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 = 'qwen2:7b'

clean_and_train = True  # False
```

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

In [12]:
```
df_ddl
```

Out[12]:

	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 [13]:

```
if clean_and_train:
    for ddl in df_ddl['sql'].to_list():
```

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

    # Sometimes you may want to add documentation about your business terminology or definitions.
    vn.train(documentation="In the chinook database invoice means order")
```

```
Adding ddl: CREATE TABLE "albums"
(
    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    Title NVARCHAR(160)  NOT NULL,
    ArtistId INTEGER  NOT NULL,
    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE sqlite_sequence(name,seq)
Adding ddl: CREATE TABLE "artists"
(
    ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    Name NVARCHAR(120)
)
Adding ddl: CREATE TABLE "customers"
(
    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    FirstName NVARCHAR(40)  NOT NULL,
    LastName NVARCHAR(20)  NOT NULL,
    Company NVARCHAR(80),
    Address NVARCHAR(70),
    City NVARCHAR(40),
    State NVARCHAR(40),
    Country NVARCHAR(40),
    PostalCode NVARCHAR(10),
    Phone NVARCHAR(24),
    Fax NVARCHAR(24),
    Email NVARCHAR(60)  NOT NULL,
    SupportRepId INTEGER,
    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE "employees"
(
    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    LastName NVARCHAR(20)  NOT NULL,
    FirstName NVARCHAR(20)  NOT NULL,
    Title NVARCHAR(30),
    ReportsTo INTEGER,
    BirthDate DATETIME,
    HireDate DATETIME,
    Address NVARCHAR(70),
```

```
    City NVARCHAR(40),
    State NVARCHAR(40),
    Country NVARCHAR(40),
    PostalCode NVARCHAR(10),
    Phone NVARCHAR(24),
    Fax NVARCHAR(24),
    Email NVARCHAR(60),
    FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE "genres"
(
    GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    Name NVARCHAR(120)
)
Adding ddl: CREATE TABLE "invoices"
(
    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    CustomerId INTEGER  NOT NULL,
    InvoiceDate DATETIME  NOT NULL,
    BillingAddress NVARCHAR(70),
    BillingCity NVARCHAR(40),
    BillingState NVARCHAR(40),
    BillingCountry NVARCHAR(40),
    BillingPostalCode NVARCHAR(10),
    Total NUMERIC(10,2)  NOT NULL,
    FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE "invoice_items"
(
    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    InvoiceId INTEGER  NOT NULL,
    TrackId INTEGER  NOT NULL,
    UnitPrice NUMERIC(10,2)  NOT NULL,
    Quantity INTEGER  NOT NULL,
    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId)
                ON DELETE NO ACTION ON UPDATE NO ACTION,
    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE "media_types"
```

```
(
    MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    Name NVARCHAR(120)
)
Adding ddl: CREATE TABLE "playlists"
(
    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    Name NVARCHAR(120)
)
Adding ddl: CREATE TABLE "playlist_track"
(
    PlaylistId INTEGER  NOT NULL,
    TrackId INTEGER  NOT NULL,
    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),
    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId)
                ON DELETE NO ACTION ON UPDATE NO ACTION,
    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE "tracks"
(
    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    Name NVARCHAR(200)  NOT NULL,
    AlbumId INTEGER,
    MediaTypeId INTEGER  NOT NULL,
    GenreId INTEGER,
    Composer NVARCHAR(220),
    Milliseconds INTEGER  NOT NULL,
    Bytes INTEGER,
    UnitPrice NUMERIC(10,2)  NOT NULL,
    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId)
                ON DELETE NO ACTION ON UPDATE NO ACTION,
    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId)
                ON DELETE NO ACTION ON UPDATE NO ACTION,
    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)
Adding ddl: CREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)
Adding ddl: CREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)
Adding ddl: CREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)
Adding ddl: CREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)
```

```
Adding ddl: CREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)
Adding ddl: CREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)
Adding ddl: CREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)
Adding ddl: CREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)
Adding ddl: CREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)
Adding ddl: CREATE TABLE sqlite_stat1(tbl,idx,stat)
Adding documentation....
```

In [14]:
```
# show training data
training_data = vn.get_training_data()
training_data
```

Out[14]:

	id	question	content	training_data_type
0	039f9d54-59f7-5f29-8c04-14dbc3e95671-ddl	None	CREATE TABLE "artists"\r\n(\r\n ArtistId IN...	ddl
1	0db84e3d-ef41-563c-803e-21c1b985dc19-ddl	None	CREATE TABLE "invoices"\r\n(\r\n InvoiceId ...	ddl
2	10cba811-ddba-5042-9e90-d764dfcd1629-ddl	None	CREATE INDEX IFK_InvoiceCustomerId ON "invoice...	ddl
3	2c711317-b93d-5f60-a728-cb1c6fcbc040-ddl	None	CREATE INDEX IFK_CustomerSupportRepId ON "cust...	ddl
4	37319c81-65f7-50ee-956b-795de244bee5-ddl	None	CREATE TABLE sqlite_stat1(tbl,idx,stat)	ddl
5	40bd77cd-e1de-5872-8693-624117ff413c-ddl	None	CREATE INDEX IFK_InvoiceLineInvoiceId ON "invo...	ddl
6	41130543-7164-562a-90a7-0fd0a409c154-ddl	None	CREATE TABLE "albums"\r\n(\r\n AlbumId INTE...	ddl
7	458debc8-8082-5450-a17a-66028bd55ace-ddl	None	CREATE TABLE "playlists"\r\n(\r\n PlaylistI...	ddl
8	4815f3fd-925b-53ce-9dfa-0e4285d5abd3-ddl	None	CREATE TABLE "invoice_items"\r\n(\r\n Invoi...	ddl
9	48d484e9-984c-58ff-b391-75521c69d486-ddl	None	CREATE INDEX IFK_PlaylistTrackTrackId ON "play...	ddl
10	551e1120-a6ee-554f-8b8a-ccf4f22d3636-ddl	None	CREATE INDEX IFK_AlbumArtistId ON "albums" (Ar...	ddl
11	5ff4911e-45c1-5a59-9566-243a9b6a3320-ddl	None	CREATE TABLE "employees"\r\n(\r\n EmployeeI...	ddl
12	65df0648-bf05-5f75-9365-c21f54b2302d-ddl	None	CREATE TABLE "media_types"\r\n(\r\n MediaTy...	ddl
13	6b585176-e66d-5b23-8d86-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-	None	CREATE INDEX IFK_TrackGenreId ON "tracks" (Gen...	ddl

	id	question	content	training_data_type
	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	2b4dda0a-a6ac-5e34-8f76-e41c0734d55e-doc	None	In the chinook database invoice means order	documentation

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 [15]:
```
ts_start = time()
```

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

In [16]:
```
vn.ask(question="Can you list all tables in the SQLite database catalog?")
```

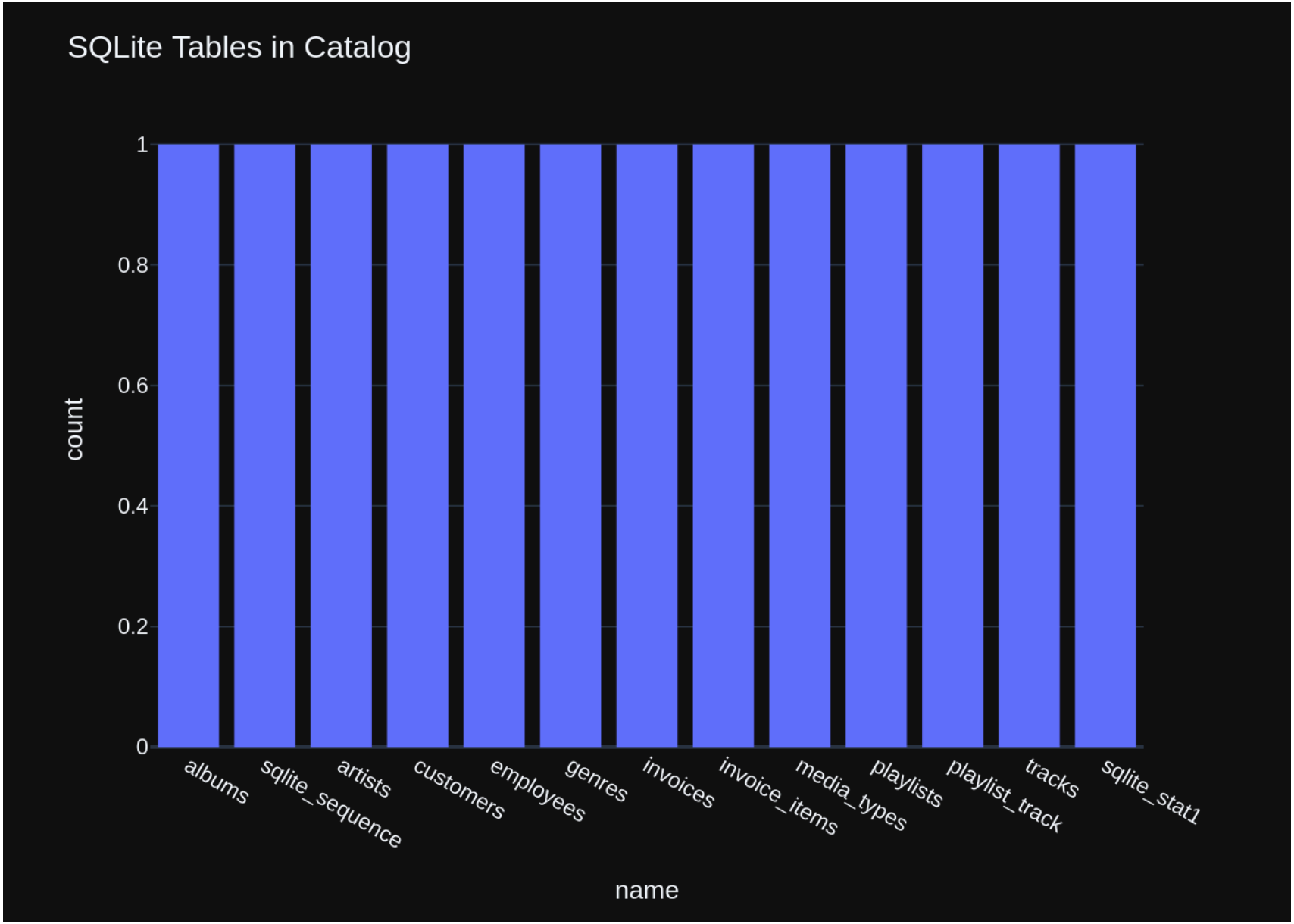
```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

[{'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 chinook 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?'}]
Ollama parameters:
model=qwen2:7b,
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 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 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 chinook database invoice means orde
r\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query w
ithout any explanations for the question. \n2. If the provided context is almost sufficient but requires kn
owledge of a specific string in a particular column, please generate an intermediate SQL query to find the
distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provi
ded context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant ta
ble(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was
given before. \n"}, {"role": "user", "content": "Can you list all tables in the SQLite database catalog?"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:36:19.113602727Z', 'message': {'role': 'assistant', 'con
tent': "SELECT name FROM sqlite_master WHERE type='table';"}, 'done_reason': 'stop', 'done': True, 'total_d
uration': 53462132196, 'load_duration': 1175882606, 'prompt_eval_count': 767, 'prompt_eval_duration': 50392
151000, 'eval_count': 11, 'eval_duration': 1844302000}
SELECT name FROM sqlite_master WHERE type='table';
Output from LLM: SELECT name FROM sqlite_master WHERE type='table';
Extracted SQL: SELECT name FROM sqlite_master WHERE type='table'
SELECT name FROM sqlite_master WHERE type='table'

	name
0	albums
1	sqlite_sequence
2	artists

```
3         customers
4         employees
5            genres
6          invoices
7     invoice_items
8       media_types
9         playlists
10   playlist_track
11           tracks
12     sqlite_stat1
Ollama parameters:
model=qwen2:7b,
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': 'qwen2:7b', 'created_at': '2024-06-15T22:36:41.166304718Z', 'message': {'role': 'assistant', 'con
tent': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n    df[\'Count\'] = 1\n    fig = p
x.bar(df, x=\'name\', y=\'Count\')\nelse:\n    fig = px.histogram(df, x=\'name\')\n\nfig.update_layout(titl
e_text="SQLite Tables in Catalog")\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duratio
n': 22020102076, 'load_duration': 43089491, 'prompt_eval_count': 146, 'prompt_eval_duration': 9530305000,
'eval_count': 74, 'eval_duration': 12397340000}
```


SQLite Tables in Catalog
count
1
0.8
0.6
0.4
0.2
0
albums
sqlite_sequence
artists
customers
employees
genres
invoices
invoice_items
media_types
playlists
playlist_track
tracks
sqlite_stat1
name

Out[16]:
```
("SELECT name FROM sqlite_master WHERE type='table'",
                name
0             albums
1    sqlite_sequence
2            artists
3          customers
4          employees
5             genres
6           invoices
7      invoice_items
8        media_types
9          playlists
10    playlist_track
11            tracks
12     sqlite_stat1,
Figure({
    'data': [{'alignmentgroup': 'True',
              'bingroup': 'x',
              'hovertemplate': 'name=%{x}<br>count=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'v',
              'showlegend': False,
              'type': 'histogram',
              'x': array(['albums', 'sqlite_sequence', 'artists', 'customers', 'employees',
                          'genres', 'invoices', 'invoice_items', 'media_types', 'playlists',
                          'playlist_track', 'tracks', 'sqlite_stat1'], dtype=object),
              'xaxis': 'x',
              'yaxis': 'y'}],
    'layout': {'barmode': 'relative',
               'legend': {'tracegroupgap': 0},
               'margin': {'t': 60},
               'template': '...',
               'title': {'text': 'SQLite Tables in Catalog'},
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'name'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'count'}}}
}))
```

In [17]:
```
vn.ask(question="which table stores customer's orders")
```

```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
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 chinook 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': "which table stores customer's orders"}]

```
Ollama parameters:
model=qwen2:7b,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    Billin
gAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCou
ntry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FORE
IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NO
T NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (Tra
ckId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n    Custom
erId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName N
VARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(4
0),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NV
ARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n
FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO A
CTION\r\n)\n\nCREATE TABLE \"employees\"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(3
0),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR
(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode N
VARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN
KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE \"playlists\"\r\n(\r\n    PlaylistId INTEGER PRI
MARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)
\n\nCREATE TABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NV
ARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES \"artist
s\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n
(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack
PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"media_types\"\r\n(\r\n    MediaTy
peId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n\n===Additional Context
\n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is s
ufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provide
d context is almost sufficient but requires knowledge of a specific string in a particular column, please g
enerate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a com
```

```
ment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be g
enerated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered befo
re, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "Can you lis
t all tables in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_m
aster WHERE type='table'"}, {"role": "user", "content": "which table stores customer's orders"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:37:49.888232313Z', 'message': {'role': 'assistant', 'con
tent': 'The table that stores customer\'s orders is "invoices".'}, 'done_reason': 'stop', 'done': True, 'to
tal_duration': 68305145141, 'load_duration': 577118, 'prompt_eval_count': 1000, 'prompt_eval_duration': 662
38967000, 'eval_count': 12, 'eval_duration': 1923660000}
The table that stores customer's orders is "invoices".
The table that stores customer's orders is "invoices".
Couldn't run sql:  Execution failed on sql 'The table that stores customer's orders is "invoices".': near
"The": syntax error
```

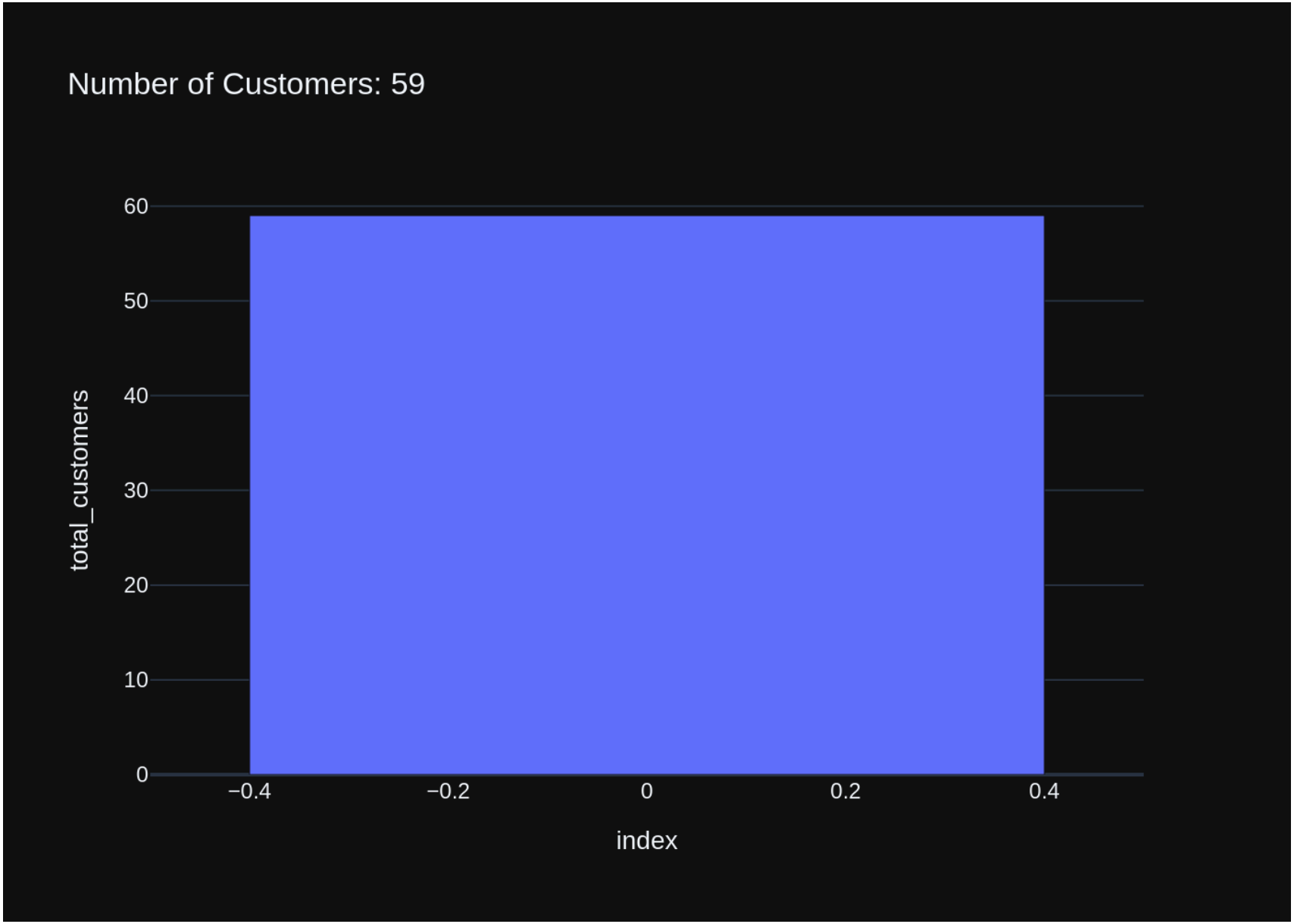
In [18]:
```
vn.ask(question="How many customers are there")
```

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

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingA
ddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCount
ry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIG
N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE 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 NVAR
CHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Ph
one NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGE
R,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPD
ATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE TABLE "invoic
e_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NO
T 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 U
PDATE 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 TAB
LE "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(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 "playlists"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name N
VARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response
Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explan
ations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a spec
ific string in a particular column, please generate an intermediate SQL query to find the distinct strings
in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is in
sufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If
the question has been asked and answered before, please repeat the answer exactly as it was given before.
\n'}, {'role': 'user', 'content': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'ass
istant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': 'How
many customers are there'}]
Ollama parameters:
model=qwen2:7b,
options={},
```

```
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    Billin
gAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCou
ntry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FORE
IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"customers
\"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NU
LL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n
City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(1
0),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    Support
RepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREAT
E TABLE \"invoice_items\"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Invo
iceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n
Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON D
ELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON \"invoice_items\"
(InvoiceId)\n\nCREATE TABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES
\"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLine
TrackId ON \"invoice_items\" (TrackId)\n\nCREATE TABLE \"employees\"\r\n(\r\n    EmployeeId INTEGER PRIMARY
KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NUL
L,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIM
E,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHA
R(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVA
RCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlists\"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCRE
MENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook 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": "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 Response:
```

{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:38:24.671040159Z', 'message': {'role': 'assistant', 'con
tent': 'SELECT COUNT(*) AS total_customers FROM customers;'}, 'done_reason': 'stop', 'done': True, 'total_d
uration': 34739628813, 'load_duration': 1435646, 'prompt_eval_count': 766, 'prompt_eval_duration': 32842056
000, 'eval_count': 11, 'eval_duration': 1755010000}
SELECT COUNT(*) AS total_customers FROM customers;
Output from LLM: SELECT COUNT(*) AS total_customers FROM customers;
Extracted SQL: SELECT COUNT(*) AS total_customers FROM customers
SELECT COUNT(*) AS total_customers FROM customers
   total_customers
0   59
Ollama parameters:
model=qwen2:7b,
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(*) AS total_customers FROM customers\n\nThe following is information about the r
esulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n total_customers   int64\ndtype: object"}, {"r
ole": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? As
sume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Ind
icator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:38:47.726045328Z', 'message': {'role': 'assistant', 'con
tent': '```python\nimport plotly.express as px\n\nif df[\'total_customers\'].nunique() == 1:\n   fig = px.
bar(df, x=df.index, y=\'total_customers\', \n   title=f"Number of Customers: {df[\'total_cust
omers\'][0]}", \n   labels={\'y\': \'Total Customers\'})\nelse:\n   fig = None\n```'}, 'done
_reason': 'stop', 'done': True, 'total_duration': 23027988331, 'load_duration': 43218156, 'prompt_eval_coun
t': 145, 'prompt_eval_duration': 9405891000, 'eval_count': 80, 'eval_duration': 13486103000}


Number of Customers: 59
60
50
40
30
20
10
0
total_customers
−0.4
−0.2
0
0.2
0.4
index

Out[18]:
```
('SELECT COUNT(*) AS total_customers FROM customers',
   total_customers
0               59,
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'index=%{x}<br>total_customers=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'v',
              'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
              'x': array([0]),
              'xaxis': 'x',
              'y': array([59]),
              'yaxis': 'y'}],
    'layout': {'barmode': 'relative',
               'legend': {'tracegroupgap': 0},
               'template': '...',
               'title': {'text': 'Number of Customers: 59'},
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'index'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'total_customers'}}}
})
```

In [ ]:
```
```

In [19]:
```
vn.ask(question="what are the top 5 countries that customers come from?")
```

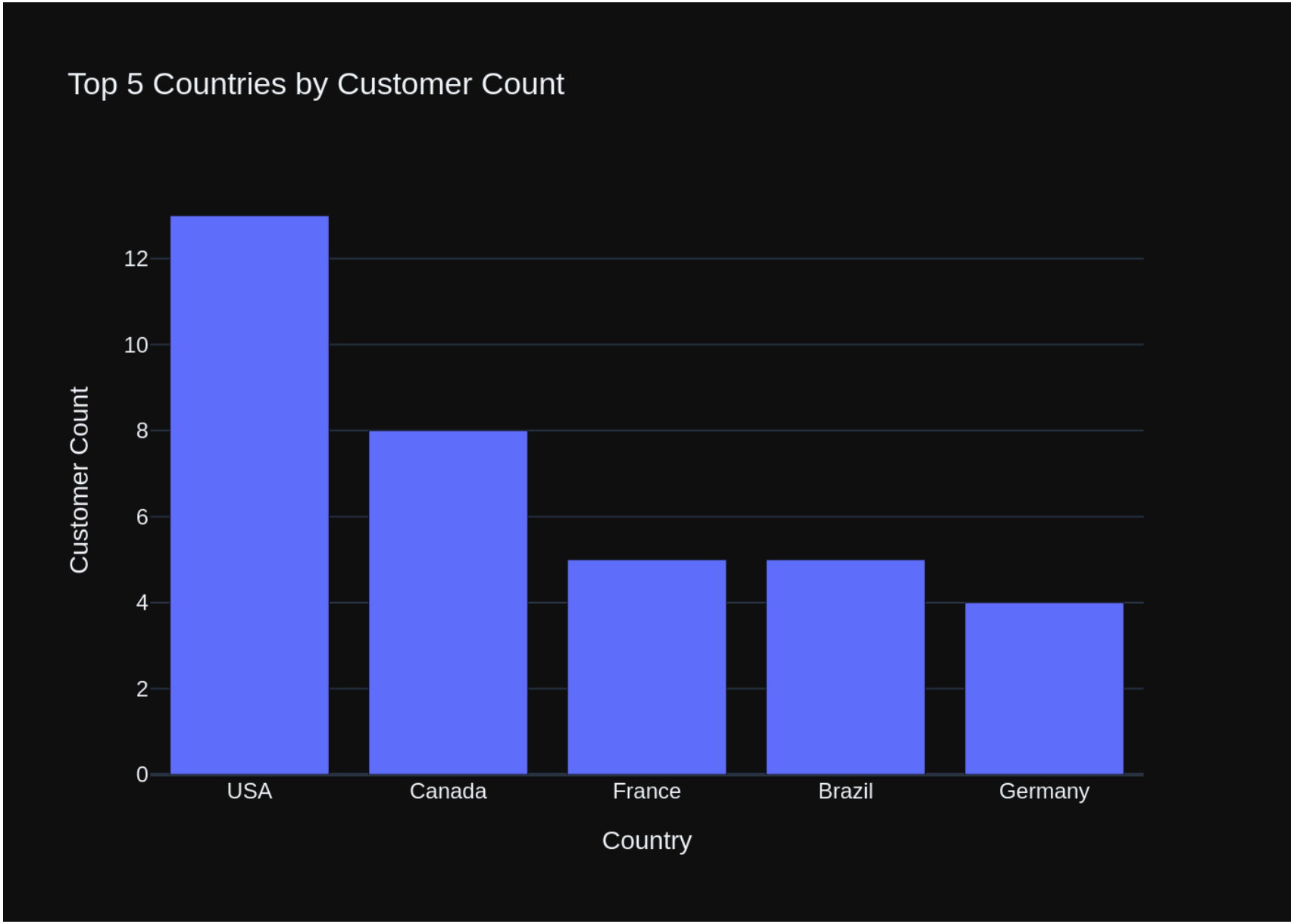
```
Number of requested results 10 is greater than number of elements in index 2, updating n_results = 2
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE 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 chinook 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': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) AS total_customers 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': 'what are the top 5 countries that customers come from?'}]
Ollama parameters:
model=qwen2:7b,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"media_types\"\r\n(\r\n   MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"employees\"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO 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 chinook database invoice means order\n\n===Response Guide
lines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanation
s 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 th
at column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insuffi
cient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the qu
estion 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
(*) AS total_customers 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": "what are the top 5 countries that customers come from?"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:40:11.122223905Z', 'message': {'role': 'assistant', 'con
tent': 'SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_
count DESC \nLIMIT 5'}, 'done_reason': 'stop', 'done': True, 'total_duration': 83295730906, 'load_duratio
n': 730790, 'prompt_eval_count': 1173, 'prompt_eval_duration': 78745272000, 'eval_count': 26, 'eval_duratio
n': 4311251000}
SELECT Country, COUNT(*) as customer_count
FROM customers
GROUP BY Country
ORDER BY customer_count DESC
LIMIT 5
SELECT Country, COUNT(*) as customer_count
FROM customers
GROUP BY Country
ORDER BY customer_count DESC
LIMIT 5
   Country  customer_count
0      USA              13
1   Canada               8
2   France               5
3   Brazil               5
4  Germany               4
Ollama parameters:
```

