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

!pip install 'vanna[chromadb]'

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]:

```
model_name = "gemma2" # 'gpt-3.5-turbo'
file_db = "~/Downloads/chinook.sqlite"
```

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)
```

Which database do you want to query?

- Postgres
- Microsoft SQL Server
- DuckDB
- Snowflake
- BigQuery
- [Selected] SQLite
- Other Database

  Use Vanna to generate queries for any SQL database

In [7]:
```
vn.run_sql_is_set
```

Out[7]:
```
True
```

In [8]:
```
clean_and_train = True  # False
```

In [9]:
```
hostname = os.uname().nodename
print("Hostname:", hostname)
```

```
Hostname: ducklover1
```

In [10]:
```
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 [11]:

```
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 [12]:

```
if clean_and_train:
    remove_collections()
```

Training

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

In [13]:

```
# show training data
training_data = vn.get_training_data()
training_data
```

Out[13]:

	id	question	content	training_data_type

In [14]:

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

```
In [15]: df_ddl
```

Out[15]:

	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 [16]:

```
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 [ ]:

```
```

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 [17]:

```
ts_start = time()
```

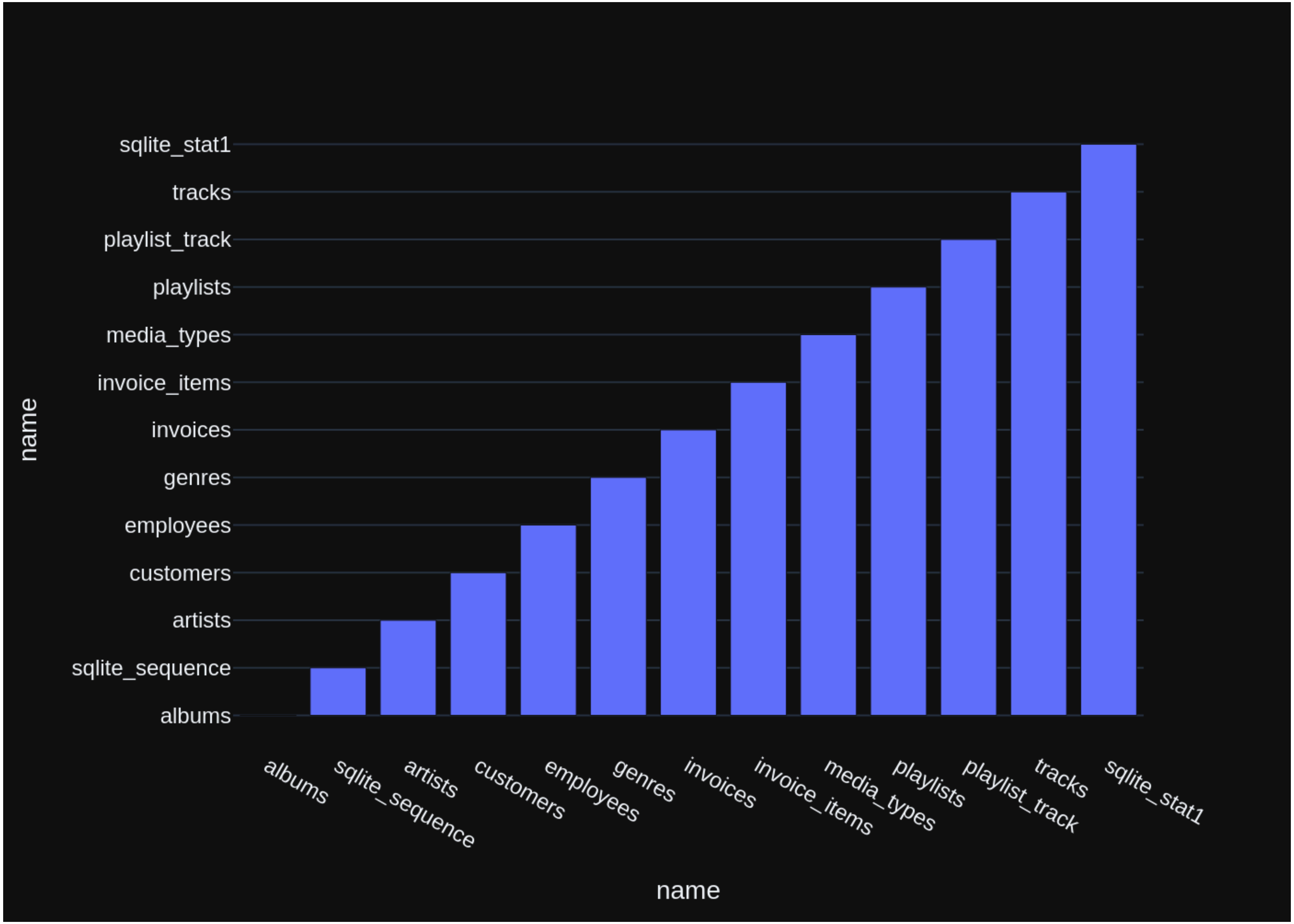
In [18]:

```
vn.ask(question="Show me a list of tables in the SQLite database")
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREA
TE TABLE "playlists"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r
\n)\n\nCREATE TABLE "media_types"\r\n(\r\n    MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NV
ARCHAR(120)\r\n)\n\nCREATE TABLE "artists"\r\n(\r\n    ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    N
ame 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 "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 N
O 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    Genr
eId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    Unit
Price NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTIO
N ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPD
ATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION O
N UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId IN
TEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (Playlist
Id) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackI
d) 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 I
NTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guideline
s \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the que
stion. \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 wit
h a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be ge
nerated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please
repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': 'Show me a list of tables in the
SQLite database'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: 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 \"pl
aylists\"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREAT
E TABLE \"media_types\"\r\n(\r\n    MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(12
0)\r\n)\n\nCREATE TABLE \"artists\"\r\n(\r\n    ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVA
RCHAR(120)\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Na
me 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\t
ON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DEL
ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREM
ENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n
GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    
UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES \"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 K
EY (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 ACT
ION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Respon
se 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 th
e query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it c
an't be generated. \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": "Show me a list of ta
bles in the SQLite database"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:03:33.092288247Z', 'message': {'role': 'assistant', 'conten
t': "SELECT name FROM sqlite_master WHERE type='table'; \n"}, 'done_reason': 'stop', 'done': True, 'total_duration':
62593879339, 'load_duration': 1124548593, 'prompt_eval_count': 849, 'prompt_eval_duration': 58242339000, 'eval_coun
t': 14, 'eval_duration': 3182305000}
LLM Response: SELECT name FROM sqlite_master WHERE type='table';

Info: 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
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: 'Show me a list of tables in the SQLite database'\n\nThe DataFrame was produced usi
ng 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\ndtype: object"}, {"role": "user", "content": "Ca
n you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas datafram
e 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."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:03:58.337772399Z', 'message': {'role': 'assistant', 'conten
t': '```python\nimport plotly.express as px\n\nif len(df) == 1:\n  fig = px.indicator(\n    df,\n    name="name",\n
value="name"\n  )\nelse:\n  fig = px.bar(df, x="name", y="name") \n```'}, 'done_reason': 'stop', 'done': True, 'tota
l_duration': 25210729091, 'load_duration': 43616418, 'prompt_eval_count': 146, 'prompt_eval_duration': 9010262000,
'eval_count': 71, 'eval_duration': 16111731000}
```


sqlite_stat1
tracks
playlist_track
playlists
media_types
invoice_items
invoices
genres
employees
customers
artists
sqlite_sequence
albums
name
albums
sqlite_sequence
artists
customers
employees
genres
invoices
invoice_items
media_types
playlists
playlist_track
tracks
sqlite_stat1
name

Out[18]:

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

In [19]:

```
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
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\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    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    Inv
oiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  N
OT 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) REFE
RENCES "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 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(24),\r\n    Fax NVA
RCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFER
ENCES "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    First
Name 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 NVARC
HAR(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 NVARCHAR(160)  NOT NUL
L,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELET
E 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===Ad
ditional 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, please generate an
intermediate 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. Please use th
e most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly
as it was given before. \n'}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'rol
e': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': "which
table stores customer's orders"}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: 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    Cus
tomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    Billing
City NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NV
ARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (Customer
Id) \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) REFERENCE
S \"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 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(24),\r\n    Fax NVA
RCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFER
ENCES \"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    Co
untry 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 UP
DATE NO ACTION\r\n)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE \"playlists\"\r\n(\r\n    PlaylistId I
NTEGER 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(16
0)  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 ACTIO
N,\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 NVA
RCHAR(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 quest
ion. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular c
olumn, 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 gener
ated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please re
peat the answer exactly as it was given before. \n"}, {"role": "user", "content": "Show me a list of tables in the S
QLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "u
ser", "content": "which table stores customer's orders"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:05:25.372558772Z', 'message': {'role': 'assistant', 'conten
t': 'invoices \n'}, 'done_reason': 'stop', 'done': True, 'total_duration': 86626175504, 'load_duration': 536306, 'pr
ompt_eval_count': 1102, 'prompt_eval_duration': 85749593000, 'eval_count': 4, 'eval_duration': 743214000}
LLM Response: invoices

invoices

Couldn't run sql:  Execution failed on sql 'invoices
': near "invoices": syntax error
```

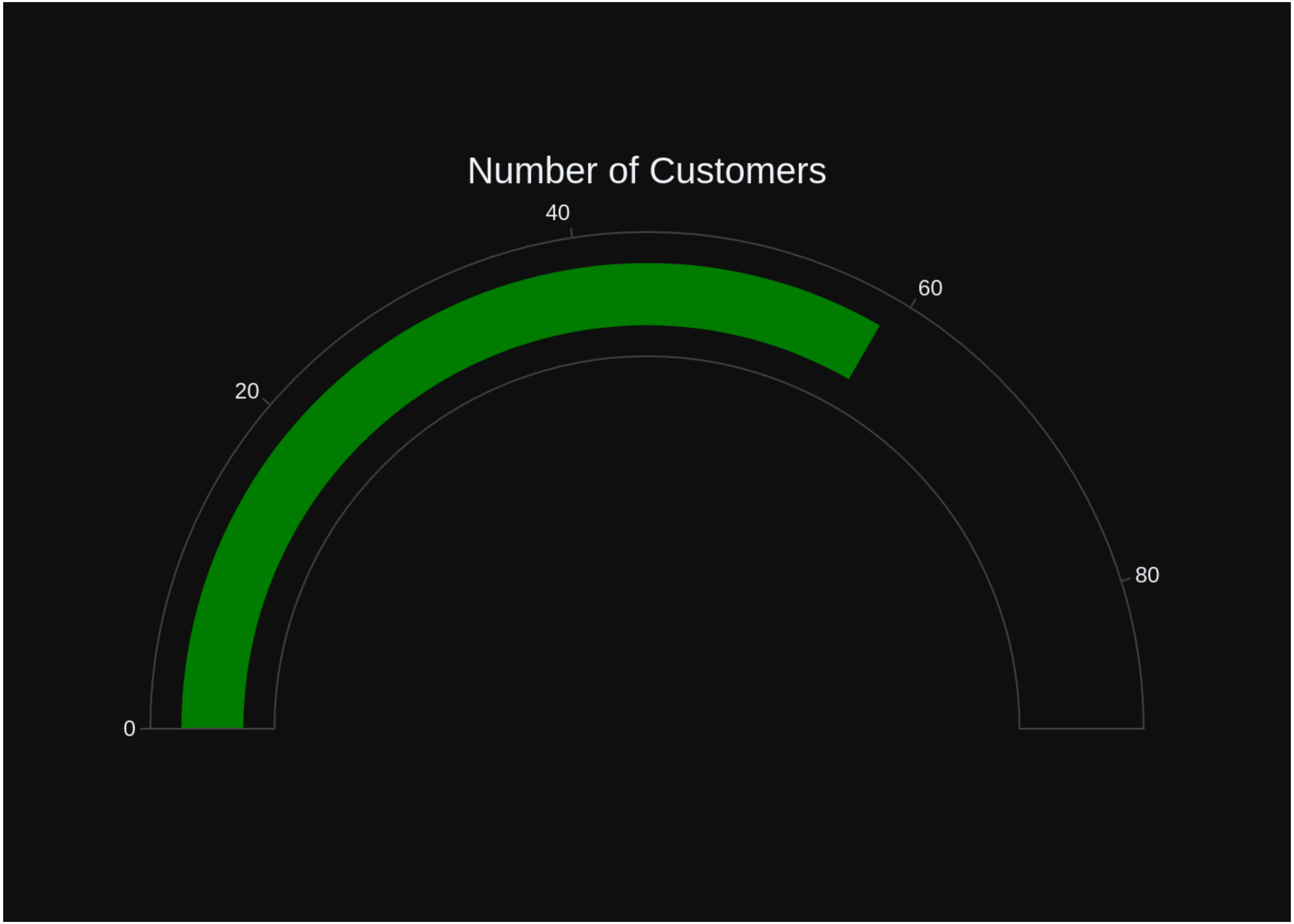
```
In [20]: vn.ask(question="How many customers are there")
```

```
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
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\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    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "cu
stomers" (SupportRepId)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\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    Posta
lCode 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 ACT
ION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE TABLE "invoi
ce_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NUL
L,\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 I
NDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGER PR
IMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FO
REIGN 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 NUL
L,\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    Posta
lCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KE
Y (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TAB
LE "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 provid
ed context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the p
rovided context is almost sufficient but requires knowledge of a specific string in a particular column, please gene
rate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying
intermediate_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 the answer
exactly as it was given before. \n'}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite databas
e'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'conten
t': 'How many customers are there'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: 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    Cus
tomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    Billing
City NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NV
ARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (Customer
Id) \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    Addres
s 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    Suppor
tRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE TABLE \"invoic
e_items\"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NUL
L,\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\n
CREATE INDEX IFK_InvoiceLineInvoiceId ON \"invoice_items\" (InvoiceId)\n\nCREATE TABLE \"albums\"\r\n(\r\n    AlbumI
d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NUL
L,\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 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(6
0),\r\n    FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO AC
TION\r\n)\n\nCREATE TABLE \"playlists\"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Na
me NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guid
elines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for th
e question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a parti
cular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the que
ry 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, p
lease repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "Show me a list of tables
in the SQLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"r
ole": "user", "content": "How many customers are there"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:06:32.337278469Z', 'message': {'role': 'assistant', 'conten
t': 'SELECT COUNT(*) FROM customers;'}, 'done_reason': 'stop', 'done': True, 'total_duration': 66923736139, 'load_du
ration': 728197, 'prompt_eval_count': 836, 'prompt_eval_duration': 65253204000, 'eval_count': 7, 'eval_duration': 15
30256000}
```

```
LLM Response: SELECT COUNT(*) FROM customers;
Info: Output from LLM: SELECT COUNT(*) FROM customers;
Extracted SQL: SELECT COUNT(*) FROM customers
SELECT COUNT(*) FROM customers
   COUNT(*)
0        59
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: 'How many customers are there'\n\nThe DataFrame was produced using this query: SELE
CT COUNT(*) FROM customers\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.d
types gives:\n COUNT(*)    int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly c
ode to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only on
e value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- ju
st the code."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:06:55.527695341Z', 'message': {'role': 'assistant', 'conten
t': "```python\nimport plotly.graph_objects as go\n\nfig = go.Figure(data=[go.Indicator(value=df['COUNT(*)'].iloc
[0],\n                                    mode='gauge',\n                                    title={'text': 'Number 
of Customers'})])\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 23170480949, 'load_du
ration': 41305451, 'prompt_eval_count': 138, 'prompt_eval_duration': 8674742000, 'eval_count': 63, 'eval_duration':
14405380000}
```


Number of Customers
0
20
40
60
80

```
Out[20]: ('SELECT COUNT(*) FROM customers',
             COUNT(*)
          0        59,
          Figure({
              'data': [{'mode': 'gauge', 'title': {'text': 'Number of Customers'}, 'type': 'indicator', 'value': 59}],
              'layout': {'template': '...'}
          }))
```

```
In [ ]:
```

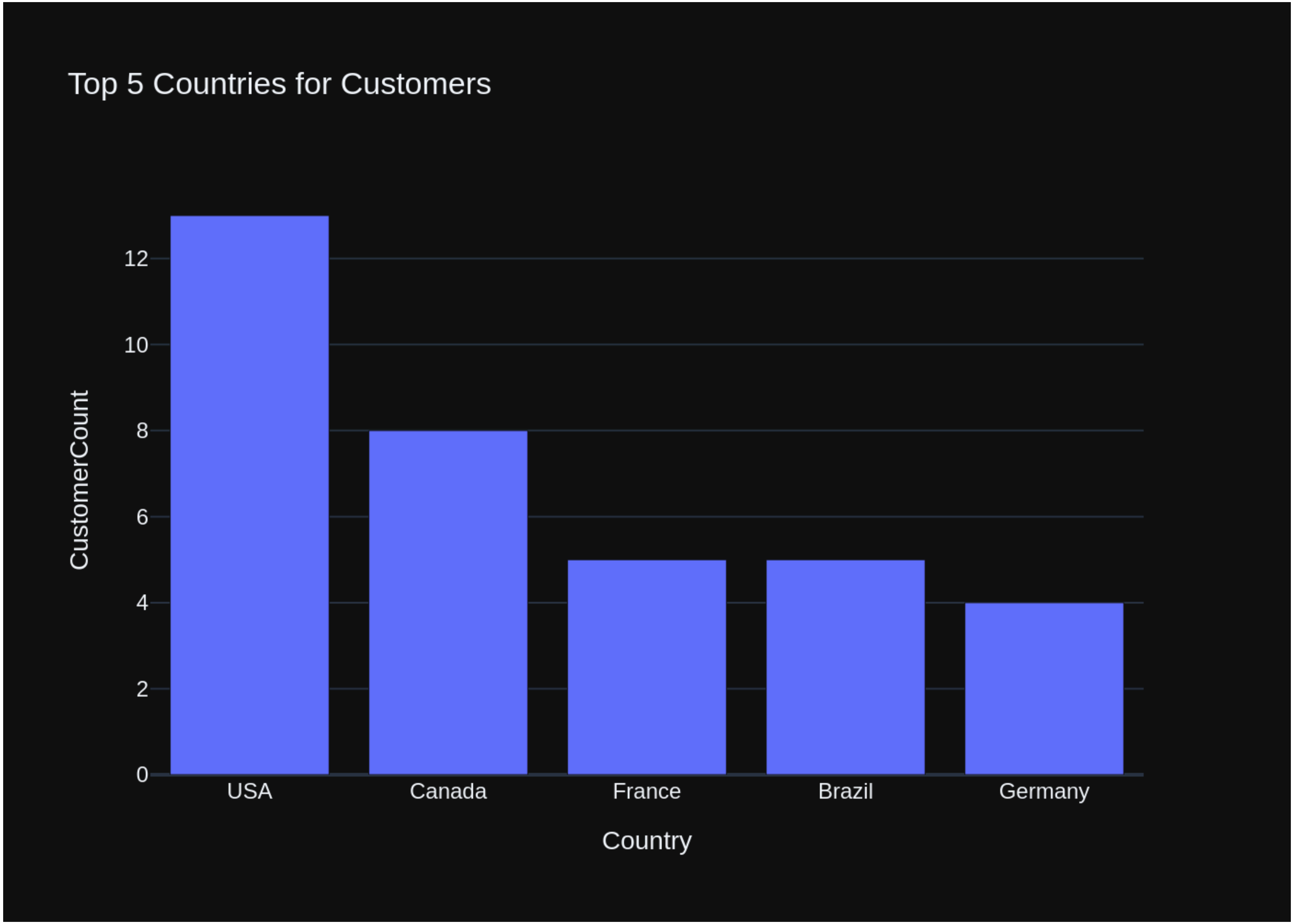
```
In [21]: vn.ask(question="what are the top 5 countries that customers come from?")
```

```
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
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\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    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n    Custome
rId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(2
0)  NOT NULL,\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 (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 I
NTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (I
nvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Tra
ckId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "media_type
s"\r\n(\r\n    MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDE
X 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 NUL
L,\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    Posta
lCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KE
Y (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\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 TABLE "playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    Tr
ackId 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    N
ame 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 N
O ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO
N,\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 \n
1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the questio
n. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular col
umn, 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 genera
ted. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please rep
```

```
eat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': 'How many customers are there'},
{'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': 'Show me a list of t
ables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='tabl
e'"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: 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    Cus
tomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    Billing
City NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NV
ARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (Customer
Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n    CustomerId INTEGE
R PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NUL
L,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State 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    I
nvoiceLineId 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 (InvoiceI
d) 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 INDE
X IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"employees\"\r\n(\r\n    EmployeeId INTE
GER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT N
ULL,\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    Posta
lCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KE
Y (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE T
ABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NUL
L,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DEL
ETE NO ACTION 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, TrackId),\r\n    F
OREIGN 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    Genr
eId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    Unit
Price NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACT
ION 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 ACT
ION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Respon
se 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 th
e query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it c
an't be generated. \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 customers a
re there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "Show me
a list of tables in the SQLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE ty
pe='table'"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:08:46.248592453Z', 'message': {'role': 'assistant', 'conten
t': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT
5;  \n'}, 'done_reason': 'stop', 'done': True, 'total_duration': 110630770977, 'load_duration': 562473, 'prompt_eval
_count': 1283, 'prompt_eval_duration': 103203374000, 'eval_count': 29, 'eval_duration': 7247340000}
LLM Response: SELECT Country, COUNT(*) AS CustomerCount
FROM customers
GROUP BY Country
ORDER BY CustomerCount DESC
LIMIT 5;

Info: Output from LLM: SELECT Country, COUNT(*) AS CustomerCount
FROM customers
GROUP BY Country
ORDER BY CustomerCount DESC
LIMIT 5;

Extracted SQL: SELECT Country, COUNT(*) AS CustomerCount
FROM customers
GROUP BY Country
ORDER BY CustomerCount DESC
LIMIT 5
SELECT Country, COUNT(*) AS CustomerCount
FROM customers
GROUP BY Country
ORDER BY CustomerCount DESC
LIMIT 5
   Country  CustomerCount
```

```
0      USA             13
1   Canada              8
2   France              5
3   Brazil              5
4  Germany              4
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: 'what are the top 5 countries that customers come from?'\n\nThe DataFrame was produ
ced using this query: SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY Customer
Count DESC\nLIMIT 5\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes g
ives:\n Country          object\nCustomerCount     int64\ndtype: object"}, {"role": "user", "content": "Can you gene
rate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 
'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer wi
th any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:09:18.581509079Z', 'message': {'role': 'assistant', 'conten
t': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n  fig = px.indicator(df, \n
name="CustomerCount",\n                    value="CustomerCount",\n                    title="Top Country for Custom
ers")\nelse:\n  fig = px.bar(df, x="Country", y="CustomerCount", title="Top 5 Countries for Customers")\n\nfig.show
()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 32312804311, 'load_duration': 41802672, 'prompt_ev
al_count': 168, 'prompt_eval_duration': 10625747000, 'eval_count': 94, 'eval_duration': 21556266000}
```


Top 5 Countries for Customers
CustomerCount
12
10
8
6
4
2
0
USA
Canada
France
Brazil
Germany
Country

```
Out[21]: ('SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT
         5',
             Country  CustomerCount
          0      USA             13
          1   Canada              8
          2   France              5
          3   Brazil              5
          4  Germany              4,
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'Country=%{x}<br>CustomerCount=%{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 for Customers'},
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Country'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerCount'}}}
          }))
```

More SQL questions

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

```
In [22]: 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
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \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  N
OT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI
ON\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHA
R(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Compo
ser 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    FO
REIGN 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 PRIMA
RY 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    Play
listId 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_TrackMediaTyp
eId ON "tracks" (MediaTypeId)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Respo
nse Guidelines \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 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 be
fore, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': 'Show me a list of
tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='tabl
e'"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant',
'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC
\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT C
OUNT(*) FROM customers'}, {'role': 'user', 'content': '  \n    List all albums and their corresponding artist names
\n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: 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 NUL
L,\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    Compos
er NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT N
ULL,\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 INTEGE
R 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 \"gen
res\"\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 mean
s order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query with
out any explanations for the question. \n2. If the provided context is almost sufficient but requires 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 insufficient, pl
ease explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been as
ked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": 
"Show me a list of tables in the SQLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master 
WHERE type='table'"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"rol
e": "assistant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY C
ustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "co
ntent": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "  \n    List all albums and their correspond
ing artist names  \n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:10:21.604237725Z', 'message': {'role': 'assistant', 'conten
t': 'SELECT \n    a.Title,\n    ar.Name\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId;'}, 'done_reaso
n': 'stop', 'done': True, 'total_duration': 62918122282, 'load_duration': 549245, 'prompt_eval_count': 755, 'prompt_
eval_duration': 54877315000, 'eval_count': 33, 'eval_duration': 7813154000}
LLM Response: SELECT 
    a.Title,
    ar.Name
FROM albums a
JOIN artists ar ON a.ArtistId = ar.ArtistId;
Info: Output from LLM: SELECT 
    a.Title,
    ar.Name
FROM albums a
JOIN artists ar ON a.ArtistId = ar.ArtistId;
Extracted SQL: SELECT 
    a.Title,
    ar.Name
FROM albums a
```

```
JOIN artists ar ON a.ArtistId = ar.ArtistId
SELECT
    a.Title,
    ar.Name
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...

                                                  Name
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]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n    List all albums and their corresponding artist names  \n'\n\nThe DataFrame
was produced using this query: SELECT \n    a.Title,\n    ar.Name\nFROM albums a\nJOIN artists ar ON a.ArtistId = a
r.ArtistId\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n T
itle    object\nName     object\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly co
de to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one
value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just
the code."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:10:52.877377872Z', 'message': {'role': 'assistant', 'conten
t': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n    px.indicator(\n        value=df[\'Title\'].
iloc[0],\n        title=f"{df[\'Name\'].iloc[0]} - {df[\'Title\'].iloc[0]}"\n    )\nelse:\n    px.bar(df, x=\'Name
\', y=\'Title\') \n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 31247728799, 'load_duration': 40887
174, 'prompt_eval_count': 173, 'prompt_eval_duration': 10878700000, 'eval_count': 89, 'eval_duration': 20278996000}
Couldn't run plotly code:  'NoneType' object has no attribute 'show'
```

```
Traceback (most recent call last):
  File "/home/gongai/anaconda3/envs/vanna/lib/python3.11/site-packages/vanna/base/base.py", line 1684, in ask
    img_bytes = fig.to_image(format="png", scale=2)
                ^^^^^^^^^^^^
AttributeError: 'NoneType' object has no attribute 'to_image'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/home/gongai/anaconda3/envs/vanna/lib/python3.11/site-packages/vanna/base/base.py", line 1687, in ask
    fig.show()
    ^^^^^^^^
AttributeError: 'NoneType' object has no attribute 'show'
```

In [23]:

```python
question = """
    Find all tracks with a name containing "What" (case-insensitive)
"""

vn.ask(question=question)
```

