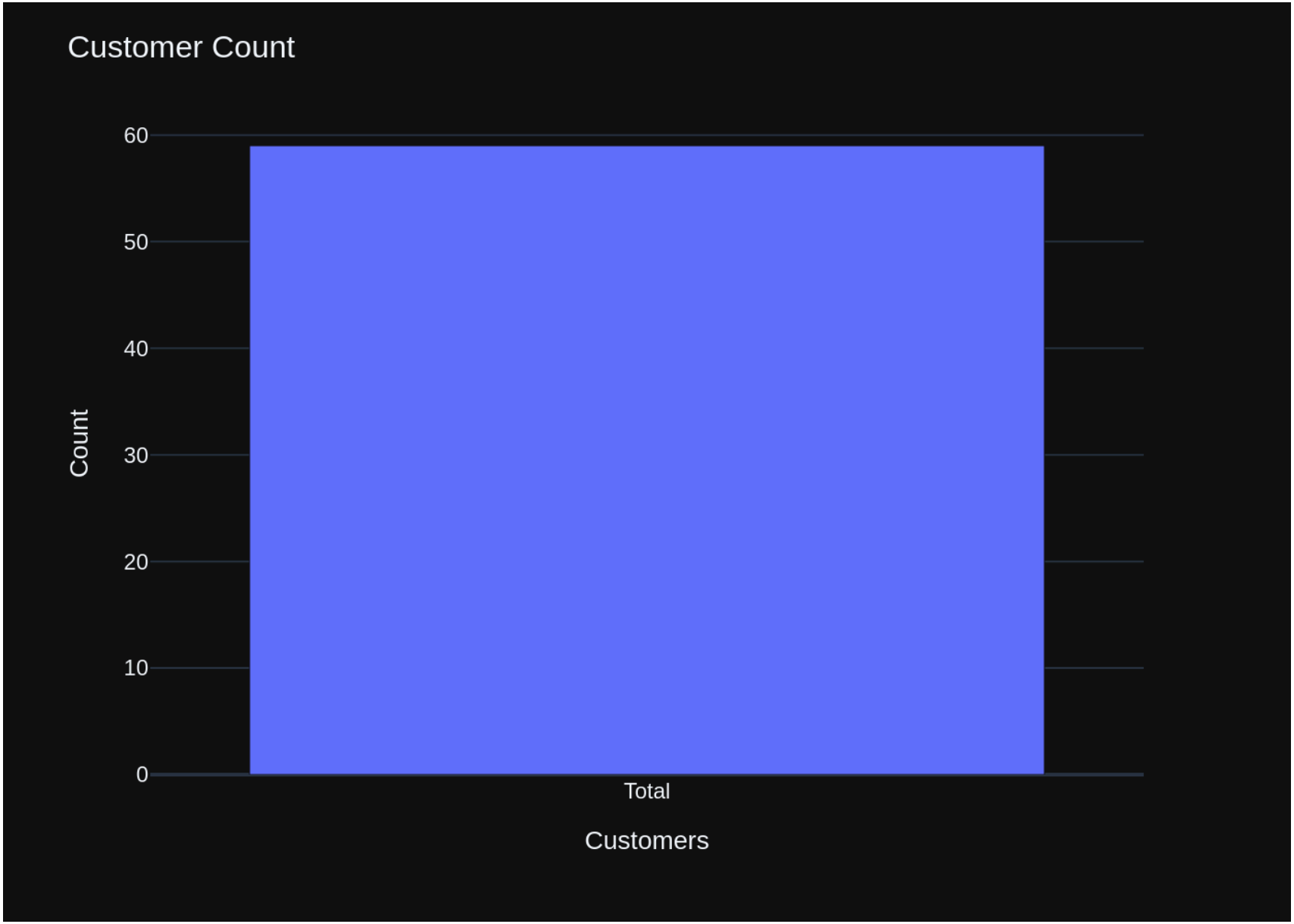
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "customers"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE TABLE "albums"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE "employees"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlists"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM "customers"'}, {'role': 'user', 'content': ' \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Hint: album quantity is found

in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   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': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': 'How many customers are there'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE \"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 NVARCHA

R(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVA
RCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlists\"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCRE
MENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the SQLite database invoice
means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid S
QL query without any explanations for the question. \n2. If the provided context is almost sufficient but r
equires knowledge of a specific string in a particular column, please generate an intermediate SQL query to
find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If
the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most re
levant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly
as it was given before. \n"}, {"role": "user", "content": "How many customers are there"}, {"role": "assist
ant", "content": "SELECT COUNT(*) FROM \"customers\""}, {"role": "user", "content": " \n   Find the 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
Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most al
bums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, C
OUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.I
nvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n
There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and
tracks are linked by AlbumId,\n   Can you find the top 10 most popular artists based on the number of trac
ks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\"
a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.
Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": "Can you list all tables in the SQ
Lite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='tabl
e'"}, {"role": "user", "content": "How many customers are there"}]

Insert of existing embedding ID: 0658ba3d-98ff-51f4-9006-a24f87045858-sql
Add of existing embedding ID: 0658ba3d-98ff-51f4-9006-a24f87045858-sql

```
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:43:18.183068493Z', 'message': {'role': 'assistant',
'content': 'SELECT COUNT(*) FROM "customers";'}, 'done_reason': 'stop', 'done': True, 'total_duration': 478
52137170, 'load_duration': 849555, 'prompt_eval_count': 1196, 'prompt_eval_duration': 46052429000, 'eval_co
unt': 8, 'eval_duration': 1343766000}
SELECT COUNT(*) FROM "customers";
Output from LLM: SELECT COUNT(*) FROM "customers";
Extracted SQL: SELECT COUNT(*) FROM "customers"
SELECT COUNT(*) FROM "customers"
   COUNT(*)
0        59
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: 'How many customers are there'\n\nThe DataFrame was produced usin
g this query: SELECT COUNT(*) FROM \"customers\"\n\nThe following is information about the resulting pandas
DataFrame 'df': \nRunning df.dtypes gives:\n COUNT(*)    int64\ndtype: object"}, {"role": "user", "conten
t": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a
pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with o
nly Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:43:32.300924355Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nfig = px.bar(x=['Total'], y=df.iloc[0], labels={'x': 'Custome
rs', 'y': 'Count'})\nfig.update_layout(title='Customer Count')\nfig.show()\n```"}, 'done_reason': 'stop',
'done': True, 'total_duration': 14097149399, 'load_duration': 777507, 'prompt_eval_count': 118, 'prompt_eva
l_duration': 4729162000, 'eval_count': 50, 'eval_duration': 9233218000}
```


Customer Count
60
50
40
30
20
10
0
Count
Total
Customers

Out[19]:
```
('SELECT COUNT(*) FROM "customers"',
   COUNT(*)
0        59,
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'Customers=%{x}<br>Count=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'v',
              'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
              'x': array(['Total'], dtype=object),
              'xaxis': 'x',
              'y': array([59]),
              'yaxis': 'y'}],
    'layout': {'barmode': 'relative',
               'legend': {'tracegroupgap': 0},
               'margin': {'t': 60},
               'template': '...',
               'title': {'text': 'Customer Count'},
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Customers'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Count'}}}
})
```

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

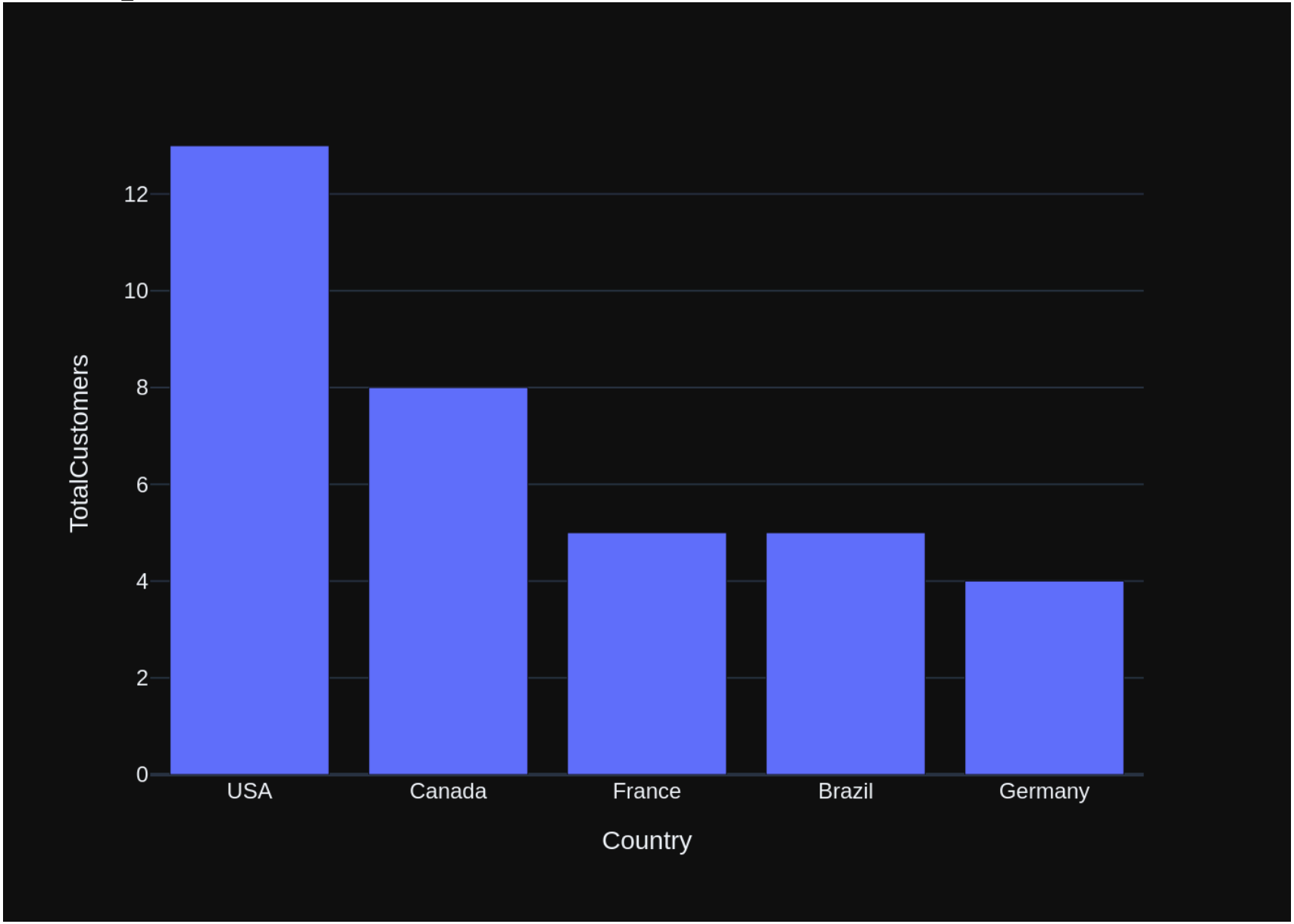
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY 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': ' \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM "customers"'}, {'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': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE \"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   N
ame NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE T
ABLE \"employees\"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARC
HAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INT
EGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHA
R(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone
NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES
\"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums
\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r
\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n   PlaylistId INTEG
ER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistI
d, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACT
ION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE \"tracks\"\r\n(\r
\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   Album
Id INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r
\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n
FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n
FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n
FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION\r\n)\n\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidel
ines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations
for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific st
ring in a particular column, please generate an intermediate SQL query to find the distinct strings in that
column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficie
nt, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the quest
ion has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"ro
le": "user", "content": " \n   Find the top 5 customers who spent the most money overall, \n   \n
Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessa
ry \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customer
s\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC
\nLIMIT 5"}, {"role": "user", "content": " \n   Hint: album quantity is found in invoice_items, \n   \n
Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role":
"assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\n
JOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\n
LIMIT 5"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "S
ELECT COUNT(*) FROM \"customers\""}, {"role": "user", "content": " \n   There are 3 tables: artists, album
s and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n
Can you find the top 10 most popular artists based on the number of tracks\n"}, {"role": "assistant", "cont
ent": "SELECT a.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\nLIM

```
IT 10"}, {"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":
"what are the top 5 countries that customers come from?"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:44:29.426446137Z', 'message': {'role': 'assistant',
'content': 'SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY
TotalCustomers DESC\nLIMIT 5'}, 'done_reason': 'stop', 'done': True, 'total_duration': 57009408520, 'load_d
uration': 922205, 'prompt_eval_count': 1299, 'prompt_eval_duration': 50744226000, 'eval_count': 31, 'eval_d
uration': 5803091000}
SELECT c.Country, COUNT(*) AS TotalCustomers
FROM "customers" c
GROUP BY c.Country
ORDER BY TotalCustomers DESC
LIMIT 5
SELECT c.Country, COUNT(*) AS TotalCustomers
FROM "customers" c
GROUP BY c.Country
ORDER BY TotalCustomers DESC
LIMIT 5
   Country  TotalCustomers
0      USA              13
1   Canada               8
2   France               5
3   Brazil               5
4  Germany               4
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: 'what are the top 5 countries that customers come from?'\n\nThe D
ataFrame was produced using this query: SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM \"customers\" c
\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5\n\nThe following is information about the resul
ting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Country           object\nTotalCustomers     int64
\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the result
s of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the
dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the
code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:44:49.355238571Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\n\nfig = px.bar(df, x='Country', y='TotalCustomers', title='To
```

p 5 Countries by Number of Customers')\n\nif df.shape[0] == 1:\n   fig.update_layout(yaxis_title='Number o
f Customers')\nelse:\n   fig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 1990349
7609, 'load_duration': 658602, 'prompt_eval_count': 174, 'prompt_eval_duration': 7252377000, 'eval_count':
66, 'eval_duration': 12516235000}



Out[20]:
```
('SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCust
omers DESC\nLIMIT 5',
    Country  TotalCustomers
 0      USA             13
 1   Canada              8
 2   France              5
 3   Brazil              5
 4  Germany              4,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'Country=%{x}<br>TotalCustomers=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['USA', 'Canada', 'France', 'Brazil', 'Germany'], dtype=object),
               'xaxis': 'x',
               'y': array([13,  8,  5,  5,  4]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Country'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalCustomers'}}}
 }))
```

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 6, updating n_results = 6
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_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   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 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': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'c

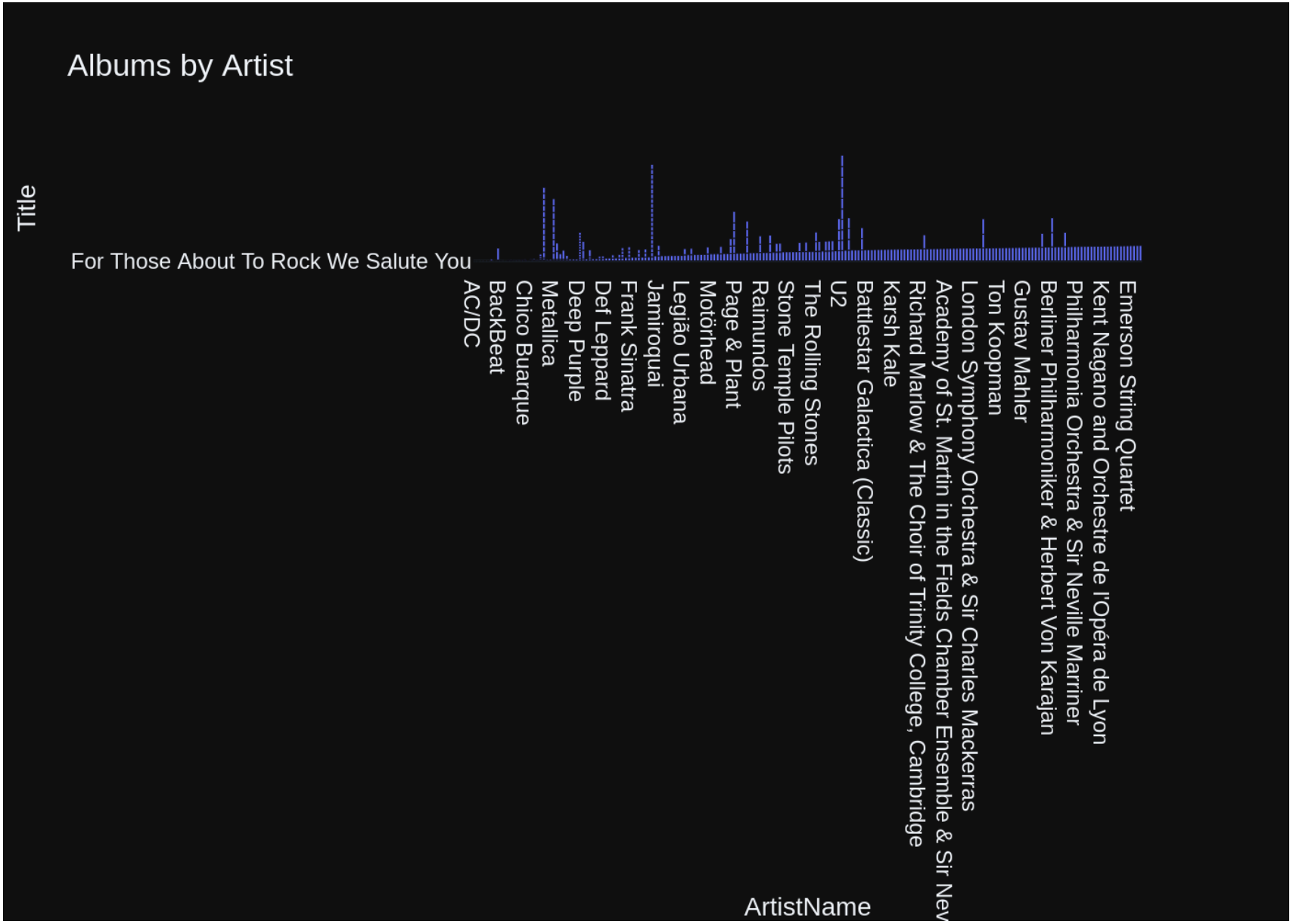
ontent': 'SELECT COUNT(*) FROM "customers"'}, {'role': 'user', 'content': ' \n   List all albums and thei
r corresponding artist names \n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat 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 P
RIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   Medi
aTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEG
ER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId)
REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreI
d) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (Med
iaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n
CREATE INDEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"artists\"\r\n(\r\n   ArtistId INTE
GER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON
\"tracks\" (GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\n\nCREATE TAB
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
There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and
tracks are linked by AlbumId,\n   Can you find the top 10 most popular artists based on the number of trac
ks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\"
a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.
Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n   Hint: album quantity is f
ound in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (a
cross all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS
TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Custo
merId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "Can you list all tables in the SQ
Lite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='tabl

```
e'"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assi
stant", "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 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\nG
ROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": "How many customers
are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM \"customers\""}, {"role": "user", "cont
ent": "  \n    List all albums and their corresponding artist names  \n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:45:38.135771676Z', 'message': {'role': 'assistant',
'content': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.Arti
stId = ar.ArtistId'}, 'done_reason': 'stop', 'done': True, 'total_duration': 48687079961, 'load_duration':
829881, 'prompt_eval_count': 1011, 'prompt_eval_duration': 41304926000, 'eval_count': 37, 'eval_duration':
6928569000}
SELECT a.Title, a.ArtistId, ar.Name AS ArtistName
FROM "albums" a
JOIN "artists" ar ON a.ArtistId = ar.ArtistId
SELECT a.Title, a.ArtistId, ar.Name AS ArtistName
FROM "albums" a
JOIN "artists" ar ON a.ArtistId = ar.ArtistId
                                                Title  ArtistId  \
0                For For Those About To Rock We Salute You         1
1                                     Balls to the Wall         2
2                                     Restless and Wild         2
3                                     Let There Be Rock         1
4                                              Big Ones         3
..                                                  ...       ...
342                             Respighi:Pines of Rome       226
343  Schubert: The Late String Quartets & String Qu...       272
344                                Monteverdi: L'Orfeo       273
345                              Mozart: Chamber Music       274
346  Koyaanisqatsi (Soundtrack from the Motion Pict...       275

                                          ArtistName
0                                              AC/DC
1                                             Accept
2                                             Accept
3                                              AC/DC
4                                           Aerosmith
..                                                ...
342                                   Eugene Ormandy
```