```
model=qwen2:7b,
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 Country, COUNT(*) as customer_count \nFROM customers \nGROUP
BY Country \nORDER BY customer_count DESC \nLIMIT 5\n\nThe following is information about the resulting pan
das DataFrame 'df': \nRunning df.dtypes gives:\n Country           object\ncustomer_count     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 datafram
e, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the cod
e."}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:40:39.691135497Z', 'message': {'role': 'assistant', 'con
tent': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n    fig = px.indicators.Number(\n
title=\'Top Country: \' + df[\'Country\'].iloc[0],\n        value=df[\'customer_count\'].iloc[0]\n    )\nel
se:\n    fig = px.bar(df, x=\'Country\', y=\'customer_count\', \n                 title=\'Top 5 Countries b
y Customer Count\')\n    fig.update_layout(xaxis_title="Country", yaxis_title="Customer Count")\nfig.show()
\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 28538030711, 'load_duration': 682178, 'prom
pt_eval_count': 170, 'prompt_eval_duration': 10048262000, 'eval_count': 110, 'eval_duration': 18353229000}
```


Top 5 Countries by Customer Count
Customer Count
0
2
4
6
8
10
12
USA
Canada
France
Brazil
Germany
Country

Out[19]:
```
('SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count
DESC \nLIMIT 5',
    Country  customer_count
 0      USA              13
 1   Canada               8
 2   France               5
 3   Brazil               5
 4  Germany               4,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'Country=%{x}<br>customer_count=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['USA', 'Canada', 'France', 'Brazil', 'Germany'], dtype=object),
               'xaxis': 'x',
               'y': array([13,  8,  5,  5,  4]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Top 5 Countries by Customer Count'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Country'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Customer Count'}}}
 }))
```

More SQL questions

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

In [20]:
```
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 3, updating n_results = 3
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_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 chinook 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': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) AS total_customers FROM customers'}, {'role': 'user', 'content': '\n   List all albums and their corresponding artist names \n'}]
Ollama parameters:
model=qwen2:7b,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"albums\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n   TrackId INTEGER P

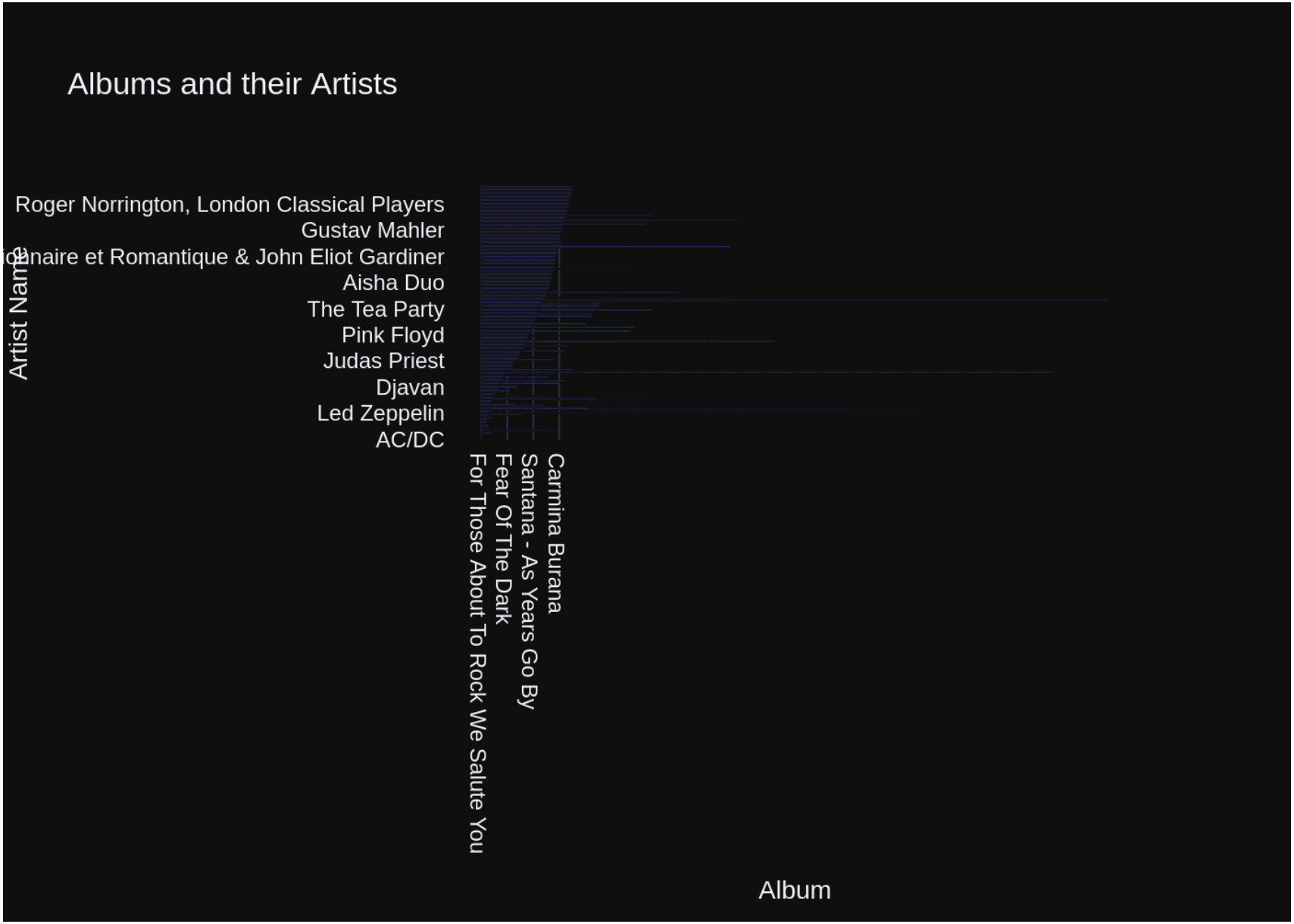
```
RIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    Medi
aTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEG
ER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    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 chinook database invoice means order\n\n===Response Guidelines \n1. If the provided cont
ext 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": "Can
you list all tables in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM s
qlite_master WHERE type='table'"}, {"role": "user", "content": "what are the top 5 countries that customers
come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) as customer_count \nFROM customers
\nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5"}, {"role": "user", "content": "How many custom
ers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) AS total_customers FROM customers"}, {"r
ole": "user", "content": "  \n    List all albums and their corresponding artist names  \n"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:41:30.42645079Z', 'message': {'role': 'assistant', 'cont
ent': 'SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId'},
'done_reason': 'stop', 'done': True, 'total_duration': 50644548287, 'load_duration': 1213552, 'prompt_eval_
count': 697, 'prompt_eval_duration': 45675842000, 'eval_count': 28, 'eval_duration': 4680430000}
SELECT a.Title, ar.Name AS ArtistName
FROM albums a
JOIN artists ar ON a.ArtistId = ar.ArtistId
SELECT a.Title, ar.Name AS ArtistName
FROM albums a
JOIN artists ar ON a.ArtistId = ar.ArtistId
                                          Title  \
0               For Those About To Rock We Salute You
1                                   Balls to the Wall
2                                   Restless and Wild
3                                   Let There Be Rock
4                                           Big Ones
```

```
..                                                 ...
342                              Respighi:Pines of Rome
343  Schubert: The Late String Quartets & String Qu...
344                                 Monteverdi: L'Orfeo
345                               Mozart: Chamber Music
346  Koyaanisqatsi (Soundtrack from the Motion Pict...

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

[347 rows x 2 columns]
Ollama parameters:
model=qwen2:7b,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all albums and their corresponding artist names
\n'\n\nThe DataFrame was produced using this query: SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJ
OIN artists ar ON a.ArtistId = ar.ArtistId\n\nThe following is information about the resulting pandas DataF
rame 'df': \nRunning df.dtypes gives:\n Title         object\nArtistName    object\ndtype: object"}, {"rol
e": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assu
me the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indic
ator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:42:03.653580026Z', 'message': {'role': 'assistant', 'con
tent': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n    fig = px.scatter(\n        data
_frame=df,\n        x=\'ArtistName\',\n        y=\'Title\',\n        size=\'ArtistName\',\n        color=
\'Title\',\n        hover_name=\'ArtistName\',\n        title=\'Album by Artist\'\n    )\nelse:\n    fig =
px.bar(\n        df, \n        x=\'Title\', \n        y=\'ArtistName\',\n        orientation=\'h\',\n
title=\'Albums and their Artists\'\n    )\n\nfig.update_layout(xaxis_title="Album", yaxis_title="Artist Nam
e")\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 33206634208, 'load_duratio
```

```
n': 42328488, 'prompt_eval_count': 170, 'prompt_eval_duration': 11035625000, 'eval_count': 131, 'eval_durat
ion': 22035905000}
```



Out[20]: ('SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId',

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

[347 rows x 2 columns],

```
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'Title=%{x}<br>ArtistName=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'h',
              'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
              'x': 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),
                'xaxis': 'x',
                'y': array(['AC/DC', 'Accept', 'Accept', ...,
                            'C. Monteverdi, Nigel Rogers - Chiaroscuro; London Baroque; London Cornett & Sa
ckbu',
                            'Nash Ensemble', 'Philip Glass Ensemble'], dtype=object),
                'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Albums and their Artists'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Album'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Artist Name'}}}
 }))
```

In [21]:
```
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 4, updating n_results = 4
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_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 chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId'}, {'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 Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) AS total_customers FROM customers'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}]
Ollama parameters:
model=qwen2:7b,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX IFK_

PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\n\nCREATE TABLE \"tracks\"\r\n(\r\n   TrackId INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   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 chinook 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 albums and their
corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, ar.Name AS ArtistName\n
FROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId"}, {"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 fro
m?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP
BY Country \nORDER BY customer_count DESC \nLIMIT 5"}, {"role": "user", "content": "How many customers are
there"}, {"role": "assistant", "content": "SELECT COUNT(*) AS total_customers FROM customers"}, {"role": "u
ser", "content": " \n   Find all tracks with a name containing \"What\" (case-insensitive)\n"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:42:53.863569356Z', 'message': {'role': 'assistant', 'con
tent': "SELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, 'done_reason': 'stop', 'done': Tru
e, 'total_duration': 49983059411, 'load_duration': 1032615, 'prompt_eval_count': 769, 'prompt_eval_duratio
n': 47243214000, 'eval_count': 15, 'eval_duration': 2410721000}
SELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE
SELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE

	TrackId	Name	AlbumId	\
0	26	What It Takes	5	
1	88	What You Are	10	
2	130	Do what cha wanna	13	

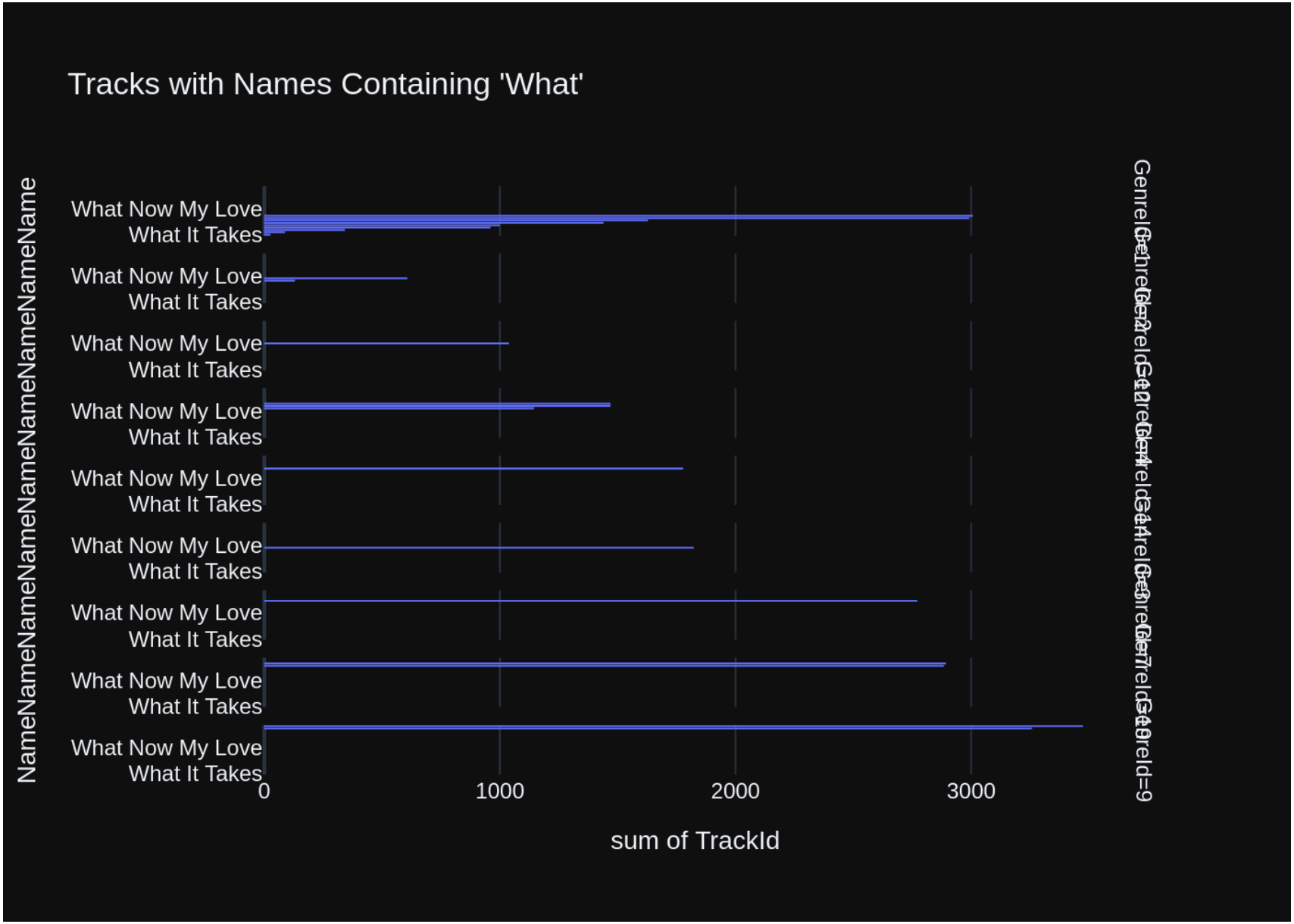
3	342	What is and Should Never Be	30	
4	607	So What	48	
5	960	What A Day	76	
6	1000	What If I Do?	80	
7	1039	What Now My Love	83	
8	1145	Whatsername	89	
9	1440	Whatever It Is, I Just Can't Stop	116	
10	1469	Look What You've Done	119	
11	1470	Get What You Need	119	
12	1628	What Is And What Should Never 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=qwen2:7b,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Find all tracks with a name containing \"What\" (case-in
sensitive)\n'\n\nThe DataFrame was produced using this query: SELECT * FROM tracks WHERE Name LIKE '%What%'
COLLATE NOCASE\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtyp
es gives:\n TrackId           int64\nName             object\nAlbumId           int64\nMediaTypeId       in
t64\nGenreId           int64\nComposer         object\nMilliseconds      int64\nBytes            int64\nUn
itPrice       float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly cod
e 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': 'qwen2:7b', 'created_at': '2024-06-15T22:43:29.475162966Z', 'message': {'role': 'assistant', 'con
```

tent': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n   fig = px.scatter(\n   df,\n   x=\'TrackId\',\n   y=\'Name\',\n   title="Single Track",\n   labels={\'x\': \'Track ID\', \'y\': \'Track Name\'}\n   )\nelse:\n   fig = px.histogram(\n   df, \n   x=\'TrackId\',\n   y=\'Name\',\n   title="Tracks with Names Containing \'What\'",\n   labels={\'x\': \'Track ID\', \'y\': \'Number of Tracks\'},\n   facet_row="GenreId"\n   )\n   \nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 35585950366, 'load_duration': 595781, 'prompt_eval_count': 209, 'prompt_eval_duration': 13508155000, 'eval_count': 131, 'eval_duration': 21984450000}


Tracks with Names Containing 'What'
NameNameNameNameNameNameNameNameName
What Now My Love
What It Takes
0
1000
2000
3000
sum of TrackId

Out[21]: ("SELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE",

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

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

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

```
 'x': array([  26,   88,  342,  960, 1000, 1440, 1628, 2992, 3007]),
 'xaxis': 'x9',
 'y': array(['What It Takes', 'What You Are', 'What is and Should Never Be',
             'What A Day', 'What If I Do?', "Whatever It Is, I Just Can't Stop",
             'What Is And What Should Never Be',
             "I Still Haven't Found What I'm Looking for",
             "I Still Haven't Found What I'm Looking For"], dtype=object),
 'yaxis': 'y9'},
{'alignmentgroup': 'True',
 'bingroup': 'y',
 'histfunc': 'sum',
 'hovertemplate': 'GenreId=2<br>sum of TrackId=%{x}<br>Name=%{y}<extra></extra>',
 'legendgroup': '',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': '',
 'offsetgroup': '',
 'orientation': 'h',
 'showlegend': False,
 'type': 'histogram',
 'x': array([130, 607]),
 'xaxis': 'x8',
 'y': array(['Do what cha wanna', 'So What'], dtype=object),
 'yaxis': 'y8'},
{'alignmentgroup': 'True',
 'bingroup': 'y',
 'histfunc': 'sum',
 'hovertemplate': 'GenreId=12<br>sum of TrackId=%{x}<br>Name=%{y}<extra></extra>',
 'legendgroup': '',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': '',
 'offsetgroup': '',
 'orientation': 'h',
 'showlegend': False,
 'type': 'histogram',
 'x': array([1039]),
 'xaxis': 'x7',
 'y': array(['What Now My Love'], dtype=object),
 'yaxis': 'y7'},
{'alignmentgroup': 'True',
 'bingroup': 'y',
 'histfunc': 'sum',
 'hovertemplate': 'GenreId=4<br>sum of TrackId=%{x}<br>Name=%{y}<extra></extra>',
```

```
 'legendgroup': '',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': '',
 'offsetgroup': '',
 'orientation': 'h',
 'showlegend': False,
 'type': 'histogram',
 'x': array([1145, 1469, 1470]),
 'xaxis': 'x6',
 'y': array(['Whatsername', "Look What You've Done", 'Get What You Need'],
           dtype=object),
 'yaxis': 'y6'},
{'alignmentgroup': 'True',
 'bingroup': 'y',
 'histfunc': 'sum',
 'hovertemplate': 'GenreId=14<br>sum of TrackId=%{x}<br>Name=%{y}<extra></extra>',
 'legendgroup': '',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': '',
 'offsetgroup': '',
 'orientation': 'h',
 'showlegend': False,
 'type': 'histogram',
 'x': array([1778]),
 'xaxis': 'x5',
 'y': array(["You're What's Happening (In The World Today)"], dtype=object),
 'yaxis': 'y5'},
{'alignmentgroup': 'True',
 'bingroup': 'y',
 'histfunc': 'sum',
 'hovertemplate': 'GenreId=3<br>sum of TrackId=%{x}<br>Name=%{y}<extra></extra>',
 'legendgroup': '',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': '',
 'offsetgroup': '',
 'orientation': 'h',
 'showlegend': False,
 'type': 'histogram',
 'x': array([1823]),
 'xaxis': 'x4',
 'y': array(['So What'], dtype=object),
 'yaxis': 'y4'},
```

```
{'alignmentgroup': 'True',
 'bingroup': 'y',
 'histfunc': 'sum',
 'hovertemplate': 'GenreId=7<br>sum of TrackId=%{x}<br>Name=%{y}<extra></extra>',
 'legendgroup': '',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': '',
 'offsetgroup': '',
 'orientation': 'h',
 'showlegend': False,
 'type': 'histogram',
 'x': array([2772]),
 'xaxis': 'x3',
 'y': array(["I Don't Know What To Do With Myself"], dtype=object),
 'yaxis': 'y3'},
{'alignmentgroup': 'True',
 'bingroup': 'y',
 'histfunc': 'sum',
 'hovertemplate': 'GenreId=19<br>sum of TrackId=%{x}<br>Name=%{y}<extra></extra>',
 'legendgroup': '',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': '',
 'offsetgroup': '',
 'orientation': 'h',
 'showlegend': False,
 'type': 'histogram',
 'x': array([2884, 2893]),
 'xaxis': 'x2',
 'y': array(['What Kate Did', 'Whatever the Case May Be'], dtype=object),
 'yaxis': 'y2'},
{'alignmentgroup': 'True',
 'bingroup': 'y',
 'histfunc': 'sum',
 'hovertemplate': 'GenreId=9<br>sum of TrackId=%{x}<br>Name=%{y}<extra></extra>',
 'legendgroup': '',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': '',
 'offsetgroup': '',
 'orientation': 'h',
 'showlegend': False,
 'type': 'histogram',
 'x': array([3258, 3475]),
```

```
                    'xaxis': 'x',
                    'y': array(['Whatever Gets You Thru the Night', 'What Is It About Men'],
                               dtype=object),
                    'yaxis': 'y'}],
          'layout': {'annotations': [{'font': {},
                                      'showarrow': False,
                                      'text': 'GenreId=9',
                                      'textangle': 90,
                                      'x': 0.98,
                                      'xanchor': 'left',
                                      'xref': 'paper',
                                      'y': 0.042222222222222223,
                                      'yanchor': 'middle',
                                      'yref': 'paper'},
                                     {'font': {},
                                      'showarrow': False,
                                      'text': 'GenreId=19',
                                      'textangle': 90,
                                      'x': 0.98,
                                      'xanchor': 'left',
                                      'xref': 'paper',
                                      'y': 0.15666666666666668,
                                      'yanchor': 'middle',
                                      'yref': 'paper'},
                                     {'font': {},
                                      'showarrow': False,
                                      'text': 'GenreId=7',
                                      'textangle': 90,
                                      'x': 0.98,
                                      'xanchor': 'left',
                                      'xref': 'paper',
                                      'y': 0.27111111111111114,
                                      'yanchor': 'middle',
                                      'yref': 'paper'},
                                     {'font': {},
                                      'showarrow': False,
                                      'text': 'GenreId=3',
                                      'textangle': 90,
                                      'x': 0.98,
                                      'xanchor': 'left',
                                      'xref': 'paper',
                                      'y': 0.3855555555555556,
```

```
 'yanchor': 'middle',
 'yref': 'paper'},
{'font': {},
 'showarrow': False,
 'text': 'GenreId=14',
 'textangle': 90,
 'x': 0.98,
 'xanchor': 'left',
 'xref': 'paper',
 'y': 0.5,
 'yanchor': 'middle',
 'yref': 'paper'},
{'font': {},
 'showarrow': False,
 'text': 'GenreId=4',
 'textangle': 90,
 'x': 0.98,
 'xanchor': 'left',
 'xref': 'paper',
 'y': 0.6144444444444443,
 'yanchor': 'middle',
 'yref': 'paper'},
{'font': {},
 'showarrow': False,
 'text': 'GenreId=12',
 'textangle': 90,
 'x': 0.98,
 'xanchor': 'left',
 'xref': 'paper',
 'y': 0.7288888888888889,
 'yanchor': 'middle',
 'yref': 'paper'},
{'font': {},
 'showarrow': False,
 'text': 'GenreId=2',
 'textangle': 90,
 'x': 0.98,
 'xanchor': 'left',
 'xref': 'paper',
 'y': 0.8433333333333333,
 'yanchor': 'middle',
 'yref': 'paper'},
```

```
                                    {'font': {},
                                     'showarrow': False,
                                     'text': 'GenreId=1',
                                     'textangle': 90,
                                     'x': 0.98,
                                     'xanchor': 'left',
                                     'xref': 'paper',
                                     'y': 0.9577777777777778,
                                     'yanchor': 'middle',
                                     'yref': 'paper'}],
                        'barmode': 'relative',
                        'legend': {'tracegroupgap': 0},
                        'template': '...',
                        'title': {'text': "Tracks with Names Containing 'What'"},
                        'xaxis': {'anchor': 'y', 'domain': [0.0, 0.98], 'title': {'text': 'sum of TrackId'}},
                        'xaxis2': {'anchor': 'y2', 'domain': [0.0, 0.98], 'matches': 'x', 'showticklabels': Fals
e},
                        'xaxis3': {'anchor': 'y3', 'domain': [0.0, 0.98], 'matches': 'x', 'showticklabels': Fals
e},
                        'xaxis4': {'anchor': 'y4', 'domain': [0.0, 0.98], 'matches': 'x', 'showticklabels': Fals
e},
                        'xaxis5': {'anchor': 'y5', 'domain': [0.0, 0.98], 'matches': 'x', 'showticklabels': Fals
e},
                        'xaxis6': {'anchor': 'y6', 'domain': [0.0, 0.98], 'matches': 'x', 'showticklabels': Fals
e},
                        'xaxis7': {'anchor': 'y7', 'domain': [0.0, 0.98], 'matches': 'x', 'showticklabels': Fals
e},
                        'xaxis8': {'anchor': 'y8', 'domain': [0.0, 0.98], 'matches': 'x', 'showticklabels': Fals
e},
                        'xaxis9': {'anchor': 'y9', 'domain': [0.0, 0.98], 'matches': 'x', 'showticklabels': Fals
e},
                        'yaxis': {'anchor': 'x', 'domain': [0.0, 0.08444444444444445], 'title': {'text': 'Name'}},
                        'yaxis2': {'anchor': 'x2',
                                   'domain': [0.11444444444444445, 0.1988888888888889],
                                   'matches': 'y',
                                   'title': {'text': 'Name'}},
                        'yaxis3': {'anchor': 'x3',
                                   'domain': [0.2288888888888889, 0.31333333333333335],
                                   'matches': 'y',
                                   'title': {'text': 'Name'}},
                        'yaxis4': {'anchor': 'x4',
                                   'domain': [0.3433333333333334, 0.4277777777777778],
```

```
                    'matches': 'y',
                    'title': {'text': 'Name'}},
                'yaxis5': {'anchor': 'x5',
                    'domain': [0.4577777777777778, 0.5422222222222223],
                    'matches': 'y',
                    'title': {'text': 'Name'}},
                'yaxis6': {'anchor': 'x6',
                    'domain': [0.5722222222222222, 0.6566666666666666],
                    'matches': 'y',
                    'title': {'text': 'Name'}},
                'yaxis7': {'anchor': 'x7',
                    'domain': [0.6866666666666668, 0.7711111111111112],
                    'matches': 'y',
                    'title': {'text': 'Name'}},
                'yaxis8': {'anchor': 'x8',
                    'domain': [0.8011111111111111, 0.8855555555555555],
                    'matches': 'y',
                    'title': {'text': 'Name'}},
                'yaxis9': {'anchor': 'x9', 'domain': [0.9155555555555556, 1.0], 'matches': 'y', 'title':
{'text': 'Name'}}}
 }))
```