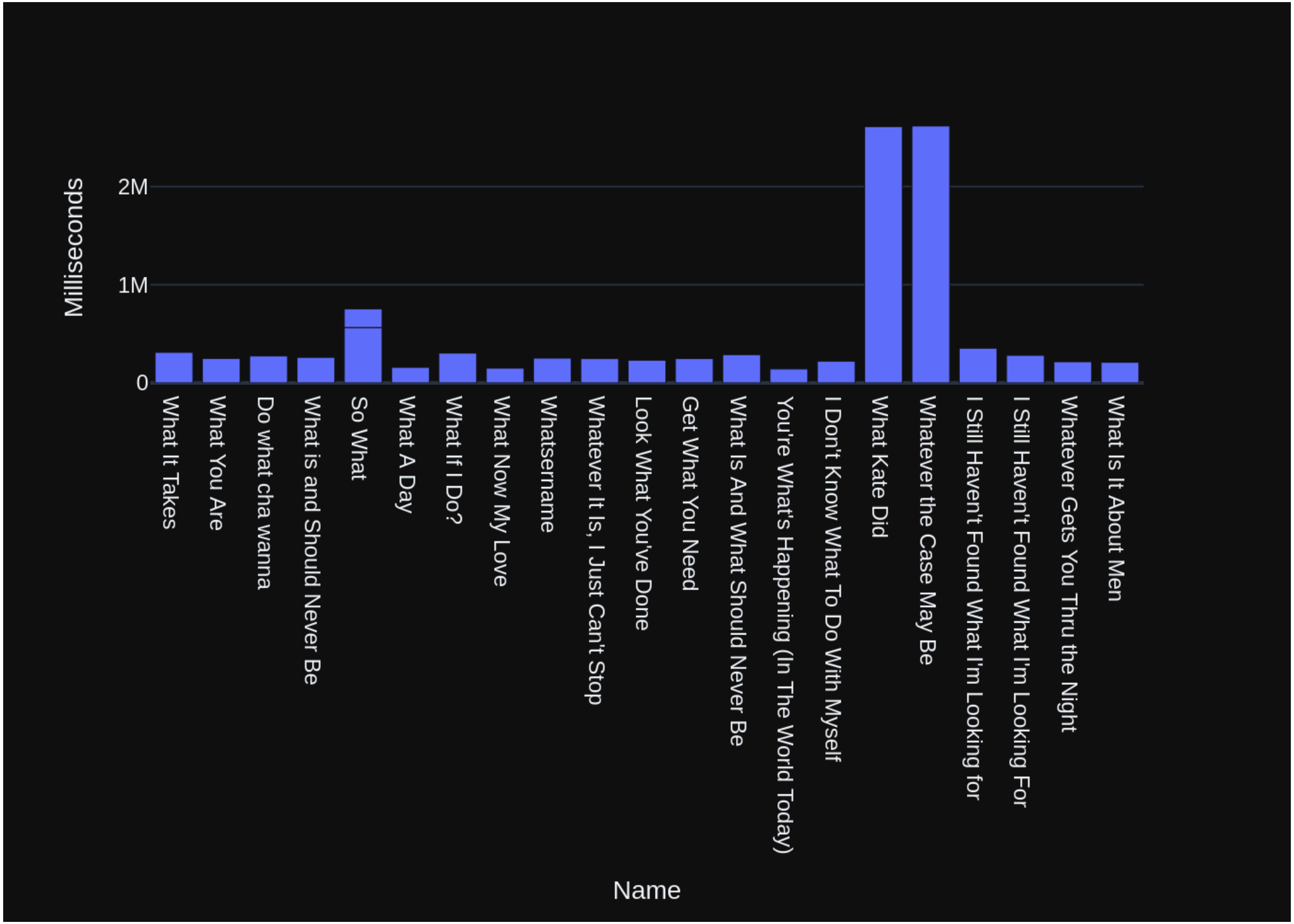
```
Number of requested results 10 is greater than number of elements in index 4, updating n_results = 4
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrac
kId 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    Genr
eId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    Unit
Price NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTIO
N ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPD
ATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION O
N 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 (Playli
stId) REFERENCES "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 IFK_InvoiceLi
neTrackId ON "invoice_items" (TrackId)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "pla
ylists"\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 prov
ided context is almost sufficient but requires knowledge of a specific string in a particular column, please generat
e an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying int
ermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please u
se the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exa
ctly as it was given before. \n'}, {'role': 'user', 'content': '  \n    List all albums and their corresponding arti
st names  \n'}, {'role': 'assistant', 'content': 'SELECT \n    a.Title,\n    ar.Name\nFROM albums a\nJOIN artists ar
ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'rol
e': '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 C
ustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content':
'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'use
r', 'content': '  \n    Find all tracks with a name containing "What" (case-insensitive)\n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: 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 \"pl
aylist_track\" (TrackId)\n\nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\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    UnitPric
e 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 UPD
ATE 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_TrackMediaType
Id ON \"tracks\" (MediaTypeId)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    T
rackId 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 K
EY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IF
K_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums\" (ArtistId)\n\nCR
EATE TABLE \"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    Name NVARC
HAR(120)\r\n)\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 quest
ion. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular c
olumn, 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 gener
ated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please re
peat 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 \n    a.Title,\n    ar.Name\nFROM albums 
a\nJOIN artists ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "Show me a list of tables in the SQLit
e database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "use
r", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT C
ountry, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"rol
e": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM custo
mers"}, {"role": "user", "content": "  \n    Find all tracks with a name containing \"What\" (case-insensitive)\n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:11:58.890971166Z', 'message': {'role': 'assistant', 'conten
t': "SELECT * FROM tracks WHERE Name LIKE '%What%' \n"}, 'done_reason': 'stop', 'done': True, 'total_duration': 6595
6302529, 'load_duration': 654344, 'prompt_eval_count': 836, 'prompt_eval_duration': 62605053000, 'eval_count': 13,
'eval_duration': 3042326000}
LLM Response: SELECT * FROM tracks WHERE Name LIKE '%What%'

SELECT * FROM tracks WHERE Name LIKE '%What%'

    TrackId                                     Name  AlbumId  \
0        26                            What It Takes        5   
1        88                             What You Are       10   
2       130                        Do what cha wanna       13   
3       342              What is and Should Never Be       30   
4       607                                  So What       48   
```

```
5      960                               What A Day       76
6     1000                             What If I Do?       80
7     1039                          What Now My Love       83
8     1145                               Whatsername       89
9     1440         Whatever It Is, I Just Can't Stop      116
10    1469                     Look What You've Done      119
11    1470                         Get What You Need      119
12    1628          What Is And What Should Never Be      133
13    1778  You're What's Happening (In The World Today)  146
14    1823                                   So What      149
15    2772        I Don't Know What To Do With Myself      223
16    2884                             What Kate Did      231
17    2893                  Whatever the Case May Be      230
18    2992    I Still Haven't Found What I'm Looking for  237
19    3007    I Still Haven't Found What I'm Looking For  238
20    3258          Whatever Gets You Thru the Night      255
21    3475                     What Is It About Men      322

    MediaTypeId  GenreId                                           Composer  \
0             1        1             Steven Tyler, Joe Perry, Desmond Child
1             1        1                             Audioslave/Chris Cornell
2             1        2                                         George Duke
3             1        1                             Jimmy Page/Robert Plant
4             1        2                                         Miles Davis
5             1        1                 Mike Bordin, Billy Gould, Mike Patton
6             1        1  Dave Grohl, Taylor Hawkins, Nate Mendel, Chris...
7             1       12          carl sigman/gilbert becaud/pierre leroyer
8             1        4                                           Green Day
9             1        1                                 Jay Kay/Kay, Jay
10            1        4                                           N. Cester
11            1        4                         C. Cester/C. Muncey/N. Cester
12            1        1                             Jimmy Page, Robert Plant
13            1       14              Allen Story/George Gordy/Robert Gordy
14            1        3                                         Culmer/Exalt
15            1        7                                                None
16            3       19                                                None
17            3       19                                                None
18            1        1     Bono/Clayton, Adam/Mullen Jr., Larry/The Edge
19            1        1                                                  U2
20            2        9                                                None
21            2        9  Delroy "Chris" Cooper, Donovan Jackson, Earl C...
```

```
    Milliseconds      Bytes  UnitPrice
0         310622   10144730       0.99
1         249391    5988186       0.99
2         274155    9018565       0.99
3         260675    8497116       0.99
4         564009   18360449       0.99
5         158275    5203430       0.99
6         302994    9929799       0.99
7         149995    4913383       0.99
8         252316    8244843       0.99
9         247222    8249453       0.99
10        230974    7517083       0.99
11        247719    8043765       0.99
12        287973    9369385       0.99
13        142027    4631104       0.99
14        189152    6162894       0.99
15        221387    7251478       0.99
16       2610250  484583988       1.99
17       2616410  183867185       1.99
18        353567   11542247       0.99
19        280764    9306737       0.99
20        215084    3499018       0.99
21        209573    3426106       0.99
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n    Find all tracks with a name containing \"What\" (case-insensitive)\n'\n\nT
he DataFrame was produced using this query: SELECT * FROM tracks WHERE Name LIKE '%What%' \n\n\nThe following is inf
ormation about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n TrackId           int64\nName      
object\nAlbumId           int64\nMediaTypeId       int64\nGenreId           int64\nComposer         object\nMillisec
onds      int64\nBytes             int64\nUnitPrice       float64\ndtype: object"}, {"role": "user", "content": "Can
you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe
called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not an
swer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:12:27.932258072Z', 'message': {'role': 'assistant', 'conten
t': "```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n  fig = px.indicator(df, \n
```

```
value='Name',\n                    title='Track Name')\nelse:\n  fig = px.bar(df, x='Name', y='Milliseconds')\n``
`"}, 'done_reason': 'stop', 'done': True, 'total_duration': 29019580727, 'load_duration': 630951, 'prompt_eval_coun
t': 205, 'prompt_eval_duration': 12967640000, 'eval_count': 69, 'eval_duration': 15921086000}
```


Milliseconds
2M
1M
0
What It Takes
What You Are
Do what cha wanna
What is and Should Never Be
So What
What A Day
What If I Do?
What Now My Love
Whatsername
Whatever It Is, I Just Can't Stop
Look What You've Done
Get What You Need
What Is And What Should Never Be
You're What's Happening (In The World Today)
I Don't Know What To Do With Myself
What Kate Did
Whatever the Case May Be
I Still Haven't Found What I'm Looking for
I Still Haven't Found What I'm Looking For
Whatever Gets You Thru the Night
What Is It About Men
Name

```
Out[23]:  ("SELECT * FROM tracks WHERE Name LIKE '%What%' \n",
               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  ,
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'Name=%{x}<br>Milliseconds=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'v',
              'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
              'x': array(['What It Takes', 'What You Are', 'Do what cha wanna',
                          'What is and Should Never Be', 'So What', 'What A Day', 'What If I Do?',
                          'What Now My Love', 'Whatsername', "Whatever It Is, I Just Can't Stop",
                          "Look What You've Done", 'Get What You Need',
                          'What Is And What Should Never Be',
                          "You're What's Happening (In The World Today)", 'So What',
                          "I Don't Know What To Do With Myself", 'What Kate Did',
                          'Whatever the Case May Be',
                          "I Still Haven't Found What I'm Looking for",
                          "I Still Haven't Found What I'm Looking For",
                          'Whatever Gets You Thru the Night', 'What Is It About Men'],
                         dtype=object),
              'xaxis': 'x',
              'y': array([ 310622,  249391,  274155,  260675,  564009,  158275,  302994,  149995,
                           252316,  247222,  230974,  247719,  287973,  142027,  189152,  221387,
                          2610250, 2616410,  353567,  280764,  215084,  209573]),
              'yaxis': 'y'}],
    'layout': {'barmode': 'relative',
               'legend': {'tracegroupgap': 0},
               'margin': {'t': 60},
               'template': '...',
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Milliseconds'}}}
}))
```

In [24]:

```
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
```

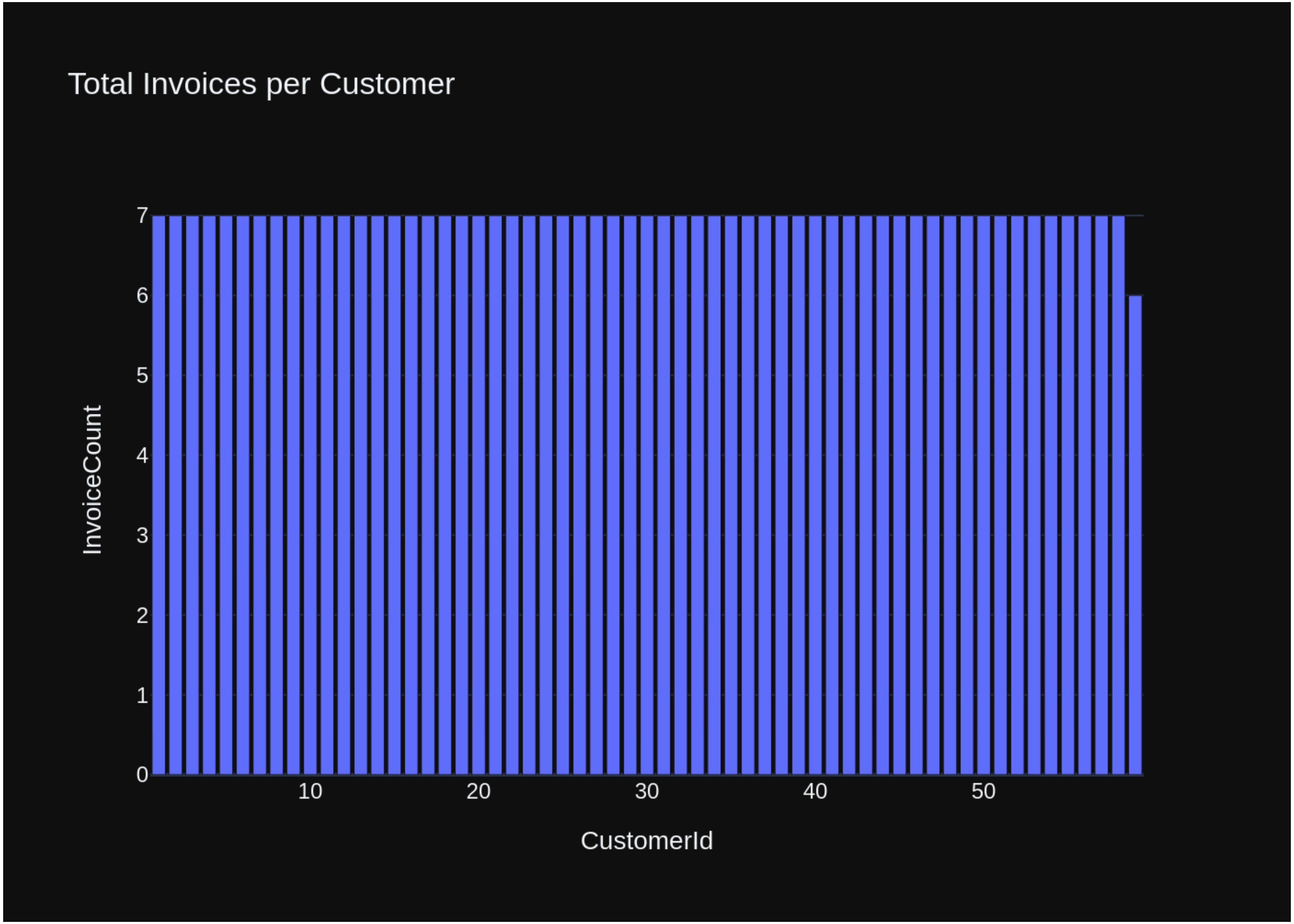
```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\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    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \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 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    FOREI
GN 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    Count
ry NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NV
ARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (Employe
eId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (S
upportRepId)\n\nCREATE TABLE "employees"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    L
astName 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(4
0),\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),\r\n    FOREIGN KEY (ReportsTo) REFERENCES "employees" (Emplo
yeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (Re
portsTo)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCH
AR(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(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    FO
REIGN 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 pr
ovided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If t
he 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 say
ing intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. P
lease use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the ans
wer exactly as it was given before. \n'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'ass
istant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': 'what are the top 5 countries tha
t customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customer
s\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    List all albums an
d their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT \n    a.Title,\n    ar.Name\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 LIK
E '%What%' \n"}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assistan
t', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': '  \n    Get the to
tal number of invoices for each customer\n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: 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    Cus
tomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    Billing
City NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NV
ARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (Customer
Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (Cust
omerId)\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    FOREI
GN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREI
GN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDE
X 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 NUL
L,\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 \"c
ustomers\" (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    Ci
ty NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phon
e NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES \"empl
oyees\" (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 NUL
L,\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    UnitPric
e 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 UPD
ATE 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 G
```

```
uidelines \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 par
ticular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the q
uery with 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 answered befor
e, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "How many customers ar
e there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "what are
the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS Custo
merCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "
\n    List all albums and their corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT \n    a.T
itle,\n    ar.Name\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 * F
ROM tracks WHERE Name LIKE '%What%' \n"}, {"role": "user", "content": "Show me a list of tables in the SQLite databa
se"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "conte
nt": "  \n    Get the total number of invoices for each customer\n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:14:15.778911363Z', 'message': {'role': 'assistant', 'conten
t': 'SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId;'}, 'done_reason': 'stop', 'don
e': True, 'total_duration': 107732244181, 'load_duration': 733806, 'prompt_eval_count': 1282, 'prompt_eval_duratio
n': 102793016000, 'eval_count': 19, 'eval_duration': 4542287000}
LLM Response: SELECT CustomerId, COUNT(*) AS InvoiceCount
FROM invoices
GROUP BY CustomerId;
Info: Output from LLM: SELECT CustomerId, COUNT(*) AS InvoiceCount
FROM invoices
GROUP BY CustomerId;
Extracted SQL: SELECT CustomerId, COUNT(*) AS InvoiceCount
FROM invoices
GROUP BY CustomerId
SELECT CustomerId, COUNT(*) AS InvoiceCount
FROM invoices
GROUP BY CustomerId
    CustomerId  InvoiceCount
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
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers 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 InvoiceCount\nFROM invoices\nGROUP BY CustomerId\n\nThe fo
llowing is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId      int64
\nInvoiceCount    int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to ch
art the results 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 -- just the co
de."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:14:46.527534059Z', 'message': {'role': 'assistant', 'conten
t': "```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n  fig = px.indicator(df, \n
value='InvoiceCount',\n                    title='Total Invoices per Customer')\nelse:\n  fig = px.bar(df, x='Custom
erId', y='InvoiceCount', title='Total Invoices per Customer')\n  \nfig.show()\n```"}, 'done_reason': 'stop', 'done':
True, 'total_duration': 30726660251, 'load_duration': 637268, 'prompt_eval_count': 165, 'prompt_eval_duration': 1035
4382000, 'eval_count': 89, 'eval_duration': 20241498000}
```


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

```
Out[24]: ('SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId',
              CustomerId  InvoiceCount
          0            1             7
          1            2             7
          2            3             7
          3            4             7
          4            5             7
          5            6             7
          6            7             7
          7            8             7
          8            9             7
          9           10             7
          10          11             7
          11          12             7
          12          13             7
          13          14             7
          14          15             7
          15          16             7
          16          17             7
          17          18             7
          18          19             7
          19          20             7
          20          21             7
          21          22             7
          22          23             7
          23          24             7
          24          25             7
          25          26             7
          26          27             7
          27          28             7
          28          29             7
          29          30             7
          30          31             7
          31          32             7
          32          33             7
          33          34             7
          34          35             7
          35          36             7
          36          37             7
          37          38             7
```

```
38          39              7
39          40              7
40          41              7
41          42              7
42          43              7
43          44              7
44          45              7
45          46              7
46          47              7
47          48              7
48          49              7
49          50              7
50          51              7
51          52              7
52          53              7
53          54              7
54          55              7
55          56              7
56          57              7
57          58              7
58          59              6,
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'CustomerId=%{x}<br>InvoiceCount=%{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},
               'template': '...',
               'title': {'text': 'Total Invoices per Customer'},
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceCount'}}}
}))
```

In [25]:

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

vn.ask(question=question)
```

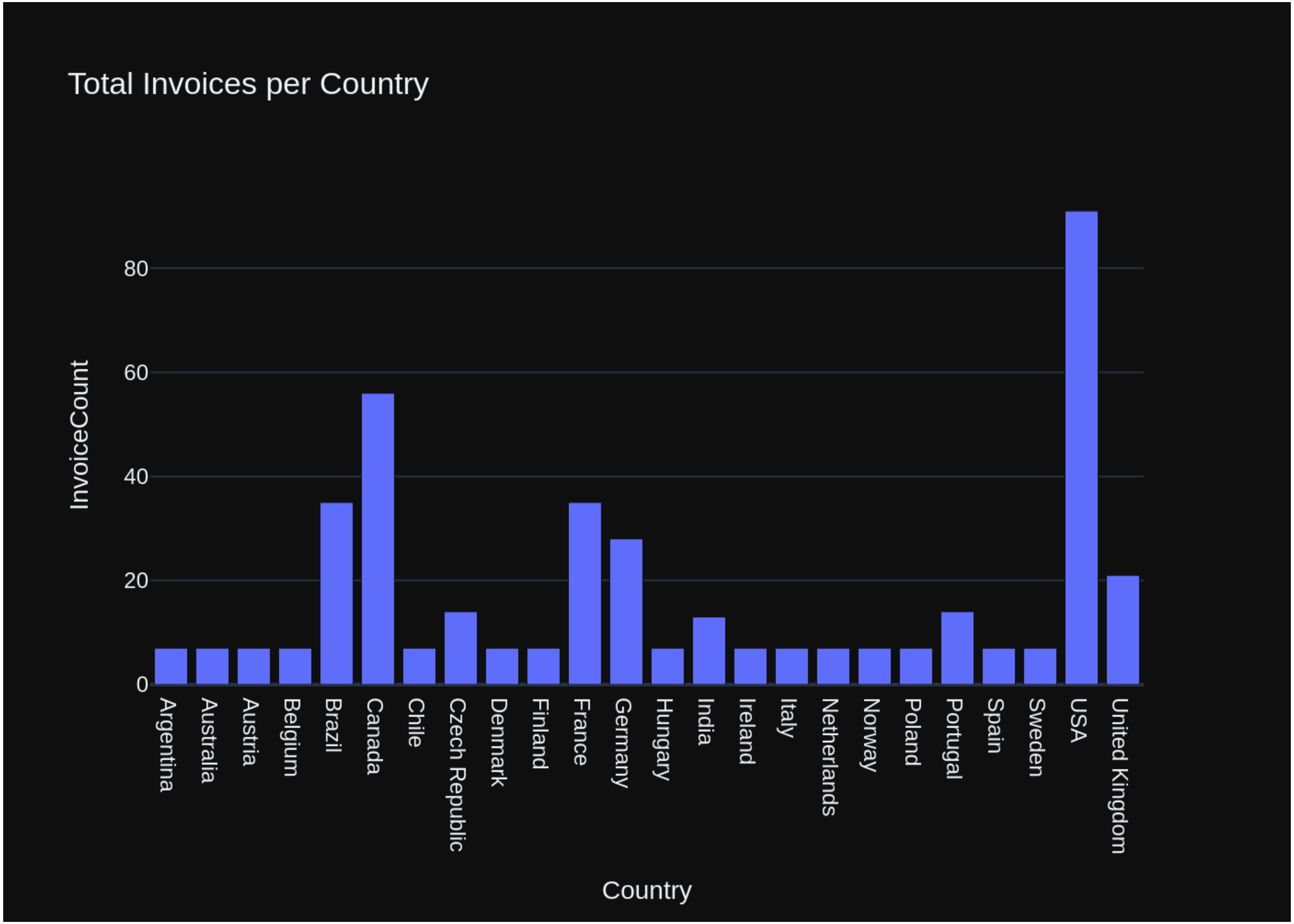
```
Number of requested results 10 is greater than number of elements in index 6, updating n_results = 6
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\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    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    Inv
oiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  N
OT 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) REFE
RENCES "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 I
FK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE "employees"\r\n(\r\n    EmployeeId INTEGER PRIMAR
Y 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 NV
ARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVAR
CHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (Reports
To) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "custom
ers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n
LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(4
0),\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    FOREIGN KEY (S
upportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABL
E "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n
ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTIO
N ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\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    UnitPric
e 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 N
O ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPD
ATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\n\n===Additional Context \n\nI
n the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, ple
ase generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost suf
ficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL quer
y 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 befor
e. \n'}, {'role': 'user', 'content': '  \n    Get the total number of invoices for each customer\n'}, {'role': 'assi
stant', 'content': 'SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId'}, {'role': 'use
r', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT C
```

```
ountry, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'rol
e': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM custo
mers'}, {'role': 'user', 'content': '  \n    List all albums and their corresponding artist names  \n'}, {'role': 'a
ssistant', 'content': 'SELECT \n    a.Title,\n    ar.Name\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistI
d'}, {'role': 'user', 'content': '  \n    Find all tracks with a name containing "What" (case-insensitive)\n'}, {'ro
le': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%' \n"}, {'role': 'user', 'content': 'Show
me a list of tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE
type='table'"}, {'role': 'user', 'content': '  \n    Find the total number of invoices per country:\n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: 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    Cus
tomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    Billing
City NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NV
ARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (Customer
Id) \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) REFERENCE
S \"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 P
RIMARY 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 DATETIME,\r\n
Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    Posta
lCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KE
Y (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE T
ABLE \"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    C
ity NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Pho
ne NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    F
OREIGN 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(16
0)  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 KE
Y AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  N
OT 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 pr
ovided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If t
he 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 say
ing intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Pl
ease use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answ
er exactly as it was given before. \n"}, {"role": "user", "content": "  \n    Get the total number of invoices for e
ach customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROU
P BY CustomerId"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role":
"assistant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY Custo
merCount DESC\nLIMIT 5"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "conten
t": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "  \n    List all albums and their corresponding
artist names  \n"}, {"role": "assistant", "content": "SELECT \n    a.Title,\n    ar.Name\nFROM albums a\nJOIN artist
s ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "  \n    Find all tracks with a name containing \"Wh
at\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks WHERE Name LIKE '%What%' \n"},
{"role": "user", "content": "Show me a list of tables in the SQLite database"}, {"role": "assistant", "content": "SE
LECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "content": "  \n    Find the total number of inv
oices per country:\n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:16:52.312489285Z', 'message': {'role': 'assistant', 'conten
t': 'SELECT \n    c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.Custom
erId = i.CustomerId\nGROUP BY c.Country; \n\n\n\n'}, 'done_reason': 'stop', 'done': True, 'total_duration': 12568624
0876, 'load_duration': 784909, 'prompt_eval_count': 1378, 'prompt_eval_duration': 112557749000, 'eval_count': 47, 'e
val_duration': 12728751000}
LLM Response: SELECT
    c.Country,
    COUNT(i.InvoiceId) AS InvoiceCount
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.Country;




Info: Output from LLM: SELECT
    c.Country,
    COUNT(i.InvoiceId) AS InvoiceCount
```

```
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.Country;




Extracted SQL: SELECT
    c.Country,
    COUNT(i.InvoiceId) AS InvoiceCount
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.Country
SELECT
    c.Country,
    COUNT(i.InvoiceId) AS InvoiceCount
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.Country
           Country  InvoiceCount
0        Argentina             7
1        Australia             7
2          Austria             7
3          Belgium             7
4           Brazil            35
5           Canada            56
6            Chile             7
7   Czech Republic            14
8          Denmark             7
9          Finland             7
10          France            35
11         Germany            28
12         Hungary             7
13           India            13
14         Ireland             7
15           Italy             7
16     Netherlands             7
17          Norway             7
18          Poland             7
19        Portugal            14
20           Spain             7
```

```
21          Sweden             7
22             USA            91
23  United Kingdom            21
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n    Find the total number of invoices per country:\n'\n\nThe DataFrame was pro
duced using this query: SELECT \n    c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOIN inv
oices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\n\nThe following is information about the resulting panda
s DataFrame 'df': \nRunning df.dtypes gives:\n Country         object\nInvoiceCount     int64\ndtype: object"}, {"ro
le": "user", "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. Respond wit
h only Python code. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:17:26.293464762Z', 'message': {'role': 'assistant', 'conten
t': "```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n  fig = px.indicator(\n    df,\n    value='Invo
iceCount',\n    title='Total Invoices per Country'\n  )\nelse:\n  fig = px.bar(df, x='Country', y='InvoiceCount', ti
tle='Total Invoices per Country')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 33955
515399, 'load_duration': 44568714, 'prompt_eval_count': 188, 'prompt_eval_duration': 12555621000, 'eval_count': 91,
'eval_duration': 21305296000}
```


Total Invoices per Country
InvoiceCount
0
20
40
60
80
Country
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