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

[347 rows x 3 columns]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all albums and their corresponding artist names
\n'\n\nThe DataFrame was produced using this query: SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM
\"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId\n\nThe following is information about the res
ulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Title         object\nArtistId      int64\nArti
stName    object\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to
chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only
one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explana
tions -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:45:53.059896473Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\n\nfig = px.bar(df, x='ArtistName', y='Title', title='Albums b
y Artist')\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 14895504329, 'load_du
ration': 41223139, 'prompt_eval_count': 185, 'prompt_eval_duration': 7692887000, 'eval_count': 37, 'eval_du
ration': 7096350000}
```


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

Out[21]: ('SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = a
r.ArtistId',

	Title	ArtistId	ArtistName
0	For Those About To Rock We Salute You	1	AC/DC
1	Balls to the Wall	2	Accept
2	Restless and Wild	2	Accept
3	Let There Be Rock	1	AC/DC
4	Big Ones	3	Aerosmith
..	...	...	...
342	Respighi:Pines of Rome	226	Eugene Ormandy
343	Schubert: The Late String Quartets & String Qu...	272	Emerson String Quartet
344	Monteverdi: L'Orfeo	273	C. Monteverdi, Nigel Rogers - Chiaroscuro; Lon...
345	Mozart: Chamber Music	274	Nash Ensemble
346	Koyaanisqatsi (Soundtrack from the Motion Pict...	275	Philip Glass Ensemble

[347 rows x 3 columns],
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'ArtistName=%{x}
Title=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'v',
              'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
              'x': array(['AC/DC', 'Accept', 'Accept', ...,
                          'C. Monteverdi, Nigel Rogers - Chiaroscuro; London Baroque; London Cornett & Sa

```
ckbu',
                            'Nash Ensemble', 'Philip Glass Ensemble'], dtype=object),
               'xaxis': 'x',
               'y': array(['For Those About To Rock We Salute You', 'Balls to the Wall',
                           'Restless and Wild', ..., "Monteverdi: L'Orfeo",
                           'Mozart: Chamber Music',
                           'Koyaanisqatsi (Soundtrack from the Motion Picture)'], dtype=object),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Albums by Artist'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'ArtistName'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Title'}}}
 }))
```

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

vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON "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   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   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': '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   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': 'what are the top 5 countries that customers come from?'}, {'role': 'assistan
t', 'content': 'SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER
BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'a
ssistant', 'content': 'SELECT COUNT(*) FROM "customers"'}, {'role': 'user', 'content': ' \n   Find all tr
acks with a name containing "What" (case-insensitive)\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON \"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   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.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTrack
s DESC\nLIMIT 10"}, {"role": "user", "content": " \n   List all albums and their corresponding artist nam
es \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM \"album

s\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "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   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": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM \"customers\""}, {"role": "user", "content": " \n   Find all tracks with a name containing \"What\" (case-insensitive)\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:46:42.449127219Z', 'message': {'role': 'assistant', 'content': 'SELECT * \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%\''}, 'done_reason': 'stop', 'done': True, 'total_duration': 49231998940, 'load_duration': 772120, 'prompt_eval_count': 1097, 'prompt_eval_duration': 45621825000, 'eval_count': 17, 'eval_duration': 3044669000}
```
SELECT *
FROM "tracks"
WHERE LOWER(Name) LIKE '%what%'
```
```
SELECT *
FROM "tracks"
WHERE LOWER(Name) LIKE '%what%'
```

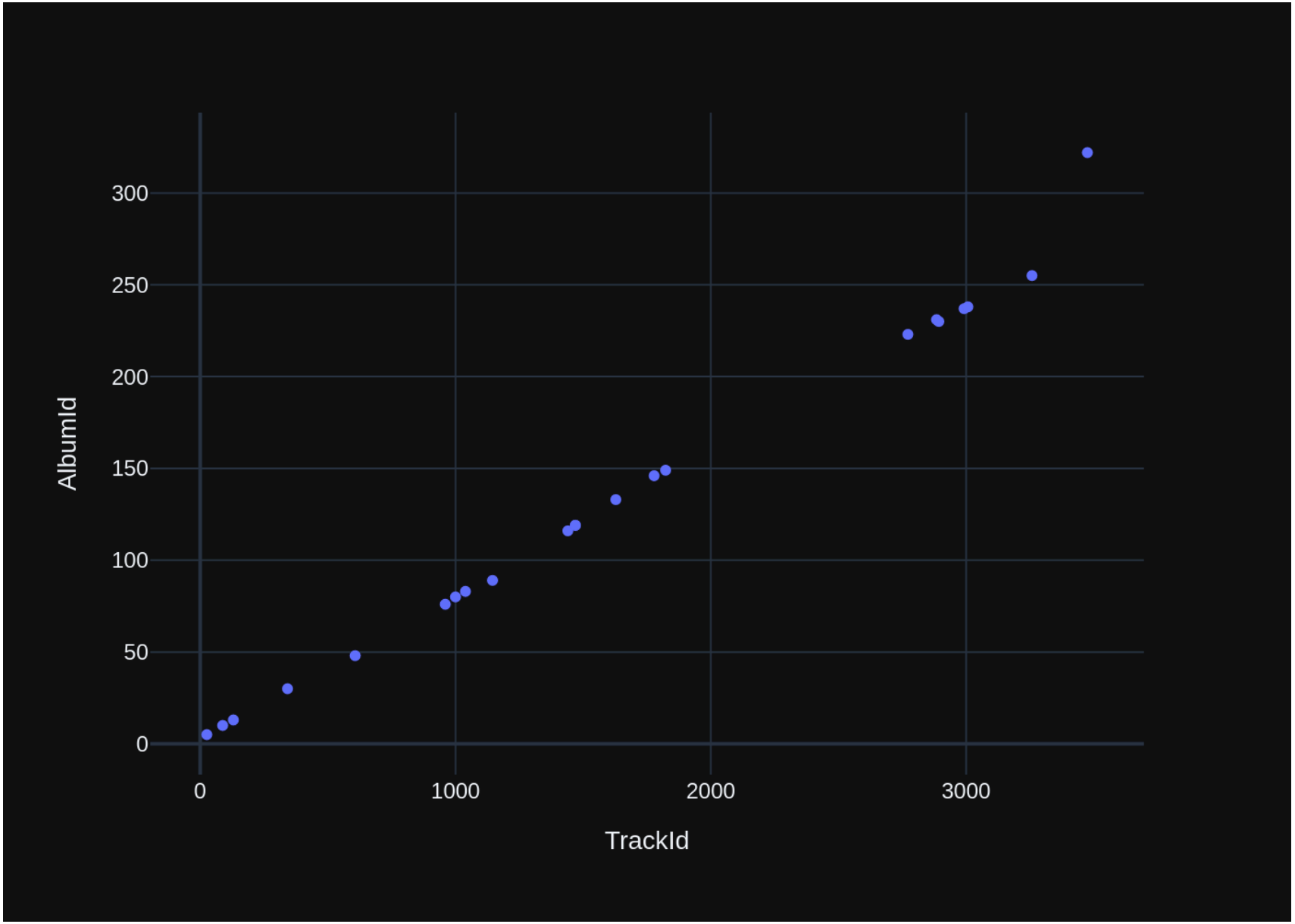
	TrackId	Name	AlbumId	\
0	26	What It Takes	5	
1	88	What You Are	10	
2	130	Do what cha wanna	13	
3	342	What is and Should Never Be	30	
4	607	So What	48	
5	960	What A Day	76	
6	1000	What If I Do?	80	
7	1039	What Now My Love	83	
8	1145	Whatsername	89	
9	1440	Whatever It Is, I Just Can't Stop	116	
10	1469	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=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Find all tracks with a name containing \"What\" (case-in
sensitive)\n'\n\nThe DataFrame was produced using this query: SELECT * \nFROM \"tracks\" \nWHERE LOWER(Nam
e) LIKE '%what%'\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dt
ypes gives:\n TrackId           int64\nName             object\nAlbumId           int64\nMediaTypeId
int64\nGenreId           int64\nComposer         object\nMilliseconds      int64\nBytes             int64\n
UnitPrice       float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly c
ode to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there i
s only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any
explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:47:14.368552993Z', 'message': {'role': 'assistant',
'content': '```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = px.bar(df, x=\'Nam
e\', y=\'Milliseconds\')\nfig.update_layout(title="Tracks with Name Containing \'What\'", xaxis_title="Trac
k Names", yaxis_title="Duration (ms)")\n\nfig2 = go.Figure(data=[go.Indicator(\n    mode = "number+delt
a",\n    value = df[\'UnitPrice\'].mean(),\n    title = "Average Price",\n    delta = {\'reference\': 10,
\'value\': int(df[\'UnitPrice\'].mean())}\n)])\n\nfig.show()\nfig2.show()\n```'}, 'done_reason': 'stop', 'd
one': True, 'total_duration': 31890708441, 'load_duration': 676142, 'prompt_eval_count': 204, 'prompt_eval_
duration': 7971342000, 'eval_count': 128, 'eval_duration': 23825815000}
```


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

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

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

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

    Milliseconds      Bytes  UnitPrice
0         310622   10144730       0.99
1         249391    5988186       0.99
2         274155    9018565       0.99
3         260675    8497116       0.99
4         564009   18360449       0.99
5         158275    5203430       0.99
6         302994    9929799       0.99
7         149995    4913383       0.99
8         252316    8244843       0.99
9         247222    8249453       0.99
10        230974    7517083       0.99
11        247719    8043765       0.99
12        287973    9369385       0.99
13        142027    4631104       0.99
14        189152    6162894       0.99
15        221387    7251478       0.99
16       2610250  484583988       1.99
17       2616410  183867185       1.99
18        353567   11542247       0.99
19        280764    9306737       0.99
20        215084    3499018       0.99
21        209573    3426106       0.99  ,
Figure({
    'data': [{'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 8, updating n_results = 8
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

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

S TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\n
ORDER 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 al
l invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAl
bums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDE
R BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'as
sistant', 'content': 'SELECT COUNT(*) FROM "customers"'}, {'role': 'user', 'content': 'what are the top 5 c
ountries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(*) AS Total
Customers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'use
r', 'content': ' \n   There are 3 tables: artists, albums and tracks, where albums and artists are linked
by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most popular artists bas
ed on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalT
racks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.Al
bumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n   List al
l albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.Ar
tistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId'}, {'role':
'user', 'content': ' \n   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': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SE
LECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': ' \n   Get the total numbe
r of invoices for each customer\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and 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 top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on inv
oices table, calculation using invoice_items detail table is unnecessary \n"}, {"role": "assistant", "conte
nt": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.Custo
merId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "conten
t": " \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bough
t the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.
CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.Inv
oiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "conte
nt": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM \"customers
\""}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assi
stant", "content": "SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM \"customers\" c\nGROUP BY c.Country
\nORDER BY TotalCustomers 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 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.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTrack
s DESC\nLIMIT 10"}, {"role": "user", "content": " \n   List all albums and their corresponding artist nam
es \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM \"album

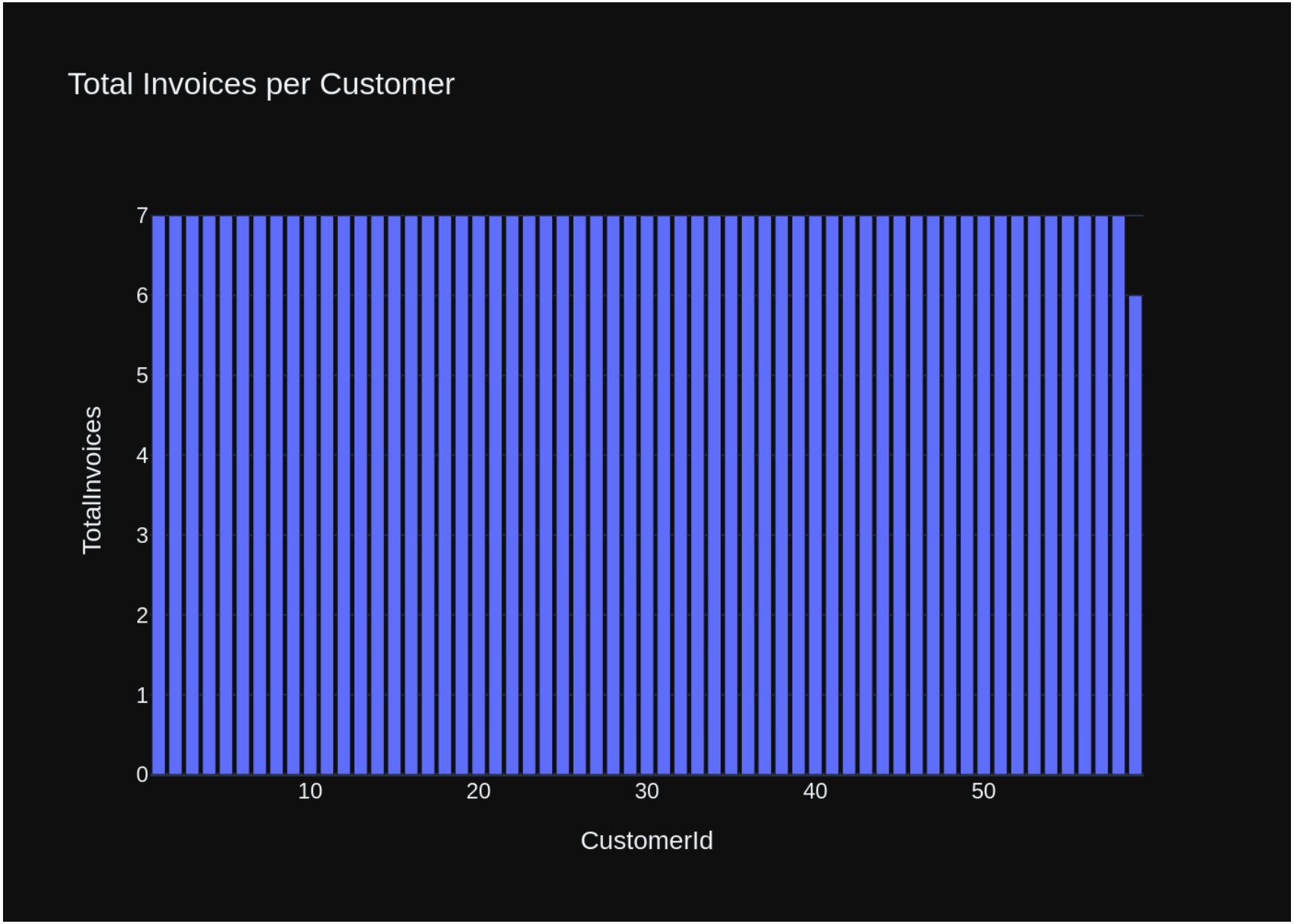
```
s\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "  \n    Find all tra
cks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * \nFR
OM \"tracks\" \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": "Can you list all tables in
the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type
='table'"}, {"role": "user", "content": "  \n    Get the total number of invoices for each customer\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:48:22.535459864Z', 'message': {'role': 'assistant',
'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i
ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, 'done_reason': 'stop', 'done': True, 'total_durati
on': 68062377612, 'load_duration': 734588, 'prompt_eval_count': 1484, 'prompt_eval_duration': 59473771000,
'eval_count': 42, 'eval_duration': 7978981000}
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=llama3:latest,
options={},
keep_alive=None
Prompt Content:
```

[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n   Get the total number of invoices for each customer\n'\n\nThe DataFrame was produced using this query: SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\n\nThe following 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': 'llama3:latest', 'created_at': '2024-06-13T21:48:39.256069523Z', 'message': {'role': 'assistant', 'content': "```\nimport plotly.express as px\nimport numpy as np\n\nfig = px.bar(df, x='CustomerId', y='TotalInvoices', title='Total Invoices per Customer')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 16693156766, 'load_duration': 755576, 'prompt_eval_count': 189, 'prompt_eval_duration': 7992210000, 'eval_count': 45, 'eval_duration': 8561328000}


Total 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},
```

```
                'template': '...',
                'title': {'text': 'Total 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 9, updating n_results = 9
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE "employees"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each custom

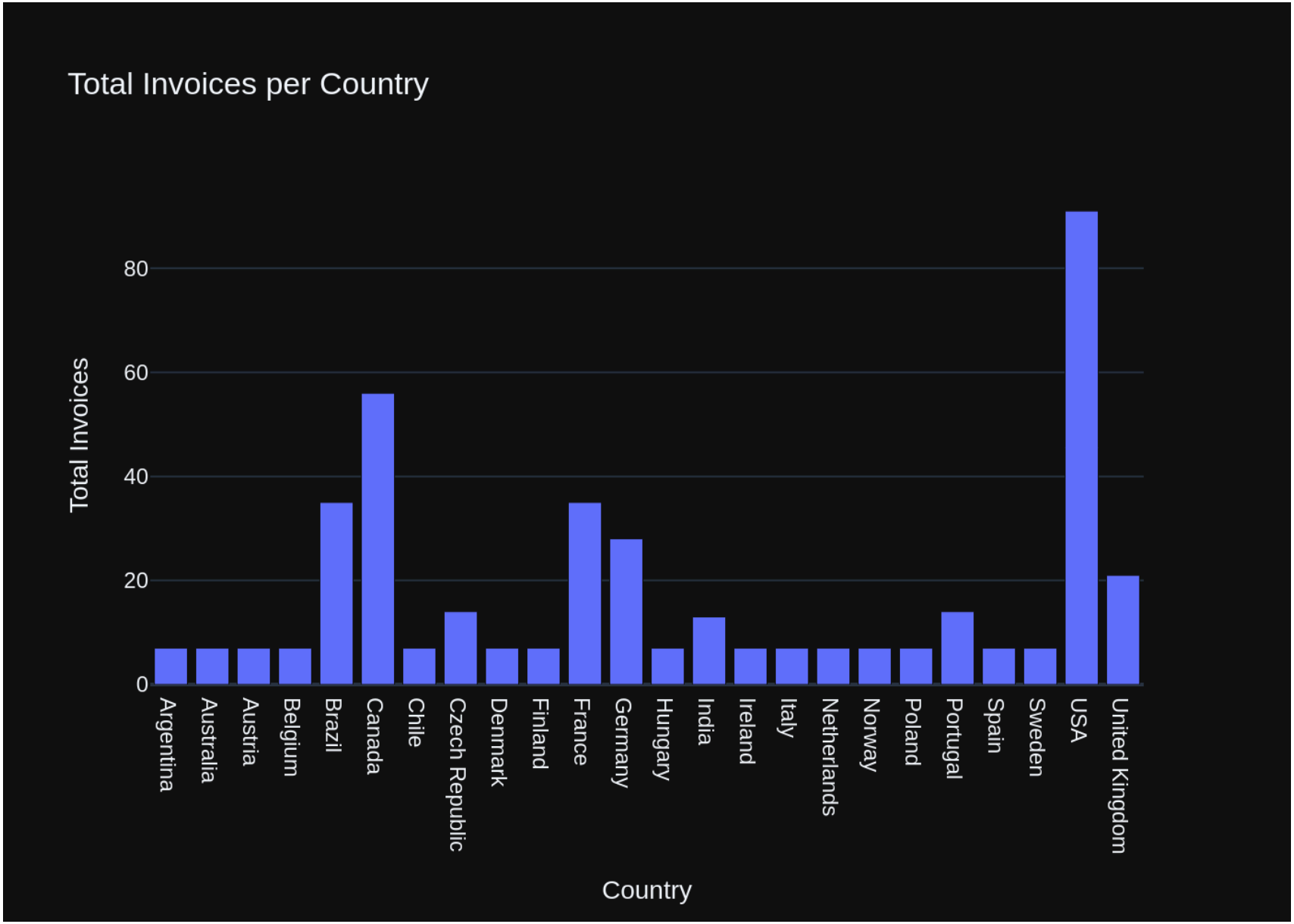
er\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "c
ustomers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'c
ontent': ' \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who
bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELE
CT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.In
voiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'cont
ent': ' \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 "invoice
s" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role':
'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'conten
t': 'SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCu
stomers DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant',
'content': 'SELECT COUNT(*) FROM "customers"'}, {'role': 'user', 'content': ' \n   There are 3 tables: art
ists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by A
lbumId,\n   Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assi
stant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON
a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY 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   Find all tracks wit
h a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM "track
s" \nWHERE LOWER(Name) LIKE \'%what%\''}, {'role': 'user', 'content': 'Can you list all tables in the SQLit
e database catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='tabl
e'"}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and 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   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   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   Find the top 5 customers who spent the most money ov
erall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items deta
il table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalS
pent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDE
R BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": "what are the top 5 countries that customers c
ome from?"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM \"custome

rs\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM \"customers\""}, {"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   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": "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   Find the total number of invoices per country:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:49:52.784456098Z', 'message': {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.BillingCountry'}, 'done_reason': 'stop', 'done': True, 'total_duration': 73427570920, 'load_duration': 885976, 'prompt_eval_count': 1592, 'prompt_eval_duration': 67816490000, 'eval_count': 26, 'eval_duration': 4961355000}
```
SELECT i.BillingCountry, COUNT(*) AS TotalInvoices
FROM "invoices" i
GROUP BY i.BillingCountry
```
```
SELECT i.BillingCountry, COUNT(*) AS TotalInvoices
FROM "invoices" i
GROUP BY i.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=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Find the total number of invoices per country:\n'\n\nThe
DataFrame was produced using this query: SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM \"invoice
s\" i\nGROUP BY i.BillingCountry\n\nThe following is information about the resulting pandas DataFrame 'df':
\nRunning df.dtypes gives:\n BillingCountry    object\nTotalInvoices      int64\ndtype: object"}, {"role":
"user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume t
he data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicato
r. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:50:14.2188506Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\n\nfig = px.bar(df, x='BillingCountry', y='TotalInvoices', tit
le='Total Invoices per Country')\n\nfig.update_layout(xaxis_title='Country',\n                   yaxis_title
='Total Invoices')\n\nif len(df) == 1:\n    fig.update_traces(type='indicator')\n\nfig.show()\n```"}, 'done
_reason': 'stop', 'done': True, 'total_duration': 21406720478, 'load_duration': 41477469, 'prompt_eval_coun
t': 171, 'prompt_eval_duration': 6881867000, 'eval_count': 75, 'eval_duration': 14434480000}
```