In [22]:
```
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 5, updating n_results = 5
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

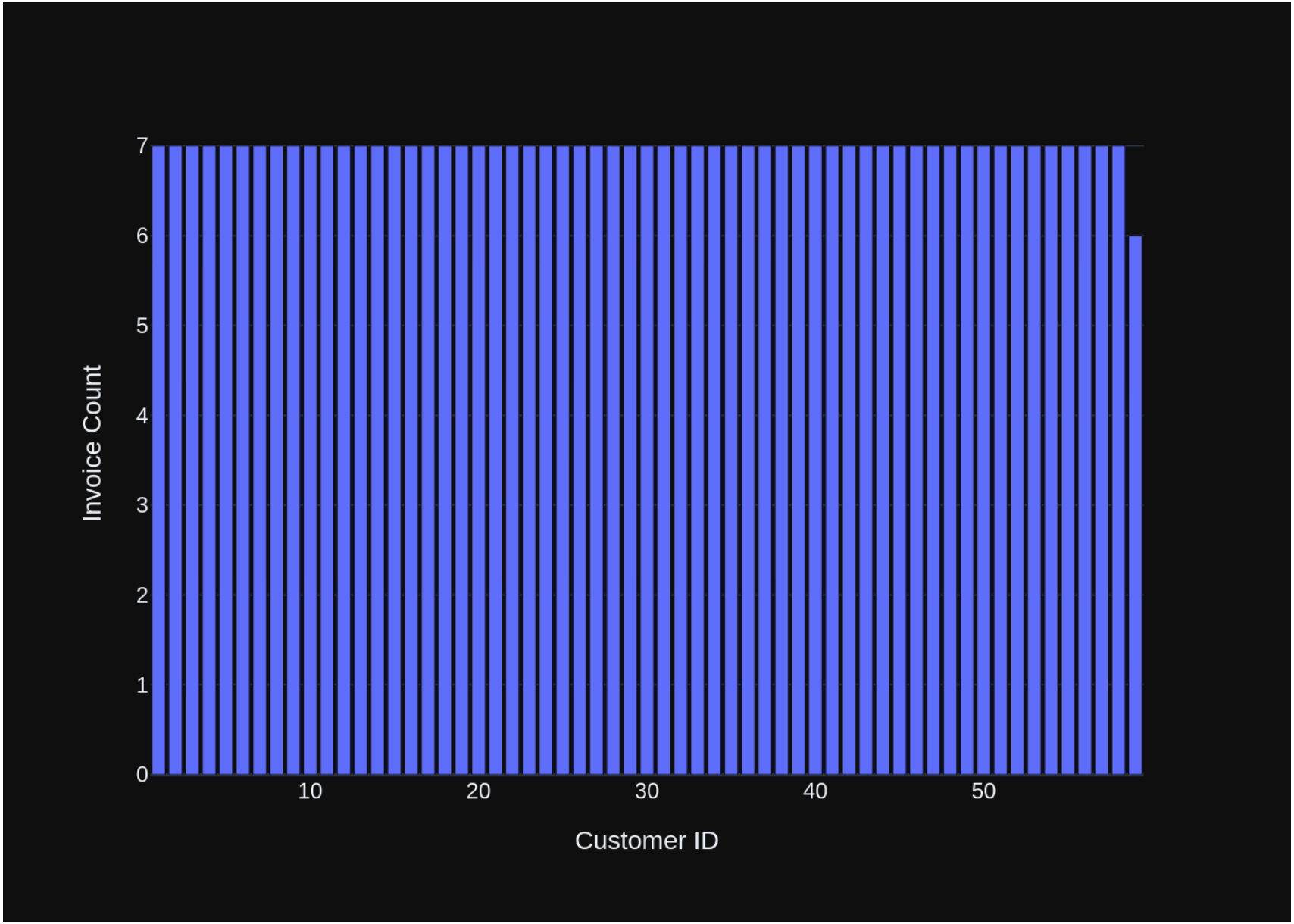
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY 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_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 chinook 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(*) AS total_customers FROM customers'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*)

as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, {'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 parameters:
model=qwen2:7b,
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 \"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   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 chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is s
ufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provide
d context is almost sufficient but requires knowledge of a specific string in a particular column, please g
enerate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a com
ment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be g
enerated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered befo
re, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "How many cu
stomers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) AS total_customers FROM customers"},
{"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistan
t", "content": "SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY c
ustomer_count DESC \nLIMIT 5"}, {"role": "user", "content": "  \n    List all albums and their correspondin
g artist names  \n"}, {"role": "assistant", "content": "SELECT a.Title, ar.Name AS ArtistName\nFROM albums
a\nJOIN artists ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "  \n    Find all tracks with
a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks W
HERE Name LIKE '%What%' COLLATE NOCASE"}, {"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    Get the total number of invoices for each customer\n"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:44:51.47757093Z', 'message': {'role': 'assistant', 'cont
ent': 'SELECT CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId'}, 'done_reason':
'stop', 'done': True, 'total_duration': 81745678020, 'load_duration': 1050284, 'prompt_eval_count': 1178,
'prompt_eval_duration': 78378589000, 'eval_count': 18, 'eval_duration': 3000291000}
SELECT CustomerId, COUNT(*) as invoice_count
FROM invoices
GROUP BY CustomerId
SELECT CustomerId, COUNT(*) as invoice_count
FROM invoices
GROUP BY CustomerId
    CustomerId  invoice_count
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=qwen2:7b,
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 CustomerId, COUNT(*) as invoice_count \nFROM invoices
\nGROUP BY CustomerId\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning
df.dtypes gives:\n CustomerId       int64\ninvoice_count    int64\ndtype: object"}, {"role": "user", "conte
nt": "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': 'qwen2:7b', 'created_at': '2024-06-15T22:45:19.22747545Z', 'message': {'role': 'assistant', 'cont
ent': '```python\nimport plotly.express as px\n\nif len(df) == 1:\n    fig = px.indicators.Value(\n
title="Total Invoices for Customer",\n        label="invoice_count",\n        value=df[\'invoice_count\'].i
loc[0]\n    )\nelse:\n    fig = px.bar(\n        df,\n        x=\'CustomerId\',\n        y=\'invoice_count
\',\n        labels={\'CustomerId\': \'Customer ID\', \'invoice_count\': \'Invoice Count\'}\n    )\n\nfig.s
how()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 27724513381, 'load_duration': 582837,
'prompt_eval_count': 166, 'prompt_eval_duration': 10686941000, 'eval_count': 101, 'eval_duration': 16943003
000}
```


Invoice Count
0
1
2
3
4
5
6
7
10
20
30
40
50
Customer ID

Out[22]: ('SELECT CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId',

	CustomerId	invoice_count
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': 'Customer ID=%{x}<br>Invoice Count=%{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, 7, 7, 6]),
              'yaxis': 'y'}],
    'layout': {'barmode': 'relative',
               'legend': {'tracegroupgap': 0},
               'margin': {'t': 60},
```

```
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Customer ID'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Invoice Count'}}}
    }))
```

In [23]:
```
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 6, updating n_results = 6
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

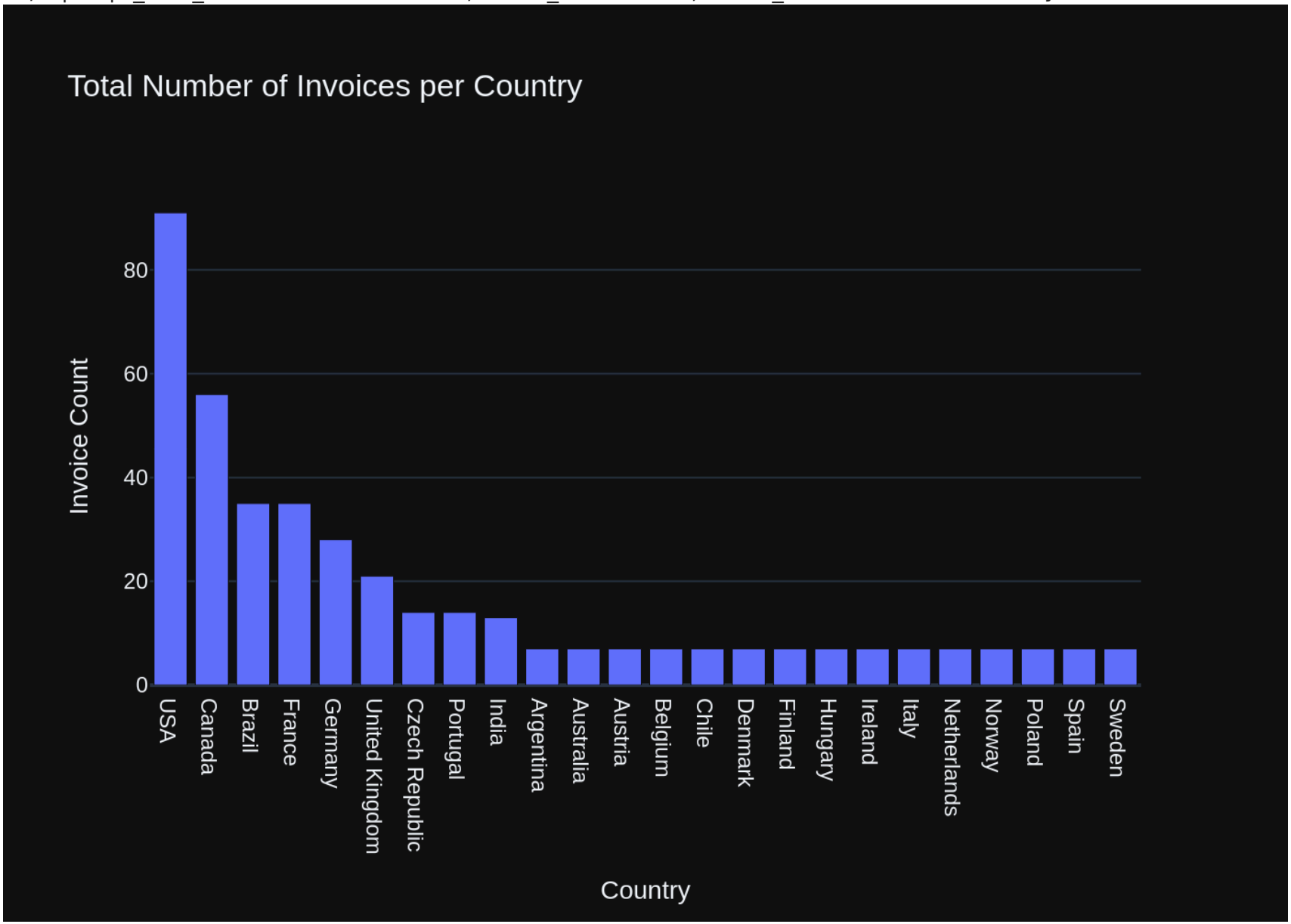
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY 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 chinook 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 CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGR
OUP BY CustomerId'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'},
{'role': 'assistant', 'content': 'SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Co
untry \nORDER BY customer_count DESC \nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are ther
e'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) AS total_customers FROM customers'}, {'role': 'use
r', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant',
'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.Artist
Id'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)
\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, {'r
ole': '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 parameters:
model=qwen2:7b,
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 c
hinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, p
lease generate a valid SQL query without any explanations for the question. \n2. If the provided context is
almost sufficient but requires knowledge of a specific string in a particular column, please generate an in
termediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying i
ntermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n
4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please r
epeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "  \n    Get the total n
umber of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) as i
nvoice_count \nFROM invoices \nGROUP BY CustomerId"}, {"role": "user", "content": "what are the top 5 count
ries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) as customer_co
unt \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5"}, {"role": "user", "conte
nt": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) AS total_customers
FROM customers"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist names
\n"}, {"role": "assistant", "content": "SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists
ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "  \n    Find all tracks with a name containi
ng \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks WHERE Name LIKE
'%What%' COLLATE NOCASE"}, {"role": "user", "content": "Can you list all tables in the SQLite database cata
log?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "us
er", "content": "  \n    Find the total number of invoices per country:\n"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:46:02.276878983Z', 'message': {'role': 'assistant', 'con
tent': 'SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices  \nGROUP BY BillingCountry'}, 'don
e_reason': 'stop', 'done': True, 'total_duration': 42953563142, 'load_duration': 676999, 'prompt_eval_coun
t': 1263, 'prompt_eval_duration': 39516787000, 'eval_count': 18, 'eval_duration': 3021893000}
SELECT BillingCountry, COUNT(*) as invoice_count
FROM invoices
GROUP BY BillingCountry
SELECT BillingCountry, COUNT(*) as invoice_count
FROM invoices
GROUP BY BillingCountry
    BillingCountry  invoice_count
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=qwen2:7b,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Find the total number of invoices per country:\n'\n\nThe
DataFrame was produced using this query: SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices
\nGROUP BY BillingCountry\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunn
ing df.dtypes gives:\n BillingCountry    object\ninvoice_count      int64\ndtype: object"}, {"role": "use
r", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the d
ata is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Re
spond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:46:32.935563991Z', 'message': {'role': 'assistant', 'con
tent': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n    fig = px.indicators.Value(\n
title="Total Invoices",\n        value=df[\'invoice_count\'].iloc[0],\n        label="Invoice Count"\n    )
\nelse:\n    fig = px.bar(df, x=\'BillingCountry\', y=\'invoice_count\', \n                 title=\'Total N
umber of Invoices per Country\', labels={\'BillingCountry\': \'Country\', \'invoice_count\': \'Invoice Coun
```

t\'})\n   \nfig.update_layout(xaxis_categoryorder=\'total descending\')\n\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 30634426916, 'load_duration': 44151680, 'prompt_eval_count': 163, 'prompt_eval_duration': 10499188000, 'eval_count': 119, 'eval_duration': 20002219000}



Out[23]:
```
('SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices  \nGROUP BY BillingCountry',
    BillingCountry  invoice_count
0        Argentina              7
1        Australia              7
2          Austria              7
3          Belgium              7
4           Brazil             35
5           Canada             56
6            Chile              7
7   Czech Republic             14
8          Denmark              7
9          Finland              7
10          France             35
11         Germany             28
12         Hungary              7
13           India             13
14         Ireland              7
15           Italy              7
16     Netherlands              7
17          Norway              7
18          Poland              7
19        Portugal             14
20           Spain              7
21          Sweden              7
22             USA             91
23  United Kingdom             21,
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'Country=%{x}<br>Invoice Count=%{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 Number of Invoices per Country'},
                    'xaxis': {'anchor': 'y',
                              'categoryorder': 'total descending',
                              'domain': [0.0, 1.0],
                              'title': {'text': 'Country'}},
                    'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Invoice Count'}}}
          }))
```

In [24]:
```
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 7, updating n_results = 7
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "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 chinook 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 CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'rol

e': 'assistant', 'content': 'SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY B
illingCountry'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'conten
t': 'SELECT COUNT(*) AS total_customers FROM customers'}, {'role': 'user', 'content': 'what are the top 5 c
ountries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) as custome
r_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5'}, {'role': 'user', 'c
ontent': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'conten
t': 'SELECT a.Title, 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 * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, {'role':
'user', 'content': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'assistant', 'conte
nt': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': ' \n   List all in
voices with a total exceeding $10:\n'}]
Ollama parameters:
model=qwen2:7b,
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

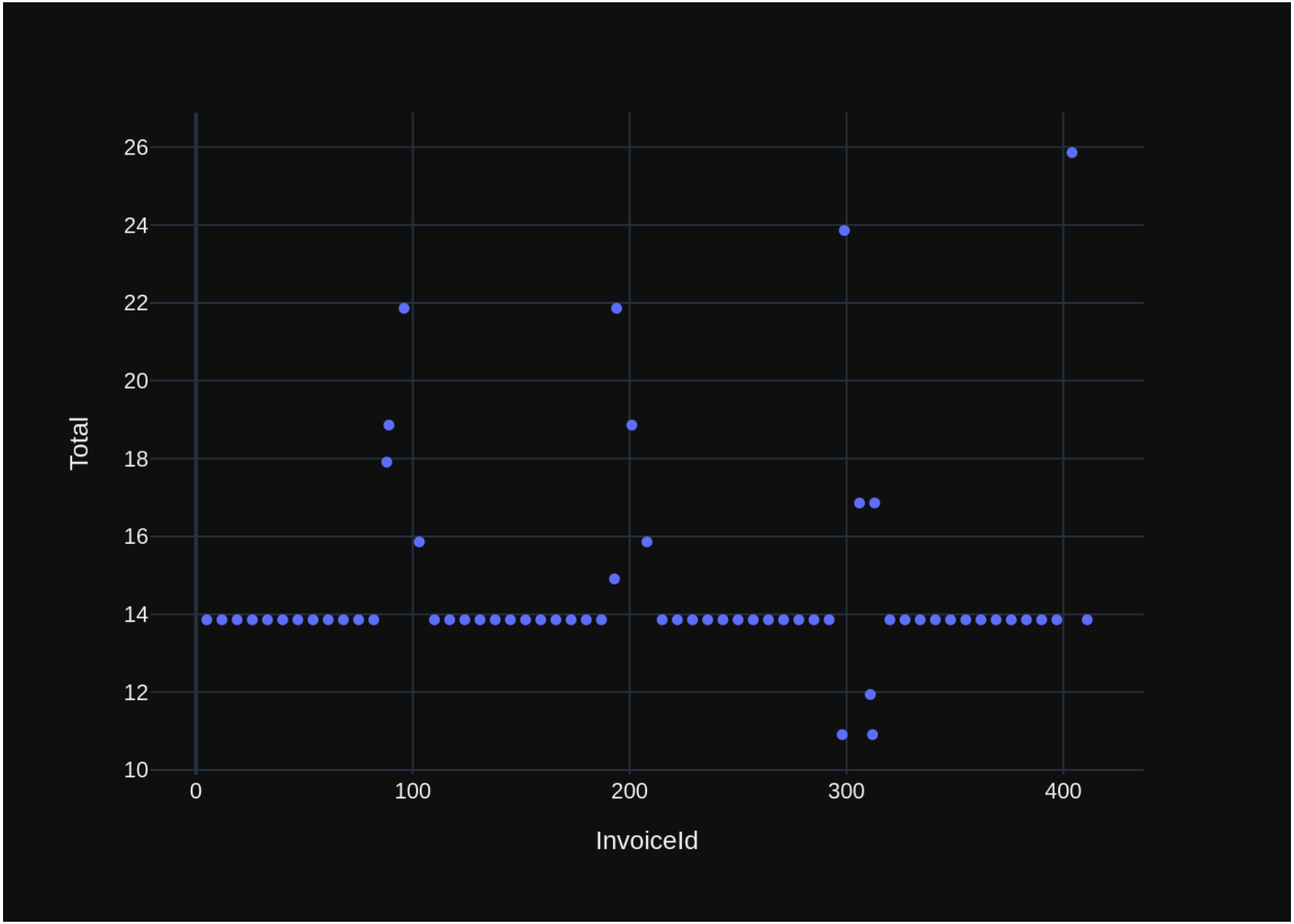
```
\n    FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION\r\n)\n\nCREATE TABLE \"employees\"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT
NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT 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 chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficie
nt, please generate a valid SQL query without any explanations for the question. \n2. If the provided conte
xt 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    Get the to
tal number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*)
as invoice_count \nFROM invoices \nGROUP BY CustomerId"}, {"role": "user", "content": "  \n    Find the tot
al number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT BillingCountry, COUNT(*) a
s invoice_count \nFROM invoices  \nGROUP BY BillingCountry"}, {"role": "user", "content": "How many custome
rs are there"}, {"role": "assistant", "content": "SELECT COUNT(*) AS total_customers FROM customers"}, {"ro
le": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "c
ontent": "SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY custome
r_count DESC \nLIMIT 5"}, {"role": "user", "content": "  \n    List all albums and their corresponding arti
st names  \n"}, {"role": "assistant", "content": "SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOI
N artists ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "  \n    Find all tracks with a nam
e containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks WHERE
Name LIKE '%What%' COLLATE NOCASE"}, {"role": "user", "content": "Can you list all tables in the SQLite dat
abase catalog?"}, {"role": "assistant", "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': 'qwen2:7b', 'created_at': '2024-06-15T22:47:59.471057437Z', 'message': {'role': 'assistant', 'con
tent': 'SELECT * FROM invoices WHERE Total > 10'}, 'done_reason': 'stop', 'done': True, 'total_duration': 8
6425065539, 'load_duration': 1024927, 'prompt_eval_count': 1254, 'prompt_eval_duration': 84151428000, 'eval
_count': 11, 'eval_duration': 1768959000}
SELECT * FROM invoices WHERE Total > 10
SELECT * FROM invoices WHERE Total > 10
    InvoiceId  CustomerId          InvoiceDate             BillingAddress  \
0           5          23  2009-01-11 00:00:00            69 Salem Street
1          12           2  2009-02-11 00:00:00    Theodor-Heuss-Straße 34
2          19          40  2009-03-14 00:00:00             8, Rue Hanovre
3          26          19  2009-04-14 00:00:00            1 Infinite Loop
4          33          57  2009-05-15 00:00:00            Calle Lira, 198
```

```
..        ...         ...                  ...                          ...
59        383          10  2013-08-12 00:00:00  Rua Dr. Falcão Filho, 155
60        390          48  2013-09-12 00:00:00      Lijnbaansgracht 120bg
61        397          27  2013-10-13 00:00:00            1033 N Park Ave
62        404           6  2013-11-13 00:00:00              Rilská 3174/6
63        411          44  2013-12-14 00:00:00            Porthaninkatu 9

   BillingCity BillingState  BillingCountry BillingPostalCode  Total
0       Boston           MA             USA              2113  13.86
1    Stuttgart         None         Germany             70174  13.86
2        Paris         None          France             75002  13.86
3    Cupertino           CA             USA             95014  13.86
4     Santiago         None           Chile              None  13.86
..         ...          ...             ...               ...    ...
59   São Paulo           SP          Brazil         01007-010  13.86
60   Amsterdam           VV     Netherlands              1016  13.86
61      Tucson           AZ             USA             85719  13.86
62      Prague         None  Czech Republic             14300  25.86
63    Helsinki         None         Finland             00530  13.86

[64 rows x 9 columns]
Ollama parameters:
model=qwen2:7b,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all invoices with a total exceeding $10:\n'\n\nThe
DataFrame was produced using this query: SELECT * FROM invoices WHERE Total > 10\n\nThe following is inform
ation about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n InvoiceId              int64
\nCustomerId             int64\nInvoiceDate           object\nBillingAddress        object\nBillingCity
object\nBillingState          object\nBillingCountry        object\nBillingPostalCode     object\nTotal
float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the
results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value
in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- j
ust the code."}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:48:19.443751249Z', 'message': {'role': 'assistant', 'con
tent': '```python\nimport plotly.express as px\n\nif df.shape[0] > 1:\n    fig = px.scatter(df, x=\'Invoice
Id\', y=\'Total\')\nelse:\n    fig = px.indicators.number(value=df[\'Total\'].values[0], title="Invoice Tot
al")\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 19944497328, 'load_duratio
```

n': 673140, 'prompt_eval_count': 196, 'prompt_eval_duration': 9046190000, 'eval_count': 65, 'eval_duratio
n': 10758895000}



Out[24]:
```
('SELECT * FROM invoices WHERE Total > 10',
    InvoiceId  CustomerId          InvoiceDate             BillingAddress  \
0           5          23  2009-01-11 00:00:00            69 Salem Street
1          12           2  2009-02-11 00:00:00    Theodor-Heuss-Straße 34
2          19          40  2009-03-14 00:00:00             8, Rue Hanovre
3          26          19  2009-04-14 00:00:00            1 Infinite Loop
4          33          57  2009-05-15 00:00:00            Calle Lira, 198
..        ...         ...                  ...                        ...
59        383          10  2013-08-12 00:00:00  Rua Dr. Falcão Filho, 155
60        390          48  2013-09-12 00:00:00      Lijnbaansgracht 120bg
61        397          27  2013-10-13 00:00:00            1033 N Park Ave
62        404           6  2013-11-13 00:00:00              Rilská 3174/6
63        411          44  2013-12-14 00:00:00            Porthaninkatu 9

   BillingCity BillingState  BillingCountry BillingPostalCode  Total
0       Boston           MA             USA              2113  13.86
1    Stuttgart         None         Germany             70174  13.86
2        Paris         None          France             75002  13.86
3    Cupertino           CA             USA             95014  13.86
4     Santiago         None           Chile              None  13.86
..         ...          ...             ...               ...    ...
59   São Paulo           SP          Brazil         01007-010  13.86
60   Amsterdam           VV     Netherlands              1016  13.86
61      Tucson           AZ             USA             85719  13.86
62      Prague         None  Czech Republic             14300  25.86
63    Helsinki         None         Finland             00530  13.86

[64 rows x 9 columns],
Figure({
    'data': [{'hovertemplate': 'InvoiceId=%{x}<br>Total=%{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([13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                            13.86, 13.86, 17.91, 18.86, 21.86, 15.86, 13.86, 13.86, 13.86, 13.86,
                            13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 14.91, 21.86,
                            18.86, 15.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                            13.86, 13.86, 13.86, 13.86, 10.91, 23.86, 16.86, 11.94, 10.91, 16.86,
                            13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                            13.86, 13.86, 25.86, 13.86]),
                'yaxis': 'y'}],
     'layout': {'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total'}}}
 }))
```

In [25]:

```
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 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 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 chinook 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 intermedia

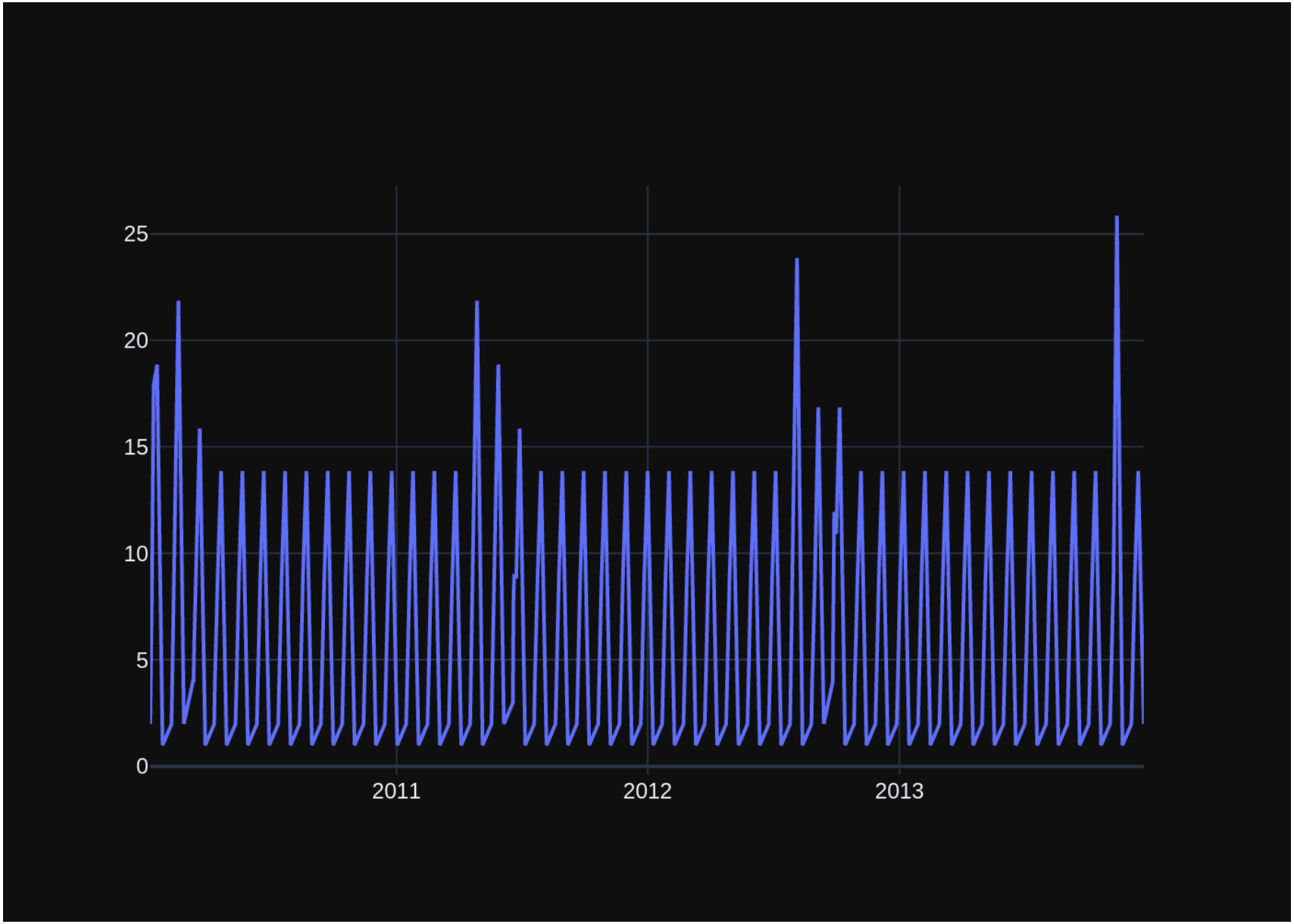
te_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCountry'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) AS total_customers FROM customers'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, {'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'}]
Ollama parameters:
model=qwen2:7b,
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 \"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 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 chinook database invoice means order\n\n===Response Guidelines \n1. If the provid
ed context is sufficient, please generate a valid SQL query without any explanations for the question. \n2.
If the provided context is almost sufficient but requires knowledge of a specific string in a particular co
lumn, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the qu
ery with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why
it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and
answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content":
" \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM
invoices WHERE Total > 10"}, {"role": "user", "content": " \n   Find the total number of invoices per cou
ntry:\n"}, {"role": "assistant", "content": "SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoic
es \nGROUP BY BillingCountry"}, {"role": "user", "content": " \n   Get the total number of invoices for
each customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) as invoice_count \nFROM in
voices \nGROUP BY CustomerId"}, {"role": "user", "content": "How many customers are there"}, {"role": "assi
stant", "content": "SELECT COUNT(*) AS total_customers FROM customers"}, {"role": "user", "content": "what
are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUN
T(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5"}, {"ro
le": "user", "content": " \n   List all albums and their corresponding artist names \n"}, {"role": "assi
stant", "content": "SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = a
r.ArtistId"}, {"role": "user", "content": " \n   Find all tracks with a name containing \"What\" (case-in
sensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOC
ASE"}, {"role": "user", "content": "Can you list all tables in the SQLite database catalog?"}, {"role": "as
sistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "content": "