```
Out[25]: ('SELECT \n    c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.Custome
         rId = i.CustomerId\nGROUP BY c.Country',
                     Country  InvoiceCount
          0        Argentina             7
          1        Australia             7
          2          Austria             7
          3          Belgium             7
          4           Brazil            35
          5           Canada            56
          6            Chile             7
          7   Czech Republic            14
          8          Denmark             7
          9          Finland             7
          10          France            35
          11         Germany            28
          12         Hungary             7
          13           India            13
          14         Ireland             7
          15           Italy             7
          16     Netherlands             7
          17          Norway             7
          18          Poland             7
          19        Portugal            14
          20           Spain             7
          21          Sweden             7
          22             USA            91
          23  United Kingdom            21,
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'Country=%{x}<br>InvoiceCount=%{y}<extra></extra>',
                        'legendgroup': '',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array(['Argentina', 'Australia', 'Austria', 'Belgium', 'Brazil', 'Canada',
                                    'Chile', 'Czech Republic', 'Denmark', 'Finland', 'France', 'Germany',
                                    'Hungary', 'India', 'Ireland', 'Italy', 'Netherlands', 'Norway',
                                    'Poland', 'Portugal', 'Spain', 'Sweden', 'USA', 'United Kingdom'],
                                   dtype=object),
                        'xaxis': 'x',
                        'y': array([ 7,  7,  7,  7, 35, 56,  7, 14,  7,  7, 35, 28,  7, 13,  7,  7,  7,  7,
                                     7, 14,  7,  7, 91, 21]),
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                         'legend': {'tracegroupgap': 0},
                         'template': '...',
                         'title': {'text': 'Total Invoices per Country'},
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Country'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceCount'}}}
          }))
```

In [26]:

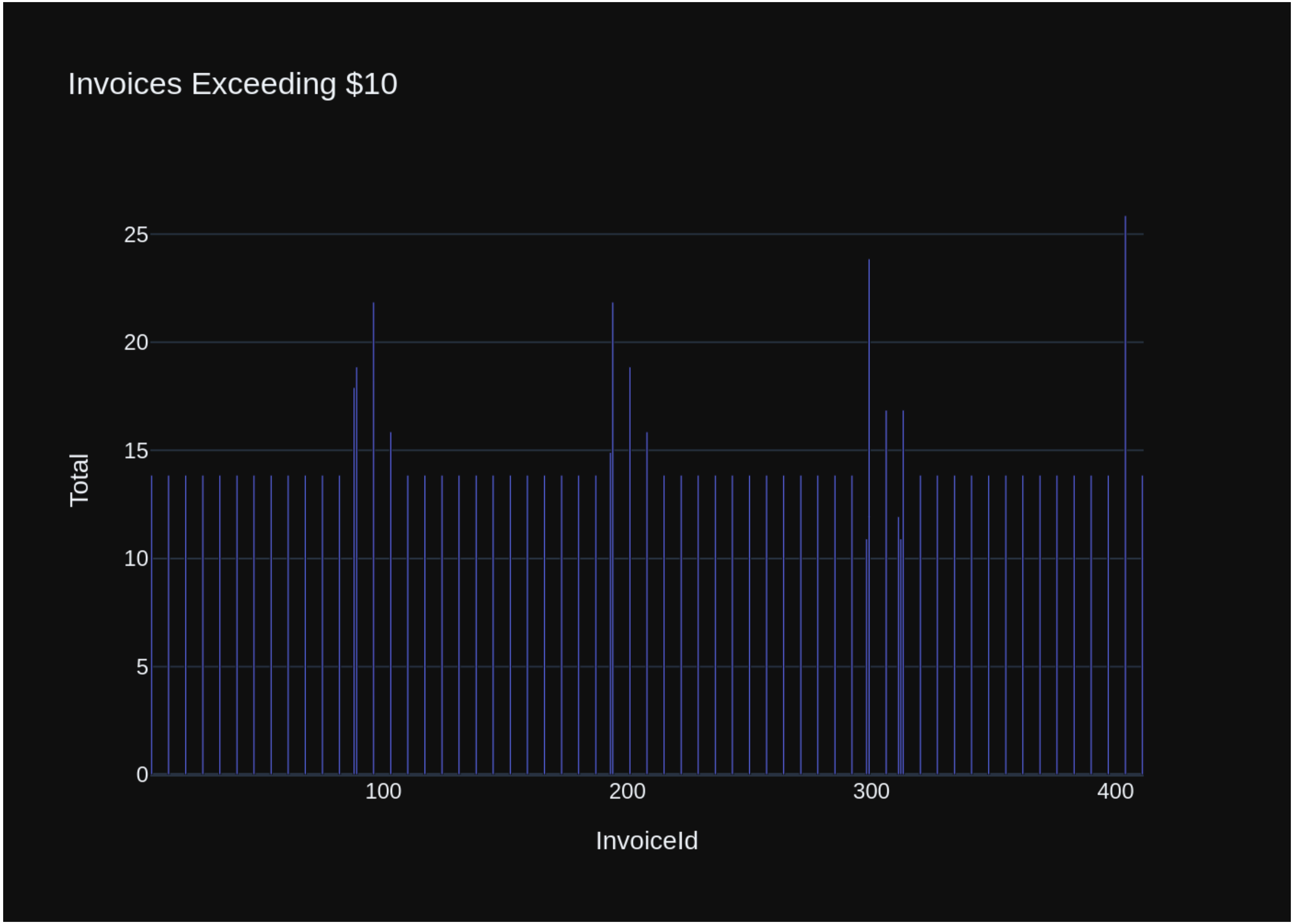
```
question = """
    List all invoices with a total exceeding $10:
"""

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
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \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 N
O 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 NU
LL,\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 N
UMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACT
ION 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 AUTOI
NCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NUL
L,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGE
R,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON D
ELETE 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 DELET
E NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE TABL
E "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 N
VARCHAR(24),\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\nCR
EATE TABLE "employees"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(2
0)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    B
irthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVA
RCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCH
AR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DEL
ETE 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 prov
ided context is almost sufficient but requires knowledge of a specific string in a particular column, please generat
e an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying int
ermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please u
se the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exa
ctly as it was given before. \n'}, {'role': 'user', 'content': '  \n    Get the total number of invoices for each cu
stomer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY C
ustomerId'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per country:\n'}, {'role': 'assi
stant', 'content': 'SELECT \n    c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOIN invoice
s i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': 'How many customers are ther
```

```
e'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': 'what are the t
op 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS CustomerCo
unt\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n
List all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT \n    a.Titl
e,\n    ar.Name\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 * FRO
M tracks WHERE Name LIKE '%What%' \n"}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite databas
e'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'conten
t': '  \n    List all invoices with a total exceeding $10:\n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question.
Your response should ONLY be based on the given context and follow the response guidelines and format instructions.
\n===Tables \nCREATE TABLE \"invoice_items\"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r
\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n
Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) 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 NUL
L,\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 N
UMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO A
CTION 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 KE
Y AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  N
OT 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 N
VARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR
(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(1
0),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INT
EGER,\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    ReportsT
o INTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(4
0),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(2
4),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (Emp
loyeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customer
s\" (SupportRepId)\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 q
uestion. \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 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, plea
se 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 InvoiceCount\nFR
OM invoices\nGROUP BY CustomerId"}, {"role": "user", "content": "  \n    Find the total number of invoices per count
ry:\n"}, {"role": "assistant", "content": "SELECT \n    c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM cu
stomers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"role": "user", "content": "How ma
ny customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user", "cont
ent": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, C
OUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user",
"content": "  \n    List all albums and their corresponding artist names  \n"}, {"role": "assistant", "content": "SE
LECT \n    a.Title,\n    ar.Name\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "co
ntent": "  \n    Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "cont
ent": "SELECT * FROM tracks WHERE Name LIKE '%What%' \n"}, {"role": "user", "content": "Show me a list of tables in
the SQLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"rol
e": "user", "content": "  \n    List all invoices with a total exceeding $10:\n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:19:24.97393451Z', 'message': {'role': 'assistant', 'conten
t': 'SELECT * FROM invoices WHERE Total > 10;'}, 'done_reason': 'stop', 'done': True, 'total_duration': 11857427183
6, 'load_duration': 624572, 'prompt_eval_count': 1394, 'prompt_eval_duration': 115201258000, 'eval_count': 12, 'eval
_duration': 2919006000}
LLM Response: SELECT * FROM invoices WHERE Total > 10;
Info: Output from LLM: SELECT * FROM invoices WHERE Total > 10;
Extracted SQL: SELECT * FROM invoices WHERE Total > 10
SELECT * FROM invoices WHERE Total > 10
    InvoiceId  CustomerId          InvoiceDate             BillingAddress  \
0           5          23  2009-01-11 00:00:00            69 Salem Street
1          12           2  2009-02-11 00:00:00    Theodor-Heuss-Straße 34
2          19          40  2009-03-14 00:00:00             8, Rue Hanovre
3          26          19  2009-04-14 00:00:00            1 Infinite Loop
4          33          57  2009-05-15 00:00:00            Calle Lira, 198
..        ...         ...                  ...                        ...
59        383          10  2013-08-12 00:00:00  Rua Dr. Falcão Filho, 155
60        390          48  2013-09-12 00:00:00      Lijnbaansgracht 120bg
```

```
61        397          27  2013-10-13 00:00:00            1033 N Park Ave
62        404           6  2013-11-13 00:00:00              Rilská 3174/6
63        411          44  2013-12-14 00:00:00            Porthaninkatu 9

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

[64 rows x 9 columns]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n    List all invoices with a total exceeding $10:\n'\n\nThe DataFrame was prod
uced using this query: SELECT * FROM invoices WHERE Total > 10\n\nThe following is information about the resulting p
andas DataFrame 'df': \nRunning df.dtypes gives:\n InvoiceId              int64\nCustomerId             int64\nInvoi
ceDate           object\nBillingAddress        object\nBillingCity           object\nBillingState          object\nB
illingCountry        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 i
s in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with onl
y Python code. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:19:56.419001795Z', 'message': {'role': 'assistant', 'conten
t': "```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n  fig = px.indicator(df, \n
value='Total',\n                    title='Invoice Total')\nelse:\n  fig = px.bar(df, x='InvoiceId', y='Total', titl
e='Invoices Exceeding $10')\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 31413669786, 'load_durati
on': 45246904, 'prompt_eval_count': 195, 'prompt_eval_duration': 12463302000, 'eval_count': 79, 'eval_duration': 188
14743000}
```


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

```
Out[26]: ('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': [{'alignmentgroup': 'True',
                        'hovertemplate': 'InvoiceId=%{x}<br>Total=%{y}<extra></extra>',
                        'legendgroup': '',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array([  5,  12,  19,  26,  33,  40,  47,  54,  61,  68,  75,  82,  88,  89,
                                96, 103, 110, 117, 124, 131, 138, 145, 152, 159, 166, 173, 180, 187,
                               193, 194, 201, 208, 215, 222, 229, 236, 243, 250, 257, 264, 271, 278,
                               285, 292, 298, 299, 306, 311, 312, 313, 320, 327, 334, 341, 348, 355,
                               362, 369, 376, 383, 390, 397, 404, 411]),
                   'xaxis': 'x',
                   'y': array([13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                               13.86, 13.86, 17.91, 18.86, 21.86, 15.86, 13.86, 13.86, 13.86, 13.86,
                               13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 14.91, 21.86,
                               18.86, 15.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                               13.86, 13.86, 13.86, 13.86, 10.91, 23.86, 16.86, 11.94, 10.91, 16.86,
                               13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                               13.86, 13.86, 25.86, 13.86]),
                   'yaxis': 'y'}],
         'layout': {'barmode': 'relative',
                    'legend': {'tracegroupgap': 0},
                    'template': '...',
                    'title': {'text': 'Invoices Exceeding $10'},
                    'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceId'}},
                    'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total'}}}
     }))
```

In [27]:

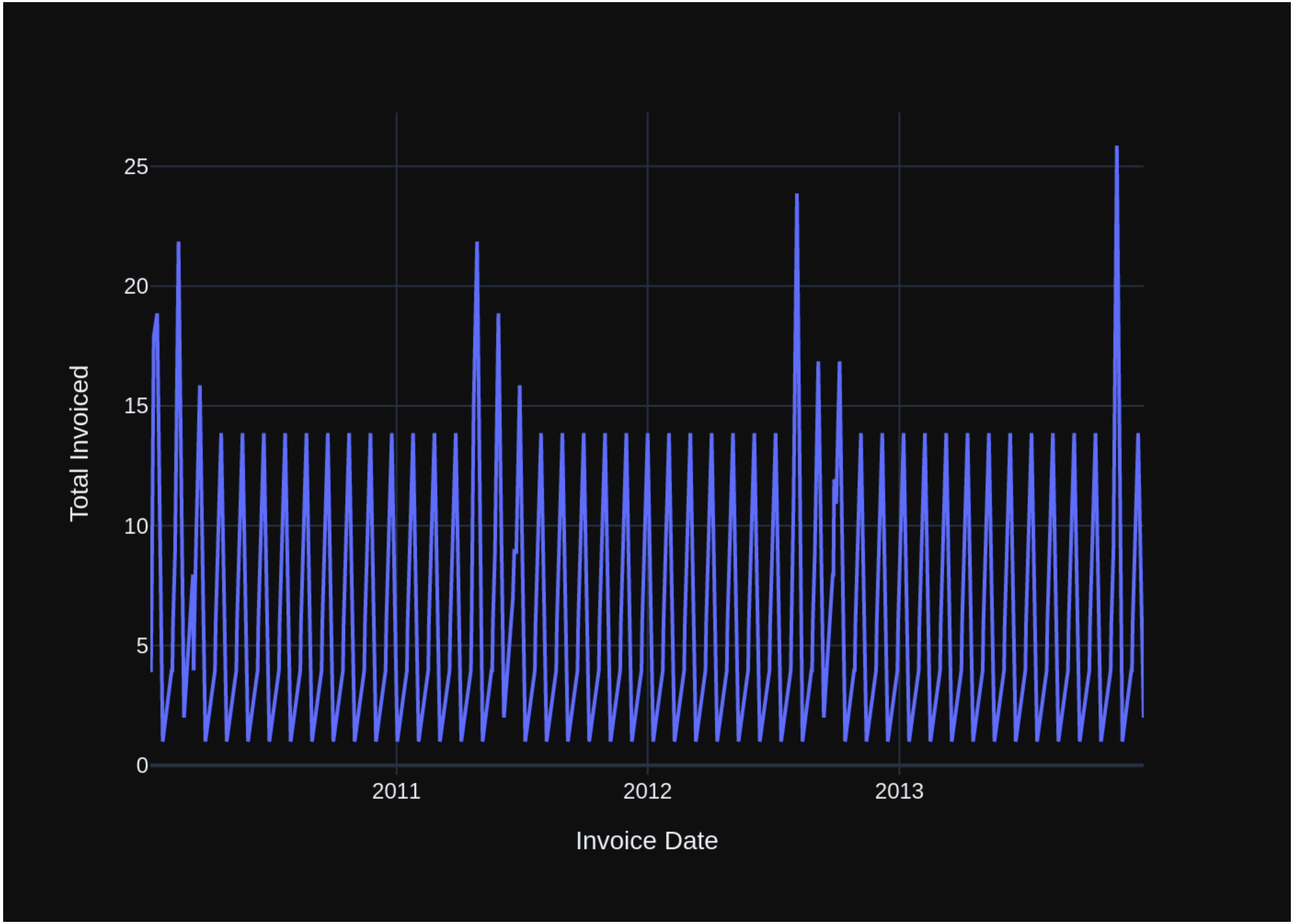
```
question = """
    Find all invoices since 2010 and the total amount invoiced:
"""

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
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\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    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    Inv
oiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  N
OT 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) REFE
RENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoic
eId 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 PRIMA
RY 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 NV
ARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVAR
CHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (Reports
To) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "custom
ers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n
LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(4
0),\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    FOREIGN KEY (S
upportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABL
E "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 (GenreI
d) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) R
EFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "album
s"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    Arti
stId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON
UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTE
GER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistI
d) 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, pleas
e generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost suffi
cient 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 pr
ovided 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 befor
```

```
e. \n'}, {'role': 'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistan
t', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'user', 'content': '  \n    Find the total numbe
r of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT \n    c.Country, \n    COUNT(i.InvoiceId) A
S InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'us
er', 'content': '  \n    Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'S
ELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId'}, {'role': 'user', 'content': 'How m
any customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'con
tent': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country,
COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'use
r', 'content': '  \n    List all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content':
'SELECT \n    a.Title,\n    ar.Name\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', 'con
tent': "SELECT * FROM tracks WHERE Name LIKE '%What%' \n"}, {'role': 'user', 'content': 'Show me a list of tables in
the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'rol
e': 'user', 'content': '  \n    Find all invoices since 2010 and the total amount invoiced:\n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: 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    Cus
tomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    Billing
City NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NV
ARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (Customer
Id) \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) REFERENCE
S \"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 O
N \"invoice_items\" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX I
FK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE TABLE \"employees\"\r\n(\r\n    EmployeeId INTEGER PR
IMARY 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    Addr
ess NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCod
e NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (R
eportsTo) 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 N
VARCHAR(24),\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\n
CREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  N
OT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARC
HAR(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    FOREIG
N 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\nCREAT
E TABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT N
ULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \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 NUL
L,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    F
OREIGN 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===A
dditional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided con
text is sufficient, 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 generate a
n intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying interm
ediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use t
he 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 $1
0:\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 \n    c.Country,
\n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY
c.Country"}, {"role": "user", "content": "  \n    Get the total number of invoices for each customer\n"}, {"role":
"assistant", "content": "SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId"}, {"role":
"user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customer
s"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "c
ontent": "SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\n
LIMIT 5"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist names  \n"}, {"role":
"assistant", "content": "SELECT \n    a.Title,\n    ar.Name\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.Artis
tId"}, {"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%' \n"}, {"role": "user", "content": "S
how me a list of tables in the SQLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master W
HERE type='table'"}, {"role": "user", "content": "  \n    Find all invoices since 2010 and the total amount invoice
d:\n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:22:24.08176123Z', 'message': {'role': 'assistant', 'conten
t': "SELECT SUM(Total) AS TotalInvoiced, InvoiceDate \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY I
nvoiceDate;\n\n\n\n"}, 'done_reason': 'stop', 'done': True, 'total_duration': 147568159226, 'load_duration': 579689,
'prompt_eval_count': 1577, 'prompt_eval_duration': 136253085000, 'eval_count': 42, 'eval_duration': 10822754000}
```

```
LLM Response: SELECT SUM(Total) AS TotalInvoiced, InvoiceDate
FROM invoices
WHERE InvoiceDate >= '2010-01-01'
GROUP BY InvoiceDate;




Info: Output from LLM: SELECT SUM(Total) AS TotalInvoiced, InvoiceDate
FROM invoices
WHERE InvoiceDate >= '2010-01-01'
GROUP BY InvoiceDate;




Extracted SQL: SELECT SUM(Total) AS TotalInvoiced, InvoiceDate
FROM invoices
WHERE InvoiceDate >= '2010-01-01'
GROUP BY InvoiceDate
SELECT SUM(Total) AS TotalInvoiced, InvoiceDate
FROM invoices
WHERE InvoiceDate >= '2010-01-01'
GROUP BY InvoiceDate
     TotalInvoiced          InvoiceDate
0             3.96  2010-01-08 00:00:00
1             3.96  2010-01-09 00:00:00
2             6.94  2010-01-10 00:00:00
3            17.91  2010-01-13 00:00:00
4            18.86  2010-01-18 00:00:00
..             ...                  ...
277           3.96  2013-12-05 00:00:00
278           5.94  2013-12-06 00:00:00
279           8.91  2013-12-09 00:00:00
280          13.86  2013-12-14 00:00:00
281           1.99  2013-12-22 00:00:00

[282 rows x 2 columns]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n    Find all invoices since 2010 and the total amount invoiced:\n'\n\nThe Data
Frame was produced using this query: SELECT SUM(Total) AS TotalInvoiced, InvoiceDate \nFROM invoices \nWHERE Invoice
Date >= '2010-01-01'\nGROUP BY InvoiceDate\n\nThe following is information about the resulting pandas DataFrame 'd
f': \nRunning df.dtypes gives:\n TotalInvoiced    float64\nInvoiceDate       object\ndtype: object"}, {"role": "use
r", "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 Pyt
hon code. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:23:07.110293755Z', 'message': {'role': 'assistant', 'conten
t': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n  fig = px.indicator(\n    df, \n    name="Tota
lInvoiced",\n    value="TotalInvoiced",\n    title="Total Invoiced Since 2010",\n    color_discrete_sequence=[\'#007
bff\']\n  )\nelse:\n  fig = px.line(df, x="InvoiceDate", y="TotalInvoiced")\nfig.update_layout(xaxis_title="Invoice
Date", yaxis_title="Total Invoiced") \n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 43000895267, 'l
oad_duration': 41126482, 'prompt_eval_count': 193, 'prompt_eval_duration': 12325903000, 'eval_count': 132, 'eval_dur
ation': 30580924000}
```


25
20
15
10
5
0
Total Invoiced
2011
2012
2013
Invoice Date

```
Out[27]: ("SELECT SUM(Total) AS TotalInvoiced, InvoiceDate \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY In
voiceDate",
      TotalInvoiced          InvoiceDate
 0             3.96  2010-01-08 00:00:00
 1             3.96  2010-01-09 00:00:00
 2             6.94  2010-01-10 00:00:00
 3            17.91  2010-01-13 00:00:00
 4            18.86  2010-01-18 00:00:00
 ..             ...                  ...
 277           3.96  2013-12-05 00:00:00
 278           5.94  2013-12-06 00:00:00
 279           8.91  2013-12-09 00:00:00
 280          13.86  2013-12-14 00:00:00
 281           1.99  2013-12-22 00:00:00

 [282 rows x 2 columns],
 Figure({
     'data': [{'hovertemplate': 'InvoiceDate=%{x}<br>TotalInvoiced=%{y}<extra></extra>',
               'legendgroup': '',
               'line': {'color': '#636efa', 'dash': 'solid'},
               'marker': {'symbol': 'circle'},
               'mode': 'lines',
               'name': '',
               'orientation': 'v',
               'showlegend': False,
               'type': 'scatter',
               'x': array(['2010-01-08 00:00:00', '2010-01-09 00:00:00', '2010-01-10 00:00:00',
                           ..., '2013-12-09 00:00:00', '2013-12-14 00:00:00',
                           '2013-12-22 00:00:00'], dtype=object),
               'xaxis': 'x',
               'y': array([ 3.96,  3.96,  6.94, ...,  8.91, 13.86,  1.99]),
               'yaxis': 'y'}],
     'layout': {'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Invoice Date'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total Invoiced'}}}
 }))
```

In [28]:

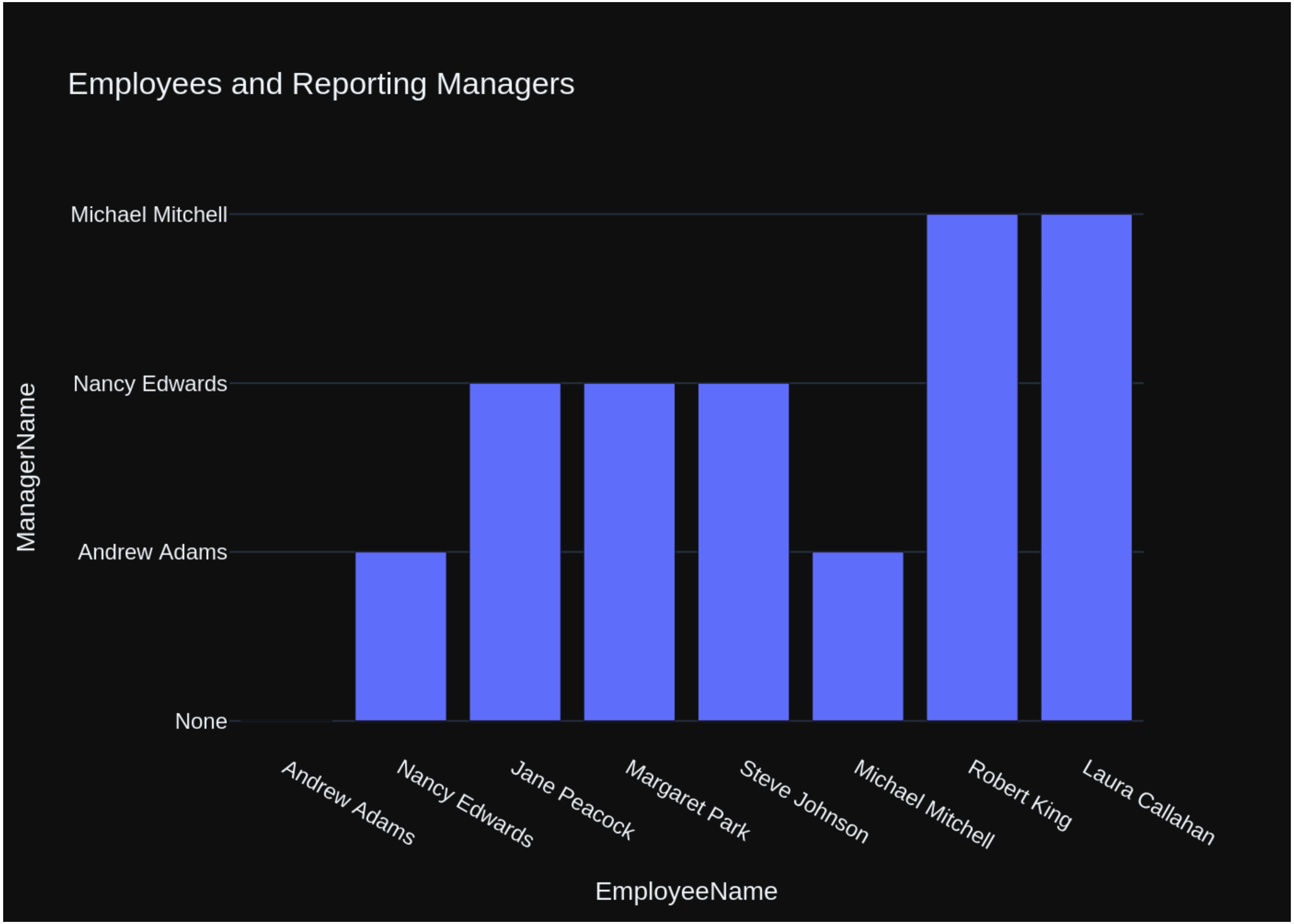
```
question = """
    List all employees and their reporting manager's name (if any):
"""

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
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE TABLE "employee
s"\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    Co
untry 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 UPDA
TE 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    Addres
s 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    Suppor
tRepId 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 "invoic
es"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    In
voiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingSta
te 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 UPD
ATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT N
ULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NUL
L,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DE
LETE 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 A
CTION 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 ACTIO
N ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) 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 pr
ovided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If t
he 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 say
ing intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. P
lease use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the ans
wer exactly as it was given before. \n'}, {'role': 'user', 'content': 'what are the top 5 countries that customers c
ome from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY C
ountry\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find all invoices since 2010 an
```