Total Invoices per Country
Total Invoices
0
20
40
60
80
Argentina
Australia
Austria
Belgium
Brazil
Canada
Chile
Czech Republic
Denmark
Finland
France
Germany
Hungary
India
Ireland
Italy
Netherlands
Norway
Poland
Portugal
Spain
Sweden
USA
United Kingdom
Country

Out[24]: ('SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.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},
                    'template': '...',
                    'title': {'text': 'Total Invoices per Country'},
                    'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Country'}},
                    'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total Invoices'}}}
          }))
```

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   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 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 Tot
alSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Hint: album quantity is found in invoice_item
s, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoice
s):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFRO
M "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY Tota
lAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each cus
tomer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM
"customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user',
'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'S
ELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role':
'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'conten
t': 'SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCu
stomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   There are 3 tables: artists, albums and track
s, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n   Can you f
ind the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'content': 'SE
LECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistI
d\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'rol
e': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) F
ROM "customers"'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist name
s \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 all tracks wi
th a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM "trac
ks" \nWHERE LOWER(Name) LIKE \'%what%\''}, {'role': 'user', 'content': 'Can you list all tables in the SQLi
te database catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='tabl
e'"}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoice_items\"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY K
EY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   Uni
tPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCE
S \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) RE
FERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_Invo
iceLineInvoiceId ON \"invoice_items\" (InvoiceId)\n\nCREATE TABLE \"invoices\"\r\n(\r\n   InvoiceId INTEGE
R PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT
NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(4
0),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2)
NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION

ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE IN
DEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE TABLE \"tracks\"\r\n(\r\n   TrackId INTEG
ER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n
MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds I
NTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (Album
Id) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (Ge
nreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY
(MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE INDEX IFK_EmployeeReportsTo ON \"employees\" (ReportsTo)\n\nCREATE TABLE \"customers\"\r\n(\r\n
CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   Last
Name NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR
(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone
NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r
\n   FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION\r\n)\n\nCREATE TABLE \"employees\"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT
NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHA
R(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCH
AR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode
NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN
KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\n\n===Additional Context \n\nIn
the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficien
t, please generate a valid SQL query without any explanations for the question. \n2. If the provided contex
t is almost sufficient but requires knowledge of a specific string in a particular column, please generate
an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment say
ing intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generate
d. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, ple
ase repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n   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 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   Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.Cu
stomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId =
i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   Find the total number of invoic
es per country:\n"}, {"role": "assistant", "content": "SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\n
FROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "content": "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", "cont
ent": " \n   There are 3 tables: artists, albums and tracks, where albums and artists are linked by Artist
Id, 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": "How many customers
are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM \"customers\""}, {"role": "user", "cont
ent": " \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   Find all tracks with a name containing \"What\" (case-i
nsensitive)\n"}, {"role": "assistant", "content": "SELECT * \nFROM \"tracks\" \nWHERE LOWER(Name) LIKE '%wh
at%'"}, {"role": "user", "content": "Can you list all tables in the SQLite database catalog?"}, {"role": "a
ssistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "content": "
\n   List all invoices with a total exceeding $10:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:51:23.63424513Z', 'message': {'role': 'assistant',
'content': 'SELECT * \nFROM "invoices" \nWHERE Total > 10.00'}, 'done_reason': 'stop', 'done': True, 'total
_duration': 69312340658, 'load_duration': 1049555, 'prompt_eval_count': 1593, 'prompt_eval_duration': 65683
433000, 'eval_count': 16, 'eval_duration': 2967954000}

```
SELECT *
FROM "invoices"
WHERE Total > 10.00
```

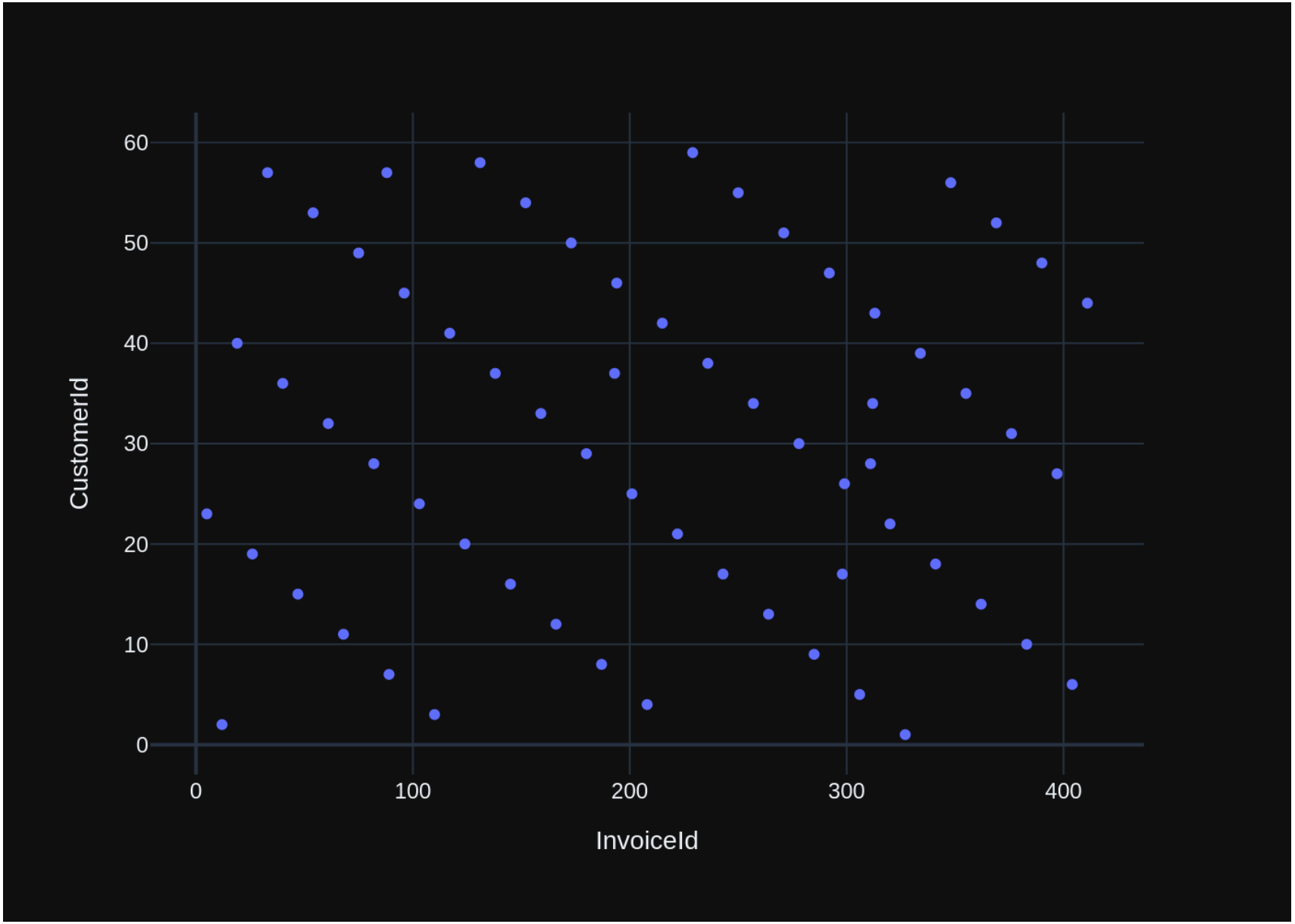
```
SELECT *
FROM "invoices"
WHERE Total > 10.00
```

	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=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all invoices with a total exceeding $10:\n'\n\nThe
DataFrame was produced using this query: SELECT * \nFROM \"invoices\" \nWHERE Total > 10.00\n\nThe followin
g is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n InvoiceId
int64\nCustomerId             int64\nInvoiceDate           object\nBillingAddress       object\nBillingCit
y           object\nBillingState          object\nBillingCountry       object\nBillingPostalCode     objec
t\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 wi
th any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:51:52.265965133Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nimport numpy as np\n\nfig = px.bar(df, x='InvoiceId', y='Tota
l')\nfig.update_layout(title='Invoices with Total Exceeding $10')\n\nif len(df) == 1:\n    fig = px.scatter
(x=['Single Invoice'], y=[df['Total'].values[0]], mode='markers+', \n                     title='Single Inv
oice', labels={'x': 'Invoice', 'y': 'Total'})\nelse:\n    fig.show()\n```"}, 'done_reason': 'stop', 'done':
True, 'total_duration': 28605333929, 'load_duration': 41002943, 'prompt_eval_count': 197, 'prompt_eval_dura
tion': 8172141000, 'eval_count': 105, 'eval_duration': 20343269000}
```


60
50
40
30
20
10
0
CustomerId
0
100
200
300
400
InvoiceId

Out[25]:
```
('SELECT * \nFROM "invoices" \nWHERE Total > 10.00',
    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   List all invoices with a t
otal exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM "invoices" \nWHERE Total > 10.0
0'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'ass
istant', 'content': 'SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.Bill
ingCountry'}, {'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': ' \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total ca
n be found on invoices table, calculation using invoice_items detail table is unnecessary \n'}, {'role': 'a
ssistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices"
i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'use
r', 'content': ' \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customer
s who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content':
'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON
i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user',
'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SEL
ECT c.Country, COUNT(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers
DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'conten
t': 'SELECT COUNT(*) FROM "customers"'}, {'role': 'user', 'content': ' \n   There are 3 tables: artists, a
lbums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumI
d,\n   Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistan
t', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\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   Find all tracks with a name containing "What" (case-inse
nsitive)\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%
\''}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'ro
le': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "art
ists" ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 a
nd the total amount invoiced:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"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

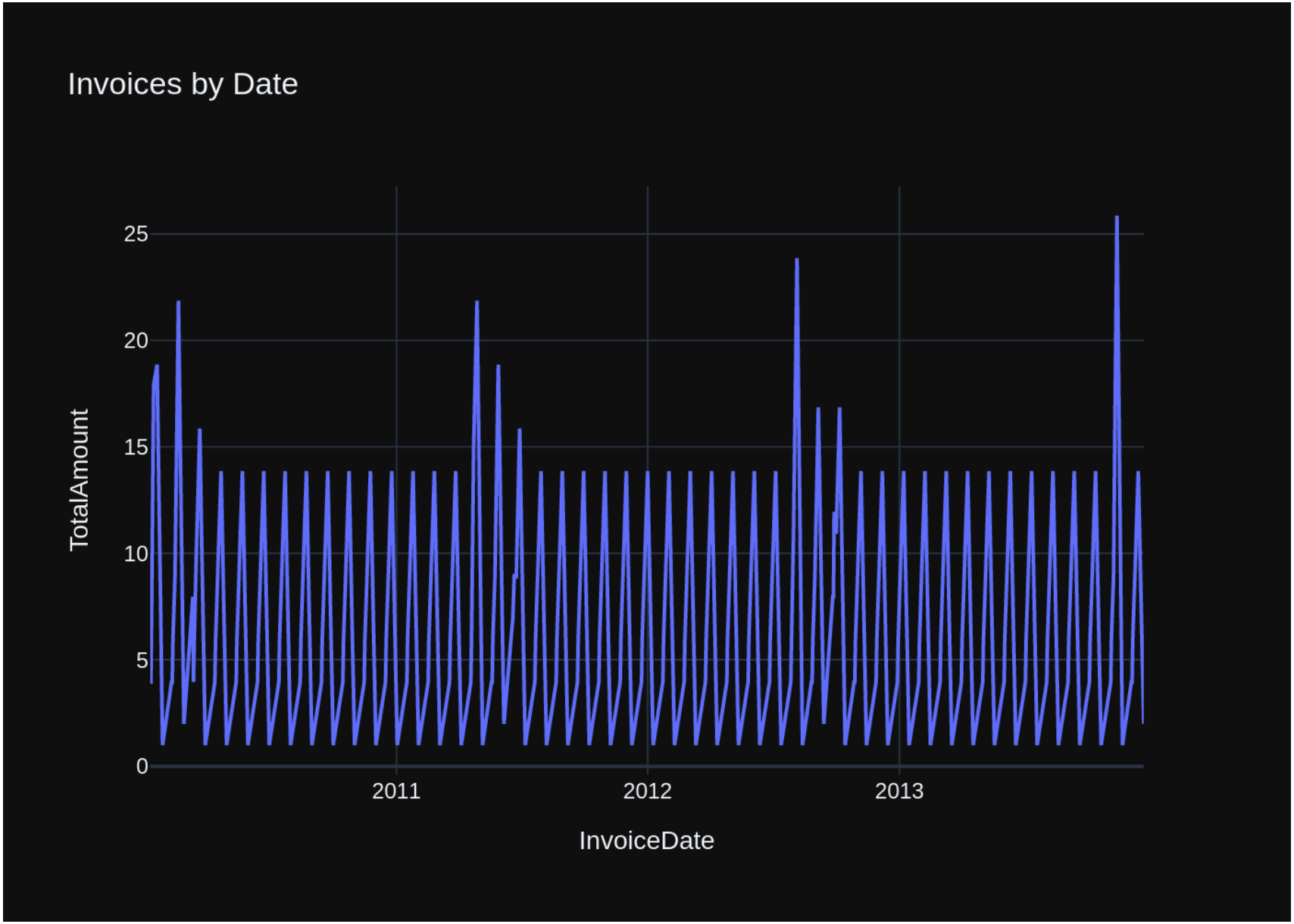
IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) 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   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   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": " \n   Find the top 5 customers who spent the most money ove
rall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detai
l table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSp
ent\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 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": "what are the top 5 countries that customers co
me from?"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM \"customer
s\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "How many c
ustomers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM \"customers\""}, {"role": "use
r", "content": " \n   There are 3 tables: artists, albums and tracks, where albums and artists are linked
by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most popular artists bas
ed on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalT
racks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId
= t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n   F
ind all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SE
LECT * \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.Title, a.Arti
stId, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"role":
"user", "content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:53:12.086616294Z', 'message': {'role': 'assistant',
'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >=
\'2010-01-01\'\nGROUP BY i.InvoiceDate'}, 'done_reason': 'stop', 'done': True, 'total_duration': 7971270294
7, 'load_duration': 839305, 'prompt_eval_count': 1726, 'prompt_eval_duration': 71140462000, 'eval_count': 4
1, 'eval_duration': 7911457000}
```
SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount
FROM "invoices" i
WHERE i.InvoiceDate >= '2010-01-01'
GROUP BY i.InvoiceDate
```
```
SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount
FROM "invoices" i
WHERE i.InvoiceDate >= '2010-01-01'
GROUP BY i.InvoiceDate
```

	InvoiceDate	TotalAmount
0	2010-01-08 00:00:00	3.96
1	2010-01-09 00:00:00	3.96

```
2    2010-01-10 00:00:00         6.94
3    2010-01-13 00:00:00        17.91
4    2010-01-18 00:00:00        18.86
..                   ...          ...
277  2013-12-05 00:00:00         3.96
278  2013-12-06 00:00:00         5.94
279  2013-12-09 00:00:00         8.91
280  2013-12-14 00:00:00        13.86
281  2013-12-22 00:00:00         1.99

[282 rows x 2 columns]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Find all invoices since 2010 and the total amount invoic
ed:\n'\n\nThe DataFrame was produced using this query: SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nF
ROM \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.InvoiceDate\n\nThe following is informa
tion about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n InvoiceDate     object\nTotalA
mount    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 explana
tions -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:53:47.536258015Z', 'message': {'role': 'assistant',
'content': '```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = px.line(df, x=\'In
voiceDate\', y=\'TotalAmount\', title=\'Invoices by Date\')\n\nif len(df) == 1:\n    fig = go.Figure(data=
[go.Indicator(\n        mode="number+delta",\n        value=df[\'TotalAmount\'].values[0],\n        delta=
{\'reference\': df[\'TotalAmount\'].values[0], \'magnitude\': None, \'orientation\': "up"},\n        title=
\'Total Invoices\',\n        domain={\'x\': [0, 1], \'y\': [0.5, 1]}\n    )])\n\nfig.show()\n```'}, 'done_r
eason': 'stop', 'done': True, 'total_duration': 35424102226, 'load_duration': 41387421, 'prompt_eval_coun
t': 190, 'prompt_eval_duration': 7797426000, 'eval_count': 141, 'eval_duration': 27536428000}
```


Invoices by Date
TotalAmount
25
20
15
10
5
0
2011
2012
2013
InvoiceDate

Out[26]:
```
('SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-0
1\'\nGROUP BY i.InvoiceDate',
             InvoiceDate  TotalAmount
 0    2010-01-08 00:00:00         3.96
 1    2010-01-09 00:00:00         3.96
 2    2010-01-10 00:00:00         6.94
 3    2010-01-13 00:00:00        17.91
 4    2010-01-18 00:00:00        18.86
 ..                   ...          ...
 277  2013-12-05 00:00:00         3.96
 278  2013-12-06 00:00:00         5.94
 279  2013-12-09 00:00:00         8.91
 280  2013-12-14 00:00:00        13.86
 281  2013-12-22 00:00:00         1.99

 [282 rows x 2 columns],
 Figure({
     'data': [{'hovertemplate': 'InvoiceDate=%{x}<br>TotalAmount=%{y}<extra></extra>',
               'legendgroup': '',
               'line': {'color': '#636efa', 'dash': 'solid'},
               'marker': {'symbol': 'circle'},
               'mode': 'lines',
               'name': '',
               'orientation': 'v',
               'showlegend': False,
               'type': 'scatter',
               'x': array(['2010-01-08 00:00:00', '2010-01-09 00:00:00', '2010-01-10 00:00:00',
                           ..., '2013-12-09 00:00:00', '2013-12-14 00:00:00',
                           '2013-12-22 00:00:00'], dtype=object),
               'xaxis': 'x',
               'y': array([ 3.96,  3.96,  6.94, ...,  8.91, 13.86,  1.99]),
               'yaxis': 'y'}],
     'layout': {'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Invoices by Date'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceDate'}},
                '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   Find th