```
\n    Find all invoices since 2010 and the total amount invoiced:\n"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:49:59.708495467Z', 'message': {'role': 'assistant', 'con
tent': "SELECT InvoiceDate, Total \nFROM invoices  \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDat
e"}, 'done_reason': 'stop', 'done': True, 'total_duration': 100163144938, 'load_duration': 863582, 'prompt_
eval_count': 1416, 'prompt_eval_duration': 94544052000, 'eval_count': 30, 'eval_duration': 5066249000}
SELECT InvoiceDate, Total
FROM invoices
WHERE InvoiceDate >= '2010-01-01'
GROUP BY InvoiceDate
SELECT InvoiceDate, Total
FROM invoices
WHERE InvoiceDate >= '2010-01-01'
GROUP BY InvoiceDate
             InvoiceDate  Total
0    2010-01-08 00:00:00   1.98
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=qwen2:7b,
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 InvoiceDate, Total \nFROM invoices  \nWHERE I
nvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate\n\nThe following is information about the resulting pandas
DataFrame 'df': \nRunning df.dtypes gives:\n InvoiceDate     object\nTotal          float64\ndtype: objec
t"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataf
rame? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, us
e an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
```

{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:50:28.460857905Z', 'message': {'role': 'assistant', 'con
tent': '```\nimport plotly.graph_objs as go\n\nif df.shape[0] == 1:\n   fig = go.Figure(go.Indicator(\n
mode="number",\n   value=df[\'Total\'].iloc[0],\n   title={"text": "Total Amount Invoiced Sinc
e 2010"},\n   ))\nelse:\n   data = [go.Scatter(x=df[\'InvoiceDate\'], y=df[\'Total\'])]\n   fig = g
o.Figure(data)\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 28725522038, 'load_duration':
41952492, 'prompt_eval_count': 181, 'prompt_eval_duration': 11756848000, 'eval_count': 100, 'eval_duratio
n': 16833696000}


25
20
15
10
5
0
2011
2012
2013

Out[25]:
```
("SELECT InvoiceDate, Total \nFROM invoices  \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate",
             InvoiceDate  Total
0    2010-01-08 00:00:00   1.98
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': [{'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),
              'y': array([ 1.98,  3.96,  6.94, ...,  8.91, 13.86,  1.99])}],
    'layout': {'template': '...'}
})
```

In [26]:
```
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 9, updating n_results = 9
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

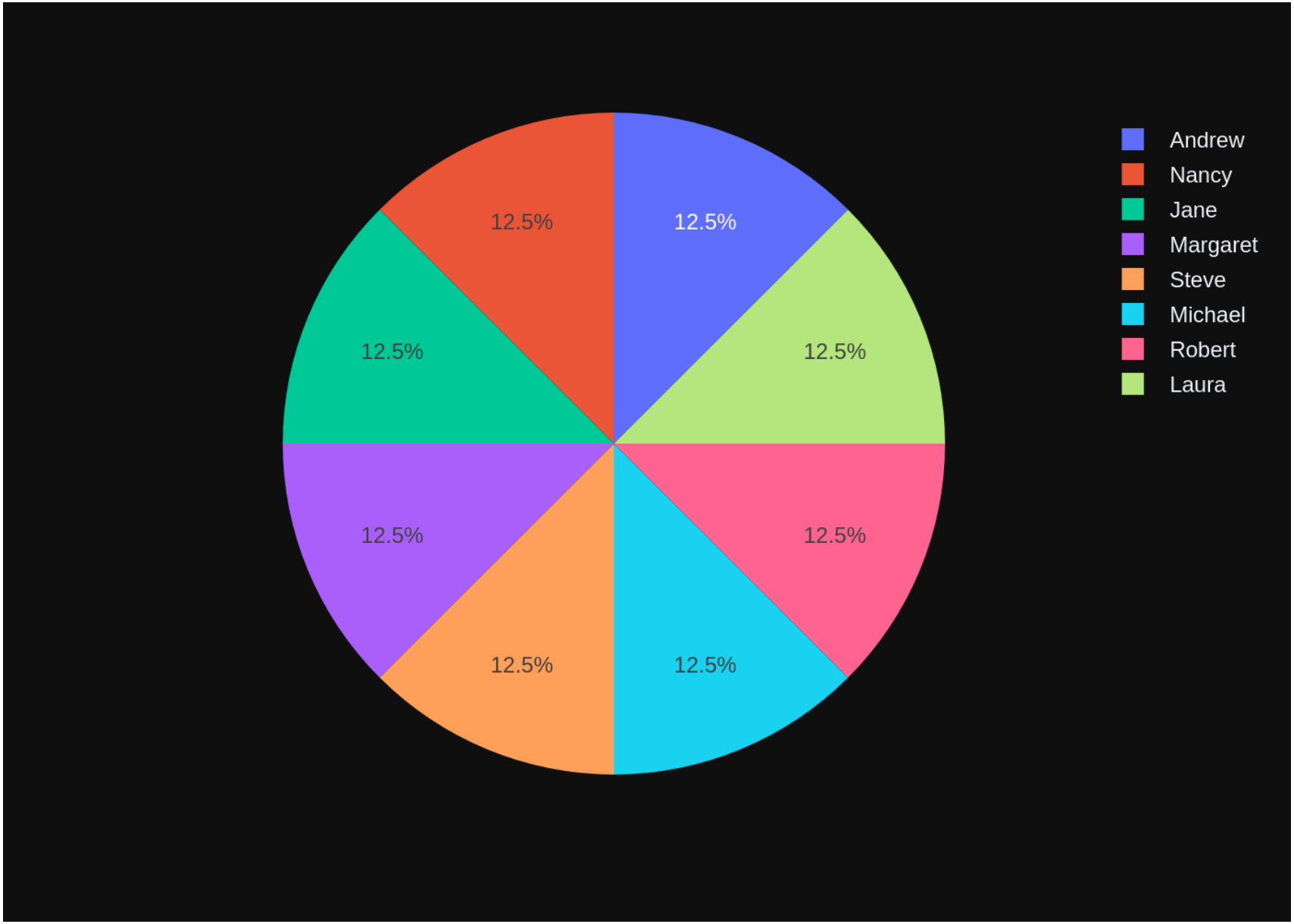
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE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

l invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceDate, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCountry'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 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'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) AS total_customers FROM customers'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, {'role': 'user', 'content': " \n   List all employees and their reporting manager's name (if any):\n"}]
Ollama parameters:
model=qwen2:7b,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_EmployeeReportsTo ON \"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 chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceDate, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5"}, {"role": "user", "content": " \n   Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCountry"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total > 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"}, {"role": "assistant", "content": "SELECT COUNT(*) AS total_customers FROM customers"}, {"role": "user", "content": " \n   Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, {"role": "user", "content": " \n   List all employees and their reporting manager's name (if any):\n"}]
Ollama Response:

{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:51:58.948982921Z', 'message': {'role': 'assistant', 'con
tent': 'SELECT e1.FirstName, e2.FirstName AS ReportingManagerName \nFROM employees e1 \nLEFT JOIN employee
s e2 ON e1.ReportsTo = e2.EmployeeId'}, 'done_reason': 'stop', 'done': True, 'total_duration': 90399813700,
'load_duration': 1055872, 'prompt_eval_count': 1396, 'prompt_eval_duration': 83732452000, 'eval_count': 35,
'eval_duration': 6077916000}
```
SELECT e1.FirstName, e2.FirstName AS ReportingManagerName
FROM employees e1
LEFT JOIN employees e2 ON e1.ReportsTo = e2.EmployeeId
```
```
SELECT e1.FirstName, e2.FirstName AS ReportingManagerName
FROM employees e1
LEFT JOIN employees e2 ON e1.ReportsTo = e2.EmployeeId
```
```
  FirstName ReportingManagerName
0    Andrew                 None
1     Nancy               Andrew
2      Jane                Nancy
3  Margaret                Nancy
4     Steve                Nancy
5   Michael               Andrew
6    Robert              Michael
7     Laura              Michael
```
```
Ollama parameters:
model=qwen2:7b,
options={},
keep_alive=None
```
```
Prompt Content:
```
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: ' \n   List all employees and their reporting manager's name (i
f any):\n'\n\nThe DataFrame was produced using this query: SELECT e1.FirstName, e2.FirstName AS ReportingMa
nagerName \nFROM employees e1 \nLEFT JOIN employees e2 ON e1.ReportsTo = e2.EmployeeId\n\nThe following is
information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n FirstName
object\nReportingManagerName   object\ndtype: object"}, {"role": "user", "content": "Can you generate the
Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called
'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not
answer with any explanations -- just the code."}]
```
Ollama Response:
```
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:52:26.397824737Z', 'message': {'role': 'assistant', 'con
tent': "```\nimport plotly.express as px\n\nif df.shape[0] > 1:\n   fig = px.treemap(df, path=['FirstNam
e', 'ReportingManagerName'], \n   values='FirstName',\n   color='FirstN
ame',\n   color_continuous_scale='Viridis')\nelse:\n   fig = px.indicators.Data(\n
dataframe=df,\n   value='FirstName',\n   title='Employee',\n   label='Employee'\n   )\n\nfi
g.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 27423461421, 'load_duration': 53030

9, 'prompt_eval_count': 182, 'prompt_eval_duration': 11765214000, 'eval_count': 93, 'eval_duration': 155659
98000}



Out[26]:
```
('SELECT e1.FirstName, e2.FirstName AS ReportingManagerName  \nFROM employees e1 \nLEFT JOIN employees e2
ON e1.ReportsTo = e2.EmployeeId',
   FirstName ReportingManagerName
 0    Andrew                 None
 1     Nancy               Andrew
 2      Jane                Nancy
 3  Margaret                Nancy
 4     Steve                Nancy
 5   Michael               Andrew
 6    Robert              Michael
 7     Laura              Michael,
 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 [27]:
```
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
```

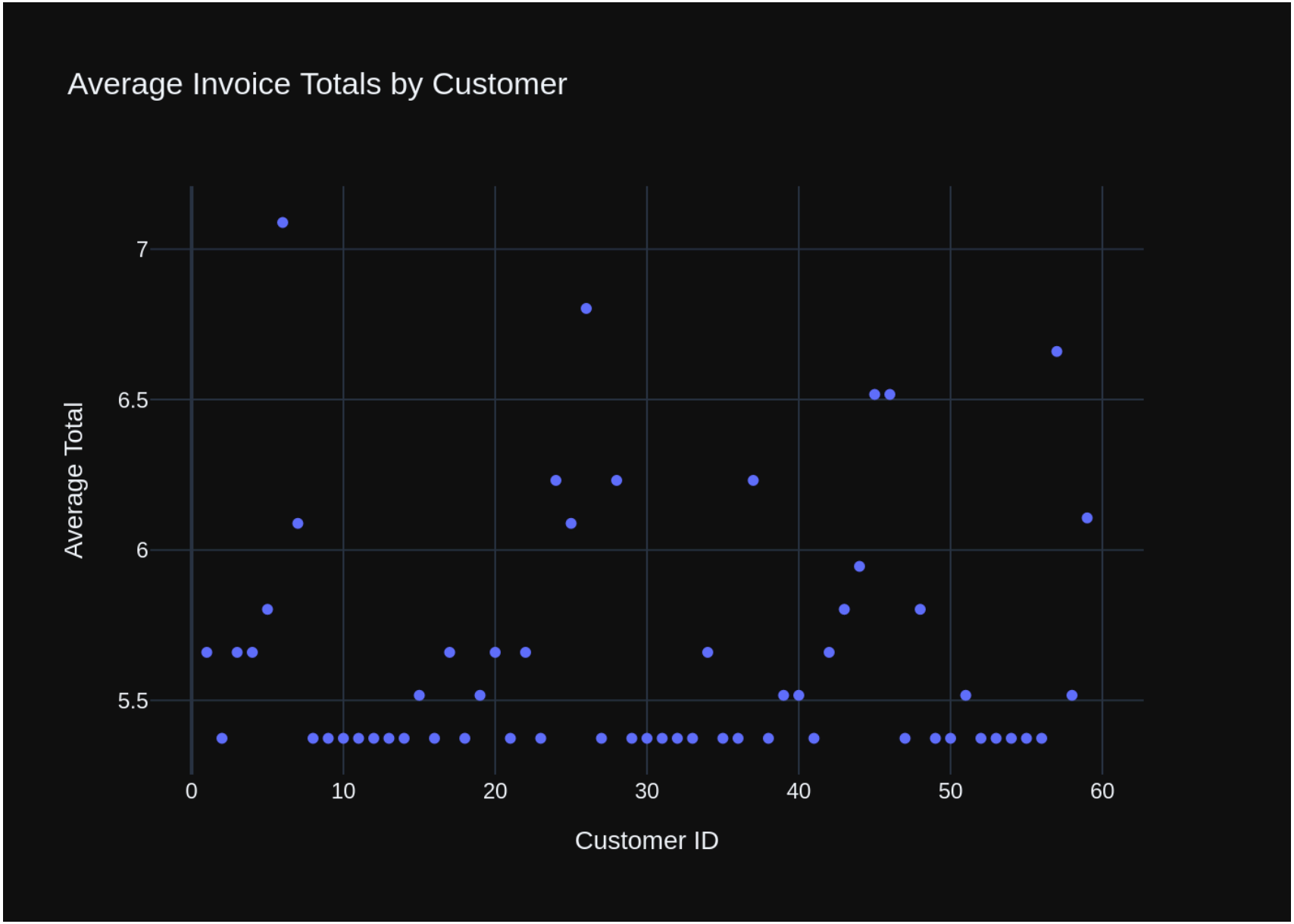
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY 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 chinook 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 CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceDate, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCountry'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT

(*) AS total_customers FROM customers'}, {'role': 'user', 'content': 'what are the top 5 countries that cus
tomers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) as customer_count \nFROM cu
stomers \nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5'}, {'role': 'user', 'content': " \n
List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': 'SELE
CT e1.FirstName, e2.FirstName AS ReportingManagerName \nFROM employees e1 \nLEFT JOIN employees e2 ON e1.R
eportsTo = e2.EmployeeId'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding ar
tist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJ
OIN artists ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find all tracks with a n
ame containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE
Name LIKE '%What%' COLLATE NOCASE"}, {'role': 'user', 'content': 'Can you list all tables in the SQLite dat
abase catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"},
{'role': 'user', 'content': ' \n   Get the average invoice total for each customer:\n'}]
Ollama parameters:
model=qwen2:7b,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   Billin
gAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCou
ntry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FORE
IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineInv
oiceId ON \"invoice_items\" (InvoiceId)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n   InvoiceLineId INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r
\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId)
REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (Tr
ackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX
IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE
INDEX IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"customers\"\r\n(\r\n   Cu
stomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   LastNa
me NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(4
0),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NV
ARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r\n
FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO A
CTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON \"employees\" (ReportsTo)\n\nCREATE TABLE \"employees
\"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NUL
L,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   Bir
thDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   St
ate NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r

\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook 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 CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceDate, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCountry"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) AS total_customers FROM customers"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5"}, {"role": "user", "content": " \n   List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT e1.FirstName, e2.FirstName AS ReportingManagerName \nFROM employees e1 \nLEFT JOIN employees e2 ON e1.ReportsTo = e2.EmployeeId"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, {"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 average invoice total for each customer:\n"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:53:53.099945045Z', 'message': {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) as avg_total \nFROM invoices \nGROUP BY CustomerId'}, 'done_reason': 'stop', 'done': True, 'total_duration': 86592428705, 'load_duration': 661458, 'prompt_eval_count': 1244, 'prompt_eval_duration': 82592276000, 'eval_count': 20, 'eval_duration': 3349044000}
SELECT CustomerId, AVG(Total) as avg_total
FROM invoices
GROUP BY CustomerId
SELECT CustomerId, AVG(Total) as avg_total
FROM invoices
GROUP BY CustomerId
    CustomerId avg_total

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=qwen2:7b,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Get the average invoice total for each customer:\n'\n\nT
he DataFrame was produced using this query: SELECT CustomerId, AVG(Total) as avg_total \nFROM invoices \nGR
OUP BY CustomerId\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.d
types gives:\n CustomerId      int64\navg_total     float64\ndtype: object"}, {"role": "user", "content":
"Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pan
das dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only
Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:54:24.712885563Z', 'message': {'role': 'assistant', 'con
tent': '```\nimport plotly.express as px\n\nif len(df) == 1:\n    fig = px.indicators.Scatter(\n        x=d
f[\'CustomerId\'],\n        y=[df[\'avg_total\'].iloc[0]],\n        title=\'Average Invoice Total for Custo
mer\',\n        labels={\'x\': \'Customer ID\', \'y\': \'Average Total\'}\n    )\nelse:\n    fig = px.scatt
er(df, x=\'CustomerId\', y=\'avg_total\', \n                      title=\'Average Invoice Totals by Customer
\')\n    \nfig.update_layout(xaxis_title="Customer ID", yaxis_title="Average Total")\nfig.show()\n```'}, 'd
one_reason': 'stop', 'done': True, 'total_duration': 31585282845, 'load_duration': 597486, 'prompt_eval_cou
nt': 167, 'prompt_eval_duration': 10722861000, 'eval_count': 124, 'eval_duration': 20763430000}
```


Average Invoice Totals by Customer
Average Total
7
6.5
6
5.5
0
10
20
30
40
50
60
Customer ID

Out[27]: ('SELECT CustomerId, AVG(Total) as avg_total \nFROM invoices \nGROUP BY CustomerId',

	CustomerId	avg_total
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': [{'hovertemplate': 'CustomerId=%{x}<br>avg_total=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'symbol': 'circle'},
              'mode': 'markers',
              'name': '',
              'orientation': 'v',
              'showlegend': False,
              'type': 'scatter',
              'x': array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
                          19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
                          37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
                          55, 56, 57, 58, 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': {'legend': {'tracegroupgap': 0},
                 'template': '...',
                 'title': {'text': 'Average Invoice Totals by Customer'},
                 'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Customer ID'}},
                 'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Average Total'}}}
 }))
```

In [28]:
```
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 chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceDate, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5'}, {'role': 'user', 'content': ' \n   Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) as avg_total \nFROM invoices \nGROUP BY CustomerId'},

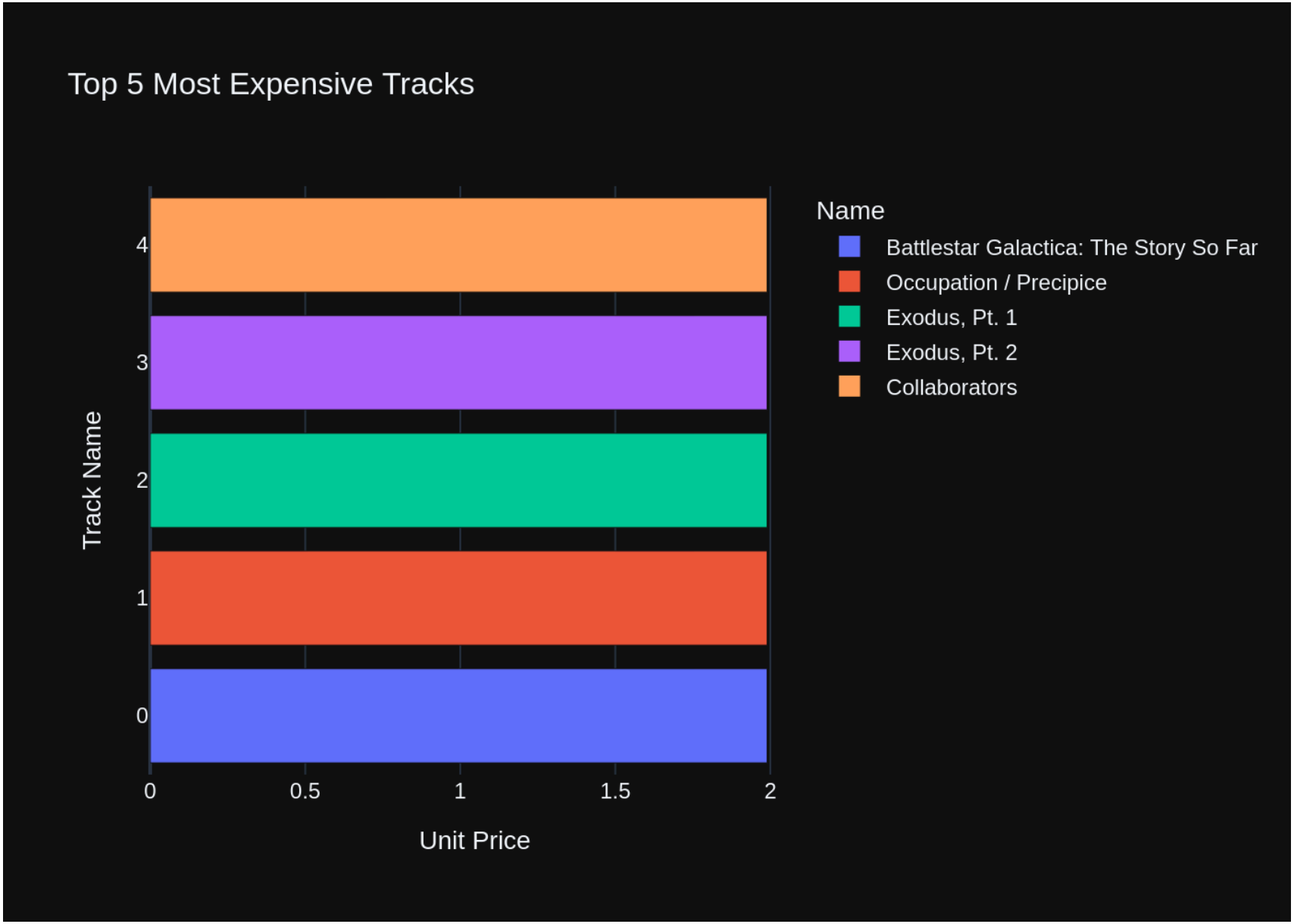
{'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistan
t', 'content': 'SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCountr
y'}, {'role': 'user', 'content': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'assi
stant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': ' \n
Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerI
d, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId'}, {'role': 'user', 'content': 'How many
customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) AS total_customers FROM customer
s'}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}]
Ollama parameters:
model=qwen2:7b,
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 chinook database invoice
means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid S
QL query without any explanations for the question. \n2. If the provided context is almost sufficient but r
equires knowledge of a specific string in a particular column, please generate an intermediate SQL query to
find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If
the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most re

levant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n   Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceDate, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5"}, {"role": "user", "content": " \n   Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT CustomerId, AVG(Total) as avg_total \nFROM invoices \nGROUP BY CustomerId"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCountry"}, {"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"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) AS total_customers FROM customers"}, {"role": "user", "content": " \n   Find the top 5 most expensive tracks (based on unit price):\n"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:55:29.574037728Z', 'message': {'role': 'assistant', 'content': 'SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5'}, 'done_reason': 'stop', 'done': True, 'total_duration': 64753329926, 'load_duration': 675678, 'prompt_eval_count': 1130, 'prompt_eval_duration': 61978617000, 'eval_count': 13, 'eval_duration': 2129581000}
SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5
SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5

	TrackId	Name	AlbumId	MediaTypeId	\
0	2819	Battlestar Galactica: The Story So Far	226	3	
1	2820	Occupation / Precipice	227	3	
2	2821	Exodus, Pt. 1	227	3	
3	2822	Exodus, Pt. 2	227	3	
4	2823	Collaborators	227	3	

	GenreId	Composer	Milliseconds	Bytes	UnitPrice
0	18	None	2622250	490750393	1.99
1	19	None	5286953	1054423946	1.99
2	19	None	2621708	475079441	1.99
3	19	None	2618000	466820021	1.99
4	19	None	2626626	483484911	1.99

```
Ollama parameters:
model=qwen2:7b,
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 * FROM tracks ORDER BY UnitPrice DESC LIMIT 5
\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Tr
ackId           int64\nName             object\nAlbumId           int64\nMediaTypeId       int64\nGenreId
int64\nComposer         object\nMilliseconds      int64\nBytes             int64\nUnitPrice       float64\n
dtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results
of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the d
ataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the
code."}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:55:56.172580967Z', 'message': {'role': 'assistant', 'con
tent': '```python\nimport plotly.express as px\n\nif len(df) == 1:\n    fig = px.indicators.Value(\n
title="Top 5 Most Expensive Tracks",\n        label=df[\'UnitPrice\'].values[0],\n    )\nelse:\n    fig = p
x.bar(df, x=\'UnitPrice\', color=\'Name\', orientation=\'h\',\n                 title=\'Top 5 Most Expensiv
e Tracks\')\n    \nfig.update_layout(xaxis_title=\'Unit Price\', yaxis_title=\'Track Name\')\nfig.show()\n`
``'}, 'done_reason': 'stop', 'done': True, 'total_duration': 26572683831, 'load_duration': 542349, 'prompt_
eval_count': 206, 'prompt_eval_duration': 8531567000, 'eval_count': 107, 'eval_duration': 17947392000}
```


Top 5 Most Expensive Tracks
Track Name
4
3
2
1
0
0
0.5
1
1.5
2
Unit Price
Name
Battlestar Galactica: The Story So Far
Occupation / Precipice
Exodus, Pt. 1
Exodus, Pt. 2
Collaborators

```
Out[28]: ('SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5',
             TrackId                                  Name  AlbumId  MediaTypeId  \
          0     2819  Battlestar Galactica: The Story So Far      226            3
          1     2820                  Occupation / Precipice      227            3
          2     2821                           Exodus, Pt. 1      227            3
          3     2822                           Exodus, Pt. 2      227            3
          4     2823                           Collaborators      227            3

             GenreId Composer  Milliseconds       Bytes  UnitPrice
          0       18     None       2622250   490750393       1.99
          1       19     None       5286953  1054423946       1.99
          2       19     None       2621708   475079441       1.99
          3       19     None       2618000   466820021       1.99
          4       19     None       2626626   483484911       1.99  ,
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': ('Name=Battlestar Galactica: The' ... '}<br>index=%{y}<extra></extra>'),
                        'legendgroup': 'Battlestar Galactica: The Story So Far',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': 'Battlestar Galactica: The Story So Far',
                        'offsetgroup': 'Battlestar Galactica: The Story So Far',
                        'orientation': 'h',
                        'showlegend': True,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array([1.99]),
                        'xaxis': 'x',
                        'y': array([0]),
                        'yaxis': 'y'},
                       {'alignmentgroup': 'True',
                        'hovertemplate': 'Name=Occupation / Precipice<br>UnitPrice=%{x}<br>index=%{y}<extra></extra
          >',
                        'legendgroup': 'Occupation / Precipice',
                        'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
                        'name': 'Occupation / Precipice',
                        'offsetgroup': 'Occupation / Precipice',
                        'orientation': 'h',
                        'showlegend': True,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array([1.99]),
                        'xaxis': 'x',
```

```
 'y': array([1]),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Exodus, Pt. 1<br>UnitPrice=%{x}<br>index=%{y}<extra></extra>',
 'legendgroup': 'Exodus, Pt. 1',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'Exodus, Pt. 1',
 'offsetgroup': 'Exodus, Pt. 1',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([1.99]),
 'xaxis': 'x',
 'y': array([2]),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Exodus, Pt. 2<br>UnitPrice=%{x}<br>index=%{y}<extra></extra>',
 'legendgroup': 'Exodus, Pt. 2',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Exodus, Pt. 2',
 'offsetgroup': 'Exodus, Pt. 2',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([1.99]),
 'xaxis': 'x',
 'y': array([3]),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Name=Collaborators<br>UnitPrice=%{x}<br>index=%{y}<extra></extra>',
 'legendgroup': 'Collaborators',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'Collaborators',
 'offsetgroup': 'Collaborators',
 'orientation': 'h',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array([1.99]),
 'xaxis': 'x',
```

```
                  'y': array([4]),
                  'yaxis': 'y'}],
      'layout': {'barmode': 'relative',
                 'legend': {'title': {'text': 'Name'}, 'tracegroupgap': 0},
                 'template': '...',
                 'title': {'text': 'Top 5 Most Expensive Tracks'},
                 'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Unit Price'}},
                 'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Track Name'}}}
 }))
```