```
d the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT SUM(Total) AS TotalInvoiced, InvoiceDate 
\nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': '  \n    Ge
t the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*)
AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId'}, {'role': 'user', 'content': '  \n    List all albums and thei
r corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT \n    a.Title,\n    ar.Name\nFROM album
s a\nJOIN artists ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': '  \n    Find the total number of in
voices per country:\n'}, {'role': 'assistant', 'content': 'SELECT \n    c.Country, \n    COUNT(i.InvoiceId) AS Invoi
ceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'c
ontent': '  \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', 'con
tent': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': '  \n    Find all tracks with a name containin
g "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%' 
\n"}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant', 'conten
t': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': "  \n    List all employees an
d their reporting manager's name (if any):\n"}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: 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 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(6
0),\r\n    FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO AC
TION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Fi
rstName 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 NV
ARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportR
epId 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 \"inv
oices\"\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    BillingS
tate NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(1
0,2)  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\t
ON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DEL
ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"artists\"\r\n(\r\n    ArtistId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AU
TOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT N
ULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTE
GER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\t
ON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DEL
ETE 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 AU
TOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (A
rtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqli
te_stat1(tbl,idx,stat)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Gui
delines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for t
he question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a part
icular column, please generate an intermediate SQL query to find the distinct strings in that column. 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, p
lease repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "what are the top 5 countr
ies that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM c
ustomers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    Find all in
voices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT SUM(Total) AS TotalIn
voiced, InvoiceDate \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "c
ontent": "  \n    Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT C
ustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId"}, {"role": "user", "content": "  \n    List
all albums and their corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT \n    a.Title,\n    
ar.Name\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "  \n    Find the
total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT \n    c.Country, \n    COUNT(i.I
nvoiceId) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"},
{"role": "user", "content": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "conte
nt": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "content": "How many customers are there"}, {"rol
e": "assistant", "content": "SELECT COUNT(*) 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%' \n"}, {"role": "user", "content": "Show me a list of tables in the SQLite database"}, {"role": "a
ssistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "content": "  \n    Lis
t all employees and their reporting manager's name (if any):\n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:25:38.792876347Z', 'message': {'role': 'assistant', 'conten
t': "SELECT \n    e.FirstName || ' ' || e.LastName AS EmployeeName,\n    CASE \n        WHEN e.ReportsTo IS NULL THE
N 'None'\n        ELSE (SELECT FirstName || ' ' || LastName FROM employees WHERE EmployeeId = e.ReportsTo)\n    END
AS ManagerName\nFROM employees e;"}, 'done_reason': 'stop', 'done': True, 'total_duration': 151582529337, 'load_dura
tion': 591500, 'prompt_eval_count': 1569, 'prompt_eval_duration': 133045805000, 'eval_count': 69, 'eval_duration': 1
```

```
7955107000}
LLM Response: SELECT
    e.FirstName || ' ' || e.LastName AS EmployeeName,
    CASE
        WHEN e.ReportsTo IS NULL THEN 'None'
        ELSE (SELECT FirstName || ' ' || LastName FROM employees WHERE EmployeeId = e.ReportsTo)
    END AS ManagerName
FROM employees e;
Info: Output from LLM: SELECT
    e.FirstName || ' ' || e.LastName AS EmployeeName,
    CASE
        WHEN e.ReportsTo IS NULL THEN 'None'
        ELSE (SELECT FirstName || ' ' || LastName FROM employees WHERE EmployeeId = e.ReportsTo)
    END AS ManagerName
FROM employees e;
Extracted SQL: SELECT
    e.FirstName || ' ' || e.LastName AS EmployeeName,
    CASE
        WHEN e.ReportsTo IS NULL THEN 'None'
        ELSE (SELECT FirstName || ' ' || LastName FROM employees WHERE EmployeeId = e.ReportsTo)
    END AS ManagerName
FROM employees e
SELECT
    e.FirstName || ' ' || e.LastName AS EmployeeName,
    CASE
        WHEN e.ReportsTo IS NULL THEN 'None'
        ELSE (SELECT FirstName || ' ' || LastName FROM employees WHERE EmployeeId = e.ReportsTo)
    END AS ManagerName
FROM employees e
       EmployeeName       ManagerName
0      Andrew Adams              None
1     Nancy Edwards      Andrew Adams
2      Jane Peacock     Nancy Edwards
3     Margaret Park     Nancy Edwards
4     Steve Johnson     Nancy Edwards
5  Michael Mitchell      Andrew Adams
6       Robert King  Michael Mitchell
7    Laura Callahan  Michael Mitchell
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n    List all employees and their reporting manager's name (if any):\n'\n\nThe 
DataFrame was produced using this query: SELECT \n    e.FirstName || ' ' || e.LastName AS EmployeeName,\n    CASE \n
WHEN e.ReportsTo IS NULL THEN 'None'\n        ELSE (SELECT FirstName || ' ' || LastName FROM employees WHERE Employe
eId = e.ReportsTo)\n    END AS ManagerName\nFROM employees e\n\nThe following is information about the resulting pan
das DataFrame 'df': \nRunning df.dtypes gives:\n EmployeeName    object\nManagerName     object\ndtype: object"},
{"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume t
he data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond
with only Python code. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:26:15.045665297Z', 'message': {'role': 'assistant', 'conten
t': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n  fig = px.indicator(df, \n
name="EmployeeName",\n                    value="ManagerName",\n                    title="Employee and Reporting Ma
nager")\nelse:\n  fig = px.bar(df, x="EmployeeName", y="ManagerName",\n               title="Employees and Reporting
Managers")\n  \nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 36226575495, 'load_duratio
n': 664489, 'prompt_eval_count': 216, 'prompt_eval_duration': 13916545000, 'eval_count': 97, 'eval_duration': 221771
85000}
```


Employees and Reporting Managers
ManagerName
Michael Mitchell
Nancy Edwards
Andrew Adams
None
Andrew Adams
Nancy Edwards
Jane Peacock
Margaret Park
Steve Johnson
Michael Mitchell
Robert King
Laura Callahan
EmployeeName

```
Out[28]: ("SELECT \n    e.FirstName || ' ' || e.LastName AS EmployeeName,\n    CASE \n        WHEN e.ReportsTo IS NULL THEN
         'None'\n        ELSE (SELECT FirstName || ' ' || LastName FROM employees WHERE EmployeeId = e.ReportsTo)\n    END
         AS ManagerName\nFROM employees e",
                EmployeeName       ManagerName
          0      Andrew Adams              None
          1     Nancy Edwards      Andrew Adams
          2      Jane Peacock     Nancy Edwards
          3     Margaret Park     Nancy Edwards
          4     Steve Johnson     Nancy Edwards
          5  Michael Mitchell      Andrew Adams
          6       Robert King  Michael Mitchell
          7    Laura Callahan  Michael Mitchell,
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'EmployeeName=%{x}<br>ManagerName=%{y}<extra></extra>',
                        'legendgroup': '',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array(['Andrew Adams', 'Nancy Edwards', 'Jane Peacock', 'Margaret Park',
                                    'Steve Johnson', 'Michael Mitchell', 'Robert King', 'Laura Callahan'],
                                   dtype=object),
                        'xaxis': 'x',
                        'y': array(['None', 'Andrew Adams', 'Nancy Edwards', 'Nancy Edwards',
                                    'Nancy Edwards', 'Andrew Adams', 'Michael Mitchell', 'Michael Mitchell'],
                                   dtype=object),
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                         'legend': {'tracegroupgap': 0},
                         'template': '...',
                         'title': {'text': 'Employees and Reporting Managers'},
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'EmployeeName'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'ManagerName'}}}
          }))
```

In [29]:

```
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
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\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    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \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 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    FOREI
GN 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_Cust
omerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KE
Y AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Co
mpany 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 NVARC
HAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (Report
sTo)\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 INTEGE
R,\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    F
ax 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\n===Additional Context \n\nIn the chinook database invoice mean
s order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query with
out any explanations for the question. \n2. If the provided context is almost sufficient but requires 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 insufficient, pl
ease explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been a
sked 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 CustomerI
d, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId'}, {'role': 'user', 'content': '  \n    Find the tot
al number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT \n    c.Country, \n    COUNT(i.Invo
iceId) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'ro
le': 'user', 'content': '  \n    Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistan
t', 'content': "SELECT SUM(Total) AS TotalInvoiced, InvoiceDate \nFROM invoices \nWHERE InvoiceDate >= '2010-01-0
1'\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': 'How many c
ustomers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'conten
t': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COU
```

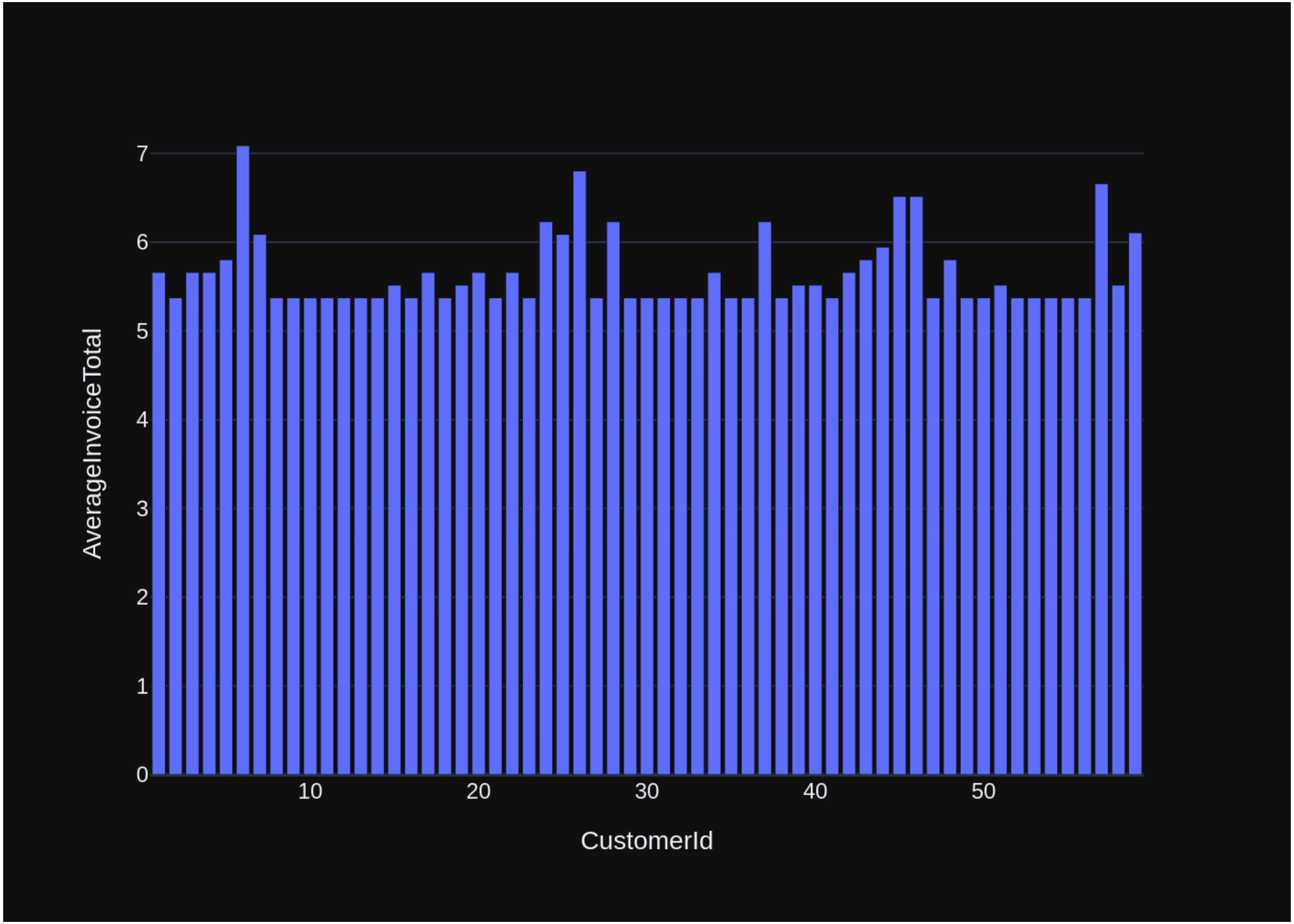
```
NT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user',
'content': "  \n    List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'cont
ent': "SELECT \n    e.FirstName || ' ' || e.LastName AS EmployeeName,\n    CASE \n        WHEN e.ReportsTo IS NULL T
HEN 'None'\n        ELSE (SELECT FirstName || ' ' || LastName FROM employees WHERE EmployeeId = e.ReportsTo)\n    EN
D AS ManagerName\nFROM employees e"}, {'role': 'user', 'content': '  \n    List all albums and their corresponding a
rtist names  \n'}, {'role': 'assistant', 'content': 'SELECT \n    a.Title,\n    ar.Name\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%' \n"}, {'rol
e': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT
name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': '  \n    Get the average invoice total for
each customer:\n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: 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    Cus
tomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    Billing
City NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NV
ARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (Customer
Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (Cust
omerId)\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    FOREI
GN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREI
GN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDE
X IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE INDEX I
FK_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 NUL
L,\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 \"empl
oyees\" (ReportsTo)\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(30),\r\n
ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVA
RCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVAR
CHAR(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 d
```

```
atabase 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 req
uires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the d
istinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided contex
t 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 InvoiceCount\nFROM invoices\nGROUP BY CustomerId"}, {"role": "user", "content": "
\n    Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT \n    c.Country,
\n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY
c.Country"}, {"role": "user", "content": "  \n    Find all invoices since 2010 and the total amount invoiced:\n"},
{"role": "assistant", "content": "SELECT SUM(Total) AS TotalInvoiced, InvoiceDate \nFROM invoices \nWHERE InvoiceDat
e >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "content": "  \n    List all invoices with a total excee
ding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "conten
t": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "u
ser", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT
Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"rol
e": "user", "content": "  \n    List all employees and their reporting manager's name (if any):\n"}, {"role": "assis
tant", "content": "SELECT \n    e.FirstName || ' ' || e.LastName AS EmployeeName,\n    CASE \n        WHEN e.Reports
To IS NULL THEN 'None'\n        ELSE (SELECT FirstName || ' ' || LastName FROM employees WHERE EmployeeId = e.Report
sTo)\n    END AS ManagerName\nFROM employees e"}, {"role": "user", "content": "  \n    List all albums and their cor
responding artist names  \n"}, {"role": "assistant", "content": "SELECT \n    a.Title,\n    ar.Name\nFROM albums a\n
JOIN artists ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "  \n    Find all tracks with a name cont
aining \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks WHERE Name LIKE '%Wha
t%' \n"}, {"role": "user", "content": "Show me a list of tables in the SQLite database"}, {"role": "assistant", "con
tent": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "content": "  \n    Get the average in
voice total for each customer:\n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:28:18.085892653Z', 'message': {'role': 'assistant', 'conten
t': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal\nFROM invoices\nGROUP BY CustomerId;'}, 'done_reason': 'st
op', 'done': True, 'total_duration': 122934554296, 'load_duration': 615970, 'prompt_eval_count': 1430, 'prompt_eval_
duration': 116853860000, 'eval_count': 22, 'eval_duration': 5408960000}
LLM Response: SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal
FROM invoices
GROUP BY CustomerId;
Info: Output from LLM: SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal
FROM invoices
GROUP BY CustomerId;
Extracted SQL: SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal
FROM invoices
GROUP BY CustomerId
```

```
SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal
FROM invoices
GROUP BY CustomerId
    CustomerId  AverageInvoiceTotal
0            1             5.660000
1            2             5.374286
2            3             5.660000
3            4             5.660000
4            5             5.802857
5            6             7.088571
6            7             6.088571
7            8             5.374286
8            9             5.374286
9           10             5.374286
10          11             5.374286
11          12             5.374286
12          13             5.374286
13          14             5.374286
14          15             5.517143
15          16             5.374286
16          17             5.660000
17          18             5.374286
18          19             5.517143
19          20             5.660000
20          21             5.374286
21          22             5.660000
22          23             5.374286
23          24             6.231429
24          25             6.088571
25          26             6.802857
26          27             5.374286
27          28             6.231429
28          29             5.374286
29          30             5.374286
30          31             5.374286
31          32             5.374286
32          33             5.374286
33          34             5.660000
34          35             5.374286
35          36             5.374286
36          37             6.231429
```

```
37          38             5.374286
38          39             5.517143
39          40             5.517143
40          41             5.374286
41          42             5.660000
42          43             5.802857
43          44             5.945714
44          45             6.517143
45          46             6.517143
46          47             5.374286
47          48             5.802857
48          49             5.374286
49          50             5.374286
50          51             5.517143
51          52             5.374286
52          53             5.374286
53          54             5.374286
54          55             5.374286
55          56             5.374286
56          57             6.660000
57          58             5.517143
58          59             6.106667
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n    Get the average invoice total for each customer:\n'\n\nThe DataFrame was p
roduced using this query: SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal\nFROM invoices\nGROUP BY CustomerId\n
\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId   
int64\nAverageInvoiceTotal    float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plo
tly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is on
ly one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations 
-- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:28:46.535959061Z', 'message': {'role': 'assistant', 'conten
t': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n    fig = px.indicator(\n        df,\n        n
ame="Average Invoice Total",\n        value="AverageInvoiceTotal",\n    )\nelse:\n    fig = px.bar(df, x="CustomerI
d", y="AverageInvoiceTotal")\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 28430267966, 'load_durat
```

```
ion': 705061, 'prompt_eval_count': 169, 'prompt_eval_duration': 10423534000, 'eval_count': 78, 'eval_duration': 1787
3034000}
```



```
Out[29]:  ('SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal\nFROM invoices\nGROUP BY CustomerId',
               CustomerId  AverageInvoiceTotal
           0            1             5.660000
           1            2             5.374286
           2            3             5.660000
           3            4             5.660000
           4            5             5.802857
           5            6             7.088571
           6            7             6.088571
           7            8             5.374286
           8            9             5.374286
           9           10             5.374286
           10          11             5.374286
           11          12             5.374286
           12          13             5.374286
           13          14             5.374286
           14          15             5.517143
           15          16             5.374286
           16          17             5.660000
           17          18             5.374286
           18          19             5.517143
           19          20             5.660000
           20          21             5.374286
           21          22             5.660000
           22          23             5.374286
           23          24             6.231429
           24          25             6.088571
           25          26             6.802857
           26          27             5.374286
           27          28             6.231429
           28          29             5.374286
           29          30             5.374286
           30          31             5.374286
           31          32             5.374286
           32          33             5.374286
           33          34             5.660000
           34          35             5.374286
           35          36             5.374286
           36          37             6.231429
           37          38             5.374286
```

```
38          39             5.517143
39          40             5.517143
40          41             5.374286
41          42             5.660000
42          43             5.802857
43          44             5.945714
44          45             6.517143
45          46             6.517143
46          47             5.374286
47          48             5.802857
48          49             5.374286
49          50             5.374286
50          51             5.517143
51          52             5.374286
52          53             5.374286
53          54             5.374286
54          55             5.374286
55          56             5.374286
56          57             6.660000
57          58             5.517143
58          59             6.106667,
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'CustomerId=%{x}<br>AverageInvoiceTotal=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'v',
              'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
              'x': array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
                          19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
                          37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
                          55, 56, 57, 58, 59]),
              'xaxis': 'x',
              'y': array([5.66      , 5.37428571, 5.66      , 5.66      , 5.80285714, 7.08857143,
                          6.08857143, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571,
                          5.37428571, 5.37428571, 5.51714286, 5.37428571, 5.66      , 5.37428571,
                          5.51714286, 5.66      , 5.37428571, 5.66      , 5.37428571, 6.23142857,
                          6.08857143, 6.80285714, 5.37428571, 6.23142857, 5.37428571, 5.37428571,
                          5.37428571, 5.37428571, 5.37428571, 5.66      , 5.37428571, 5.37428571,
                          6.23142857, 5.37428571, 5.51714286, 5.51714286, 5.37428571, 5.66      ,
                          5.80285714, 5.94571429, 6.51714286, 6.51714286, 5.37428571, 5.80285714,
                          5.37428571, 5.37428571, 5.51714286, 5.37428571, 5.37428571, 5.37428571,
                          5.37428571, 5.37428571, 6.66      , 5.51714286, 6.10666667]),
              'yaxis': 'y'}],
    'layout': {'barmode': 'relative',
               'legend': {'tracegroupgap': 0},
               'margin': {'t': 60},
               'template': '...',
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'AverageInvoiceTotal'}}}
}))
```

In [30]:

```python
question = """
    Find the top 5 most expensive tracks (based on unit price):
"""

vn.ask(question=question)
```

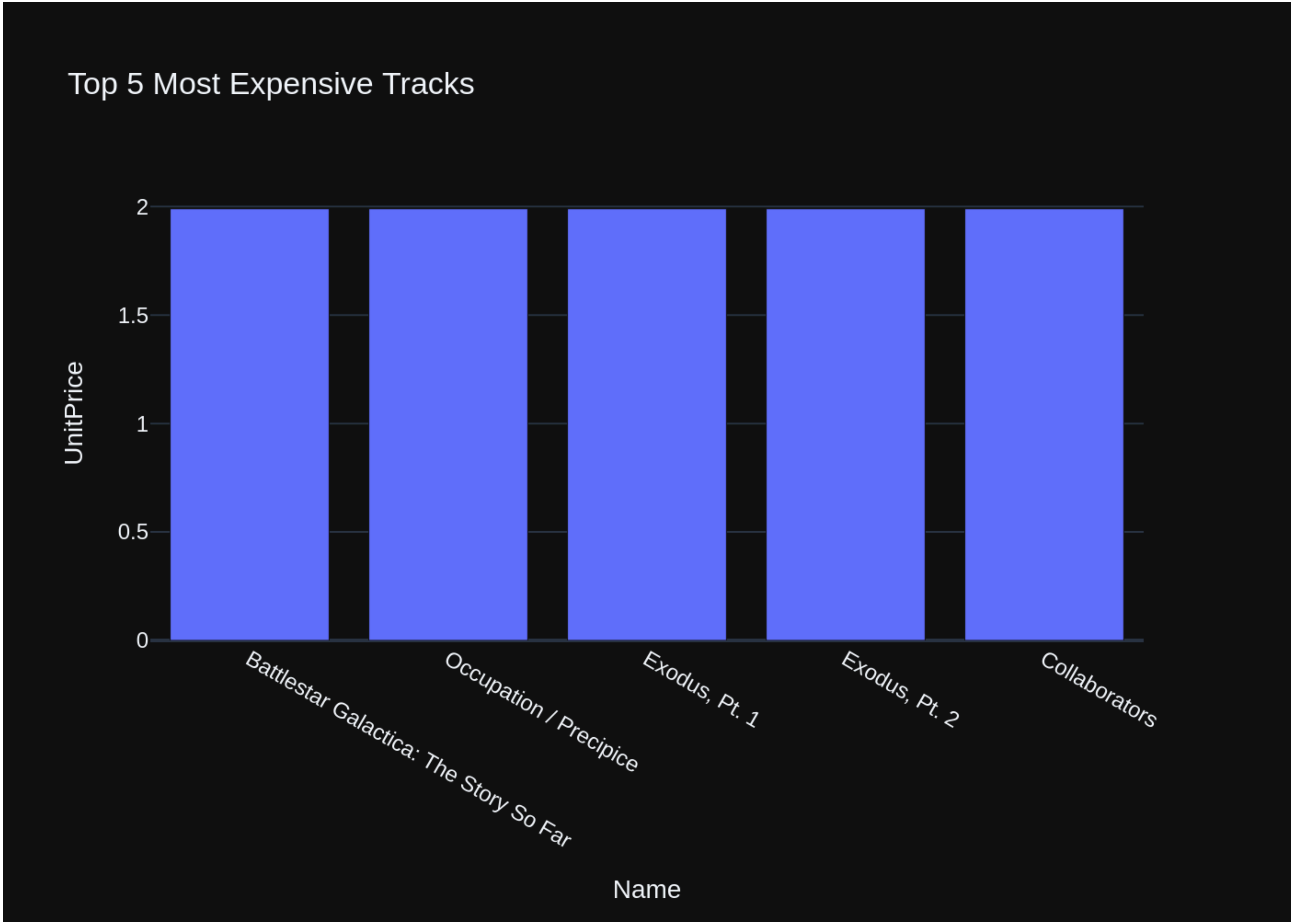
```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \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 INTEGE
R,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUME
RIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION\r\n)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (G
enreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE INDEX IFK_InvoiceLineTrackI
d 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 NUL
L,\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 T
ABLE "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" (Playlis
tId) \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\nCREAT
E TABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NUL
L,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELET
E 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 expl
anations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific st
ring in a particular column, please generate an intermediate SQL query to find the distinct strings in that column.
Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explai
n why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and an
swered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': '  \n    F
ind all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM
tracks WHERE Name LIKE '%What%' \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    List all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT \n    a.T
itle,\n    ar.Name\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 SUM(Total)
AS TotalInvoiced, InvoiceDate \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 CustomerId, AVG(Total) AS AverageInvoiceTotal\nFROM invoices\nGROUP BY CustomerId'}, {'role': 'user', 'conte
nt': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, CO
UNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user',
'content': '  \n    Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT \n
c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerI
```

```
d\nGROUP BY c.Country'}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'a
ssistant', '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
InvoiceCount\nFROM invoices\nGROUP BY CustomerId'}, {'role': 'user', 'content': 'How many customers are there'}, {'r
ole': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': '  \n    Find the top 5
most expensive tracks (based on unit price):\n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question.
Your response should ONLY be based on the given context and follow the response guidelines and format instructions.
\n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NV
ARCHAR(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 ACTI
ON,\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 AC
TION\r\n)\n\nCREATE INDEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (G
enreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\n\nCREATE INDEX IFK_InvoiceLineTrac
kId 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  NO
T NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NUL
L,\r\n    FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO
N,\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 \"playlis
ts\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks
\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums\" (A
rtistId)\n\nCREATE TABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVA
RCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES \"artists\" (Artis
tId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database inv
oice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL q
uery without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowl
edge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct str
ings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insuff
icient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question h
as 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", "c
ontent": "SELECT * FROM tracks WHERE Name LIKE '%What%' \n"}, {"role": "user", "content": "  \n    List all invoices
```

```
with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total > 10"}, {"rol
e": "user", "content": "  \n    List all albums and their corresponding artist names  \n"}, {"role": "assistant", "c
ontent": "SELECT \n    a.Title,\n    ar.Name\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 SUM(Total) AS TotalInvoiced, InvoiceDate \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGRO
UP BY InvoiceDate"}, {"role": "user", "content": "  \n    Get the average invoice total for each customer:\n"}, {"ro
le": "assistant", "content": "SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal\nFROM invoices\nGROUP BY Customer
Id"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant",
"content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC
\nLIMIT 5"}, {"role": "user", "content": "  \n    Find the total number of invoices per country:\n"}, {"role": "assi
stant", "content": "SELECT \n    c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOIN invoice
s i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"role": "user", "content": "Show me a list of tables in t
he SQLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"rol
e": "user", "content": "  \n    Get the total number of invoices for each customer\n"}, {"role": "assistant", "conte
nt": "SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId"}, {"role": "user", "content":
"How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "use
r", "content": "  \n    Find the top 5 most expensive tracks (based on unit price):\n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:30:32.261879724Z', 'message': {'role': 'assistant', 'conten
t': 'SELECT *\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5;'}, 'done_reason': 'stop', 'done': True, 'total_duratio
n': 105617973581, 'load_duration': 693346, 'prompt_eval_count': 1272, 'prompt_eval_duration': 100877606000, 'eval_co
unt': 17, 'eval_duration': 4061201000}
LLM Response: SELECT *
FROM tracks
ORDER BY UnitPrice DESC
LIMIT 5;
Info: Output from LLM: SELECT *
FROM tracks
ORDER BY UnitPrice DESC
LIMIT 5;
Extracted SQL: 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
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n    Find the top 5 most expensive tracks (based on unit price):\n'\n\nThe Data
Frame was produced using this query: SELECT *\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5\n\nThe following is inf
ormation about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n TrackId           int64\nName
object\nAlbumId           int64\nMediaTypeId       int64\nGenreId           int64\nComposer         object\nMillisec
onds      int64\nBytes             int64\nUnitPrice       float64\ndtype: object"}, {"role": "user", "content": "Can
you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe
called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not an
swer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:31:07.052409661Z', 'message': {'role': 'assistant', 'conten
t': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n  fig = px.indicator(\n    df,\n    name="UnitP
rice",\n    value="UnitPrice",\n    title="Most Expensive Track"\n  )\nelse:\n  fig = px.bar(df, x="Name", y="UnitPr
ice", title="Top 5 Most Expensive Tracks")\n\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duratio
n': 34764593904, 'load_duration': 632964, 'prompt_eval_count': 208, 'prompt_eval_duration': 13073608000, 'eval_coun
t': 94, 'eval_duration': 21559501000}
```


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

Out[30]:

