Generating SQL for SQLite using Ollama, ChromaDB

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

Which LLM do you want to use?

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

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

- Vanna Hosted Vector DB (Recommended)
  Use Vanna.AIs hosted vector database (pgvector) for free. This is usable across machines with no additional setup.
- [Selected] ChromaDB
  Use ChromaDBs open-source vector database for free locally. No additional setup is necessary -- all database files will be created and stored locally.
- Marqo
  Use Marqo locally for free. Requires additional setup. Or use their hosted option.
- Other VectorDB

Use any other vector database. Requires additional setup.

Setup

!pwd!pip install vanna!pip install 'vanna[chromadb]'!pip install ollama!pip show vanna # 0.5.5, 0.2.1!pip show ollama # 0.2.0

In [1]:
```
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)

config = {
    'model': 'llama3.1'  # 'mistral' # "starcoder2"
}
vn = MyVanna(config=config)
```

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 [3]:
```
import os.path
import re
from time import time
```

In [4]:
```
# file_db = "./db/gpt3sql.sqlite"

file_db = "~/Downloads/chinook.sqlite"
file_db = os.path.abspath(os.path.expanduser(file_db))
vn.connect_to_sqlite(file_db)
```

In [5]:
```
vn.run_sql_is_set
```

Out[5]:
```
True
```

In [6]:
```
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 [7]:
```
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 [8]:
```
if True:
    remove_collections()
```

Training

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

In [9]:
```
# show training data
training_data = vn.get_training_data()
training_data
```

Out[9]:

id	question	content	training_data_type

In [10]:
```
df_ddl = vn.run_sql("SELECT type, sql FROM sqlite_master WHERE sql is not null")
```

In [11]:
```
df_ddl
```

Out[11]:

	type	sql
0	table	CREATE TABLE [Album]\n(\n [AlbumId] INTEGER...
1	table	CREATE TABLE [Artist]\n(\n [ArtistId] INTEG...
2	table	CREATE TABLE [Customer]\n(\n [CustomerId] I...
3	table	CREATE TABLE [Employee]\n(\n [EmployeeId] I...
4	table	CREATE TABLE [Genre]\n(\n [GenreId] INTEGER...
5	table	CREATE TABLE [Invoice]\n(\n [InvoiceId] INT...
6	table	CREATE TABLE [InvoiceLine]\n(\n [InvoiceLin...
7	table	CREATE TABLE [MediaType]\n(\n [MediaTypeId]...
8	table	CREATE TABLE [Playlist]\n(\n [PlaylistId] I...
9	table	CREATE TABLE [PlaylistTrack]\n(\n [Playlist...
10	table	CREATE TABLE [Track]\n(\n [TrackId] INTEGER...
11	index	CREATE INDEX [IFK_AlbumArtistId] ON [Album] ([...
12	index	CREATE INDEX [IFK_CustomerSupportRepId] ON [Cu...
13	index	CREATE INDEX [IFK_EmployeeReportsTo] ON [Emplo...
14	index	CREATE INDEX [IFK_InvoiceCustomerId] ON [Invoi...
15	index	CREATE INDEX [IFK_InvoiceLineInvoiceId] ON [In...
16	index	CREATE INDEX [IFK_InvoiceLineTrackId] ON [Invo...
17	index	CREATE INDEX [IFK_PlaylistTrackTrackId] ON [Pl...
18	index	CREATE INDEX [IFK_TrackAlbumId] ON [Track] ([A...
19	index	CREATE INDEX [IFK_TrackGenreId] ON [Track] ([G...
20	index	CREATE INDEX [IFK_TrackMediaTypeId] ON [Track]...

In [12]:

```
for ddl in df_ddl['sql'].to_list():
    ddl = strip_brackets(ddl)
    vn.train(ddl=ddl)
```

```
Adding ddl: CREATE TABLE Album
(
    AlbumId INTEGER  NOT NULL,
    Title NVARCHAR(160)  NOT NULL,
    ArtistId INTEGER  NOT NULL,
    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),
    FOREIGN KEY (ArtistId) REFERENCES Artist (ArtistId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE Artist
(
    ArtistId INTEGER  NOT NULL,
    Name NVARCHAR(120),
    CONSTRAINT PK_Artist PRIMARY KEY  (ArtistId)
)
Adding ddl: CREATE TABLE Customer
(
    CustomerId INTEGER  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,
    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),
    FOREIGN KEY (SupportRepId) REFERENCES Employee (EmployeeId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE Employee
(
    EmployeeId INTEGER  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),
    CONSTRAINT PK_Employee PRIMARY KEY  (EmployeeId),
    FOREIGN KEY (ReportsTo) REFERENCES Employee (EmployeeId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE Genre
(
    GenreId INTEGER  NOT NULL,
    Name NVARCHAR(120),
    CONSTRAINT PK_Genre PRIMARY KEY  (GenreId)
)
Adding ddl: CREATE TABLE Invoice
(
    InvoiceId INTEGER  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,
    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),
    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE InvoiceLine
(
    InvoiceLineId INTEGER  NOT NULL,
    InvoiceId INTEGER  NOT NULL,
    TrackId INTEGER  NOT NULL,
    UnitPrice NUMERIC(10,2)  NOT NULL,
    Quantity INTEGER  NOT NULL,
    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),
    FOREIGN KEY (InvoiceId) REFERENCES Invoice (InvoiceId)
                ON DELETE NO ACTION ON UPDATE NO ACTION,
    FOREIGN KEY (TrackId) REFERENCES Track (TrackId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE MediaType
(
    MediaTypeId INTEGER  NOT NULL,
    Name NVARCHAR(120),
    CONSTRAINT PK_MediaType PRIMARY KEY  (MediaTypeId)
)
Adding ddl: CREATE TABLE Playlist
(
    PlaylistId INTEGER  NOT NULL,
    Name NVARCHAR(120),
    CONSTRAINT PK_Playlist PRIMARY KEY  (PlaylistId)
)
Adding ddl: CREATE TABLE PlaylistTrack
(
    PlaylistId INTEGER  NOT NULL,
    TrackId INTEGER  NOT NULL,
    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),
    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId)
                ON DELETE NO ACTION ON UPDATE NO ACTION,
    FOREIGN KEY (TrackId) REFERENCES Track (TrackId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE Track
(
    TrackId INTEGER  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,
    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),
    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId)
                ON DELETE NO ACTION ON UPDATE NO ACTION,
    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId)
```

```
                ON DELETE NO ACTION ON UPDATE NO ACTION,
    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaTypeId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE INDEX IFK_AlbumArtistId ON Album (ArtistId)
Adding ddl: CREATE INDEX IFK_CustomerSupportRepId ON Customer (SupportRepId)
Adding ddl: CREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)
Adding ddl: CREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)
Adding ddl: CREATE INDEX IFK_InvoiceLineInvoiceId ON InvoiceLine (InvoiceId)
Adding ddl: CREATE INDEX IFK_InvoiceLineTrackId ON InvoiceLine (TrackId)
Adding ddl: CREATE INDEX IFK_PlaylistTrackTrackId ON PlaylistTrack (TrackId)
Adding ddl: CREATE INDEX IFK_TrackAlbumId ON Track (AlbumId)
Adding ddl: CREATE INDEX IFK_TrackGenreId ON Track (GenreId)
Adding ddl: CREATE INDEX IFK_TrackMediaTypeId ON Track (MediaTypeId)
```

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 [13]:
```
ts_start = time()
```

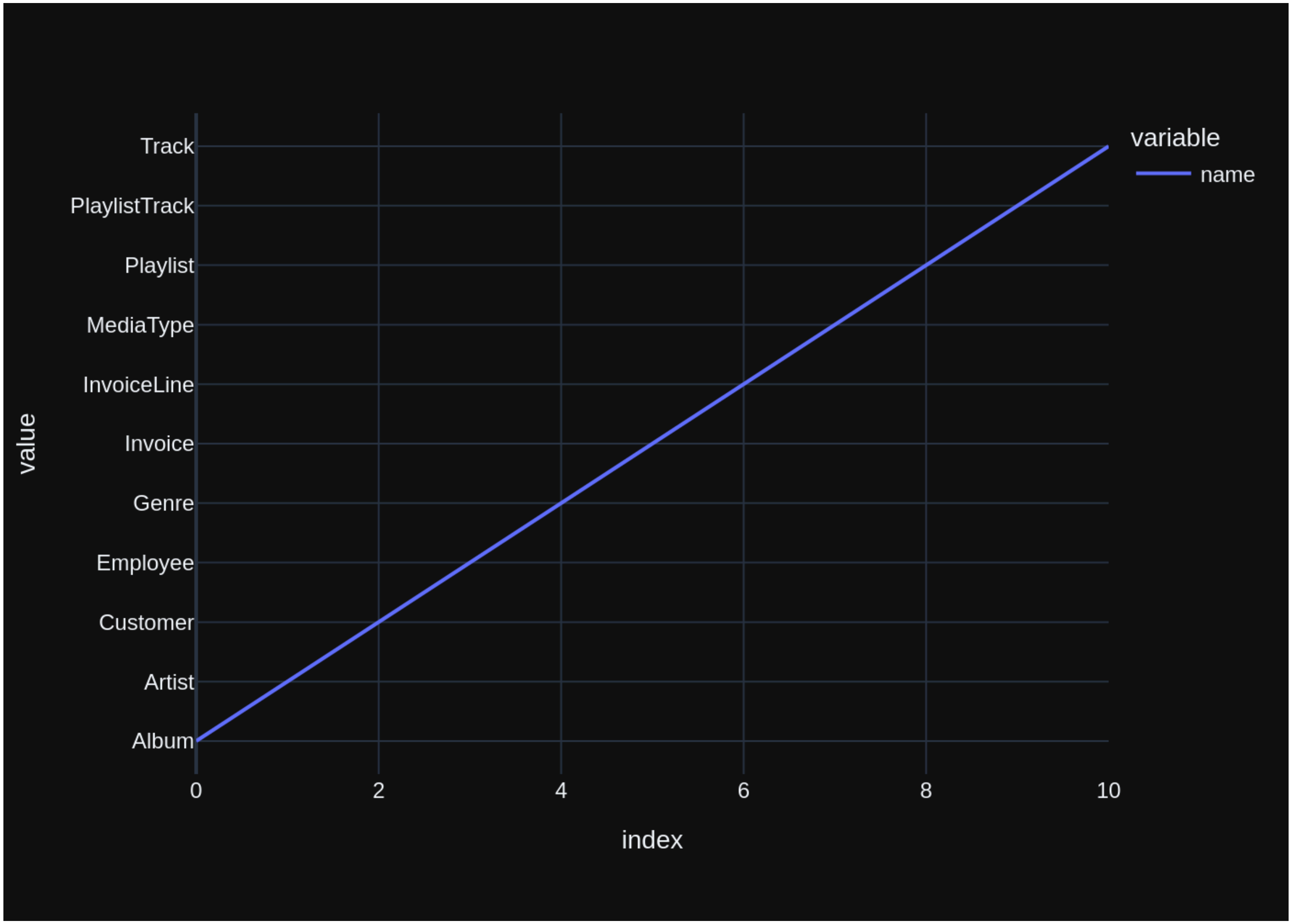
In [39]:
```
vn.ask(question="What are the table names in the database")
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name
NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n
Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,
2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album
(AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (Genre
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (Med
iaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Playlist\n(\n    PlaylistId INTE
GER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Playlist PRIMARY KEY  (PlaylistId)\n)\n\nCREATE
TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  N
OT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (Ar
tistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE MediaType\n(\n    MediaTypeId INTE
GER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_MediaType PRIMARY KEY  (MediaTypeId)\n)\n\nCREAT
E TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    InvoiceId INTEGER  NOT NULL,\n    TrackId
INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quantity INTEGER  NOT NULL,\n    CONSTRAINT
PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (InvoiceId) REFERENCES Invoice (InvoiceId) \n
\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Genre\n(\n    GenreId INTEGER  NOT NULL,\n    Name
NVARCHAR(120),\n    CONSTRAINT PK_Genre PRIMARY KEY  (GenreId)\n)\n\nCREATE TABLE PlaylistTrack\n(\n    Pla
ylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (P
laylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTI
ON ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON
UPDATE NO ACTION\n)\n\nCREATE TABLE Artist\n(\n    ArtistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n
CONSTRAINT PK_Artist PRIMARY KEY  (ArtistId)\n)\n\nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NUL
L,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(7
0),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    B
illingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY
(InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UP
DATE NO ACTION\n)\n\nCREATE TABLE Employee\n(\n    EmployeeId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)
NOT NULL,\n    FirstName NVARCHAR(20)  NOT NULL,\n    Title NVARCHAR(30),\n    ReportsTo INTEGER,\n    Birt
hDate DATETIME,\n    HireDate DATETIME,\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARC
HAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHA
R(24),\n    Email NVARCHAR(60),\n    CONSTRAINT PK_Employee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (Re
portsTo) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response G
uidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explana
tions for the question. \n2. If the provided context is almost sufficient but requires knowledge of a speci
fic string in a particular column, please generate an intermediate SQL query to find the distinct strings i
n that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is ins
ufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If th
e question has been asked and answered before, please repeat the answer exactly as it was given before.
\n"}, {'role': 'user', 'content': '  \n    List all albums and their corresponding artist names  \n'}, {'ro
```

```
le': 'assistant', 'content': 'SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId
= ART.ArtistId'}, {'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.TrackId) AS NumberOfTracks \nFROM Genre G
\nJOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name'}, {'role': 'user', 'content': 'what are the top
5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(CustomerI
d) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5'}, {'rol
e': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assistant', 'content':
'SELECT COUNT(*) FROM customer'}, {'role': 'user', 'content': "  \n    List all employees and their reporti
ng manager's name (if any):\n"}, {'role': 'assistant', 'content': "SELECT E.FirstName + ' ' + E.LastName AS
EmployeeName,\n       COALESCE(R.FirstName + ' ' + R.LastName, '--') AS ManagerName\nFROM Employee E\nLEFT
JOIN Employee R ON E.ReportsTo = R.EmployeeId"}, {'role': 'user', 'content': '  \n     Identify artists who
have albums with tracks appearing in multiple genres:\n'}, {'role': 'assistant', 'content': 'SELECT A.Name,
COUNT(DISTINCT T.GenreId) AS NumberOfGenres\nFROM Artist A\nJOIN Album AS ALBUM ON A.ArtistId = ALBUM.Artis
tId\nJOIN Track T ON ALBUM.AlbumId = T.AlbumId\nGROUP BY A.Name\nHAVING COUNT(DISTINCT T.GenreId) > 1'},
{'role': 'user', 'content': '  \n    Get the total number of invoices for each customer\n'}, {'role': 'assi
stant', 'content': 'SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoi
ce I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId'}, {'role': 'user', 'content': '  \n     Find t
he customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT C.Email FROM Customer C\n
JOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.CustomerId, C.FirstName, C.LastName, C.Email\nORD
ER BY COUNT(I.InvoiceId) DESC\nLIMIT 1'}, {'role': 'user', 'content': '  \n    Find the total number of inv
oices per country:\n'}, {'role': 'assistant', 'content': 'SELECT COUNT(DISTINCT InvoiceId), BillingCountry
\nFROM Invoice'}, {'role': 'user', 'content': '  \n     List all customers from Canada and their email addr
esses:\n'}, {'role': 'assistant', 'content': "SELECT Country, Email FROM Customer WHERE Country = 'Canad
a'"}, {'role': 'user', 'content': 'What are the table names in the database'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(2
00)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Compose
r NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT
NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumI
d) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n
\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaType
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Playlist\n(\n    PlaylistId INTEGER  N
OT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Playlist PRIMARY KEY  (PlaylistId)\n)\n\nCREATE TABLE
Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NUL
L,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (ArtistI
d) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE MediaType\n(\n    MediaTypeId INTEGER
NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_MediaType PRIMARY KEY  (MediaTypeId)\n)\n\nCREATE TAB
LE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    InvoiceId INTEGER  NOT NULL,\n    TrackId INTE
GER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quantity INTEGER  NOT NULL,\n    CONSTRAINT PK_
InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (InvoiceId) REFERENCES Invoice (InvoiceId) \n\t
\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON D
ELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Genre\n(\n    GenreId INTEGER  NOT NULL,\n    Name N
VARCHAR(120),\n    CONSTRAINT PK_Genre PRIMARY KEY  (GenreId)\n)\n\nCREATE TABLE PlaylistTrack\n(\n    Play
listId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (Pl
aylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTIO
N ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON U
PDATE NO ACTION\n)\n\nCREATE TABLE Artist\n(\n    ArtistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n
CONSTRAINT PK_Artist PRIMARY KEY  (ArtistId)\n)\n\nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NUL
L,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(7
0),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    B
illingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY
(InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UP
DATE NO ACTION\n)\n\nCREATE TABLE Employee\n(\n    EmployeeId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)
NOT NULL,\n    FirstName NVARCHAR(20)  NOT NULL,\n    Title NVARCHAR(30),\n    ReportsTo INTEGER,\n    Birt
hDate DATETIME,\n    HireDate DATETIME,\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARC
HAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHA
R(24),\n    Email NVARCHAR(60),\n    CONSTRAINT PK_Employee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (Re
portsTo) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response G
uidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explana
tions for the question. \n2. If the provided context is almost sufficient but requires knowledge of a speci
fic string in a particular column, please generate an intermediate SQL query to find the distinct strings i
n that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is ins
ufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If th
e question has been asked and answered before, please repeat the answer exactly as it was given before.
\n"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist names  \n"}, {"ro
le": "assistant", "content": "SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId
= ART.ArtistId"}, {"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.TrackId) AS NumberOfTracks \nFROM Genre G
\nJOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name"}, {"role": "user", "content": "what are the top
5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(CustomerI
d) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5"}, {"rol
e": "user", "content": "How many records are in table called customer"}, {"role": "assistant", "content":
"SELECT COUNT(*) FROM customer"}, {"role": "user", "content": "  \n    List all employees and their reporti
ng manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT E.FirstName + ' ' + E.LastName AS
EmployeeName,\n       COALESCE(R.FirstName + ' ' + R.LastName, '--') AS ManagerName\nFROM Employee E\nLEFT
JOIN Employee R ON E.ReportsTo = R.EmployeeId"}, {"role": "user", "content": "  \n     Identify artists who
have albums with tracks appearing in multiple genres:\n"}, {"role": "assistant", "content": "SELECT A.Name,
COUNT(DISTINCT T.GenreId) AS NumberOfGenres\nFROM Artist A\nJOIN Album AS ALBUM ON A.ArtistId = ALBUM.Artis
```

```
tId\nJOIN Track T ON ALBUM.AlbumId = T.AlbumId\nGROUP BY A.Name\nHAVING COUNT(DISTINCT T.GenreId) > 1"},
{"role": "user", "content": "  \n    Get the total number of invoices for each customer\n"}, {"role": "assi
stant", "content": "SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoi
ce I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId"}, {"role": "user", "content": "  \n     Find t
he customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT C.Email FROM Customer C\n
JOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.CustomerId, C.FirstName, C.LastName, C.Email\nORD
ER BY COUNT(I.InvoiceId) DESC\nLIMIT 1"}, {"role": "user", "content": "  \n    Find the total number of inv
oices per country:\n"}, {"role": "assistant", "content": "SELECT COUNT(DISTINCT InvoiceId), BillingCountry
\nFROM Invoice"}, {"role": "user", "content": "  \n     List all customers from Canada and their email addr
esses:\n"}, {"role": "assistant", "content": "SELECT Country, Email FROM Customer WHERE Country = 'Canad
a'"}, {"role": "user", "content": "What are the table names in the database"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T11:16:47.564154264Z', 'message': {'role': 'assistan
t', 'content': "```sql\nSELECT name FROM sqlite_master WHERE type='table';\n```"}, 'done_reason': 'stop',
'done': True, 'total_duration': 75285814862, 'load_duration': 2310359314, 'prompt_eval_count': 2041, 'promp
t_eval_duration': 70624591000, 'eval_count': 15, 'eval_duration': 2290937000}
LLM Response: ```sql
SELECT name FROM sqlite_master WHERE type='table';
```
Info: Output from LLM: ```sql
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           Album
1          Artist
2        Customer
3        Employee
4           Genre
5         Invoice
6     InvoiceLine
7       MediaType
8        Playlist
9   PlaylistTrack
10          Track
Info: Ollama parameters:
model=llama3.1: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 answers the question the user asked: 'What are the table names in the database'\n\nThe DataFrame was p
roduced using this query: SELECT name FROM sqlite_master WHERE type='table'\n\nThe following is information
about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n name    object\ndtype: object"},
{"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe?
Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an I
ndicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T11:17:18.956710777Z', 'message': {'role': 'assistan
t', 'content': '```python\nimport plotly.express as px\n\nfig = px.bar(df, x=\'name\', y=0)\nfig.update_lay
out(title=\'Table Names in Database\',\n                  xaxis_title=\'Table Name\',\n                  ya
xis_title=\'Count\')\n\nif len(df) == 1:\n    fig.data[0].type = \'indicator\'\n    fig.update_layout(\n
title_text="Table Names in Database",\n        annotations=[dict(x=df[\'name\'].values[0], y=0, text=f"1",
showarrow=False, xref=\'x\', yref=\'y\')])\n    \nfig.show()\n```'}, 'done_reason': 'stop', 'done': True,
'total_duration': 31278086360, 'load_duration': 80562253, 'prompt_eval_count': 228, 'prompt_eval_duration':
7469543000, 'eval_count': 125, 'eval_duration': 23670058000}
```



variable
name
value
Track
PlaylistTrack
Playlist
MediaType
InvoiceLine
Invoice
Genre
Employee
Customer
Artist
Album
index
0
2
4
6
8
10

```
Out[39]: ("SELECT name FROM sqlite_master WHERE type='table'",
                       name
          0           Album
          1          Artist
          2        Customer
          3        Employee
          4           Genre
          5         Invoice
          6     InvoiceLine
          7       MediaType
          8        Playlist
          9   PlaylistTrack
          10          Track,
          Figure({
              'data': [{'hovertemplate': 'variable=name<br>index=%{x}<br>value=%{y}<extra></extra>',
                        'legendgroup': 'name',
                        'line': {'color': '#636efa', 'dash': 'solid'},
                        'marker': {'symbol': 'circle'},
                        'mode': 'lines',
                        'name': 'name',
                        'orientation': 'v',
                        'showlegend': True,
                        'type': 'scatter',
                        'x': array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10]),
                        'xaxis': 'x',
                        'y': array(['Album', 'Artist', 'Customer', 'Employee', 'Genre', 'Invoice',
                                    'InvoiceLine', 'MediaType', 'Playlist', 'PlaylistTrack', 'Track'],
                                   dtype=object),
                        'yaxis': 'y'}],
              'layout': {'legend': {'title': {'text': 'variable'}, 'tracegroupgap': 0},
                         'margin': {'t': 60},
                         'template': '...',
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'index'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'value'}}}
          }))
```

```
In [14]: vn.ask(question="Show me a list of tables in the SQLite database")
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE TABLE Playlist\n(\n    PlaylistId INTEGER  NOT NULL,\n
Name NVARCHAR(120),\n    CONSTRAINT PK_Playlist PRIMARY KEY  (PlaylistId)\n)\n\nCREATE TABLE InvoiceLine\n
(\n    InvoiceLineId INTEGER  NOT NULL,\n    InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n
UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY
KEY  (InvoiceLineId),\n    FOREIGN KEY (InvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPD
ATE NO ACTION\n)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER
NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistI
d) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackI
d) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Track\n(\n
TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGE
R  NOT NULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    B
ytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n
FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIG
N KEY (GenreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY
(MediaTypeId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE
TABLE MediaType\n(\n    MediaTypeId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_MediaTyp
e PRIMARY KEY  (MediaTypeId)\n)\n\nCREATE TABLE Artist\n(\n    ArtistId INTEGER  NOT NULL,\n    Name NVARCH
AR(120),\n    CONSTRAINT PK_Artist PRIMARY KEY  (ArtistId)\n)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER
NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRI
MARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON
UPDATE NO ACTION\n)\n\nCREATE TABLE Genre\n(\n    GenreId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n
CONSTRAINT PK_Genre PRIMARY KEY  (GenreId)\n)\n\nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NUL
L,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(7
0),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    B
illingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY
(InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UP
DATE NO ACTION\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\n\n===Response Guidelines
\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for
the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string
in a particular column, please generate an intermediate SQL query to find the distinct strings in that colu
mn. 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': 'Show me a list of tables in the SQLite database'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE TABLE Playlist\n(\n    PlaylistId INTEGER  NOT NULL,\n    Name NVAR
CHAR(120),\n    CONSTRAINT PK_Playlist PRIMARY KEY  (PlaylistId)\n)\n\nCREATE TABLE InvoiceLine\n(\n    Inv
oiceLineId INTEGER  NOT NULL,\n    InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPr
ice NUMERIC(10,2)  NOT NULL,\n    Quantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY
(InvoiceLineId),\n    FOREIGN KEY (InvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON U
PDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE
NO ACTION\n)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT
NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) RE
FERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) RE
FERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Track\n(\n    Tra
ckId INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER
NOT NULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Byte
s INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FO
REIGN KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN
KEY (GenreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (M
ediaTypeId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE T
ABLE MediaType\n(\n    MediaTypeId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_MediaType
PRIMARY KEY  (MediaTypeId)\n)\n\nCREATE TABLE Artist\n(\n    ArtistId INTEGER  NOT NULL,\n    Name NVARCHAR
(120),\n    CONSTRAINT PK_Artist PRIMARY KEY  (ArtistId)\n)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER
NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRI
MARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON
UPDATE NO ACTION\n)\n\nCREATE TABLE Genre\n(\n    GenreId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n
CONSTRAINT PK_Genre PRIMARY KEY  (GenreId)\n)\n\nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NUL
L,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(7
0),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    B
illingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY
(InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UP
DATE NO ACTION\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\n\n===Response Guidelines
\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for
the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string
in a particular column, please generate an intermediate SQL query to find the distinct strings in that colu
mn. 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": "Show me a list of tables in the SQLite database"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:14:46.615585176Z', 'message': {'role': 'assistan
t', 'content': 'Here is the list of tables from the provided context:\n\n1. Playlist\n2. InvoiceLine\n3. Pl
aylistTrack\n4. Track\n5. MediaType\n6. Artist\n7. Album\n8. Genre\n9. Invoice\n10. Employee \n\nNote that
there is no `Customer` table in the provided context, which might be relevant for some questions.'}, 'done_
reason': 'stop', 'done': True, 'total_duration': 84294634266, 'load_duration': 12459073085, 'prompt_eval_co
unt': 1765, 'prompt_eval_duration': 60020937000, 'eval_count': 76, 'eval_duration': 11759051000}
LLM Response: Here is the list of tables from the provided context:

1. Playlist
2. InvoiceLine
3. PlaylistTrack
4. Track
5. MediaType
6. Artist
7. Album
8. Genre
9. Invoice
10. Employee

Note that there is no `Customer` table in the provided context, which might be relevant for some questions.
Here is the list of tables from the provided context:

1. Playlist
2. InvoiceLine
3. PlaylistTrack
4. Track
5. MediaType
6. Artist
7. Album
8. Genre
9. Invoice
10. Employee

Note that there is no `Customer` table in the provided context, which might be relevant for some questions.
Couldn't run sql:  Execution failed on sql 'Here is the list of tables from the provided context:

1. Playlist
2. InvoiceLine
3. PlaylistTrack
4. Track
5. MediaType
6. Artist
7. Album
8. Genre
9. Invoice
10. Employee
```