In [29]:
```
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 chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count 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 total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCountry'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceDate, Total

\nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': '
\n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT Cust
omerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId'}, {'role': 'user', 'content': 'How
many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) AS total_customers FROM custo
mers'}, {'role': 'user', 'content': ' \n   List all genres and the number of tracks in each genre:\n'}]
Ollama parameters:
model=qwen2:7b,
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 chinook databas
e 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 suffici
ent 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 e
xactly as it was given before. \n"}, {"role": "user", "content": " \n   Find the top 5 most expensive tra
cks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT * FROM tracks ORDER BY UnitPrice D
ESC LIMIT 5"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names
\n"}, {"role": "assistant", "content": "SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists
ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   Find all tracks with a name containi

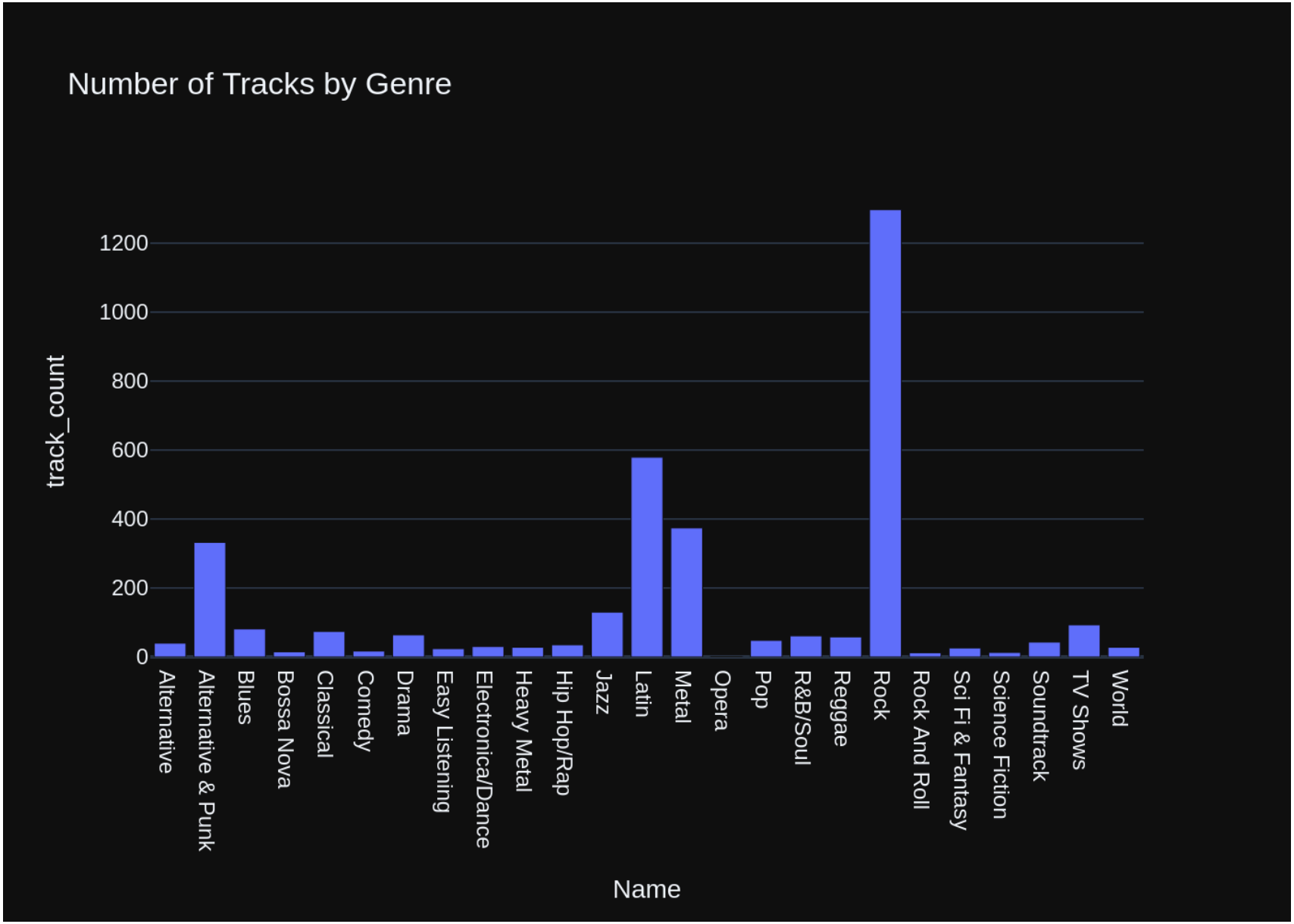
ng \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count 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 total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCountry"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceDate, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "content": " \n   Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) AS total_customers FROM customers"}, {"role": "user", "content": " \n   List all genres and the number of tracks in each genre:\n"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:57:13.297659832Z', 'message': {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.TrackId) as track_count\nFROM genres g \nJOIN tracks t ON g.GenreId = t.GenreId \nGROUP BY g.Name'}, 'done_reason': 'stop', 'done': True, 'total_duration': 77022716758, 'load_duration': 647524, 'prompt_eval_count': 1053, 'prompt_eval_duration': 70245457000, 'eval_count': 36, 'eval_duration': 6134432000}

```
SELECT g.Name, COUNT(t.TrackId) as track_count
FROM genres g
JOIN tracks t ON g.GenreId = t.GenreId
GROUP BY g.Name
```

```
SELECT g.Name, COUNT(t.TrackId) as track_count
FROM genres g
JOIN tracks t ON g.GenreId = t.GenreId
GROUP BY g.Name
```

	Name	track_count
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=qwen2:7b,
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.TrackId) as track_count\nFROM
genres g \nJOIN tracks t ON g.GenreId = t.GenreId \nGROUP BY g.Name\n\nThe following is information about t
he resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name           object\ntrack_count     int
64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the resu
lts of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in t
he dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just
the code."}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:57:38.45183592Z', 'message': {'role': 'assistant', 'cont
ent': "```python\nimport plotly.express as px\n\nif df.shape[0] > 1:\n    fig = px.bar(df, x='Name', y='tra
ck_count', title='Number of Tracks by Genre')\n    fig.show()\nelse:\n    fig = px.scatter(df.index, df['tr
ack_count'], title='Single Genre with Track Count',\n                     labels={'df.index': 'Genre'}, \n
hover_data=['Name'])\n    fig.update_traces(indicator=True)\n    fig.show()\n```"}, 'done_reason': 'stop',
'done': True, 'total_duration': 25128529617, 'load_duration': 572436, 'prompt_eval_count': 183, 'prompt_eva
l_duration': 7760920000, 'eval_count': 103, 'eval_duration': 17276347000}
```


Number of Tracks by Genre
track_count
0
200
400
600
800
1000
1200
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[29]:
```
('SELECT g.Name, COUNT(t.TrackId) as track_count\nFROM genres g \nJOIN tracks t ON g.GenreId = t.GenreId
\nGROUP BY g.Name',
                    Name  track_count
 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>track_count=%{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': 'Number of Tracks by Genre'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'track_count'}}}
 }))
```

In [30]:
```
question = """
    Get all genres that do not have any tracks associated with them:
"""

vn.ask(question=question)
```

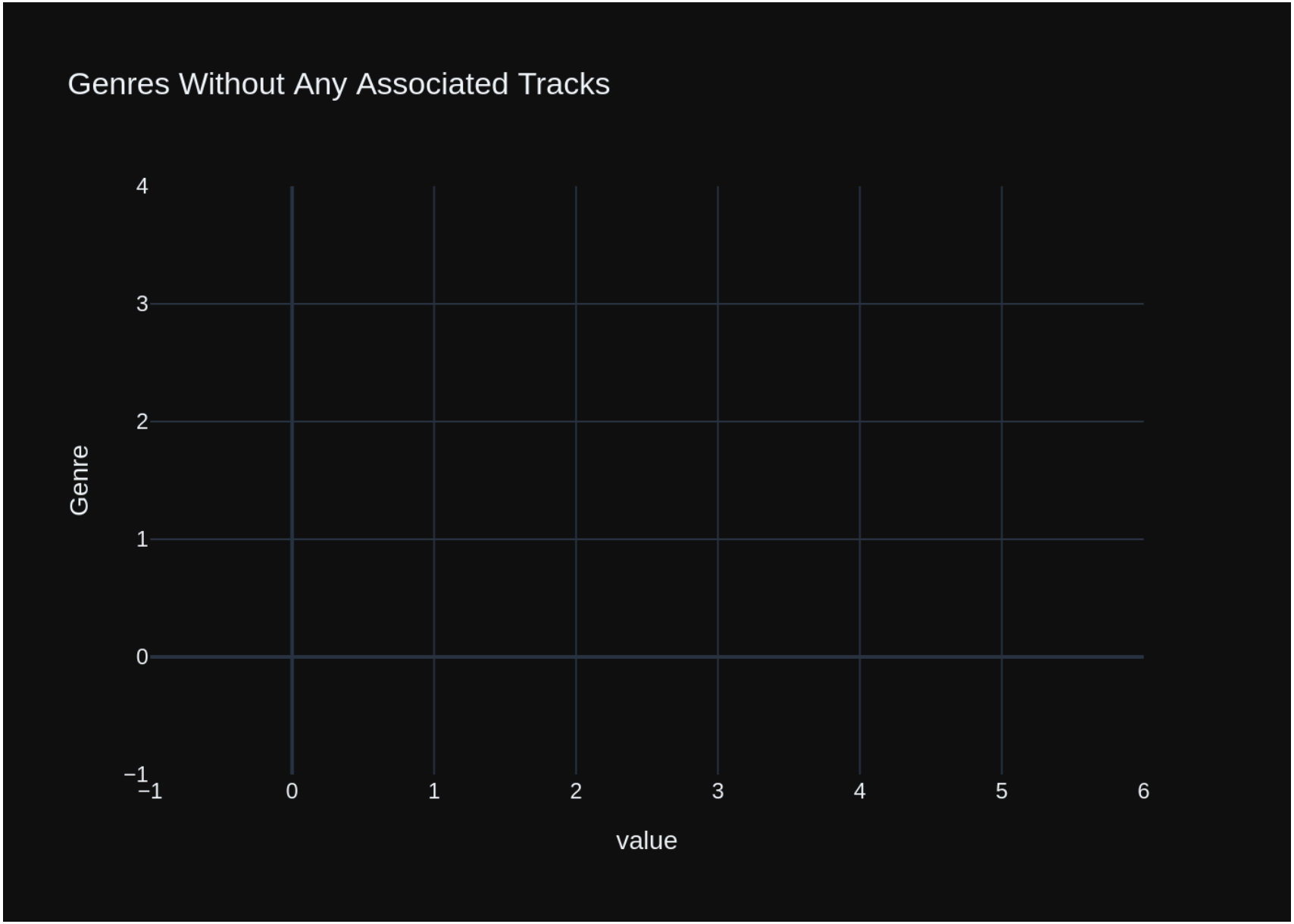
```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

[{'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 chinook 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.TrackId) as track_count\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 * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks ORDER BY UnitPrice DESC 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': "SELECT InvoiceDate, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'conten

t': 'SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_cou
nt DESC \nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'co
ntent': 'SELECT COUNT(*) AS total_customers FROM customers'}, {'role': 'user', 'content': " \n   List all
employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': 'SELECT e1.Fir
stName, e2.FirstName AS ReportingManagerName \nFROM employees e1 \nLEFT JOIN employees e2 ON e1.ReportsTo
= e2.EmployeeId'}, {'role': 'user', 'content': ' \n   Get all genres that do not have any tracks associat
ed with them:\n'}]
Ollama parameters:
model=qwen2:7b,
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 chinook
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   List all genres and the
number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.TrackId) as t
rack_count\nFROM genres g \nJOIN tracks t ON g.GenreId = t.GenreId \nGROUP BY g.Name"}, {"role": "user", "c

ontent": " \n   Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assista
nt", "content": "SELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, {"role": "user", "conten
t": " \n   List all albums and their corresponding artist names \n"}, {"role": "assistant", "content":
"SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId"}, {"rol
e": "user", "content": " \n   Find the top 5 most expensive tracks (based on unit price):\n"}, {"role":
"assistant", "content": "SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5"}, {"role": "user", "conten
t": "Can you list all tables in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT na
me FROM sqlite_master WHERE type='table'"}, {"role": "user", "content": " \n   Find all invoices since 20
10 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceDate, Total \nFROM i
nvoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "content": " \n
List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices
WHERE Total > 10"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"},
{"role": "assistant", "content": "SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Co
untry \nORDER BY customer_count DESC \nLIMIT 5"}, {"role": "user", "content": "How many customers are ther
e"}, {"role": "assistant", "content": "SELECT COUNT(*) AS total_customers FROM customers"}, {"role": "use
r", "content": " \n   List all employees and their reporting manager's name (if any):\n"}, {"role": "assi
stant", "content": "SELECT e1.FirstName, e2.FirstName AS ReportingManagerName \nFROM employees e1 \nLEFT J
OIN employees e2 ON e1.ReportsTo = e2.EmployeeId"}, {"role": "user", "content": " \n   Get all genres tha
t do not have any tracks associated with them:\n"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:58:36.734158696Z', 'message': {'role': 'assistant', 'con
tent': 'SELECT g.Name FROM genres g LEFT JOIN tracks t ON g.GenreId = t.GenreId WHERE t.TrackId IS NULL;'},
'done_reason': 'stop', 'done': True, 'total_duration': 58154477696, 'load_duration': 658554, 'prompt_eval_c
ount': 1095, 'prompt_eval_duration': 52786496000, 'eval_count': 28, 'eval_duration': 4732982000}
SELECT g.Name FROM genres g LEFT JOIN tracks t ON g.GenreId = t.GenreId WHERE t.TrackId IS NULL;
Output from LLM: SELECT g.Name FROM genres g LEFT JOIN tracks t ON g.GenreId = t.GenreId WHERE t.TrackId IS
NULL;
Extracted SQL: SELECT g.Name FROM genres g LEFT JOIN tracks t ON g.GenreId = t.GenreId WHERE t.TrackId IS N
ULL
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=qwen2:7b,
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 FROM genres g LEFT JOIN tracks t
ON g.GenreId = t.GenreId WHERE t.TrackId IS NULL\n\nThe following is information about the resulting pandas

DataFrame 'df': \nRunning df.dtypes gives:\n Name   object\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T22:59:02.478968065Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n   fig = px.indicators(value=df['Name'][0],\n   label='Unique Genre',\n   title='Genres Without Associated Tracks')\nelse:\n   fig = px.bar(df, x=df.index, y='Name', \n   labels={'Name': 'Genre'},\n   title='Genres Without Any Associated Tracks')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 25742679742, 'load_duration': 655187, 'prompt_eval_count': 168, 'prompt_eval_duration': 10895363000, 'eval_count': 88, 'eval_duration': 14753846000}


Genres Without Any Associated Tracks
Genre
4
3
2
1
0
−1
−1
0
1
2
3
4
5
6
value

Out[30]:
```
('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: [],
 Figure({
     'data': [],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Genres Without Any Associated Tracks'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'value'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Genre'}}}
 }))
```

In [31]:
```
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
```

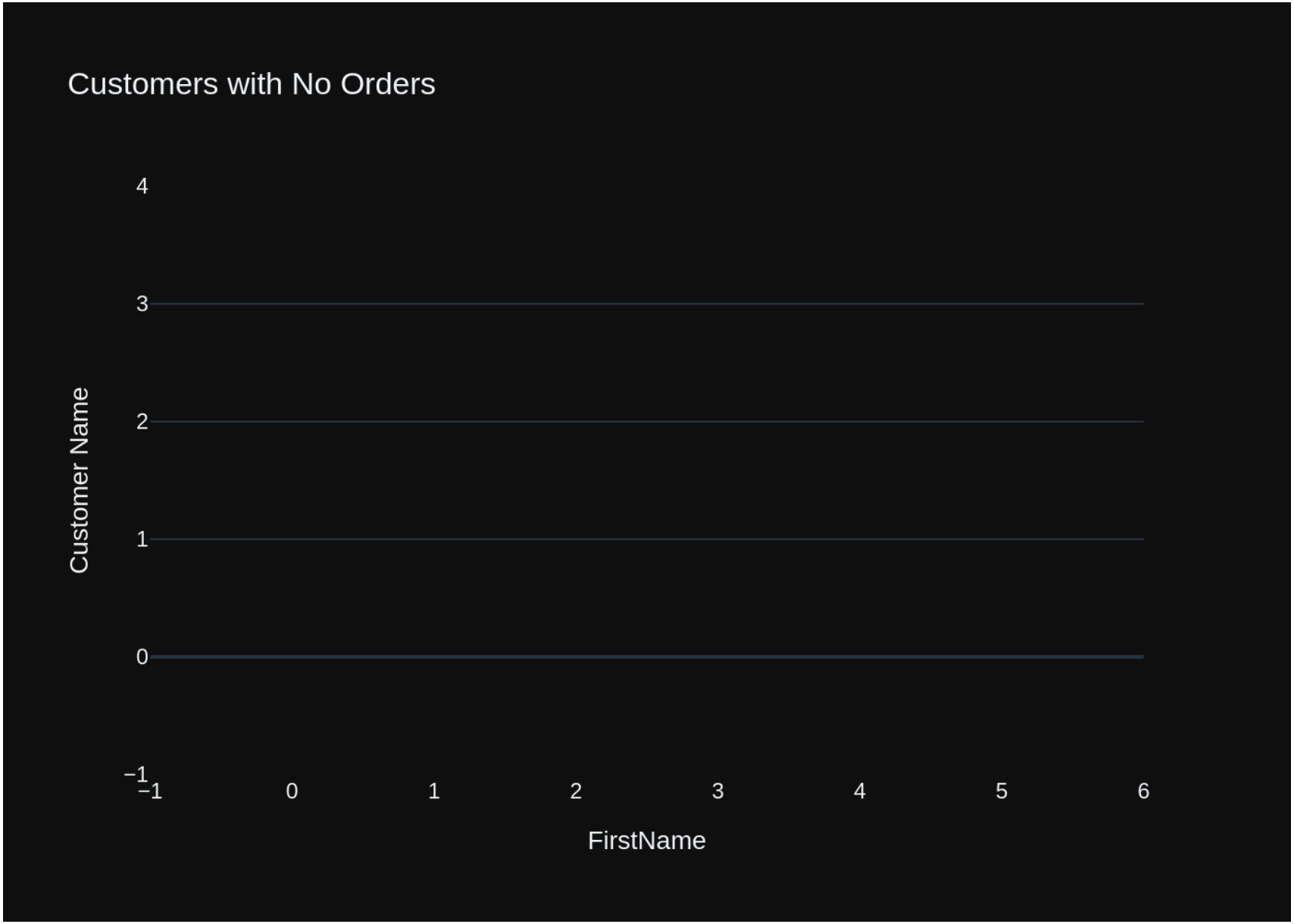
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY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

end the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please e
xplain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has bee
n asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'use
r', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content':
'SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count D
ESC \nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'conten
t': 'SELECT COUNT(*) AS total_customers FROM customers'}, {'role': 'user', 'content': ' \n   Get the tota
l number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) a
s invoice_count \nFROM invoices \nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   List all invo
ices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total
> 10'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amount invoice
d:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceDate, Total \nFROM invoices \nWHERE InvoiceDate >=
'2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': ' \n   Get the average invoice total fo
r each customer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) as avg_total \nFROM i
nvoices \nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per
country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(*) as invoice_count \nFROM inv
oices \nGROUP BY BillingCountry'}, {'role': 'user', 'content': " \n   List all employees and their repor
ting manager's name (if any):\n"}, {'role': 'assistant', 'content': 'SELECT e1.FirstName, e2.FirstName AS R
eportingManagerName \nFROM employees e1 \nLEFT JOIN employees e2 ON e1.ReportsTo = e2.EmployeeId'}, {'rol
e': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assis
tant', 'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = a
r.ArtistId'}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit pric
e):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5'}, {'role':
'user', 'content': ' \n   List all customers who have not placed any orders:\n'}]
Ollama parameters:
model=qwen2:7b,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   Billin
gAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCou
ntry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FORE
IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
FirstName NVARCHAR(40) NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n
Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n
PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT
NULL,\r\n   SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n   InvoiceL

ineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGE
R NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KE
Y (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   F
OREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n
\nCREATE TABLE \"employees\"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Last
Name NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   Rep
ortsTo INTEGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   Ci
ty NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r
\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsT
o) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TAB
LE \"playlist_track\"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CO
NSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES
\"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) RE
FERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums
\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r
\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (Supp
ortRepId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER
NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n
Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES \"albums\"
(AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES \"genres
\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES
\"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_Invo
iceCustomerId ON \"invoices\" (CustomerId)\n\n\n===Additional Context \n\nIn the chinook 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": "what are the top 5 countries that customers come
from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) as customer_count \nFROM customers \nGR
OUP BY Country \nORDER BY customer_count DESC \nLIMIT 5"}, {"role": "user", "content": "How many customers
are there"}, {"role": "assistant", "content": "SELECT COUNT(*) AS total_customers FROM customers"}, {"rol
e": "user", "content": " \n   Get the total number of invoices for each customer\n"}, {"role": "assistan
t", "content": "SELECT CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId"}, {"rol
e": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "c
ontent": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "content": " \n   Find all invoices
since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceDate, Total
\nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "content": "

\n   Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT Custom
erId, AVG(Total) as avg_total \nFROM invoices \nGROUP BY CustomerId"}, {"role": "user", "content": " \n
Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT BillingCountr
y, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCountry"}, {"role": "user", "content": "
\n   List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "content":
"SELECT e1.FirstName, e2.FirstName AS ReportingManagerName \nFROM employees e1 \nLEFT JOIN employees e2 ON
e1.ReportsTo = e2.EmployeeId"}, {"role": "user", "content": " \n   List all albums and their correspondin
g artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, ar.Name AS ArtistName\nFROM albums
a\nJOIN artists ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   Find the top 5 most
expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT * FROM tracks ORDER B
Y UnitPrice DESC LIMIT 5"}, {"role": "user", "content": " \n   List all customers who have not placed any
orders:\n"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T23:00:53.562497436Z', 'message': {'role': 'assistant', 'con
tent': 'SELECT c.CustomerId, c.FirstName, c.LastName \nFROM customers c \nLEFT JOIN invoices i ON c.Custom
erId = i.CustomerId \nWHERE i.CustomerId IS NULL'}, 'done_reason': 'stop', 'done': True, 'total_duration':
110997194152, 'load_duration': 691807, 'prompt_eval_count': 1538, 'prompt_eval_duration': 104319576000, 'ev
al_count': 35, 'eval_duration': 6029269000}
SELECT c.CustomerId, c.FirstName, c.LastName
FROM customers c
LEFT JOIN invoices i ON c.CustomerId = i.CustomerId
WHERE i.CustomerId IS NULL
SELECT c.CustomerId, c.FirstName, c.LastName
FROM customers c
LEFT JOIN invoices i ON c.CustomerId = i.CustomerId
WHERE i.CustomerId IS NULL
Empty DataFrame
Columns: [CustomerId, FirstName, LastName]
Index: []
Ollama parameters:
model=qwen2:7b,
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 customer
s c \nLEFT JOIN invoices i ON c.CustomerId = i.CustomerId \nWHERE i.CustomerId IS NULL\n\nThe following is
information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId   object\nF
irstName   object\nLastName   object\ndtype: object"}, {"role": "user", "content": "Can you generate t
he Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe calle
d 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do no

```
t answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T23:01:20.749126882Z', 'message': {'role': 'assistant', 'con
tent': "```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n    fig = px.indicators.Number(\n
value=df['CustomerId'].values[0],\n        title='Single Customer ID'\n    )\nelse:\n    fig = px.bar(df, x
='FirstName', y='LastName', text='CustomerId',\n                 labels={'LastName': 'Customer Name', 'Cust
omerId': 'Customer ID'},\n                 title='Customers with No Orders')\n    fig.update_traces(textpos
ition='outside')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 27184406173,
'load_duration': 475962, 'prompt_eval_count': 182, 'prompt_eval_duration': 8918462000, 'eval_count': 108,
'eval_duration': 18172045000}
```


Customers with No Orders
Customer Name
4
3
2
1
0
−1
−1
0
1
2
3
4
5
6
FirstName

Out[31]:
```
('SELECT c.CustomerId, c.FirstName, c.LastName \nFROM customers c  \nLEFT JOIN invoices i ON c.CustomerId
= i.CustomerId \nWHERE i.CustomerId IS NULL',
 Empty DataFrame
 Columns: [CustomerId, FirstName, LastName]
 Index: [],
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'FirstName=%{x}<br>Customer Name=%{y}<br>Customer ID=%{text}<extra></extra
>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'text': array([], dtype=object),
               'textposition': 'outside',
               'type': 'bar',
               'x': array([], dtype=object),
               'xaxis': 'x',
               'y': array([], dtype=object),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Customers with No Orders'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'FirstName'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Customer Name'}}}
 }))
```

In [32]:
```
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)
```

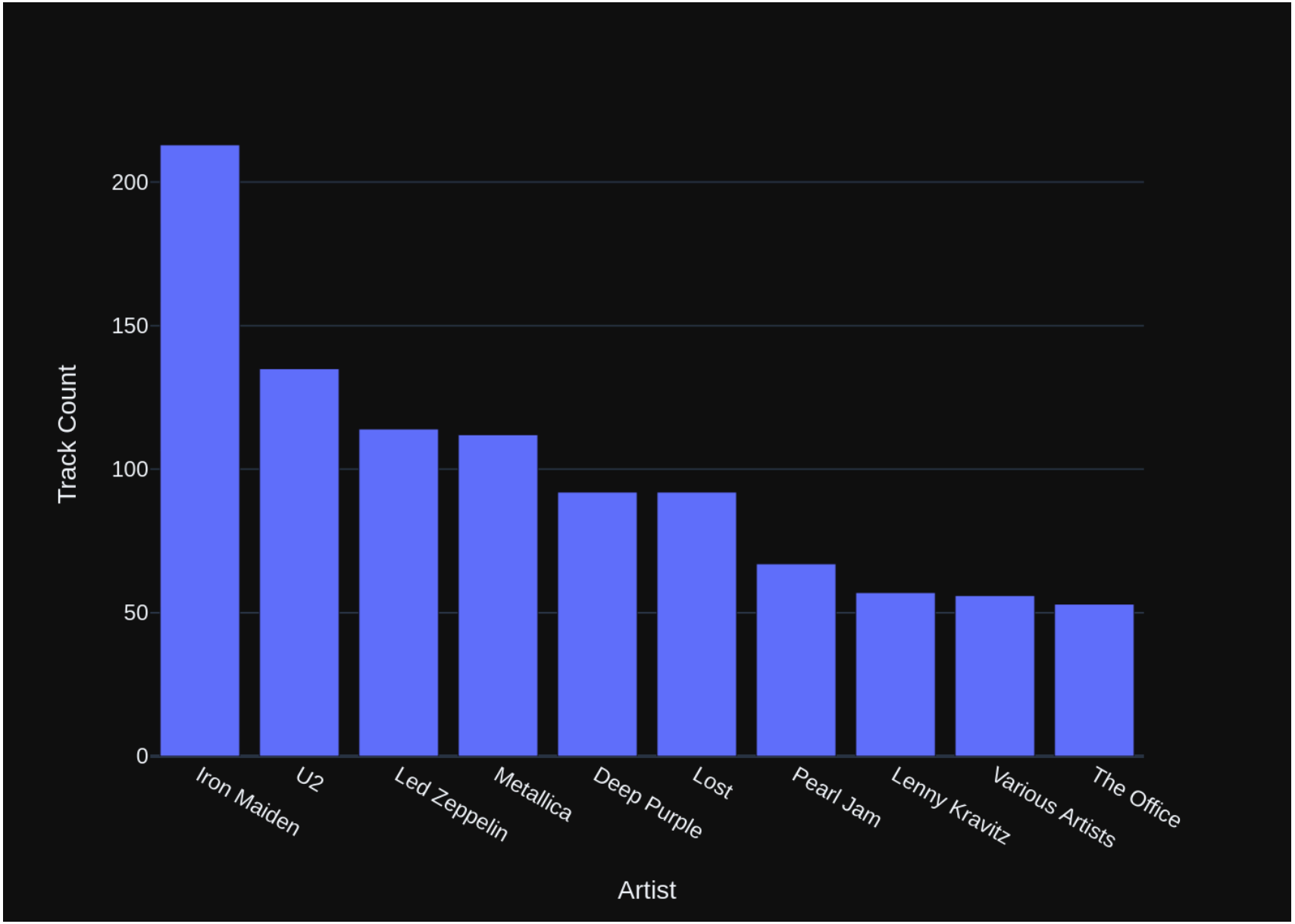
```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