```
('SELECT *\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 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=%{x}<br>UnitPrice=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Battlestar Galactica: The Story So Far', 'Occupation / Precipice',
                           'Exodus, Pt. 1', 'Exodus, Pt. 2', 'Collaborators'], dtype=object),
               'xaxis': 'x',
               'y': array([1.99, 1.99, 1.99, 1.99, 1.99]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Top 5 Most Expensive Tracks'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'UnitPrice'}}}
 }))
```

In [31]:

```
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
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \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 INTEGE
R,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUME
RIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE TABLE "genres"\r\n(\r\n    GenreId IN
TEGER 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_Trac
kMediaTypeId 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    FOREI
GN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FORE
IGN 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 ACTIO
N ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\n\n===Additional Context \n\nI
n the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, ple
ase generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost suf
ficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL quer
y 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 befor
e. \n'}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive tracks (based on unit price):\n'}, {'rol
e': 'assistant', 'content': 'SELECT *\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content':
'  \n    List all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT \n
a.Title,\n    ar.Name\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%' \n"}, {'role': 'user', 'content': 'what are the top 5 countries that customer
s come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP B
Y Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find the total number of inv
oices per country:\n'}, {'role': 'assistant', 'content': 'SELECT \n    c.Country, \n    COUNT(i.InvoiceId) AS Invoic
eCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'co
ntent': '  \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 SUM(Total) AS TotalInvoiced, InvoiceDate \nFROM invoices \nW
HERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': 'Show me a list of tables in t
he SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'rol
e': 'user', 'content': '  \n    Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'conte
```

```
nt': 'SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId'}, {'role': 'user', 'content':
'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'use
r', 'content': '  \n    List all genres and the number of tracks in each genre:\n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question.
Your response should ONLY be based on the given context and follow the response guidelines and format instructions.
\n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NV
ARCHAR(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 ACTI
ON,\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 AC
TION\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE TABLE \"genres\"\r\n(\r\n    GenreId IN
TEGER 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 ACTIO
N,\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(16
0)  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 prov
ided context is almost sufficient but requires knowledge of a specific string in a particular column, please generat
e an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying int
ermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please us
e the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exac
tly as it was given before. \n"}, {"role": "user", "content": "  \n    Find the top 5 most expensive tracks (based o
n unit price):\n"}, {"role": "assistant", "content": "SELECT *\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"r
ole": "user", "content": "  \n    List all albums and their corresponding artist names  \n"}, {"role": "assistant",
"content": "SELECT \n    a.Title,\n    ar.Name\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId"}, {"rol
e": "user", "content": "  \n    Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "as
sistant", "content": "SELECT * FROM tracks WHERE Name LIKE '%What%' \n"}, {"role": "user", "content": "what are the
top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS CustomerC
ount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "  \n
```

```
Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT \n    c.Country, \n
COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Cou
ntry"}, {"role": "user", "content": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "assistan
t", "content": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "content": "  \n    Find all invoices si
nce 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT SUM(Total) AS TotalInvoiced, I
nvoiceDate \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "content":
"Show me a list of tables in the SQLite database"}, {"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 InvoiceCount\nFROM invoices\nGROUP BY CustomerId"},
{"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM
customers"}, {"role": "user", "content": "  \n    List all genres and the number of tracks in each genre:\n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:32:51.900000234Z', 'message': {'role': 'assistant', 'conten
t': 'SELECT g.Name, COUNT(t.GenreId) AS TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.GenreId\nGROUP BY
g.Name;'}, 'done_reason': 'stop', 'done': True, 'total_duration': 104751984492, 'load_duration': 707064, 'prompt_eva
l_count': 1192, 'prompt_eval_duration': 93973746000, 'eval_count': 41, 'eval_duration': 10103227000}
LLM Response: SELECT g.Name, COUNT(t.GenreId) AS TrackCount
FROM genres g
JOIN tracks t ON g.GenreId = t.GenreId
GROUP BY g.Name;
Info: Output from LLM: SELECT g.Name, COUNT(t.GenreId) AS TrackCount
FROM genres g
JOIN tracks t ON g.GenreId = t.GenreId
GROUP BY g.Name;
Extracted SQL: SELECT g.Name, COUNT(t.GenreId) AS TrackCount
FROM genres g
JOIN tracks t ON g.GenreId = t.GenreId
GROUP BY g.Name
SELECT g.Name, COUNT(t.GenreId) AS TrackCount
FROM genres g
JOIN tracks t ON g.GenreId = t.GenreId
GROUP BY g.Name
                  Name  TrackCount
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
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n    List all genres and the number of tracks in each genre:\n'\n\nThe DataFram
e was produced using this query: SELECT g.Name, COUNT(t.GenreId) AS TrackCount\nFROM genres g\nJOIN tracks t ON g.Ge
nreId = t.GenreId\nGROUP BY g.Name\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunn
ing df.dtypes gives:\n Name          object\nTrackCount     int64\ndtype: object"}, {"role": "user", "content": "Can
you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe
called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not an
swer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:33:24.079974374Z', 'message': {'role': 'assistant', 'conten
t': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n  px.indicator(\n    df,\n    value="TrackCoun
t",\n    title=df["Name"].iloc[0],\n    color=df["Name"].iloc[0]\n  )\nelse:\n  px.bar(df, x="Name", y="TrackCount")
\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 32160011466, 'load_duration': 635653, 'prompt_eval_c
ount': 187, 'prompt_eval_duration': 11759930000, 'eval_count': 89, 'eval_duration': 20268914000}
Couldn't run plotly code:  'NoneType' object has no attribute 'show'
```

```
Traceback (most recent call last):
  File "/home/gongai/anaconda3/envs/vanna/lib/python3.11/site-packages/vanna/base/base.py", line 1684, in ask
    img_bytes = fig.to_image(format="png", scale=2)
                ^^^^^^^^^^^^
AttributeError: 'NoneType' object has no attribute 'to_image'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/home/gongai/anaconda3/envs/vanna/lib/python3.11/site-packages/vanna/base/base.py", line 1687, in ask
    fig.show()
    ^^^^^^^^
AttributeError: 'NoneType' object has no attribute 'show'
```

In [32]:

```
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
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \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  N
OT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "al
bums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (G
enreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" 
(MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "play
list_track" (TrackId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE INDEX IFK_TrackAlbumI
d 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    Playlis
tId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistI
d, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDA
TE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO AC
TION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "playlists"\r\n(\r\n    PlaylistI
d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\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, please gen
erate 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 fin
d the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided
context is insufficient, please explain why it can\'t be generated. \n4. Please use the most 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': 'assistan
t', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS TrackCount\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-insens
itive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%' \n"}, {'role': 'user',
'content': '  \n    Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'conten
t': 'SELECT *\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    List all albums
and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT \n    a.Title,\n    ar.Name\nFR
OM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': 'Show me a list of tables in
the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'rol
e': '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 th
e total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT SUM(Total) AS TotalInvoiced, InvoiceDate \nFR
OM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': "  \n    List a
ll employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': "SELECT \n    e.Firs
tName || ' ' || e.LastName AS EmployeeName,\n    CASE \n        WHEN e.ReportsTo IS NULL THEN 'None'\n        ELSE 
(SELECT FirstName || ' ' || LastName FROM employees WHERE EmployeeId = e.ReportsTo)\n    END AS ManagerName\nFROM em
```

```
ployees e"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assist
ant', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCoun
t DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SE
LECT COUNT(*) FROM customers'}, {'role': 'user', 'content': '  \n    Get all genres that do not have any tracks asso
ciated with them:\n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: 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 TABLE \"tracks\"\r\n(\r\n    TrackId I
NTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    Media
TypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NU
LL,\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\" (Ge
nreId) \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 \"pla
ylist_track\" (TrackId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE INDEX IFK_TrackAl
bumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"genres\"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT N
ULL,\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) REF
ERENCES \"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 PR
IMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON D
ELETE 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 \"play
lists\"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n\n===Ad
ditional 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, please generate an 
intermediate 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. 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 gen
re:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS TrackCount\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 cont
aining \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks WHERE Name LIKE '%Wha
t%' \n"}, {"role": "user", "content": "  \n    Find the top 5 most expensive tracks (based on unit price):\n"}, {"ro
le": "assistant", "content": "SELECT *\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content":
```

```
"  \n    List all albums and their corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT \n
a.Title,\n    ar.Name\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "Sh
ow me a list of tables in the SQLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WH
ERE 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 all in
voices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT SUM(Total) AS TotalIn
voiced, InvoiceDate \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "c
ontent": "  \n    List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "conten
t": "SELECT \n    e.FirstName || ' ' || e.LastName AS EmployeeName,\n    CASE \n        WHEN e.ReportsTo IS NULL THE
N 'None'\n        ELSE (SELECT FirstName || ' ' || LastName FROM employees WHERE EmployeeId = e.ReportsTo)\n    END
AS ManagerName\nFROM employees e"}, {"role": "user", "content": "what are the top 5 countries that customers come fr
om?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country
\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "How many customers are there"}, {"role": "ass
istant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "  \n    Get all genres that do n
ot have any tracks associated with them:\n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:35:13.372438784Z', 'message': {'role': 'assistant', 'conten
t': 'SELECT g.Name \nFROM genres g\nLEFT JOIN tracks t ON g.GenreId = t.GenreId\nWHERE t.TrackId IS NULL\nGROUP BY
g.Name;\n\n\n\n'}, 'done_reason': 'stop', 'done': True, 'total_duration': 109238726928, 'load_duration': 719141, 'pr
ompt_eval_count': 1247, 'prompt_eval_duration': 98493854000, 'eval_count': 41, 'eval_duration': 10082708000}
LLM Response: SELECT g.Name
FROM genres g
LEFT JOIN tracks t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
GROUP BY g.Name;




Info: Output from LLM: SELECT g.Name
FROM genres g
LEFT JOIN tracks t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
GROUP BY g.Name;




Extracted SQL: SELECT g.Name
FROM genres g
LEFT JOIN tracks t ON g.GenreId = t.GenreId
```

```
WHERE t.TrackId IS NULL
GROUP BY g.Name
SELECT g.Name
FROM genres g
LEFT JOIN tracks t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
GROUP BY g.Name
Empty DataFrame
Columns: [Name]
Index: []
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n    Get all genres that do not have any tracks associated with them:\n'\n\nThe
DataFrame was produced using this query: SELECT g.Name \nFROM genres g\nLEFT JOIN tracks t ON g.GenreId = t.GenreId
\nWHERE t.TrackId IS NULL\nGROUP BY g.Name\n\nThe following is information about the resulting pandas DataFrame 'd
f': \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 t
here is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any exp
lanations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:35:58.896198336Z', 'message': {'role': 'assistant', 'conten
t': '```python\nimport plotly.graph_objects as go\n\nfig = go.Figure(data=[go.Indicator(\n    value=len(df),\n    mo
de="gauge+number",\n    title={"text": "Genres with No Tracks"},\n    gauge={"axis": {"range": [0, 100], "tickmode":
"linear"},\n           "bar":{"color":"red"},\n           "shape":"circle"}\n)])\n\nfig.update_layout(\n    height=3
00,\n    width=300,\n    margin=dict(l=50, r=50, t=50, b=50)\n)\n```'}, 'done_reason': 'stop', 'done': True, 'total_
duration': 45521529811, 'load_duration': 977401, 'prompt_eval_count': 180, 'prompt_eval_duration': 11116667000, 'eva
l_count': 148, 'eval_duration': 34271809000}
```

Out[32]:
```
('SELECT g.Name \nFROM genres g\nLEFT JOIN tracks t ON g.GenreId = t.GenreId\nWHERE t.TrackId IS NULL\nGROUP BY g.
Name',
 Empty DataFrame
 Columns: [Name]
 Index: [],
 Figure({
     'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
               'hovertemplate': 'Name=%{label}<extra></extra>',
               'labels': array([], dtype=object),
               'legendgroup': '',
               'name': '',
               'showlegend': True,
               'type': 'pie'}],
     'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'template': '...'}
 }))
```

In [33]:
```
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
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\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    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n    Custome
rId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(2
0)  NOT NULL,\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 (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 I
NTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (I
nvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Tra
ckId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "employee
s"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT 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    Co
untry 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 UPDA
TE 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) REF
ERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFER
ENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n    A
lbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NO
T NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO
N\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 INT
EGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Millisecon
ds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) R
EFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCE
S "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "m
edia_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 fo
r the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a p
articular 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 befor
```

```
e, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': 'what are the top 5 co
untries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFR
OM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many custom
ers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': '
\n    Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, C
OUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId'}, {'role': 'user', 'content': '  \n    Find the total n
umber of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT \n    c.Country, \n    COUNT(i.InvoiceI
d) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role':
'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SEL
ECT * FROM invoices WHERE Total > 10'}, {'role': 'user', 'content': '  \n    Get the average invoice total for each
customer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal\nFROM invoices
\nGROUP BY CustomerId'}, {'role': 'user', 'content': "  \n    List all employees and their reporting manager's name
(if any):\n"}, {'role': 'assistant', 'content': "SELECT \n    e.FirstName || ' ' || e.LastName AS EmployeeName,\n
CASE \n        WHEN e.ReportsTo IS NULL THEN 'None'\n        ELSE (SELECT FirstName || ' ' || LastName FROM employee
s WHERE EmployeeId = e.ReportsTo)\n    END AS ManagerName\nFROM employees e"}, {'role': 'user', 'content': '  \n
Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT SUM(Total)
AS TotalInvoiced, InvoiceDate \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role':
'user', 'content': '  \n    List all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'conte
nt': 'SELECT \n    a.Title,\n    ar.Name\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId'}, {'role': 'us
er', 'content': '  \n    Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'con
tent': 'SELECT *\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    List all cus
tomers who have not placed any orders:\n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: 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    Cus
tomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    Billing
City NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NV
ARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (Customer
Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n    CustomerId INTEGE
R PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NUL
L,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State 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    I
nvoiceLineId 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 (InvoiceI
d) 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    Fir
stName 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 NVARC
HAR(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) REF
ERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REF
ERENCES \"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 INTE
GER  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 (Genre
Id) REFERENCES \"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 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 withou
t any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a s
pecific string in a particular column, please generate an intermediate SQL query to find the distinct strings in tha
t column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, ple
ase explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been ask
ed and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content":
"what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT
(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "co
ntent": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"rol
e": "user", "content": "  \n    Get the total number of invoices for each customer\n"}, {"role": "assistant", "conte
nt": "SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId"}, {"role": "user", "content":
"  \n    Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT \n    c.Countr
y, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP
BY c.Country"}, {"role": "user", "content": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "as
sistant", "content": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "content": "  \n    Get the averag
e invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT CustomerId, AVG(Total) AS AverageIn
voiceTotal\nFROM invoices\nGROUP BY CustomerId"}, {"role": "user", "content": "  \n    List all employees and their
reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT \n    e.FirstName || ' ' || e.LastN
ame AS EmployeeName,\n    CASE \n        WHEN e.ReportsTo IS NULL THEN 'None'\n        ELSE (SELECT FirstName || ' '
|| LastName FROM employees WHERE EmployeeId = e.ReportsTo)\n    END AS ManagerName\nFROM employees e"}, {"role": "us
er", "content": "  \n    Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "con
tent": "SELECT SUM(Total) AS TotalInvoiced, InvoiceDate \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP B
Y InvoiceDate"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist names  \n"},
{"role": "assistant", "content": "SELECT \n    a.Title,\n    ar.Name\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 *\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "conte
nt": "  \n    List all customers who have not placed any orders:\n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:38:37.291973081Z', 'message': {'role': 'assistant', 'conten
t': 'SELECT * FROM customers\nWHERE CustomerId NOT IN (SELECT CustomerId FROM invoices);\n\n\n\n'}, 'done_reason':
'stop', 'done': True, 'total_duration': 158311753253, 'load_duration': 657122, 'prompt_eval_count': 1764, 'prompt_ev
al_duration': 152843013000, 'eval_count': 19, 'eval_duration': 4797257000}
LLM Response: SELECT * FROM customers
WHERE CustomerId NOT IN (SELECT CustomerId FROM invoices);




Info: Output from LLM: SELECT * FROM customers
WHERE CustomerId NOT IN (SELECT CustomerId FROM invoices);




Extracted SQL: SELECT * FROM customers
WHERE CustomerId NOT IN (SELECT CustomerId FROM invoices)
SELECT * FROM customers
WHERE CustomerId NOT IN (SELECT CustomerId FROM invoices)
Empty DataFrame
Columns: [CustomerId, FirstName, LastName, Company, Address, City, State, Country, PostalCode, Phone, Fax, Email, Su
pportRepId]
Index: []
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n    List all customers who have not placed any orders:\n'\n\nThe DataFrame was
produced using this query: SELECT * FROM customers\nWHERE CustomerId NOT IN (SELECT CustomerId FROM invoices)\n\nThe
```

```
following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId      obj
ect\nFirstName       object\nLastName        object\nCompany         object\nAddress         object\nCity
object\nState           object\nCountry         object\nPostalCode      object\nPhone           object\nFax
object\nEmail           object\nSupportRepId    object\ndtype: object"}, {"role": "user", "content": "Can you genera
te the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'd
f'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with
any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:39:08.891792088Z', 'message': {'role': 'assistant', 'conten
t': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n    fig = px.indicator(df, \n
value="CustomerId", \n                       title="Customers with No Orders")\nelse:\n    fig = px.bar(df, x="Custo
merId", \n                 title="Customers with No Orders")\n```'}, 'done_reason': 'stop', 'done': True, 'total_dur
ation': 31596656957, 'load_duration': 706983, 'prompt_eval_count': 207, 'prompt_eval_duration': 13441553000, 'eval_c
ount': 78, 'eval_duration': 18013517000}
```

Customers with No Orders
4
3
2
1
0
−1
count
−1
0
1
2
3
4
5
6
CustomerId

```
Out[33]: ('SELECT * FROM customers\nWHERE CustomerId NOT IN (SELECT CustomerId FROM invoices)',
          Empty DataFrame
          Columns: [CustomerId, FirstName, LastName, Company, Address, City, State, Country, PostalCode, Phone, Fax, Email,
         SupportRepId]
          Index: [],
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'CustomerId=%{x}<br>count=%{y}<extra></extra>',
                        'legendgroup': '',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array([], dtype=object),
                        'xaxis': 'x',
                        'y': array([], dtype=int64),
                        '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': 'CustomerId'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'count'}}}
          }))
```

```
In [34]: question = """
             Get the top 10 most popular artists (based on the number of tracks):
         """

         vn.ask(question=question)
```

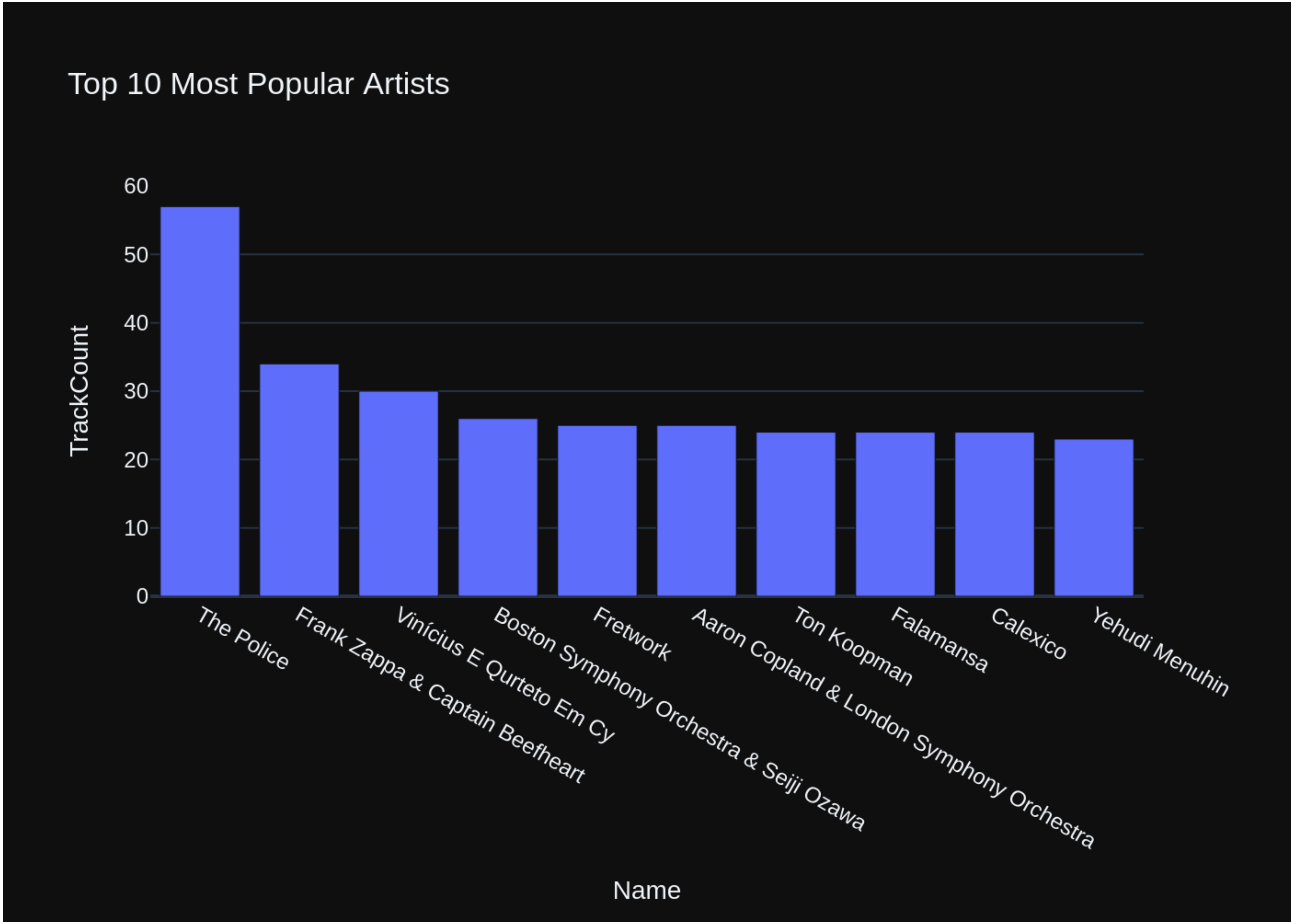
```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \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 INTEGE
R,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUME
RIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks"
(AlbumId)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGE
R 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\nCREAT
E TABLE "artists"\r\n(\r\n    ArtistId 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_TrackMediaTypeId ON "track
s" (MediaTypeId)\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 IN
TEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (Playlist
Id) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackI
d) 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 q
uery to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If t
he provided 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 be
fore. \n'}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive tracks (based on unit price):\n'},
{'role': 'assistant', 'content': 'SELECT *\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'conte
nt': '  \n    List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT
g.Name, COUNT(t.GenreId) AS TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.GenreId\nGROUP BY g.Name'},
{'role': 'user', 'content': '  \n    List all albums and their corresponding artist names  \n'}, {'role': 'assistan
t', 'content': 'SELECT \n    a.Title,\n    ar.Name\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': 'assistant', 'conten
t': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT
5'}, {'role': 'user', 'content': '  \n    Find all tracks with a name containing "What" (case-insensitive)\n'}, {'ro
le': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%' \n"}, {'role': 'user', 'content': '  \n
List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE To
tal > 10'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT CO
UNT(*) FROM customers'}, {'role': 'user', 'content': '  \n    Get the average invoice total for each customer:\n'},
{'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal\nFROM invoices\nGROUP BY Cust
omerId'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per country:\n'}, {'role': 'assista
nt', 'content': 'SELECT \n    c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOIN invoices i
```

```
ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': 'Show me a list of tables in the S
QLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'u
ser', 'content': '  \n    Get the top 10 most popular artists (based on the number of tracks):\n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question.
Your response should ONLY be based on the given context and follow the response guidelines and format instructions.
\n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NV
ARCHAR(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 ACTI
ON,\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 AC
TION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE INDEX IFK_TrackAlbumId ON \"tracks\"
(AlbumId)\n\nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE TABLE \"albums\"\r\n(\r\n    AlbumId IN
TEGER 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(1
20)\r\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\n\nCREATE INDEX IFK_TrackMediaType
Id ON \"tracks\" (MediaTypeId)\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 \"playlist_track\"\r\n(\r\n    PlaylistId INTEGER  NOT NU
LL,\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===A
dditional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided con
text is sufficient, 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 generate a
n intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying interm
ediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use t
he 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 un
it price):\n"}, {"role": "assistant", "content": "SELECT *\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"rol
e": "user", "content": "  \n    List all genres and the number of tracks in each genre:\n"}, {"role": "assistant",
"content": "SELECT g.Name, COUNT(t.GenreId) AS TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.GenreId\nGR
OUP BY g.Name"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist names  \n"},
{"role": "assistant", "content": "SELECT \n    a.Title,\n    ar.Name\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": "assi
stant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCo
```

```
unt DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    Find all tracks with a name containing \"What\" (case-inse
nsitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks WHERE Name LIKE '%What%' \n"}, {"role": "user",
"content": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FR
OM invoices WHERE Total > 10"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant",
"content": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "  \n    Get the average invoice total for
each customer:\n"}, {"role": "assistant", "content": "SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal\nFROM inv
oices\nGROUP BY CustomerId"}, {"role": "user", "content": "  \n    Find the total number of invoices per countr
y:\n"}, {"role": "assistant", "content": "SELECT \n    c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM cus
tomers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"role": "user", "content": "Show me
a list of tables in the SQLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE ty
pe='table'"}, {"role": "user", "content": "  \n    Get the top 10 most popular artists (based on the number of track
s):\n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:40:58.169559503Z', 'message': {'role': 'assistant', 'conten
t': 'SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artists a\nJOIN tracks t ON a.ArtistId = t.AlbumId\nGROUP B
Y a.Name\nORDER BY TrackCount DESC\nLIMIT 10;'}, 'done_reason': 'stop', 'done': True, 'total_duration': 10914771398
5, 'load_duration': 680044, 'prompt_eval_count': 1196, 'prompt_eval_duration': 95681036000, 'eval_count': 52, 'eval_
duration': 12804772000}
LLM Response: SELECT a.Name, COUNT(t.TrackId) AS TrackCount
FROM artists a
JOIN tracks t ON a.ArtistId = t.AlbumId
GROUP BY a.Name
ORDER BY TrackCount DESC
LIMIT 10;
Info: Output from LLM: SELECT a.Name, COUNT(t.TrackId) AS TrackCount
FROM artists a
JOIN tracks t ON a.ArtistId = t.AlbumId
GROUP BY a.Name
ORDER BY TrackCount DESC
LIMIT 10;
Extracted SQL: SELECT a.Name, COUNT(t.TrackId) AS TrackCount
FROM artists a
JOIN tracks t ON a.ArtistId = t.AlbumId
GROUP BY a.Name
ORDER BY TrackCount DESC
LIMIT 10
SELECT a.Name, COUNT(t.TrackId) AS TrackCount
FROM artists a
JOIN tracks t ON a.ArtistId = t.AlbumId
GROUP BY a.Name
ORDER BY TrackCount DESC
```