```
Note that there is no `Customer` table in the provided context, which might be relevant for some question
s.': near "Here": syntax error
```

In [15]:
```
vn.ask(question="which tables store order information")
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NUL
L,\n    InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NUL
L,\n    Quantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREI
GN KEY (InvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREI
GN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABL
E PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_Pl
aylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (Playlist
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n
\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n
Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGE
R,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUM
ERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCE
S Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genr
e (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaT
ype (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Employee\n(\n    Employe
eId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    FirstName NVARCHAR(20)  NOT NULL,\n    Ti
tle NVARCHAR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIME,\n    HireDate DATETIME,\n    Address NVA
RCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVAR
CHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60),\n    CONSTRAINT PK_Emplo
yee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Employee (EmployeeId) \n\t\tON DELET
E NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n    Custom
erId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(70),\n    Billing
City NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    BillingPostalCode
NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),\n
FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)
\n\nCREATE TABLE Playlist\n(\n    PlaylistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK
_Playlist PRIMARY KEY  (PlaylistId)\n)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title
NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumI
d),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\n)\n\nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    FirstName NVARCHAR(40)  NOT NULL,\n
LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Address NVARCHAR(70),\n    City NVARCHAR(4
0),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(2
4),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    SupportRepId INTEGER,\n    CONSTRAINT PK
_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId) REFERENCES Employee (EmployeeId) \n\t
\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE MediaType\n(\n    MediaTypeId INTEGER  NOT NUL
L,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_MediaType PRIMARY KEY  (MediaTypeId)\n)\n\nCREATE TABLE Arti
st\n(\n    ArtistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Artist PRIMARY KEY  (Art
istId)\n)\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL
query without any explanations for the question. \n2. If the provided context is almost sufficient but requ
ires knowledge of a specific string in a particular column, please generate an intermediate SQL query to fi
```

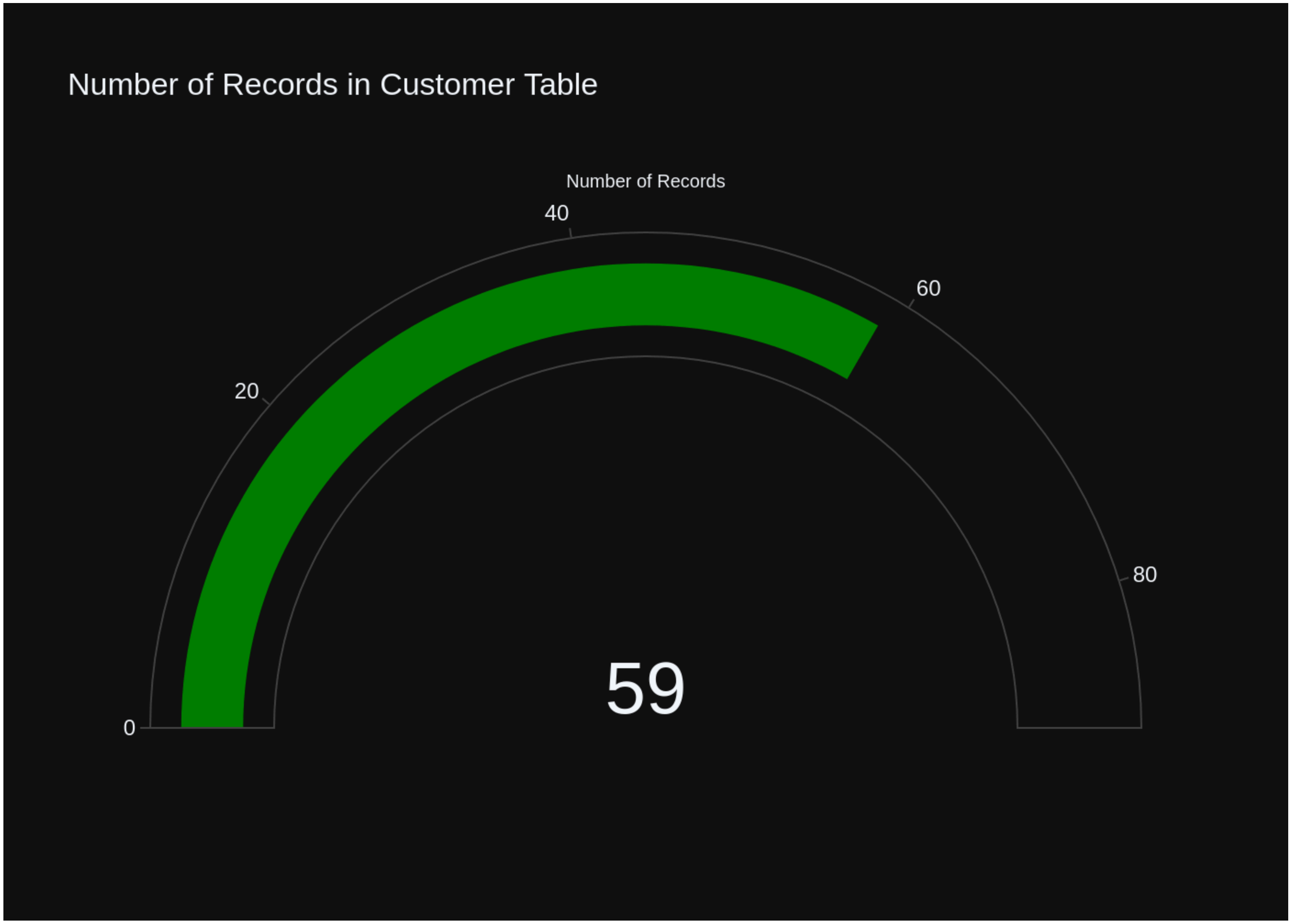
```
nd the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If th
e provided context is insufficient, please explain why it can't be generated. \n4. Please use the most rele
vant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as
it was given before. \n"}, {'role': 'user', 'content': 'which tables store order information'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    Inv
oiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quan
tity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (Invo
iceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (Trac
kId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE PlaylistTr
ack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack
PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE
NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHA
R(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Comp
oser NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)
NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (Alb
umId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId)
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaTy
peId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Employee\n(\n    EmployeeId INTEGER
NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    FirstName NVARCHAR(20)  NOT NULL,\n    Title NVARCHAR
(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIME,\n    HireDate DATETIME,\n    Address NVARCHAR(70),\n
City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    P
hone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60),\n    CONSTRAINT PK_Employee PRIMARY KEY
(EmployeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UP
DATE NO ACTION\n)\n\nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT
NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(4
0),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    BillingPostalCode NVARCHAR(1
0),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),\n    FOREIGN K
EY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE
TABLE Playlist\n(\n    PlaylistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Playlist P
RIMARY KEY  (PlaylistId)\n)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(16
0)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIG
N KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TA
BLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    FirstName NVARCHAR(40)  NOT NULL,\n    LastName NVA
RCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    Sta
te NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax
NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    SupportRepId INTEGER,\n    CONSTRAINT PK_Customer PRI
MARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE MediaType\n(\n    MediaTypeId INTEGER  NOT NULL,\n    Name NV
ARCHAR(120),\n    CONSTRAINT PK_MediaType PRIMARY KEY  (MediaTypeId)\n)\n\nCREATE TABLE Artist\n(\n    Arti
stId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Artist PRIMARY KEY  (ArtistId)\n)\n\n==
=Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without
any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge
of a specific string in a particular column, please generate an intermediate SQL query to find the distinct
strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided conte
xt 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 b
efore. \n"}, {"role": "user", "content": "which tables store order information"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:15:29.282710605Z', 'message': {'role': 'assistan
t', 'content': 'In a typical e-commerce or online ordering system, the following tables can store order inf
ormation:\n\n1. **Orders Table**: This is the primary table that stores all order-related data, such as:\n
\t* Order ID (primary key)\n\t* Customer ID (foreign key referencing Customers table)\n\t* Order date\n\t*
Total cost\n\t* Status (e.g., "pending", "shipped", "delivered")\n2. **Order Items Table**: This table stor
es individual items in an order, such as:\n\t* Order ID (foreign key referencing Orders table)\n\t* Product
ID (foreign key referencing Products table)\n\t* Quantity\n\t* Price\n3. **Customers Table**: While not dir
ectly related to orders, this table can store customer information that is referenced in the Orders table,
such as:\n\t* Customer ID (primary key)\n\t* Name\n\t* Email\n\t* Address\n4. **Payments Table** (optiona
l): If your system allows for multiple payment methods or needs to track payments separately from orders, y
ou might have a Payments table with columns like:\n\t* Payment ID (primary key)\n\t* Order ID (foreign key
referencing Orders table)\n\t* Payment method (e.g., credit card, PayPal)\n\t* Payment date\n\t* Amount\n\n
These are the most common tables related to order storage. The specific schema will depend on your applicat
ion\'s requirements and architecture.'}, 'done_reason': 'stop', 'done': True, 'total_duration': 4261378975
7, 'load_duration': 30142591, 'prompt_eval_count': 14, 'prompt_eval_duration': 506449000, 'eval_count': 28
7, 'eval_duration': 42016798000}
LLM Response: In a typical e-commerce or online ordering system, the following tables can store order infor
mation:

1. **Orders Table**: This is the primary table that stores all order-related data, such as:
        * Order ID (primary key)
        * Customer ID (foreign key referencing Customers table)
        * Order date
        * Total cost
        * Status (e.g., "pending", "shipped", "delivered")
2. **Order Items Table**: This table stores individual items in an order, such as:
        * Order ID (foreign key referencing Orders table)
        * Product ID (foreign key referencing Products table)
        * Quantity
        * Price
3. **Customers Table**: While not directly related to orders, this table can store customer information tha
t is referenced in the Orders table, such as:
        * Customer ID (primary key)
        * Name
        * Email
        * Address
4. **Payments Table** (optional): If your system allows for multiple payment methods or needs to track paym
ents separately from orders, you might have a Payments table with columns like:
        * Payment ID (primary key)
        * Order ID (foreign key referencing Orders table)
        * Payment method (e.g., credit card, PayPal)
        * Payment date
        * Amount

These are the most common tables related to order storage. The specific schema will depend on your applicat
ion's requirements and architecture.
In a typical e-commerce or online ordering system, the following tables can store order information:

1. **Orders Table**: This is the primary table that stores all order-related data, such as:
        * Order ID (primary key)
        * Customer ID (foreign key referencing Customers table)
        * Order date
        * Total cost
        * Status (e.g., "pending", "shipped", "delivered")
2. **Order Items Table**: This table stores individual items in an order, such as:
        * Order ID (foreign key referencing Orders table)
        * Product ID (foreign key referencing Products table)
        * Quantity
        * Price
3. **Customers Table**: While not directly related to orders, this table can store customer information tha
t is referenced in the Orders table, such as:
        * Customer ID (primary key)
        * Name
        * Email
        * Address
4. **Payments Table** (optional): If your system allows for multiple payment methods or needs to track paym
ents separately from orders, you might have a Payments table with columns like:
        * Payment ID (primary key)
        * Order ID (foreign key referencing Orders table)
        * Payment method (e.g., credit card, PayPal)
        * Payment date
        * Amount

These are the most common tables related to order storage. The specific schema will depend on your applicat
ion's requirements and architecture.
Couldn't run sql:  Execution failed on sql 'In a typical e-commerce or online ordering system, the followin
g tables can store order information:

1. **Orders Table**: This is the primary table that stores all order-related data, such as:
        * Order ID (primary key)
        * Customer ID (foreign key referencing Customers table)
        * Order date
        * Total cost
        * Status (e.g., "pending", "shipped", "delivered")
2. **Order Items Table**: This table stores individual items in an order, such as:
        * Order ID (foreign key referencing Orders table)
        * Product ID (foreign key referencing Products table)
        * Quantity
        * Price
3. **Customers Table**: While not directly related to orders, this table can store customer information tha
t is referenced in the Orders table, such as:
        * Customer ID (primary key)
        * Name
        * Email
        * Address
4. **Payments Table** (optional): If your system allows for multiple payment methods or needs to track paym
ents separately from orders, you might have a Payments table with columns like:
        * Payment ID (primary key)
        * Order ID (foreign key referencing Orders table)
        * Payment method (e.g., credit card, PayPal)
        * Payment date
        * Amount

These are the most common tables related to order storage. The specific schema will depend on your applicat
ion's requirements and architecture.': near "In": syntax error
```

In [16]:
```
vn.ask(question="How many records are in table called customer")
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n
FirstName NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Add
ress NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalC
ode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    S
upportRepId INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId)
REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Invoice\n
(\n    InvoiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NUL
L,\n    BillingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    Bi
llingCountry NVARCHAR(40),\n    BillingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CO
NSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerI
d) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEG
ER  NOT NULL,\n    InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,
2)  NOT NULL,\n    Quantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineI
d),\n    FOREIGN KEY (InvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTI
ON,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)
\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId
INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES
Artist (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON
Invoice (CustomerId)\n\nCREATE TABLE Employee\n(\n    EmployeeId INTEGER  NOT NULL,\n    LastName NVARCHAR
(20)  NOT NULL,\n    FirstName NVARCHAR(20)  NOT NULL,\n    Title NVARCHAR(30),\n    ReportsTo INTEGER,\n
BirthDate DATETIME,\n    HireDate DATETIME,\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State N
VARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVA
RCHAR(24),\n    Email NVARCHAR(60),\n    CONSTRAINT PK_Employee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY
(ReportsTo) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TAB
LE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    M
ediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER
NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY
(TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n
FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON Customer (SupportRepId)\n\nCREATE TABLE Playlist\n(\n    Pl
aylistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Playlist PRIMARY KEY  (PlaylistId)
\n)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n
CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Pl
aylist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Tr
ack (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response Guidelines \n1. If the provid
ed context is sufficient, please generate a valid SQL query without any explanations for the question. \n2.
If the provided context is almost sufficient but requires knowledge of a specific string in a particular co
lumn, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the qu
ery with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why
```

```
it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and
answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content':
'How many records are in table called customer'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    FirstName
NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Address NVARC
HAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCH
AR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    SupportRepI
d INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId) REFERENCE
S Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Invoice\n(\n    In
voiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    Bil
lingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountr
y NVARCHAR(40),\n    BillingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK
_Invoice PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NUL
L,\n    InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NUL
L,\n    Quantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREI
GN KEY (InvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREI
GN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABL
E Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT N
ULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (Artist
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (Cust
omerId)\n\nCREATE TABLE Employee\n(\n    EmployeeId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NUL
L,\n    FirstName NVARCHAR(20)  NOT NULL,\n    Title NVARCHAR(30),\n    ReportsTo INTEGER,\n    BirthDate D
ATETIME,\n    HireDate DATETIME,\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(4
0),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(2
4),\n    Email NVARCHAR(60),\n    CONSTRAINT PK_Employee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (Repor
tsTo) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Tra
ck\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTy
peId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT N
ULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (Tra
ckId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO
N,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n
FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON Customer (SupportRepId)\n\nCREATE TABLE Playlist\n(\n    Pl
aylistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Playlist PRIMARY KEY  (PlaylistId)
```

~~~
\n)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n
CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Pl
aylist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Tr
ack (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response Guidelines \n1. If the provid
ed context is sufficient, please generate a valid SQL query without any explanations for the question. \n2.
If the provided context is almost sufficient but requires knowledge of a specific string in a particular co
lumn, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the qu
ery with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why
it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and
answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content":
"How many records are in table called customer"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:15:30.892947124Z', 'message': {'role': 'assistan
t', 'content': 'SELECT COUNT(*) FROM customer;'}, 'done_reason': 'stop', 'done': True, 'total_duration': 15
35783464, 'load_duration': 20456929, 'prompt_eval_count': 17, 'prompt_eval_duration': 578976000, 'eval_coun
t': 7, 'eval_duration': 870474000}
LLM Response: SELECT COUNT(*) FROM customer;
Info: Output from LLM: SELECT COUNT(*) FROM customer;
Extracted SQL: SELECT COUNT(*) FROM customer
SELECT COUNT(*) FROM customer
   COUNT(*)
0        59
Info: Ollama parameters:
model=llama3.1: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 answers the question the user asked: 'How many records are in table called customer'\n\nThe DataFrame
was produced using this query: SELECT COUNT(*) FROM customer\n\nThe following is information about the resu
lting pandas DataFrame 'df': \nRunning df.dtypes gives:\n COUNT(*)    int64\ndtype: object"}, {"role": "use
r", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the d
ata is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Re
spond with only Python code. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:15:53.897310016Z', 'message': {'role': 'assistan
t', 'content': '```python\nimport plotly.graph_objects as go\n\nif len(df) == 1:\n    fig = go.Figure(go.In
dicator(\n        mode = \'gauge+number\',\n        value= df[\'COUNT(*)\'].iloc[0],\n        title=\'Numbe
r of Records\',\n        number = dict(font_size = 40)\n    ))\nelse:\n    fig = go.Figure(data=[go.Histogr
am(x=df[\'COUNT(*)\'])])\nfig.update_layout(title_text="Number of Records in Customer Table")\nfig.show()\n
```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 22800438352, 'load_duration': 83283428, 'prom
pt_eval_count': 224, 'prompt_eval_duration': 6924248000, 'eval_count': 107, 'eval_duration': 15736787000}
~~~


Number of Records in Customer Table
Number of Records
0
20
40
60
80
59

```
Out[16]: ('SELECT COUNT(*) FROM customer',
             COUNT(*)
          0        59,
          Figure({
              'data': [{'mode': 'gauge+number',
                        'number': {'font': {'size': 40}},
                        'title': {'text': 'Number of Records'},
                        'type': 'indicator',
                        'value': 59}],
              'layout': {'template': '...', 'title': {'text': 'Number of Records in Customer Table'}}
          }))
```

```
In [ ]:
```

```
In [17]: vn.ask(question="what are the top 5 countries that customers come from?")
```

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

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n
FirstName NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Add
ress NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalC
ode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    S
upportRepId INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId)
REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Invoice\n
(\n    InvoiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NUL
L,\n    BillingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    Bi
llingCountry NVARCHAR(40),\n    BillingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CO
NSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerI
d) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEG
ER  NOT NULL,\n    InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,
2)  NOT NULL,\n    Quantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineI
d),\n    FOREIGN KEY (InvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTI
ON,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)
\n\nCREATE TABLE Employee\n(\n    EmployeeId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n
FirstName NVARCHAR(20)  NOT NULL,\n    Title NVARCHAR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIM
E,\n    HireDate DATETIME,\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n
Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Em
ail NVARCHAR(60),\n    CONSTRAINT PK_Employee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFER
ENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE MediaType\n(\n
MediaTypeId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_MediaType PRIMARY KEY  (MediaTyp
eId)\n)\n\nCREATE TABLE Playlist\n(\n    PlaylistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTR
AINT PK_Playlist PRIMARY KEY  (PlaylistId)\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON Customer (Support
RepId)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NUL
L,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFER
ENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFER
ENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Track\n(\n    TrackI
d INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT
NULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes IN
TEGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIG
N KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY
(GenreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (Media
TypeId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE
Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NUL
L,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (ArtistI
d) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response Guidelines \n1. If the provided context
is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the pro
vided context is almost sufficient but requires knowledge of a specific string in a particular column, plea
se generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a
```

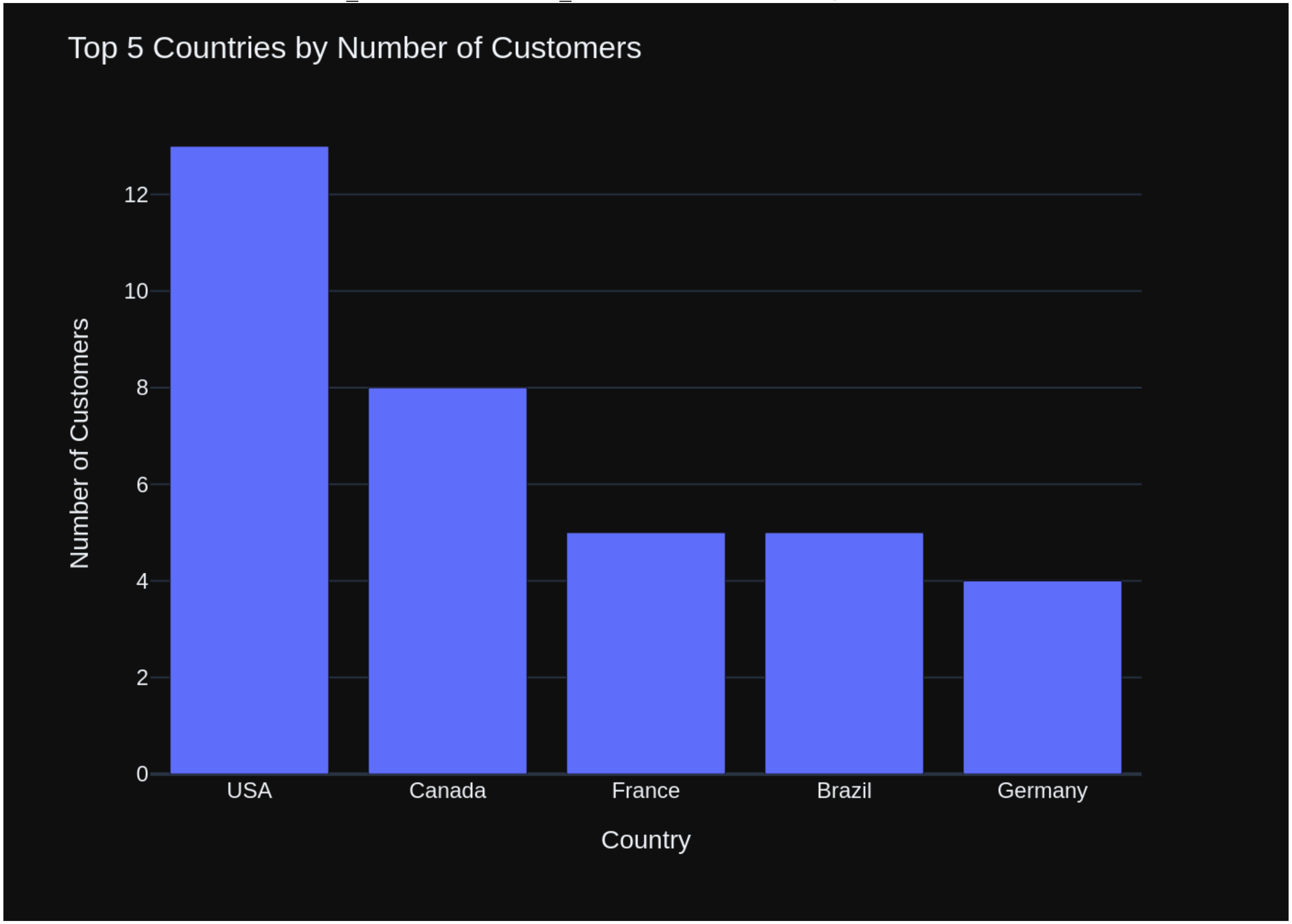
```
comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't b
e generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered b
efore, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': 'How many
records are in table called customer'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customer'},
{'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    FirstName
NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Address NVARC
HAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCH
AR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    SupportRepI
d INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId) REFERENCE
S Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Invoice\n(\n    In
voiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    Bil
lingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountr
y NVARCHAR(40),\n    BillingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK
_Invoice PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NUL
L,\n    InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NUL
L,\n    Quantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREI
GN KEY (InvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREI
GN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABL
E Employee\n(\n    EmployeeId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    FirstName NVARC
HAR(20)  NOT NULL,\n    Title NVARCHAR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIME,\n    HireDate
DATETIME,\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR
(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(6
0),\n    CONSTRAINT PK_Employee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Employee
(EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE MediaType\n(\n    MediaTypeId
INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_MediaType PRIMARY KEY  (MediaTypeId)\n)\n\nC
REATE TABLE Playlist\n(\n    PlaylistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Play
list PRIMARY KEY  (PlaylistId)\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON Customer (SupportRepId)\n\nCR
EATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRA
INT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist
(PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (Tr
ackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT
NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    Gen
reId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    Un
itPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumI
d) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REF
ERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFER
ENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Album\n(\n
AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n    CONSTR
AINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\tON DEL
ETE NO ACTION ON UPDATE NO ACTION\n)\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 i
s 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. \n
4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please r
epeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "How many records are in
table called customer"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customer"}, {"role": "use
r", "content": "what are the top 5 countries that customers come from?"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:16:47.086541576Z', 'message': {'role': 'assistan
t', 'content': '```sql\nSELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM Customer\nGROUP BY Cou
ntry\nORDER BY NumberOfCustomers DESC\nLIMIT 5;\n```\n\nThis query will return the top 5 countries by numbe
r of customers. The `GROUP BY` clause groups the customers by country, and the `COUNT` function counts the
number of customers in each group. The `ORDER BY` clause sorts the results in descending order ( highest co
unt first), and the `LIMIT` clause limits the output to the top 5 countries.'}, 'done_reason': 'stop', 'don
e': True, 'total_duration': 49506157340, 'load_duration': 15116167, 'prompt_eval_count': 1026, 'prompt_eval
_duration': 33069174000, 'eval_count': 108, 'eval_duration': 16361756000}
LLM Response: ```sql
SELECT Country, COUNT(CustomerId) AS NumberOfCustomers
FROM Customer
GROUP BY Country
ORDER BY NumberOfCustomers DESC
LIMIT 5;
```

This query will return the top 5 countries by number of customers. The `GROUP BY` clause groups the custome
rs by country, and the `COUNT` function counts the number of customers in each group. The `ORDER BY` clause
sorts the results in descending order (highest count first), and the `LIMIT` clause limits the output to t
he top 5 countries.
Info: Output from LLM: ```sql
SELECT Country, COUNT(CustomerId) AS NumberOfCustomers
FROM Customer
GROUP BY Country
ORDER BY NumberOfCustomers DESC
LIMIT 5;
```
```

```
```

This query will return the top 5 countries by number of customers. The `GROUP BY` clause groups the custome
rs by country, and the `COUNT` function counts the number of customers in each group. The `ORDER BY` clause
sorts the results in descending order (highest count first), and the `LIMIT` clause limits the output to t
he top 5 countries.
Extracted SQL: SELECT Country, COUNT(CustomerId) AS NumberOfCustomers
FROM Customer
GROUP BY Country
ORDER BY NumberOfCustomers DESC
LIMIT 5
SELECT Country, COUNT(CustomerId) AS NumberOfCustomers
FROM Customer
GROUP BY Country
ORDER BY NumberOfCustomers DESC
LIMIT 5
   Country NumberOfCustomers
0   USA   13
1   Canada   8
2   France   5
3   Brazil   5
4 Germany   4
Info: Ollama parameters:
model=llama3.1: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 answers the question the user asked: 'what are the top 5 countries that customers come from?'\n\nThe D
ataFrame was produced using this query: SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM Custom
er\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5\n\nThe following is information about the re
sulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Country   object\nNumberOfCustomers
int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the r
esults of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value i
n the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- ju
st the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:17:08.914463017Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Country', y='NumberOfCustomer
s')\nfig.update_layout(title_text='Top 5 Countries by Number of Customers',\n   xaxis_title_
text='Country',\n   yaxis_title_text='Number of Customers')\n\nif len(df) == 1:\n   fig = p
x.funnel(df, values='NumberOfCustomers', names='Country')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'don
```
```

```
e': True, 'total_duration': 21652723887, 'load_duration': 14822000, 'prompt_eval_count': 282, 'prompt_eval_
duration': 8486173000, 'eval_count': 89, 'eval_duration': 13104683000}
```



Out[17]:
```
('SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY Number
OfCustomers DESC\nLIMIT 5',
    Country  NumberOfCustomers
 0      USA                 13
 1   Canada                  8
 2   France                  5
 3   Brazil                  5
 4  Germany                  4,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'Country=%{x}<br>NumberOfCustomers=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['USA', 'Canada', 'France', 'Brazil', 'Germany'], dtype=object),
               'xaxis': 'x',
               'y': array([13,  8,  5,  5,  4]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'title': {'text': 'Top 5 Countries by Number of Customers'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Country'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Number of Customers'}}}
 }))
```

More SQL questions

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

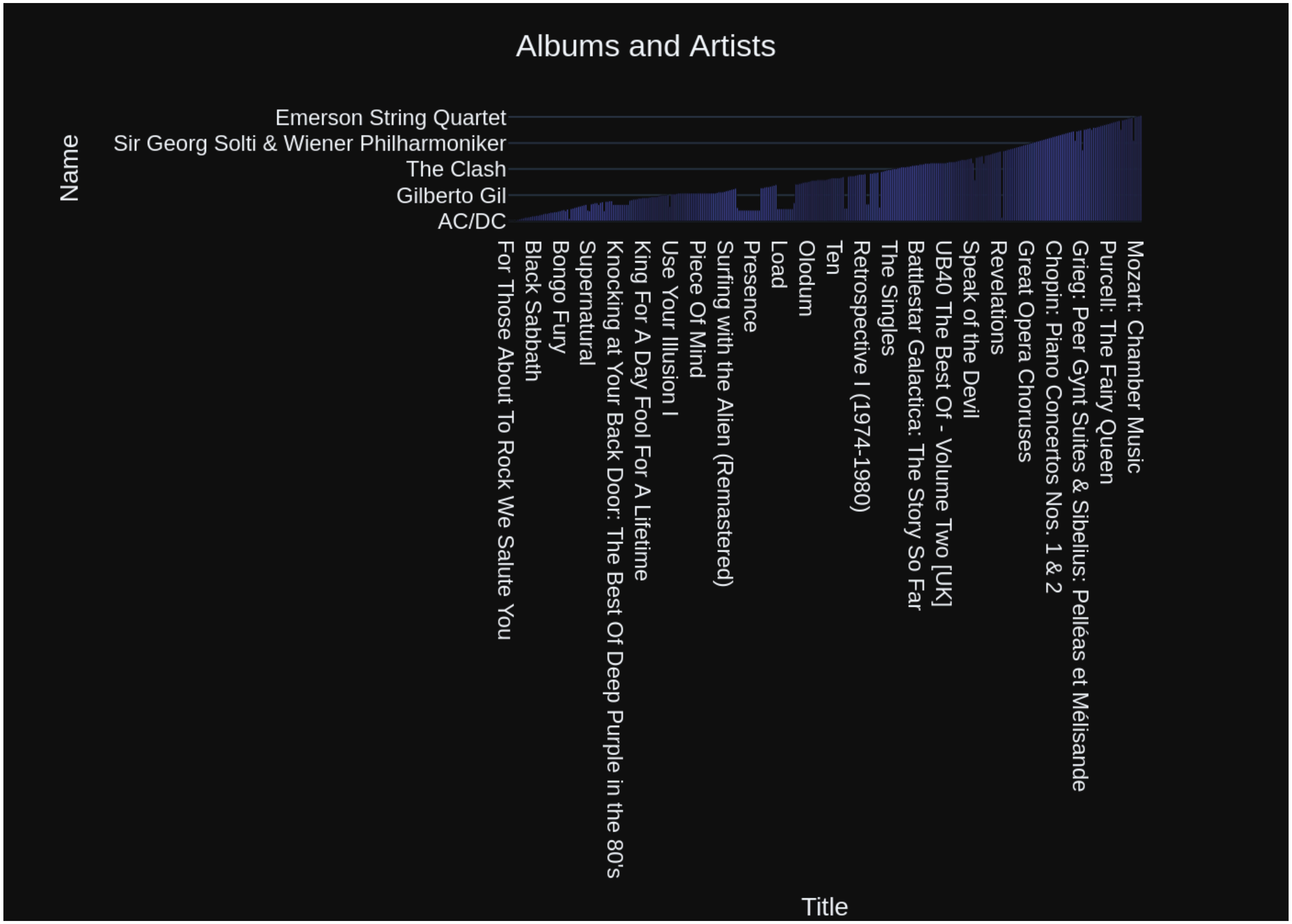
In [18]:
```
question = """
    List all albums and their corresponding artist names
"""
```

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

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

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE INDEX IFK_AlbumArtistId ON Album (ArtistId)\n\nCREATE TA
BLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT
NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (Artis
tId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NU
LL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    Genre
Id INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    Unit
Price NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId)
REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERE
NCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENC
ES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_TrackAlbumI
d ON Track (AlbumId)\n\nCREATE TABLE Artist\n(\n    ArtistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n
CONSTRAINT PK_Artist PRIMARY KEY  (ArtistId)\n)\n\nCREATE INDEX IFK_TrackGenreId ON Track (GenreId)\n\nCREA
TE INDEX IFK_PlaylistTrackTrackId ON PlaylistTrack (TrackId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON Track
(MediaTypeId)\n\nCREATE TABLE Playlist\n(\n    PlaylistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n
CONSTRAINT PK_Playlist PRIMARY KEY  (PlaylistId)\n)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGE
R  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, Tra
ckId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTI
ON\n)\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL que
ry without any explanations for the question. \n2. If the provided context is almost sufficient but require
s 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 p
rovided 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 given before. \n"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come fro
m?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM Custom
er\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many rec
ords are in table called customer'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customer'}, {'r
ole': 'user', 'content': '  \n    List all albums and their corresponding artist names  \n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_AlbumArtistId ON Album (ArtistId)\n\nCREATE TABLE Album\n
(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n
CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\t
ON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Na
me NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGE
R,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUM
ERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCE
S Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genr
e (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaT
ype (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_TrackAlbumId ON Trac
k (AlbumId)\n\nCREATE TABLE Artist\n(\n    ArtistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTR
AINT PK_Artist PRIMARY KEY  (ArtistId)\n)\n\nCREATE INDEX IFK_TrackGenreId ON Track (GenreId)\n\nCREATE IND
EX IFK_PlaylistTrackTrackId ON PlaylistTrack (TrackId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON Track (Media
TypeId)\n\nCREATE TABLE Playlist\n(\n    PlaylistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTR
AINT PK_Playlist PRIMARY KEY  (PlaylistId)\n)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT
NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackI
d),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO A
CTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\n)\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 k
nowledge of a specific string in a particular column, please generate an intermediate SQL query to find the
distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provi
ded context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant ta
ble(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was
given before. \n"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"},
{"role": "assistant", "content": "SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM Customer\nGR
OUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "How many records a
re in table called customer"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customer"}, {"role":
"user", "content": "  \n    List all albums and their corresponding artist names  \n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:17:57.503877291Z', 'message': {'role': 'assistan
t', 'content': 'SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistI
d'}, 'done_reason': 'stop', 'done': True, 'total_duration': 47999536414, 'load_duration': 14473470, 'prompt
_eval_count': 1350, 'prompt_eval_duration': 43620805000, 'eval_count': 27, 'eval_duration': 4008376000}
LLM Response: SELECT A.Title, ART.Name
FROM Album AS A
JOIN Artist AS ART ON A.ArtistId = ART.ArtistId
SELECT A.Title, ART.Name
FROM Album AS A
JOIN Artist AS ART ON A.ArtistId = ART.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=llama3.1: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 answers the question the user asked: '  \n    List all albums and their corresponding artist names
\n'\n\nThe DataFrame was produced using this query: SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artis
t AS ART ON A.ArtistId = ART.ArtistId\n\nThe following is information about the resulting pandas DataFrame
'df': \nRunning df.dtypes gives:\n Title    object\nName     object\ndtype: object"}, {"role": "user", "con
tent": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is i
n a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond wi
th only Python code. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:18:17.925035893Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nif len(df) == 1:\n    fig = px.bar(x=df['Title'],
y=df['Name'])\nelse:\n    fig = px.bar(df, x='Title', y='Name')\nfig.update_layout(title_text='Albums and A
rtists', title_x=0.5)\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 2030678552
6, 'load_duration': 80915497, 'prompt_eval_count': 276, 'prompt_eval_duration': 8463552000, 'eval_count': 7
4, 'eval_duration': 11662745000}
```