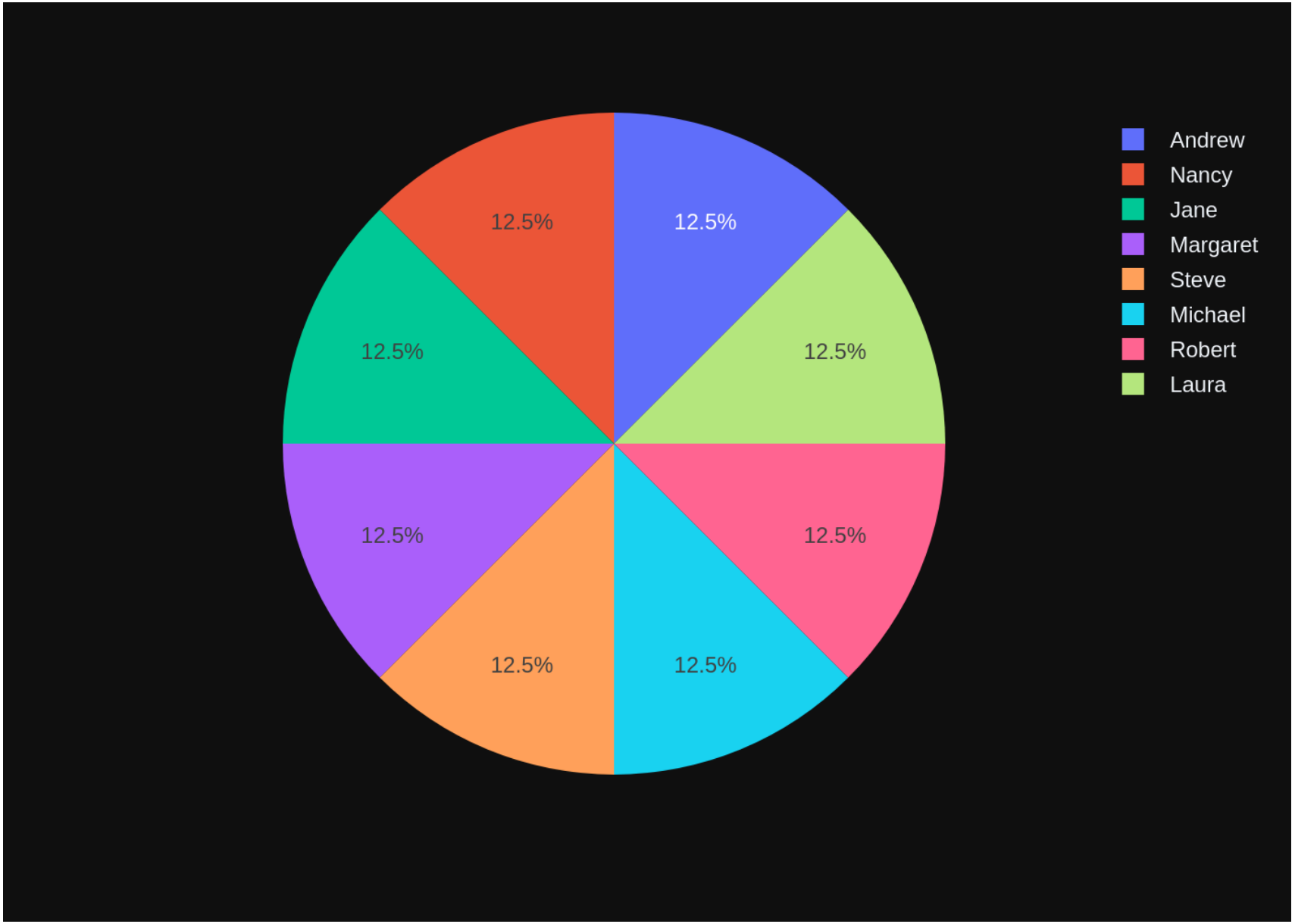
e top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoice
s table, calculation using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content':
'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId =
i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': 'what
are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT c.Country, CO
UNT(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5'},
{'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assi
stant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "inv
oices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n   Find
all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.Invo
iceDate, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.
InvoiceDate'}, {'role': 'user', 'content': ' \n   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   Find the total numb
er of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(*) AS Tot
alInvoices\nFROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': ' \n   Hint: albu
m quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in to
tal quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.In
voiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROU
P BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   There are 3 ta
bles: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are li
nked by AlbumId,\n   Can you find the top 10 most popular artists based on the number of tracks\n'}, {'rol
e': 'assistant', 'content': 'SELECT a.Name, 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.Name\nORDER BY Tot
alTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $1
0:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM "invoices" \nWHERE Total > 10.00'}, {'role': 'use
r', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM "cus
tomers"'}, {'role': 'user', 'content': " \n   List all employees and their reporting manager's name (if a
ny):\n"}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_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   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be
found on invoices table, calculation using invoice_items detail table is unnecessary \n"}, {"role": "assist
ant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\"
i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "use
r", "content": "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 TotalCust
omers DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Get the total number of invoices for each custo
mer\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 all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistan
t", "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   List all albums and their c
orresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name AS A
rtistName\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "conten
t": " \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": "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
\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role":
"user", "content": " \n   There are 3 tables: artists, albums and tracks, where albums and artists are lin
ked 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 Tot
alTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.Album
Id = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n
List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM \"invo
ices\" \nWHERE Total > 10.00"}, {"role": "user", "content": "How many customers are there"}, {"role": "assi
stant", "content": "SELECT COUNT(*) FROM \"customers\""}, {"role": "user", "content": " \n   List all emp
loyees and their reporting manager's name (if any):\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:55:07.388099702Z', 'message': {'role': 'assistant',
'content': 'SELECT e.FirstName, e.LastName, mt.FirstName AS ManagerFirstName, mt.LastName AS ManagerLastNam
e\nFROM "employees" e\nLEFT JOIN "employees" mt ON e.ReportsTo = mt.EmployeeId'}, 'done_reason': 'stop', 'd
one': True, 'total_duration': 79745159239, 'load_duration': 803509, 'prompt_eval_count': 1673, 'prompt_eval
_duration': 71128496000, 'eval_count': 41, 'eval_duration': 7868682000}
SELECT e.FirstName, e.LastName, mt.FirstName AS ManagerFirstName, mt.LastName AS ManagerLastName
FROM "employees" e
LEFT JOIN "employees" mt ON e.ReportsTo = mt.EmployeeId
SELECT e.FirstName, e.LastName, mt.FirstName AS ManagerFirstName, mt.LastName AS ManagerLastName
FROM "employees" e
LEFT JOIN "employees" mt ON e.ReportsTo = mt.EmployeeId

	FirstName	LastName	ManagerFirstName	ManagerLastName
0	Andrew	Adams	None	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=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all employees and their reporting manager's name (i
f any):\n'\n\nThe DataFrame was produced using this query: SELECT e.FirstName, e.LastName, mt.FirstName AS
ManagerFirstName, mt.LastName AS ManagerLastName\nFROM \"employees\" e\nLEFT JOIN \"employees\" mt ON e.Rep
ortsTo = mt.EmployeeId\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning
df.dtypes gives:\n FirstName           object\nLastName            object\nManagerFirstName    object\nMana
gerLastName     object\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': 'llama3:latest', 'created_at': '2024-06-13T21:55:43.419138296Z', 'message': {'role': 'assistant',
'content': '```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = go.Figure()\n\nif
df.shape[0] == 1:\n    fig.add_trace(go.Indicator(\n        name="Manager",\n        domain={\'x\': [0, 1],
\'y\': [0, 1]},\n        value=df[\'ManagerFirstName\'].iloc[0],\n        number=dict/font=dict(size=30),\n
))\nelse:\n    fig = px.scatter(df, x=\'LastName\', y=\'ManagerFirstName\', hover_name=\'LastName\')\n\nfi
g.update_layout(title="Employees and their Reporting Manager", xaxis_title="Employee Name", yaxis_title="Re
porting Manager")\n\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 35998317329,
'load_duration': 746704, 'prompt_eval_count': 196, 'prompt_eval_duration': 8202849000, 'eval_count': 142,
'eval_duration': 27653759000}
```


12.5%
12.5%
12.5%
12.5%
12.5%
12.5%
12.5%
Andrew
Nancy
Jane
Margaret
Steve
Michael
Robert
Laura

Out[27]:
```
('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',
   FirstName  LastName ManagerFirstName ManagerLastName
 0    Andrew     Adams             None             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': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
               'hovertemplate': 'FirstName=%{label}<extra></extra>',
               'labels': array(['Andrew', 'Nancy', 'Jane', 'Margaret', 'Steve', 'Michael', 'Robert',
                                'Laura'], dtype=object),
               'legendgroup': '',
               'name': '',
               'showlegend': True,
               'type': 'pie'}],
     'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'template': '...'}
 }))
```

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 total number of invoices for each customer \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find 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.InvoiceDa

te >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': ' \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM "invoices" \nWHERE Total > 10.00'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM "customers"'}, {'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 employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': '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   Get the average invoice total for each customer:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE 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   LastNa

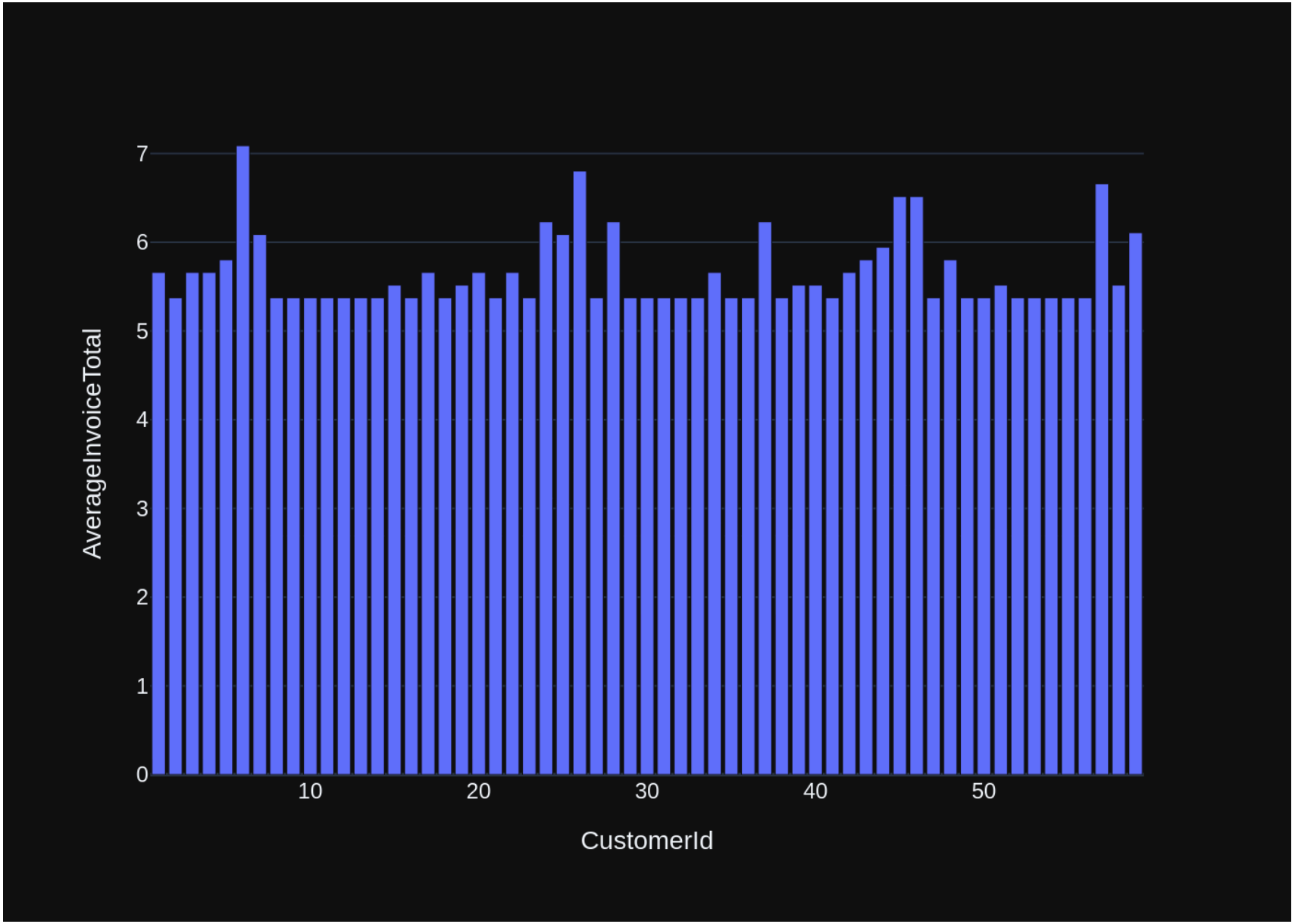
me NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r\n   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 total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "content": " \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM \"invoices\" \nWHERE Total > 10.00"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM \"customers\""}, {"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": "u
ser", "content": " \n   List all employees and their reporting manager's name (if any):\n"}, {"role": "as
sistant", "content": "SELECT e.FirstName, e.LastName, mt.FirstName AS ManagerFirstName, mt.LastName AS Mana
gerLastName\nFROM \"employees\" e\nLEFT JOIN \"employees\" mt ON e.ReportsTo = mt.EmployeeId"}, {"role": "u
ser", "content": " \n   Get the average invoice total for each customer:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:56:55.775823437Z', 'message': {'role': 'assistant',
'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i
ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, 'done_reason': 'stop', 'done': True, 'total_durati
on': 72256030409, 'load_duration': 879340, 'prompt_eval_count': 1480, 'prompt_eval_duration': 63769109000,
'eval_count': 40, 'eval_duration': 7716006000}
```
SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
```
```
SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
```

	CustomerId	AverageInvoiceTotal
0	1	5.660000
1	2	5.374286
2	3	5.660000
3	4	5.660000
4	5	5.802857
5	6	7.088571
6	7	6.088571
7	8	5.374286
8	9	5.374286
9	10	5.374286
10	11	5.374286
11	12	5.374286
12	13	5.374286
13	14	5.374286
14	15	5.517143
15	16	5.374286
16	17	5.660000
17	18	5.374286
18	19	5.517143
19	20	5.660000
20	21	5.374286
21	22	5.660000

```
22          23          5.374286
23          24          6.231429
24          25          6.088571
25          26          6.802857
26          27          5.374286
27          28          6.231429
28          29          5.374286
29          30          5.374286
30          31          5.374286
31          32          5.374286
32          33          5.374286
33          34          5.660000
34          35          5.374286
35          36          5.374286
36          37          6.231429
37          38          5.374286
38          39          5.517143
39          40          5.517143
40          41          5.374286
41          42          5.660000
42          43          5.802857
43          44          5.945714
44          45          6.517143
45          46          6.517143
46          47          5.374286
47          48          5.802857
48          49          5.374286
49          50          5.374286
50          51          5.517143
51          52          5.374286
52          53          5.374286
53          54          5.374286
54          55          5.374286
55          56          5.374286
56          57          6.660000
57          58          5.517143
58          59          6.106667
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
```

[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n   Get the average invoice total for each customer:\n'\n\nThe DataFrame was produced using this query: SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId   int64\nAverageInvoiceTotal   float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:57:27.767012663Z', 'message': {'role': 'assistant', 'content': '```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = px.bar(df, x=\'CustomerId\', y=\'AverageInvoiceTotal\', barmode=\'group\')\n\n# if there\'s only one value in the dataframe\nif len(df) == 1:\n   fig = go.Figure(data=[go.Indicator(\n   mode="number",\n   value=df[\'AverageInvoiceTotal\'].values[0],\n   title={\'text\': \'Average Invoice Total\'},\n   domain={\'x\': [0, .9], \'y\': [0, .7]}\n   )])\n\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 31963063703, 'load_duration': 777949, 'prompt_eval_count': 186, 'prompt_eval_duration': 7392452000, 'eval_count': 126, 'eval_duration': 24435825000}


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',

	CustomerId	AverageInvoiceTotal
0	1	5.660000
1	2	5.374286
2	3	5.660000
3	4	5.660000
4	5	5.802857
5	6	7.088571
6	7	6.088571
7	8	5.374286
8	9	5.374286
9	10	5.374286
10	11	5.374286
11	12	5.374286
12	13	5.374286
13	14	5.374286
14	15	5.517143
15	16	5.374286
16	17	5.660000
17	18	5.374286
18	19	5.517143
19	20	5.660000
20	21	5.374286
21	22	5.660000
22	23	5.374286
23	24	6.231429
24	25	6.088571
25	26	6.802857
26	27	5.374286
27	28	6.231429
28	29	5.374286
29	30	5.374286
30	31	5.374286
31	32	5.374286
32	33	5.374286
33	34	5.660000
34	35	5.374286
35	36	5.374286
36	37	6.231429
37	38	5.374286
38	39	5.517143

```
39          40            5.517143
40          41            5.374286
41          42            5.660000
42          43            5.802857
43          44            5.945714
44          45            6.517143
45          46            6.517143
46          47            5.374286
47          48            5.802857
48          49            5.374286
49          50            5.374286
50          51            5.517143
51          52            5.374286
52          53            5.374286
53          54            5.374286
54          55            5.374286
55          56            5.374286
56          57            6.660000
57          58            5.517143
58          59            6.106667,
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([ 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([5.66      , 5.37428571, 5.66      , 5.66      , 5.80285714, 7.08857143,
                          6.08857143, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571,
                          5.37428571, 5.37428571, 5.51714286, 5.37428571, 5.66      , 5.37428571,
                          5.51714286, 5.66      , 5.37428571, 5.66      , 5.37428571, 6.23142857,
                          6.08857143, 6.80285714, 5.37428571, 6.23142857, 5.37428571, 5.37428571,
                          5.37428571, 5.37428571, 5.37428571, 5.66      , 5.37428571, 5.37428571,
```

```
                            6.23142857, 5.37428571, 5.51714286, 5.51714286, 5.37428571, 5.66      ,
                            5.80285714, 5.94571429, 6.51714286, 6.51714286, 5.37428571, 5.80285714,
                            5.37428571, 5.37428571, 5.51714286, 5.37428571, 5.37428571, 5.37428571,
                            5.37428571, 5.37428571, 6.66      , 5.51714286, 6.10666667]),
                 'yaxis': 'y'}],
      'layout': {'barmode': 'group',
                 'legend': {'tracegroupgap': 0},
                 'margin': {'t': 60},
                 'template': '...',
                 'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                 'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'AverageInvoiceTotal'}}}
 }))
```

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

vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "albums"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   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 i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.C

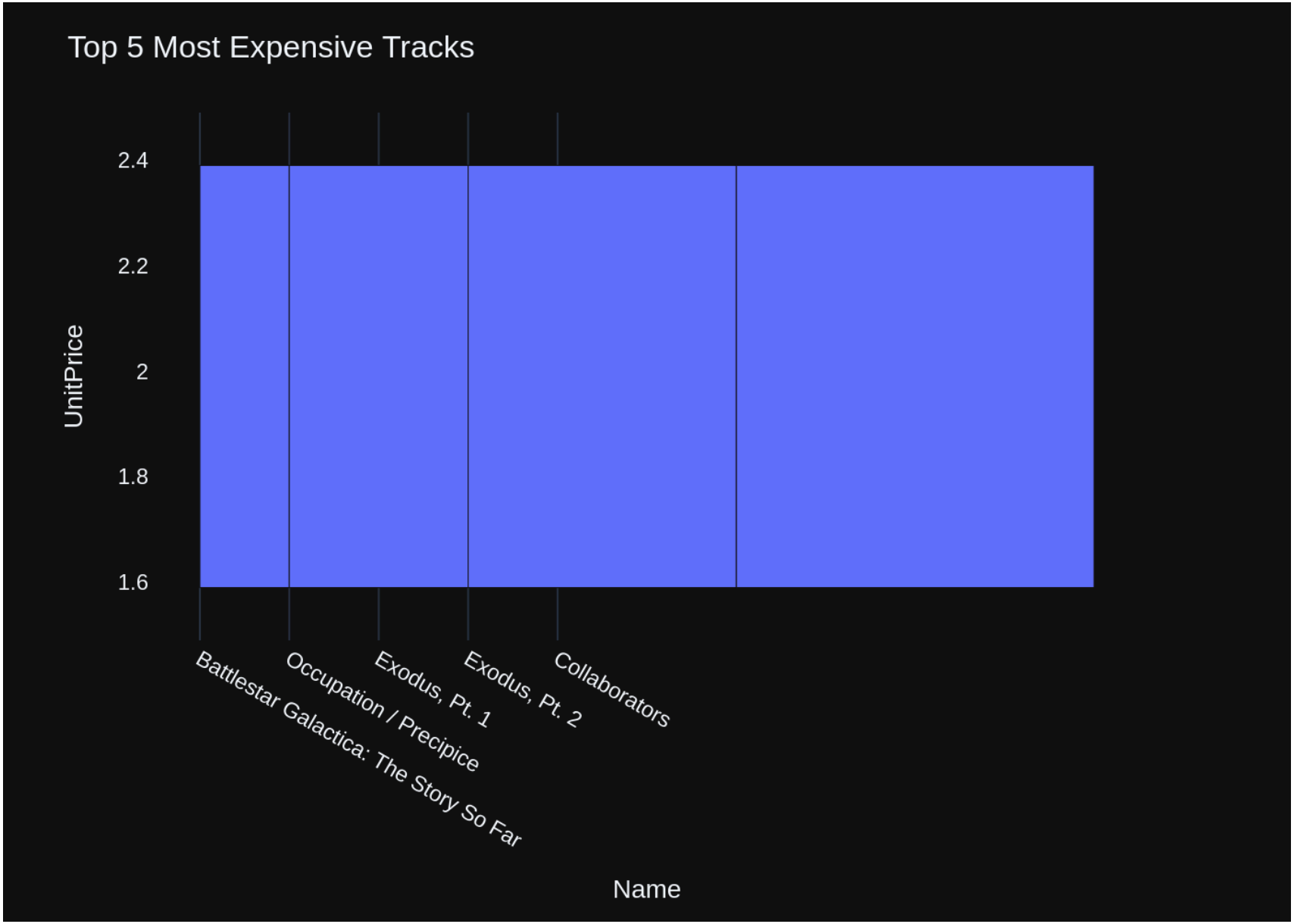
ustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n
Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': 'SEL
ECT * \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%\''}, {'role': 'user', 'content': ' \n   List all
invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM "invoices" \nWH
ERE Total > 10.00'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist na
mes \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "album
s" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': 'what are the top 5 coun
tries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(*) AS TotalCus
tomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'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.Custo
merId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n   Find all invoices since
2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM(i.Tota
l) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'ro
le': '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.BillingCountr
y'}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"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   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   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   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 \"c
ustomers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpen
t DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find all tracks with a name containing \"What\" (ca
se-insensitive)\n"}, {"role": "assistant", "content": "SELECT * \nFROM \"tracks\" \nWHERE LOWER(Name) LIKE
'%what%'"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"rol
e": "assistant", "content": "SELECT * \nFROM \"invoices\" \nWHERE Total > 10.00"}, {"role": "user", "conten
t": " \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": "what are the top 5 countries that customers come from?"}, {"rol
e": "assistant", "content": "SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM \"customers\" c\nGROUP BY
c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n   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   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 i.BillingCount
ry, COUNT(*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "conten
t": " \n   Find the top 5 most expensive tracks (based on unit price):\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:58:34.754204557Z', 'message': {'role': 'assistant',

'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5'},
'done_reason': 'stop', 'done': True, 'total_duration': 66890042775, 'load_duration': 920604, 'prompt_eval_c
ount': 1416, 'prompt_eval_duration': 60603826000, 'eval_count': 29, 'eval_duration': 5539669000}
```
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=llama3:latest,
options={},
keep_alive=None
Prompt Content:
```
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: ' \n   Find the top 5 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\nThe following is information about the resulting pandas DataFra
me 'df': \nRunning df.dtypes gives:\n TrackId   int64\nName   object\nUnitPrice   float64\ndty
pe: 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': 'llama3:latest', 'created_at': '2024-06-13T21:58:52.301581731Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nimport numpy as np\n\nfig = px.bar(df, x='Name', y='UnitPric
e', orientation='h')\n\nfig.update_layout(title_text='Top 5 Most Expensive Tracks')\nfig.show()\n```"}, 'do
ne_reason': 'stop', 'done': True, 'total_duration': 17527206852, 'load_duration': 736742, 'prompt_eval_coun
t': 183, 'prompt_eval_duration': 7315916000, 'eval_count': 53, 'eval_duration': 10074968000}


Top 5 Most Expensive Tracks
UnitPrice
2.4
2.2
2
1.8
1.6
Battlestar Galactica: The Story So Far
Occupation / Precipice
Exodus, Pt. 1
Exodus, Pt. 2
Collaborators
Name

Out[29]:
```
('SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 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,
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'Name=%{x}<br>UnitPrice=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'h',
              'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
              'x': array(['Battlestar Galactica: The Story So Far', 'Occupation / Precipice',
                          'Exodus, Pt. 1', 'Exodus, Pt. 2', 'Collaborators'], dtype=object),
              'xaxis': 'x',
              'y': array([1.99, 1.99, 1.99, 1.99, 1.99]),
              'yaxis': 'y'}],
    'layout': {'barmode': 'relative',
               'legend': {'tracegroupgap': 0},
               'margin': {'t': 60},
               'template': '...',
               'title': {'text': 'Top 5 Most Expensive Tracks'},
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'UnitPrice'}}}
})
```

In [30]:
```
question = """
    List all genres and the number of tracks in each genre:
"""