```
LIMIT 10
                                        Name  TrackCount
0                                 The Police          57
1            Frank Zappa & Captain Beefheart          34
2                   Vinícius E Qurteto Em Cy          30
3    Boston Symphony Orchestra & Seiji Ozawa          26
4                                   Fretwork          25
5  Aaron Copland & London Symphony Orchestra          25
6                                Ton Koopman          24
7                                  Falamansa          24
8                                   Calexico          24
9                             Yehudi Menuhin          23
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n    Get the top 10 most popular artists (based on the number of tracks):\n'\n
\nThe DataFrame was produced using this query: SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artists a\nJOIN t
racks t ON a.ArtistId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMIT 10\n\nThe following is informati
on about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name          object\nTrackCount     int6
4\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."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:41:32.980971177Z', 'message': {'role': 'assistant', 'conten
t': "```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n  fig = px.indicator(df, \n
value='TrackCount', \n                    title='Top Artist by Track Count')\nelse:\n  fig = px.bar(df, x='Name', y
='TrackCount', title='Top 10 Most Popular Artists')\n  \nfig.show() \n```"}, 'done_reason': 'stop', 'done': True, 't
otal_duration': 34785271268, 'load_duration': 678448, 'prompt_eval_count': 203, 'prompt_eval_duration': 12897854000,
'eval_count': 93, 'eval_duration': 21751710000}
```


Top 10 Most Popular Artists
TrackCount
0
10
20
30
40
50
60
The Police
Frank Zappa & Captain Beefheart
Vinícius E Qurteto Em Cy
Boston Symphony Orchestra & Seiji Ozawa
Fretwork
Aaron Copland & London Symphony Orchestra
Ton Koopman
Falamansa
Calexico
Yehudi Menuhin
Name

```
Out[34]:  ('SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artists a\nJOIN tracks t ON a.ArtistId = t.AlbumId\nGROUP BY
          a.Name\nORDER BY TrackCount DESC\nLIMIT 10',
                                                   Name  TrackCount
           0                               The Police          57
           1          Frank Zappa & Captain Beefheart          34
           2                 Vinícius E Qurteto Em Cy          30
           3    Boston Symphony Orchestra & Seiji Ozawa          26
           4                                 Fretwork          25
           5  Aaron Copland & London Symphony Orchestra          25
           6                              Ton Koopman          24
           7                                Falamansa          24
           8                                 Calexico          24
           9                           Yehudi Menuhin          23,
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'Name=%{x}<br>TrackCount=%{y}<extra></extra>',
                         'legendgroup': '',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                         'name': '',
                         'offsetgroup': '',
                         'orientation': 'v',
                         'showlegend': False,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array(['The Police', 'Frank Zappa & Captain Beefheart',
                                     'Vinícius E Qurteto Em Cy', 'Boston Symphony Orchestra & Seiji Ozawa',
                                     'Fretwork', 'Aaron Copland & London Symphony Orchestra', 'Ton Koopman',
                                     'Falamansa', 'Calexico', 'Yehudi Menuhin'], dtype=object),
                         'xaxis': 'x',
                         'y': array([57, 34, 30, 26, 25, 25, 24, 24, 24, 23]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'template': '...',
                          'title': {'text': 'Top 10 Most Popular Artists'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TrackCount'}}}
           }))
```

In [35]:

```
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
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "cust
omers"\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 NVARCHA
R(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR
(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  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 TA
BLE "invoices"\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(40),\r\n 
BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total N
UMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACT
ION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE TABLE "emplo
yees"\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    Co
untry 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 UPDA
TE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NU
LL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NUL
L,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DE
LETE 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  (Pla
ylistId, TrackId),\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 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  N
OT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI
ON\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. \n
2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column,
please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comm
ent 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 t
he answer exactly as it was given before. \n'}, {'role': 'user', 'content': 'what are the top 5 countries that custo
mers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROU
P BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find the total number of
invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT \n    c.Country, \n    COUNT(i.InvoiceId) AS Inv
oiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user',
'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'ro
```

```
le': 'user', 'content': '  \n    Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'cont
ent': 'SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId'}, {'role': 'user', 'conten
t': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invo
ices WHERE Total > 10'}, {'role': 'user', 'content': '  \n    Get the average invoice total for each customer:\n'},
{'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal\nFROM invoices\nGROUP BY Cust
omerId'}, {'role': 'user', 'content': "  \n    List all employees and their reporting manager's name (if any):\n"},
{'role': 'assistant', 'content': "SELECT \n    e.FirstName || ' ' || e.LastName AS EmployeeName,\n    CASE \n        
WHEN e.ReportsTo IS NULL THEN 'None'\n        ELSE (SELECT FirstName || ' ' || LastName FROM employees WHERE Employe
eId = e.ReportsTo)\n    END AS ManagerName\nFROM employees e"}, {'role': 'user', 'content': '  \n    Find all invoic
es since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT SUM(Total) AS TotalInvoic
ed, InvoiceDate \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'conte
nt': '  \n    Get the top 10 most popular artists (based on the number of tracks):\n'}, {'role': 'assistant', 'conte
nt': 'SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artists a\nJOIN tracks t ON a.ArtistId = t.AlbumId\nGROUP
BY a.Name\nORDER BY TrackCount DESC\nLIMIT 10'}, {'role': 'user', 'content': '  \n    List all albums and their corr
esponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT \n    a.Title,\n    ar.Name\nFROM albums a\nJ
OIN artists ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': '  \n     List all customers from Canada a
nd their email addresses:\n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: 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    La
stName 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 (Suppor
tRepId) 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 NUL
L,\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 N
UMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO A
CTION 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 N
ULL,\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(4
0),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(2
4),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELET
E 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 NUMER
IC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (Invoi
ceId) \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 \"pla
ylist_track\"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_Playl
istTrack 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 withou
t any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a s
pecific string in a particular column, please generate an intermediate SQL query to find the distinct strings in tha
t column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, ple
ase explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been ask
ed and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content":
"what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT
(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "co
ntent": "  \n    Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT \n
c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerI
d\nGROUP BY c.Country"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "conten
t": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "  \n    Get the total number of invoices for eac
h customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP
BY CustomerId"}, {"role": "user", "content": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "a
ssistant", "content": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "content": "  \n    Get the avera
ge invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT CustomerId, AVG(Total) AS AverageI
nvoiceTotal\nFROM invoices\nGROUP BY CustomerId"}, {"role": "user", "content": "  \n    List all employees and their
reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT \n    e.FirstName || ' ' || e.LastN
ame AS EmployeeName,\n    CASE \n        WHEN e.ReportsTo IS NULL THEN 'None'\n        ELSE (SELECT FirstName || ' '
|| LastName FROM employees WHERE EmployeeId = e.ReportsTo)\n    END AS ManagerName\nFROM employees e"}, {"role": "us
er", "content": "  \n    Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "con
tent": "SELECT SUM(Total) AS TotalInvoiced, InvoiceDate \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP B
Y InvoiceDate"}, {"role": "user", "content": "  \n    Get the top 10 most popular artists (based on the number of tr
acks):\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artists a\nJOIN tr
acks t ON a.ArtistId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMIT 10"}, {"role": "user", "content":
"  \n    List all albums and their corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT \n
a.Title,\n    ar.Name\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "
\n     List all customers from Canada and their email addresses:\n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:43:57.488493807Z', 'message': {'role': 'assistant', 'conten
```

```
t': "SELECT FirstName, LastName, Email FROM customers WHERE Country = 'Canada';"}, 'done_reason': 'stop', 'done': Tr
ue, 'total_duration': 144407465263, 'load_duration': 691979, 'prompt_eval_count': 1617, 'prompt_eval_duration': 1401
30316000, 'eval_count': 15, 'eval_duration': 3603095000}
LLM Response: SELECT FirstName, LastName, Email FROM customers WHERE Country = 'Canada';
Info: Output from LLM: SELECT FirstName, LastName, Email FROM customers WHERE Country = 'Canada';
Extracted SQL: 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
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n     List all customers from Canada and their email addresses:\n'\n\nThe DataF
rame was produced using this query: SELECT FirstName, LastName, Email FROM customers WHERE Country = 'Canada'\n\nThe
following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n FirstName    object
\nLastName     object\nEmail        object\ndtype: object"}, {"role": "user", "content": "Can you generate the Pytho
n 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 explanat
ions -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:44:26.707432776Z', 'message': {'role': 'assistant', 'conten
t': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n  px.indicator(\n    value=df[\'Email\'].iloc
[0],\n    title="Customer Email from Canada",\n    color="blue"\n  )\nelse:\n  px.bar(df, x=\'FirstName\', y=\'LastN
ame\', color=\'Email\') \n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 29197705000, 'load_duratio
n': 631895, 'prompt_eval_count': 161, 'prompt_eval_duration': 10015855000, 'eval_count': 84, 'eval_duration': 190482
08000}
Couldn't run plotly code:  'NoneType' object has no attribute 'show'
```

```
Traceback (most recent call last):
  File "/home/gongai/anaconda3/envs/vanna/lib/python3.11/site-packages/vanna/base/base.py", line 1684, in ask
    img_bytes = fig.to_image(format="png", scale=2)
                ^^^^^^^^^^^^
AttributeError: 'NoneType' object has no attribute 'to_image'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/home/gongai/anaconda3/envs/vanna/lib/python3.11/site-packages/vanna/base/base.py", line 1687, in ask
    fig.show()
    ^^^^^^^^
AttributeError: 'NoneType' object has no attribute 'show'
```

In [36]:

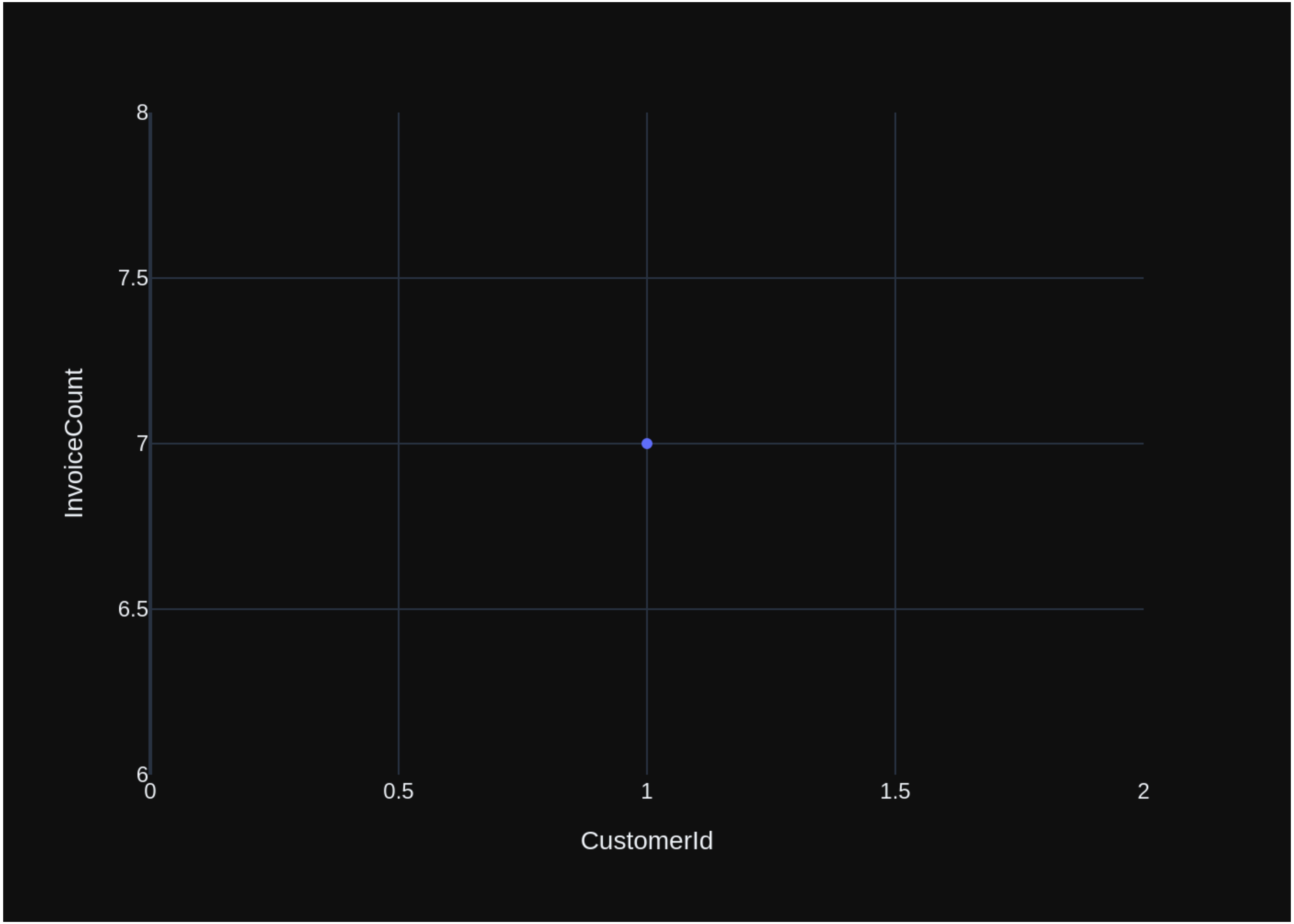
```
question = """
     Find the customer with the most invoices
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\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    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \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 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    FOREI
GN 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    Count
ry NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NV
ARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (Employe
eId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (S
upportRepId)\n\nCREATE TABLE "employees"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    L
astName 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(4
0),\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),\r\n    FOREIGN KEY (ReportsTo) REFERENCES "employees" (Emplo
yeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (Re
portsTo)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCH
AR(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(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    FO
REIGN 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 pr
ovided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If t
he 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 say
ing intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. P
lease use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the ans
wer 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 InvoiceCount\nFROM invoices\nGRO
UP BY CustomerId'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per country:\n'}, {'rol
e': 'assistant', 'content': 'SELECT \n    c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOI
N invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': '  \n    List all inv
```

```
oices 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': 'assi
stant', 'content': "SELECT SUM(Total) AS TotalInvoiced, InvoiceDate \nFROM invoices \nWHERE InvoiceDate >= '2010-01-
01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': '  \n    Get the average invoice total for each custome
r:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal\nFROM invoices\nGROUP
BY CustomerId'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'as
sistant', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY Customer
Count DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive tracks (based on unit pric
e):\n'}, {'role': 'assistant', 'content': 'SELECT *\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'use
r', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'},
{'role': 'user', 'content': '  \n     List all customers from Canada and their email addresses:\n'}, {'role': 'assis
tant', 'content': "SELECT FirstName, LastName, Email FROM customers WHERE Country = 'Canada'"}, {'role': 'user', 'co
ntent': '  \n    Get the top 10 most popular artists (based on the number of tracks):\n'}, {'role': 'assistant', 'co
ntent': 'SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artists a\nJOIN tracks t ON a.ArtistId = t.AlbumId\nGRO
UP BY a.Name\nORDER BY TrackCount DESC\nLIMIT 10'}, {'role': 'user', 'content': '  \n     Find the customer with the
most invoices \n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: 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    Cus
tomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    Billing
City NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NV
ARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (Customer
Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (Cust
omerId)\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    FOREI
GN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREI
GN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDE
X 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 NUL
L,\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 \"c
ustomers\" (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    Ci
ty NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phon
e NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES \"empl
oyees\" (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 NUL
L,\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    UnitPric
e 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 UPD
ATE 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 G
uidelines \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 par
ticular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the q
uery with 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 answered befor
e, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "  \n    Get the total
number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) AS InvoiceCo
unt\nFROM invoices\nGROUP BY CustomerId"}, {"role": "user", "content": "  \n    Find the total number of invoices pe
r country:\n"}, {"role": "assistant", "content": "SELECT \n    c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\n
FROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"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 invoice
d:\n"}, {"role": "assistant", "content": "SELECT SUM(Total) AS TotalInvoiced, InvoiceDate \nFROM invoices \nWHERE In
voiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "content": "  \n    Get the average invoice tota
l for each customer:\n"}, {"role": "assistant", "content": "SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal\nFR
OM invoices\nGROUP BY CustomerId"}, {"role": "user", "content": "what are the top 5 countries that customers come fr
om?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country
\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    Find the top 5 most expensive tracks
(based on unit price):\n"}, {"role": "assistant", "content": "SELECT *\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT
5"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*)
FROM customers"}, {"role": "user", "content": "  \n     List all customers from Canada and their email addresse
s:\n"}, {"role": "assistant", "content": "SELECT FirstName, LastName, Email FROM customers WHERE Country = 'Canad
a'"}, {"role": "user", "content": "  \n    Get the top 10 most popular artists (based on the number of tracks):\n"},
{"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artists a\nJOIN tracks t ON a.
ArtistId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMIT 10"}, {"role": "user", "content": "  \n     F
ind the customer with the most invoices \n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:46:48.521081641Z', 'message': {'role': 'assistant', 'conten
t': 'SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId\nORDER BY InvoiceCount DESC\nLI
MIT 1; \n\n\n\n'}, 'done_reason': 'stop', 'done': True, 'total_duration': 141751018652, 'load_duration': 770421, 'pr
```

```
ompt_eval_count': 1580, 'prompt_eval_duration': 133102103000, 'eval_count': 31, 'eval_duration': 7979131000}
LLM Response: SELECT CustomerId, COUNT(*) AS InvoiceCount
FROM invoices
GROUP BY CustomerId
ORDER BY InvoiceCount DESC
LIMIT 1;




Info: Output from LLM: SELECT CustomerId, COUNT(*) AS InvoiceCount
FROM invoices
GROUP BY CustomerId
ORDER BY InvoiceCount DESC
LIMIT 1;




Extracted SQL: SELECT CustomerId, COUNT(*) AS InvoiceCount
FROM invoices
GROUP BY CustomerId
ORDER BY InvoiceCount DESC
LIMIT 1
SELECT CustomerId, COUNT(*) AS InvoiceCount
FROM invoices
GROUP BY CustomerId
ORDER BY InvoiceCount DESC
LIMIT 1
   CustomerId  InvoiceCount
0           1             7
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n     Find the customer with the most invoices \n'\n\nThe DataFrame was produce
d using this query: SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId\nORDER BY Invoic
eCount DESC\nLIMIT 1\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes
gives:\n CustomerId      int64\nInvoiceCount    int64\ndtype: object"}, {"role": "user", "content": "Can you generat
e the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'd
f'. 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."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:47:19.522986409Z', 'message': {'role': 'assistant', 'conten
t': "```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n  fig = px.indicator(\n    df, \n    value='Inv
oiceCount',\n    title='Customer with the Most Invoices'\n  )\nelse:\n  fig = px.bar(df, x='CustomerId', y='InvoiceC
ount', title='Customer Invoice Count')\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 30974898256,
'load_duration': 673470, 'prompt_eval_count': 174, 'prompt_eval_duration': 11092268000, 'eval_count': 86, 'eval_dura
tion': 19751164000}
```


8
7.5
7
6.5
6
InvoiceCount
0
0.5
1
1.5
2
CustomerId

Out[36]:

```
('SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId\nORDER BY InvoiceCount DESC\nLIM
IT 1',
    CustomerId  InvoiceCount
 0           1             7,
 Figure({
     'data': [{'hovertemplate': 'CustomerId=%{x}<br>InvoiceCount=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'symbol': 'circle'},
               'mode': 'markers',
               'name': '',
               'orientation': 'v',
               'showlegend': False,
               'type': 'scatter',
               'x': array([1]),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'}],
     'layout': {'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceCount'}}}
 }))
```

In [ ]:

Advanced SQL questions

In [37]:

```
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
```

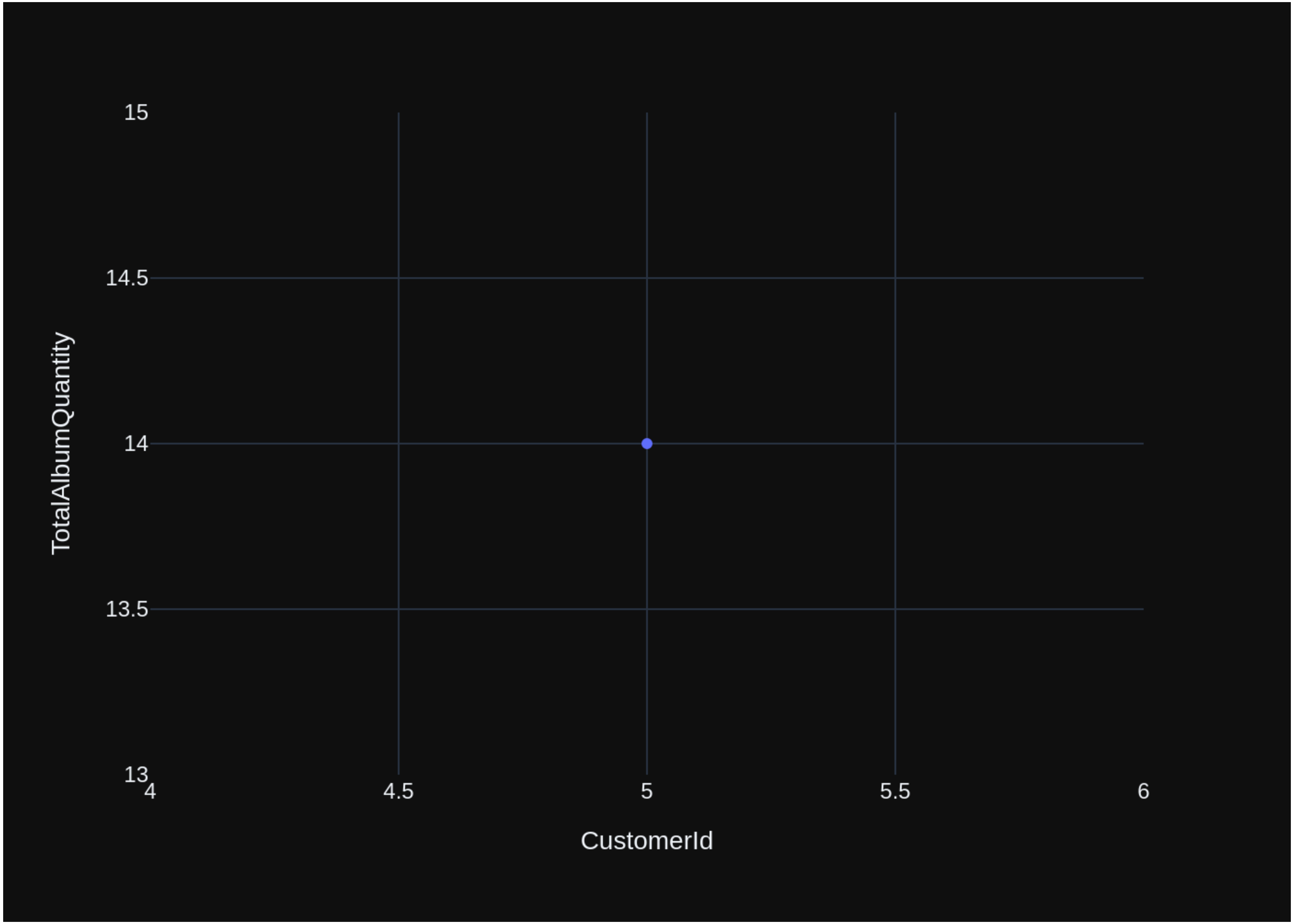
```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n    TrackId 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 INTEGE
R,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUME
RIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE N
O 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)  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 ACTIO
N ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES
"artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albu
ms" (ArtistId)\n\nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    C
ustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    Billi
ngCity 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" (Customer
Id) \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_InvoiceCustome
rId 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 Cont
ext \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is suffic
ient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is a
lmost sufficient but requires knowledge of a specific string in a particular column, please generate an 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 relevan
t table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was give
n before. \n'}, {'role': 'user', 'content': '  \n     Find the customer with the most invoices \n'}, {'role': 'assis
tant', 'content': 'SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId\nORDER BY Invoice
Count DESC\nLIMIT 1'}, {'role': 'user', 'content': '  \n    Get the total number of invoices for each customer\n'},
{'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId'},
{'role': 'user', 'content': '  \n    Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assi
stant', 'content': "SELECT SUM(Total) AS TotalInvoiced, InvoiceDate \nFROM invoices \nWHERE InvoiceDate >= '2010-01-
01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': '  \n    Get the top 10 most popular artists (based on the
number of tracks):\n'}, {'role': 'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artist
s a\nJOIN tracks t ON a.ArtistId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMIT 10'}, {'role': 'use
r', 'content': '  \n    Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT
\n    c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.Cus
tomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': '  \n    List all invoices with a total exceeding $1
```

```
0:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'user', 'content': '
\n    Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT *\nF
ROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Get the average invoice total f
or each customer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal\nFROM
invoices\nGROUP BY CustomerId'}, {'role': 'user', 'content': '  \n    List all albums and their corresponding artist
names  \n'}, {'role': 'assistant', 'content': 'SELECT \n    a.Title,\n    ar.Name\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 ge
nre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS TrackCount\nFROM genres g\nJOIN track
s 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'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question.
Your response should ONLY be based on the given context and follow the response guidelines and format instructions.
\n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NV
ARCHAR(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 ACTI
ON,\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 AC
TION\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 NUL
L,\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 \"a
lbums\" (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    BillingPostal
Code NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (C
ustomerId) \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_Invo
iceCustomerId 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 co
ntext is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provid
ed 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 inter
mediate_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     Find the customer with the most invoices \n"},
{"role": "assistant", "content": "SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId\nO
RDER BY InvoiceCount DESC\nLIMIT 1"}, {"role": "user", "content": "  \n    Get the total number of invoices for each
customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY
CustomerId"}, {"role": "user", "content": "  \n    Find all invoices since 2010 and the total amount invoiced:\n"},
{"role": "assistant", "content": "SELECT SUM(Total) AS TotalInvoiced, InvoiceDate \nFROM invoices \nWHERE InvoiceDat
e >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "content": "  \n    Get the top 10 most popular artists
(based on the number of tracks):\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TrackCoun
t\nFROM artists a\nJOIN tracks t ON a.ArtistId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMIT 10"},
{"role": "user", "content": "  \n    Find the total number of invoices per country:\n"}, {"role": "assistant", "cont
ent": "SELECT \n    c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.Cust
omerId = i.CustomerId\nGROUP BY c.Country"}, {"role": "user", "content": "  \n    List all invoices with a total exc
eeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "cont
ent": "  \n    Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SE
LECT *\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    Get the average invoic
e total for each customer:\n"}, {"role": "assistant", "content": "SELECT CustomerId, AVG(Total) AS AverageInvoiceTot
al\nFROM invoices\nGROUP BY CustomerId"}, {"role": "user", "content": "  \n    List all albums and their correspondi
ng artist names  \n"}, {"role": "assistant", "content": "SELECT \n    a.Title,\n    ar.Name\nFROM albums a\nJOIN art
ists ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "  \n    List all genres and the number of tracks
in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS TrackCount\nFROM genres g\n
JOIN tracks t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"role": "user", "content": "  \n     Find the customer w
ho bought the most albums in total quantity (across all invoices): \n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:49:31.020338821Z', 'message': {'role': 'assistant', 'conten
t': 'SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbumQuantity\nFROM customers c\nJOIN invoice_items ii ON c.Custo
merId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbumQuantity DESC\nLIMIT 1;\n\n\n\n'}, 'done_reason': 's
top', 'done': True, 'total_duration': 131408311447, 'load_duration': 690714, 'prompt_eval_count': 1414, 'prompt_eval
_duration': 117109796000, 'eval_count': 54, 'eval_duration': 13638948000}
LLM Response: SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbumQuantity
FROM customers c
JOIN invoice_items ii ON c.CustomerId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbumQuantity DESC
LIMIT 1;
```