Albums and Artists
Name
Emerson String Quartet
Sir Georg Solti & Wiener Philharmoniker
The Clash
Gilberto Gil
AC/DC
For Those About To Rock We Salute You
Black Sabbath
Bongo Fury
Supernatural
Knocking at Your Back Door: The Best Of Deep Purple in the 80's
King For A Day Fool For A Lifetime
Use Your Illusion I
Piece Of Mind
Surfing with the Alien (Remastered)
Presence
Load
Olodum
Ten
Retrospective I (1974-1980)
The Singles
Battlestar Galactica: The Story So Far
UB40 The Best Of - Volume Two [UK]
Speak of the Devil
Revelations
Great Opera Choruses
Chopin: Piano Concertos Nos. 1 & 2
Grieg: Peer Gynt Suites & Sibelius: Pelléas et Mélisande
Purcell: The Fairy Queen
Mozart: Chamber Music
Title

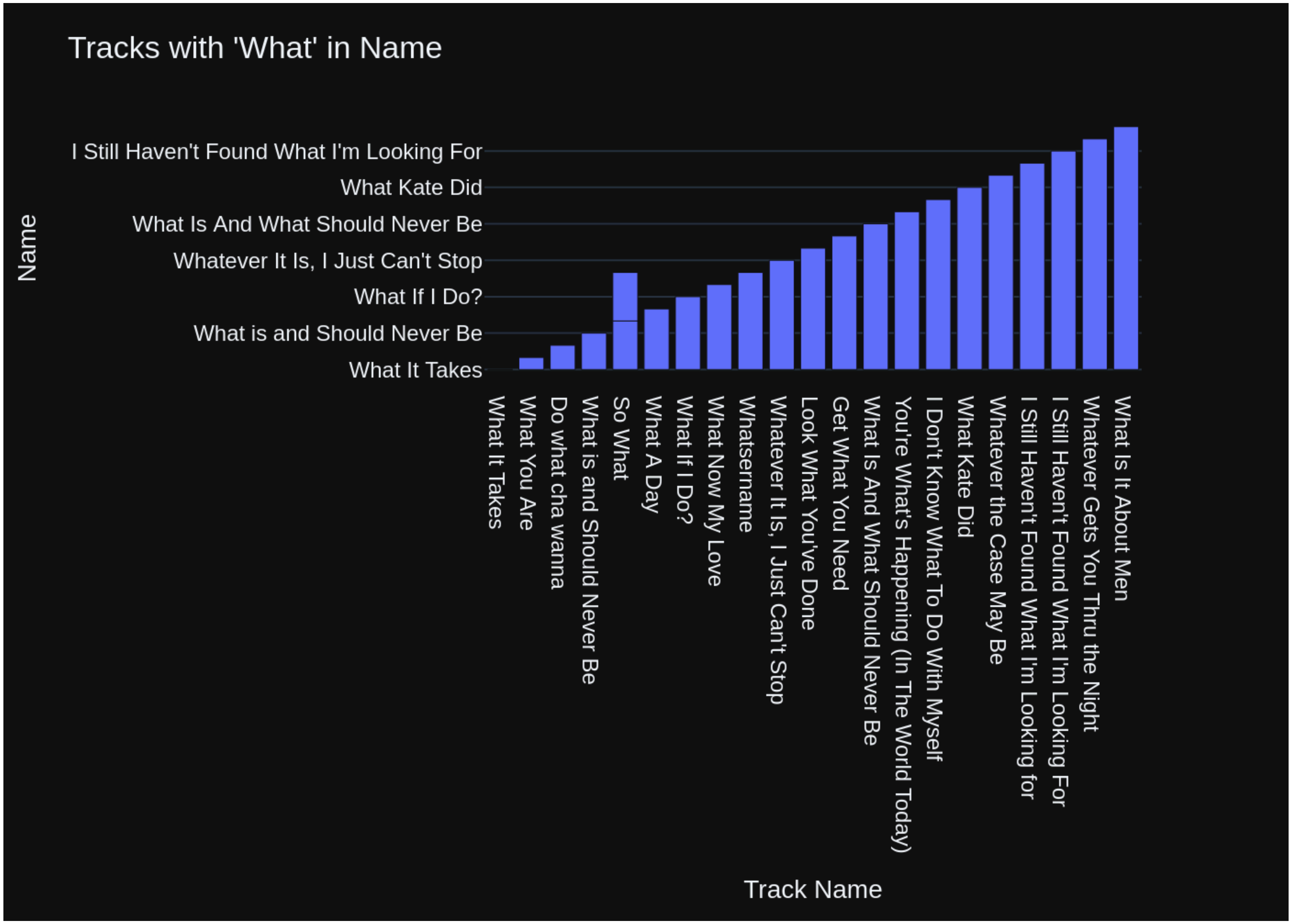
```
Out[18]: ('SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId',
                                                  Title  \
      0                For Those About To Rock We Salute You
      1                                    Balls to the Wall
      2                                    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],
      Figure({
          'data': [{'alignmentgroup': 'True',
                    'hovertemplate': 'Title=%{x}<br>Name=%{y}<extra></extra>',
                    'legendgroup': '',
                    'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                    'name': '',
                    'offsetgroup': '',
                    'orientation': 'v',
                    'showlegend': False,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['For Those About To Rock We Salute You', 'Balls to the Wall',
                                'Restless and Wild', ..., "Monteverdi: L'Orfeo",
                                'Mozart: Chamber Music',
                                'Koyaanisqatsi (Soundtrack from the Motion Picture)'], dtype=object),
                    'xaxis': 'x',
                    'y': array(['AC/DC', 'Accept', 'Accept', ...,
                                'C. Monteverdi, Nigel Rogers - Chiaroscuro; London Baroque; London Cornett & Sa
     ckbu',
                                'Nash Ensemble', 'Philip Glass Ensemble'], dtype=object),
                    'yaxis': 'y'}],
          'layout': {'barmode': 'relative',
                     'legend': {'tracegroupgap': 0},
                     'margin': {'t': 60},
                     'template': '...',
                     'title': {'text': 'Albums and Artists', 'x': 0.5},
                     'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Title'}},
                     'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}}}
      }))
```

In [19]:
```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 3, updating n_results = 3
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON Track (GenreId)\n\nCREATE TABL
E Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    Me
diaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER
NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY
(TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n
FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\n)\n\nCREATE INDEX IFK_TrackAlbumId ON Track (AlbumId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON Playlis
tTrack (TrackId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON Track (MediaTypeId)\n\nCREATE INDEX IFK_InvoiceLin
eTrackId ON InvoiceLine (TrackId)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n
TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIG
N KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOR
EIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE IN
DEX IFK_AlbumArtistId ON Album (ArtistId)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Tit
le NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumI
d),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\n)\n\nCREATE TABLE Playlist\n(\n    PlaylistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT
PK_Playlist PRIMARY KEY  (PlaylistId)\n)\n\n===Response Guidelines \n1. If the provided context is sufficie
nt, please generate a valid SQL query without any explanations for the question. \n2. If the provided conte
xt is almost sufficient but requires knowledge of a specific string in a particular column, please generate
an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment say
ing intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generate
d. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, ple
ase repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': '  \n    List all a
lbums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT A.Title, ART.Nam
e \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': 'what
are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUN
T(CustomerId) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT
5'}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assistant', 'c
ontent': 'SELECT COUNT(*) FROM customer'}, {'role': 'user', 'content': '  \n    Find all tracks with a name
containing "What" (case-insensitive)\n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON Track (GenreId)\n\nCREATE TABLE Track\n
(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId
INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NUL
L,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (Track
Id),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n
FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIG
N KEY (MediaTypeId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n
CREATE INDEX IFK_TrackAlbumId ON Track (AlbumId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON PlaylistTrack
(TrackId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON Track (MediaTypeId)\n\nCREATE INDEX IFK_InvoiceLineTrackI
d ON InvoiceLine (TrackId)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId
INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (P
laylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY
(TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_A
lbumArtistId ON Album (ArtistId)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCH
AR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    F
OREIGN KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREA
TE TABLE Playlist\n(\n    PlaylistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Playlis
t PRIMARY KEY  (PlaylistId)\n)\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 almos
t sufficient but requires knowledge of a specific string in a particular column, please generate an interme
diate 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. Ple
ase use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat
the answer exactly as it was given before. \n"}, {"role": "user", "content": "  \n    List all albums and t
heir corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT A.Title, ART.Name \nFROM Al
bum AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": "what are the top
5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(CustomerI
d) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5"}, {"rol
e": "user", "content": "How many records are in table called customer"}, {"role": "assistant", "content":
"SELECT COUNT(*) FROM customer"}, {"role": "user", "content": "  \n    Find all tracks with a name containi
ng \"What\" (case-insensitive)\n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:19:05.15070236Z', 'message': {'role': 'assistan
t', 'content': "SELECT Name \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, 'done_reason': 'stop', 'don
e': True, 'total_duration': 46790984190, 'load_duration': 14545162, 'prompt_eval_count': 1367, 'prompt_eval
_duration': 44008625000, 'eval_count': 15, 'eval_duration': 2197569000}
LLM Response: SELECT Name
FROM Track
WHERE LOWER(Name) LIKE '%what%'
SELECT Name
FROM Track
WHERE LOWER(Name) LIKE '%what%'
                                            Name
0                                  What It Takes
```

```
1                                   What You Are
2                               Do what cha wanna
3                      What is and Should Never Be
4                                        So What
5                                     What A Day
6                                   What If I Do?
7                                 What Now My Love
8                                     Whatsername
9                 Whatever It Is, I Just Can't Stop
10                             Look What You've Done
11                                 Get What You Need
12                  What Is And What Should Never Be
13  You're What's Happening (In The World Today)
14                                         So What
15               I Don't Know What To Do With Myself
16                                   What Kate Did
17                          Whatever the Case May Be
18    I Still Haven't Found What I'm Looking for
19    I Still Haven't Found What I'm Looking For
20                  Whatever Gets You Thru the Night
21                            What Is It About Men
Info: Ollama parameters:
model=llama3.1: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 answers the question the user asked: '  \n    Find all tracks with a name containing \"What\" (case-in
sensitive)\n'\n\nThe DataFrame was produced using this query: SELECT Name \nFROM Track \nWHERE LOWER(Name)
LIKE '%what%'\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtype
s gives:\n Name    object\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly
code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there
is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any
explanations -- just the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:19:24.458846527Z', 'message': {'role': 'assistan
t', 'content': '```python\nimport plotly.express as px\n\nif len(df) == 1:\n    fig = px.bar(x=df[\'Name
\'], y=[df[\'Name\']])\nelse:\n    fig = px.bar(df, x=\'Name\', y=\'Name\')\nfig.update_layout(title="Track
s with \'What\' in Name", xaxis_title="Track Name")\nfig.show()\n```'}, 'done_reason': 'stop', 'done': Tru
e, 'total_duration': 19187952428, 'load_duration': 81988880, 'prompt_eval_count': 256, 'prompt_eval_duratio
n': 7782139000, 'eval_count': 77, 'eval_duration': 11268512000}
```


Tracks with 'What' in Name
Name
I Still Haven't Found What I'm Looking For
What Kate Did
What Is And What Should Never Be
Whatever It Is, I Just Can't Stop
What If I Do?
What is and Should Never Be
What It Takes
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
Track Name

```
Out[19]: ("SELECT Name \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'",
                                                   Name
           0                                What It Takes
           1                                 What You Are
           2                             Do what cha wanna
           3                   What is and Should Never Be
           4                                       So What
           5                                    What A Day
           6                                 What If I Do?
           7                              What Now My Love
           8                                   Whatsername
           9             Whatever It Is, I Just Can't Stop
           10                        Look What You've Done
           11                            Get What You Need
           12             What Is And What Should Never Be
           13  You're What's Happening (In The World Today)
           14                                      So What
           15          I Don't Know What To Do With Myself
           16                                What Kate Did
           17                      Whatever the Case May Be
           18    I Still Haven't Found What I'm Looking for
           19    I Still Haven't Found What I'm Looking For
           20             Whatever Gets You Thru the Night
           21                         What Is It About Men,
           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(['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(['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),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'title': {'text': "Tracks with 'What' in Name"},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Track Name'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}}}
           }))
```

In [20]:

```
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 4, updating n_results = 4
```

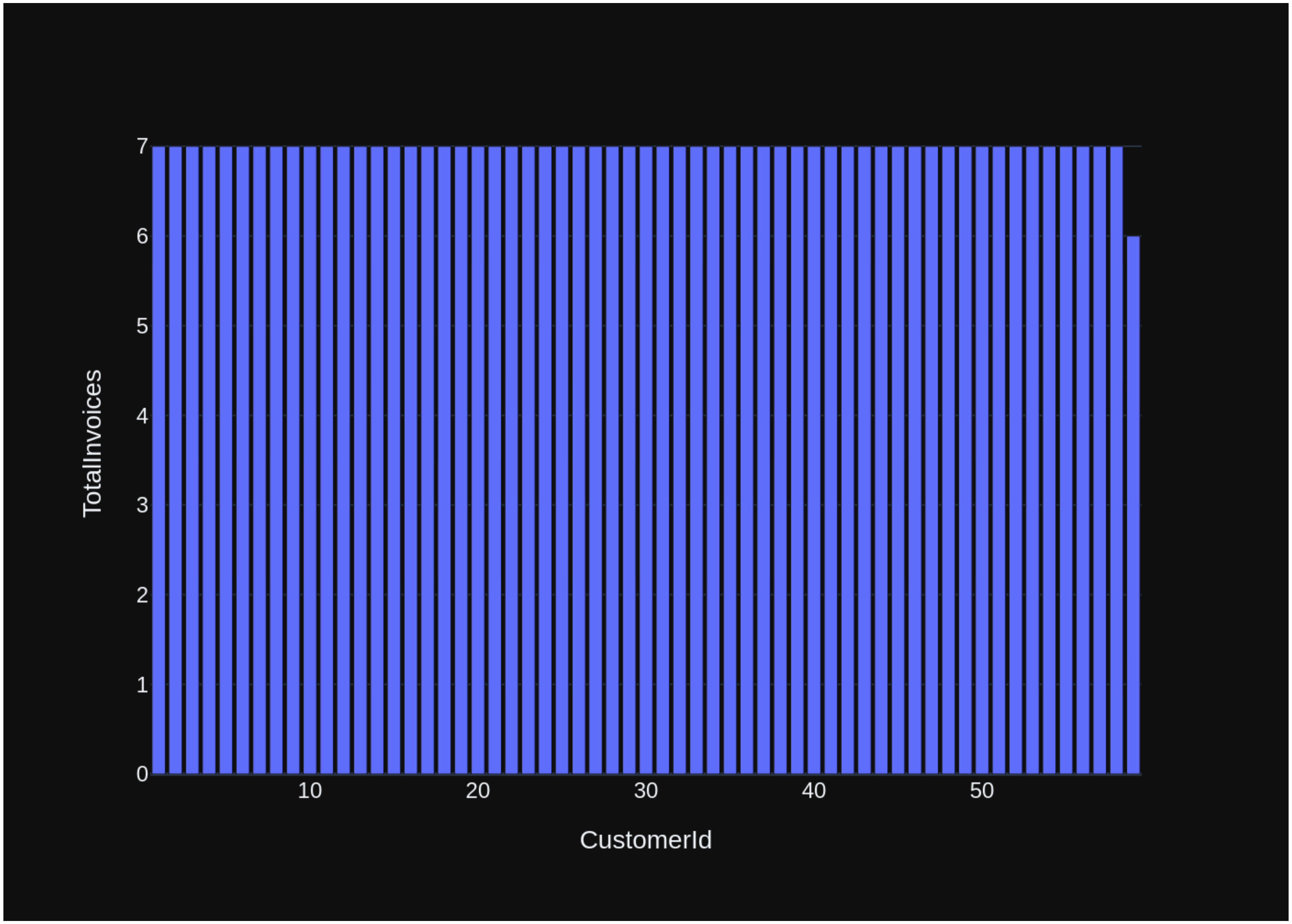
```
SQL Prompt: [{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to
answer the question. Your response should ONLY be based on the given context and follow the response guidel
ines and format instructions. \n===Tables \nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nC
REATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDat
e DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n    BillingState NV
ARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    BillingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)
NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Cus
tomer (CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceI
d ON InvoiceLine (InvoiceId)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    Invo
iceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quant
ity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (Invoi
ceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (Track
Id) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_Invoice
LineTrackId ON InvoiceLine (TrackId)\n\nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    Fi
rstName NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Addre
ss NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCod
e NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    Sup
portRepId INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId) R
EFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_Custom
erSupportRepId ON Customer (SupportRepId)\n\nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\n\n
CREATE TABLE Employee\n(\n    EmployeeId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Firs
tName NVARCHAR(20)  NOT NULL,\n    Title NVARCHAR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIME,\n
HireDate DATETIME,\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country
NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVAR
CHAR(60),\n    CONSTRAINT PK_Employee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Em
ployee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Track\n(\n    TrackId
INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT N
ULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INT
EGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN
KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (G
enreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTy
peId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response Gu
idelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanat
ions for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specif
ic 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 insuf
ficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the
question has been asked and answered before, please repeat the answer exactly as it was given before. \n"},
{'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assistant', 'conten
t': 'SELECT COUNT(*) FROM customer'}, {'role': 'user', 'content': 'what are the top 5 countries that custom
ers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(CustomerId) AS NumberOfCustomers
\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content':
```

```
'  \n    List all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELE
CT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId'}, {'role': 'use
r', 'content': '  \n    Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'ass
istant', 'content': "SELECT Name \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'conten
t': '  \n    Get the total number of invoices for each customer\n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE TABLE
Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME
NOT NULL,\n    BillingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(4
0),\n    BillingCountry NVARCHAR(40),\n    BillingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NU
LL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer
(CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON I
nvoiceLine (InvoiceId)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    InvoiceId
INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quantity IN
TEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (InvoiceId)
REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) RE
FERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceLineTr
ackId ON InvoiceLine (TrackId)\n\nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    FirstNam
e NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Address NVA
RCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVAR
CHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    SupportRe
pId INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId) REFEREN
CES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_CustomerSupp
ortRepId ON Customer (SupportRepId)\n\nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\n\nCREATE
TABLE Employee\n(\n    EmployeeId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    FirstName N
VARCHAR(20)  NOT NULL,\n    Title NVARCHAR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIME,\n    HireD
ate DATETIME,\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVAR
CHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR
(60),\n    CONSTRAINT PK_Employee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Employ
ee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Track\n(\n    TrackId INTE
GER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NUL
L,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEG
ER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN K
EY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (Ge
nreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTyp
eId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response Gui
```

```
delines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanati
ons for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specifi
c 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 insuf
ficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the
question has been asked and answered before, please repeat the answer exactly as it was given before. \n"},
{"role": "user", "content": "How many records are in table called customer"}, {"role": "assistant", "conten
t": "SELECT COUNT(*) FROM customer"}, {"role": "user", "content": "what are the top 5 countries that custom
ers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(CustomerId) AS NumberOfCustomers
\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5"}, {"role": "user", "content":
"  \n    List all albums and their corresponding artist names  \n"}, {"role": "assistant", "content": "SELE
CT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId"}, {"role": "use
r", "content": "  \n    Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "a
ssistant", "content": "SELECT Name \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "cont
ent": "  \n    Get the total number of invoices for each customer\n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:20:39.447996275Z', 'message': {'role': 'assistan
t', 'content': 'SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I
ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId'}, 'done_reason': 'stop', 'done': True, 'total_durat
ion': 74835131591, 'load_duration': 19242982, 'prompt_eval_count': 2046, 'prompt_eval_duration': 6824503000
0, 'eval_count': 38, 'eval_duration': 5956631000}
LLM Response: SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices
FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
GROUP BY C.CustomerId
SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices
FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
GROUP BY C.CustomerId
    CustomerId  TotalInvoices
0            1              7
1            2              7
2            3              7
3            4              7
4            5              7
5            6              7
6            7              7
7            8              7
8            9              7
9           10              7
10          11              7
11          12              7
```

```
       12           13              7
       13           14              7
       14           15              7
       15           16              7
       16           17              7
       17           18              7
       18           19              7
       19           20              7
       20           21              7
       21           22              7
       22           23              7
       23           24              7
       24           25              7
       25           26              7
       26           27              7
       27           28              7
       28           29              7
       29           30              7
       30           31              7
       31           32              7
       32           33              7
       33           34              7
       34           35              7
       35           36              7
       36           37              7
       37           38              7
       38           39              7
       39           40              7
       40           41              7
       41           42              7
       42           43              7
       43           44              7
       44           45              7
       45           46              7
       46           47              7
       47           48              7
       48           49              7
       49           50              7
       50           51              7
       51           52              7
       52           53              7
       53           54              7
```

```
54          55              7
55          56              7
56          57              7
57          58              7
58          59              6
Info: Ollama parameters:
model=llama3.1: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 answers the question the user asked: '  \n    Get the total number of invoices for each customer\n'\n
\nThe DataFrame was produced using this query: SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nF
ROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId\n\nThe following is
information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId       int64
\nTotalInvoices    int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly
code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there
is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any
explanations -- just the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:21:03.609493422Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nif df['TotalInvoices'].nunique() == 1:\n    fig =
px.indicator(value=df['TotalInvoices'].iloc[0],\n                        title='Total Invoices',\n
number=pxIndicatorNumber.INDICATOR_VALUE,\n                        color_discrete_seq=['#3498db'])\nelse:\n
fig = px.bar(df, x='CustomerId', y='TotalInvoices')\n    \nfig.show()\n```"}, 'done_reason': 'stop', 'don
e': True, 'total_duration': 24061121908, 'load_duration': 53995742, 'prompt_eval_count': 310, 'prompt_eval_
duration': 9478613000, 'eval_count': 98, 'eval_duration': 14468954000}
```


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

```
Out[20]: ('SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I ON C.Custome
         rId = I.CustomerId \nGROUP BY C.CustomerId',
              CustomerId  TotalInvoices
          0            1              7
          1            2              7
          2            3              7
          3            4              7
          4            5              7
          5            6              7
          6            7              7
          7            8              7
          8            9              7
          9           10              7
          10          11              7
          11          12              7
          12          13              7
          13          14              7
          14          15              7
          15          16              7
          16          17              7
          17          18              7
          18          19              7
          19          20              7
          20          21              7
          21          22              7
          22          23              7
          23          24              7
          24          25              7
          25          26              7
          26          27              7
          27          28              7
          28          29              7
          29          30              7
          30          31              7
          31          32              7
          32          33              7
          33          34              7
          34          35              7
          35          36              7
          36          37              7
          37          38              7
          38          39              7
```

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

In [21]:

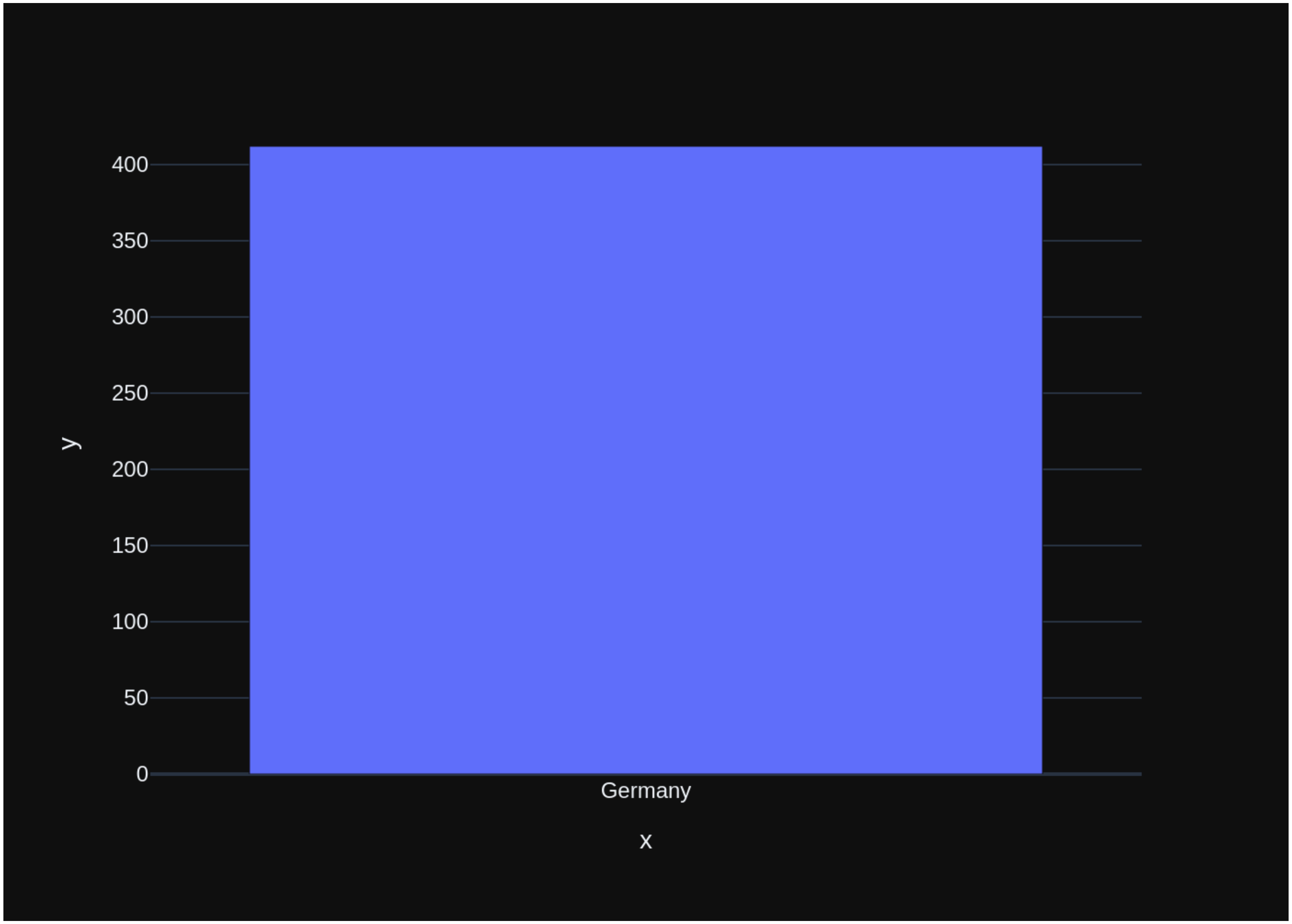
```
question = """
    Find the total number of invoices per country:
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n    
CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(70),\n    B
illingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    BillingPost
alCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceI
d),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO A
CTION\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON InvoiceLine (InvoiceId)\n\nCREATE INDEX IFK_InvoiceCus
tomerId ON Invoice (CustomerId)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    I
nvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Qu
antity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (In
voiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (Tr
ackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_Invo
iceLineTrackId ON InvoiceLine (TrackId)\n\nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    
FirstName NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Add
ress NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalC
ode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    S
upportRepId INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId)
REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Employee
\n(\n    EmployeeId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    FirstName NVARCHAR(20)  N
OT NULL,\n    Title NVARCHAR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIME,\n    HireDate DATETIM
E,\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n
PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60),\n    CONS
TRAINT PK_Employee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Employee (EmployeeId)
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n
Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGE
R,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUM
ERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCE
S Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genr
e (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaT
ype (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON
Employee (ReportsTo)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT
NULL,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (A
rtistId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response Guide
lines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanation
s for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific
string in a particular column, please generate an intermediate SQL query to find the distinct strings in th
at column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insuffi
cient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the qu
estion has been asked and answered before, please repeat the answer exactly as it was given before. \n"},
{'role': 'user', 'content': '  \n    Get the total number of invoices for each customer\n'}, {'role': 'assi
stant', 'content': 'SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoi
```

```
ce I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId'}, {'role': 'user', 'content': 'what are the to
p 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(Customer
Id) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5'}, {'ro
le': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assistant', 'content':
'SELECT COUNT(*) FROM customer'}, {'role': 'user', 'content': '  \n    List all albums and their correspond
ing artist names  \n'}, {'role': 'assistant', 'content': 'SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN
Artist AS ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': '  \n    Find all tracks with a n
ame containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT Name \nFROM Track \n
WHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': '  \n    Find the total number of invoices p
er country:\n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n    CustomerId
INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(70),\n    BillingCity
NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    BillingPostalCode NVAR
CHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),\n    FOR
EIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n
CREATE INDEX IFK_InvoiceLineInvoiceId ON InvoiceLine (InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON I
nvoice (CustomerId)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    InvoiceId INT
EGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quantity INTEG
ER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (InvoiceId) REF
ERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFER
ENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceLineTrack
Id ON InvoiceLine (TrackId)\n\nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    FirstName N
VARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Address NVARCH
AR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHA
R(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    SupportRepId
INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId) REFERENCES
Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Employee\n(\n    Emp
loyeeId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    FirstName NVARCHAR(20)  NOT NULL,\n
Title NVARCHAR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIME,\n    HireDate DATETIME,\n    Address N
VARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NV
ARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60),\n    CONSTRAINT PK_Emp
loyee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Employee (EmployeeId) \n\t\tON DEL
ETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVA
RCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n
Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,
2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album
(AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (Genre
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (Med
iaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON Employe
e (ReportsTo)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NUL
L,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (Arti
stId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response Guidelin
es \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations f
or the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific str
ing in a particular column, please generate an intermediate SQL query to find the distinct strings in that
column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficie
nt, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the quest
ion has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"ro
le": "user", "content": "  \n    Get the total number of invoices for each customer\n"}, {"role": "assistan
t", "content": "SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I
ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId"}, {"role": "user", "content": "what are the top 5 c
ountries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(CustomerId) A
S NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5"}, {"role":
"user", "content": "How many records are in table called customer"}, {"role": "assistant", "content": "SELE
CT COUNT(*) FROM customer"}, {"role": "user", "content": "  \n    List all albums and their corresponding a
rtist names  \n"}, {"role": "assistant", "content": "SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Arti
st AS ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": "  \n    Find all tracks with a name
containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT Name \nFROM Track \nWH
ERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": "  \n    Find the total number of invoices per
country:\n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:22:14.256579032Z', 'message': {'role': 'assistan
t', 'content': 'SELECT COUNT(DISTINCT InvoiceId), BillingCountry\nFROM Invoice'}, 'done_reason': 'stop', 'd
one': True, 'total_duration': 70375534175, 'load_duration': 15199447, 'prompt_eval_count': 2047, 'prompt_ev
al_duration': 67990697000, 'eval_count': 14, 'eval_duration': 2160649000}
LLM Response: SELECT COUNT(DISTINCT InvoiceId), BillingCountry
FROM Invoice
SELECT COUNT(DISTINCT InvoiceId), BillingCountry
FROM Invoice
   COUNT(DISTINCT InvoiceId) BillingCountry
0                        412        Germany
Info: Ollama parameters:
model=llama3.1: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 answers the question the user asked: '  \n    Find the total number of invoices per country:\n'\n\nThe
DataFrame was produced using this query: SELECT COUNT(DISTINCT InvoiceId), BillingCountry\nFROM Invoice\n\n
The following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n COUNT
(DISTINCT InvoiceId)     int64\nBillingCountry               object\ndtype: object"}, {"role": "user", "con
tent": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is i
n a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond wi
th only Python code. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:22:34.533390739Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nif df.shape[0] == 1:\n    fig = px.bar(x=df['Bill
ingCountry'], y=df['COUNT(DISTINCT InvoiceId)'])\nelse:\n    fig = px.bar(df, x='BillingCountry', y='COUNT
(DISTINCT InvoiceId)', title='Total Number of Invoices per Country')\nfig.show()\n```"}, 'done_reason': 'st
op', 'done': True, 'total_duration': 20167208665, 'load_duration': 80807542, 'prompt_eval_count': 266, 'pro
mpt_eval_duration': 8101149000, 'eval_count': 81, 'eval_duration': 11885372000}
```