vn.ask(question=question)
```

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

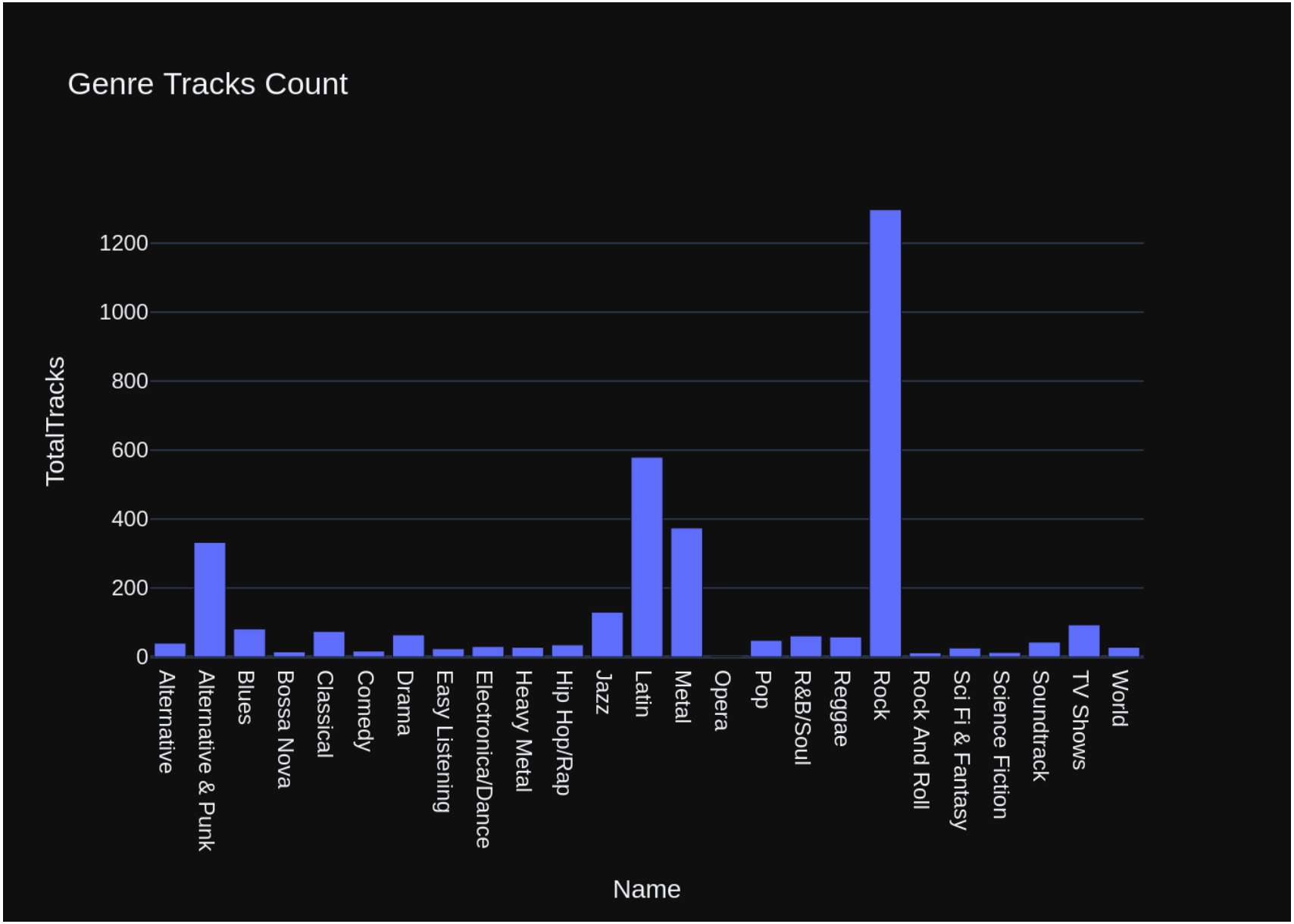
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE TABLE "genres"\r\n(\r\n   GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "playlists"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE TABLE "playlist_track"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n   Find the 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   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   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': 'what are the top 5 countries that customers come from?'}, {'rol
e': 'assistant', 'content': 'SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.C
ountry\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the total number
of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(*) AS TotalI
nvoices\nFROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': ' \n   Find the top
5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices tabl
e, calculation using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELE
CT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Cust
omerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Fi
nd all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.I
nvoiceDate, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY
i.InvoiceDate'}, {'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   List all genres and the number of tracks in each genre:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and 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   There are 3 tables: artists, album
s and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n
Can you find the top 10 most popular artists based on the number of tracks\n"}, {"role": "assistant", "cont
ent": "SELECT a.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\nLIM
IT 10"}, {"role": "user", "content": " \n   Find the top 5 most expensive tracks (based on unit pric
e):\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   List all albums and their correspondin
g artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\n
FROM \"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 al
bums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, C
OUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.I
nvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n
Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "S
ELECT * \nFROM \"tracks\" \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": "what are the to
p 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 total number of invoices per country:\n"}, {"role": "assistant", "cont
ent": "SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountr
y"}, {"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 \"c
ustomers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpen
t DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amount i
nvoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"i
nvoices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"role": "user", "content": "Ca
n you list all tables in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM
sqlite_master WHERE type='table'"}, {"role": "user", "content": " \n   List all genres and the number of
tracks in each genre:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T21:59:56.139357937Z', 'message': {'role': 'assistant',
'content': 'SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId =
t.GenreId\nGROUP BY g.Name'}, 'done_reason': 'stop', 'done': True, 'total_duration': 63741058921, 'load_dur
ation': 801770, 'prompt_eval_count': 1326, 'prompt_eval_duration': 55213214000, 'eval_count': 41, 'eval_dur
ation': 7783323000}
SELECT g.Name, COUNT(t.GenreId) AS TotalTracks
FROM "genres" g

```
JOIN "tracks" t ON g.GenreId = t.GenreId
GROUP BY g.Name
SELECT g.Name, COUNT(t.GenreId) AS TotalTracks
FROM "genres" g
JOIN "tracks" t ON g.GenreId = t.GenreId
GROUP BY g.Name
                   Name  TotalTracks
0          Alternative           40
1   Alternative & Punk          332
2                Blues           81
3           Bossa Nova           15
4            Classical           74
5               Comedy           17
6                Drama           64
7       Easy Listening           24
8    Electronica/Dance           30
9          Heavy Metal           28
10         Hip Hop/Rap           35
11                Jazz          130
12               Latin          579
13               Metal          374
14               Opera            1
15                 Pop           48
16            R&B/Soul           61
17              Reggae           58
18                Rock         1297
19       Rock And Roll           12
20    Sci Fi & Fantasy           26
21     Science Fiction           13
22          Soundtrack           43
23            TV Shows           93
24               World           28
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all genres and the number of tracks in each 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\n\nThe following is information a
bout the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name           object\nTotalTracks
```

int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the r
esults of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value i
n the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- ju
st the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:00:23.134533975Z', 'message': {'role': 'assistant',
'content': '```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = px.bar(df, x=\'Nam
e\', y=\'TotalTracks\', title=\'Genre Tracks Count\')\n\nif df.shape[0] == 1:\n   fig = go.Figure(data=[g
o.Indicator(\n   mode="number+delta",\n   value=df[\'TotalTracks\'].values[0],\n   delta=
{\'reference\': None, \'valueChange\': None}\n   )])\n\nfig.show()\n```'}, 'done_reason': 'stop', 'done':
True, 'total_duration': 26969039815, 'load_duration': 742008, 'prompt_eval_count': 187, 'prompt_eval_durati
on': 7449519000, 'eval_count': 100, 'eval_duration': 19385196000}


Genre Tracks Count
TotalTracks
1200
1000
800
600
400
200
0
Alternative
Alternative & Punk
Blues
Bossa Nova
Classical
Comedy
Drama
Easy Listening
Electronica/Dance
Heavy Metal
Hip Hop/Rap
Jazz
Latin
Metal
Opera
Pop
R&B/Soul
Reggae
Rock
Rock And Roll
Sci Fi & Fantasy
Science Fiction
Soundtrack
TV Shows
World
Name

Out[30]: ('SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.Name',

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

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

```
                            'Reggae', 'Rock', 'Rock And Roll', 'Sci Fi & Fantasy',
                            'Science Fiction', 'Soundtrack', 'TV Shows', 'World'], dtype=object),
                'xaxis': 'x',
                'y': array([  40,  332,   81,   15,   74,   17,   64,   24,   30,   28,   35,  130,
                             579,  374,    1,   48,   61,   58, 1297,   12,   26,   13,   43,   93,
                              28]),
                'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Genre Tracks Count'},
                '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   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 genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.Name'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%\''}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names\n'}, {'role': 'assistant', '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   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most album

s in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUN
T(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceI
d\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': 'Can you list a
ll tables in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_mast
er WHERE type='table'"}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total am
ount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFR
OM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'role': 'user', 'conten
t': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT c.C
ountry, COUNT(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nL
IMIT 5'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role':
'assistant', 'content': 'SELECT * \nFROM "invoices" \nWHERE Total > 10.00'}, {'role': 'user', 'content': '
\n   Get all genres that do not have any tracks associated with them:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and 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   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   List all genres and the number of tracks in each
genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres
\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"role": "user", "content": " \n   F
ind all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SE
LECT * \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.Title, a.Arti
stId, 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": "assi
stant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLI
MIT 5"}, {"role": "user", "content": " \n   Hint: album quantity is found in invoice_items, \n   \n   F
ind the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role":
"assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\n
JOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\n
LIMIT 5"}, {"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   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SE
LECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\n
GROUP BY i.InvoiceDate"}, {"role": "user", "content": "what are the top 5 countries that customers come fro
m?"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM \"customers\" c
\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n   List al
l invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM \"invoices\"
\nWHERE Total > 10.00"}, {"role": "user", "content": " \n   Get all genres that do not have any tracks as
sociated with them:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:01:24.568855172Z', 'message': {'role': 'assistant',
'content': 'SELECT g.Name\nFROM "genres" g\nLEFT JOIN "tracks" t ON g.GenreId = t.GenreId\nWHERE t.TrackId
IS NULL'}, 'done_reason': 'stop', 'done': True, 'total_duration': 61331102311, 'load_duration': 767435, 'pr
ompt_eval_count': 1284, 'prompt_eval_duration': 54086700000, 'eval_count': 34, 'eval_duration': 6483725000}
SELECT g.Name
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
SELECT g.Name
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId

```
WHERE t.TrackId IS NULL
Empty DataFrame
Columns: [Name]
Index: []
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Get all genres that do not have any tracks associated wi
th them:\n'\n\nThe DataFrame was produced using this query: SELECT g.Name\nFROM \"genres\" g\nLEFT JOIN \"t
racks\" t ON g.GenreId = t.GenreId\nWHERE t.TrackId IS NULL\n\nThe following is information about the resul
ting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name    object\ndtype: object"}, {"role": "user",
"content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data
is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respon
d with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:01:39.048098367Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nfig = px.bar(df, x='Name', y='')\nfig.update_layout(title_tex
t='Genres Without Tracks')\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 14476
753810, 'load_duration': 748437, 'prompt_eval_count': 175, 'prompt_eval_duration': 7249050000, 'eval_coun
t': 37, 'eval_duration': 7088598000}
```

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

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

vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "employees"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "playlists"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepe

nd the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please ex
plain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been
asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user',
'content': ' \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order t
otal can be found on invoices table, calculation using invoice_items detail table is unnecessary \n'}, {'ro
le': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "in
voices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'rol
e': 'user', 'content': ' \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5
customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'c
ontent': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_ite
ms" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role':
'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'conten
t': 'SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCu
stomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each cus
tomer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM
"customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user',
'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM "custome
rs"'}, {'role': 'user', 'content': ' \n   Get the average invoice total for each customer:\n'}, {'role':
'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN
"invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n
Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT
i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP
BY i.InvoiceDate'}, {'role': 'user', 'content': ' \n   Find the 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   List all invoices with a total exce
eding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM "invoices" \nWHERE Total > 10.00'}, {'rol
e': 'user', 'content': " \n   List all employees and their reporting manager's name (if any):\n"}, {'rol
e': 'assistant', 'content': '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 customers who have not placed any orders:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and 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   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. 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.Custome

rId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM \"customers\""}, {"role": "user", "content": " \n   Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM \"invoices\" \nWHERE Total > 10.00"}, {"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.LastName AS ManagerLastName\nFROM \"employees\" e\nLEFT JOIN \"employees\" mt ON e.ReportsTo = mt.EmployeeId"}, {"role": "user", "content": " \n   List all customers who have not placed any orders:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:02:58.147507657Z', 'message': {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.FirstName, c.LastName\nFROM "customers" c\nWHERE c.CustomerId NOT IN (SELECT i.CustomerId FROM "invoices" i)'}, 'done_reason': 'stop', 'done': True, 'total_duration': 79015867862, 'load_duration': 1472353, 'prompt_eval_count': 1694, 'prompt_eval_duration': 71487248000, 'eval_count': 35, 'eval_duration': 6793171000}
SELECT c.CustomerId, c.FirstName, c.LastName
FROM "customers" c
WHERE c.CustomerId NOT IN (SELECT i.CustomerId FROM "invoices" i)
SELECT c.CustomerId, c.FirstName, c.LastName
FROM "customers" c
WHERE c.CustomerId NOT IN (SELECT i.CustomerId FROM "invoices" i)
Empty DataFrame
Columns: [CustomerId, FirstName, LastName]
Index: []
Ollama parameters:
model=llama3:latest,
options={},

```
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all customers who have not placed any orders:\n'\n
\nThe DataFrame was produced using this query: SELECT c.CustomerId, c.FirstName, c.LastName\nFROM \"custome
rs\" c\nWHERE c.CustomerId NOT IN (SELECT i.CustomerId FROM \"invoices\" i)\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 generate the Python plo
tly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If th
ere is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with
any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:03:26.444555073Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nimport numpy as np\n\nfig = px.bar(df, x='CustomerId', y='Las
tName')\nfig.update_layout(title='Customers who have not placed orders')\n\n# If there is only one value in
the dataframe\nif df.shape[0] == 1:\n    fig = px.bar(x=['Only One'], y=[df['LastName'].values[0]])\n    fi
g.update_layout(title='Customer who has not placed an order')\n```"}, 'done_reason': 'stop', 'done': True,
'total_duration': 28294560716, 'load_duration': 784857, 'prompt_eval_count': 181, 'prompt_eval_duration': 8
014616000, 'eval_count': 99, 'eval_duration': 20131321000}
```


Customers who have not placed orders
LastName
CustomerId
4
3
2
1
0
−1
−1
0
1
2
3
4
5
6

Out[32]:
```
('SELECT c.CustomerId, c.FirstName, c.LastName\nFROM "customers" c\nWHERE c.CustomerId NOT IN (SELECT i.Cu
stomerId FROM "invoices" i)',
 Empty DataFrame
 Columns: [CustomerId, FirstName, LastName]
 Index: [],
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'CustomerId=%{x}<br>LastName=%{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': 'Customers who have not placed orders'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'LastName'}}}
 }))
```

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   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.Name'}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'},