```
Info: Output from LLM: SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbumQuantity
FROM customers c
JOIN invoice_items ii ON c.CustomerId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbumQuantity DESC
LIMIT 1;




Extracted SQL: SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbumQuantity
FROM customers c
JOIN invoice_items ii ON c.CustomerId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbumQuantity DESC
LIMIT 1
SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbumQuantity
FROM customers c
JOIN invoice_items ii ON c.CustomerId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbumQuantity DESC
LIMIT 1
   CustomerId  TotalAlbumQuantity
0           5                  14
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n     Find the customer who bought the most albums in total quantity (across al
l invoices): \n'\n\nThe DataFrame was produced using this query: SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbum
Quantity\nFROM customers c\nJOIN invoice_items ii ON c.CustomerId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY To
talAlbumQuantity DESC\nLIMIT 1\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning
df.dtypes gives:\n CustomerId            int64\nTotalAlbumQuantity    int64\ndtype: object"}, {"role": "user", "cont
ent": "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."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:50:05.345889243Z', 'message': {'role': 'assistant', 'conten
t': "```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n  fig = px.indicator(df, \n
```

```
value='TotalAlbumQuantity', \n                    title='Total Albums Purchased')\nelse:\n  fig = px.bar(df, x='Cust
omerId', y='TotalAlbumQuantity', title='Total Albums Purchased by Customer')\n  \nfig.show()\n```"}, 'done_reason':
'stop', 'done': True, 'total_duration': 34305851259, 'load_duration': 647175, 'prompt_eval_count': 208, 'prompt_eval
_duration': 13500563000, 'eval_count': 90, 'eval_duration': 20673705000}
```


15
14.5
14
13.5
13
TotalAlbumQuantity
4
4.5
5
5.5
6
CustomerId

Out[37]:

```
('SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbumQuantity\nFROM customers c\nJOIN invoice_items ii ON c.Custom
erId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbumQuantity DESC\nLIMIT 1',
    CustomerId  TotalAlbumQuantity
 0           5                  14,
 Figure({
     'data': [{'hovertemplate': 'CustomerId=%{x}<br>TotalAlbumQuantity=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'symbol': 'circle'},
               'mode': 'markers',
               'name': '',
               'orientation': 'v',
               'showlegend': False,
               'type': 'scatter',
               'x': array([5]),
               'xaxis': 'x',
               'y': array([14]),
               'yaxis': 'y'}],
     'layout': {'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalAlbumQuantity'}}}
 }))
```

In [38]:

```python
question = """
     Find the top 5 customer who bought the most albums in total quantity (across all invoices):
"""

vn.ask(question=question)
```

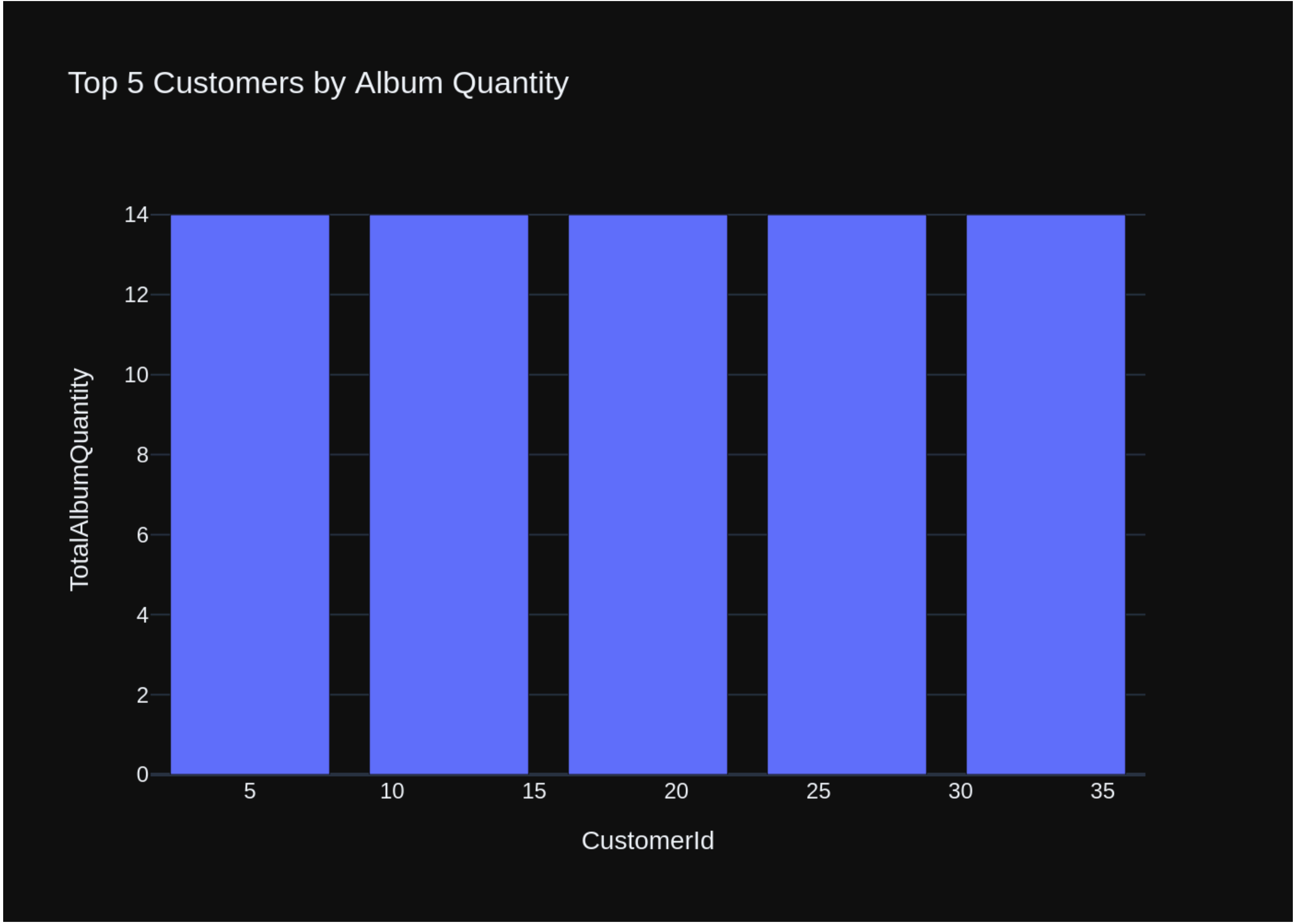
```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \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 INTEGE
R,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUME
RIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE N
O 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" (Arti
stId) \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) REFERENCE
S "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "t
racks" (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 "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Custo
merId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    BillingCi
ty NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVAR
CHAR(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" (Trac
kId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "i
nvoice_items" (InvoiceId)\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\nI
n the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, ple
ase generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost suf
ficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL quer
y 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 befor
e. \n'}, {'role': 'user', 'content': '  \n     Find the customer who bought the most albums in total quantity (acros
s all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbumQuantity
\nFROM customers c\nJOIN invoice_items ii ON c.CustomerId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbum
Quantity DESC\nLIMIT 1'}, {'role': 'user', 'content': '  \n    Get the top 10 most popular artists (based on the num
ber of tracks):\n'}, {'role': 'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artists a
\nJOIN tracks t ON a.ArtistId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMIT 10'}, {'role': 'user',
'content': '  \n     Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT Custome
rId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId\nORDER BY InvoiceCount DESC\nLIMIT 1'}, {'role':
'user', 'content': '  \n    Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant',
'content': 'SELECT *\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    List all
invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 1
0'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per country:\n'}, {'role': 'assistant',
'content': 'SELECT \n    c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON
c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': '  \n    Get the total number of invo
ices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invo
ices\nGROUP BY CustomerId'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'},
{'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER
BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Get the average invoice total for each custom
er:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal\nFROM invoices\nGROU
P BY CustomerId'}, {'role': 'user', 'content': '  \n    List all albums and their corresponding artist names  \n'},
{'role': 'assistant', 'content': 'SELECT \n    a.Title,\n    ar.Name\nFROM albums a\nJOIN artists ar ON a.ArtistId =
ar.ArtistId'}, {'role': 'user', 'content': '  \n     Find the top 5 customer who bought the most albums in total qua
ntity (across all invoices):\n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question.
Your response should ONLY be based on the given context and follow the response guidelines and format instructions.
\n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NV
ARCHAR(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 ACTI
ON,\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 AC
TION\r\n)\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 \"artists\" (Arti
stId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n    InvoiceLine
Id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NUL
L,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFER
ENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFEREN
CES \"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 \"invoices\"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\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    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customer
s\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"in
voice_items\" (TrackId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK_Invoi
ceLineInvoiceId ON \"invoice_items\" (InvoiceId)\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 prov
ided 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 gen
erate 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 ex
actly as it was given before. \n"}, {"role": "user", "content": "  \n     Find the customer who bought the most albu
ms in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(ii.Quan
tity) AS TotalAlbumQuantity\nFROM customers c\nJOIN invoice_items ii ON c.CustomerId = ii.InvoiceId\nGROUP BY c.Cust
omerId\nORDER BY TotalAlbumQuantity DESC\nLIMIT 1"}, {"role": "user", "content": "  \n    Get the top 10 most popula
r artists (based on the number of tracks):\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS
TrackCount\nFROM artists a\nJOIN tracks t ON a.ArtistId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMI
T 10"}, {"role": "user", "content": "  \n     Find the customer with the most invoices \n"}, {"role": "assistant",
"content": "SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId\nORDER BY InvoiceCount D
ESC\nLIMIT 1"}, {"role": "user", "content": "  \n    Find the top 5 most expensive tracks (based on unit pric
e):\n"}, {"role": "assistant", "content": "SELECT *\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "use
r", "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 count
ry:\n"}, {"role": "assistant", "content": "SELECT \n    c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM cu
stomers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"role": "user", "content": "  \n
Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT
(*) AS InvoiceCount\nFROM invoices\nGROUP BY CustomerId"}, {"role": "user", "content": "what are the top 5 countries
that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM custo
mers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    Get the average
invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT CustomerId, AVG(Total) AS AverageInvo
iceTotal\nFROM invoices\nGROUP BY CustomerId"}, {"role": "user", "content": "  \n    List all albums and their corre
sponding artist names  \n"}, {"role": "assistant", "content": "SELECT \n    a.Title,\n    ar.Name\nFROM albums a\nJO
IN artists ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "  \n     Find the top 5 customer who bough
t the most albums in total quantity (across all invoices):\n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:52:17.533500354Z', 'message': {'role': 'assistant', 'conten
t': 'SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbumQuantity\nFROM customers c\nJOIN invoice_items ii ON c.Custo
merId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbumQuantity DESC\nLIMIT 5;'}, 'done_reason': 'stop', 'd
one': True, 'total_duration': 132098916114, 'load_duration': 694208, 'prompt_eval_count': 1411, 'prompt_eval_duratio
n': 117828339000, 'eval_count': 53, 'eval_duration': 13605504000}
LLM Response: SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbumQuantity
FROM customers c
JOIN invoice_items ii ON c.CustomerId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbumQuantity DESC
LIMIT 5;
Info: Output from LLM: SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbumQuantity
FROM customers c
```

```
JOIN invoice_items ii ON c.CustomerId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbumQuantity DESC
LIMIT 5;
Extracted SQL: SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbumQuantity
FROM customers c
JOIN invoice_items ii ON c.CustomerId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbumQuantity DESC
LIMIT 5
SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbumQuantity
FROM customers c
JOIN invoice_items ii ON c.CustomerId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbumQuantity DESC
LIMIT 5
   CustomerId  TotalAlbumQuantity
0           5                  14
1          12                  14
2          19                  14
3          26                  14
4          33                  14
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n     Find the top 5 customer who bought the most albums in total quantity (acr
oss all invoices):\n'\n\nThe DataFrame was produced using this query: SELECT c.CustomerId, SUM(ii.Quantity) AS Total
AlbumQuantity\nFROM customers c\nJOIN invoice_items ii ON c.CustomerId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER
BY TotalAlbumQuantity DESC\nLIMIT 5\n\nThe following is information about the resulting pandas DataFrame 'df': \nRun
ning df.dtypes gives:\n CustomerId            int64\nTotalAlbumQuantity    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 p
andas 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."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:52:53.26032175Z', 'message': {'role': 'assistant', 'conten
t': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n  fig = px.indicator(df, \n
value="TotalAlbumQuantity",\n                    title="Top Customer by Album Quantity")\nelse:\n  fig = px.bar(df,
x="CustomerId", y="TotalAlbumQuantity", \n               title="Top 5 Customers by Album Quantity")\nfig.show() \n``
```

```
`'}, 'done_reason': 'stop', 'done': True, 'total_duration': 35699299149, 'load_duration': 642532, 'prompt_eval_coun
t': 210, 'prompt_eval_duration': 13495616000, 'eval_count': 95, 'eval_duration': 22069203000}
```



Out[38]:

```
('SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbumQuantity\nFROM customers c\nJOIN invoice_items ii ON c.Custom
erId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbumQuantity DESC\nLIMIT 5',
    CustomerId  TotalAlbumQuantity
 0           5                  14
 1          12                  14
 2          19                  14
 3          26                  14
 4          33                  14,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'CustomerId=%{x}<br>TotalAlbumQuantity=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([ 5, 12, 19, 26, 33]),
               'xaxis': 'x',
               'y': array([14, 14, 14, 14, 14]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Top 5 Customers by Album Quantity'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalAlbumQuantity'}}}
 }))
```

In [39]:

```
question = """
     Find the top 3 customers who spent the most money overall:
"""

vn.ask(question=question)
```

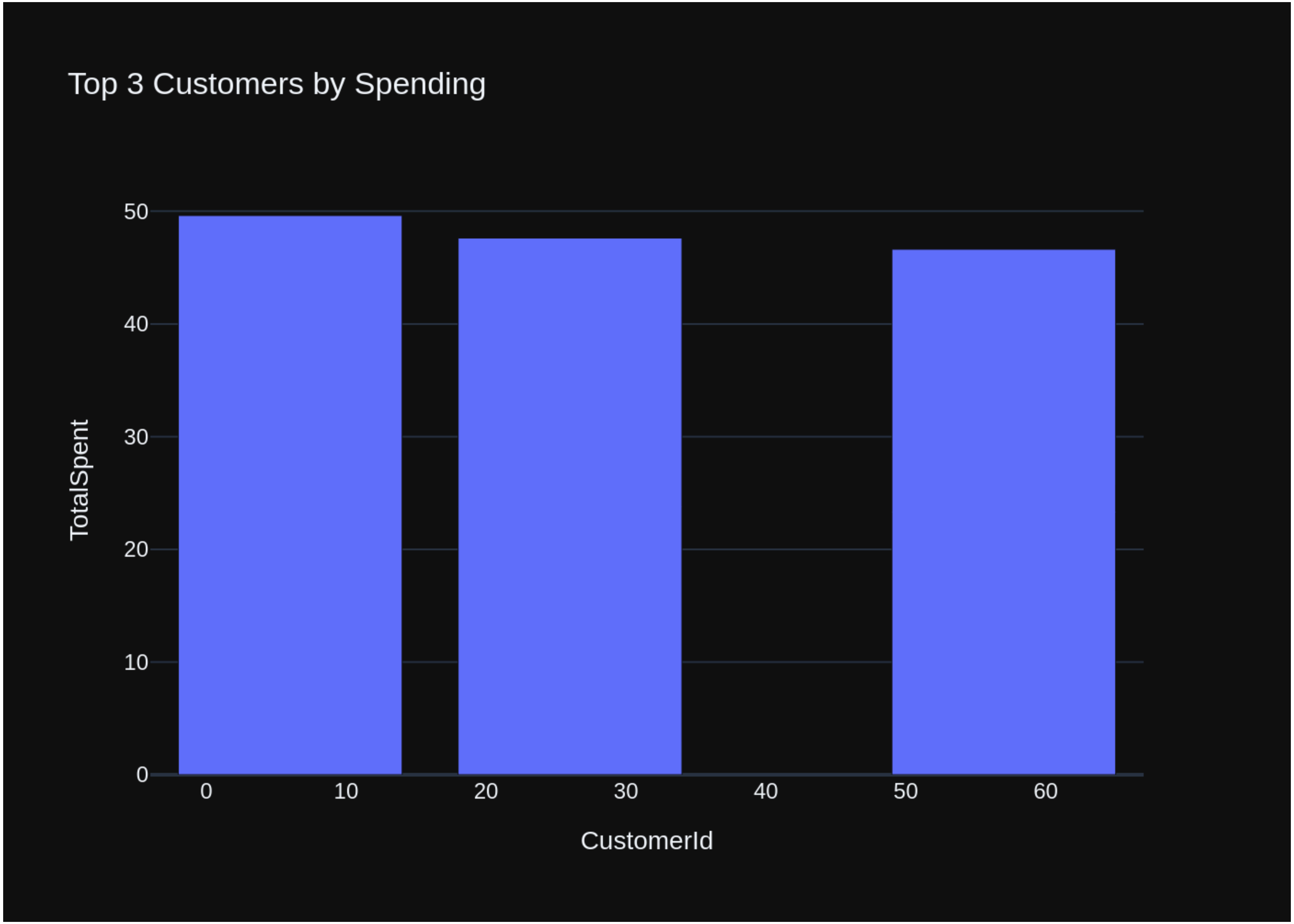
```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\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    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "cu
stomers" (SupportRepId)\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 N
O ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREME
NT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHA
R(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(4
0),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  N
OT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\t
ON DELETE NO ACTION ON UPDATE NO ACTION\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 NUL
L,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGE
R,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON D
ELETE 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 DELET
E NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE TABL
E "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" (PlaylistI
d) \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_InvoiceLineTrackId ON "invoice_items" (TrackId)\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 prov
ided context is almost sufficient but requires knowledge of a specific string in a particular column, please generat
e an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying int
ermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please u
se the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exa
ctly as it was given before. \n'}, {'role': 'user', 'content': '  \n     Find the top 5 customer who bought the most
albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(ii.Q
uantity) AS TotalAlbumQuantity\nFROM customers c\nJOIN invoice_items ii ON c.CustomerId = ii.InvoiceId\nGROUP BY c.C
ustomerId\nORDER BY TotalAlbumQuantity DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find the top 5 most exp
ensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT *\nFROM tracks\nORDER BY UnitPric
e DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n     Find the customer who bought the most albums in total quant
ity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlb
```

```
umQuantity\nFROM customers c\nJOIN invoice_items ii ON c.CustomerId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY
TotalAlbumQuantity DESC\nLIMIT 1'}, {'role': 'user', 'content': '  \n     Find the customer with the most invoices
\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices\nGROUP BY Custome
rId\nORDER BY InvoiceCount DESC\nLIMIT 1'}, {'role': 'user', 'content': 'what are the top 5 countries that customers
come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY
Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Get the average invoice total
for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal\nFROM
invoices\nGROUP BY CustomerId'}, {'role': 'user', 'content': '  \n    Get the top 10 most popular artists (based on
the number of tracks):\n'}, {'role': 'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM ar
tists a\nJOIN tracks t ON a.ArtistId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMIT 10'}, {'role': 'u
ser', 'content': '  \n    Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELE
CT \n    c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.
CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant',
'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': '  \n    Get the total number of invoices
for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices
\nGROUP BY CustomerId'}, {'role': 'user', 'content': '  \n     Find the top 3 customers who spent the most money ove
rall:\n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: 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    Cus
tomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    Billing
City NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NV
ARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (Customer
Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\"
(SupportRepId)\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)  NOT NULL,\r
\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELE
TE 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 AUTOINCREME
NT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHA
R(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(4
0),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  N
OT 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 AU
TOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT N
ULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTE
GER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\t
ON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DEL
ETE 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 \"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 \"playlis
ts\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks
\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoice
s\" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\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 quest
ion. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular c
olumn, 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 gener
ated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please re
peat the answer exactly as it was given before. \n"}, {"role": "user", "content": "  \n     Find the top 5 customer
who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.
CustomerId, SUM(ii.Quantity) AS TotalAlbumQuantity\nFROM customers c\nJOIN invoice_items ii ON c.CustomerId = ii.Inv
oiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbumQuantity DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    Fin
d the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT *\nFROM track
s\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "  \n     Find the customer who bought the most a
lbums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(ii.Q
uantity) AS TotalAlbumQuantity\nFROM customers c\nJOIN invoice_items ii ON c.CustomerId = ii.InvoiceId\nGROUP BY c.C
ustomerId\nORDER BY TotalAlbumQuantity DESC\nLIMIT 1"}, {"role": "user", "content": "  \n     Find the customer with
the most invoices \n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) AS InvoiceCount\nFROM invoices
\nGROUP BY CustomerId\nORDER BY InvoiceCount DESC\nLIMIT 1"}, {"role": "user", "content": "what are the top 5 countr
ies that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM c
ustomers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    Get the ave
rage invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT CustomerId, AVG(Total) AS Averag
eInvoiceTotal\nFROM invoices\nGROUP BY CustomerId"}, {"role": "user", "content": "  \n    Get the top 10 most popula
r artists (based on the number of tracks):\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS
TrackCount\nFROM artists a\nJOIN tracks t ON a.ArtistId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMI
T 10"}, {"role": "user", "content": "  \n    Find the total number of invoices per country:\n"}, {"role": "assistan
t", "content": "SELECT \n    c.Country, \n    COUNT(i.InvoiceId) AS InvoiceCount\nFROM customers c\nJOIN invoices i
ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"role": "user", "content": "How many customers are there"},
{"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "  \n    Get the tot
al number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) AS Invoic
eCount\nFROM invoices\nGROUP BY CustomerId"}, {"role": "user", "content": "  \n     Find the top 3 customers who spe
nt the most money overall:\n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:55:21.701182654Z', 'message': {'role': 'assistant', 'conten
```

```
t': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.Customer
Id\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 3;'}, 'done_reason': 'stop', 'done': True, 'total_duratio
n': 148351772312, 'load_duration': 742949, 'prompt_eval_count': 1582, 'prompt_eval_duration': 135050132000, 'eval_co
unt': 48, 'eval_duration': 12623285000}
LLM Response: SELECT c.CustomerId, SUM(i.Total) AS TotalSpent
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 3;
Info: Output from LLM: SELECT c.CustomerId, SUM(i.Total) AS TotalSpent
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 3;
Extracted SQL: SELECT c.CustomerId, SUM(i.Total) AS TotalSpent
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 3
SELECT c.CustomerId, SUM(i.Total) AS TotalSpent
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 3
   CustomerId  TotalSpent
0           6       49.62
1          26       47.62
2          57       46.62
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n     Find the top 3 customers who spent the most money overall:\n'\n\nThe Data
Frame was produced using this query: SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM customers c\nJOIN invoice
s i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 3\n\nThe following is inf
```

```
ormation about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId      int64\nTotalSpent
float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results
of the dataframe? Assume 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."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:55:55.352252087Z', 'message': {'role': 'assistant', 'conten
t': "```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n  fig = px.indicator(df, \n
value='TotalSpent',\n                     title='Top Customer Spending')\nelse:\n  fig = px.bar(df, x='CustomerId',
y='TotalSpent', \n               title='Top 3 Customers by Spending')\n\nfig.show()\n```"}, 'done_reason': 'stop',
'done': True, 'total_duration': 33623741856, 'load_duration': 657290, 'prompt_eval_count': 198, 'prompt_eval_duratio
n': 12860844000, 'eval_count': 89, 'eval_duration': 20631804000}
```


Top 3 Customers by Spending
TotalSpent
0
10
20
30
40
50
CustomerId
0
10
20
30
40
50
60

Out[39]:

```
('SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerI
d\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 3',
    CustomerId  TotalSpent
 0           6       49.62
 1          26       47.62
 2          57       46.62,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'CustomerId=%{x}<br>TotalSpent=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([ 6, 26, 57]),
               'xaxis': 'x',
               'y': array([49.62, 47.62, 46.62]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Top 3 Customers by Spending'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalSpent'}}}
 }))
```

In [40]:

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

vn.ask(question=question)
```