400
350
300
250
200
150
100
50
0
y
Germany
x

```
Out[21]: ('SELECT COUNT(DISTINCT InvoiceId), BillingCountry\nFROM Invoice',
             COUNT(DISTINCT InvoiceId) BillingCountry
          0                         412        Germany,
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'x=%{x}<br>y=%{y}<extra></extra>',
                        'legendgroup': '',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array(['Germany'], dtype=object),
                        'xaxis': 'x',
                        'y': array([412]),
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                         'legend': {'tracegroupgap': 0},
                         'margin': {'t': 60},
                         'template': '...',
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'x'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'y'}}}
          }))
```

```
In [22]: question = """
             List all invoices with a total exceeding $10:
         """

         vn.ask(question=question)
```

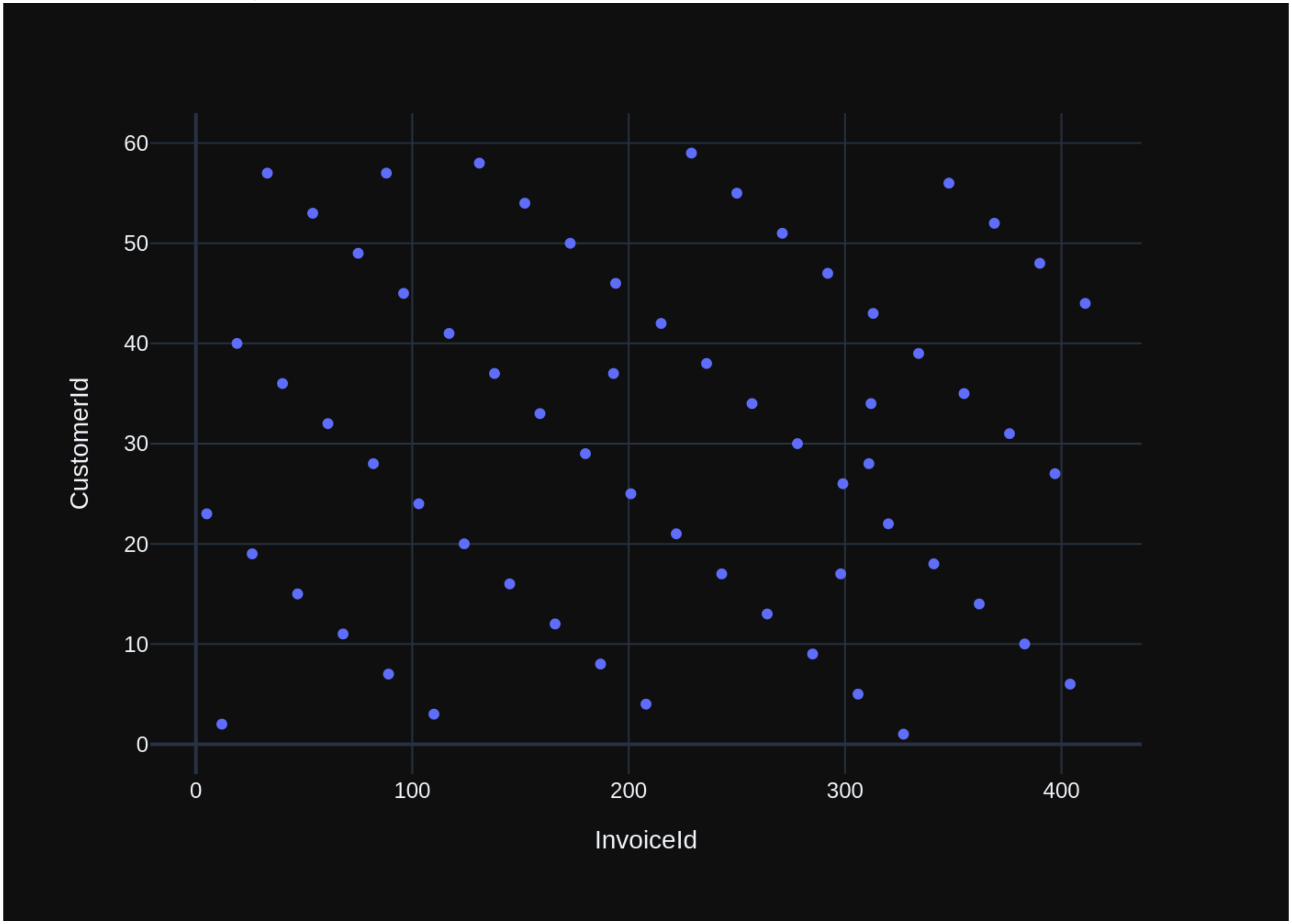
```
Number of requested results 10 is greater than number of elements in index 6, updating n_results = 6
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NUL
L,\n    InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NUL
L,\n    Quantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREI
GN KEY (InvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREI
GN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDE
X IFK_InvoiceLineInvoiceId ON InvoiceLine (InvoiceId)\n\nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NO
T NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHA
R(70),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n
BillingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY
(InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UP
DATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE INDEX IFK_Invoice
LineTrackId ON InvoiceLine (TrackId)\n\nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVA
RCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n
Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,
2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album
(AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (Genre
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (Med
iaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON Employe
e (ReportsTo)\n\nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    FirstName NVARCHAR(40)  N
OT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Address NVARCHAR(70),\n    C
ity NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Ph
one NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    SupportRepId INTEGER,\n
CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId) REFERENCES Employee (Empl
oyeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Employee\n(\n    EmployeeId INTEGE
R  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    FirstName NVARCHAR(20)  NOT NULL,\n    Title NVARCH
AR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIME,\n    HireDate DATETIME,\n    Address NVARCHAR(7
0),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(1
0),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60),\n    CONSTRAINT PK_Employee PR
IMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO A
CTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON Customer (SupportRepId)\n\n===Resp
onse Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any e
xplanations 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 stri
ngs in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context i
s 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'}, {'rol
e': 'assistant', 'content': 'SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJ
OIN Invoice I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId'}, {'role': 'user', 'content': '  \n
Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT COUNT(DISTINCT
InvoiceId), BillingCountry\nFROM Invoice'}, {'role': 'user', 'content': 'How many records are in table call
ed customer'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customer'}, {'role': 'user', 'conten
t': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Cou
ntry, COUNT(CustomerId) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers D
ESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    List all albums and their corresponding artist names 
\n'}, {'role': 'assistant', 'content': 'SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON
A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': '  \n    Find all tracks with a name containing "W
hat" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT Name \nFROM Track \nWHERE LOWER(Name)
LIKE '%what%'"}, {'role': 'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    Inv
oiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quan
tity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (Invo
iceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (Trac
kId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_Invoic
eLineInvoiceId ON InvoiceLine (InvoiceId)\n\nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n
CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(70),\n    B
illingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    BillingPost
alCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceI
d),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO A
CTION\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineTrack
Id ON InvoiceLine (TrackId)\n\nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(20
0)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Composer
NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT N
ULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId)
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n\t\t
ON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaTypeId)
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON Employee (Reports
To)\n\nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    FirstName NVARCHAR(40)  NOT NULL,\n
LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Address NVARCHAR(70),\n    City NVARCHAR(4
0),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(2
4),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    SupportRepId INTEGER,\n    CONSTRAINT PK
_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId) REFERENCES Employee (EmployeeId) \n\t
\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Employee\n(\n    EmployeeId INTEGER  NOT NUL
L,\n    LastName NVARCHAR(20)  NOT NULL,\n    FirstName NVARCHAR(20)  NOT NULL,\n    Title NVARCHAR(30),\n
ReportsTo INTEGER,\n    BirthDate DATETIME,\n    HireDate DATETIME,\n    Address NVARCHAR(70),\n    City NV
```

```
ARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NV
ARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60),\n    CONSTRAINT PK_Employee PRIMARY KEY  (Emplo
yeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON Customer (SupportRepId)\n\n===Response Guidelines
\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for
the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string
in a particular column, please generate an intermediate SQL query to find the distinct strings in that colu
mn. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient,
please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question
has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role":
"user", "content": "  \n    Get the total number of invoices for each customer\n"}, {"role": "assistant",
"content": "SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I ON
C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId"}, {"role": "user", "content": "  \n    Find the total
number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT COUNT(DISTINCT InvoiceId), Bi
llingCountry\nFROM Invoice"}, {"role": "user", "content": "How many records are in table called customer"},
{"role": "assistant", "content": "SELECT COUNT(*) FROM customer"}, {"role": "user", "content": "what are th
e top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(Cust
omerId) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5"},
{"role": "user", "content": "  \n    List all albums and their corresponding artist names  \n"}, {"role":
"assistant", "content": "SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = AR
T.ArtistId"}, {"role": "user", "content": "  \n    Find all tracks with a name containing \"What\" (case-in
sensitive)\n"}, {"role": "assistant", "content": "SELECT Name \nFROM Track \nWHERE LOWER(Name) LIKE '%wha
t%'"}, {"role": "user", "content": "  \n    List all invoices with a total exceeding $10:\n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:23:44.874358758Z', 'message': {'role': 'assistan
t', 'content': 'SELECT * FROM Invoice WHERE Total > 10'}, 'done_reason': 'stop', 'done': True, 'total_durat
ion': 70232082374, 'load_duration': 15472654, 'prompt_eval_count': 2046, 'prompt_eval_duration': 6794326500
0, 'eval_count': 10, 'eval_duration': 1538458000}
LLM Response: SELECT * FROM Invoice WHERE Total > 10
SELECT * FROM Invoice 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=llama3.1: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 answers the question the user asked: '  \n    List all invoices with a total exceeding $10:\n'\n\nThe
DataFrame was produced using this query: SELECT * FROM Invoice WHERE Total > 10\n\nThe following is informa
tion about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n InvoiceId              int64\n
CustomerId             int64\nInvoiceDate           object\nBillingAddress        object\nBillingCity
object\nBillingState          object\nBillingCountry        object\nBillingPostalCode     object\nTotal
float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the
results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value
in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- j
ust the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:24:30.527363502Z', 'message': {'role': 'assistan
t', 'content': '```python\nimport plotly.express as px\nimport pandas as pd\n\nif len(df) == 1:\n    fig =
px.bar(df, x=\'InvoiceId\', y=\'Total\', title=\'Single Invoice Total\')\nelif len(df) > 1:\n    fig = px.b
ar(df, x=\'InvoiceId\', y=\'Total\', title=\'Multiple Invoices Totals\')\nelse:\n    fig = px.density_mapbo
x(df, lat=0, lon=0, z=\'Total\', radius=100,\n                             color_discrete_sequence=["#f7b2c
5"],\n                             hover_data={\'Total\': True},\n                             mapbox_style
="carto-positron",\n                             center=dict(lat=0,lon=0),\n                             zo
om=1)\nelse:\n    fig = px.density_mapbox(df, lat=[0], lon=[0], z=\'Total\', radius=100,\n
color_discrete_sequence=["#f7b2c5"],\n                             hover_data={\'Total\': True},\n
mapbox_style="carto-positron",\n                             center=dict(lat=[0],lon=[0]),\n
zoom=1)\n\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 45407313858, 'load_dur
```

```
ation': 16775699, 'prompt_eval_count': 322, 'prompt_eval_duration': 9915433000, 'eval_count': 237, 'eval_du
ration': 35331230000}
```



```
Out[22]: ('SELECT * FROM Invoice WHERE Total > 10',
              InvoiceId  CustomerId          InvoiceDate             BillingAddress  \
          0           5          23  2009-01-11 00:00:00            69 Salem Street   
          1          12           2  2009-02-11 00:00:00    Theodor-Heuss-Straße 34   
          2          19          40  2009-03-14 00:00:00             8, Rue Hanovre   
          3          26          19  2009-04-14 00:00:00            1 Infinite Loop   
          4          33          57  2009-05-15 00:00:00            Calle Lira, 198   
          ..        ...         ...                  ...                        ...   
          59        383          10  2013-08-12 00:00:00  Rua Dr. Falcão Filho, 155   
          60        390          48  2013-09-12 00:00:00      Lijnbaansgracht 120bg   
          61        397          27  2013-10-13 00:00:00            1033 N Park Ave   
          62        404           6  2013-11-13 00:00:00              Rilská 3174/6   
          63        411          44  2013-12-14 00:00:00            Porthaninkatu 9   

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

          [64 rows x 9 columns],
          Figure({
              'data': [{'hovertemplate': 'InvoiceId=%{x}<br>CustomerId=%{y}<extra></extra>',
                        'legendgroup': '',
                        'marker': {'color': '#636efa', 'symbol': 'circle'},
                        'mode': 'markers',
                        'name': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'type': 'scatter',
                        'x': array([  5,  12,  19,  26,  33,  40,  47,  54,  61,  68,  75,  82,  88,  89,
                                     96, 103, 110, 117, 124, 131, 138, 145, 152, 159, 166, 173, 180, 187,
                                    193, 194, 201, 208, 215, 222, 229, 236, 243, 250, 257, 264, 271, 278,
                                    285, 292, 298, 299, 306, 311, 312, 313, 320, 327, 334, 341, 348, 355,
                                    362, 369, 376, 383, 390, 397, 404, 411]),
                        'xaxis': 'x',
                        'y': array([23,  2, 40, 19, 57, 36, 15, 53, 32, 11, 49, 28, 57,  7, 45, 24,  3, 41,
                                    20, 58, 37, 16, 54, 33, 12, 50, 29,  8, 37, 46, 25,  4, 42, 21, 59, 38,
                                    17, 55, 34, 13, 51, 30,  9, 47, 17, 26,  5, 28, 34, 43, 22,  1, 39, 18,
                                    56, 35, 14, 52, 31, 10, 48, 27,  6, 44]),
                        'yaxis': 'y'}],
              'layout': {'legend': {'tracegroupgap': 0},
                         'margin': {'t': 60},
                         'template': '...',
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceId'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}}}
          }))
```

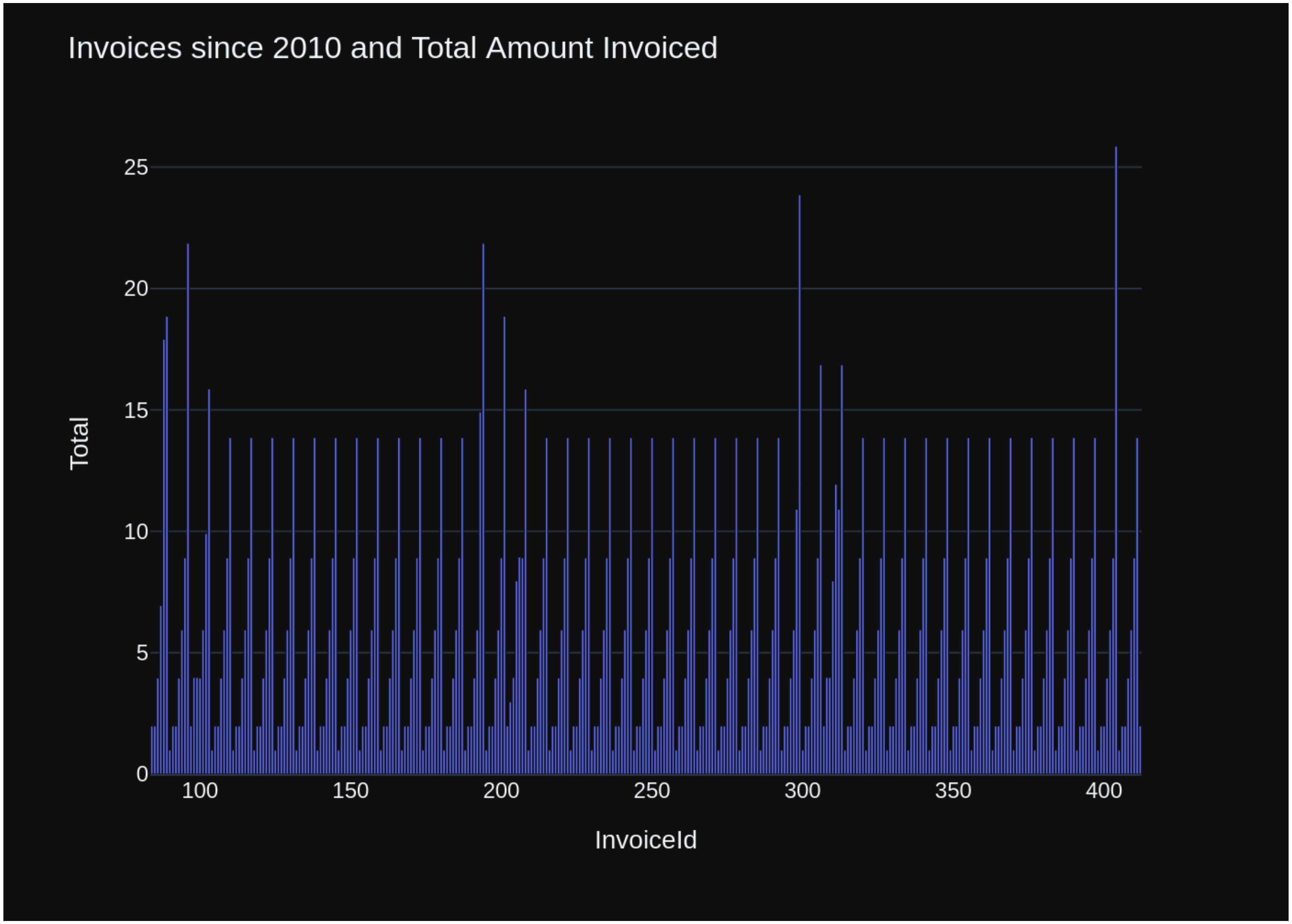
In [23]:
```
question = """
    Find all invoices since 2010 and the total amount invoiced:
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n    
CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(70),\n    B
illingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    BillingPost
alCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceI
d),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO A
CTION\n)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    InvoiceId INTEGER  NOT N
ULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quantity INTEGER  NOT NUL
L,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (InvoiceId) REFERENCES Inv
oice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track
(TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON Invo
iceLine (InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE INDEX IFK_Invoi
ceLineTrackId ON InvoiceLine (TrackId)\n\nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    
FirstName NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Add
ress NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalC
ode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    S
upportRepId INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId)
REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Employee
\n(\n    EmployeeId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    FirstName NVARCHAR(20)  N
OT NULL,\n    Title NVARCHAR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIME,\n    HireDate DATETIM
E,\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n
PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60),\n    CONS
TRAINT PK_Employee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Employee (EmployeeId)
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n
Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGE
R,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUM
ERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCE
S Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genr
e (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaT
ype (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE PlaylistTrack\n(\n    Pl
aylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  
(PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO AC
TION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION O
N UPDATE NO ACTION\n)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NO
T NULL,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY 
(ArtistId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response Gui
delines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanati
ons for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specifi
c 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 insuf
ficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the
```

```
question has been asked and answered before, please repeat the answer exactly as it was given before. \n"},
{'role': 'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistan
t', 'content': 'SELECT * FROM Invoice WHERE Total > 10'}, {'role': 'user', 'content': '  \n    Find the tot
al number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT COUNT(DISTINCT InvoiceId),
BillingCountry\nFROM Invoice'}, {'role': 'user', 'content': '  \n    Get the total number of invoices for e
ach customer\n'}, {'role': 'assistant', 'content': 'SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoice
s \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId'}, {'role': 'us
er', 'content': 'How many records are in table called customer'}, {'role': 'assistant', 'content': 'SELECT
COUNT(*) FROM customer'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come fro
m?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM Custom
er\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find
all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT N
ame \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': '  \n    List all albums
and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT A.Title, ART.Name \nFR
OM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': '  \n    Fin
d all invoices since 2010 and the total amount invoiced:\n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n    CustomerId
INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(70),\n    BillingCity
NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    BillingPostalCode NVAR
CHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),\n    FOR
EIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n
CREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    InvoiceId INTEGER  NOT NULL,\n    Tr
ackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quantity INTEGER  NOT NULL,\n    CONS
TRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (InvoiceId) REFERENCES Invoice (Invoic
eId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId)
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON InvoiceLine (I
nvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineTrac
kId ON InvoiceLine (TrackId)\n\nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    FirstName
NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Address NVARC
HAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCH
AR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    SupportRepI
d INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId) REFERENCE
S Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Employee\n(\n    E
mployeeId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    FirstName NVARCHAR(20)  NOT NULL,\n
Title NVARCHAR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIME,\n    HireDate DATETIME,\n    Address N
VARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NV
ARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60),\n    CONSTRAINT PK_Emp
loyee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Employee (EmployeeId) \n\t\tON DEL
ETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVA
RCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    
Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,
2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album 
(AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (Genre
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (Med
iaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId
INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistI
d, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UP
DATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION\n)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n 
ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) RE
FERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response Guidelines \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
particular column, please generate an intermediate SQL query to find the distinct strings in that column. P
repend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, pleas
e explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has b
een asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "use
r", "content": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content":
"SELECT * FROM Invoice WHERE Total > 10"}, {"role": "user", "content": "  \n    Find the total number of in
voices per country:\n"}, {"role": "assistant", "content": "SELECT COUNT(DISTINCT InvoiceId), BillingCountry
\nFROM Invoice"}, {"role": "user", "content": "  \n    Get the total number of invoices for each customer 
\n"}, {"role": "assistant", "content": "SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Cus
tomer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId"}, {"role": "user", "conten
t": "How many records are in table called customer"}, {"role": "assistant", "content": "SELECT COUNT(*) FRO
M customer"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"rol
e": "assistant", "content": "SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM Customer\nGROUP B
Y Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    Find all tracks
with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT Name \nFRO
M Track \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": "  \n    List all albums and their
corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT A.Title, ART.Name \nFROM Album A
S A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": "  \n    Find all invo
ices since 2010 and the total amount invoiced:\n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:25:06.953772241Z', 'message': {'role': 'assistan
t', 'content': "SELECT DISTINCT InvoiceId, Total\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'"}, 'done_
reason': 'stop', 'done': True, 'total_duration': 36279895479, 'load_duration': 15696126, 'prompt_eval_coun
t': 1026, 'prompt_eval_duration': 32747398000, 'eval_count': 23, 'eval_duration': 3369643000}
```

```
LLM Response: SELECT DISTINCT InvoiceId, Total
FROM Invoice
WHERE InvoiceDate >= '2010-01-01'
SELECT DISTINCT InvoiceId, Total
FROM Invoice
WHERE InvoiceDate >= '2010-01-01'
     InvoiceId  Total
0           84   1.98
1           85   1.98
2           86   3.96
3           87   6.94
4           88  17.91
..         ...    ...
324        408   3.96
325        409   5.94
326        410   8.91
327        411  13.86
328        412   1.99

[329 rows x 2 columns]
Info: Ollama parameters:
model=llama3.1: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 answers the question the user asked: '  \n    Find all invoices since 2010 and the total amount invoic
ed:\n'\n\nThe DataFrame was produced using this query: SELECT DISTINCT InvoiceId, Total\nFROM Invoice\nWHER
E InvoiceDate >= '2010-01-01'\n\nThe following is information about the resulting pandas DataFrame 'df': \n
Running df.dtypes gives:\n InvoiceId      int64\nTotal        float64\ndtype: object"}, {"role": "user", "c
ontent": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is
in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond w
ith only Python code. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:25:26.584815455Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='InvoiceId', y='Total')\nif le
n(df) == 1:\n    fig.update_layout(xaxis_title='', yaxis_title='')\nelse:\n    fig.update_layout(title_text
='Invoices since 2010 and Total Amount Invoiced')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': Tru
e, 'total_duration': 19518753219, 'load_duration': 81479807, 'prompt_eval_count': 282, 'prompt_eval_duratio
n': 8597364000, 'eval_count': 73, 'eval_duration': 10698815000}
```


Invoices since 2010 and Total Amount Invoiced
Total
0
5
10
15
20
25
InvoiceId
100
150
200
250
300
350
400

```
Out[23]: ("SELECT DISTINCT InvoiceId, Total\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'",
               InvoiceId  Total
          0           84   1.98
          1           85   1.98
          2           86   3.96
          3           87   6.94
          4           88  17.91
          ..         ...    ...
          324        408   3.96
          325        409   5.94
          326        410   8.91
          327        411  13.86
          328        412   1.99

          [329 rows x 2 columns],
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'InvoiceId=%{x}<br>Total=%{y}<extra></extra>',
                        'legendgroup': '',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array([ 84,  85,  86, ..., 410, 411, 412]),
                        'xaxis': 'x',
                        'y': array([ 1.98,  1.98,  3.96, ...,  8.91, 13.86,  1.99]),
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                         'legend': {'tracegroupgap': 0},
                         'margin': {'t': 60},
                         'template': '...',
                         'title': {'text': 'Invoices since 2010 and Total Amount Invoiced'},
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceId'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total'}}}
          }))
```

```
In [24]: 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 8, updating n_results = 8
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\n\nC
REATE TABLE Employee\n(\n    EmployeeId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    First
Name NVARCHAR(20)  NOT NULL,\n    Title NVARCHAR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIME,\n    
HireDate DATETIME,\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country
NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVAR
CHAR(60),\n    CONSTRAINT PK_Employee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Em
ployee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Customer\n(\n    Custo
merId INTEGER  NOT NULL,\n    FirstName NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    
Company NVARCHAR(80),\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Coun
try NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email 
NVARCHAR(60)  NOT NULL,\n    SupportRepId INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n
FOREIGN KEY (SupportRepId) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION 
\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON Customer (SupportRepId)\n\nCREATE TABLE Invoice\n(\n    Inv
oiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    Bill
ingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry
NVARCHAR(40),\n    BillingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_I
nvoice PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DE
LETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NUL
L,\n    InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NUL
L,\n    Quantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREI
GN KEY (InvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREI
GN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABL
E Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    Me
diaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER 
NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY 
(TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n
FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION 
\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE TABLE Artist\n(\n    ArtistId I
NTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Artist PRIMARY KEY  (ArtistId)\n)\n\nCREATE T
ABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK
_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (Playl
istId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId)
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\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 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 co
mment 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 bef
ore, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': 'what are t
```