[{'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 chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, 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.TrackId) as track_count\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 * FROM tracks ORDER BY UnitPrice DESC LIMIT 5'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count 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 * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'user', 'content': ' \n   Get the total number of invoi

ces for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) as invoice_count
\nFROM invoices \nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   Find the total number of invo
ices per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(*) as invoice_count
\nFROM invoices \nGROUP BY BillingCountry'}, {'role': 'user', 'content': ' \n   Get the average invoice
total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) as avg_total
\nFROM invoices \nGROUP BY CustomerId'}, {'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'}]
Ollama parameters:
model=qwen2:7b,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREM
ENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT
NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n
Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES \"albums\"
(AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES \"genres
\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES
\"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums
\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r
\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"artists\"\r\n(\r\n   ArtistId INTEGER PRIMAR
Y KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums
\" (ArtistId)\n\nCREATE INDEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"playlists\"\r\n(\r
\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE
\"genres\"\r\n(\r\n   GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n
\nCREATE TABLE \"playlist_track\"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NUL
L,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) R
EFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (T
rackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDE
X IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_track\"
(TrackId)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guideli
nes \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   List all albums and their corresponding artist names \n"}, {"role": "assi

stant", "content": "SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = a
r.ArtistId"}, {"role": "user", "content": " \n   List all genres and the number of tracks in each genr
e:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.TrackId) as track_count\nFROM genres g \nJ
OIN tracks t ON g.GenreId = t.GenreId \nGROUP BY g.Name"}, {"role": "user", "content": " \n   Find the to
p 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT * FROM track
s ORDER BY UnitPrice DESC LIMIT 5"}, {"role": "user", "content": "what are the top 5 countries that custome
rs come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) as customer_count \nFROM custom
ers \nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5"}, {"role": "user", "content": "Can you lis
t all tables in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_m
aster WHERE type='table'"}, {"role": "user", "content": " \n   Find all tracks with a name containing \"W
hat\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks WHERE Name LIKE '%Wha
t%' COLLATE NOCASE"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $1
0:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "cont
ent": " \n   Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SE
LECT CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId"}, {"role": "user", "conte
nt": " \n   Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT
BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCountry"}, {"role": "user", "c
ontent": " \n   Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "S
ELECT CustomerId, AVG(Total) as avg_total \nFROM invoices \nGROUP BY CustomerId"}, {"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"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T23:02:47.904724522Z', 'message': {'role': 'assistant', 'con
tent': 'SELECT a.Name AS artist_name, COUNT(t.TrackId) as track_count\nFROM artists a\nJOIN albums al ON a.
ArtistId = al.ArtistId \nJOIN tracks t ON al.AlbumId = t.AlbumId \nGROUP BY a.ArtistId\nORDER BY track_cou
nt DESC\nLIMIT 10'}, 'done_reason': 'stop', 'done': True, 'total_duration': 87014155669, 'load_duration': 8
00137, 'prompt_eval_count': 1119, 'prompt_eval_duration': 74976140000, 'eval_count': 66, 'eval_duration': 1
1389873000}
SELECT a.Name AS artist_name, COUNT(t.TrackId) as track_count
FROM artists a
JOIN albums al ON a.ArtistId = al.ArtistId
JOIN tracks t ON al.AlbumId = t.AlbumId
GROUP BY a.ArtistId
ORDER BY track_count DESC
LIMIT 10
SELECT a.Name AS artist_name, COUNT(t.TrackId) as track_count
FROM artists a
JOIN albums al ON a.ArtistId = al.ArtistId
JOIN tracks t ON al.AlbumId = t.AlbumId
GROUP BY a.ArtistId
ORDER BY track_count DESC

```
LIMIT 10
       artist_name  track_count
0      Iron Maiden          213
1               U2          135
2     Led Zeppelin          114
3        Metallica          112
4      Deep Purple           92
5             Lost           92
6        Pearl Jam           67
7    Lenny Kravitz           57
8  Various Artists           56
9       The Office           53
Ollama parameters:
model=qwen2:7b,
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 AS artist_name, COUNT(t.TrackId) as track_count\nFROM artists a\nJOIN albums al ON a.ArtistId =
al.ArtistId \nJOIN tracks t ON al.AlbumId = t.AlbumId  \nGROUP BY a.ArtistId\nORDER BY track_count DESC\nLI
MIT 10\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes give
s:\n artist_name    object\ntrack_count     int64\ndtype: object"}, {"role": "user", "content": "Can you ge
nerate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas datafra
me called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python cod
e. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T23:03:17.676933249Z', 'message': {'role': 'assistant', 'con
tent': "```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n    fig = px.indicators.Ifs(df['art
ist_name'], max_value=df['track_count'])\nelse:\n    fig = px.bar(df, x='artist_name', y='track_count')\n
fig.update_layout(xaxis_title='Artist', yaxis_title='Track Count')\n```"}, 'done_reason': 'stop', 'done': T
rue, 'total_duration': 29746492034, 'load_duration': 566187, 'prompt_eval_count': 252, 'prompt_eval_duratio
n': 16344376000, 'eval_count': 79, 'eval_duration': 13310001000}
```


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

Out[32]:
```
('SELECT a.Name AS artist_name, COUNT(t.TrackId) as track_count\nFROM artists a\nJOIN albums al ON a.Artis
tId = al.ArtistId \nJOIN tracks t ON al.AlbumId = t.AlbumId  \nGROUP BY a.ArtistId\nORDER BY track_count D
ESC\nLIMIT 10',
       artist_name  track_count
 0      Iron Maiden          213
 1               U2          135
 2     Led Zeppelin          114
 3        Metallica          112
 4      Deep Purple           92
 5             Lost           92
 6        Pearl Jam           67
 7    Lenny Kravitz           57
 8  Various Artists           56
 9       The Office           53,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'artist_name=%{x}<br>track_count=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Iron Maiden', 'U2', 'Led Zeppelin', 'Metallica', 'Deep Purple', 'Lost',
                           'Pearl Jam', 'Lenny Kravitz', 'Various Artists', 'The Office'],
                          dtype=object),
               'xaxis': 'x',
               'y': array([213, 135, 114, 112,  92,  92,  67,  57,  56,  53]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Artist'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Track Count'}}}
 }))
```

In [33]:
```
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 chinook 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 Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) AS total_customers FROM customer

s'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCountry'}, {'role': 'user', 'content': " \n   List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': 'SELECT e1.FirstName, e2.FirstName AS ReportingManagerName \nFROM employees e1 \nLEFT JOIN employees e2 ON e1.ReportsTo = e2.EmployeeId'}, {'role': 'user', 'content': ' \n   Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) as avg_total \nFROM invoices \nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices \nWHERE Total > 10'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceDate, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY 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   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5'}, {'role': 'user', 'content': ' \n   List all customers from Canada and their email addresses:\n'}]
Ollama parameters:
model=qwen2:7b,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_CustomerSupportRepId ON \"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_i

tems\"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tracks\" (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 chinook 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 Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) AS total_customers FROM customers"}, {"role": "user", "content": " \n   Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCountry"}, {"role": "user", "content": " \n   List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT e1.FirstName, e2.FirstName AS ReportingManagerName \nFROM employees e1 \nLEFT JOIN employees e2 ON e1.ReportsTo = e2.EmployeeId"}, {"role": "user", "content": " \n   Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT CustomerId, AVG(Total) as avg_total \nFROM invoices \nGROUP BY CustomerId"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceDate, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY 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   Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5"}, {"role": "user", "content": " \n   List all customers from Canada and their email addresses:\n"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T23:04:51.548589715Z', 'message': {'role': 'assistant', 'con

tent': "SELECT FirstName, LastName, Email \nFROM customers \nWHERE Country = 'Canada'"}, 'done_reason':
'stop', 'done': True, 'total_duration': 93753618414, 'load_duration': 700048, 'prompt_eval_count': 1354, 'p
rompt_eval_duration': 90269668000, 'eval_count': 17, 'eval_duration': 2845137000}

```
SELECT FirstName, LastName, Email
FROM customers
WHERE Country = 'Canada'
```

```
SELECT FirstName, LastName, Email
FROM customers
WHERE Country = 'Canada'
```

	FirstName	LastName	Email
0	François	Tremblay	ftremblay@gmail.com
1	Mark	Philips	mphilips12@shaw.ca
2	Jennifer	Peterson	jenniferp@rogers.ca
3	Robert	Brown	robbrown@shaw.ca
4	Edward	Francis	edfrancis@yachoo.ca
5	Martha	Silk	marthasilk@gmail.com
6	Aaron	Mitchell	aaronmitchell@yahoo.ca
7	Ellie	Sullivan	ellie.sullivan@shaw.ca

Ollama parameters:

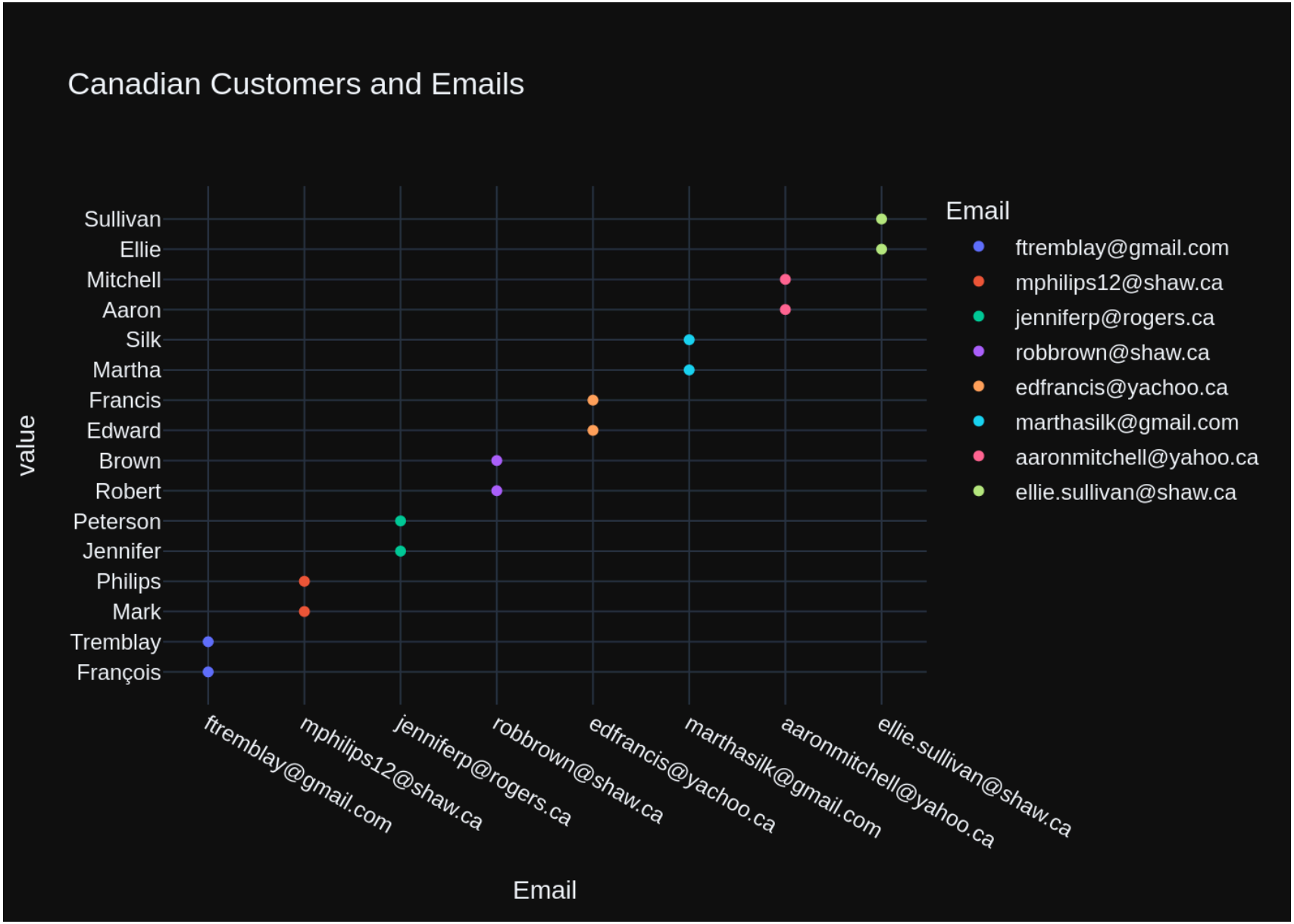
```
model=qwen2:7b,
options={},
keep_alive=None
```

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 FirstName, LastName, Email \nFROM customers
\nWHERE Country = 'Canada'\n\nThe following is information about the resulting pandas DataFrame 'df': \nRun
ning df.dtypes gives:\n FirstName   object\nLastName   object\nEmail   object\ndtype: object"}, {"r
ole": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? As
sume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Ind
icator. Respond with only Python code. Do not answer with any explanations -- just the code."}]

Ollama Response:

{'model': 'qwen2:7b', 'created_at': '2024-06-15T23:05:19.189645024Z', 'message': {'role': 'assistant', 'con
tent': "```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n   fig = px.scatter(x=df['Email'].
iloc[0], y=[len(df.index)], \n   title='Single Canadian Customer and Email', \n
labels={'x': 'Customer Email', 'y': 'Count'})\nelse:\n   fig = px.scatter(df, x='Email', y=['FirstName'] +
['LastName'] * len(df), \n   color='Email', hover_name='Email', \n   ti
tle='Canadian Customers and Emails', \n   labels={'x': 'Customer Email', 'hover_name': 'F
ull Name'})\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 27615348279, 'load
_duration': 557198, 'prompt_eval_count': 162, 'prompt_eval_duration': 4744190000, 'eval_count': 135, 'eval_
duration': 22775861000}


Canadian Customers and Emails
value
Sullivan
Ellie
Mitchell
Aaron
Silk
Martha
Francis
Edward
Brown
Robert
Peterson
Jennifer
Philips
Mark
Tremblay
François
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
Email
Email
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

```
Out[33]: ("SELECT FirstName, LastName, Email  \nFROM customers  \nWHERE Country = 'Canada'",
```

	FirstName	LastName	Email
0	François	Tremblay	ftremblay@gmail.com
1	Mark	Philips	mphilips12@shaw.ca
2	Jennifer	Peterson	jenniferp@rogers.ca
3	Robert	Brown	robbrown@shaw.ca
4	Edward	Francis	edfrancis@yachoo.ca
5	Martha	Silk	marthasilk@gmail.com
6	Aaron	Mitchell	aaronmitchell@yahoo.ca
7	Ellie	Sullivan	ellie.sullivan@shaw.ca,

```
Figure({
    'data': [{'hovertemplate': '<b>%{hovertext}</b><br><br>Email=%{x}<br>value=%{y}<extra></extra>',
              'hovertext': array(['ftremblay@gmail.com', 'ftremblay@gmail.com'], dtype=object),
              'legendgroup': 'ftremblay@gmail.com',
              'marker': {'color': '#636efa', 'symbol': 'circle'},
              'mode': 'markers',
              'name': 'ftremblay@gmail.com',
              'orientation': 'v',
              'showlegend': True,
              'type': 'scatter',
              'x': array(['ftremblay@gmail.com', 'ftremblay@gmail.com'], dtype=object),
              'xaxis': 'x',
              'y': array(['François', 'Tremblay'], dtype=object),
              'yaxis': 'y'},
             {'hovertemplate': '<b>%{hovertext}</b><br><br>Email=%{x}<br>value=%{y}<extra></extra>',
              'hovertext': array(['mphilips12@shaw.ca', 'mphilips12@shaw.ca'], dtype=object),
              'legendgroup': 'mphilips12@shaw.ca',
              'marker': {'color': '#EF553B', 'symbol': 'circle'},
              'mode': 'markers',
              'name': 'mphilips12@shaw.ca',
              'orientation': 'v',
              'showlegend': True,
              'type': 'scatter',
              'x': array(['mphilips12@shaw.ca', 'mphilips12@shaw.ca'], dtype=object),
              'xaxis': 'x',
              'y': array(['Mark', 'Philips'], dtype=object),
              'yaxis': 'y'},
             {'hovertemplate': '<b>%{hovertext}</b><br><br>Email=%{x}<br>value=%{y}<extra></extra>',
              'hovertext': array(['jenniferp@rogers.ca', 'jenniferp@rogers.ca'], dtype=object),
              'legendgroup': 'jenniferp@rogers.ca',
              'marker': {'color': '#00cc96', 'symbol': 'circle'},
              'mode': 'markers',
```

```
 'name': 'jenniferp@rogers.ca',
 'orientation': 'v',
 'showlegend': True,
 'type': 'scatter',
 'x': array(['jenniferp@rogers.ca', 'jenniferp@rogers.ca'], dtype=object),
 'xaxis': 'x',
 'y': array(['Jennifer', 'Peterson'], dtype=object),
 'yaxis': 'y'},
{'hovertemplate': '<b>%{hovertext}</b><br><br>Email=%{x}<br>value=%{y}<extra></extra>',
 'hovertext': array(['robbrown@shaw.ca', 'robbrown@shaw.ca'], dtype=object),
 'legendgroup': 'robbrown@shaw.ca',
 'marker': {'color': '#ab63fa', 'symbol': 'circle'},
 'mode': 'markers',
 'name': 'robbrown@shaw.ca',
 'orientation': 'v',
 'showlegend': True,
 'type': 'scatter',
 'x': array(['robbrown@shaw.ca', 'robbrown@shaw.ca'], dtype=object),
 'xaxis': 'x',
 'y': array(['Robert', 'Brown'], dtype=object),
 'yaxis': 'y'},
{'hovertemplate': '<b>%{hovertext}</b><br><br>Email=%{x}<br>value=%{y}<extra></extra>',
 'hovertext': array(['edfrancis@yachoo.ca', 'edfrancis@yachoo.ca'], dtype=object),
 'legendgroup': 'edfrancis@yachoo.ca',
 'marker': {'color': '#FFA15A', 'symbol': 'circle'},
 'mode': 'markers',
 'name': 'edfrancis@yachoo.ca',
 'orientation': 'v',
 'showlegend': True,
 'type': 'scatter',
 'x': array(['edfrancis@yachoo.ca', 'edfrancis@yachoo.ca'], dtype=object),
 'xaxis': 'x',
 'y': array(['Edward', 'Francis'], dtype=object),
 'yaxis': 'y'},
{'hovertemplate': '<b>%{hovertext}</b><br><br>Email=%{x}<br>value=%{y}<extra></extra>',
 'hovertext': array(['marthasilk@gmail.com', 'marthasilk@gmail.com'], dtype=object),
 'legendgroup': 'marthasilk@gmail.com',
 'marker': {'color': '#19d3f3', 'symbol': 'circle'},
 'mode': 'markers',
 'name': 'marthasilk@gmail.com',
 'orientation': 'v',
 'showlegend': True,
```

```
                  'type': 'scatter',
                  'x': array(['marthasilk@gmail.com', 'marthasilk@gmail.com'], dtype=object),
                  'xaxis': 'x',
                  'y': array(['Martha', 'Silk'], dtype=object),
                  'yaxis': 'y'},
                 {'hovertemplate': '<b>%{hovertext}</b><br><br>Email=%{x}<br>value=%{y}<extra></extra>',
                  'hovertext': array(['aaronmitchell@yahoo.ca', 'aaronmitchell@yahoo.ca'], dtype=object),
                  'legendgroup': 'aaronmitchell@yahoo.ca',
                  'marker': {'color': '#FF6692', 'symbol': 'circle'},
                  'mode': 'markers',
                  'name': 'aaronmitchell@yahoo.ca',
                  'orientation': 'v',
                  'showlegend': True,
                  'type': 'scatter',
                  'x': array(['aaronmitchell@yahoo.ca', 'aaronmitchell@yahoo.ca'], dtype=object),
                  'xaxis': 'x',
                  'y': array(['Aaron', 'Mitchell'], dtype=object),
                  'yaxis': 'y'},
                 {'hovertemplate': '<b>%{hovertext}</b><br><br>Email=%{x}<br>value=%{y}<extra></extra>',
                  'hovertext': array(['ellie.sullivan@shaw.ca', 'ellie.sullivan@shaw.ca'], dtype=object),
                  'legendgroup': 'ellie.sullivan@shaw.ca',
                  'marker': {'color': '#B6E880', 'symbol': 'circle'},
                  'mode': 'markers',
                  'name': 'ellie.sullivan@shaw.ca',
                  'orientation': 'v',
                  'showlegend': True,
                  'type': 'scatter',
                  'x': array(['ellie.sullivan@shaw.ca', 'ellie.sullivan@shaw.ca'], dtype=object),
                  'xaxis': 'x',
                  'y': array(['Ellie', 'Sullivan'], dtype=object),
                  'yaxis': 'y'}],
 'layout': {'legend': {'title': {'text': 'Email'}, 'tracegroupgap': 0},
            'template': '...',
            'title': {'text': 'Canadian Customers and Emails'},
            'xaxis': {'anchor': 'y',
                      'categoryarray': [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],
                      'categoryorder': 'array',
```

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

In [34]:
```
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
```

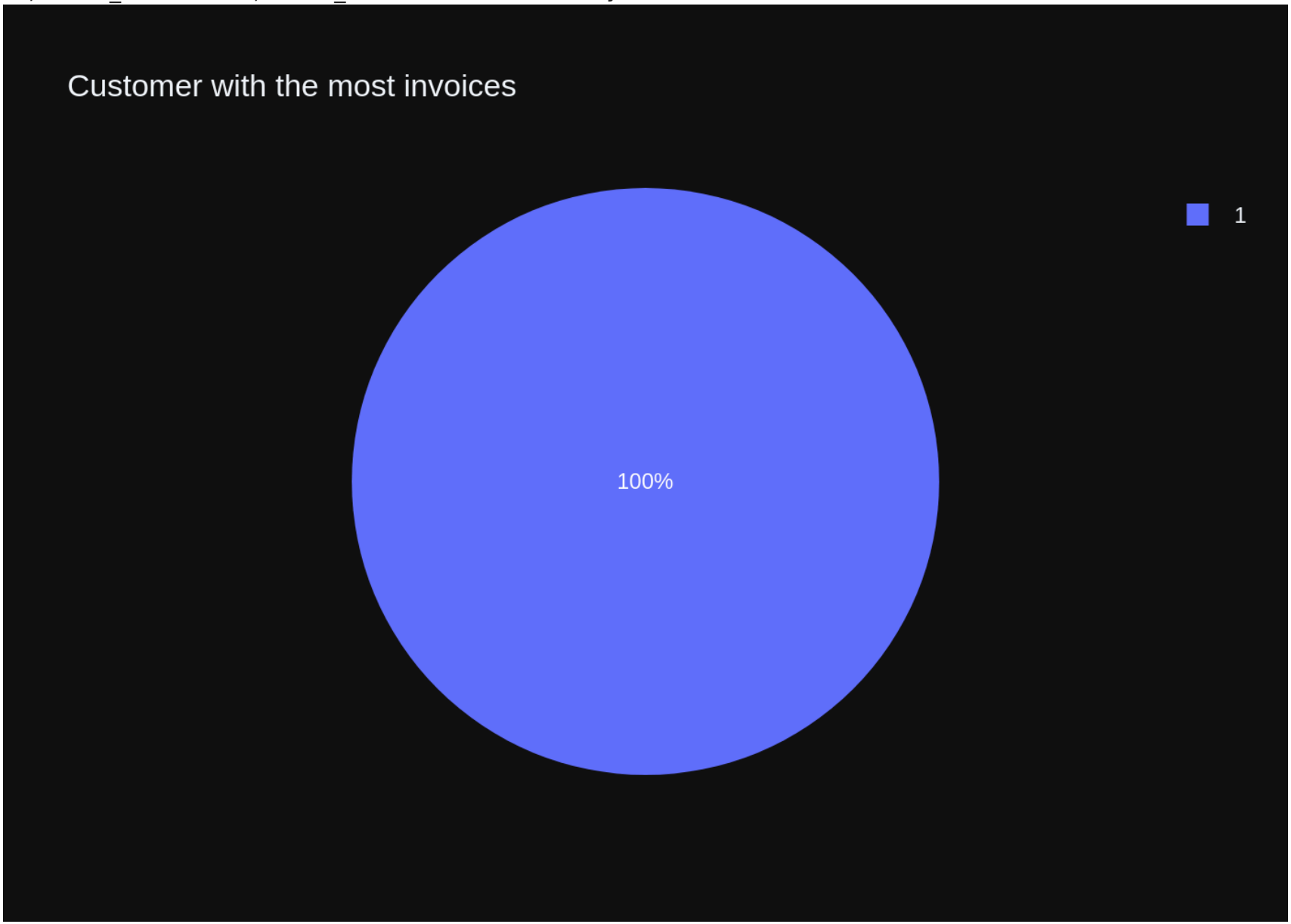
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY 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 chinook 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 CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $1

0:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'user', 'cont
ent': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'cont
ent': "SELECT InvoiceDate, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDat
e"}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'ass
istant', 'content': 'SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCo
untry'}, {'role': 'user', 'content': ' \n   Get the average invoice total for each customer:\n'}, {'rol
e': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) as avg_total \nFROM invoices \nGROUP BY Customer
Id'}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'},
{'role': 'assistant', 'content': 'SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5'}, {'role': 'user',
'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SEL
ECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC
\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content':
'SELECT COUNT(*) AS total_customers FROM customers'}, {'role': 'user', 'content': ' \n   List all custom
ers from Canada and their email addresses:\n'}, {'role': 'assistant', 'content': "SELECT FirstName, LastNam
e, Email \nFROM customers \nWHERE Country = 'Canada'"}, {'role': 'user', 'content': " \n   List all emp
loyees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': 'SELECT e1.FirstN
ame, e2.FirstName AS ReportingManagerName \nFROM employees e1 \nLEFT JOIN employees e2 ON e1.ReportsTo = e
2.EmployeeId'}, {'role': 'user', 'content': ' \n   Find the customer with the most invoices \n'}]
Ollama parameters:
model=qwen2:7b,
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 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 chinook 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 CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceDate, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCountry"}, {"role": "user", "content": " \n   Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT CustomerId, AVG(Total) as avg_total \nFROM invoices \nGROUP BY CustomerId"}, {"role": "user", "content": " \n   Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC \nLIMIT 5"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) AS total_customers FROM customers"}, {"role": "user", "content": " \n   List all customers from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SELECT FirstName, LastName, Email \nFROM customers \nWHERE Country = 'Canada'"}, {"role": "user", "content": " \n   List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT e1.FirstName, e2.FirstName AS ReportingManagerName \nFROM employees e1 \nLEFT JOIN employees e2 ON e1.ReportsTo = e2.EmployeeId"}, {"role": "user", "content": " \n   Find the customer with the most invoices \n"}]