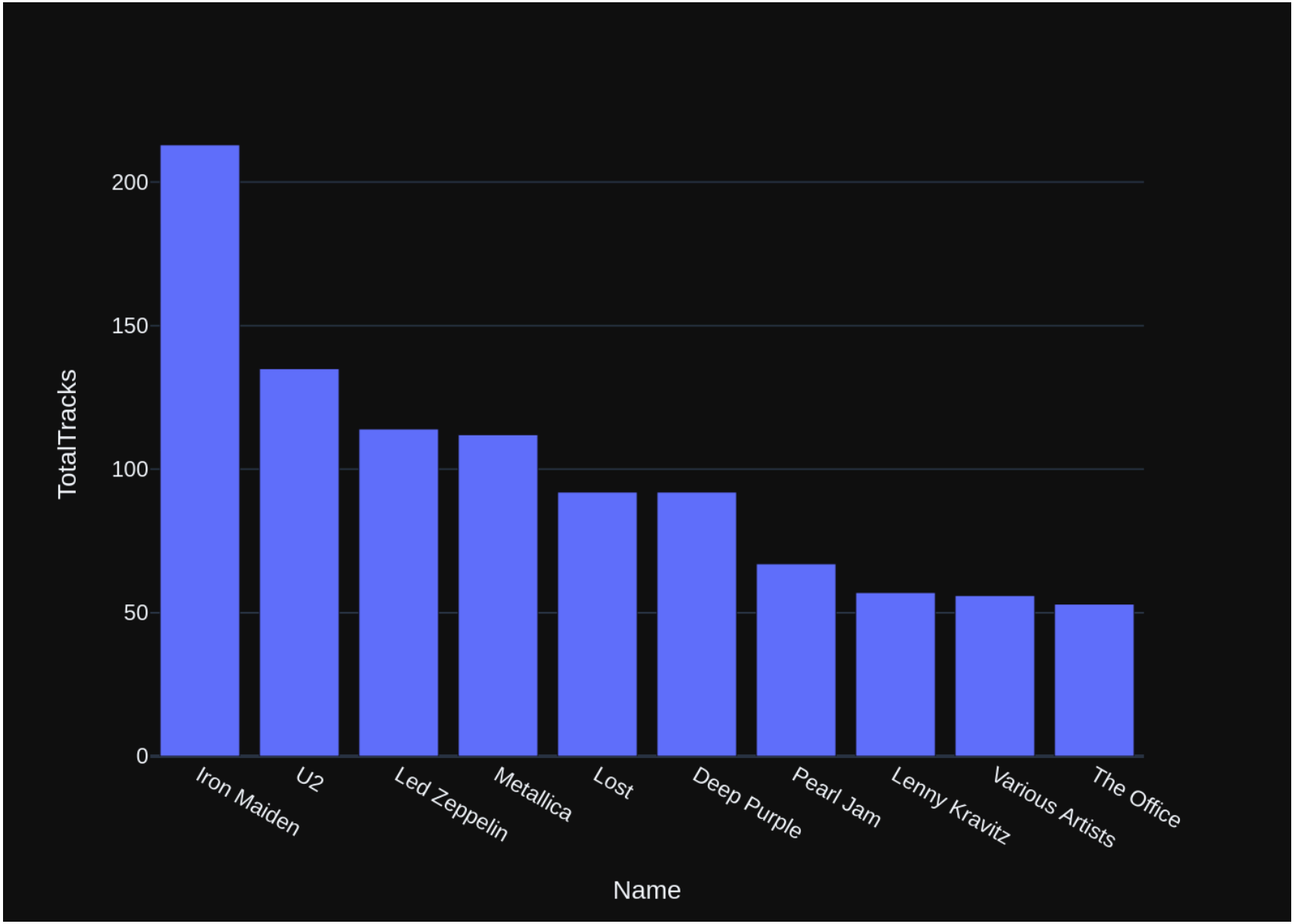
{'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items 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': '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': '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   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 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   There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most popular artists based on the number of tracks\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE \"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 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 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   List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n   List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"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   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": "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": "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   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 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   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"}]

Insert of existing embedding ID: 127fd4bd-b9af-539d-9313-1d0234d073b7-sql
Add of existing embedding ID: 127fd4bd-b9af-539d-9313-1d0234d073b7-sql

```
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:04:41.32955456Z', 'message': {'role': 'assistant',
'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistI
d = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIM
IT 10'}, 'done_reason': 'stop', 'done': True, 'total_duration': 74788066302, 'load_duration': 771741, 'prom
pt_eval_count': 1387, 'prompt_eval_duration': 60990363000, 'eval_count': 66, 'eval_duration': 13036610000}
SELECT a.Name, 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.Name
ORDER BY TotalTracks DESC
LIMIT 10
SELECT a.Name, 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.Name
ORDER BY TotalTracks DESC
LIMIT 10
              Name  TotalTracks
0      Iron Maiden          213
1               U2          135
2     Led Zeppelin          114
3        Metallica          112
4             Lost           92
5      Deep Purple           92
6        Pearl Jam           67
7    Lenny Kravitz           57
8  Various Artists           56
9       The Office           53
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: ' \n    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.Name, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.Art
istId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10\n
```

\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name object\nTotalTracks   int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:05:02.062343509Z', 'message': {'role': 'assistant', 'content': "```\nimport plotly.express as px\nimport numpy as np\n\nfig = px.bar(df, x='Name', y='TotalTracks', title='Top 10 Most Popular Artists')\n\nfig.update_xaxis_categories(df['Name'])\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 20706418894, 'load_duration': 704668, 'prompt_eval_count': 249, 'prompt_eval_duration': 10361959000, 'eval_count': 53, 'eval_duration': 10209795000}


200
150
100
50
0
TotalTracks
Iron Maiden
U2
Led Zeppelin
Metallica
Lost
Deep Purple
Pearl Jam
Lenny Kravitz
Various Artists
The Office
Name

Out[33]:
```
('SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.Ar
tistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10',
              Name  TotalTracks
 0      Iron Maiden          213
 1               U2          135
 2     Led Zeppelin          114
 3        Metallica          112
 4             Lost           92
 5      Deep Purple           92
 6        Pearl Jam           67
 7    Lenny Kravitz           57
 8  Various Artists           56
 9       The Office           53,
 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(['Iron Maiden', 'U2', 'Led Zeppelin', 'Metallica', 'Lost', 'Deep Purple',
                           '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},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalTracks'}}}
 }))
```

In [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
```

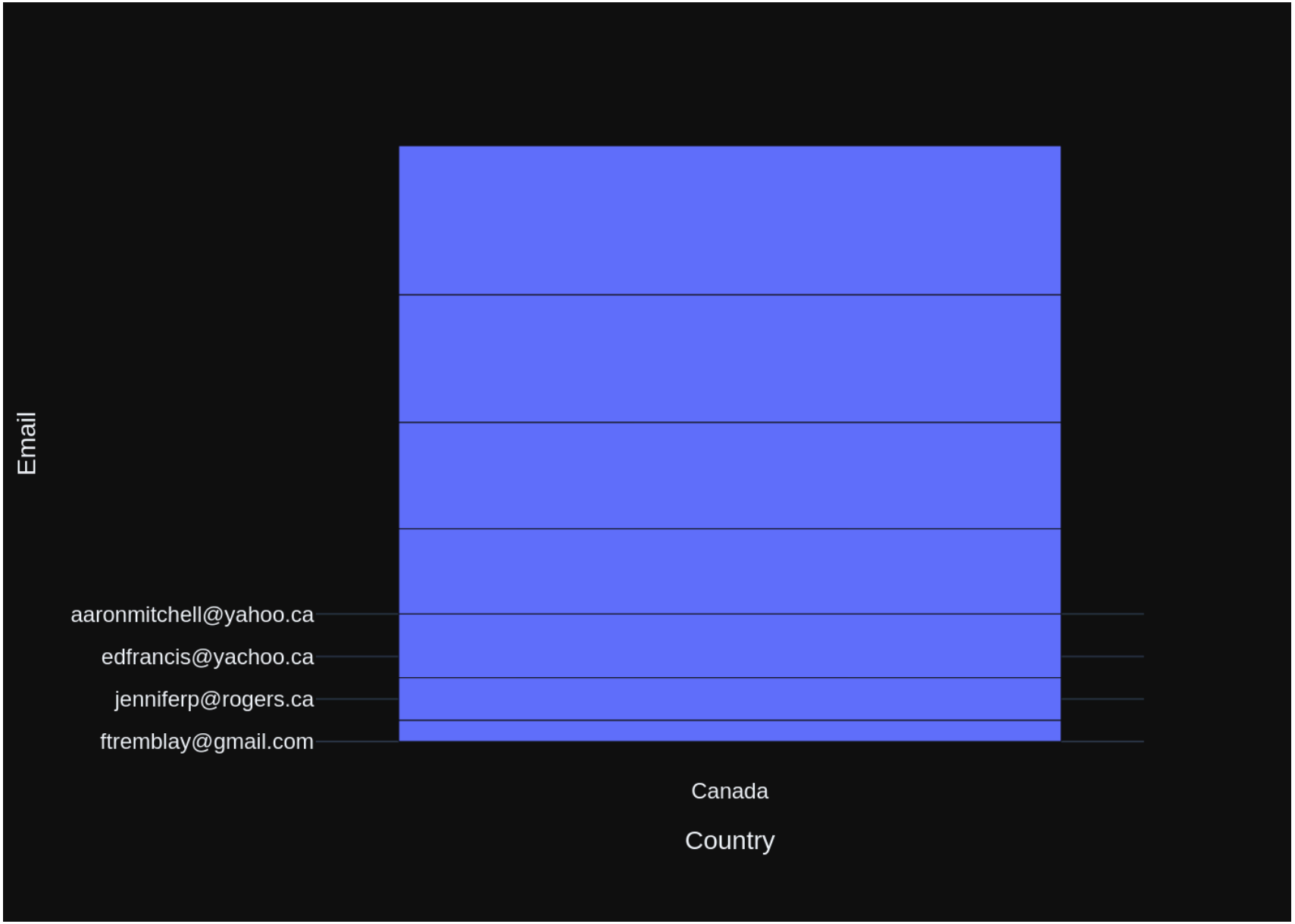
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE 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': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.Customer

Id, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Customer Id\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assista nt', 'content': 'SELECT COUNT(*) FROM "customers"'}, {'role': 'user', 'content': ' \n   Find the top 5 c ustomers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Custome rId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, C OUNT(*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most al bums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, C OUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.Invoi ceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AV G(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGR OUP BY c.CustomerId'}, {'role': 'user', 'content': " \n   List all employees and their reporting manage r's name (if any):\n"}, {'role': 'assistant', 'content': 'SELECT e.FirstName, e.LastName, mt.FirstName AS M anagerFirstName, mt.LastName AS ManagerLastName\nFROM "employees" e\nLEFT JOIN "employees" mt ON e.ReportsT o = mt.EmployeeId'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $1 0:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM "invoices" \nWHERE Total > 10.00'}, {'role': 'use r', 'content': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistan t', '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   List all customers from C anada and their email addresses:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and 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(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": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM \"customers\""}, {"role": "user", "content": " \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "content": " \n   Hint: album quantity is found in invoice_items, \n\n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT 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 average invoice total for each customer:\n"},
{"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customer
s\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "conte
nt": "  \n    List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "c
ontent": "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", "cont
ent": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT
* \nFROM \"invoices\" \nWHERE Total > 10.00"}, {"role": "user", "content": "  \n    Find all invoices since
2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceDate, SUM(i.Tota
l) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"ro
le": "user", "content": "  \n     List all customers from Canada and their email addresses:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:06:14.377336709Z', 'message': {'role': 'assistant',
'content': 'SELECT c.Email, c.Country\nFROM "customers" c\nWHERE c.Country = \'Canada\''}, 'done_reason':
'stop', 'done': True, 'total_duration': 72203775745, 'load_duration': 859002, 'prompt_eval_count': 1533, 'p
rompt_eval_duration': 67365453000, 'eval_count': 21, 'eval_duration': 4071656000}
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=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n     List all customers from Canada and their email 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': 'llama3:latest', 'created_at': '2024-06-13T22:06:28.089953606Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nimport pandas as pd\n\nfig = px.bar(df, x='Country', y='Emai
l')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 13686630585, 'load_duratio
n': 788033, 'prompt_eval_count': 162, 'prompt_eval_duration': 7071680000, 'eval_count': 34, 'eval_duratio
n': 6475678000}


Email
aaronmitchell@yahoo.ca
edfrancis@yachoo.ca
jenniferp@rogers.ca
ftremblay@gmail.com
Canada
Country

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': [{'alignmentgroup': 'True',
               'hovertemplate': 'Country=%{x}<br>Email=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Canada', 'Canada', 'Canada', 'Canada', 'Canada', 'Canada', 'Canada',
                           'Canada'], dtype=object),
               'xaxis': 'x',
               'y': array(['ftremblay@gmail.com', 'mphilips12@shaw.ca', 'jenniferp@rogers.ca',
                           'robbrown@shaw.ca', 'edfrancis@yachoo.ca', 'marthasilk@gmail.com',
                           'aaronmitchell@yahoo.ca', 'ellie.sullivan@shaw.ca'], dtype=object),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Country'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Email'}}}
 }))
```

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
```

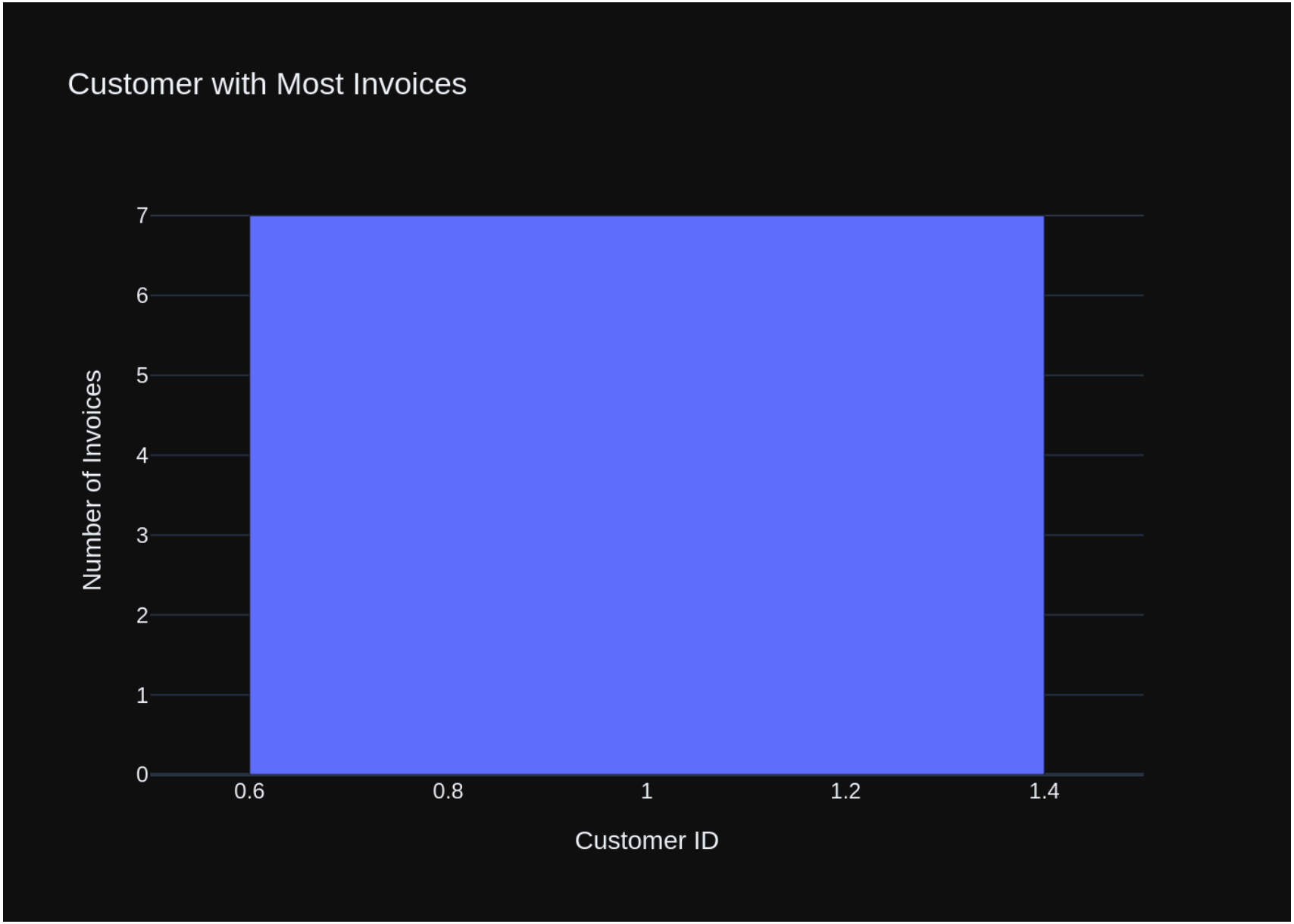
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY 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 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) A

S TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\n
ORDER 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 al
l invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAl
bums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDE
R BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for
each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoic
es\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role':
'user', 'content': ' \n   Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'co
ntent': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i ON
c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n   List all invoice
s with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM "invoices" \nWHERE Tot
al > 10.00'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'ro
le': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM "invoices" i\nGROUP
BY i.BillingCountry'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amou
nt invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM
"invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'role': 'user', 'content':
' \n   Find the top 5 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': 'us
er', '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': 'How many customers are there'}, {'role': 'assistant', 'co
ntent': 'SELECT COUNT(*) FROM "customers"'}, {'role': 'user', 'content': ' \n   Find the customer with t
he most invoices \n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and 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 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 top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n   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   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   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM \"invoices\" \nWHERE Total > 10.00"}, {"role": "user", "content": "

```
\n    Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT i.Billin
gCountry, COUNT(*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "co
ntent": "  \n    Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "co
ntent": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate >= '20
10-01-01'\nGROUP BY i.InvoiceDate"}, {"role": "user", "content": "  \n    Find the top 5 most expensive tra
cks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFR
OM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "what are the top 5 coun
tries that customers come from?"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(*) AS TotalCus
tomers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user",
"content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM \"custom
ers\""}, {"role": "user", "content": "  \n     Find the customer with the most invoices \n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:07:45.627052873Z', 'message': {'role': 'assistant',
'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i
ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, 'done_reaso
n': 'stop', 'done': True, 'total_duration': 77395471767, 'load_duration': 909497, 'prompt_eval_count': 155
1, 'prompt_eval_duration': 65844467000, 'eval_count': 53, 'eval_duration': 10808076000}
SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalInvoices DESC
LIMIT 1
SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalInvoices DESC
LIMIT 1
   CustomerId  TotalInvoices
0           1              7
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n     Find the customer with the most invoices \n'\n\nThe Dat
aFrame was produced using this query: SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"cust
omers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoice
s DESC\nLIMIT 1\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dty
pes 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': 'llama3:latest', 'created_at': '2024-06-13T22:08:07.735804204Z', 'message': {'role': 'assistant', 'content': "```\nimport plotly.graph_objects as go\n\nfig = go.Figure(data=[go.Bar(x=df['CustomerId'], y=df['TotalInvoices'])])\nfig.update_layout(title='Customer with Most Invoices', xaxis_title='Customer ID', yaxis_title='Number of Invoices')\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 22067970038, 'load_duration': 1124776, 'prompt_eval_count': 198, 'prompt_eval_duration': 9576907000, 'eval_count': 64, 'eval_duration': 12356762000}


Customer with Most Invoices
Number of Invoices
0
1
2
3
4
5
6
7
0.6
0.8
1
1.2
1.4
Customer ID

Out[35]:
```
('SELECT 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',
    CustomerId  TotalInvoices
 0           1             7,
 Figure({
     'data': [{'type': 'bar', 'x': array([1]), 'y': array([7])}],
     'layout': {'template': '...',
                'title': {'text': 'Customer with Most Invoices'},
                'xaxis': {'title': {'text': 'Customer ID'}},
                'yaxis': {'title': {'text': 'Number of Invoices'}}}
 }))
```

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   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   Find 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