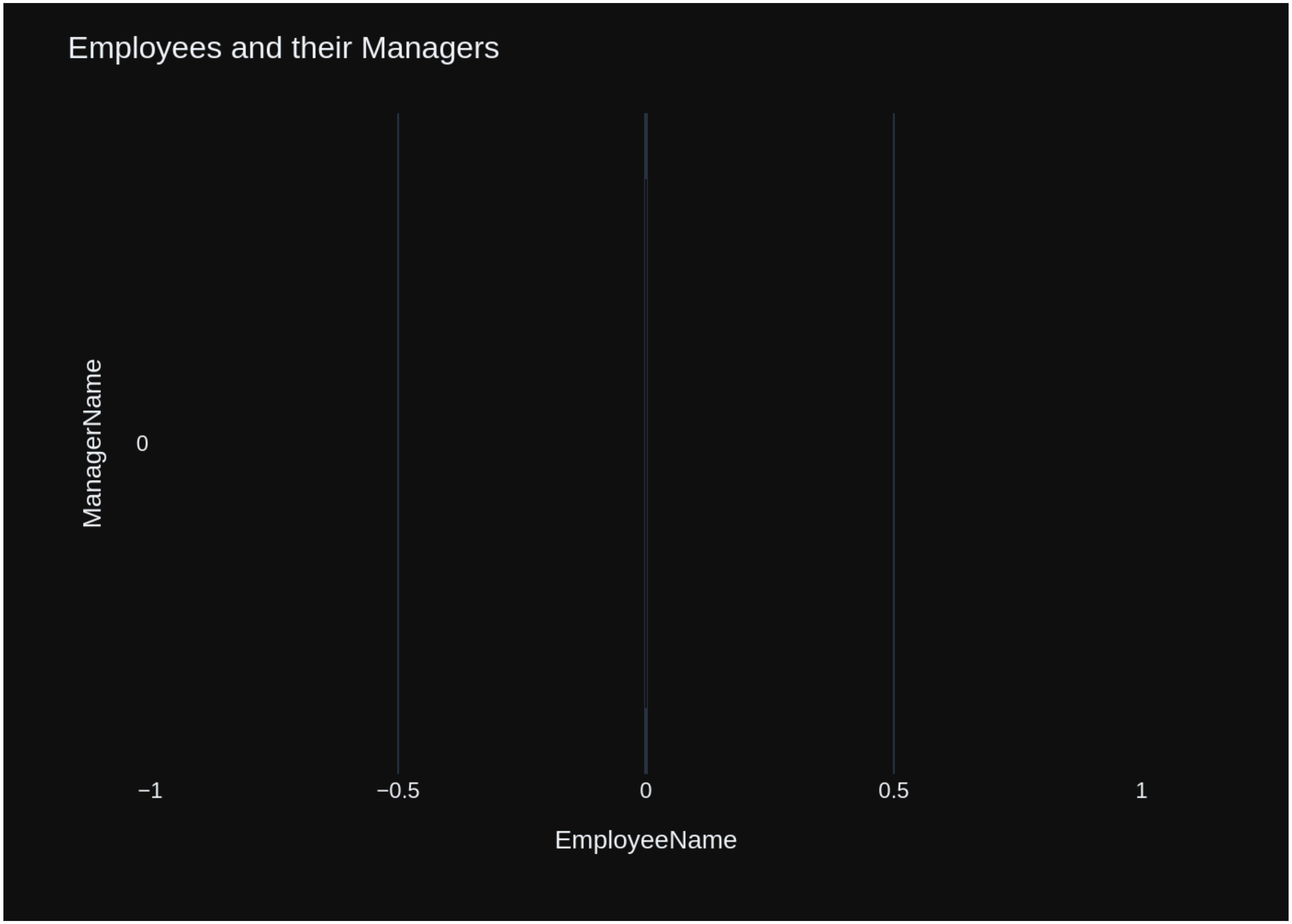
```
he top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(Cus
tomerId) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5'},
{'role': 'user', 'content': '  \n    Get the total number of invoices for each customer\n'}, {'role': 'assi
stant', 'content': 'SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoi
ce I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId'}, {'role': 'user', 'content': '  \n    Find al
l invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT DISTINCT
InvoiceId, Total\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'"}, {'role': 'user', 'content': '  \n    L
ist all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT A.Titl
e, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'conten
t': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT *
FROM Invoice WHERE Total > 10'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per
country:\n'}, {'role': 'assistant', 'content': 'SELECT COUNT(DISTINCT InvoiceId), BillingCountry\nFROM Invo
ice'}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assistant',
'content': 'SELECT COUNT(*) FROM customer'}, {'role': 'user', 'content': '  \n    Find all tracks with a na
me containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT Name \nFROM Track \nW
HERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': "  \n    List all employees and their reporti
ng manager's name (if any):\n"}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\n\nCREATE TABLE
Employee\n(\n    EmployeeId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    FirstName NVARCHA
R(20)  NOT NULL,\n    Title NVARCHAR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIME,\n    HireDate DA
TETIME,\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(4
0),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(6
0),\n    CONSTRAINT PK_Employee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Employee
(EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Customer\n(\n    CustomerId I
NTEGER  NOT NULL,\n    FirstName NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company
NVARCHAR(80),\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVAR
CHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR
(60)  NOT NULL,\n    SupportRepId INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FORE
IGN KEY (SupportRepId) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n
\nCREATE INDEX IFK_CustomerSupportRepId ON Customer (SupportRepId)\n\nCREATE TABLE Invoice\n(\n    InvoiceI
d INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAd
dress NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVAR
CHAR(40),\n    BillingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoi
ce PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE
NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n
```

```
InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Q
uantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (I
nvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (T
rackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Track\n
(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId
INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NUL
L,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (Track
Id),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n
FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIG
N KEY (MediaTypeId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n
CREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE TABLE Artist\n(\n    ArtistId INTEGER
NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Artist PRIMARY KEY  (ArtistId)\n)\n\nCREATE TABLE Pla
ylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_Playlis
tTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId)
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\t
ON DELETE NO ACTION ON UPDATE NO ACTION\n)\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 con
text is almost sufficient but requires knowledge of a specific string in a particular column, please genera
te 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, p
lease 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(CustomerI
d) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5"}, {"rol
e": "user", "content": "  \n    Get the total number of invoices for each customer\n"}, {"role": "assistan
t", "content": "SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I
ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId"}, {"role": "user", "content": "  \n    Find all inv
oices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT DISTINCT Invo
iceId, Total\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'"}, {"role": "user", "content": "  \n    List
all albums and their corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT A.Title, AR
T.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": "
\n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM In
voice WHERE Total > 10"}, {"role": "user", "content": "  \n    Find the total number of invoices per countr
y:\n"}, {"role": "assistant", "content": "SELECT COUNT(DISTINCT InvoiceId), BillingCountry\nFROM Invoice"},
{"role": "user", "content": "How many records are in table called customer"}, {"role": "assistant", "conten
t": "SELECT COUNT(*) FROM customer"}, {"role": "user", "content": "  \n    Find all tracks with a name cont
aining \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT Name \nFROM Track \nWHERE
LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": "  \n    List all employees and their reporting ma
nager's name (if any):\n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:26:14.500929876Z', 'message': {'role': 'assistan
t', 'content': "```sql\nSELECT E.FirstName + ' ' + E.LastName AS EmployeeName,\n       COALESCE(R.FirstName
```

```
+ ' ' + R.LastName, '--') AS ManagerName\nFROM Employee E\nLEFT JOIN Employee R ON E.ReportsTo = R.Employee
Id;\n```\n\nThis query will return all employees with their reporting manager's name. If an employee does n
ot have a reporting manager (i.e., `ReportsTo` is NULL), the manager's name will be '--'."}, 'done_reason':
'stop', 'done': True, 'total_duration': 47768456000, 'load_duration': 14953515, 'prompt_eval_count': 1026,
'prompt_eval_duration': 32819865000, 'eval_count': 97, 'eval_duration': 14791609000}
LLM Response: ```sql
SELECT E.FirstName + ' ' + E.LastName AS EmployeeName,
       COALESCE(R.FirstName + ' ' + R.LastName, '--') AS ManagerName
FROM Employee E
LEFT JOIN Employee R ON E.ReportsTo = R.EmployeeId;
```

This query will return all employees with their reporting manager's name. If an employee does not have a re
porting manager (i.e., `ReportsTo` is NULL), the manager's name will be '--'.
Info: Output from LLM: ```sql
SELECT E.FirstName + ' ' + E.LastName AS EmployeeName,
       COALESCE(R.FirstName + ' ' + R.LastName, '--') AS ManagerName
FROM Employee E
LEFT JOIN Employee R ON E.ReportsTo = R.EmployeeId;
```

This query will return all employees with their reporting manager's name. If an employee does not have a re
porting manager (i.e., `ReportsTo` is NULL), the manager's name will be '--'.
Extracted SQL: SELECT E.FirstName + ' ' + E.LastName AS EmployeeName,
       COALESCE(R.FirstName + ' ' + R.LastName, '--') AS ManagerName
FROM Employee E
LEFT JOIN Employee R ON E.ReportsTo = R.EmployeeId
SELECT E.FirstName + ' ' + E.LastName AS EmployeeName,
       COALESCE(R.FirstName + ' ' + R.LastName, '--') AS ManagerName
FROM Employee E
LEFT JOIN Employee R ON E.ReportsTo = R.EmployeeId
   EmployeeName ManagerName
0   0   --
1   0   0
2   0   0
3   0   0
4   0   0
5   0   0
6   0   0
7   0   0
Info: Ollama parameters:
model=llama3.1: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 answers the question the user asked: '  \n    List all employees and their reporting manager's name (i
f any):\n'\n\nThe DataFrame was produced using this query: SELECT E.FirstName + ' ' + E.LastName AS Employe
eName,\n       COALESCE(R.FirstName + ' ' + R.LastName, '--') AS ManagerName\nFROM Employee E\nLEFT JOIN Em
ployee R ON E.ReportsTo = R.EmployeeId\n\nThe following is information about the resulting pandas DataFrame
'df': \nRunning df.dtypes gives:\n EmployeeName     int64\nManagerName     object\ndtype: object"}, {"rol
e": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assu
me the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indic
ator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:26:41.738621615Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='EmployeeName', y='ManagerNam
e', barmode='group')\n\nif len(df) == 1:\n    fig.update_layout(\n        title='Single Employee',\n
annotations=[dict(text=df['ManagerName'].values[0], x=0.5, y=-0.25, showarrow=False, xref='paper', yref='pa
per')]\n    )\nelse:\n    fig.update_layout(title='Employees and their Managers')\n\nfig.show()\n```"}, 'do
ne_reason': 'stop', 'done': True, 'total_duration': 27055211356, 'load_duration': 14879629, 'prompt_eval_co
unt': 336, 'prompt_eval_duration': 10334213000, 'eval_count': 112, 'eval_duration': 16576142000}
```


Employees and their Managers
ManagerName
0
−1
−0.5
0
0.5
1
EmployeeName

Out[24]:
```
("SELECT E.FirstName + ' ' + E.LastName AS EmployeeName,\n       COALESCE(R.FirstName + ' ' + R.LastName,
'--') AS ManagerName\nFROM Employee E\nLEFT JOIN Employee R ON E.ReportsTo = R.EmployeeId",
    EmployeeName ManagerName
 0             0          --
 1             0           0
 2             0           0
 3             0           0
 4             0           0
 5             0           0
 6             0           0
 7             0           0,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'EmployeeName=%{x}<br>ManagerName=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'h',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([0, 0, 0, 0, 0, 0, 0, 0]),
               'xaxis': 'x',
               'y': array(['--', 0, 0, 0, 0, 0, 0, 0], dtype=object),
               'yaxis': 'y'}],
     'layout': {'barmode': 'group',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'title': {'text': 'Employees and their Managers'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'EmployeeName'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'ManagerName'}}}
 }))
```

In [25]:
```
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 9, updating n_results = 9
```

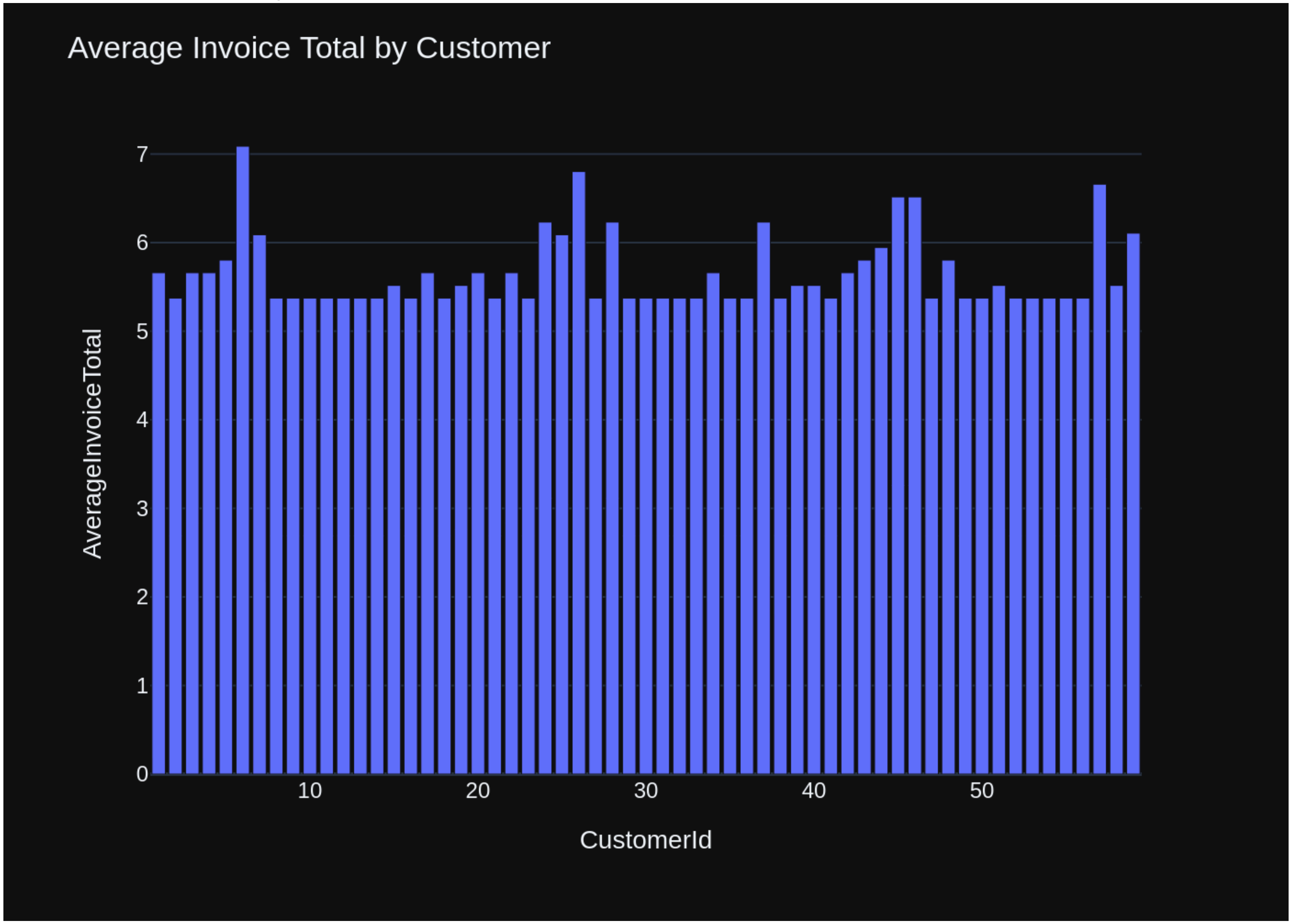
```
SQL Prompt: [{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to
answer the question. Your response should ONLY be based on the given context and follow the response guidel
ines and format instructions. \n===Tables \nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nC
REATE INDEX IFK_InvoiceLineInvoiceId ON InvoiceLine (InvoiceId)\n\nCREATE TABLE Invoice\n(\n    InvoiceId I
NTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddre
ss NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHA
R(40),\n    BillingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice 
PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    Inv
oiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quan
tity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (Invo
iceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (Trac
kId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_Invoic
eLineTrackId ON InvoiceLine (TrackId)\n\nCREATE INDEX IFK_CustomerSupportRepId ON Customer (SupportRepId)\n
\nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    FirstName NVARCHAR(40)  NOT NULL,\n    L
astName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Address NVARCHAR(70),\n    City NVARCHAR(4
0),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(2
4),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    SupportRepId INTEGER,\n    CONSTRAINT PK
_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId) REFERENCES Employee (EmployeeId) \n\t
\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)
\n\nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId I
NTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Millise
conds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Tra
ck PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION 
ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPD
ATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION 
ON UPDATE NO ACTION\n)\n\nCREATE TABLE Employee\n(\n    EmployeeId INTEGER  NOT NULL,\n    LastName NVARCHA
R(20)  NOT NULL,\n    FirstName NVARCHAR(20)  NOT NULL,\n    Title NVARCHAR(30),\n    ReportsTo INTEGER,\n 
BirthDate DATETIME,\n    HireDate DATETIME,\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State N
VARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVA
RCHAR(24),\n    Email NVARCHAR(60),\n    CONSTRAINT PK_Employee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY
(ReportsTo) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Respons
e 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 sp
ecific string in a particular column, please generate an intermediate SQL query to find the distinct string
s in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is 
insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If
the question has been asked and answered before, please repeat the answer exactly as it was given before. 
\n"}, {'role': 'user', 'content': '  \n    Get the total number of invoices for each customer\n'}, {'role':
'assistant', 'content': 'SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN
Invoice I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId'}, {'role': 'user', 'content': '  \n    Fi
nd all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT DIS
```

```
TINCT InvoiceId, Total\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'"}, {'role': 'user', 'content': '
\n    Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT COUNT(DI
STINCT InvoiceId), BillingCountry\nFROM Invoice'}, {'role': 'user', 'content': '  \n    List all invoices w
ith a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM Invoice WHERE Total > 10'},
{'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assistant', 'conten
t': 'SELECT COUNT(*) FROM customer'}, {'role': 'user', 'content': 'what are the top 5 countries that custom
ers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(CustomerId) AS NumberOfCustomers
\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content':
"  \n    List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'conten
t': "SELECT E.FirstName + ' ' + E.LastName AS EmployeeName,\n       COALESCE(R.FirstName + ' ' + R.LastNam
e, '--') AS ManagerName\nFROM Employee E\nLEFT JOIN Employee R ON E.ReportsTo = R.EmployeeId"}, {'role': 'u
ser', 'content': '  \n    Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'a
ssistant', 'content': "SELECT Name \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'cont
ent': '  \n    List all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content':
'SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId'}, {'role':
'user', 'content': '  \n    Get the average invoice total for each customer:\n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE INDEX
IFK_InvoiceLineInvoiceId ON InvoiceLine (InvoiceId)\n\nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT
NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR
(70),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n
BillingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY
(InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UP
DATE NO ACTION\n)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    InvoiceId INTEG
ER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quantity INTEGER
NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (InvoiceId) REFEREN
CES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCE
S Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId O
N InvoiceLine (TrackId)\n\nCREATE INDEX IFK_CustomerSupportRepId ON Customer (SupportRepId)\n\nCREATE TABLE
Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    FirstName NVARCHAR(40)  NOT NULL,\n    LastName NVARCHA
R(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State N
VARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVA
RCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    SupportRepId INTEGER,\n    CONSTRAINT PK_Customer PRIMAR
Y KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO AC
TION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\n\nCREATE TABLE T
rack\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    Media
TypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT
NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (Tr
ackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO
N,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n
FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\n)\n\nCREATE TABLE Employee\n(\n    EmployeeId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n
FirstName NVARCHAR(20)  NOT NULL,\n    Title NVARCHAR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIM
E,\n    HireDate DATETIME,\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n
Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Em
ail NVARCHAR(60),\n    CONSTRAINT PK_Employee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFER
ENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response Guidelines \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
particular column, please generate an intermediate SQL query to find the distinct strings in that column. P
repend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, pleas
e explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has b
een asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "use
r", "content": "  \n    Get the total number of invoices for each customer\n"}, {"role": "assistant", "cont
ent": "SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I ON C.Cus
tomerId = I.CustomerId \nGROUP BY C.CustomerId"}, {"role": "user", "content": "  \n    Find all invoices si
nce 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT DISTINCT InvoiceId, T
otal\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'"}, {"role": "user", "content": "  \n    Find the tota
l number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT COUNT(DISTINCT InvoiceId),
BillingCountry\nFROM Invoice"}, {"role": "user", "content": "  \n    List all invoices with a total exceedi
ng $10:\n"}, {"role": "assistant", "content": "SELECT * FROM Invoice WHERE Total > 10"}, {"role": "user",
"content": "How many records are in table called customer"}, {"role": "assistant", "content": "SELECT COUNT
(*) FROM customer"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"},
{"role": "assistant", "content": "SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM Customer\nGR
OUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    List all e
mployees and their reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT E.First
Name + ' ' + E.LastName AS EmployeeName,\n       COALESCE(R.FirstName + ' ' + R.LastName, '--') AS ManagerN
ame\nFROM Employee E\nLEFT JOIN Employee R ON E.ReportsTo = R.EmployeeId"}, {"role": "user", "content": "
\n    Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "conten
t": "SELECT Name \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": "  \n    Lis
t all albums and their corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT A.Title,
ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content":
"  \n    Get the average invoice total for each customer:\n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:27:53.660779356Z', 'message': {'role': 'assistan
t', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerI
d'}, 'done_reason': 'stop', 'done': True, 'total_duration': 71775875210, 'load_duration': 16984094, 'prompt
_eval_count': 2040, 'prompt_eval_duration': 67777112000, 'eval_count': 21, 'eval_duration': 3362974000}
```

```
LLM Response: SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal
FROM Invoice
GROUP BY CustomerId
SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal
FROM Invoice
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=llama3.1: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 answers the question the user asked: '  \n    Get the average invoice total for each customer:\n'\n\nT
he DataFrame was produced using this query: SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM Inv
oice \nGROUP BY CustomerId\n\nThe following is information about the resulting pandas DataFrame 'df': \nRun
ning df.dtypes gives:\n CustomerId               int64\nAverageInvoiceTotal    float64\ndtype: object"},
{"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe?
Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an I
ndicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:28:13.894543128Z', 'message': {'role': 'assistan
t', 'content': '```python\nimport plotly.express as px\n\nfig = px.bar(df, x=\'CustomerId\', y=\'AverageInv
oiceTotal\')\nif len(df) == 1:\n    fig.update_layout(\n        title_text="Indicator: Average Invoice Tota
l",\n        yaxis_title="Value"\n    )\nelse:\n    fig.update_layout(title_text="Average Invoice Total by
```

```
Customer")\n\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 20126134169, 'load_
duration': 16333411, 'prompt_eval_count': 274, 'prompt_eval_duration': 8367267000, 'eval_count': 79, 'eval_
duration': 11610915000}
```



```
Out[25]: ('SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM Invoice \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': '...',
                       'title': {'text': 'Average Invoice Total by Customer'},
                       'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                       'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'AverageInvoiceTotal'}}}
        }))
```

In [26]:

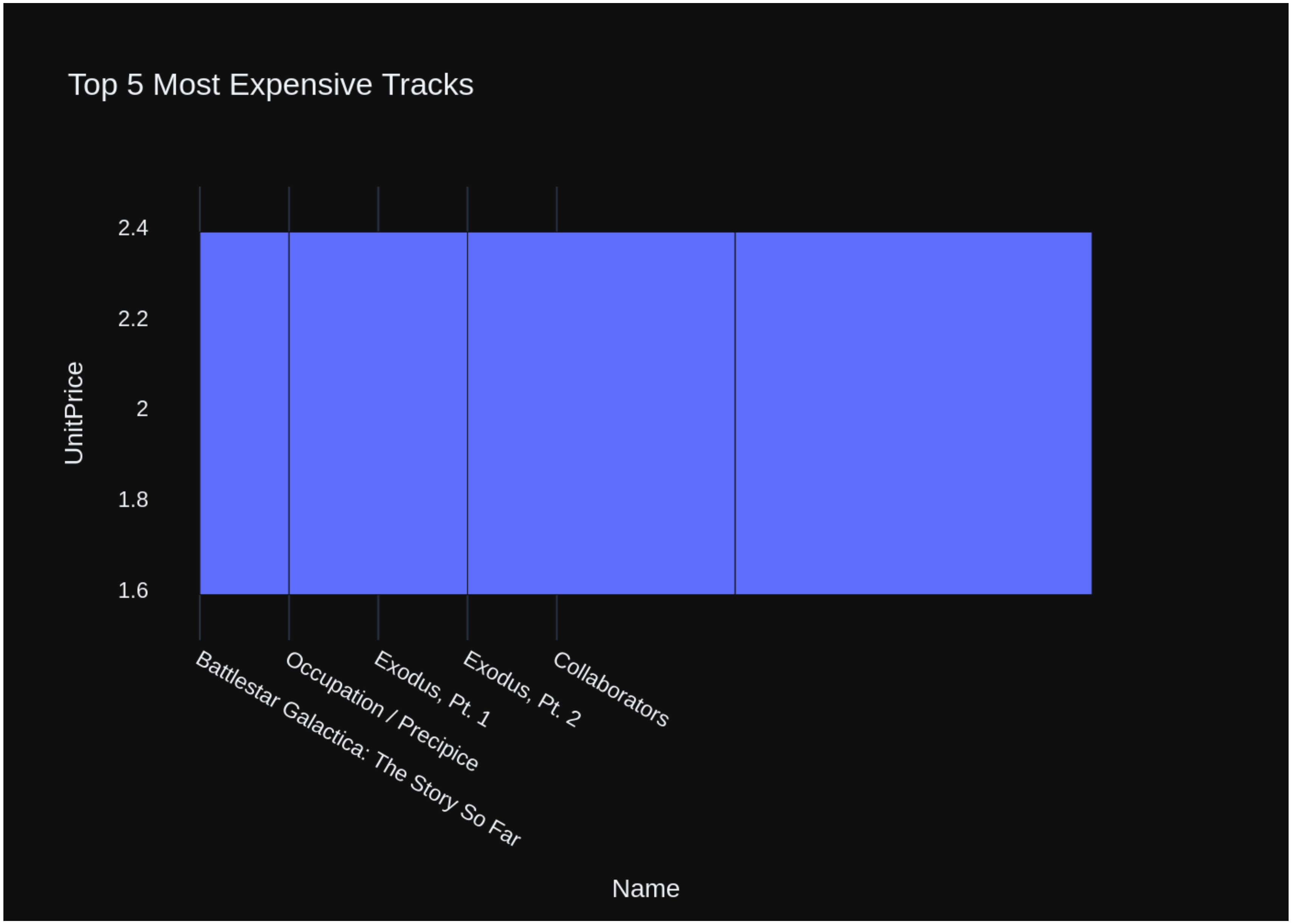
```python
question = """
    Find the top 5 most expensive tracks (based on unit price):
"""

vn.ask(question=question)
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name
NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n
Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,
2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album
(AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (Genre
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (Med
iaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_TrackAlbumId ON Track (Album
Id)\n\nCREATE INDEX IFK_TrackGenreId ON Track (GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON Playlis
tTrack (TrackId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON InvoiceLine (TrackId)\n\nCREATE INDEX IFK_TrackM
ediaTypeId ON Track (MediaTypeId)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n
InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Q
uantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (I
nvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (T
rackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Playlis
tTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTra
ck PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t
\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON D
ELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_AlbumArtistId ON Album (ArtistId)\n\nCREATE TABL
E Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT N
ULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (Artist
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response Guidelines \n1. If the provided context
is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the pro
vided context is almost sufficient but requires knowledge of a specific string in a particular column, plea
se generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a
comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't b
e generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered b
efore, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': '  \n
Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SEL
ECT Name \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': '  \n    List all in
voices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM Invoice WHERE Total
> 10'}, {'role': 'user', 'content': '  \n    List all albums and their corresponding artist names  \n'},
{'role': 'assistant', 'content': 'SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.Arti
stId = ART.ArtistId'}, {'role': 'user', 'content': '  \n    Find all invoices since 2010 and the total amou
nt invoiced:\n'}, {'role': 'assistant', 'content': "SELECT DISTINCT InvoiceId, Total\nFROM Invoice\nWHERE I
nvoiceDate >= '2010-01-01'"}, {'role': 'user', 'content': '  \n    Get the average invoice total for each c
ustomer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM
Invoice \nGROUP BY CustomerId'}, {'role': 'user', 'content': 'what are the top 5 countries that customers c
ome from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM
Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n
Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT C.Customer
```

```
Id, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId
\nGROUP BY C.CustomerId'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per count
ry:\n'}, {'role': 'assistant', 'content': 'SELECT COUNT(DISTINCT InvoiceId), BillingCountry\nFROM Invoic
e'}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assistant', 'c
ontent': 'SELECT COUNT(*) FROM customer'}, {'role': 'user', 'content': "  \n    List all employees and thei
r reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': "SELECT E.FirstName + ' ' + E.La
stName AS EmployeeName,\n       COALESCE(R.FirstName + ' ' + R.LastName, '--') AS ManagerName\nFROM Employe
e E\nLEFT JOIN Employee R ON E.ReportsTo = R.EmployeeId"}, {'role': 'user', 'content': '  \n    Find the to
p 5 most expensive tracks (based on unit price):\n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(2
00)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Compose
r NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT
NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumI
d) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n
\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaType
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_TrackAlbumId ON Track (AlbumId)\n
\nCREATE INDEX IFK_TrackGenreId ON Track (GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON PlaylistTrac
k (TrackId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON InvoiceLine (TrackId)\n\nCREATE INDEX IFK_TrackMediaT
ypeId ON Track (MediaTypeId)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    Invo
iceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quant
ity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (Invoi
ceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (Track
Id) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE PlaylistTra
ck\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack P
RIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE
NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_AlbumArtistId ON Album (ArtistId)\n\nCREATE TABLE Albu
m\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n
CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\t
ON DELETE NO ACTION ON UPDATE NO ACTION\n)\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 con
text is almost sufficient but requires knowledge of a specific string in a particular column, please genera
te 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, p
```

```
lease repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "  \n    Find all
tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT Nam
e \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": "  \n    List all invoices
with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM Invoice WHERE Total > 1
0"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist names  \n"}, {"rol
e": "assistant", "content": "SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId
= ART.ArtistId"}, {"role": "user", "content": "  \n    Find all invoices since 2010 and the total amount in
voiced:\n"}, {"role": "assistant", "content": "SELECT DISTINCT InvoiceId, Total\nFROM Invoice\nWHERE Invoic
eDate >= '2010-01-01'"}, {"role": "user", "content": "  \n    Get the average invoice total for each custom
er:\n"}, {"role": "assistant", "content": "SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM Invo
ice \nGROUP BY CustomerId"}, {"role": "user", "content": "what are the top 5 countries that customers come
from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM Cus
tomer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    G
et the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT C.CustomerI
d, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId \n
GROUP BY C.CustomerId"}, {"role": "user", "content": "  \n    Find the total number of invoices per countr
y:\n"}, {"role": "assistant", "content": "SELECT COUNT(DISTINCT InvoiceId), BillingCountry\nFROM Invoice"},
{"role": "user", "content": "How many records are in table called customer"}, {"role": "assistant", "conten
t": "SELECT COUNT(*) FROM customer"}, {"role": "user", "content": "  \n    List all employees and their rep
orting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT E.FirstName + ' ' + E.LastNam
e AS EmployeeName,\n       COALESCE(R.FirstName + ' ' + R.LastName, '--') AS ManagerName\nFROM Employee E\n
LEFT JOIN Employee R ON E.ReportsTo = R.EmployeeId"}, {"role": "user", "content": "  \n    Find the top 5 m
ost expensive tracks (based on unit price):\n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:29:19.845787894Z', 'message': {'role': 'assistan
t', 'content': 'SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY UnitPrice DESC \nLIMIT 5'}, 'done_reas
on': 'stop', 'done': True, 'total_duration': 65829314854, 'load_duration': 16694175, 'prompt_eval_count': 1
858, 'prompt_eval_duration': 61076210000, 'eval_count': 20, 'eval_duration': 2968982000}
LLM Response: SELECT Track.Name, UnitPrice
FROM Track
ORDER BY UnitPrice DESC
LIMIT 5
SELECT Track.Name, UnitPrice
FROM Track
ORDER BY UnitPrice DESC
LIMIT 5
                                     Name  UnitPrice
0  Battlestar Galactica: The Story So Far       1.99
1                 Occupation / Precipice       1.99
2                          Exodus, Pt. 1       1.99
3                          Exodus, Pt. 2       1.99
4                          Collaborators       1.99
```

```
Info: Ollama parameters:
model=llama3.1: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 answers the question the user asked: '  \n    Find the top 5 most expensive tracks (based on unit pric
e):\n'\n\nThe DataFrame was produced using this query: SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY
UnitPrice DESC \nLIMIT 5\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunni
ng df.dtypes gives:\n Name          object\nUnitPrice    float64\ndtype: object"}, {"role": "user", "conten
t": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a
pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with o
nly Python code. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:29:40.476215767Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nif len(df) == 1:\n    fig = px.treemap(df, values
='UnitPrice', path=['Name'], title='Most Expensive Track')\nelse:\n    fig = px.bar(df, x='Name', y='UnitPr
ice', orientation='h', title='Top 5 Most Expensive Tracks')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'd
one': True, 'total_duration': 20520176827, 'load_duration': 81196935, 'prompt_eval_count': 276, 'prompt_eva
l_duration': 8431677000, 'eval_count': 81, 'eval_duration': 11868251000}
```