```
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T23:06:55.365156614Z', 'message': {'role': 'assistant', 'con
tent': 'SELECT c.CustomerId, COUNT(i.InvoiceId) as invoice_count  \nFROM customers c \nJOIN invoices i ON
c.CustomerId = i.CustomerId \nGROUP BY c.CustomerId  \nORDER BY invoice_count DESC  \nLIMIT 1'}, 'done_reas
on': 'stop', 'done': True, 'total_duration': 96031630344, 'load_duration': 768731, 'prompt_eval_count': 139
2, 'prompt_eval_duration': 87203150000, 'eval_count': 47, 'eval_duration': 8187396000}
SELECT c.CustomerId, COUNT(i.InvoiceId) as invoice_count
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY invoice_count DESC
LIMIT 1
SELECT c.CustomerId, COUNT(i.InvoiceId) as invoice_count
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY invoice_count DESC
LIMIT 1
   CustomerId  invoice_count
0           1              7
Ollama parameters:
model=qwen2:7b,
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 invoice_count  \nFROM cust
omers c \nJOIN invoices i ON c.CustomerId = i.CustomerId \nGROUP BY c.CustomerId  \nORDER BY invoice_count
DESC  \nLIMIT 1\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dty
pes gives:\n CustomerId       int64\ninvoice_count    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 pan
das dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only
Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T23:07:27.594104209Z', 'message': {'role': 'assistant', 'con
tent': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n    fig = px.pie(df, values=\'invoi
ce_count\', names=\'CustomerId\', title="Customer with the most invoices")\nelse:\n    fig = px.bar(df, x=
\'CustomerId\', y=\'invoice_count\', title="Customers by number of Invoices", labels={\'CustomerId\': \'Cus
tomer ID\', \'invoice_count\': \'Number of Invoices\'})\n    \nfig.update_layout(xaxis_title=\'Customer ID
\', yaxis_title=\'Number of Invoices\')\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_dur
```

ation': 32203896770, 'load_duration': 605637, 'prompt_eval_count': 193, 'prompt_eval_duration': 1241548000
0, 'eval_count': 116, 'eval_duration': 19695833000}



Out[34]:
```
('SELECT c.CustomerId, COUNT(i.InvoiceId) as invoice_count  \nFROM customers c \nJOIN invoices i ON c.Cust
omerId = i.CustomerId \nGROUP BY c.CustomerId  \nORDER BY invoice_count DESC  \nLIMIT 1',
    CustomerId  invoice_count
 0           1              7,
 Figure({
     'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
               'hovertemplate': 'CustomerId=%{label}<br>invoice_count=%{value}<extra></extra>',
               'labels': array([1]),
               'legendgroup': '',
               'name': '',
               'showlegend': True,
               'type': 'pie',
               'values': array([7])}],
     'layout': {'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Customer with the most invoices'},
                'xaxis': {'title': {'text': 'Customer ID'}},
                'yaxis': {'title': {'text': 'Number of Invoices'}}}
 }))
```

In [ ]:

Advanced SQL questions

In [35]:
```
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 chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) as invoice_count \nFROM customers c \nJOIN invoices i ON c.CustomerId = i.CustomerId \nGROUP BY c.CustomerId \nORDER BY invoice_count DESC \nLIMIT 1'}, {'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 AS artist_name, COUNT(t.TrackId) as track_count\nFROM artists a\nJOIN albums al ON a.ArtistId = al.ArtistId \nJOIN tracks t ON al.AlbumId = t.AlbumId \nGROUP BY a.ArtistId\nORDER BY track_count DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the t

otal amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceDate, Total \nFROM invoices \nW
HERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': ' \n   Find the top
5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks
ORDER BY UnitPrice DESC LIMIT 5'}, {'role': 'user', 'content': ' \n   List all invoices with a total exce
eding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'use
r', 'content': ' \n   Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'conten
t': 'SELECT CustomerId, AVG(Total) as avg_total \nFROM invoices \nGROUP BY CustomerId'}, {'role': 'user',
'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'S
ELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCountry'}, {'role': 'use
r', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant',
'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.Artist
Id'}, {'role': 'user', 'content': ' \n   List all genres and the number of tracks in each genre:\n'}, {'r
ole': 'assistant', 'content': 'SELECT g.Name, COUNT(t.TrackId) as track_count\nFROM genres g \nJOIN tracks
t ON g.GenreId = t.GenreId \nGROUP BY g.Name'}, {'role': 'user', 'content': ' \n   Find the customer who
bought the most albums in total quantity (across all invoices): \n'}]
Ollama parameters:
model=qwen2:7b,
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 chinook database invoice means order\n\n===Response Guidelines \n1. If the pr
ovided context is sufficient, please generate a valid SQL query without any explanations for the question.
\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particul
ar column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend t
he query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explai
n why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been aske
d and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "con
tent": " \n   Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.
CustomerId, COUNT(i.InvoiceId) as invoice_count \nFROM customers c \nJOIN invoices i ON c.CustomerId = i.C
ustomerId \nGROUP BY c.CustomerId \nORDER BY invoice_count DESC \nLIMIT 1"}, {"role": "user", "content":
" \n   There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, al
bums and tracks are linked by AlbumId,\n   Can you find the top 10 most popular artists based on the numbe
r of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name AS artist_name, COUNT(t.TrackId) as track_
count\nFROM artists a\nJOIN albums al ON a.ArtistId = al.ArtistId \nJOIN tracks t ON al.AlbumId = t.AlbumId
\nGROUP BY a.ArtistId\nORDER BY track_count DESC\nLIMIT 10"}, {"role": "user", "content": " \n   Get the
total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT
(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId"}, {"role": "user", "content": " \n   Find all
invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceDat
e, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "co
ntent": " \n   Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "co
ntent": "SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5"}, {"role": "user", "content": " \n   List
all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHER
E Total > 10"}, {"role": "user", "content": " \n   Get the average invoice total for each customer:\n"},
{"role": "assistant", "content": "SELECT CustomerId, AVG(Total) as avg_total \nFROM invoices \nGROUP BY Cus
tomerId"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"rol
e": "assistant", "content": "SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY B
illingCountry"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names
\n"}, {"role": "assistant", "content": "SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists
ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   List all genres and the number of tr
acks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.TrackId) as track_count\n
FROM genres g \nJOIN tracks t ON g.GenreId = t.GenreId \nGROUP BY g.Name"}, {"role": "user", "content": "
\n   Find the customer who bought the most albums in total quantity (across all invoices): \n"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T23:09:03.9674524Z', 'message': {'role': 'assistant', 'conte
nt': 'SELECT c.CustomerId, SUM(i.Quantity * ai.TrackId) as total_albums_bought\nFROM customers c \nJOIN inv
oices i ON c.CustomerId = i.CustomerId \nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId \nJOIN IFK_I
nvoiceLineTrackId (ii) ON ii.TrackId = t.TrackId \nGROUP BY c.CustomerId \nORDER BY total_albums_bought D
ESC\nLIMIT 1'}, 'done_reason': 'stop', 'done': True, 'total_duration': 96228837715, 'load_duration': 94873
1, 'prompt_eval_count': 1313, 'prompt_eval_duration': 79894275000, 'eval_count': 90, 'eval_duration': 15655

```
514000}
SELECT c.CustomerId, SUM(i.Quantity * ai.TrackId) as total_albums_bought
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId
JOIN IFK_InvoiceLineTrackId (ii) ON ii.TrackId = t.TrackId
GROUP BY c.CustomerId
ORDER BY total_albums_bought DESC
LIMIT 1
SELECT c.CustomerId, SUM(i.Quantity * ai.TrackId) as total_albums_bought
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId
JOIN IFK_InvoiceLineTrackId (ii) ON ii.TrackId = t.TrackId
GROUP BY c.CustomerId
ORDER BY total_albums_bought DESC
LIMIT 1
Couldn't run sql:  Execution failed on sql 'SELECT c.CustomerId, SUM(i.Quantity * ai.TrackId) as total_albu
ms_bought
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId
JOIN IFK_InvoiceLineTrackId (ii) ON ii.TrackId = t.TrackId
GROUP BY c.CustomerId
ORDER BY total_albums_bought DESC
LIMIT 1': no such table: IFK_InvoiceLineTrackId
```

In [36]:
```
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
```

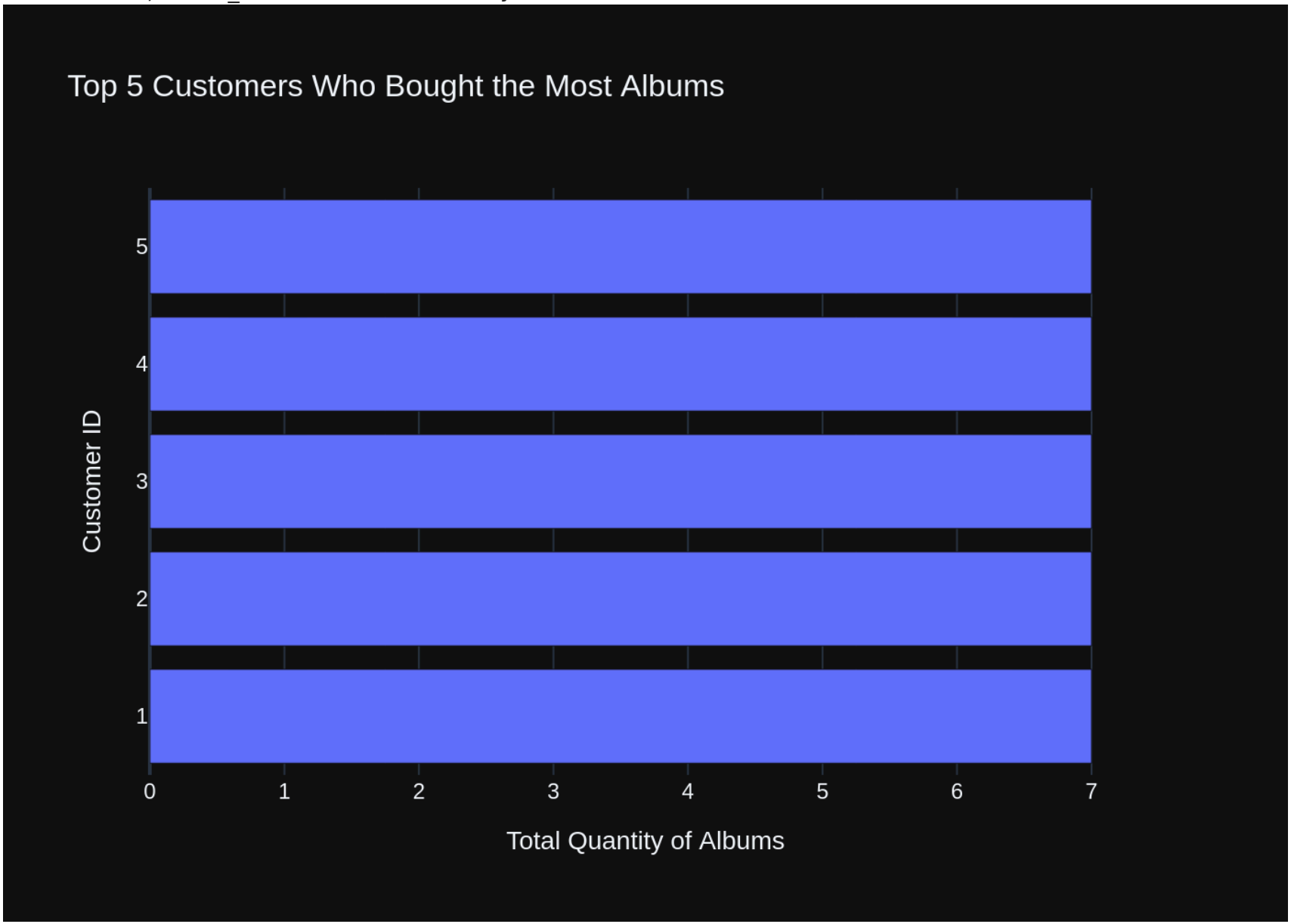
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId 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 chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) as invoice_count \nFROM customers c \nJOIN invoices i ON c.CustomerId = i.CustomerId \nGROUP BY c.CustomerId \nORDER BY invoice_count DESC \nLIMIT 1'}, {'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 AS artist_name, COUNT(t.TrackId) as track_count\nFROM artists a\nJOIN albums al ON a.ArtistId = al.ArtistId \nJOIN tracks t ON al.AlbumId = t.AlbumId \nGROUP BY a.ArtistId\nORDER BY track_count DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role':

'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'user', 'content': ' \n   Ge
t the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId,
COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   Fin
d the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, C
OUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCountry'}, {'role': 'user', 'content': ' \n
Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT I
nvoiceDate, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'u
ser', 'content': ' \n   Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'cont
ent': 'SELECT CustomerId, AVG(Total) as avg_total \nFROM invoices \nGROUP BY CustomerId'}, {'role': 'user',
'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'cont
ent': 'SELECT a.Title, 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?'}, {'role': 'assistan
t', 'content': 'SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY c
ustomer_count DESC \nLIMIT 5'}, {'role': 'user', 'content': ' \n   Hint: album quantity is found in invoi
ce_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all inv
oices):\n'}]
Ollama parameters:
model=qwen2:7b,
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 chinook 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 customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.Custome
rId, COUNT(i.InvoiceId) as invoice_count \nFROM customers c \nJOIN invoices i ON c.CustomerId = i.Customer
Id \nGROUP BY c.CustomerId \nORDER BY invoice_count DESC \nLIMIT 1"}, {"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 AS artist_name, COUNT(t.TrackId) as track_count\nFR
OM artists a\nJOIN albums al ON a.ArtistId = al.ArtistId \nJOIN tracks t ON al.AlbumId = t.AlbumId \nGROUP
BY a.ArtistId\nORDER BY track_count DESC\nLIMIT 10"}, {"role": "user", "content": " \n   Find the top 5 m
ost expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT * FROM tracks ORD
ER BY UnitPrice DESC LIMIT 5"}, {"role": "user", "content": " \n   List all invoices with a total exceedi
ng $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user",
"content": " \n   Get the total number of invoices for each customer\n"}, {"role": "assistant", "conten
t": "SELECT CustomerId, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId"}, {"role": "user",
"content": " \n   Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "S
ELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY BillingCountry"}, {"role": "use
r", "content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistan
t", "content": "SELECT InvoiceDate, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY In
voiceDate"}, {"role": "user", "content": " \n   Get the average invoice total for each customer:\n"}, {"r
ole": "assistant", "content": "SELECT CustomerId, AVG(Total) as avg_total \nFROM invoices \nGROUP BY Custom
erId"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names \n"},
{"role": "assistant", "content": "SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON
a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "what are the top 5 countries that customers come f
rom?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) as customer_count \nFROM customers \nGRO
UP BY Country \nORDER BY customer_count 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 tota
l quantity (across all invoices):\n"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T23:10:17.276367552Z', 'message': {'role': 'assistant', 'con
tent': 'SELECT i.CustomerId, SUM(ii.Quantity) as total_quantity \nFROM invoice_items ii \nJOIN invoices i
ON ii.InvoiceLineId = i.InvoiceId \nGROUP BY i.CustomerId \nORDER BY total_quantity DESC \nLIMIT 5'}, 'd
one_reason': 'stop', 'done': True, 'total_duration': 73266621642, 'load_duration': 704584, 'prompt_eval_cou

```
nt': 1272, 'prompt_eval_duration': 64086191000, 'eval_count': 49, 'eval_duration': 8500540000}
SELECT i.CustomerId, SUM(ii.Quantity) as total_quantity
FROM invoice_items ii
JOIN invoices i ON ii.InvoiceLineId = i.InvoiceId
GROUP BY i.CustomerId
ORDER BY total_quantity DESC
LIMIT 5
SELECT i.CustomerId, SUM(ii.Quantity) as total_quantity
FROM invoice_items ii
JOIN invoices i ON ii.InvoiceLineId = i.InvoiceId
GROUP BY i.CustomerId
ORDER BY total_quantity DESC
LIMIT 5
   CustomerId  total_quantity
0           1               7
1           2               7
2           3               7
3           4               7
4           5               7
Ollama parameters:
model=qwen2:7b,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Hint: album quantity is found in invoice_items, \n    \n
Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'\n\nThe Data
Frame was produced using this query: SELECT i.CustomerId, SUM(ii.Quantity) as total_quantity  \nFROM invoic
e_items ii \nJOIN invoices i ON ii.InvoiceLineId = i.InvoiceId  \nGROUP BY i.CustomerId  \nORDER BY total_q
uantity DESC  \nLIMIT 5\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunnin
g df.dtypes gives:\n CustomerId        int64\ntotal_quantity    int64\ndtype: object"}, {"role": "user", "c
ontent": "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 w
ith only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T23:10:55.282473646Z', 'message': {'role': 'assistant', 'con
tent': "```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n    fig = px.bar(df, x=df.CustomerI
d, y='total_quantity', \n                 title=f'Top Customer for Albums: {df.CustomerId.values[0]} - Tota
l Quantity: {df.total_quantity.values[0]}')\nelse:\n    fig = px.bar(df, x='total_quantity', y='CustomerI
d', orientation='h', \n                 title='Top 5 Customers Who Bought the Most Albums',\n
labels={'CustomerId': 'Customer ID', 'total_quantity': 'Total Album Quantity'})\nfig.update_layout(xaxis_ti
tle='Total Quantity of Albums')\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration':
```

37986293114, 'load_duration': 678108, 'prompt_eval_count': 219, 'prompt_eval_duration': 14106784000, 'eval_count': 140, 'eval_duration': 23746008000}



Out[36]:
```
('SELECT i.CustomerId, SUM(ii.Quantity) as total_quantity  \nFROM invoice_items ii \nJOIN invoices i ON i
i.InvoiceLineId = i.InvoiceId  \nGROUP BY i.CustomerId  \nORDER BY total_quantity DESC  \nLIMIT 5',
    CustomerId  total_quantity
 0           1              7
 1           2              7
 2           3              7
 3           4              7
 4           5              7,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'Total Album Quantity=%{x}<br>Customer ID=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'h',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([7, 7, 7, 7, 7]),
               'xaxis': 'x',
               'y': array([1, 2, 3, 4, 5]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Top 5 Customers Who Bought the Most Albums'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Total Quantity of Album
s'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Customer ID'}}}
 }))
```

```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 [37]:
```
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
```

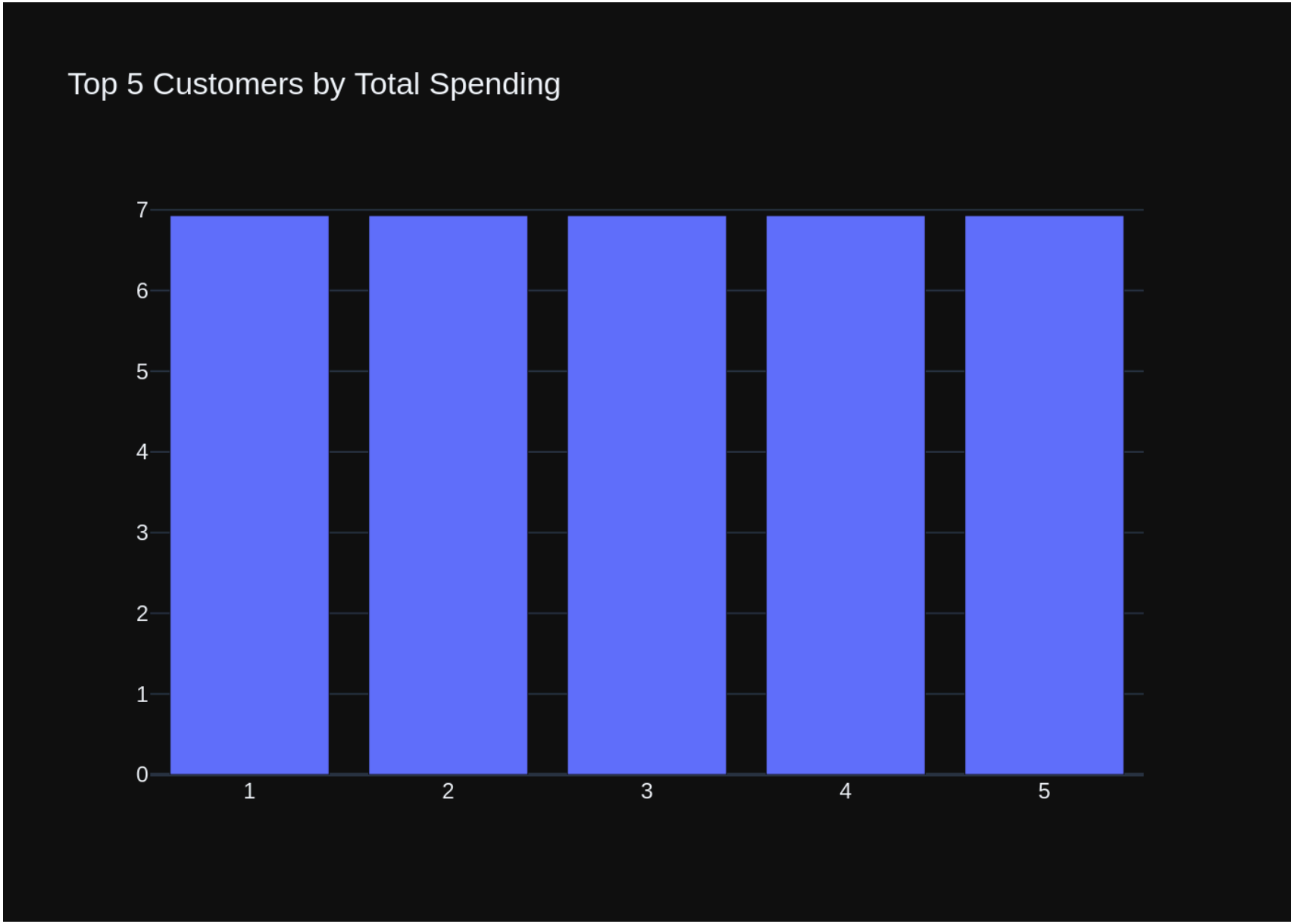
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY 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 chinook 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 qu
antity 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, SUM(ii.Quantit
y) as total_quantity \nFROM invoice_items ii \nJOIN invoices i ON ii.InvoiceLineId = i.InvoiceId \nGROUP
BY i.CustomerId \nORDER BY total_quantity DESC \nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find th
e customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.Invo
iceId) as invoice_count \nFROM customers c \nJOIN invoices i ON c.CustomerId = i.CustomerId \nGROUP BY c.C
ustomerId \nORDER BY invoice_count DESC \nLIMIT 1'}, {'role': 'user', 'content': ' \n   Get the average
invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) as av
g_total \nFROM invoices \nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   Get the total number
of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) as invoice
_count \nFROM invoices \nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   Find the top 5 most ex
pensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks ORDER BY
UnitPrice DESC LIMIT 5'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per countr
y:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(*) as invoice_count \nFROM invoices
\nGROUP BY BillingCountry'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding
$10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'user', 'co
ntent': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT
Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC \nLI
MIT 5'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amount invoice
d:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceDate, Total \nFROM invoices \nWHERE InvoiceDate >=
'2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': ' \n   There are 3 tables: artists, albu
ms 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 AS artist_name, COUNT(t.TrackId) as track_count\nFROM artists a\nJOIN albums al ON a.A
rtistId = al.ArtistId \nJOIN tracks t ON al.AlbumId = t.AlbumId \nGROUP BY a.ArtistId\nORDER BY track_coun
t DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n   Find the top 5 customers who spent the most money
overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items de
tail table is unnecessary \n'}]
Ollama parameters:
model=qwen2:7b,
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 chinook 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, SUM(ii.Quantity) as total_quantity \nFROM invoice_items ii \nJOIN invoices i ON ii.InvoiceLineId =
i.InvoiceId \nGROUP BY i.CustomerId \nORDER BY total_quantity DESC \nLIMIT 5"}, {"role": "user", "conten
t": " \n   Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.Cus
tomerId, COUNT(i.InvoiceId) as invoice_count \nFROM customers c \nJOIN invoices i ON c.CustomerId = i.Cust
omerId \nGROUP BY c.CustomerId \nORDER BY invoice_count DESC \nLIMIT 1"}, {"role": "user", "content": "

```
\n    Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT Custom
erId, AVG(Total) as avg_total \nFROM invoices \nGROUP BY CustomerId"}, {"role": "user", "content": "  \n
Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT CustomerI
d, COUNT(*) as invoice_count \nFROM invoices \nGROUP BY CustomerId"}, {"role": "user", "content": "  \n
Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT *
FROM tracks ORDER BY UnitPrice DESC LIMIT 5"}, {"role": "user", "content": "  \n    Find the total number o
f invoices per country:\n"}, {"role": "assistant", "content": "SELECT BillingCountry, COUNT(*) as invoice_c
ount \nFROM invoices  \nGROUP BY BillingCountry"}, {"role": "user", "content": "  \n    List all invoices w
ith a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total > 1
0"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assis
tant", "content": "SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER B
Y customer_count DESC \nLIMIT 5"}, {"role": "user", "content": "  \n    Find all invoices since 2010 and th
e total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceDate, Total \nFROM invoices
\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "content": " \n    There are
3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks ar
e linked by AlbumId,\n    Can you find the top 10 most popular artists based on the number of tracks\n"},
{"role": "assistant", "content": "SELECT a.Name AS artist_name, COUNT(t.TrackId) as track_count\nFROM artis
ts a\nJOIN albums al ON a.ArtistId = al.ArtistId \nJOIN tracks t ON al.AlbumId = t.AlbumId  \nGROUP BY a.Ar
tistId\nORDER BY track_count DESC\nLIMIT 10"}, {"role": "user", "content": "  \n     Find the top 5 custome
rs who spent the most money overall, \n     \n     Hint: order total can be found on invoices table, calcul
ation using invoice_items detail table is unnecessary \n"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T23:12:59.078130097Z', 'message': {'role': 'assistant', 'con
tent': 'SELECT i.CustomerId, SUM(ii.UnitPrice * ii.Quantity) as total_spent  \nFROM invoices i  \nJOIN invo
ice_items ii ON i.InvoiceId = ii.InvoiceLineId  \nGROUP BY i.CustomerId  \nORDER BY total_spent DESC  \nLIM
IT 5'}, 'done_reason': 'stop', 'done': True, 'total_duration': 123704745564, 'load_duration': 751587, 'prom
pt_eval_count': 1659, 'prompt_eval_duration': 113308921000, 'eval_count': 55, 'eval_duration': 9707568000}
SELECT i.CustomerId, SUM(ii.UnitPrice * ii.Quantity) as total_spent
FROM invoices i
JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceLineId
GROUP BY i.CustomerId
ORDER BY total_spent DESC
LIMIT 5
SELECT i.CustomerId, SUM(ii.UnitPrice * ii.Quantity) as total_spent
FROM invoices i
JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceLineId
GROUP BY i.CustomerId
ORDER BY total_spent DESC
LIMIT 5
   CustomerId  total_spent
0           1         6.93
1           2         6.93
```