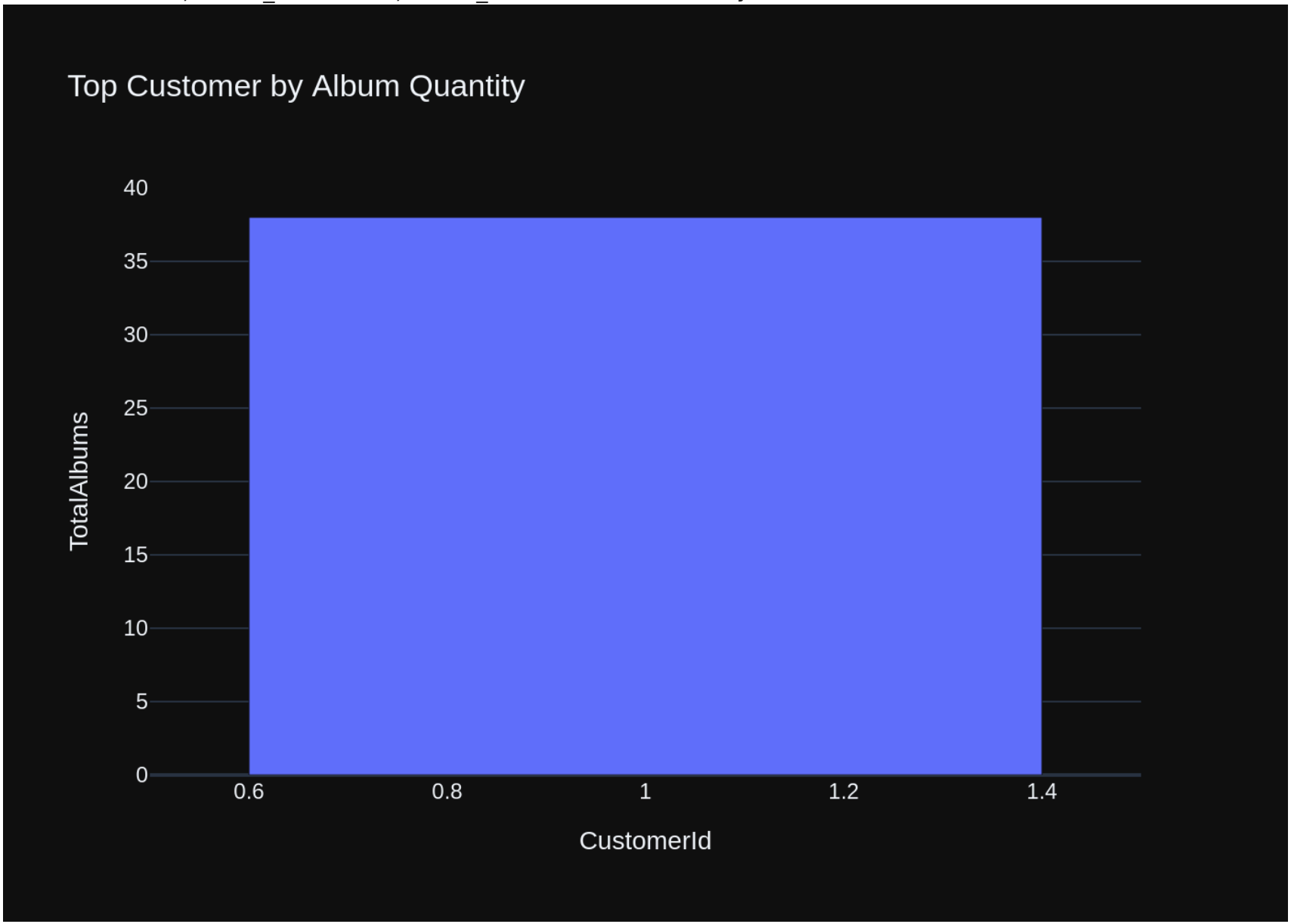
c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   There are 3 tables:
artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked b
y AlbumId,\n   Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'a
ssistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al O
N a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTrack
s DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amount
invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM "i
nvoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'role': 'user', 'content': '
\n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.Cu
stomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Cu
stomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive track
s (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   Find the total numbe
r of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(*) AS Tota
lInvoices\nFROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': ' \n   Get the ave
rage invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Tot
al) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY
c.CustomerId'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'r
ole': 'assistant', 'content': 'SELECT * \nFROM "invoices" \nWHERE Total > 10.00'}, {'role': 'user', 'conten
t': ' \n   Find the customer who bought the most albums in total quantity (across all invoices): \n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"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   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
i.CustomerId, COUNT(ii.InvoiceLineId) 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"}, {"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 the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be
found on invoices table, calculation using invoice_items detail table is unnecessary \n"}, {"role": "assist
ant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\"
i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "use
r", "content": " \n   There are 3 tables: artists, albums and tracks, where albums and artists are linked
by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most popular artists bas
ed on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalT
racks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId
= t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n   F
ind 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 B
Y i.InvoiceDate"}, {"role": "user", "content": " \n   Get the total number of invoices for each customer
\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"cu
stomers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user",
"content": " \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   Find the total number of invoices per country:\n"}, {"role": "assistan
t", "content": "SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.Billing
Country"}, {"role": "user", "content": " \n   Get the average invoice total for each customer:\n"}, {"rol
e": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c

\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM \"invoices\" \nWHERE Total > 10.00"}, {"role": "user", "content": " \n   Find the customer who bought the most albums in total quantity (across all invoices): \n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:09:29.560291824Z', 'message': {'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'}, 'done_reason': 'stop', 'done': True, 'total_duration': 81731471410, 'load_duration': 1021267, 'prompt_eval_count': 1484, 'prompt_eval_duration': 67409259000, 'eval_count': 69, 'eval_duration': 13588015000}
```
SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
JOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1
```
```
SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
JOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1
```
```
   CustomerId  TotalAlbums
0           1           38
```
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n   Find the customer who bought the most albums in total quantity (across all invoices): \n'\n\nThe DataFrame was produced using this query: 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\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId   int64\nTotalAlbums   int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]

Ollama Response:

{'model': 'llama3:latest', 'created_at': '2024-06-13T22:09:48.243193503Z', 'message': {'role': 'assistant', 'content': "```\nimport plotly.express as px\nimport pandas as pd\n\nfig = px.bar(df, x='CustomerId', y='TotalAlbums', title='Top Customer by Album Quantity')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 18654342324, 'load_duration': 821653, 'prompt_eval_count': 224, 'prompt_eval_duration': 9887049000, 'eval_count': 45, 'eval_duration': 8625738000}



Out[36]:
```
('SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.Custom
erId = i.CustomerId\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER B
Y TotalAlbums DESC\nLIMIT 1',
    CustomerId  TotalAlbums
 0           1           38,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'CustomerId=%{x}<br>TotalAlbums=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([1]),
               'xaxis': 'x',
               'y': array([38]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Top Customer by Album Quantity'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalAlbums'}}}
 }))
```

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

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

vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n   ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the customer 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 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.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'}, {'role': 'user', 'content': ' \n   There are 3 tables: artists, a
lbums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumI
d,\n   Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistan
t', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\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   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 most expensive tracks (based on unit price):\n'}, {'rol
e': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice D
ESC\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 "cus
tomers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'con
tent': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELEC
T i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'us
er', 'content': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assista
nt', '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   Get the average invoice t
otal for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS Average
InvoiceTotal\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerI
d'}, {'role': 'user', 'content': ' \n   Hint: album quantity is found in invoice_items, \n   \n   Find
the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoice_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

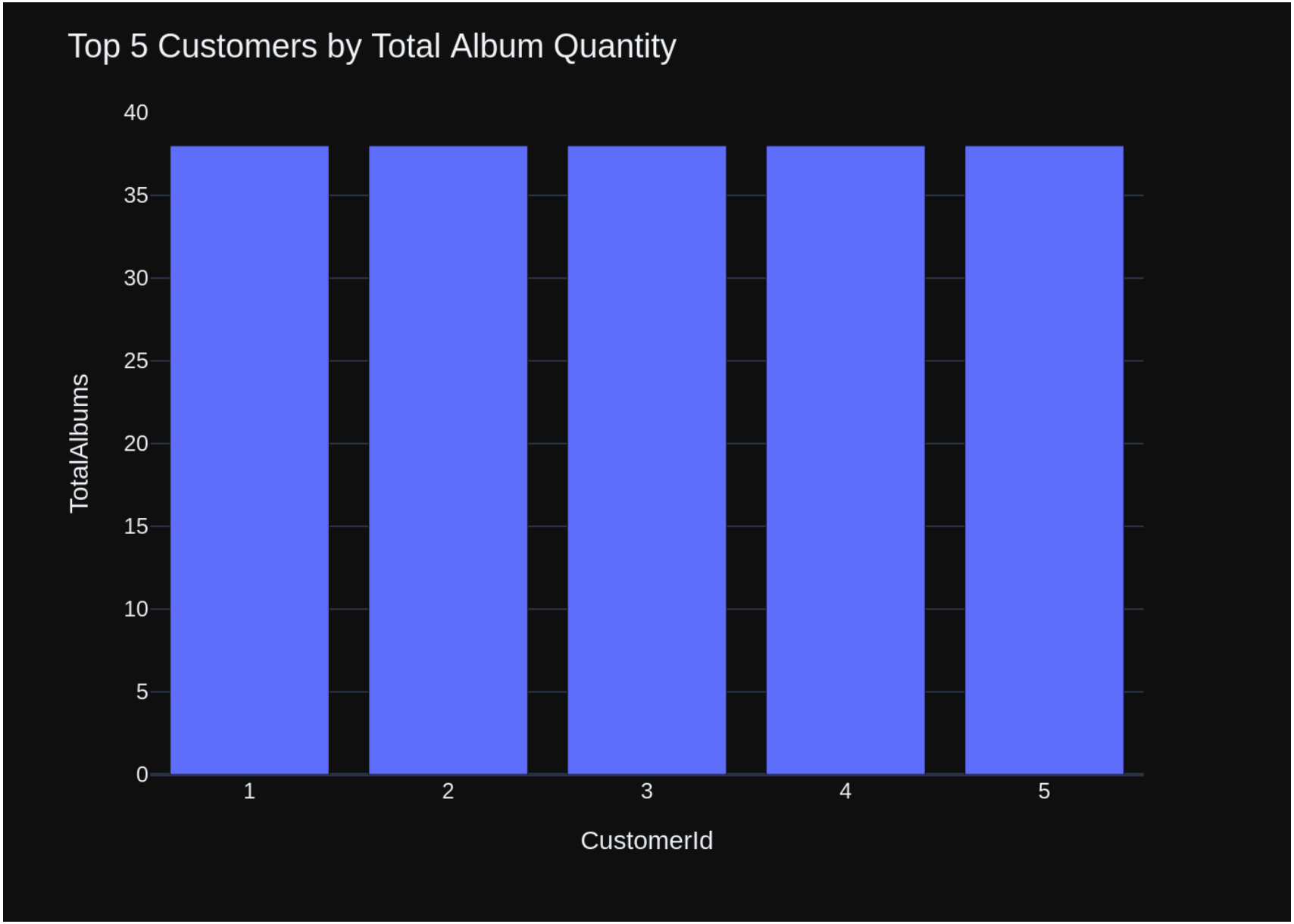
T NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (Ar
tistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IN
DEX IFK_AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON \"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 i.Cust
omerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.Invoice
Id = 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"}, {"rol
e": "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
\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n   Find t
he top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoic
es table, calculation using invoice_items detail table is unnecessary \n"}, {"role": "assistant", "conten
t": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.Custom
erId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "conten
t": " \n   There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistI
d, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most popular artists based on the
number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFRO
M \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumI
d\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n   Find the c
ustomer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.Invoice
Id) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.
CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n   Find the top 5 mos
t 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 the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.Customer
Id, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Cust

omerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   Find 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 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   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   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:11:10.90133083Z', 'message': {'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'}, 'done_reason': 'stop', 'done': True, 'total_duration': 82568866119, 'load_duration': 852618, 'prompt_eval_count': 1559, 'prompt_eval_duration': 67813112000, 'eval_count': 69, 'eval_duration': 14017274000}
```
SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
JOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5
```
```
SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
JOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5
```

	CustomerId	TotalAlbums
0	1	38
1	2	38
2	3	38
3	4	38
4	5	38

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

[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: ' \n   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 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\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId   int64\nTotalAlbums   int64 \ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:11:31.528814716Z', 'message': {'role': 'assistant', 'content': "```\nimport plotly.express as px\n\nfig = px.bar(df, x='CustomerId', y='TotalAlbums')\nfig.update_layout(title_text='Top 5 Customers by Total Album Quantity', title_font_size=18)\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 20606399035, 'load_duration': 763626, 'prompt_eval_count': 238, 'prompt_eval_duration': 10129721000, 'eval_count': 53, 'eval_duration': 10340477000}


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

Out[37]:
```
('SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.Custom
erId = i.CustomerId\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER B
Y TotalAlbums DESC\nLIMIT 5',
    CustomerId  TotalAlbums
 0           1           38
 1           2           38
 2           3           38
 3           4           38
 4           5           38,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'CustomerId=%{x}<br>TotalAlbums=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([1, 2, 3, 4, 5]),
               'xaxis': 'x',
               'y': array([38, 38, 38, 38, 38]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'title': {'font': {'size': 18}, 'text': 'Top 5 Customers by Total Album Quantity'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalAlbums'}}}
 }))
```

```sql
SELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums
FROM Customers c
JOIN invoices i  ON c.CustomerId = i.CustomerId
JOIN invoice_items il ON i.InvoiceId = il.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5
```

In [38]:
```
question = """
     Find the top 5 customers who spent the most money overall,

     Hint: order total can be found on invoices table, calculation using invoice_items detail table is unne
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and 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.Custome
rId\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 i
n total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i
i.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invo
ice_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 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 "invoi
ce_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'},
{'role': 'user', 'content': '  \n     Find the customer with the most invoices \n'}, {'role': 'assistant',
'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i
ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'us
er', 'content': '  \n     Find the customer who bought the most albums in total quantity (across all invoic
es): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "c
ustomers" 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.CustomerI
d, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerI
d\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': '  \n    Get the total number of invoices for each
customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nF
ROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'use
r', 'content': '  \n    Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistan
t', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT
5'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assis
tant', 'content': 'SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nOR
DER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find all invoices since 2010 an
d the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS T
otalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'role': 'u
ser', 'content': '  \n     Find the top 5 customers who spent the most money overall, \n     \n     Hint: o
rder total can be found on invoices table, calculation using invoice_items detail table is unnecessary
\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
```

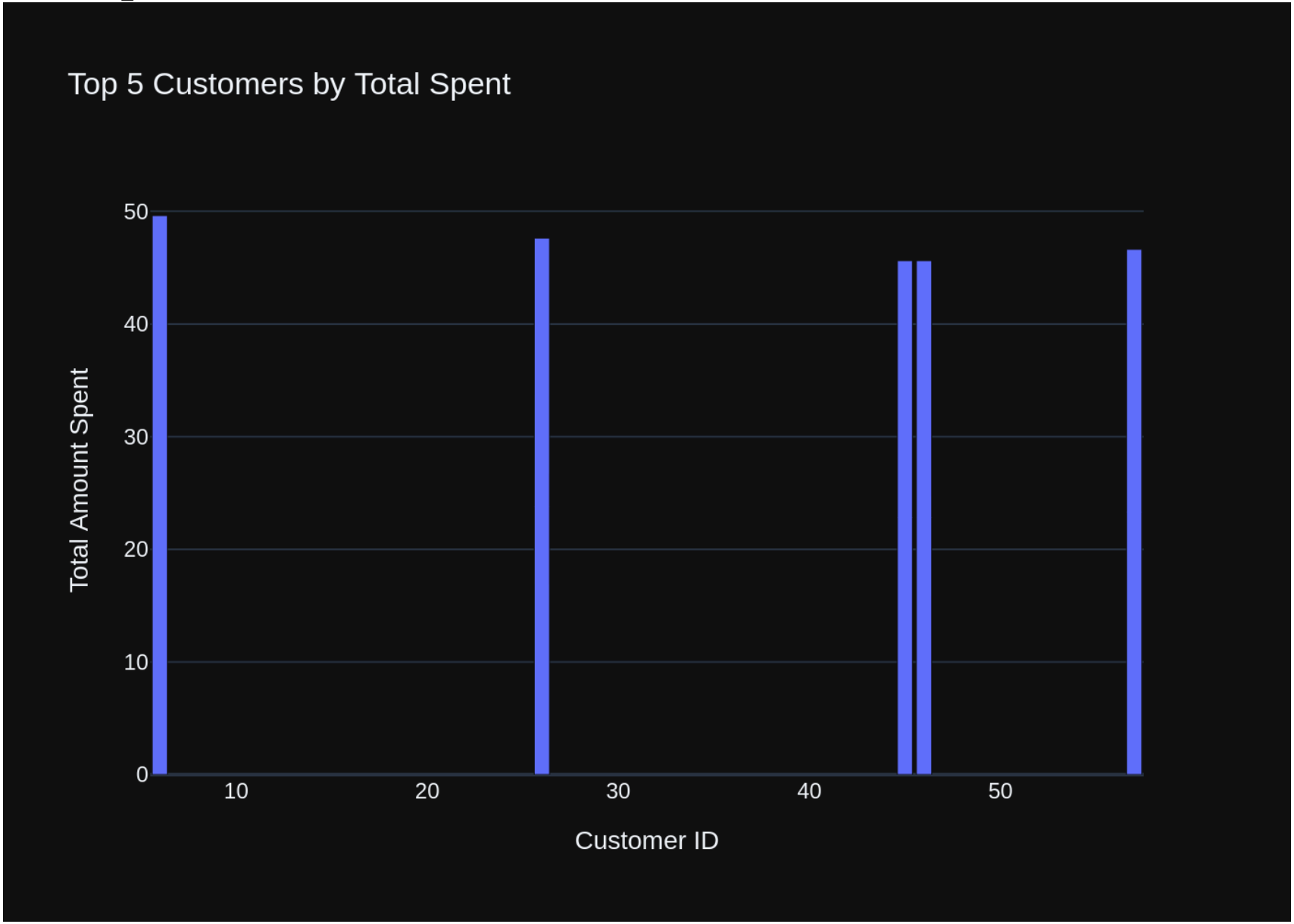
rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   Billin
gAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   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"}, {"role": "use
r", "content": " \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customer
s who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content":
"SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.Cust
omerId = i.CustomerId\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDE
R BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Hint: album quantity is found in inv
oice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all i
nvoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbum
s\nFROM \"invoices\" i\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORD
ER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find the customer with the most in
voices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFR
OM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY Tot
alInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n   Find the customer who bought the most alb
ums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, C
OUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\n
JOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\n
LIMIT 1"}, {"role": "user", "content": " \n   Get the average invoice total for each customer:\n"}, {"rol
e": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c
\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"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.CustomerI
d = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   Find the top 5 most expensi
ve tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPri
ce\nFROM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "what are the top
5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(*) AS To
talCustomers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role":
"user", "content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assi
stant", "content": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.Invoice
Date >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"role": "user", "content": " \n   Find the top 5 custom
ers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calcu
lation using invoice_items detail table is unnecessary \n"}]

Insert of existing embedding ID: 32b99e7b-31ab-55d8-8431-fb010fa7af85-sql
Add of existing embedding ID: 32b99e7b-31ab-55d8-8431-fb010fa7af85-sql

```
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:13:04.645016676Z', 'message': {'role': 'assistant',
'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.Cus
tomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, 'done_reason': 'stop',
'done': True, 'total_duration': 93019459056, 'load_duration': 964022, 'prompt_eval_count': 1832, 'prompt_ev
al_duration': 82003529000, 'eval_count': 51, 'eval_duration': 10270848000}
SELECT c.CustomerId, SUM(i.Total) AS TotalSpent
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 5
SELECT c.CustomerId, SUM(i.Total) AS TotalSpent
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 5
   CustomerId  TotalSpent
0           6       49.62
1          26       47.62
2          57       46.62
3          45       45.62
4          46       45.62
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n     Find the top 5 customers who spent the most money overa
ll, \n     \n     Hint: order total can be found on invoices table, calculation using invoice_items detail
table is unnecessary \n'\n\nThe DataFrame was produced using this query: SELECT c.CustomerId, SUM(i.Total)
AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Custome
rId\nORDER BY TotalSpent DESC\nLIMIT 5\n\nThe following is information about the resulting pandas DataFrame
'df': \nRunning df.dtypes gives:\n CustomerId      int64\nTotalSpent    float64\ndtype: object"}, {"role":
"user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume t
he data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicato
r. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:13:27.625037731Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\n\nfig = px.bar(df, x='CustomerId', y='TotalSpent', title='Top
```

5 Customers by Total Spent')\n\nfig.update_layout(xaxis_title='Customer ID',\n yaxis_title
='Total Amount Spent')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 2295393
2026, 'load_duration': 699180, 'prompt_eval_count': 224, 'prompt_eval_duration': 9963281000, 'eval_count':
63, 'eval_duration': 12854993000}



Out[38]:
```
('SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId =
i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5',
    CustomerId  TotalSpent
 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>TotalSpent=%{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},
                'template': '...',
                'title': {'text': 'Top 5 Customers by Total Spent'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Customer ID'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total Amount Spent'}}}
 }))
```

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   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   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.Cus

tomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%\''}, {'role': 'user', 'content': ' \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the top 5 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   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   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': '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   Get all playlists containing at least 10 tracks and the total duration of those tracks:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and 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