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

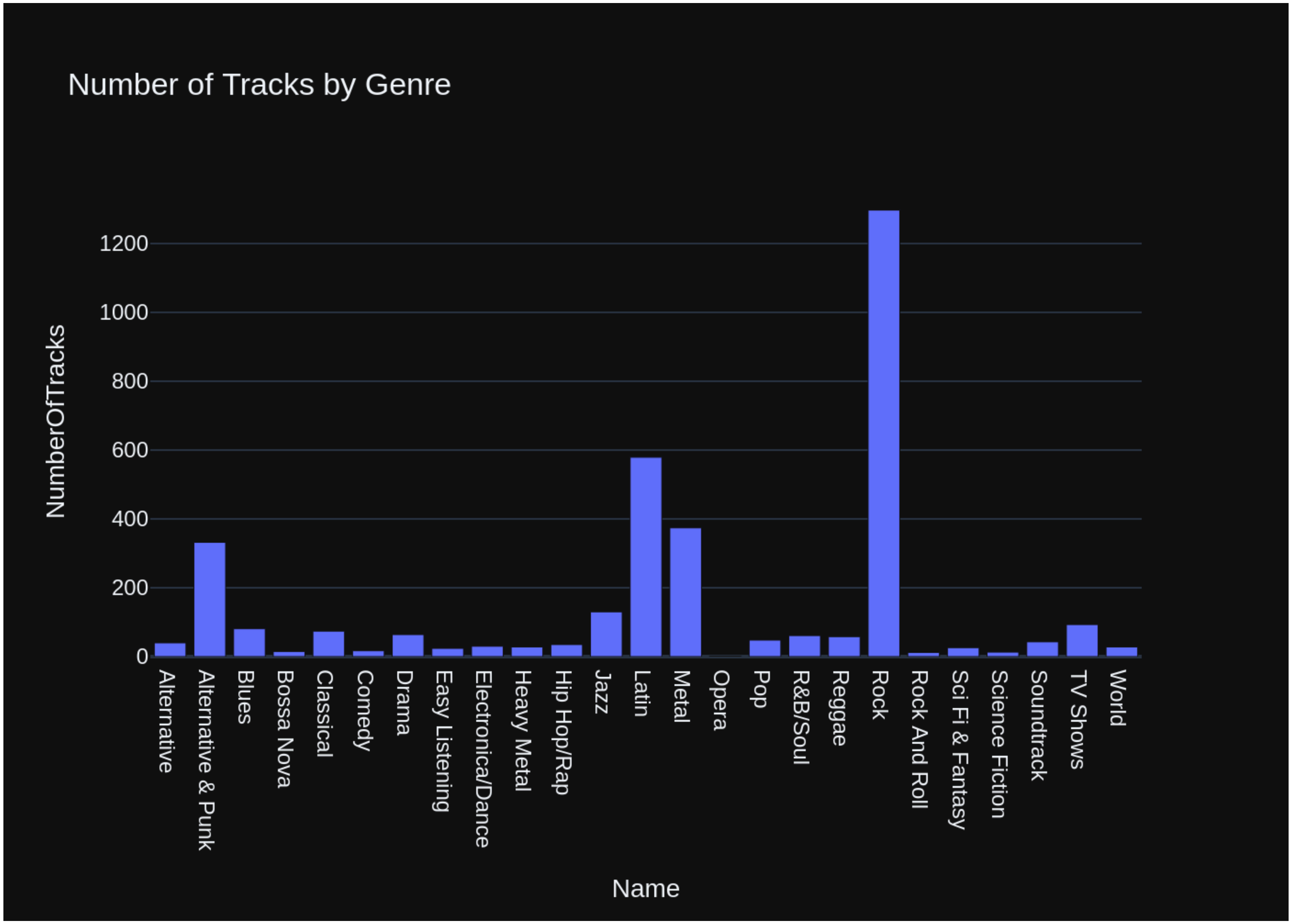
Out[26]:
```
('SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY UnitPrice DESC \nLIMIT 5',
                                       Name  UnitPrice
 0  Battlestar Galactica: The Story So Far       1.99
 1                 Occupation / Precipice       1.99
 2                         Exodus, Pt. 1       1.99
 3                         Exodus, Pt. 2       1.99
 4                         Collaborators       1.99,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'Name=%{x}<br>UnitPrice=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'h',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Battlestar Galactica: The Story So Far', 'Occupation / Precipice',
                           'Exodus, Pt. 1', 'Exodus, Pt. 2', 'Collaborators'], dtype=object),
               'xaxis': 'x',
               'y': array([1.99, 1.99, 1.99, 1.99, 1.99]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                '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 [27]:
```
question = """
    List all genres and the number of tracks in each genre:
"""

vn.ask(question=question)
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name
NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n
Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,
2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album
(AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (Genre
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (Med
iaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_TrackGenreId ON Track (Genre
Id)\n\nCREATE TABLE Genre\n(\n    GenreId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Ge
nre PRIMARY KEY  (GenreId)\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON PlaylistTrack (TrackId)\n\nCREATE
INDEX IFK_TrackAlbumId ON Track (AlbumId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON Track (MediaTypeId)\n\nCR
EATE TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEG
ER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artis
t (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE PlaylistTrack\n(\n    Playlis
tId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (Playl
istId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION O
N UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDA
TE NO ACTION\n)\n\nCREATE INDEX IFK_AlbumArtistId ON Album (ArtistId)\n\nCREATE TABLE Playlist\n(\n    Play
listId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Playlist PRIMARY KEY  (PlaylistId)\n)
\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query wi
thout any explanations for the question. \n2. If the provided context is almost sufficient but requires kno
wledge of a specific string in a particular column, please generate an intermediate SQL query to find the d
istinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provid
ed context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant tab
le(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was
given before. \n"}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive tracks (based on uni
t price):\n'}, {'role': 'assistant', 'content': 'SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY UnitP
rice DESC \nLIMIT 5'}, {'role': 'user', 'content': '  \n    List all albums and their corresponding artist
names  \n'}, {'role': 'assistant', 'content': 'SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS
ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': '  \n    Find all tracks with a name contai
ning "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT Name \nFROM Track \nWHERE LOWE
R(Name) LIKE '%what%'"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come fro
m?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM Custom
er\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    List
all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM Invoice WHERE
Total > 10'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per country:\n'}, {'ro
le': 'assistant', 'content': 'SELECT COUNT(DISTINCT InvoiceId), BillingCountry\nFROM Invoice'}, {'role': 'u
ser', 'content': '  \n    Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'co
ntent': 'SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I ON C.C
ustomerId = I.CustomerId \nGROUP BY C.CustomerId'}, {'role': 'user', 'content': '  \n    Find all invoices
since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT DISTINCT InvoiceId,
Total\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'"}, {'role': 'user', 'content': 'How many records are
in table called customer'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customer'}, {'role': 'us
er', 'content': '  \n    Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'conte
nt': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerId'}, {'role':
'user', 'content': '  \n    List all genres and the number of tracks in each genre:\n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(2
00)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Compose
r NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT
NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumI
d) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n
\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaType
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_TrackGenreId ON Track (GenreId)\n
\nCREATE TABLE Genre\n(\n    GenreId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Genre P
RIMARY KEY  (GenreId)\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON PlaylistTrack (TrackId)\n\nCREATE INDE
X IFK_TrackAlbumId ON Track (AlbumId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON Track (MediaTypeId)\n\nCREATE
TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  N
OT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (Ar
tistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId I
NTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistI
d, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UP
DATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION\n)\n\nCREATE INDEX IFK_AlbumArtistId ON Album (ArtistId)\n\nCREATE TABLE Playlist\n(\n    Playlist
Id INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Playlist PRIMARY KEY  (PlaylistId)\n)\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 knowled
ge of a specific string in a particular column, please generate an intermediate SQL query to find the disti
nct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided c
ontext 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 gi
ven before. \n"}, {"role": "user", "content": "  \n    Find the top 5 most expensive tracks (based on unit
price):\n"}, {"role": "assistant", "content": "SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY UnitPri
ce DESC \nLIMIT 5"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist na
mes  \n"}, {"role": "assistant", "content": "SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS AR
T ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": "  \n    Find all tracks with a name containi
ng \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT Name \nFROM Track \nWHERE LOWE
```

```
R(Name) LIKE '%what%'"}, {"role": "user", "content": "what are the top 5 countries that customers come fro
m?"}, {"role": "assistant", "content": "SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM Custom
er\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    List
all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM Invoice WHERE
Total > 10"}, {"role": "user", "content": "  \n    Find the total number of invoices per country:\n"}, {"ro
le": "assistant", "content": "SELECT COUNT(DISTINCT InvoiceId), BillingCountry\nFROM Invoice"}, {"role": "u
ser", "content": "  \n    Get the total number of invoices for each customer\n"}, {"role": "assistant", "co
ntent": "SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I ON C.C
ustomerId = I.CustomerId \nGROUP BY C.CustomerId"}, {"role": "user", "content": "  \n    Find all invoices
since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT DISTINCT InvoiceId,
Total\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'"}, {"role": "user", "content": "How many records are
in table called customer"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customer"}, {"role": "us
er", "content": "  \n    Get the average invoice total for each customer:\n"}, {"role": "assistant", "conte
nt": "SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerId"}, {"role":
"user", "content": "  \n    List all genres and the number of tracks in each genre:\n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:30:43.311118262Z', 'message': {'role': 'assistan
t', 'content': 'SELECT G.Name, COUNT(T.TrackId) AS NumberOfTracks \nFROM Genre G \nJOIN Track T ON G.GenreI
d = T.GenreId \nGROUP BY G.Name'}, 'done_reason': 'stop', 'done': True, 'total_duration': 62668104751, 'loa
d_duration': 21494862, 'prompt_eval_count': 1699, 'prompt_eval_duration': 55473145000, 'eval_count': 36, 'e
val_duration': 5485311000}
LLM Response: SELECT G.Name, COUNT(T.TrackId) AS NumberOfTracks
FROM Genre G
JOIN Track T ON G.GenreId = T.GenreId
GROUP BY G.Name
SELECT G.Name, COUNT(T.TrackId) AS NumberOfTracks
FROM Genre G
JOIN Track T ON G.GenreId = T.GenreId
GROUP BY G.Name
                  Name  NumberOfTracks
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=llama3.1: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 answers the question the user asked: '  \n    List all genres and the number of tracks in each genr
e:\n'\n\nThe DataFrame was produced using this query: SELECT G.Name, COUNT(T.TrackId) AS NumberOfTracks \nF
ROM Genre G \nJOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name\n\nThe following is information about
the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name              object\nNumberOfTracks
int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the r
esults of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value i
n the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- ju
st the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:31:04.528996061Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Name', y='NumberOfTracks', ti
tle='Number of Tracks by Genre')\nfig.show()\n\n# if there's only one row in the dataframe\nif len(df) ==
1:\n    fig = px.funnel(df, values='NumberOfTracks', names='Name')\n    fig.show()\n```"}, 'done_reason':
'stop', 'done': True, 'total_duration': 21101431763, 'load_duration': 80997634, 'prompt_eval_count': 304,
'prompt_eval_duration': 9284092000, 'eval_count': 79, 'eval_duration': 11596625000}
```


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

```
Out[27]: ('SELECT G.Name, COUNT(T.TrackId) AS NumberOfTracks \nFROM Genre G \nJOIN Track T ON G.GenreId = T.GenreId
         \nGROUP BY G.Name',
                           Name  NumberOfTracks
          0          Alternative              40
          1   Alternative & Punk             332
          2                Blues              81
          3           Bossa Nova              15
          4            Classical              74
          5               Comedy              17
          6                Drama              64
          7       Easy Listening              24
          8    Electronica/Dance              30
          9          Heavy Metal              28
          10         Hip Hop/Rap              35
          11                Jazz             130
          12               Latin             579
          13               Metal             374
          14               Opera               1
          15                 Pop              48
          16            R&B/Soul              61
          17              Reggae              58
          18                Rock            1297
          19       Rock And Roll              12
          20    Sci Fi & Fantasy              26
          21     Science Fiction              13
          22          Soundtrack              43
          23            TV Shows              93
          24               World              28,
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'Name=%{x}<br>NumberOfTracks=%{y}<extra></extra>',
                        'legendgroup': '',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array(['Alternative', 'Alternative & Punk', 'Blues', 'Bossa Nova', 'Classical',
                                    'Comedy', 'Drama', 'Easy Listening', 'Electronica/Dance', 'Heavy Metal',
                                    'Hip Hop/Rap', 'Jazz', 'Latin', 'Metal', 'Opera', 'Pop', 'R&B/Soul',
                                    'Reggae', 'Rock', 'Rock And Roll', 'Sci Fi & Fantasy',
                                    'Science Fiction', 'Soundtrack', 'TV Shows', 'World'], dtype=object),
                        'xaxis': 'x',
                        'y': array([  40,  332,   81,   15,   74,   17,   64,   24,   30,   28,   35,  130,
                                     579,  374,    1,   48,   61,   58, 1297,   12,   26,   13,   43,   93,
                                      28]),
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                         'legend': {'tracegroupgap': 0},
                         'template': '...',
                         'title': {'text': 'Number of Tracks by Genre'},
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'NumberOfTracks'}}}
          })
```

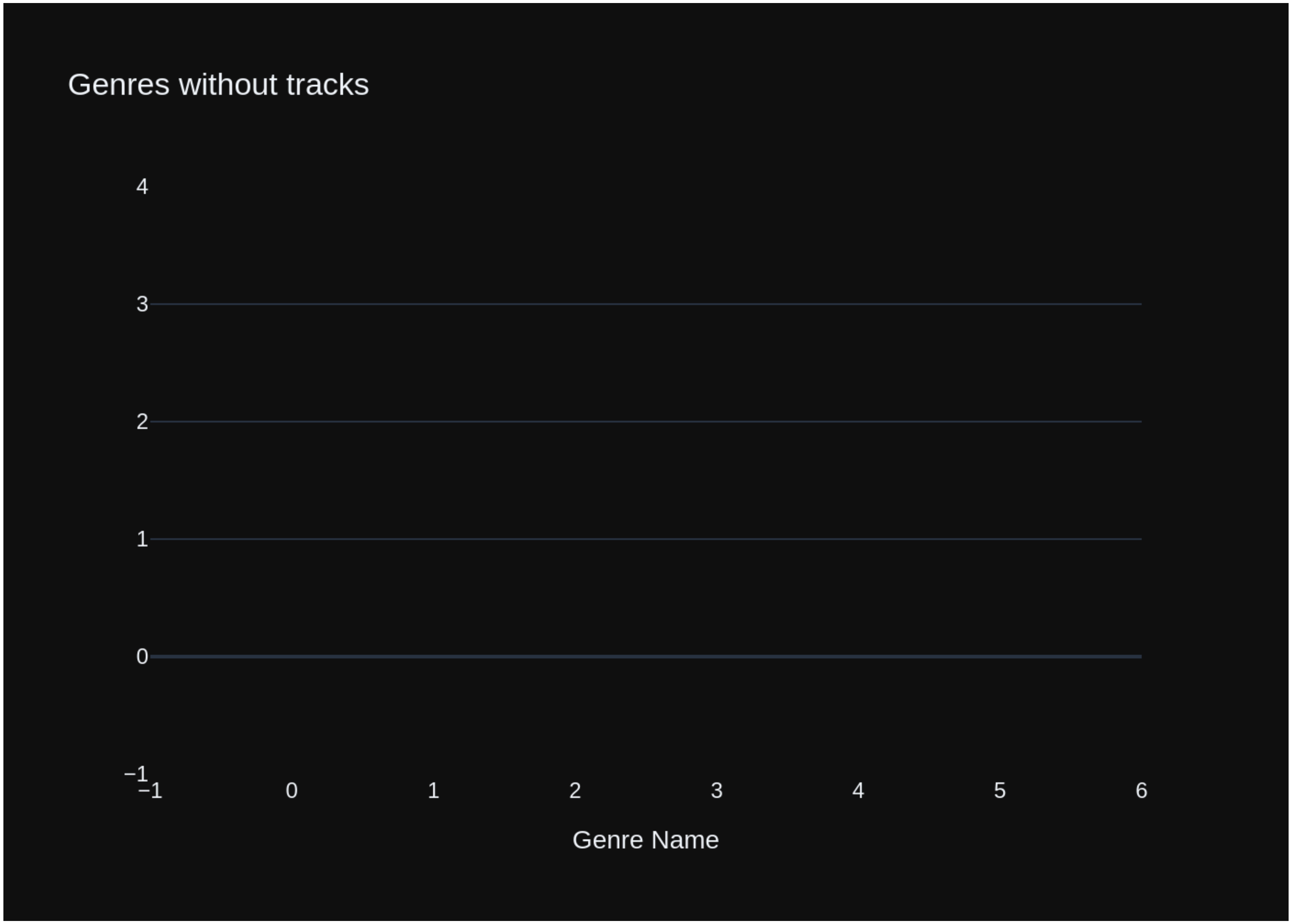
In [28]:
```python
question = """
    Get all genres that do not have any tracks associated with them:
"""

vn.ask(question=question)
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON Track (GenreId)\n\nCREATE TABL
E Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    Me
diaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER 
NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY 
(TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n
FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION 
\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON PlaylistTrack (TrackId)\n\nCREATE INDEX IFK_TrackAlbumId ON
Track (AlbumId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON Track (MediaTypeId)\n\nCREATE TABLE Genre\n(\n    G
enreId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Genre PRIMARY KEY  (GenreId)\n)\n\nCR
EATE TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEG
ER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artis
t (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_AlbumArtistId ON Album (A
rtistId)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NUL
L,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFER
ENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFER
ENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Artist\n(\n    Artis
tId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Artist PRIMARY KEY  (ArtistId)\n)\n\n===
Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without a
ny explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge 
of a specific string in a particular column, please generate an intermediate SQL query to find the distinct
strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided conte
xt 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 b
efore. \n"}, {'role': 'user', 'content': '  \n    List all genres and the number of tracks in each genr
e:\n'}, {'role': 'assistant', 'content': 'SELECT G.Name, COUNT(T.TrackId) AS NumberOfTracks \nFROM Genre G 
\nJOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name'}, {'role': 'user', 'content': '  \n    Find the 
top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT Track.Nam
e, UnitPrice \nFROM Track \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 N
ame \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': '  \n    List all albums 
and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT A.Title, ART.Name \nFR
OM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': '  \n    Fin
d all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT DIST
INCT InvoiceId, Total\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'"}, {'role': 'user', 'content': '  \n
List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM Invoice 
WHERE Total > 10'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, 
{'role': 'assistant', 'content': 'SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM Customer\nGR
OUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': "  \n    List all e
mployees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': "SELECT E.First
```

```
Name + ' ' + E.LastName AS EmployeeName,\n       COALESCE(R.FirstName + ' ' + R.LastName, '--') AS ManagerN
ame\nFROM Employee E\nLEFT JOIN Employee R ON E.ReportsTo = R.EmployeeId"}, {'role': 'user', 'content': 'Ho
w many records are in table called customer'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM custo
mer'}, {'role': 'user', 'content': '  \n    Get the average invoice total for each customer:\n'}, {'role':
'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY Cus
tomerId'}, {'role': 'user', 'content': '  \n    Get all genres that do not have any tracks associated with
them:\n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON Track (GenreId)\n\nCREATE TABLE Track\n
(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId
INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NUL
L,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (Track
Id),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n
FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIG
N KEY (MediaTypeId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n
CREATE INDEX IFK_PlaylistTrackTrackId ON PlaylistTrack (TrackId)\n\nCREATE INDEX IFK_TrackAlbumId ON Track
(AlbumId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON Track (MediaTypeId)\n\nCREATE TABLE Genre\n(\n    GenreId
INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Genre PRIMARY KEY  (GenreId)\n)\n\nCREATE TA
BLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT
NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (Artis
tId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_AlbumArtistId ON Album (ArtistId)
\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    C
ONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Pla
ylist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Tra
ck (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Artist\n(\n    ArtistId INTEG
ER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Artist PRIMARY KEY  (ArtistId)\n)\n\n===Response
Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explan
ations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a spec
ific string in a particular column, please generate an intermediate SQL query to find the distinct strings
in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is in
sufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If t
he 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"}, {"r
ole": "assistant", "content": "SELECT G.Name, COUNT(T.TrackId) AS NumberOfTracks \nFROM Genre G \nJOIN Trac
k T ON G.GenreId = T.GenreId \nGROUP BY G.Name"}, {"role": "user", "content": "  \n    Find the top 5 most
expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT Track.Name, UnitPrice
```

```
\nFROM Track \nORDER BY UnitPrice DESC \nLIMIT 5"}, {"role": "user", "content": "  \n    Find all tracks wi
th a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT Name \nFROM
Track \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": "  \n    List all albums and their c
orresponding artist names  \n"}, {"role": "assistant", "content": "SELECT A.Title, ART.Name \nFROM Album AS
A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": "  \n    Find all invoic
es since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT DISTINCT Invoice
Id, Total\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'"}, {"role": "user", "content": "  \n    List all
invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM Invoice WHERE Tot
al > 10"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role":
"assistant", "content": "SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM Customer\nGROUP BY Co
untry\nORDER BY NumberOfCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    List all employees
and their reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT E.FirstName + '
' + E.LastName AS EmployeeName,\n       COALESCE(R.FirstName + ' ' + R.LastName, '--') AS ManagerName\nFROM
Employee E\nLEFT JOIN Employee R ON E.ReportsTo = R.EmployeeId"}, {"role": "user", "content": "How many rec
ords are in table called customer"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customer"}, {"r
ole": "user", "content": "  \n    Get the average invoice total for each customer:\n"}, {"role": "assistan
t", "content": "SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerI
d"}, {"role": "user", "content": "  \n    Get all genres that do not have any tracks associated with the
m:\n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:32:05.889069108Z', 'message': {'role': 'assistan
t', 'content': 'SELECT Name FROM Genre WHERE GenreId NOT IN ( SELECT GenreId FROM Track )'}, 'done_reason':
'stop', 'done': True, 'total_duration': 61222033383, 'load_duration': 14744015, 'prompt_eval_count': 1742,
'prompt_eval_duration': 56947003000, 'eval_count': 17, 'eval_duration': 2505449000}
LLM Response: SELECT Name FROM Genre WHERE GenreId NOT IN ( SELECT GenreId FROM Track )
SELECT Name FROM Genre WHERE GenreId NOT IN ( SELECT GenreId FROM Track )
Empty DataFrame
Columns: [Name]
Index: []
Info: Ollama parameters:
model=llama3.1: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 answers the question the user asked: '  \n    Get all genres that do not have any tracks associated wi
th them:\n'\n\nThe DataFrame was produced using this query: SELECT Name FROM Genre WHERE GenreId NOT IN ( S
ELECT GenreId FROM Track )\n\nThe following is information about the resulting pandas DataFrame 'df': \nRun
ning df.dtypes gives:\n Name    object\ndtype: object"}, {"role": "user", "content": "Can you generate the
Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called
'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not
answer with any explanations -- just the code."}]
```

```
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:32:31.764372979Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.graph_objects as go\n\nif len(df) == 1:\n    fig = go.Figure(data=
[go.Indicator(\n        title='Genres without tracks',\n        value=df['Name'].values[0],\n        number
_dict=dict(decimals=0),\n        mode='number+gauge')])\nelse:\n    fig = go.Figure(data=[go.Bar(x=df['Nam
e'], y=[1]*len(df))])\n\nfig.update_layout(\n    title='Genres without tracks',\n    xaxis_title='Genre Nam
e',\n    yaxis_title=''\n)\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 258
73230224, 'load_duration': 14496954, 'prompt_eval_count': 254, 'prompt_eval_duration': 7736305000, 'eval_co
unt': 122, 'eval_duration': 17992239000}
```


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

Out[28]:
```
('SELECT Name FROM Genre WHERE GenreId NOT IN ( SELECT GenreId FROM Track )',
 Empty DataFrame
 Columns: [Name]
 Index: [],
 Figure({
     'data': [{'type': 'bar', 'x': array([], dtype=object), 'y': []}],
     'layout': {'template': '...',
                'title': {'text': 'Genres without tracks'},
                'xaxis': {'title': {'text': 'Genre Name'}},
                'yaxis': {'title': {'text': ''}}}
 }))
```

In [29]:
```python
question = """
    List all customers who have not placed any orders:
"""

vn.ask(question=question)
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n
FirstName NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Add
ress NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalC
ode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    S
upportRepId INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId)
REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Invoice\n
(\n    InvoiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NUL
L,\n    BillingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    Bi
llingCountry NVARCHAR(40),\n    BillingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CO
NSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerI
d) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEG
ER  NOT NULL,\n    InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,
2)  NOT NULL,\n    Quantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineI
d),\n    FOREIGN KEY (InvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTI
ON,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)
\n\nCREATE TABLE Employee\n(\n    EmployeeId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n
FirstName NVARCHAR(20)  NOT NULL,\n    Title NVARCHAR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIM
E,\n    HireDate DATETIME,\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n
Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Em
ail NVARCHAR(60),\n    CONSTRAINT PK_Employee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFER
ENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_CustomerSu
pportRepId ON Customer (SupportRepId)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n
TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIG
N KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOR
EIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TA
BLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT
NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (Artis
tId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (Cus
tomerId)\n\nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    A
lbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n
Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT
PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO A
CTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO A
CTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Playlist\n(\n    PlaylistId INTEGER  NOT NULL,\n    Name NVARC
HAR(120),\n    CONSTRAINT PK_Playlist PRIMARY KEY  (PlaylistId)\n)\n\n===Response Guidelines \n1. If the pr
ovided context is sufficient, please generate a valid SQL query without any explanations for the question.
\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particul
ar column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend t
he query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explai
```

```
n why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been aske
d and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'con
tent': '  \n    Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'S
ELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I ON C.CustomerId
= I.CustomerId \nGROUP BY C.CustomerId'}, {'role': 'user', 'content': 'what are the top 5 countries that cu
stomers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(CustomerId) AS NumberOfCustom
ers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5'}, {'role': 'user', 'conten
t': 'How many records are in table called customer'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FRO
M customer'}, {'role': 'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'rol
e': 'assistant', 'content': 'SELECT * FROM Invoice WHERE Total > 10'}, {'role': 'user', 'content': '  \n
Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT D
ISTINCT InvoiceId, Total\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'"}, {'role': 'user', 'content': '
\n    Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT Custom
erId, AVG(Total) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerId'}, {'role': 'user', 'content':
'  \n    Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT COUNT
(DISTINCT InvoiceId), BillingCountry\nFROM Invoice'}, {'role': 'user', 'content': "  \n    List all employe
es and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': "SELECT E.FirstName +
' ' + E.LastName AS EmployeeName,\n       COALESCE(R.FirstName + ' ' + R.LastName, '--') AS ManagerName\nFR
OM Employee E\nLEFT JOIN Employee R ON E.ReportsTo = R.EmployeeId"}, {'role': 'user', 'content': '  \n    L
ist all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT A.Titl
e, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'conten
t': '  \n    Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'conten
t': 'SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY UnitPrice DESC \nLIMIT 5'}, {'role': 'user', 'con
tent': '  \n    List all customers who have not placed any orders:\n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    FirstName
NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Address NVARC
HAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCH
AR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    SupportRepI
d INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId) REFERENCE
S Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Invoice\n(\n    In
voiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    Bil
lingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountr
y NVARCHAR(40),\n    BillingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK
_Invoice PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NUL
L,\n    InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NUL
L,\n    Quantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREI
GN KEY (InvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREI
GN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABL
E Employee\n(\n    EmployeeId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    FirstName NVARC
HAR(20)  NOT NULL,\n    Title NVARCHAR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIME,\n    HireDate 
DATETIME,\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR
(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(6
0),\n    CONSTRAINT PK_Employee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Employee
(EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON C
ustomer (SupportRepId)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INT
EGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (Play
listId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (T
rackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Album\n
(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n    
CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\t
ON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n
\nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INT
EGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseco
nds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track
PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON 
UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE
NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON U
PDATE NO ACTION\n)\n\nCREATE TABLE Playlist\n(\n    PlaylistId INTEGER  NOT NULL,\n    Name NVARCHAR(12
0),\n    CONSTRAINT PK_Playlist PRIMARY KEY  (PlaylistId)\n)\n\n===Response Guidelines \n1. If the provided
context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If
the provided context is almost sufficient but requires knowledge of a specific string in a particular colum
n, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query
with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it 
can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and ans
wered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "
\n    Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT C.Cu
stomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.Custom
erId \nGROUP BY C.CustomerId"}, {"role": "user", "content": "what are the top 5 countries that customers co
me from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM 
Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "How ma
ny records are in table called customer"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM custome
r"}, {"role": "user", "content": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "assi
stant", "content": "SELECT * FROM Invoice WHERE Total > 10"}, {"role": "user", "content": "  \n    Find all
invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT DISTINCT I
nvoiceId, Total\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'"}, {"role": "user", "content": "  \n    Ge
t the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT CustomerId, AV
```