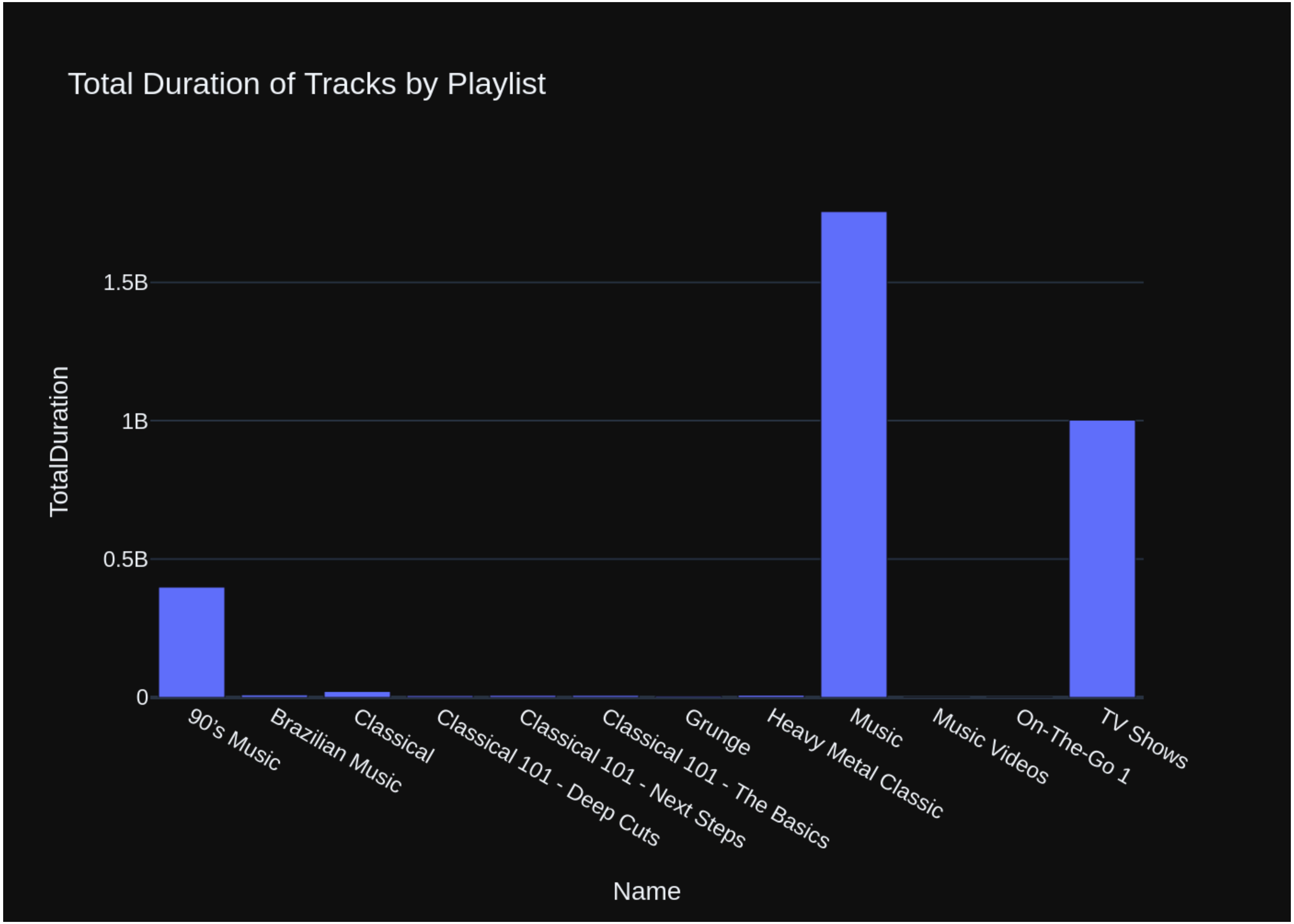
```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE TABLE "play
lists"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE T
ABLE "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" (Playlis
tId) \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 INT
EGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tO
N 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 DE
LETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK_T
rackAlbumId 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) REF
ERENCES "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, pleas
e generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost suffi
cient 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 pr
ovided 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 befor
e. \n'}, {'role': 'user', 'content': '  \n    List all genres and the number of tracks in each genre:\n'}, {'role': 
'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = 
t.GenreId\nGROUP BY g.Name'}, {'role': 'user', 'content': '  \n    Get the top 10 most popular artists (based on the 
number of tracks):\n'}, {'role': 'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artist
s a\nJOIN tracks t ON a.ArtistId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMIT 10'}, {'role': 'use
r', 'content': '  \n    Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 
'content': "SELECT * FROM tracks WHERE Name LIKE '%What%' \n"}, {'role': 'user', 'content': '  \n    Find the top 5 
most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT *\nFROM tracks\nORDER BY 
UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n     Find the top 5 customer who bought the most albums 
in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(ii.Quantit
y) AS TotalAlbumQuantity\nFROM customers c\nJOIN invoice_items ii ON c.CustomerId = ii.InvoiceId\nGROUP BY c.Custome
rId\nORDER BY TotalAlbumQuantity DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n     Find the customer who bought 
the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerI
d, SUM(ii.Quantity) AS TotalAlbumQuantity\nFROM customers c\nJOIN invoice_items ii ON c.CustomerId = ii.InvoiceId\nG
ROUP BY c.CustomerId\nORDER BY TotalAlbumQuantity DESC\nLIMIT 1'}, {'role': 'user', 'content': '  \n    List all alb
ums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT \n    a.Title,\n    ar.Name
```

```
\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': '  \n    Find all invoic
es since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT SUM(Total) AS TotalInvoic
ed, InvoiceDate \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'conte
nt': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM inv
oices WHERE Total > 10'}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role':
'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': '  \n
Get all playlists containing at least 10 tracks and the total duration of those tracks:\n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question.
Your response should ONLY be based on the given context and follow the response guidelines and format instructions.
\n===Tables \nCREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\n\nCREATE TABLE \"playlists\"\r
\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"p
laylist_track\"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_Pla
ylistTrack 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 AU
TOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT N
ULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTE
GER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\t
ON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DEL
ETE 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 IN
DEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCRE
ATE 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 TABL
E \"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 co
ntext is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provid
ed 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 inter
mediate_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    List all genres and the number of tracks in eac
h genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS TrackCount\nFROM genres g\nJOIN t
racks t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"role": "user", "content": "  \n    Get the top 10 most popula
r artists (based on the number of tracks):\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS
```

```
TrackCount\nFROM artists a\nJOIN tracks t ON a.ArtistId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMI
T 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%' \n"}, {"role": "user", "content": "
\n    Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT *\nF
ROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "  \n     Find the top 5 customer who bo
ught the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.Custom
erId, SUM(ii.Quantity) AS TotalAlbumQuantity\nFROM customers c\nJOIN invoice_items ii ON c.CustomerId = ii.InvoiceId
\nGROUP BY c.CustomerId\nORDER BY TotalAlbumQuantity DESC\nLIMIT 5"}, {"role": "user", "content": "  \n     Find the
customer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content":
"SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbumQuantity\nFROM customers c\nJOIN invoice_items ii ON c.CustomerI
d = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbumQuantity DESC\nLIMIT 1"}, {"role": "user", "content": "
\n    List all albums and their corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT \n    a.T
itle,\n    ar.Name\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 SUM(Total)
AS TotalInvoiced, InvoiceDate \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": "SEL
ECT * FROM invoices WHERE Total > 10"}, {"role": "user", "content": "Show me a list of tables in the SQLite databas
e"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "conten
t": "  \n     Get all playlists containing at least 10 tracks and the total duration of those tracks:\n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:58:08.98894483Z', 'message': {'role': 'assistant', 'conten
t': 'SELECT p.Name, SUM(t.Milliseconds) AS TotalDuration\nFROM playlists p\nJOIN playlist_track pt ON p.PlaylistId =
pt.PlaylistId\nJOIN tracks t ON pt.TrackId = t.TrackId\nGROUP BY p.Name\nHAVING SUM(t.Milliseconds) >= 10000; -- Ass
uming 10000 milliseconds = 10 seconds per track\n\n\n\n'}, 'done_reason': 'stop', 'done': True, 'total_duration': 13
3547040562, 'load_duration': 756087, 'prompt_eval_count': 1337, 'prompt_eval_duration': 110347426000, 'eval_count':
89, 'eval_duration': 22529874000}
LLM Response: SELECT p.Name, SUM(t.Milliseconds) AS TotalDuration
FROM playlists p
JOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId
JOIN tracks t ON pt.TrackId = t.TrackId
GROUP BY p.Name
HAVING SUM(t.Milliseconds) >= 10000; -- Assuming 10000 milliseconds = 10 seconds per track




Info: Output from LLM: SELECT p.Name, SUM(t.Milliseconds) AS TotalDuration
FROM playlists p
JOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId
JOIN tracks t ON pt.TrackId = t.TrackId
GROUP BY p.Name
```

```
HAVING SUM(t.Milliseconds) >= 10000; -- Assuming 10000 milliseconds = 10 seconds per track




Extracted SQL: SELECT p.Name, SUM(t.Milliseconds) AS TotalDuration
FROM playlists p
JOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId
JOIN tracks t ON pt.TrackId = t.TrackId
GROUP BY p.Name
HAVING SUM(t.Milliseconds) >= 10000
SELECT p.Name, SUM(t.Milliseconds) AS TotalDuration
FROM playlists p
JOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId
JOIN tracks t ON pt.TrackId = t.TrackId
GROUP BY p.Name
HAVING SUM(t.Milliseconds) >= 10000
                          Name  TotalDuration
0                   90’s Music      398705153
1              Brazilian Music        9486559
2                    Classical       21770592
3    Classical 101 - Deep Cuts        6755730
4   Classical 101 - Next Steps        7575051
5   Classical 101 - The Basics        7439811
6                       Grunge        4122018
7          Heavy Metal Classic        8206312
8                        Music     1755366166
9                 Music Videos         294294
10                 On-The-Go 1         197459
11                    TV Shows     1002189914
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers 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, SUM(t.Milliseconds) AS TotalDuration
\nFROM playlists p\nJOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId\nJOIN tracks t ON pt.TrackId = t.TrackId
\nGROUP BY p.Name\nHAVING SUM(t.Milliseconds) >= 10000\n\nThe following is information about the resulting pandas Da
taFrame 'df': \nRunning df.dtypes gives:\n Name             object\nTotalDuration     int64\ndtype: object"}, {"rol
e": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the da
ta 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."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T01:58:43.801323642Z', 'message': {'role': 'assistant', 'conten
t': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n  fig = px.indicator(df, name="TotalDuration",
value="TotalDuration", title="Total Duration of Tracks")\nelse:\n  fig = px.bar(df, x="Name", y="TotalDuration", tit
le="Total Duration of Tracks by Playlist")\n  \nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_durat
ion': 34782591439, 'load_duration': 685245, 'prompt_eval_count': 224, 'prompt_eval_duration': 14415996000, 'eval_cou
nt': 89, 'eval_duration': 20233426000}
```


Total Duration of Tracks by Playlist
TotalDuration
1.5B
1B
0.5B
0
90's Music
Brazilian Music
Classical
Classical 101 - Deep Cuts
Classical 101 - Next Steps
Classical 101 - The Basics
Grunge
Heavy Metal Classic
Music
Music Videos
On-The-Go 1
TV Shows
Name

```
Out[40]:  ('SELECT p.Name, SUM(t.Milliseconds) AS TotalDuration\nFROM playlists p\nJOIN playlist_track pt ON p.PlaylistId =
          pt.PlaylistId\nJOIN tracks t ON pt.TrackId = t.TrackId\nGROUP BY p.Name\nHAVING SUM(t.Milliseconds) >= 10000',
                                     Name  TotalDuration
           0                   90’s Music      398705153
           1              Brazilian Music        9486559
           2                    Classical       21770592
           3    Classical 101 - Deep Cuts        6755730
           4   Classical 101 - Next Steps        7575051
           5   Classical 101 - The Basics        7439811
           6                       Grunge        4122018
           7          Heavy Metal Classic        8206312
           8                        Music     1755366166
           9                 Music Videos         294294
           10                 On-The-Go 1         197459
           11                    TV Shows     1002189914,
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'Name=%{x}<br>TotalDuration=%{y}<extra></extra>',
                         'legendgroup': '',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                         'name': '',
                         'offsetgroup': '',
                         'orientation': 'v',
                         'showlegend': False,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array(['90’s Music', 'Brazilian Music', 'Classical',
                                     'Classical 101 - Deep Cuts', 'Classical 101 - Next Steps',
                                     'Classical 101 - The Basics', 'Grunge', 'Heavy Metal Classic', 'Music',
                                     'Music Videos', 'On-The-Go 1', 'TV Shows'], dtype=object),
                         'xaxis': 'x',
                         'y': array([ 398705153,    9486559,   21770592,    6755730,    7575051,    7439811,
                                        4122018,    8206312, 1755366166,     294294,     197459, 1002189914]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'template': '...',
                          'title': {'text': 'Total Duration of Tracks by Playlist'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                        'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalDuration'}}}
          }))
```

```
In [41]: 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
```

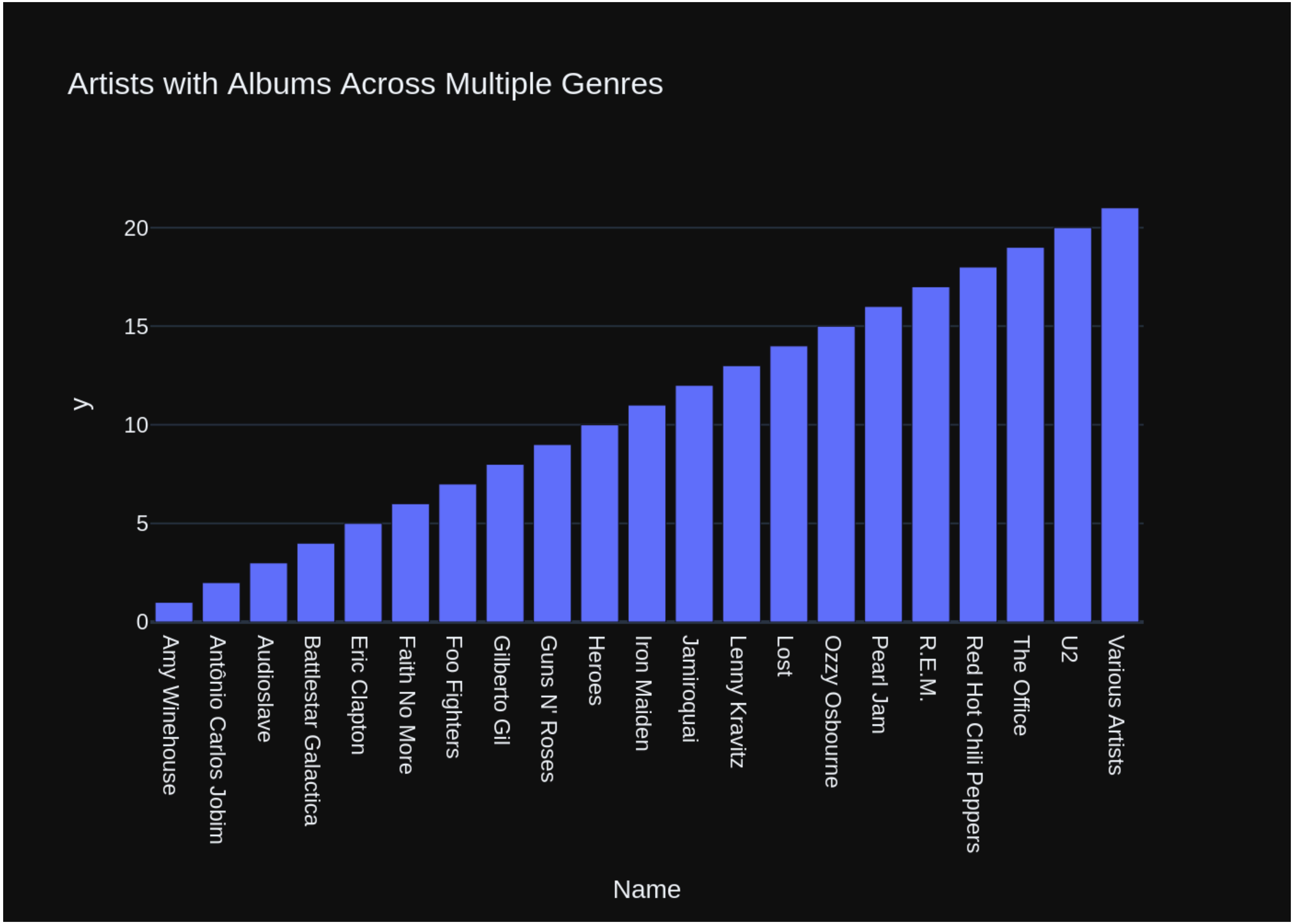
```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n    TrackId 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 INTEGE
R,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUME
RIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE N
O 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 INTEGE
R 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\nCREAT
E 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_tra
ck" (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) REF
ERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFER
ENCES "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 ge
nerate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficien
t but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to f
ind the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provid
ed context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s).
\n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before.
\n'}, {'role': 'user', 'content': '  \n    Get the top 10 most popular artists (based on the number of tracks):\n'},
{'role': 'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artists a\nJOIN tracks t ON a.
ArtistId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMIT 10'}, {'role': 'user', 'content': '  \n    Li
st all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT \n    a.Title,\n
ar.Name\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.GenreI
d) AS TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.GenreId\nGROUP BY g.Name'}, {'role': 'user', 'conten
t': '  \n     Get all playlists containing at least 10 tracks and the total duration of those tracks:\n'}, {'role':
'assistant', 'content': 'SELECT p.Name, SUM(t.Milliseconds) AS TotalDuration\nFROM playlists p\nJOIN playlist_track
pt ON p.PlaylistId = pt.PlaylistId\nJOIN tracks t ON pt.TrackId = t.TrackId\nGROUP BY p.Name\nHAVING SUM(t.Milliseco
nds) >= 10000'}, {'role': 'user', 'content': '  \n     Find the customer who bought the most albums in total quantit
y (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(ii.Quantity) AS TotalAlbum
Quantity\nFROM customers c\nJOIN invoice_items ii ON c.CustomerId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY To
talAlbumQuantity DESC\nLIMIT 1'}, {'role': 'user', 'content': '  \n     Find the top 5 customer who bought the most
albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(ii.Q
uantity) AS TotalAlbumQuantity\nFROM customers c\nJOIN invoice_items ii ON c.CustomerId = ii.InvoiceId\nGROUP BY c.C
```

```
ustomerId\nORDER BY TotalAlbumQuantity DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find the top 5 most exp
ensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT *\nFROM tracks\nORDER BY UnitPric
e DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find all tracks with a name containing "What" (case-insensit
ive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%' \n"}, {'role': 'user', 'co
ntent': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlit
e_master WHERE type='table'"}, {'role': 'user', 'content': '  \n     Find the top 3 customers who spent the most mon
ey overall:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM customers c
\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 3'}, {'rol
e': 'user', 'content': '  \n     Identify artists who have albums with tracks appearing in multiple genres:\n'}]
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question.
Your response should ONLY be based on the given context and follow the response guidelines and format instructions.
\n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NV
ARCHAR(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 ACTI
ON,\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 AC
TION\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 IN
TEGER 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 INT
EGER 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    FOREI
GN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FO
REIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Add
itional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided conte
xt is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided
context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an
intermediate 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. 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 top 10 most popular artists (based on the nu
mber of tracks):\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artists
```

```
a\nJOIN tracks t ON a.ArtistId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMIT 10"}, {"role": "user",
"content": "  \n    List all albums and their corresponding artist names  \n"}, {"role": "assistant", "content": "SE
LECT \n    a.Title,\n    ar.Name\nFROM albums a\nJOIN artists ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "co
ntent": "  \n    List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELE
CT g.Name, COUNT(t.GenreId) AS TrackCount\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 thos
e tracks:\n"}, {"role": "assistant", "content": "SELECT p.Name, SUM(t.Milliseconds) AS TotalDuration\nFROM playlists
p\nJOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId\nJOIN tracks t ON pt.TrackId = t.TrackId\nGROUP BY p.Name
\nHAVING SUM(t.Milliseconds) >= 10000"}, {"role": "user", "content": "  \n     Find the customer who bought the most
albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(ii.
Quantity) AS TotalAlbumQuantity\nFROM customers c\nJOIN invoice_items ii ON c.CustomerId = ii.InvoiceId\nGROUP BY c.
CustomerId\nORDER BY TotalAlbumQuantity DESC\nLIMIT 1"}, {"role": "user", "content": "  \n     Find the top 5 custom
er who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT
c.CustomerId, SUM(ii.Quantity) AS TotalAlbumQuantity\nFROM customers c\nJOIN invoice_items ii ON c.CustomerId = ii.I
nvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbumQuantity DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    F
ind the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT *\nFROM tra
cks\nORDER BY UnitPrice DESC\nLIMIT 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 '%What%'
\n"}, {"role": "user", "content": "Show me a list of tables in the SQLite database"}, {"role": "assistant", "conten
t": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "content": "  \n     Find the top 3 custo
mers who spent the most money overall:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS T
otalSpent\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSp
ent DESC\nLIMIT 3"}, {"role": "user", "content": "  \n     Identify artists who have albums with tracks appearing in
multiple genres:\n"}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T02:00:58.637184128Z', 'message': {'role': 'assistant', 'conten
t': 'SELECT a.Name\nFROM artists a\nJOIN albums alb ON a.ArtistId = alb.ArtistId\nJOIN tracks tr ON alb.AlbumId = t
r.AlbumId\nJOIN genres g ON tr.GenreId = g.GenreId\nGROUP BY a.Name\nHAVING COUNT(DISTINCT g.GenreId) > 1;\n\n\n
\n'}, 'done_reason': 'stop', 'done': True, 'total_duration': 134732818408, 'load_duration': 777932, 'prompt_eval_cou
nt': 1402, 'prompt_eval_duration': 115126773000, 'eval_count': 74, 'eval_duration': 18938596000}
LLM Response: SELECT a.Name
FROM artists a
JOIN albums alb ON a.ArtistId = alb.ArtistId
JOIN tracks tr ON alb.AlbumId = tr.AlbumId
JOIN genres g ON tr.GenreId = g.GenreId
GROUP BY a.Name
HAVING COUNT(DISTINCT g.GenreId) > 1;
```

```
Info: Output from LLM: SELECT a.Name
FROM artists a
JOIN albums alb ON a.ArtistId = alb.ArtistId
JOIN tracks tr ON alb.AlbumId = tr.AlbumId
JOIN genres g ON tr.GenreId = g.GenreId
GROUP BY a.Name
HAVING COUNT(DISTINCT g.GenreId) > 1;




Extracted SQL: SELECT a.Name
FROM artists a
JOIN albums alb ON a.ArtistId = alb.ArtistId
JOIN tracks tr ON alb.AlbumId = tr.AlbumId
JOIN genres g ON tr.GenreId = g.GenreId
GROUP BY a.Name
HAVING COUNT(DISTINCT g.GenreId) > 1
SELECT a.Name
FROM artists a
JOIN albums alb ON a.ArtistId = alb.ArtistId
JOIN tracks tr ON alb.AlbumId = tr.AlbumId
JOIN genres g ON tr.GenreId = g.GenreId
GROUP BY a.Name
HAVING COUNT(DISTINCT g.GenreId) > 1
                     Name
0           Amy Winehouse
1    Antônio Carlos Jobim
2             Audioslave
3    Battlestar Galactica
4            Eric Clapton
5           Faith No More
6            Foo Fighters
7            Gilberto Gil
8           Guns N' Roses
9                  Heroes
10            Iron Maiden
11             Jamiroquai
12          Lenny Kravitz
13                   Lost
14          Ozzy Osbourne
```

~~~~
15              Pearl Jam
16                 R.E.M.
17  Red Hot Chili Peppers
18             The Office
19                     U2
20        Various Artists
Info: Ollama parameters:
model=gemma2:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answ
ers the question the user asked: '  \n     Identify artists who have albums with tracks appearing in multiple genre
s:\n'\n\nThe DataFrame was produced using this query: SELECT a.Name\nFROM artists a\nJOIN albums alb ON a.ArtistId =
alb.ArtistId\nJOIN tracks tr ON alb.AlbumId = tr.AlbumId\nJOIN genres g ON tr.GenreId = g.GenreId\nGROUP BY a.Name\n
HAVING COUNT(DISTINCT g.GenreId) > 1\n\nThe following is information about the resulting pandas DataFrame 'df': \nRu
nning df.dtypes gives:\n Name    object\ndtype: object"}, {"role": "user", "content": "Can you generate the Python p
lotly 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 explanation
s -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:latest', 'created_at': '2024-07-22T02:01:33.603523869Z', 'message': {'role': 'assistant', 'conten
t': '```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n  fig = px.indicator(df, \n
value="Name",\n                    title="Artists with Albums Across Multiple Genres")\nelse:\n  fig = px.bar(df, x
="Name", y=df.index+1, title="Artists with Albums Across Multiple Genres") \nfig.show()\n```'}, 'done_reason': 'sto
p', 'done': True, 'total_duration': 34938278883, 'load_duration': 707864, 'prompt_eval_count': 212, 'prompt_eval_dur
ation': 13483406000, 'eval_count': 92, 'eval_duration': 21324524000}
~~~~


Artists with Albums Across Multiple Genres
y
0
5
10
15
20
Amy Winehouse
Antônio Carlos Jobim
Audioslave
Battlestar Galactica
Eric Clapton
Faith No More
Foo Fighters
Gilberto Gil
Guns N' Roses
Heroes
Iron Maiden
Jamiroquai
Lenny Kravitz
Lost
Ozzy Osbourne
Pearl Jam
R.E.M.
Red Hot Chili Peppers
The Office
U2
Various Artists
Name

```
Out[41]: ('SELECT a.Name\nFROM artists a\nJOIN albums alb ON a.ArtistId = alb.ArtistId\nJOIN tracks tr ON alb.AlbumId = tr.
         AlbumId\nJOIN genres g ON tr.GenreId = g.GenreId\nGROUP BY a.Name\nHAVING COUNT(DISTINCT g.GenreId) > 1',
                              Name
          0           Amy Winehouse
          1    Antônio Carlos Jobim
          2              Audioslave
          3    Battlestar Galactica
          4            Eric Clapton
          5           Faith No More
          6            Foo Fighters
          7            Gilberto Gil
          8           Guns N' Roses
          9                  Heroes
          10            Iron Maiden
          11             Jamiroquai
          12          Lenny Kravitz
          13                   Lost
          14          Ozzy Osbourne
          15              Pearl Jam
          16                 R.E.M.
          17  Red Hot Chili Peppers
          18             The Office
          19                     U2
          20        Various Artists,
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'Name=%{x}<br>y=%{y}<extra></extra>',
                        'legendgroup': '',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array(['Amy Winehouse', 'Antônio Carlos Jobim', 'Audioslave',
                                    'Battlestar Galactica', 'Eric Clapton', 'Faith No More', 'Foo Fighters',
                                    'Gilberto Gil', "Guns N' Roses", 'Heroes', 'Iron Maiden', 'Jamiroquai',
                                    'Lenny Kravitz', 'Lost', 'Ozzy Osbourne', 'Pearl Jam', 'R.E.M.',
                                    'Red Hot Chili Peppers', 'The Office', 'U2', 'Various Artists'],
                                   dtype=object),
                        'xaxis': 'x',
                        'y': array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
                                    19, 20, 21]),
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                         'legend': {'tracegroupgap': 0},
                         'template': '...',
                         'title': {'text': 'Artists with Albums Across Multiple Genres'},
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'y'}}}
          }))
```

Check completion time

In [42]:

```python
ts_stop = time()

elapsed_time = ts_stop - ts_start
print(f"test running on '{hostname}' with '{model_name}' LLM took : {elapsed_time:.2f} sec")
```

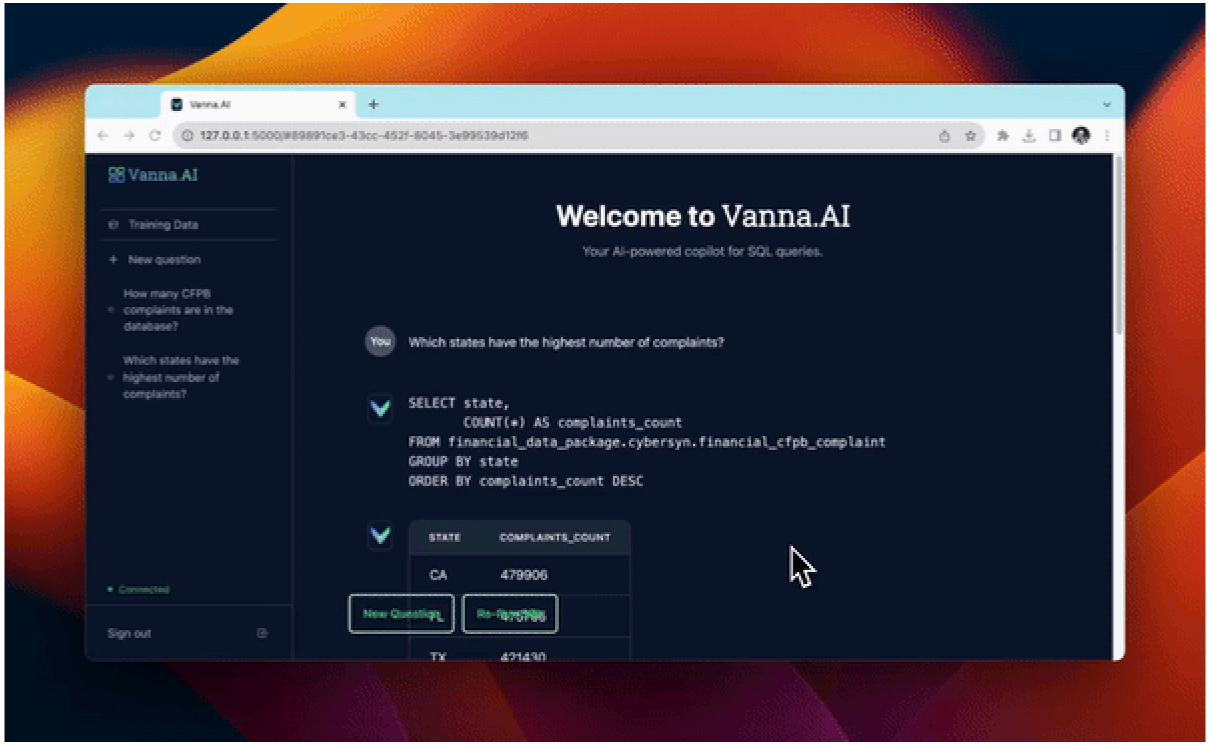
```
test running on 'ducklover1' with 'gemma2' LLM took : 3543.22 sec
```

In [43]:

```python
from datetime import datetime
print(datetime.now())
```

```
2024-07-21 22:01:33.690666
```

Launch the User Interface



Welcome to Vanna.AI

Your AI-powered copilot for SQL queries.

How many CFPB complaints are in the database?

Which states have the highest number of complaints?

Which states have the highest number of complaints?

```
SELECT state,
       COUNT(*) AS complaints_count
FROM financial_data_package.cybersyn.financial_cfpb_complaint
GROUP BY state
ORDER BY complaints_count DESC
```

STATE	COMPLAINTS_COUNT
CA	479906
FL	[illegible]
TX	421430

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