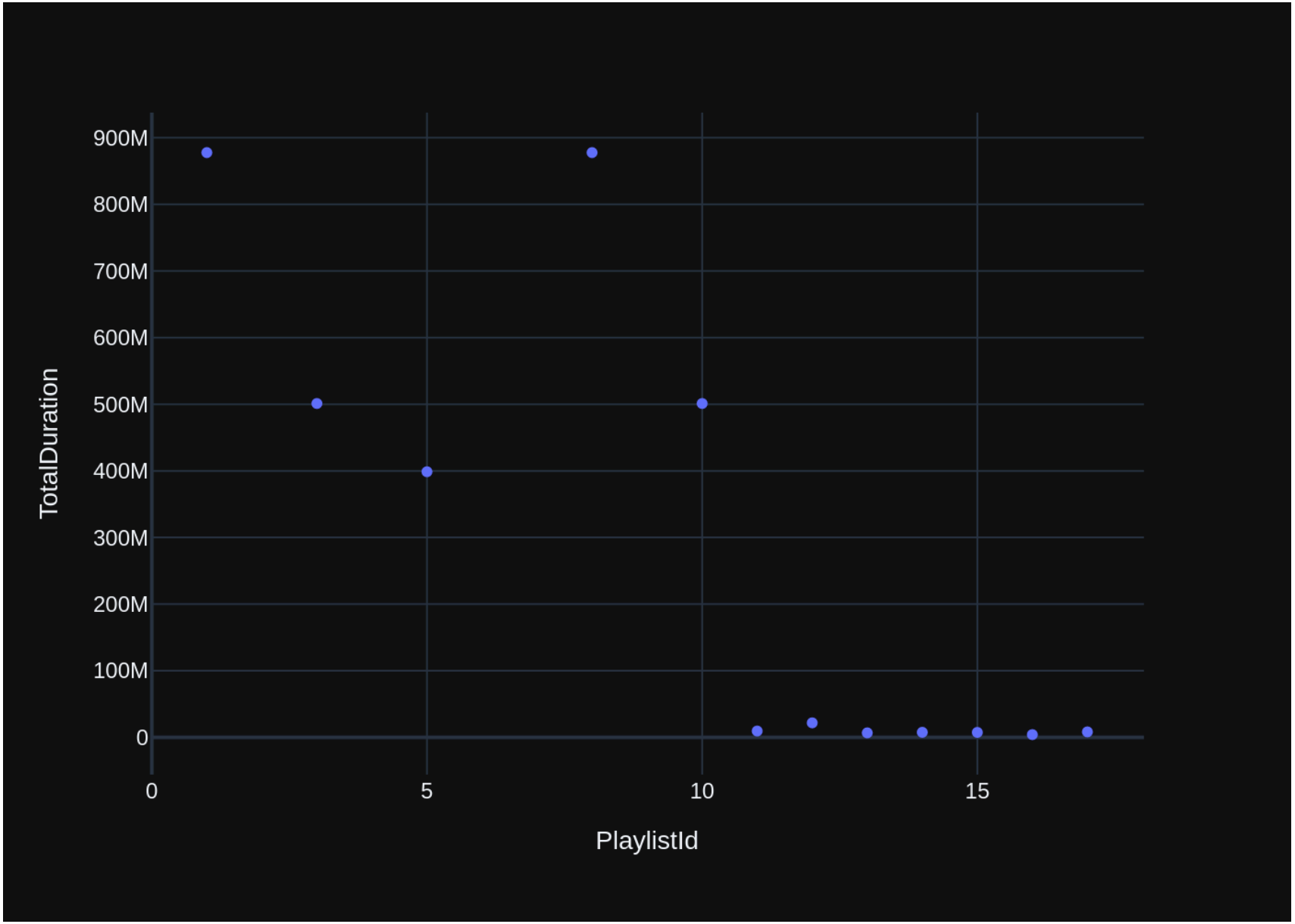
KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the SQLite dat
abase invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please gene
rate a valid SQL query without any explanations for the question. \n2. If the provided 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   List all genres and the n
umber of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS To
talTracks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"role": "use
r", "content": " \n   There are 3 tables: artists, albums and tracks, where albums and artists are linked
by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most popular artists bas
ed on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalT
racks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId
= t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n
Find the customer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "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 * \nFROM \"tracks\" \nWHERE LOWER(Name) LIKE '%what%'"}, {"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
\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role":
"user", "content": " \n   Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assi
stant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLI
MIT 5"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names \n"},
{"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM \"albums\" a\nJOI
N \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   Find all invoices sinc
e 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceDate, SUM(i.To
tal) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.InvoiceDate"},
{"role": "user", "content": "Can you list all tables in the SQLite database catalog?"}, {"role": "assistan
t", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "content": " \n
Get all playlists containing at least 10 tracks and the total duration of those tracks:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:14:47.685860128Z', 'message': {'role': 'assistant',
'content': 'SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM "playl

ist_track" pt\nJOIN "playlists" p ON pt.PlaylistId = p.PlaylistId\nJOIN "tracks" t ON pt.TrackId = t.TrackI
d\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10'}, 'done_reason': 'stop', 'done': True,
'total_duration': 79950896097, 'load_duration': 765958, 'prompt_eval_count': 1409, 'prompt_eval_duration':
62890920000, 'eval_count': 80, 'eval_duration': 16295786000}
```
SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration
FROM "playlist_track" pt
JOIN "playlists" p ON pt.PlaylistId = p.PlaylistId
JOIN "tracks" t ON pt.TrackId = t.TrackId
GROUP BY pt.PlaylistId, p.Name
HAVING COUNT(pt.TrackId) >= 10
```
```
SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration
FROM "playlist_track" pt
JOIN "playlists" p ON pt.PlaylistId = p.PlaylistId
JOIN "tracks" t ON pt.TrackId = t.TrackId
GROUP BY pt.PlaylistId, p.Name
HAVING COUNT(pt.TrackId) >= 10
```

	PlaylistId	PlaylistName	TotalDuration
0	1	Music	877683083
1	3	TV Shows	501094957
2	5	90’s Music	398705153
3	8	Music	877683083
4	10	TV Shows	501094957
5	11	Brazilian Music	9486559
6	12	Classical	21770592
7	13	Classical 101 - Deep Cuts	6755730
8	14	Classical 101 - Next Steps	7575051
9	15	Classical 101 - The Basics	7439811
10	16	Grunge	4122018
11	17	Heavy Metal Classic	8206312

Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: ' \n   Get all playlists containing at least 10 tracks and the
total duration of those tracks:\n'\n\nThe DataFrame was produced using this query: SELECT 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.N
ame\nHAVING COUNT(pt.TrackId) >= 10\n\nThe following is information about the resulting pandas DataFrame 'd
f': \nRunning df.dtypes gives:\n PlaylistId   int64\nPlaylistName   object\nTotalDuration   int64
\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the result

s of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:15:22.255156485Z', 'message': {'role': 'assistant', 'content': '```\nimport plotly.express as px\nimport numpy as np\n\nfig = px.bar(df, x="PlaylistName", y="TotalDuration")\nfig.update_layout(title=\'Total Duration of Tracks in Playlists\')\nfig.show()\n\n# If there is only one value:\nimport plotly.graph_objects as go\n\nfig = go.Indicator(\n   mode="number",\n   value=np.sum(df[\'TotalDuration\']),\n   number={\'font\': {\'size\': 40}, \'text\': "Total Duration"},\n)\nfig.update_layout(title=\'Total Duration\', height=400, width=200)\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 34540730600, 'load_duration': 761949, 'prompt_eval_count': 238, 'prompt_eval_duration': 10317191000, 'eval_count': 123, 'eval_duration': 24083369000}


900M
800M
700M
600M
500M
400M
300M
200M
100M
0
TotalDuration
0
5
10
15
PlaylistId

Out[39]:
```
('SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM "playlist_trac
k" pt\nJOIN "playlists" p ON pt.PlaylistId = p.PlaylistId\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROU
P BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10',
     PlaylistId               PlaylistName  TotalDuration
 0            1                      Music      877683083
 1            3                  TV Shows      501094957
 2            5                 90’s Music      398705153
 3            8                      Music      877683083
 4           10                  TV Shows      501094957
 5           11            Brazilian Music        9486559
 6           12                  Classical       21770592
 7           13   Classical 101 - Deep Cuts        6755730
 8           14  Classical 101 - Next Steps        7575051
 9           15  Classical 101 - The Basics        7439811
 10          16                     Grunge        4122018
 11          17        Heavy Metal Classic        8206312,
 Figure({
     'data': [{'hovertemplate': 'PlaylistId=%{x}<br>TotalDuration=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'symbol': 'circle'},
               'mode': 'markers',
               'name': '',
               'orientation': 'v',
               'showlegend': False,
               'type': 'scatter',
               'x': array([ 1,  3,  5,  8, 10, 11, 12, 13, 14, 15, 16, 17]),
               'xaxis': 'x',
               'y': array([877683083, 501094957, 398705153, 877683083, 501094957,   9486559,
                           21770592,   6755730,   7575051,   7439811,   4122018,   8206312]),
               'yaxis': 'y'}],
     'layout': {'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'PlaylistId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalDuration'}}}
 }))
```

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
```

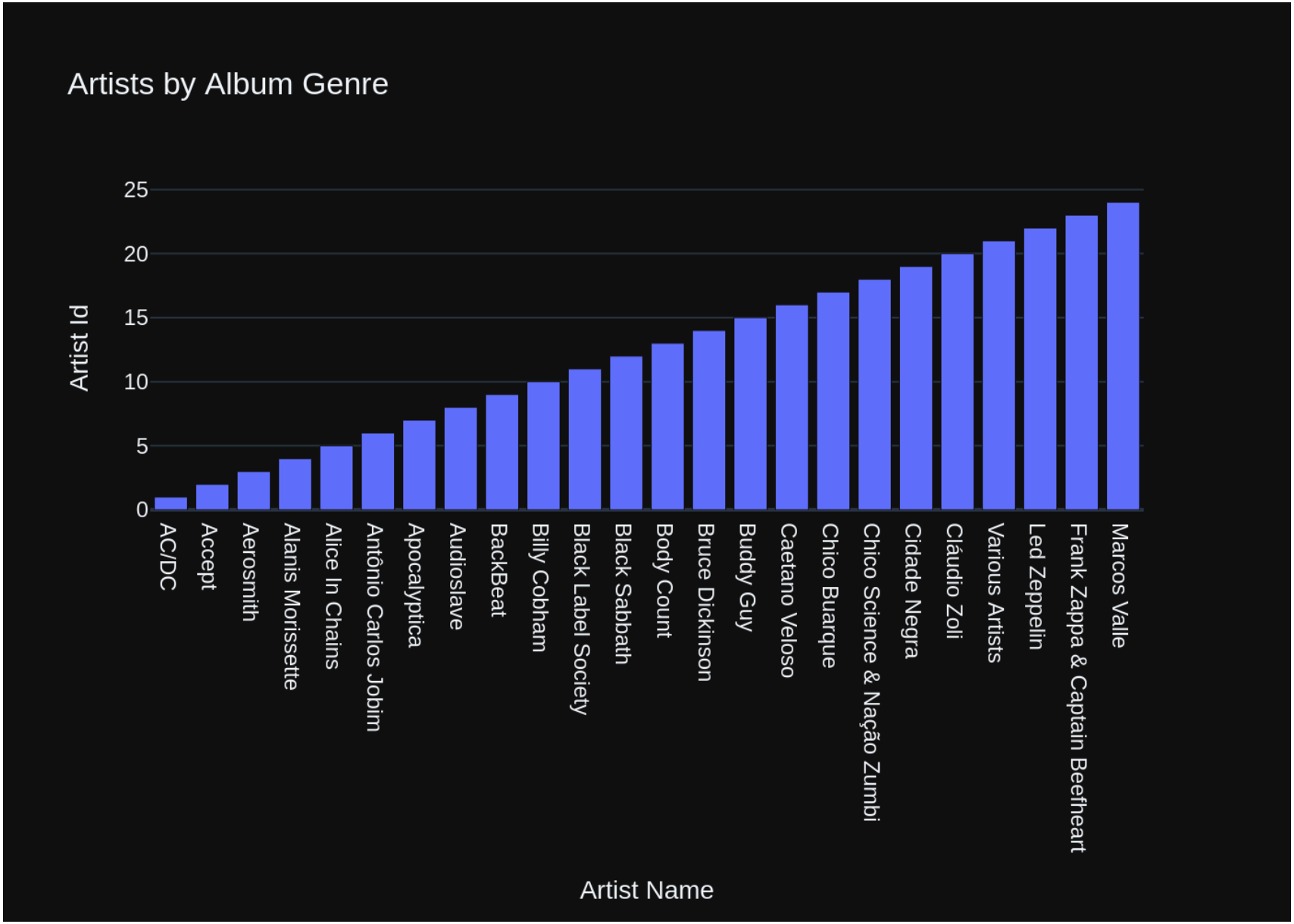
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY 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   There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.Name'}, {'role': 'user', 'content': ' \n   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 al

l invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums
\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice_items" ii ON i.Invoic
eId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'conten
t': ' \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bough
t the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.
CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice_items" ii ON i.Invoice
Id = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content':
' \n   Get all playlists containing at least 10 tracks and the total duration of those tracks:\n'}, {'ro
le': 'assistant', 'content': 'SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDur
ation\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', 'co
ntent': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'co
ntent': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'ro
le': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'ro
le': 'assistant', 'content': 'SELECT * \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%\''}, {'role': 'use
r', '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   Identify artist
s who have albums with tracks appearing in multiple genres:\n\n\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
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_Albu
mArtistId ON \"albums\" (ArtistId)\n\nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX
IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"albums\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY
AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOR
EIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE INDEX IFK_TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE TABLE \"genres\"\r\n(\r\n   Ge
nreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_Playli
stTrackTrackId ON \"playlist_track\" (TrackId)\n\nCREATE TABLE \"artists\"\r\n(\r\n   ArtistId INTEGER PRI
MARY 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\tO

N 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 invoic
e 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 t
o find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. I
f the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most
relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactl
y as it was given before. \n"}, {"role": "user", "content": " \n   There are 3 tables: artists, albums and
tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n   Can
you find the top 10 most popular artists based on the number of tracks\n"}, {"role": "assistant", "conten
t": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId =
al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT
10"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names \n"}, {"ro
le": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN
\"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   List all 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"}, {"role": "use
r", "content": " \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customer
s who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content":
"SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.Cust
omerId = i.CustomerId\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDE
R BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find the customer who bought the mo
st albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.Custome
rId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Custom
erId\nJOIN \"invoice_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums
DESC\nLIMIT 1"}, {"role": "user", "content": " \n   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 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\nHAV
ING COUNT(pt.TrackId) >= 10"}, {"role": "user", "content": " \n   Find the top 5 most expensive tracks (b
ased on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"t
racks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find all tracks with
a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * \nFROM \"trac
ks\" \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": "Can you list all tables in the SQLit
e database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='tabl
e'"}, {"role": "user", "content": " \n   Identify artists who have albums with tracks appearing in multi
ple genres:\n\n\n"}]

```
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:16:51.385198023Z', 'message': {'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'}, 'don
e_reason': 'stop', 'done': True, 'total_duration': 88988353273, 'load_duration': 980392, 'prompt_eval_coun
t': 1445, 'prompt_eval_duration': 65581305000, 'eval_count': 111, 'eval_duration': 22681831000}
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
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
    ArtistId                      ArtistName
0          1                           AC/DC
1          2                          Accept
2          3                       Aerosmith
3          4                Alanis Morissette
4          5                  Alice In Chains
5          6             Antônio Carlos Jobim
6          7                      Apocalyptica
7          8                        Audioslave
8          9                          BackBeat
9         10                      Billy Cobham
10        11               Black Label Society
11        12                     Black Sabbath
```

```
12        13                      Body Count
13        14                  Bruce Dickinson
14        15                       Buddy Guy
15        16                   Caetano Veloso
16        17                    Chico Buarque
17        18      Chico Science & Nação Zumbi
18        19                     Cidade Negra
19        20                     Cláudio Zoli
20        21                  Various Artists
21        22                     Led Zeppelin
22        23  Frank Zappa & Captain Beefheart
23        24                     Marcos Valle
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n     Identify artists who have albums with tracks appearing
in multiple genres:\n\n\n'\n\nThe DataFrame was produced using this query: SELECT a.ArtistId, a.Name AS Art
istName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n  SELEC
T 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\n\nThe following is information about the
resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n ArtistId       int64\nArtistName    object\nd
type: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results o
f the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the da
taframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the c
ode."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-13T22:17:17.121199492Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = go.Figure(data=[g
o.Bar(x=df['ArtistName'], y=df['ArtistId'])])\n\nfig.update_layout(title='Artists by Album Genre', xaxis_ti
tle='Artist Name', yaxis_title='Artist Id')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'tot
al_duration': 25707988214, 'load_duration': 763858, 'prompt_eval_count': 258, 'prompt_eval_duration': 11415
846000, 'eval_count': 70, 'eval_duration': 14146704000}
```


Artists by Album Genre
Artist Id
25
20
15
10
5
0
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
Cidade Negra
Cláudio Zoli
Various Artists
Led Zeppelin
Frank Zappa & Captain Beefheart
Marcos Valle
Artist Name

Out[40]:
```
('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.Ge
nreId\n  GROUP BY g2.GenreId\n  HAVING COUNT(g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name',
     ArtistId                      ArtistName
 0          1                      AC/DC
 1          2                      Accept
 2          3                      Aerosmith
 3          4                      Alanis Morissette
 4          5                      Alice In Chains
 5          6                      Antônio Carlos Jobim
 6          7                      Apocalyptica
 7          8                      Audioslave
 8          9                      BackBeat
 9         10                      Billy Cobham
 10        11                      Black Label Society
 11        12                      Black Sabbath
 12        13                      Body Count
 13        14                      Bruce Dickinson
 14        15                      Buddy Guy
 15        16                      Caetano Veloso
 16        17                      Chico Buarque
 17        18      Chico Science & Nação Zumbi
 18        19                      Cidade Negra
 19        20                      Cláudio Zoli
 20        21                      Various Artists
 21        22                      Led Zeppelin
 22        23  Frank Zappa & Captain Beefheart
 23        24                      Marcos Valle,
 Figure({
     'data': [{'type': 'bar',
               'x': array(['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', 'Cidade Negra', 'Cláudio Zoli',
                           'Various Artists', 'Led Zeppelin', 'Frank Zappa & Captain Beefheart',
                           'Marcos Valle'], dtype=object),
               'y': 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])}],
     'layout': {'template': '...',
                'title': {'text': 'Artists by Album Genre'},
                'xaxis': {'title': {'text': 'Artist Name'}},
```

```
                'yaxis': {'title': {'text': 'Artist Id'}}}
    }))
```

Check completion time

In [ ]:

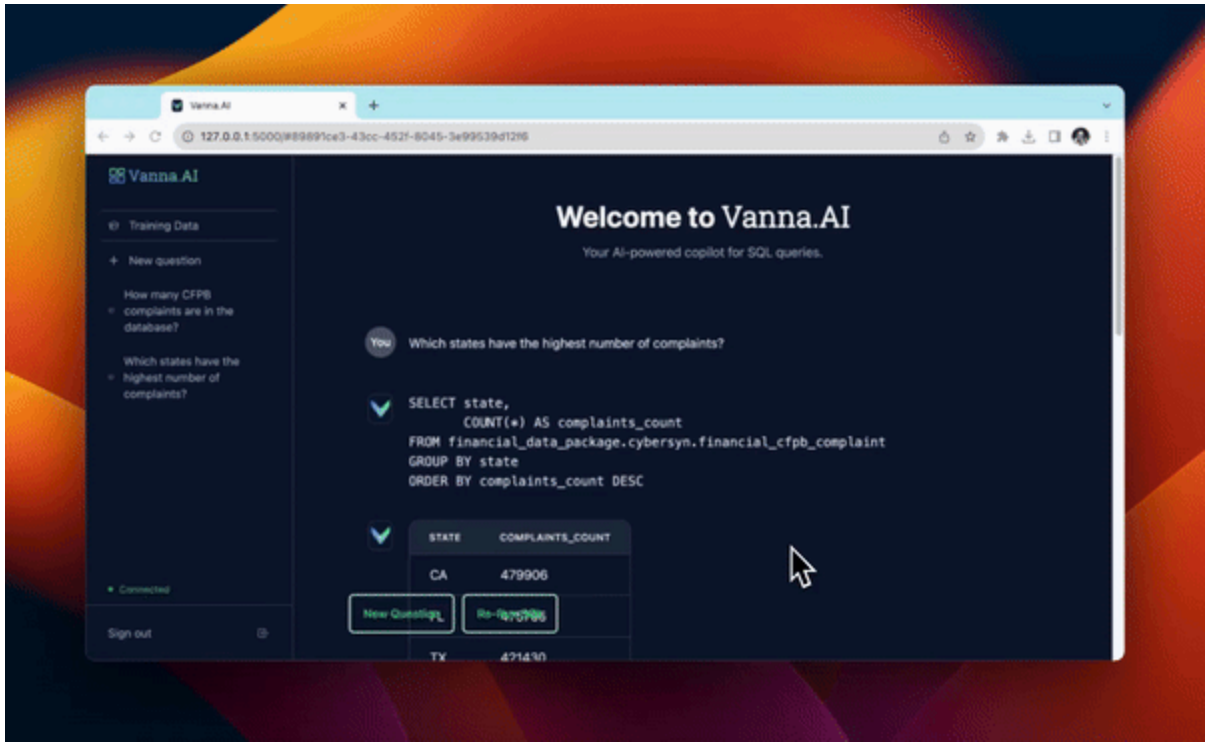
In [41]:
```
ts_stop = time()

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 'llama3' LLM took : 2189.45 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