```
G(Total) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerId"}, {"role": "user", "content": "  \n
Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT COUNT(DISTINCT
InvoiceId), BillingCountry\nFROM Invoice"}, {"role": "user", "content": "  \n    List all employees and the
ir reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT E.FirstName + ' ' + E.L
astName AS EmployeeName,\n       COALESCE(R.FirstName + ' ' + R.LastName, '--') AS ManagerName\nFROM Employ
ee E\nLEFT JOIN Employee R ON E.ReportsTo = R.EmployeeId"}, {"role": "user", "content": "  \n    List all a
lbums and their corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT A.Title, ART.Nam
e \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": "  \n
Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT T
rack.Name, UnitPrice \nFROM Track \nORDER BY UnitPrice DESC \nLIMIT 5"}, {"role": "user", "content": "  \n
List all customers who have not placed any orders:\n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:33:06.616938595Z', 'message': {'role': 'assistan
t', 'content': 'SELECT CustomerId FROM Customer EXCEPT SELECT CustomerId FROM Invoice'}, 'done_reason': 'st
op', 'done': True, 'total_duration': 34758994127, 'load_duration': 15369501, 'prompt_eval_count': 1026, 'pr
ompt_eval_duration': 32759018000, 'eval_count': 13, 'eval_duration': 1841138000}
LLM Response: SELECT CustomerId FROM Customer EXCEPT SELECT CustomerId FROM Invoice
SELECT CustomerId FROM Customer EXCEPT SELECT CustomerId FROM Invoice
Empty DataFrame
Columns: [CustomerId]
Index: []
Info: Ollama parameters:
model=llama3.1: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 answers the question the user asked: '  \n    List all customers who have not placed any orders:\n'\n
\nThe DataFrame was produced using this query: SELECT CustomerId FROM Customer EXCEPT SELECT CustomerId FRO
M Invoice\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gi
ves:\n CustomerId    object\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plot
ly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If the
re is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with
any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:33:32.036563314Z', 'message': {'role': 'assistan
t', 'content': '```python\nimport plotly.express as px\n\nfig = px.bar(df, x=\'CustomerId\', y=0, title=\'C
ustomers without orders\')\nfig.show()\n\n# or if there is only one value\nif len(df) == 1:\n    fig = px.d
ensity_histogram(df[\'CustomerId\'], histnorm=\'probability density\',\n                                tit
le="Customers without orders",\n                                height=400,\n
width=800)\n    fig.show()\nelse:\n    fig = px.bar(df, x=\'CustomerId\', y=0, title=\'Customers without or
ders\')\n    fig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 25417616601, 'load_d
```

```
uration': 14208600, 'prompt_eval_count': 244, 'prompt_eval_duration': 7448465000, 'eval_count': 121, 'eval_
duration': 17819815000}
```

```
Out[29]: ('SELECT CustomerId FROM Customer EXCEPT SELECT CustomerId FROM Invoice',
          Empty DataFrame
          Columns: [CustomerId]
          Index: [],
          Figure({
              'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
                        'hovertemplate': 'CustomerId=%{label}<extra></extra>',
                        'labels': array([], dtype=object),
                        'legendgroup': '',
                        'name': '',
                        'showlegend': True,
                        'type': 'pie'}],
              'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'template': '...'}
          }))
```

```
In [30]: question = """
             Get the top 10 most popular artists (based on the number of tracks):
         """

         vn.ask(question=question)
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name
NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n
Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,
2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album
(AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (Genre
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (Med
iaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_AlbumArtistId ON Album (Arti
stId)\n\nCREATE INDEX IFK_TrackAlbumId ON Track (AlbumId)\n\nCREATE TABLE Artist\n(\n    ArtistId INTEGER
NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Artist PRIMARY KEY  (ArtistId)\n)\n\nCREATE INDEX IFK
_TrackGenreId ON Track (GenreId)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCH
AR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    F
OREIGN KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREA
TE INDEX IFK_PlaylistTrackTrackId ON PlaylistTrack (TrackId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON Track
(MediaTypeId)\n\nCREATE TABLE Playlist\n(\n    PlaylistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n
CONSTRAINT PK_Playlist PRIMARY KEY  (PlaylistId)\n)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGE
R  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, Tra
ckId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTI
ON\n)\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL que
ry without any explanations for the question. \n2. If the provided context is almost sufficient but require
s 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 p
rovided 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 given before. \n"}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive tracks (based on
unit price):\n'}, {'role': 'assistant', 'content': 'SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY Un
itPrice DESC \nLIMIT 5'}, {'role': 'user', 'content': '  \n    List all genres and the number of tracks in
each genre:\n'}, {'role': 'assistant', 'content': 'SELECT G.Name, COUNT(T.TrackId) AS NumberOfTracks \nFROM
Genre G \nJOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name'}, {'role': 'user', 'content': '  \n    L
ist all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT A.Titl
e, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'conten
t': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Cou
ntry, COUNT(CustomerId) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers D
ESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find all tracks with a name containing "What" (case-in
sensitive)\n'}, {'role': 'assistant', 'content': "SELECT Name \nFROM Track \nWHERE LOWER(Name) LIKE '%wha
t%'"}, {'role': 'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'as
sistant', 'content': 'SELECT * FROM Invoice WHERE Total > 10'}, {'role': 'user', 'content': '  \n    Get th
e average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(To
tal) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerId'}, {'role': 'user', 'content': '  \n    Ge
t the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT C.CustomerI
```

```
d, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId \n
GROUP BY C.CustomerId'}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'ro
le': 'assistant', 'content': 'SELECT COUNT(*) FROM customer'}, {'role': 'user', 'content': '  \n    Find th
e total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT COUNT(DISTINCT Invoic
eId), BillingCountry\nFROM Invoice'}, {'role': 'user', 'content': '  \n    Get the top 10 most popular arti
sts (based on the number of tracks):\n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(2
00)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Compose
r NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT 
NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumI
d) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n
\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaType
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_AlbumArtistId ON Album (ArtistId) 
\n\nCREATE INDEX IFK_TrackAlbumId ON Track (AlbumId)\n\nCREATE TABLE Artist\n(\n    ArtistId INTEGER  NOT N
ULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Artist PRIMARY KEY  (ArtistId)\n)\n\nCREATE INDEX IFK_Trac
kGenreId ON Track (GenreId)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(16
0)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIG
N KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE IN
DEX IFK_PlaylistTrackTrackId ON PlaylistTrack (TrackId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON Track (Medi
aTypeId)\n\nCREATE TABLE Playlist\n(\n    PlaylistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONST
RAINT PK_Playlist PRIMARY KEY  (PlaylistId)\n)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NO
T NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackI
d),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO A
CTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION 
\n)\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 k
nowledge of a specific string in a particular column, please generate an intermediate SQL query to find the
distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provi
ded context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant ta
ble(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was
given before. \n"}, {"role": "user", "content": "  \n    Find the top 5 most expensive tracks (based on uni
t price):\n"}, {"role": "assistant", "content": "SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY UnitP
rice DESC \nLIMIT 5"}, {"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.TrackId) AS NumberOfTracks \nFROM Ge
nre G \nJOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name"}, {"role": "user", "content": "  \n    Lis
```

```
t all albums and their corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT A.Title,
ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content":
"what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Countr
y, COUNT(CustomerId) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC
\nLIMIT 5"}, {"role": "user", "content": "  \n    Find all tracks with a name containing \"What\" (case-ins
ensitive)\n"}, {"role": "assistant", "content": "SELECT Name \nFROM Track \nWHERE LOWER(Name) LIKE '%wha
t%'"}, {"role": "user", "content": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "as
sistant", "content": "SELECT * FROM Invoice WHERE Total > 10"}, {"role": "user", "content": "  \n    Get th
e average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT CustomerId, AVG(To
tal) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerId"}, {"role": "user", "content": "  \n    Ge
t the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT C.CustomerI
d, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId \n
GROUP BY C.CustomerId"}, {"role": "user", "content": "How many records are in table called customer"}, {"ro
le": "assistant", "content": "SELECT COUNT(*) FROM customer"}, {"role": "user", "content": "  \n    Find th
e total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT COUNT(DISTINCT Invoic
eId), BillingCountry\nFROM Invoice"}, {"role": "user", "content": "  \n    Get the top 10 most popular arti
sts (based on the number of tracks):\n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:34:40.94504808Z', 'message': {'role': 'assistan
t', 'content': 'SELECT T.ArtistName, COUNT(T.TrackId) AS NumberOfTracks \nFROM (\n  SELECT A.Name AS Artist
Name, T.Trackid FROM Track AS T \n  JOIN Album AS A ON T.AlbumId = A.AlbumId\n) AS T \nGROUP BY T.ArtistNam
e \nORDER BY NumberOfTracks DESC \nLIMIT 10'}, 'done_reason': 'stop', 'done': True, 'total_duration': 68608
015643, 'load_duration': 20358215, 'prompt_eval_count': 1714, 'prompt_eval_duration': 56031899000, 'eval_co
unt': 70, 'eval_duration': 10832561000}
LLM Response: SELECT T.ArtistName, COUNT(T.TrackId) AS NumberOfTracks
FROM (
  SELECT A.Name AS ArtistName, T.Trackid FROM Track AS T
  JOIN Album AS A ON T.AlbumId = A.AlbumId
) AS T
GROUP BY T.ArtistName
ORDER BY NumberOfTracks DESC
LIMIT 10
SELECT T.ArtistName, COUNT(T.TrackId) AS NumberOfTracks
FROM (
  SELECT A.Name AS ArtistName, T.Trackid FROM Track AS T
  JOIN Album AS A ON T.AlbumId = A.AlbumId
) AS T
GROUP BY T.ArtistName
ORDER BY NumberOfTracks DESC
LIMIT 10
Couldn't run sql:  Execution failed on sql 'SELECT T.ArtistName, COUNT(T.TrackId) AS NumberOfTracks
FROM (
  SELECT A.Name AS ArtistName, T.Trackid FROM Track AS T
  JOIN Album AS A ON T.AlbumId = A.AlbumId
) AS T
GROUP BY T.ArtistName
ORDER BY NumberOfTracks DESC
LIMIT 10': no such column: A.Name
```

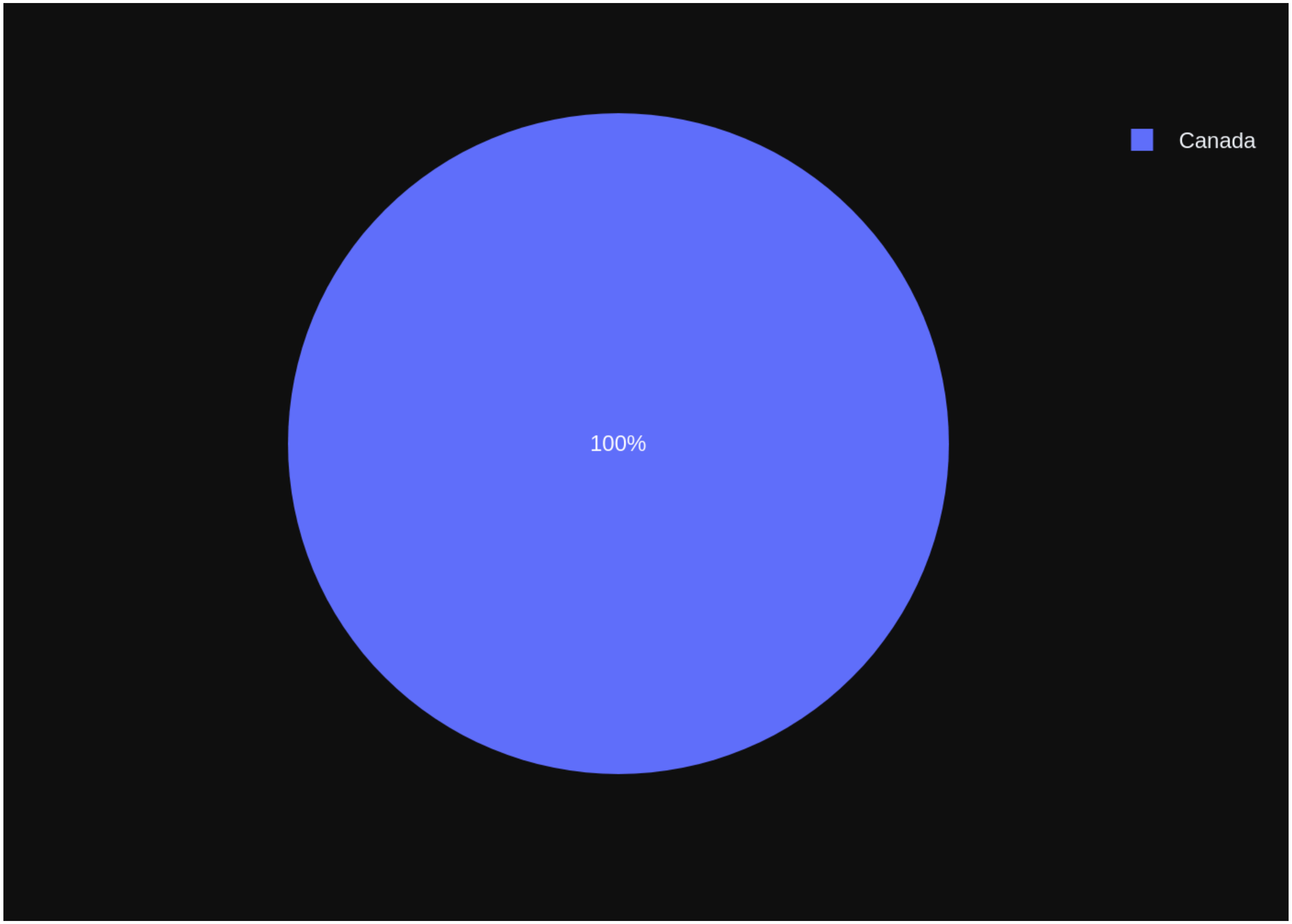
In [31]:

```python
question = """
     List all customers from Canada and their email addresses:
"""

vn.ask(question=question)
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n
FirstName NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Add
ress NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalC
ode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    S
upportRepId INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId)
REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_Custo
merSupportRepId ON Customer (SupportRepId)\n\nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n
CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(70),\n    B
illingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    BillingPost
alCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceI
d),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO A
CTION\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE TABLE Employee\n(\n    Emp
loyeeId INTEGER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    FirstName NVARCHAR(20)  NOT NULL,\n
Title NVARCHAR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIME,\n    HireDate DATETIME,\n    Address N
VARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NV
ARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60),\n    CONSTRAINT PK_Emp
loyee PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Employee (EmployeeId) \n\t\tON DEL
ETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n
InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Q
uantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (I
nvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (T
rackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_Inv
oiceLineTrackId ON InvoiceLine (TrackId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON InvoiceLine (InvoiceI
d)\n\nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\n\nCREATE TABLE PlaylistTrack\n(\n    Play
listId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (Pl
aylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTIO
N ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON U
PDATE NO ACTION\n)\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a
valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficien
t but requires knowledge of a specific string in a particular column, please generate an intermediate SQL q
uery to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql
\n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the
most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer e
xactly as it was given before. \n"}, {'role': 'user', 'content': 'what are the top 5 countries that custome
rs come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\n
FROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': '
\n    Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT C.Cu
stomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.Custom
erId \nGROUP BY C.CustomerId'}, {'role': 'user', 'content': 'How many records are in table called custome
r'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customer'}, {'role': 'user', 'content': '  \n
```

```
Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT COUNT(DISTINCT
InvoiceId), BillingCountry\nFROM Invoice'}, {'role': 'user', 'content': '  \n    List all invoices with a t
otal exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM Invoice WHERE Total > 10'}, {'rol
e': 'user', 'content': '  \n    Get the average invoice total for each customer:\n'}, {'role': 'assistant',
'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerId'},
{'role': 'user', 'content': "  \n    List all employees and their reporting manager's name (if any):\n"},
{'role': 'assistant', 'content': "SELECT E.FirstName + ' ' + E.LastName AS EmployeeName,\n       COALESCE
(R.FirstName + ' ' + R.LastName, '--') AS ManagerName\nFROM Employee E\nLEFT JOIN Employee R ON E.ReportsTo
= R.EmployeeId"}, {'role': 'user', 'content': '  \n    Find all invoices since 2010 and the total amount in
voiced:\n'}, {'role': 'assistant', 'content': "SELECT DISTINCT InvoiceId, Total\nFROM Invoice\nWHERE Invoic
eDate >= '2010-01-01'"}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive tracks (based o
n unit price):\n'}, {'role': 'assistant', 'content': 'SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY
UnitPrice DESC \nLIMIT 5'}, {'role': 'user', 'content': '  \n    List all albums and their corresponding ar
tist names  \n'}, {'role': 'assistant', 'content': 'SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artis
t AS ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'content': '  \n     List all customers from Cana
da and their email addresses:\n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    FirstName
NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Address NVARC
HAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCH
AR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    SupportRepI
d INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId) REFERENCE
S Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_CustomerSuppor
tRepId ON Customer (SupportRepId)\n\nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n    Custome
rId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(70),\n    BillingC
ity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    BillingPostalCode
NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),\n
FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)
\n\nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE TABLE Employee\n(\n    EmployeeId
INTEGER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    FirstName NVARCHAR(20)  NOT NULL,\n    Title
NVARCHAR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIME,\n    HireDate DATETIME,\n    Address NVARCHA
R(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR
(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60),\n    CONSTRAINT PK_Employee
PRIMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    Inv
oiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quan
tity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (Invo
iceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (Trac
kId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_Invoic
eLineTrackId ON InvoiceLine (TrackId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON InvoiceLine (InvoiceId)\n
\nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\n\nCREATE TABLE PlaylistTrack\n(\n    Playlist
Id INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (Playli
stId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON
UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE
NO ACTION\n)\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid
SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but
requires knowledge of a specific string in a particular column, please generate an intermediate SQL query t
o find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. I
f the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most
relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactl
y as it was given before. \n"}, {"role": "user", "content": "what are the top 5 countries that customers co
me from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM
Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "  \n
Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT C.Customer
Id, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId
\nGROUP BY C.CustomerId"}, {"role": "user", "content": "How many records are in table called customer"},
{"role": "assistant", "content": "SELECT COUNT(*) FROM customer"}, {"role": "user", "content": "  \n    Fin
d the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT COUNT(DISTINCT In
voiceId), BillingCountry\nFROM Invoice"}, {"role": "user", "content": "  \n    List all invoices with a tot
al exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM Invoice WHERE Total > 10"}, {"role":
"user", "content": "  \n    Get the average invoice total for each customer:\n"}, {"role": "assistant", "co
ntent": "SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerId"}, {"rol
e": "user", "content": "  \n    List all employees and their reporting manager's name (if any):\n"}, {"rol
e": "assistant", "content": "SELECT E.FirstName + ' ' + E.LastName AS EmployeeName,\n       COALESCE(R.Firs
tName + ' ' + R.LastName, '--') AS ManagerName\nFROM Employee E\nLEFT JOIN Employee R ON E.ReportsTo = R.Em
ployeeId"}, {"role": "user", "content": "  \n    Find all invoices since 2010 and the total amount invoice
d:\n"}, {"role": "assistant", "content": "SELECT DISTINCT InvoiceId, Total\nFROM Invoice\nWHERE InvoiceDate
>= '2010-01-01'"}, {"role": "user", "content": "  \n    Find the top 5 most expensive tracks (based on unit
price):\n"}, {"role": "assistant", "content": "SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY UnitPri
ce DESC \nLIMIT 5"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist na
mes  \n"}, {"role": "assistant", "content": "SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS AR
T ON A.ArtistId = ART.ArtistId"}, {"role": "user", "content": "  \n     List all customers from Canada and
their email addresses:\n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:35:50.396058561Z', 'message': {'role': 'assistan
t', 'content': "SELECT Country, Email FROM Customer WHERE Country = 'Canada'"}, 'done_reason': 'stop', 'don
e': True, 'total_duration': 69389926066, 'load_duration': 21248783, 'prompt_eval_count': 2039, 'prompt_eval
_duration': 66233287000, 'eval_count': 13, 'eval_duration': 2002132000}
```

```
LLM Response: SELECT Country, Email FROM Customer WHERE Country = 'Canada'
SELECT Country, Email FROM Customer WHERE Country = 'Canada'
  Country                   Email
0  Canada     ftremblay@gmail.com
1  Canada      mphilips12@shaw.ca
2  Canada     jenniferp@rogers.ca
3  Canada        robbrown@shaw.ca
4  Canada     edfrancis@yachoo.ca
5  Canada    marthasilk@gmail.com
6  Canada  aaronmitchell@yahoo.ca
7  Canada  ellie.sullivan@shaw.ca
Info: Ollama parameters:
model=llama3.1: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 answers the question the user asked: '  \n     List all customers from Canada and their email addresse
s:\n'\n\nThe DataFrame was produced using this query: SELECT Country, Email FROM Customer WHERE Country =
'Canada'\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes giv
es:\n Country    object\nEmail      object\ndtype: object"}, {"role": "user", "content": "Can you generate
the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe call
ed 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do n
ot answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:36:14.772692241Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Country', y='Email')\nfig.upd
ate_layout(title='Customers from Canada and their email addresses',\n                  xaxis_title='Countr
y',\n                  yaxis_title='Email')\n\nif len(df) == 1:\n    fig = px.density_error_bars(x=df['Cou
ntry'], y=df['Email'],\n                                 title='Customers from Canada and their email addre
sses',\n                                 xaxis_title='Country',\n                                 yaxis_tit
le='Email')\nelse:\n    fig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 242778341
91, 'load_duration': 15210207, 'prompt_eval_count': 248, 'prompt_eval_duration': 7672921000, 'eval_count':
112, 'eval_duration': 16500040000}
```


100%
Canada

```
Out[31]: ("SELECT Country, Email FROM Customer WHERE Country = 'Canada'",
            Country                    Email
          0  Canada     ftremblay@gmail.com
          1  Canada      mphilips12@shaw.ca
          2  Canada     jenniferp@rogers.ca
          3  Canada        robbrown@shaw.ca
          4  Canada     edfrancis@yachoo.ca
          5  Canada    marthasilk@gmail.com
          6  Canada  aaronmitchell@yahoo.ca
          7  Canada  ellie.sullivan@shaw.ca,
          Figure({
              'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
                        'hovertemplate': 'Country=%{label}<extra></extra>',
                        'labels': array(['Canada', 'Canada', 'Canada', 'Canada', 'Canada', 'Canada', 'Canada',
                                         'Canada'], dtype=object),
                        'legendgroup': '',
                        'name': '',
                        'showlegend': True,
                        'type': 'pie'}],
              'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'template': '...'}
          }))
```

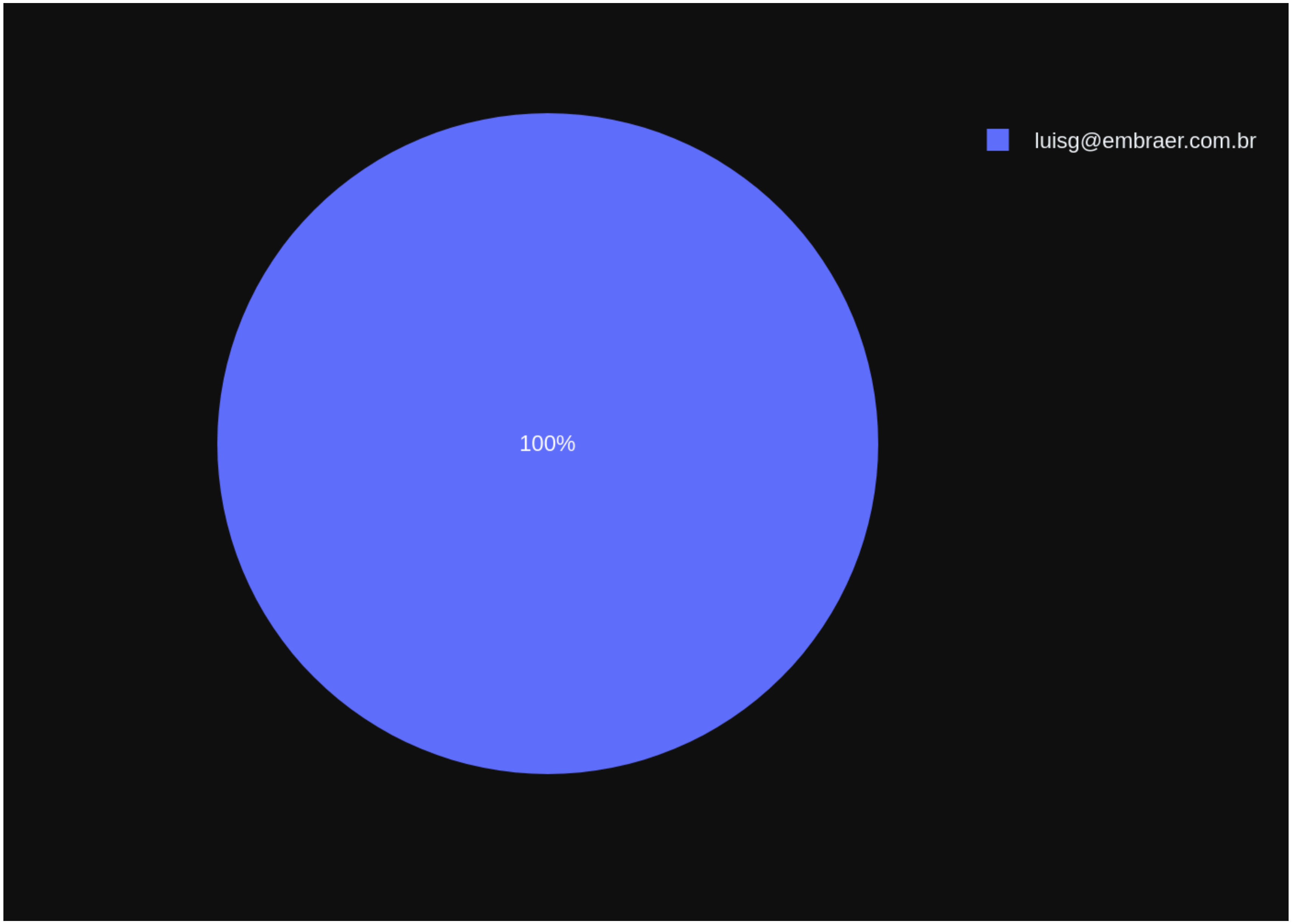
```
In [32]: question = """
              Find the customer with the most invoices
         """

         vn.ask(question=question)
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nC
REATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDat
e DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n    BillingState NV
ARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    BillingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)
NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Cus
tomer (CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceI
d ON InvoiceLine (InvoiceId)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    Invo
iceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quant
ity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (Invoi
ceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (Track
Id) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_Invoice
LineTrackId ON InvoiceLine (TrackId)\n\nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    Fi
rstName NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Addre
ss NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCod
e NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    Sup
portRepId INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId) R
EFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_Custom
erSupportRepId ON Customer (SupportRepId)\n\nCREATE TABLE Employee\n(\n    EmployeeId INTEGER  NOT NULL,\n
LastName NVARCHAR(20)  NOT NULL,\n    FirstName NVARCHAR(20)  NOT NULL,\n    Title NVARCHAR(30),\n    Repor
tsTo INTEGER,\n    BirthDate DATETIME,\n    HireDate DATETIME,\n    Address NVARCHAR(70),\n    City NVARCHA
R(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHA
R(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60),\n    CONSTRAINT PK_Employee PRIMARY KEY  (EmployeeI
d),\n    FOREIGN KEY (ReportsTo) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO AC
TION\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\n\nCREATE TABLE Track\n(\n    TrackId
INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT N
ULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INT
EGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN
KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (G
enreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTy
peId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response Gu
idelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanat
ions for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specif
ic 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 insuf
ficient, 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': 'assi
stant', 'content': 'SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoi
ce I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId'}, {'role': 'user', 'content': '  \n    List al
l invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM Invoice WHERE T
```

```
otal > 10'}, {'role': 'user', 'content': '  \n    Find all invoices since 2010 and the total amount invoice
d:\n'}, {'role': 'assistant', 'content': "SELECT DISTINCT InvoiceId, Total\nFROM Invoice\nWHERE InvoiceDate
>= '2010-01-01'"}, {'role': 'user', 'content': '  \n    Find the total number of invoices per country:\n'},
{'role': 'assistant', 'content': 'SELECT COUNT(DISTINCT InvoiceId), BillingCountry\nFROM Invoice'}, {'rol
e': 'user', 'content': '  \n    Get the average invoice total for each customer:\n'}, {'role': 'assistant',
'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerId'},
{'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistan
t', 'content': 'SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nOR
DER BY NumberOfCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find the top 5 most expensiv
e tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT Track.Name, UnitPrice \nFROM
Track \nORDER BY UnitPrice DESC \nLIMIT 5'}, {'role': 'user', 'content': 'How many records are in table cal
led customer'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customer'}, {'role': 'user', 'conten
t': '  \n     List all customers from Canada and their email addresses:\n'}, {'role': 'assistant', 'conten
t': "SELECT Country, Email FROM Customer WHERE Country = 'Canada'"}, {'role': 'user', 'content': "  \n    L
ist all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': "SELEC
T E.FirstName + ' ' + E.LastName AS EmployeeName,\n       COALESCE(R.FirstName + ' ' + R.LastName, '--') AS
ManagerName\nFROM Employee E\nLEFT JOIN Employee R ON E.ReportsTo = R.EmployeeId"}, {'role': 'user', 'conte
nt': '  \n     Find the customer with the most invoices \n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE TABLE
Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME
NOT NULL,\n    BillingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(4
0),\n    BillingCountry NVARCHAR(40),\n    BillingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NU
LL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer
(CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON I
nvoiceLine (InvoiceId)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    InvoiceId
INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quantity IN
TEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (InvoiceId)
REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) RE
FERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceLineTr
ackId ON InvoiceLine (TrackId)\n\nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT NULL,\n    FirstNam
e NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(80),\n    Address NVA
RCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVAR
CHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)  NOT NULL,\n    SupportRe
pId INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KEY (SupportRepId) REFEREN
CES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_CustomerSupp
```