```
2           3         6.93
3           4         6.93
4           5         6.93
Ollama parameters:
model=qwen2:7b,
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 i.CustomerId, SUM(ii.UnitPr
ice * ii.Quantity) as total_spent  \nFROM invoices i  \nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceLi
neId  \nGROUP BY i.CustomerId  \nORDER BY total_spent DESC  \nLIMIT 5\n\nThe following is information about
the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId       int64\ntotal_spent    fl
oat64\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': 'qwen2:7b', 'created_at': '2024-06-15T23:13:29.890815492Z', 'message': {'role': 'assistant', 'con
tent': 'import plotly.graph_objs as go\n\nif df.shape[0] == 1:\n    data = [go.Indicator(mode="number", val
ue=df[\'total_spent\'].iloc[0])]\nelse:\n    data = [\n        go.Bar(x=df[\'CustomerId\'], y=df[\'total_sp
ent\'])\n    ]\n\nlayout = go.Layout(title=\'Top 5 Customers by Total Spending\')\n\nfig = go.Figure(data=d
ata, layout=layout)\nfig.show()'}, 'done_reason': 'stop', 'done': True, 'total_duration': 30787590695, 'loa
d_duration': 677400, 'prompt_eval_count': 230, 'prompt_eval_duration': 14905866000, 'eval_count': 93, 'eval
_duration': 15739276000}
```


Top 5 Customers by Total Spending
7
6
5
4
3
2
1
0
1
2
3
4
5

Out[37]:
```
('SELECT i.CustomerId, SUM(ii.UnitPrice * ii.Quantity) as total_spent  \nFROM invoices i  \nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceLineId  \nGROUP BY i.CustomerId  \nORDER BY total_spent DESC  \nLIMIT 5',
    CustomerId  total_spent
 0           1         6.93
 1           2         6.93
 2           3         6.93
 3           4         6.93
 4           5         6.93,
 Figure({
     'data': [{'type': 'bar', 'x': array([1, 2, 3, 4, 5]), 'y': array([6.93, 6.93, 6.93, 6.93, 6.93])}],
     'layout': {'template': '...', 'title': {'text': 'Top 5 Customers by Total Spending'}}
 }))
```

In [38]:
```
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
```

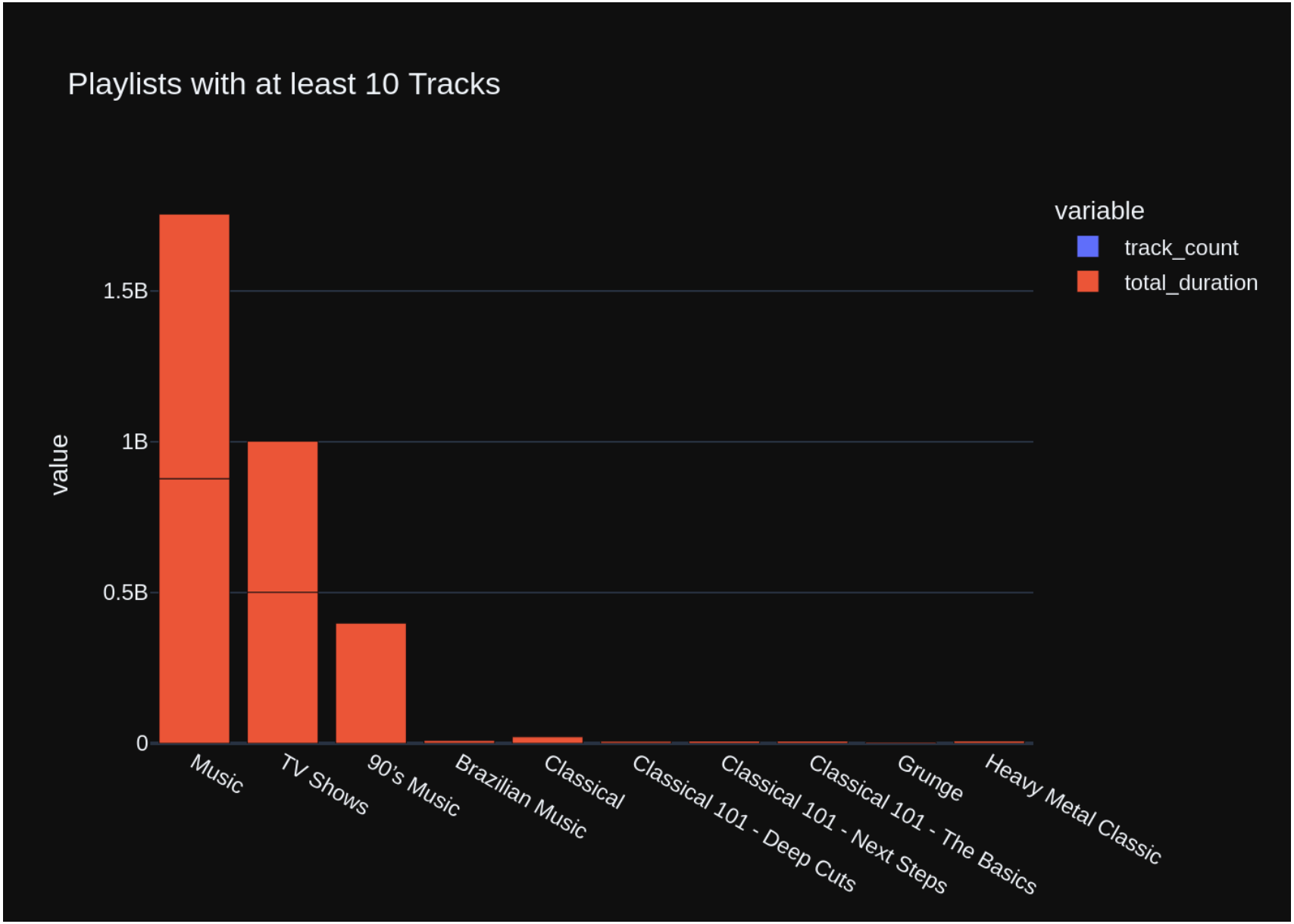
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (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 chinook 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.TrackId) as track_count\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 AS artist_name, COUNT(t.TrackId) as track_count\nFROM artists a\nJOIN albums al ON a.ArtistId = al.ArtistId \nJOIN tracks t ON al.AlbumId = t.AlbumId \nGROUP BY a.ArtistId\nORDER BY track_count DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names\n'}, {'role': 'assistant', 'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId'}, {'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 i

nvoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, SUM(ii.Quantity) as total_quantity
\nFROM invoice_items ii \nJOIN invoices i ON ii.InvoiceLineId = i.InvoiceId \nGROUP BY i.CustomerId \nORD
ER BY total_quantity DESC \nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 a
nd the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceDate, Total \nFROM invoi
ces \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': 'Can you list
all tables in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_mas
ter WHERE type='table'"}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $1
0:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'user', '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 i.CustomerId, SUM(ii.UnitPrice * ii.Quantity) as total_spent \nFROM invoic
es i \nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceLineId \nGROUP BY i.CustomerId \nORDER BY total_
spent DESC \nLIMIT 5'}, {'role': 'user', 'content': ' \n   Get all playlists containing at least 10 tra
cks and the total duration of those tracks:\n'}]
Ollama parameters:
model=qwen2:7b,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\n\nC
REATE TABLE \"playlists\"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NV
ARCHAR(120)\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackI
d INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN
KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n
FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE TABLE \"tracks\"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVA
RCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGE
R,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitP
rice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELET
E NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DE
LETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (Gen
reId)\n\nCREATE INDEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON \"tr
acks\" (MediaTypeId)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"albums
\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r
\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n   GenreId INTEGER PRIMARY
KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook da
tabase invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please gen
erate a valid SQL query without any explanations for the question. \n2. If the provided context is almost s

ufficient but requires knowledge of a specific string in a particular column, please generate an intermedia
te SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermedi
ate_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.TrackId) as tr
ack_count\nFROM genres g \nJOIN tracks t ON g.GenreId = t.GenreId \nGROUP BY g.Name"}, {"role": "user", "co
ntent": " \n   There are 3 tables: artists, albums and tracks, where albums and artists are linked by Arti
stId, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most popular artists based on t
he number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name AS artist_name, COUNT(t.TrackId) a
s track_count\nFROM artists a\nJOIN albums al ON a.ArtistId = al.ArtistId \nJOIN tracks t ON al.AlbumId =
t.AlbumId \nGROUP BY a.ArtistId\nORDER BY track_count DESC\nLIMIT 10"}, {"role": "user", "content": " \n
Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "S
ELECT * FROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, {"role": "user", "content": " \n   Find th
e top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT * FROM t
racks ORDER BY UnitPrice DESC LIMIT 5"}, {"role": "user", "content": " \n   List all albums and their cor
responding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, ar.Name AS ArtistName\nFRO
M 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 tota
l quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, SUM(ii.Quanti
ty) as total_quantity \nFROM invoice_items ii \nJOIN invoices i ON ii.InvoiceLineId = i.InvoiceId \nGROUP
BY i.CustomerId \nORDER BY total_quantity DESC \nLIMIT 5"}, {"role": "user", "content": " \n   Find all
invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceDat
e, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "co
ntent": "Can you list all tables in the SQLite database catalog?"}, {"role": "assistant", "content": "SELEC
T name FROM sqlite_master WHERE type='table'"}, {"role": "user", "content": " \n   List all invoices with
a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total > 10"},
{"role": "user", "content": " \n   Find the top 5 customers who spent the most money overall, \n   \n
Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessa
ry \n"}, {"role": "assistant", "content": "SELECT i.CustomerId, SUM(ii.UnitPrice * ii.Quantity) as total_sp
ent \nFROM invoices i \nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceLineId \nGROUP BY i.CustomerId
\nORDER BY total_spent DESC \nLIMIT 5"}, {"role": "user", "content": " \n   Get all playlists containin
g at least 10 tracks and the total duration of those tracks:\n"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T23:15:10.086323634Z', 'message': {'role': 'assistant', 'con
tent': 'SELECT p.Name, COUNT(t.TrackId) as track_count, SUM(t.Milliseconds) as total_duration \nFROM playli
sts p \nJOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId \nJOIN tracks t ON pt.TrackId = t.TrackId
\nGROUP BY p.PlaylistId\nHAVING COUNT(t.TrackId) >= 10'}, 'done_reason': 'stop', 'done': True, 'total_durat
ion': 100119491953, 'load_duration': 774823, 'prompt_eval_count': 1289, 'prompt_eval_duration': 8695506100
0, 'eval_count': 72, 'eval_duration': 12477569000}
SELECT p.Name, COUNT(t.TrackId) as track_count, SUM(t.Milliseconds) as total_duration
FROM playlists p

```
JOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId
JOIN tracks t ON pt.TrackId = t.TrackId
GROUP BY p.PlaylistId
HAVING COUNT(t.TrackId) >= 10
SELECT p.Name, COUNT(t.TrackId) as track_count, SUM(t.Milliseconds) as total_duration
FROM playlists p
JOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId
JOIN tracks t ON pt.TrackId = t.TrackId
GROUP BY p.PlaylistId
HAVING COUNT(t.TrackId) >= 10
                         Name  track_count  total_duration
0                       Music         3290       877683083
1                    TV Shows          213       501094957
2                  90’s Music         1477       398705153
3                       Music         3290       877683083
4                    TV Shows          213       501094957
5             Brazilian Music           39         9486559
6                    Classical           75        21770592
7    Classical 101 - Deep Cuts           25         6755730
8   Classical 101 - Next Steps           25         7575051
9   Classical 101 - The Basics           25         7439811
10                     Grunge           15         4122018
11        Heavy Metal Classic           26         8206312
Ollama parameters:
model=qwen2:7b,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n     Get all playlists containing at least 10 tracks and the
total duration of those tracks:\n'\n\nThe DataFrame was produced using this query: SELECT p.Name, COUNT(t.T
rackId) as track_count, SUM(t.Milliseconds) as total_duration \nFROM playlists p \nJOIN playlist_track pt O
N p.PlaylistId = pt.PlaylistId \nJOIN tracks t ON pt.TrackId = t.TrackId  \nGROUP BY p.PlaylistId\nHAVING C
OUNT(t.TrackId) >= 10\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning
df.dtypes gives:\n Name              object\ntrack_count        int64\ntotal_duration     int64\ndtype: obj
ect"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dat
aframe? 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': 'qwen2:7b', 'created_at': '2024-06-15T23:15:51.148095241Z', 'message': {'role': 'assistant', 'con
tent': '```python\nimport plotly.express as px\n\nif len(df) == 1:\n    fig = px.indicators.Islot(\n
title="Playlist with Tracks",\n        value=df[\'total_duration\'].values[0],\n        labels={\'value\':
```

\'Total Duration\'},\n   secondary_value=df[\'track_count\'].values[0],\n   secondary_label=\'Num
ber of Tracks\'\n   )\nelse:\n   fig = px.bar(df, x=\'Name\', y=[\'track_count\', \'total_duration\'], \n
title="Playlists with at least 10 Tracks",\n   labels={\'x\': \'Playlist Name\',\n
\'y\': [\'Tracks Count\', \'Total Duration (milliseconds)\']})\n\nfig.update_layout(xaxis_title=None)\nfig.
show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 41036603390, 'load_duration': 672661,
'prompt_eval_count': 232, 'prompt_eval_duration': 15052260000, 'eval_count': 152, 'eval_duration': 25845273
000}


Playlists with at least 10 Tracks
variable
track_count
total_duration
value
1.5B
1B
0.5B
0
Music
TV Shows
90's Music
Brazilian Music
Classical
Classical 101 - Deep Cuts
Classical 101 - Next Steps
Classical 101 - The Basics
Grunge
Heavy Metal Classic

Out[38]:
```
('SELECT p.Name, COUNT(t.TrackId) as track_count, SUM(t.Milliseconds) as total_duration \nFROM playlists p
\nJOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId \nJOIN tracks t ON pt.TrackId = t.TrackId  \nGROU
P BY p.PlaylistId\nHAVING COUNT(t.TrackId) >= 10',
                          Name  track_count  total_duration
 0                       Music         3290       877683083
 1                    TV Shows          213       501094957
 2                  90’s Music         1477       398705153
 3                       Music         3290       877683083
 4                    TV Shows          213       501094957
 5              Brazilian Music           39         9486559
 6                    Classical           75        21770592
 7    Classical 101 - Deep Cuts           25         6755730
 8   Classical 101 - Next Steps           25         7575051
 9   Classical 101 - The Basics           25         7439811
 10                     Grunge           15         4122018
 11         Heavy Metal Classic           26         8206312,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'variable=track_count<br>Name=%{x}<br>value=%{y}<extra></extra>',
               'legendgroup': 'track_count',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': 'track_count',
               'offsetgroup': 'track_count',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Music', 'TV Shows', '90’s Music', 'Music', 'TV Shows',
                           'Brazilian Music', 'Classical', 'Classical 101 - Deep Cuts',
                           'Classical 101 - Next Steps', 'Classical 101 - The Basics', 'Grunge',
                           'Heavy Metal Classic'], dtype=object),
               'xaxis': 'x',
               'y': array([3290,  213, 1477, 3290,  213,   39,   75,   25,   25,   25,   15,   26]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'variable=total_duration<br>Name=%{x}<br>value=%{y}<extra></extra>',
               'legendgroup': 'total_duration',
               'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
               'name': 'total_duration',
               'offsetgroup': 'total_duration',
               'orientation': 'v',
               'showlegend': True,
```

```
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['Music', 'TV Shows', '90’s Music', 'Music', 'TV Shows',
                                'Brazilian Music', 'Classical', 'Classical 101 - Deep Cuts',
                                'Classical 101 - Next Steps', 'Classical 101 - The Basics', 'Grunge',
                                'Heavy Metal Classic'], dtype=object),
                    'xaxis': 'x',
                    'y': array([877683083, 501094957, 398705153, 877683083, 501094957,   9486559,
                                 21770592,   6755730,   7575051,   7439811,   4122018,   8206312]),
                    'yaxis': 'y'}],
         'layout': {'barmode': 'relative',
                    'legend': {'title': {'text': 'variable'}, 'tracegroupgap': 0},
                    'template': '...',
                    'title': {'text': 'Playlists with at least 10 Tracks'},
                    'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {}},
                    'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'value'}}}
     }))
```

In [39]:

```
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 chinook 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 AS artist_name, COUNT(t.TrackId) as track_count\nFROM artists a\nJOIN albums al ON a.ArtistId = al.ArtistId \nJOIN tracks t ON al.AlbumId = t.AlbumId \nGROUP BY a.ArtistId\nORDER BY track_count DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.TrackId) as track_count\nFROM genres g \nJOIN tracks t ON g.GenreId = t.GenreId \nGROUP BY g.Name'}, {'role': 'user', 'content': ' \n   Get all playlists containing at least 10 tracks and the total duration of those tracks:\n'}, {'role': 'assistant', 'content': 'SELECT p.Name, COUNT(t.TrackId) as track_count, SUM(t.Milliseconds) as total_duration \nFROM playlists p \nJOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId \nJOIN tracks t ON pt.TrackId = t.TrackId \nGROUP BY p.PlaylistId\nHAVING COUNT(t.TrackId) >= 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 quantit

y (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, SUM(ii.Quantity) as to
tal_quantity \nFROM invoice_items ii \nJOIN invoices i ON ii.InvoiceLineId = i.InvoiceId \nGROUP BY i.Cus
tomerId \nORDER BY total_quantity DESC \nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the top 5 m
ost expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks ORD
ER BY UnitPrice DESC LIMIT 5'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing
"What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%Wha
t%' COLLATE NOCASE"}, {'role': 'user', 'content': 'Can you list all tables in the SQLite database catalo
g?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'use
r', 'content': ' \n   Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'S
ELECT c.CustomerId, COUNT(i.InvoiceId) as invoice_count \nFROM customers c \nJOIN invoices i ON c.Customer
Id = i.CustomerId \nGROUP BY c.CustomerId \nORDER BY invoice_count DESC \nLIMIT 1'}, {'role': 'user', 'co
ntent': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT
Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nORDER BY customer_count DESC \nLI
MIT 5'}, {'role': 'user', 'content': ' \n   Identify artists who have albums with tracks appearing in mu
ltiple genres:\n\n\n'}]
Ollama parameters:
model=qwen2:7b,
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 chinook database invoi
ce means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a vali

d SQL query without any explanations for the question. \n2. If the provided context is almost sufficient bu
t requires knowledge of a specific string in a particular column, please generate an intermediate SQL query
to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3.
If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most
relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer 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 AS artist_name, COUNT(t.TrackId) as track_count\nFROM artists a\nJOIN albums al ON a.Art
istId = al.ArtistId \nJOIN tracks t ON al.AlbumId = t.AlbumId \nGROUP BY a.ArtistId\nORDER BY track_count
DESC\nLIMIT 10"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names
\n"}, {"role": "assistant", "content": "SELECT a.Title, ar.Name AS ArtistName\nFROM albums a\nJOIN artists
ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   List all genres and the number of tr
acks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.TrackId) as track_count\n
FROM genres g \nJOIN tracks t ON g.GenreId = t.GenreId \nGROUP BY g.Name"}, {"role": "user", "content": "
\n   Get all playlists containing at least 10 tracks and the total duration of those tracks:\n"}, {"rol
e": "assistant", "content": "SELECT p.Name, COUNT(t.TrackId) as track_count, SUM(t.Milliseconds) as total_d
uration \nFROM playlists p \nJOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId \nJOIN tracks t ON pt.T
rackId = t.TrackId \nGROUP BY p.PlaylistId\nHAVING COUNT(t.TrackId) >= 10"}, {"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, SUM(ii.Quantity) as total_quantity \nFROM invoice_items ii \nJOIN invoices i ON ii.InvoiceLineId = i.
InvoiceId \nGROUP BY i.CustomerId \nORDER BY total_quantity DESC \nLIMIT 5"}, {"role": "user", "conten
t": " \n   Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "conten
t": "SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5"}, {"role": "user", "content": " \n   Find all
tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * F
ROM tracks WHERE Name LIKE '%What%' COLLATE NOCASE"}, {"role": "user", "content": "Can you list all tables
in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE t
ype='table'"}, {"role": "user", "content": " \n   Find the customer with the most invoices \n"}, {"rol
e": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) as invoice_count \nFROM customers c
\nJOIN invoices i ON c.CustomerId = i.CustomerId \nGROUP BY c.CustomerId \nORDER BY invoice_count DESC \n
LIMIT 1"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role":
"assistant", "content": "SELECT Country, COUNT(*) as customer_count \nFROM customers \nGROUP BY Country \nO
RDER BY customer_count DESC \nLIMIT 5"}, {"role": "user", "content": " \n   Identify artists who have al
bums with tracks appearing in multiple genres:\n\n\n"}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T23:17:29.349574766Z', 'message': {'role': 'assistant', 'con
tent': 'SELECT a.Name AS artist_name, g1.Name AS genre1, g2.Name AS genre2 FROM artists a \nJOIN albums al
ON a.ArtistId = al.ArtistId \nJOIN tracks t ON al.AlbumId = t.AlbumId \nJOIN genres g1 ON t.GenreId = g1.G
enreId\nJOIN tracks t2 ON t.TrackId != t2.TrackId AND t.AlbumId = t2.AlbumId\nJOIN genres g2 ON t2.GenreId
= g2.GenreId\nGROUP BY a.ArtistId, t.GenreId, t2.GenreId\nHAVING COUNT(DISTINCT g1.Name) > 1 OR COUNT(DISTI
NCT g2.Name) > 1'}, 'done_reason': 'stop', 'done': True, 'total_duration': 98087959434, 'load_duration': 72

```
7230, 'prompt_eval_count': 1305, 'prompt_eval_duration': 70026338000, 'eval_count': 154, 'eval_duration': 2
7365556000}
SELECT a.Name AS artist_name, g1.Name AS genre1, g2.Name AS genre2 FROM artists a
JOIN albums al ON a.ArtistId = al.ArtistId
JOIN tracks t ON al.AlbumId = t.AlbumId
JOIN genres g1 ON t.GenreId = g1.GenreId
JOIN tracks t2 ON t.TrackId != t2.TrackId AND t.AlbumId = t2.AlbumId
JOIN genres g2 ON t2.GenreId = g2.GenreId
GROUP BY a.ArtistId, t.GenreId, t2.GenreId
HAVING COUNT(DISTINCT g1.Name) > 1 OR COUNT(DISTINCT g2.Name) > 1
SELECT a.Name AS artist_name, g1.Name AS genre1, g2.Name AS genre2 FROM artists a
JOIN albums al ON a.ArtistId = al.ArtistId
JOIN tracks t ON al.AlbumId = t.AlbumId
JOIN genres g1 ON t.GenreId = g1.GenreId
JOIN tracks t2 ON t.TrackId != t2.TrackId AND t.AlbumId = t2.AlbumId
JOIN genres g2 ON t2.GenreId = g2.GenreId
GROUP BY a.ArtistId, t.GenreId, t2.GenreId
HAVING COUNT(DISTINCT g1.Name) > 1 OR COUNT(DISTINCT g2.Name) > 1
Empty DataFrame
Columns: [artist_name, genre1, genre2]
Index: []
Ollama parameters:
model=qwen2:7b,
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.Name AS artist_name, g
1.Name AS genre1, g2.Name AS genre2 FROM artists a \nJOIN albums al ON a.ArtistId = al.ArtistId \nJOIN trac
ks t ON al.AlbumId = t.AlbumId  \nJOIN genres g1 ON t.GenreId = g1.GenreId\nJOIN tracks t2 ON t.TrackId !=
t2.TrackId AND t.AlbumId = t2.AlbumId\nJOIN genres g2 ON t2.GenreId = g2.GenreId\nGROUP BY a.ArtistId, t.Ge
nreId, t2.GenreId\nHAVING COUNT(DISTINCT g1.Name) > 1 OR COUNT(DISTINCT g2.Name) > 1\n\nThe following is in
formation about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n artist_name    object\nge
nre1         object\ngenre2         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 call
ed 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do n
ot answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'qwen2:7b', 'created_at': '2024-06-15T23:18:14.242274702Z', 'message': {'role': 'assistant', 'con
tent': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n    fig = px.bar(df,\n
x=df.index,\n                y=list(df.columns),\n                title="Unique Artist: " + str(list(df.i
```

```
ndex)[0]),\n                labels={list(df.index)[0]: "Artist Name",\n                        list(df.co
lumns)[0]: "Genre", \n                          list(df.columns)[1]: "Genre 2"})\nelse:\n    fig = px.scatte
r_geo(df,\n                         locations="artist_name",\n                         locationmode=\'count
ry names\',\n                           hover_name=[\'genre1\', \'genre2\'],\n                         title
="Artists with Multiple Genres",\n                          projection="natural earth")\n    \nfig.show()\n`
``'}, 'done_reason': 'stop', 'done': True, 'total_duration': 44872480034, 'load_duration': 668372, 'prompt_
eval_count': 305, 'prompt_eval_duration': 19758817000, 'eval_count': 146, 'eval_duration': 24972693000}
```

Out[39]:
```
('SELECT a.Name AS artist_name, g1.Name AS genre1, g2.Name AS genre2 FROM artists a \nJOIN albums al ON a.
ArtistId = al.ArtistId \nJOIN tracks t ON al.AlbumId = t.AlbumId  \nJOIN genres g1 ON t.GenreId = g1.Genre
Id\nJOIN tracks t2 ON t.TrackId != t2.TrackId AND t.AlbumId = t2.AlbumId\nJOIN genres g2 ON t2.GenreId = g
2.GenreId\nGROUP BY a.ArtistId, t.GenreId, t2.GenreId\nHAVING COUNT(DISTINCT g1.Name) > 1 OR COUNT(DISTINC
T g2.Name) > 1',
 Empty DataFrame
 Columns: [artist_name, genre1, genre2]
 Index: [],
 Figure({
     'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
               'hovertemplate': 'artist_name=%{label}<extra></extra>',
               'labels': array([], dtype=object),
               'legendgroup': '',
               'name': '',
               'showlegend': True,
               'type': 'pie'}],
     'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'template': '...'}
 }))
```

Check completion time

In [ ]:

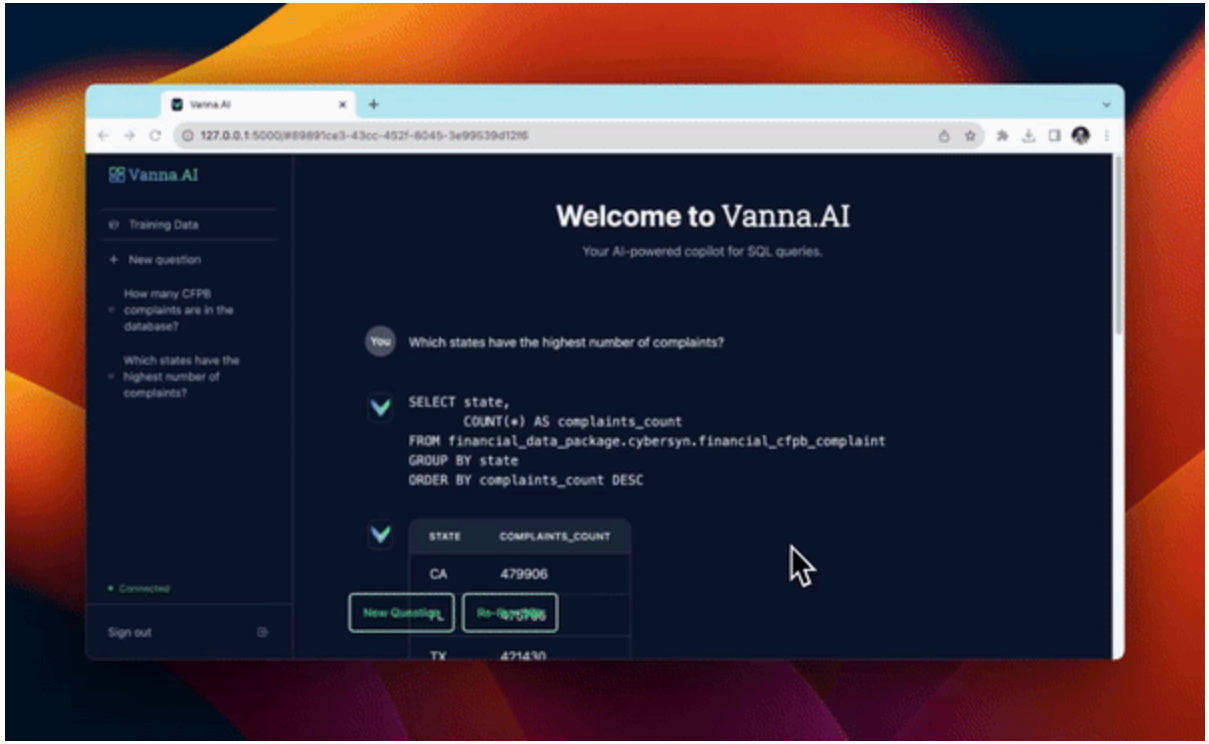
In [40]:
```
ts_stop = time()

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 'qwen2:7b' LLM took : 2568.68 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