```
ortRepId ON Customer (SupportRepId)\n\nCREATE TABLE Employee\n(\n    EmployeeId INTEGER  NOT NULL,\n    Las
tName NVARCHAR(20)  NOT NULL,\n    FirstName NVARCHAR(20)  NOT NULL,\n    Title NVARCHAR(30),\n    ReportsT
o INTEGER,\n    BirthDate DATETIME,\n    HireDate DATETIME,\n    Address NVARCHAR(70),\n    City NVARCHAR(4
0),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(2
4),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60),\n    CONSTRAINT PK_Employee PRIMARY KEY  (EmployeeI
d),\n    FOREIGN KEY (ReportsTo) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO AC
TION\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\n\nCREATE TABLE Track\n(\n    TrackId
INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT N
ULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INT
EGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN
KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (G
enreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTy
peId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response Gu
idelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanat
ions for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specif
ic 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 insuf
ficient, 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": "assi
stant", "content": "SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoi
ce I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId"}, {"role": "user", "content": "  \n    List al
l invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM Invoice WHERE T
otal > 10"}, {"role": "user", "content": "  \n    Find all invoices since 2010 and the total amount invoice
d:\n"}, {"role": "assistant", "content": "SELECT DISTINCT InvoiceId, Total\nFROM Invoice\nWHERE InvoiceDate
>= '2010-01-01'"}, {"role": "user", "content": "  \n    Find the total number of invoices per country:\n"},
{"role": "assistant", "content": "SELECT COUNT(DISTINCT InvoiceId), BillingCountry\nFROM Invoice"}, {"rol
e": "user", "content": "  \n    Get the average invoice total for each customer:\n"}, {"role": "assistant",
"content": "SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerId"},
{"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistan
t", "content": "SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nOR
DER BY NumberOfCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    Find the top 5 most expensiv
e tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT Track.Name, UnitPrice \nFROM
Track \nORDER BY UnitPrice DESC \nLIMIT 5"}, {"role": "user", "content": "How many records are in table cal
led customer"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customer"}, {"role": "user", "conten
t": "  \n     List all customers from Canada and their email addresses:\n"}, {"role": "assistant", "conten
t": "SELECT Country, Email FROM Customer WHERE Country = 'Canada'"}, {"role": "user", "content": "  \n    L
ist all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELEC
T E.FirstName + ' ' + E.LastName AS EmployeeName,\n       COALESCE(R.FirstName + ' ' + R.LastName, '--') AS
ManagerName\nFROM Employee E\nLEFT JOIN Employee R ON E.ReportsTo = R.EmployeeId"}, {"role": "user", "conte
nt": "  \n     Find the customer with the most invoices \n"}]
Info: Ollama Response:
```

```
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:37:30.295896527Z', 'message': {'role': 'assistan
t', 'content': 'SELECT C.Email FROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.C
ustomerId, C.FirstName, C.LastName, C.Email\nORDER BY COUNT(I.InvoiceId) DESC\nLIMIT 1'}, 'done_reason': 's
top', 'done': True, 'total_duration': 75406400883, 'load_duration': 17536913, 'prompt_eval_count': 2028, 'p
rompt_eval_duration': 67235433000, 'eval_count': 48, 'eval_duration': 7543040000}
LLM Response: SELECT C.Email FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
GROUP BY C.CustomerId, C.FirstName, C.LastName, C.Email
ORDER BY COUNT(I.InvoiceId) DESC
LIMIT 1
SELECT C.Email FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
GROUP BY C.CustomerId, C.FirstName, C.LastName, C.Email
ORDER BY COUNT(I.InvoiceId) DESC
LIMIT 1
                  Email
0  luisg@embraer.com.br
Info: Ollama parameters:
model=llama3.1: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 answers the question the user asked: '  \n     Find the customer with the most invoices \n'\n\nThe Dat
aFrame was produced using this query: SELECT C.Email FROM Customer C\nJOIN Invoice I ON C.CustomerId = I.Cu
stomerId\nGROUP BY C.CustomerId, C.FirstName, C.LastName, C.Email\nORDER BY COUNT(I.InvoiceId) DESC\nLIMIT
1\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n E
mail    object\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 on
e value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanati
ons -- just the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:37:52.550621712Z', 'message': {'role': 'assistan
t', 'content': '```python\nimport plotly.express as px\n\nif len(df) == 1:\n    fig = px.bar(x=df[\'Email
\'], y=1)\nelse:\n    fig = px.bar(df, x=\'Email\', y=\'value\')\n\nfig.update_layout(\n    title_text="Cus
tomer with the Most Invoices",\n    font_size_title=20,\n    font_size_subtitle=16\n)\n\nfig.show()\n```'},
'done_reason': 'stop', 'done': True, 'total_duration': 22183889992, 'load_duration': 13947703, 'prompt_eval
_count': 308, 'prompt_eval_duration': 9419264000, 'eval_count': 85, 'eval_duration': 12652570000}
```


100%
luisg@embraer.com.br

Out[32]:
```
('SELECT C.Email FROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.CustomerId, C.
FirstName, C.LastName, C.Email\nORDER BY COUNT(I.InvoiceId) DESC\nLIMIT 1',
                    Email
 0  luisg@embraer.com.br,
 Figure({
     'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
               'hovertemplate': 'Email=%{label}<extra></extra>',
               'labels': array(['luisg@embraer.com.br'], dtype=object),
               'legendgroup': '',
               'name': '',
               'showlegend': True,
               'type': 'pie'}],
     'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'template': '...'}
 }))
```

In [ ]:

Advanced SQL questions

In [33]:
```
question = """
     Find the customer who bought the most albums in total quantity (across all invoices):
"""

vn.ask(question=question)
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name
NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n
Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,
2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album
(AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (Genre
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (Med
iaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  N
OT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIM
ARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON U
PDATE NO ACTION\n)\n\nCREATE INDEX IFK_AlbumArtistId ON Album (ArtistId)\n\nCREATE TABLE InvoiceLine\n(\n
InvoiceLineId INTEGER  NOT NULL,\n    InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    Uni
tPrice NUMERIC(10,2)  NOT NULL,\n    Quantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY
(InvoiceLineId),\n    FOREIGN KEY (InvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON U
PDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE
NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE TABLE Invoice\n(\n
InvoiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    B
illingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCoun
try NVARCHAR(40),\n    BillingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT
PK_Invoice PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_TrackAlbumId ON Track (AlbumId)\n\nCREATE IND
EX IFK_InvoiceLineInvoiceId ON InvoiceLine (InvoiceId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON InvoiceLin
e (TrackId)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT N
ULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REF
ERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REF
ERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response Guidelines \n1. If
the provided context is sufficient, please generate a valid SQL query without any explanations for the ques
tion. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a pa
rticular column, please generate an intermediate SQL query to find the distinct strings in that column. Pre
pend 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 bee
n asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'use
r', 'content': '  \n     Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'S
ELECT C.Email FROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.CustomerId, C.Firs
tName, C.LastName, C.Email\nORDER BY COUNT(I.InvoiceId) DESC\nLIMIT 1'}, {'role': 'user', 'content': '  \n
Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT C.Customer
Id, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId
\nGROUP BY C.CustomerId'}, {'role': 'user', 'content': '  \n    Find all invoices since 2010 and the total
amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT DISTINCT InvoiceId, Total\nFROM Invoice\nWHE
RE InvoiceDate >= '2010-01-01'"}, {'role': 'user', 'content': '  \n    List all invoices with a total excee
ding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM Invoice WHERE Total > 10'}, {'role': 'user',
'content': '  \n    Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant',
'content': 'SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY UnitPrice DESC \nLIMIT 5'}, {'role': 'use
r', 'content': '  \n    Find the total number of invoices per country:\n'}, {'role': 'assistant', 'conten
t': 'SELECT COUNT(DISTINCT InvoiceId), BillingCountry\nFROM Invoice'}, {'role': 'user', 'content': '  \n
Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId,
AVG(Total) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerId'}, {'role': 'user', 'content': '  \n
List all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT A.Titl
e, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId'}, {'role': 'user', 'conten
t': '  \n    List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content':
'SELECT G.Name, COUNT(T.TrackId) AS NumberOfTracks \nFROM Genre G \nJOIN Track T ON G.GenreId = T.GenreId
\nGROUP BY G.Name'}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'role':
'assistant', 'content': 'SELECT COUNT(*) FROM customer'}, {'role': 'user', 'content': '  \n     Find the cu
stomer who bought the most albums in total quantity (across all invoices): \n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(2
00)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Compose
r NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT
NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumI
d) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n
\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaType
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  NOT NUL
L,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KE
Y  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE
NO ACTION\n)\n\nCREATE INDEX IFK_AlbumArtistId ON Album (ArtistId)\n\nCREATE TABLE InvoiceLine\n(\n    Invo
iceLineId INTEGER  NOT NULL,\n    InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPri
ce NUMERIC(10,2)  NOT NULL,\n    Quantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (I
nvoiceLineId),\n    FOREIGN KEY (InvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPD
ATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE TABLE Invoice\n(\n    Inv
oiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    Bill
ingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry
NVARCHAR(40),\n    BillingPostalCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_I
nvoice PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DE
LETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_TrackAlbumId ON Track (AlbumId)\n\nCREATE INDEX I
FK_InvoiceLineInvoiceId ON InvoiceLine (InvoiceId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON InvoiceLine (T
rackId)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NUL
L,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFER
ENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFER
ENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response Guidelines \n1. If th
e provided context is sufficient, please generate a valid SQL query without any explanations for the questi
on. \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. Prepe
nd the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please ex
plain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been
asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user",
"content": "  \n     Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELEC
T C.Email FROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.CustomerId, C.FirstNam
e, C.LastName, C.Email\nORDER BY COUNT(I.InvoiceId) DESC\nLIMIT 1"}, {"role": "user", "content": "  \n    G
et the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT C.CustomerI
d, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId \n
GROUP BY C.CustomerId"}, {"role": "user", "content": "  \n    Find all invoices since 2010 and the total am
ount invoiced:\n"}, {"role": "assistant", "content": "SELECT DISTINCT InvoiceId, Total\nFROM Invoice\nWHERE
InvoiceDate >= '2010-01-01'"}, {"role": "user", "content": "  \n    List all invoices with a total exceedin
g $10:\n"}, {"role": "assistant", "content": "SELECT * FROM Invoice WHERE Total > 10"}, {"role": "user", "c
ontent": "  \n    Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "c
ontent": "SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY UnitPrice DESC \nLIMIT 5"}, {"role": "user",
"content": "  \n    Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "S
ELECT COUNT(DISTINCT InvoiceId), BillingCountry\nFROM Invoice"}, {"role": "user", "content": "  \n    Get t
he average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT CustomerId, AVG(T
otal) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerId"}, {"role": "user", "content": "  \n    L
ist all albums and their corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT A.Titl
e, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "conten
t": "  \n    List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content":
"SELECT G.Name, COUNT(T.TrackId) AS NumberOfTracks \nFROM Genre G \nJOIN Track T ON G.GenreId = T.GenreId
\nGROUP BY G.Name"}, {"role": "user", "content": "How many records are in table called customer"}, {"role":
"assistant", "content": "SELECT COUNT(*) FROM customer"}, {"role": "user", "content": "  \n     Find the cu
stomer who bought the most albums in total quantity (across all invoices): \n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:39:13.399568541Z', 'message': {'role': 'assistan
t', 'content': "SELECT C.CustomerId, SUM(IL.Quantity) AS TotalAlbumsPurchased \nFROM Customer C \nJOIN Invo
ice I ON C.CustomerId = I.CustomerId \nJOIN InvoiceLine IL ON I.InvoiceId = IL.InvoiceId \nWHERE IL.TrackId
IN ( SELECT TrackId FROM Track WHERE Genre = 'Album' ) \nGROUP BY C.CustomerId \nORDER BY TotalAlbumsPurcha
sed DESC LIMIT 1"}, 'done_reason': 'stop', 'done': True, 'total_duration': 80714314276, 'load_duration': 14
935354, 'prompt_eval_count': 2008, 'prompt_eval_duration': 65912572000, 'eval_count': 85, 'eval_duration':
13249972000}
LLM Response: SELECT C.CustomerId, SUM(IL.Quantity) AS TotalAlbumsPurchased
FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
JOIN InvoiceLine IL ON I.InvoiceId = IL.InvoiceId
WHERE IL.TrackId IN ( SELECT TrackId FROM Track WHERE Genre = 'Album' )
GROUP BY C.CustomerId
ORDER BY TotalAlbumsPurchased DESC LIMIT 1
SELECT C.CustomerId, SUM(IL.Quantity) AS TotalAlbumsPurchased
FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
JOIN InvoiceLine IL ON I.InvoiceId = IL.InvoiceId
WHERE IL.TrackId IN ( SELECT TrackId FROM Track WHERE Genre = 'Album' )
GROUP BY C.CustomerId
ORDER BY TotalAlbumsPurchased DESC LIMIT 1
Couldn't run sql:  Execution failed on sql 'SELECT C.CustomerId, SUM(IL.Quantity) AS TotalAlbumsPurchased
FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
JOIN InvoiceLine IL ON I.InvoiceId = IL.InvoiceId
WHERE IL.TrackId IN ( SELECT TrackId FROM Track WHERE Genre = 'Album' )
GROUP BY C.CustomerId
ORDER BY TotalAlbumsPurchased DESC LIMIT 1': no such column: Genre
```

In [34]:
```
question = """
     Find the top 5 customer who bought the most albums in total quantity (across all invoices):
"""

vn.ask(question=question)
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name
NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n
Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,
2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album
(AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (Genre
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (Med
iaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  N
OT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIM
ARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON U
PDATE NO ACTION\n)\n\nCREATE INDEX IFK_AlbumArtistId ON Album (ArtistId)\n\nCREATE TABLE InvoiceLine\n(\n
InvoiceLineId INTEGER  NOT NULL,\n    InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    Uni
tPrice NUMERIC(10,2)  NOT NULL,\n    Quantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY
(InvoiceLineId),\n    FOREIGN KEY (InvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON U
PDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE
NO ACTION\n)\n\nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NUL
L,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n
BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    BillingPostalCode NVARCHAR(10),\n    Tota
l NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerI
d) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_In
voiceCustomerId ON Invoice (CustomerId)\n\nCREATE INDEX IFK_TrackAlbumId ON Track (AlbumId)\n\nCREATE INDEX
IFK_InvoiceLineTrackId ON InvoiceLine (TrackId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON InvoiceLine (In
voiceId)\n\nCREATE TABLE Artist\n(\n    ArtistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAIN
T PK_Artist PRIMARY KEY  (ArtistId)\n)\n\n===Response Guidelines \n1. If the provided context is sufficien
t, please generate a valid SQL query without any explanations for the question. \n2. If the provided contex
t is almost sufficient but requires knowledge of a specific string in a particular column, please generate
an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment say
ing intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generate
d. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, ple
ase repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': '  \n     Find the
customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT C.Email FROM Customer C\nJOI
N Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.CustomerId, C.FirstName, C.LastName, C.Email\nORDER
BY COUNT(I.InvoiceId) DESC\nLIMIT 1'}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive t
racks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT Track.Name, UnitPrice \nFROM Tra
ck \nORDER BY UnitPrice DESC \nLIMIT 5'}, {'role': 'user', 'content': '  \n    Get the total number of invo
ices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT C.CustomerId, COUNT(I.InvoiceId) AS To
talInvoices \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId'},
{'role': 'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistan
t', 'content': 'SELECT * FROM Invoice WHERE Total > 10'}, {'role': 'user', 'content': 'what are the top 5 c
ountries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(CustomerId) A
S NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5'}, {'role':
```

```
'user', 'content': '  \n    Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assi
stant', 'content': "SELECT DISTINCT InvoiceId, Total\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'"},
{'role': 'user', 'content': '  \n    Get the average invoice total for each customer:\n'}, {'role': 'assist
ant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerI
d'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per country:\n'}, {'role': 'ass
istant', 'content': 'SELECT COUNT(DISTINCT InvoiceId), BillingCountry\nFROM Invoice'}, {'role': 'user', 'co
ntent': '  \n    List all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'conten
t': 'SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId'}, {'rol
e': 'user', 'content': '  \n    List all genres and the number of tracks in each genre:\n'}, {'role': 'assi
stant', 'content': 'SELECT G.Name, COUNT(T.TrackId) AS NumberOfTracks \nFROM Genre G \nJOIN Track T ON G.Ge
nreId = T.GenreId \nGROUP BY G.Name'}, {'role': 'user', 'content': '  \n     Find the top 5 customer who bo
ught the most albums in total quantity (across all invoices):\n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(2
00)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Compose
r NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT
NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumI
d) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n
\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaType
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  NOT NUL
L,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KE
Y  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE
NO ACTION\n)\n\nCREATE INDEX IFK_AlbumArtistId ON Album (ArtistId)\n\nCREATE TABLE InvoiceLine\n(\n    Invo
iceLineId INTEGER  NOT NULL,\n    InvoiceId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPri
ce NUMERIC(10,2)  NOT NULL,\n    Quantity INTEGER  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (I
nvoiceLineId),\n    FOREIGN KEY (InvoiceId) REFERENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPD
ATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION\n)\n\nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NULL,\n
InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n    Billi
ngState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    BillingPostalCode NVARCHAR(10),\n    Total NUM
ERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerId) RE
FERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_Invoice
CustomerId ON Invoice (CustomerId)\n\nCREATE INDEX IFK_TrackAlbumId ON Track (AlbumId)\n\nCREATE INDEX IFK_
InvoiceLineTrackId ON InvoiceLine (TrackId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON InvoiceLine (Invoic
eId)\n\nCREATE TABLE Artist\n(\n    ArtistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK
_Artist PRIMARY KEY  (ArtistId)\n)\n\n===Response Guidelines \n1. If the provided context is sufficient, pl
```

```
ease generate a valid SQL query without any explanations for the question. \n2. If the provided context is
almost sufficient but requires knowledge of a specific string in a particular column, please generate an in
termediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying i
ntermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n
4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please r
epeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "  \n     Find the custo
mer with the most invoices \n"}, {"role": "assistant", "content": "SELECT C.Email FROM Customer C\nJOIN Inv
oice I ON C.CustomerId = I.CustomerId\nGROUP BY C.CustomerId, C.FirstName, C.LastName, C.Email\nORDER BY CO
UNT(I.InvoiceId) DESC\nLIMIT 1"}, {"role": "user", "content": "  \n    Find the top 5 most expensive tracks
(based on unit price):\n"}, {"role": "assistant", "content": "SELECT Track.Name, UnitPrice \nFROM Track \nO
RDER BY UnitPrice DESC \nLIMIT 5"}, {"role": "user", "content": "  \n    Get the total number of invoices f
or each customer\n"}, {"role": "assistant", "content": "SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInv
oices \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId"}, {"role":
"user", "content": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "conte
nt": "SELECT * FROM Invoice WHERE Total > 10"}, {"role": "user", "content": "what are the top 5 countries t
hat customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(CustomerId) AS NumberOf
Customers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers DESC\nLIMIT 5"}, {"role": "user", "c
ontent": "  \n    Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "c
ontent": "SELECT DISTINCT InvoiceId, Total\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'"}, {"role": "us
er", "content": "  \n    Get the average invoice total for each customer:\n"}, {"role": "assistant", "conte
nt": "SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerId"}, {"role":
"user", "content": "  \n    Find the total number of invoices per country:\n"}, {"role": "assistant", "cont
ent": "SELECT COUNT(DISTINCT InvoiceId), BillingCountry\nFROM Invoice"}, {"role": "user", "content": "  \n
List all albums and their corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT A.Titl
e, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId"}, {"role": "user", "conten
t": "  \n    List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content":
"SELECT G.Name, COUNT(T.TrackId) AS NumberOfTracks \nFROM Genre G \nJOIN Track T ON G.GenreId = T.GenreId
\nGROUP BY G.Name"}, {"role": "user", "content": "  \n     Find the top 5 customer who bought the most albu
ms in total quantity (across all invoices):\n"}]
```

```
Info: Ollama Response:
```

```
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:40:29.571530281Z', 'message': {'role': 'assistan
t', 'content': "SELECT C.CustomerId, SUM(Quantity) AS TotalAlbums \nFROM Customer C \nJOIN InvoiceLine I ON
C.CustomerId = I.CustomerId \nWHERE Track.Name LIKE 'Album%' \nGROUP BY C.CustomerId \nORDER BY TotalAlbums
DESC \nLIMIT 5"}, 'done_reason': 'stop', 'done': True, 'total_duration': 76105778556, 'load_duration': 1870
7925, 'prompt_eval_count': 1997, 'prompt_eval_duration': 65739654000, 'eval_count': 56, 'eval_duration': 86
82377000}
```

```
LLM Response: SELECT C.CustomerId, SUM(Quantity) AS TotalAlbums
FROM Customer C
JOIN InvoiceLine I ON C.CustomerId = I.CustomerId
WHERE Track.Name LIKE 'Album%'
GROUP BY C.CustomerId
ORDER BY TotalAlbums DESC
```

```
LIMIT 5
SELECT C.CustomerId, SUM(Quantity) AS TotalAlbums
FROM Customer C
JOIN InvoiceLine I ON C.CustomerId = I.CustomerId
WHERE Track.Name LIKE 'Album%'
GROUP BY C.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5
Couldn't run sql:  Execution failed on sql 'SELECT C.CustomerId, SUM(Quantity) AS TotalAlbums
FROM Customer C
JOIN InvoiceLine I ON C.CustomerId = I.CustomerId
WHERE Track.Name LIKE 'Album%'
GROUP BY C.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5': no such column: Track.Name
```

In [40]:

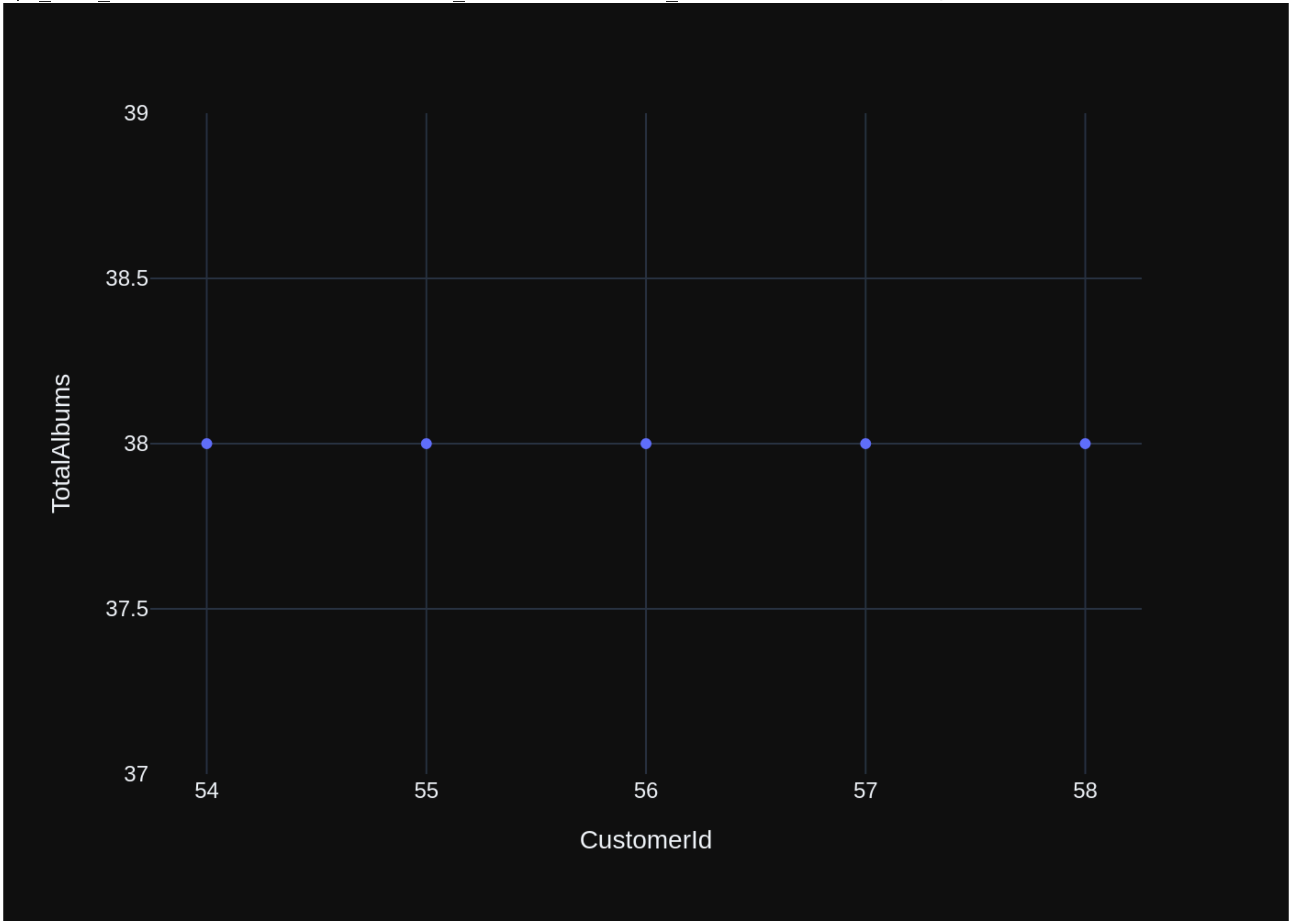
```
question = """
Hint: album quantity is found in invoiceLine,
Find the top 5 customers who bought the most albums in total quantity (across all invoices):
"""

vn.ask(question=question)
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name
NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n
Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,
2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album
(AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (Genre
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (Med
iaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  N
OT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIM
ARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON U
PDATE NO ACTION\n)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    InvoiceId INTE
GER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quantity INTEGE
R  NOT NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (InvoiceId) REFE
RENCES Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERE
NCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_AlbumArtistId ON
Album (ArtistId)\n\nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT
NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(4
0),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    BillingPostalCode NVARCHAR(1
0),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),\n    FOREIGN K
EY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE
INDEX IFK_InvoiceLineInvoiceId ON InvoiceLine (InvoiceId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON Invoice
Line (TrackId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE INDEX IFK_TrackAlbum
Id ON Track (AlbumId)\n\nCREATE TABLE Artist\n(\n    ArtistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n
CONSTRAINT PK_Artist PRIMARY KEY  (ArtistId)\n)\n\n===Response Guidelines \n1. If the provided context is s
ufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provide
d context is almost sufficient but requires knowledge of a specific string in a particular column, please g
enerate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a com
ment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be g
enerated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered befo
re, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': '  \n     Fi
nd the top 3 customers who spent the most money overall:\n'}, {'role': 'assistant', 'content': 'SELECT Cust
omerId, SUM(Total) AS TotalSpent\nFROM Invoice\nGROUP BY CustomerId\nORDER BY TotalSpent DESC\nLIMIT 3'},
{'role': 'user', 'content': '  \n     Find the customer with the most invoices \n'}, {'role': 'assistant',
'content': 'SELECT C.Email FROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.Custo
merId, C.FirstName, C.LastName, C.Email\nORDER BY COUNT(I.InvoiceId) DESC\nLIMIT 1'}, {'role': 'user', 'con
tent': '  \n    Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'con
tent': 'SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY UnitPrice DESC \nLIMIT 5'}, {'role': 'user',
'content': '  \n     Identify artists who have albums with tracks appearing in multiple genres:\n'}, {'rol
e': 'assistant', 'content': 'SELECT A.Name, COUNT(DISTINCT T.GenreId) AS NumberOfGenres\nFROM Artist A\nJOI
N Album AS ALBUM ON A.ArtistId = ALBUM.ArtistId\nJOIN Track T ON ALBUM.AlbumId = T.AlbumId\nGROUP BY A.Name
\nHAVING COUNT(DISTINCT T.GenreId) > 1'}, {'role': 'user', 'content': '  \n    Get the total number of invo
```

```
ices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT C.CustomerId, COUNT(I.InvoiceId) AS To
talInvoices \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId'},
{'role': 'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistan
t', 'content': 'SELECT * FROM Invoice WHERE Total > 10'}, {'role': 'user', 'content': '  \n    Find all inv
oices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT DISTINCT Invo
iceId, Total\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'"}, {'role': 'user', 'content': '  \n    Find
the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT COUNT(DISTINCT Invo
iceId), BillingCountry\nFROM Invoice'}, {'role': 'user', 'content': '  \n    Get the average invoice total
for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTo
tal \nFROM Invoice \nGROUP BY CustomerId'}, {'role': 'user', 'content': '  \n     Get all playlists contain
ing at least 10 tracks and the total duration of those tracks:\n'}, {'role': 'assistant', 'content': 'SELEC
T P.Name, SUM(T.Milliseconds / 1000) AS TotalDuration\nFROM Playlist P \nJOIN PlaylistTrack PT ON P.Playlis
tId = PT.PlaylistId \nJOIN Track T ON PT.TrackId = T.TrackId \nGROUP BY P.Name \nHAVING COUNT(DISTINCT T.Tr
ackId) >= 10'}, {'role': 'user', 'content': '  \nHint: album quantity is found in invoiceLine,           \n
Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(2
00)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Compose
r NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT
NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumI
d) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n
\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaType
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  NOT NUL
L,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KE
Y  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE
NO ACTION\n)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    InvoiceId INTEGER  N
OT NULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quantity INTEGER  NOT
NULL,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (InvoiceId) REFERENCES
Invoice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Tr
ack (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_AlbumArtistId ON Album
(ArtistId)\n\nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NUL
L,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(70),\n    BillingCity NVARCHAR(40),\n
BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    BillingPostalCode NVARCHAR(10),\n    Tota
l NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),\n    FOREIGN KEY (CustomerI
d) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_In
voiceLineInvoiceId ON InvoiceLine (InvoiceId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON InvoiceLine (TrackI
d)\n\nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE INDEX IFK_TrackAlbumId ON Track
(AlbumId)\n\nCREATE TABLE Artist\n(\n    ArtistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAI
NT PK_Artist PRIMARY KEY  (ArtistId)\n)\n\n===Response Guidelines \n1. If the provided context is sufficien
t, please generate a valid SQL query without any explanations for the question. \n2. If the provided contex
t is almost sufficient but requires knowledge of a specific string in a particular column, please generate
an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment say
ing intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generate
d. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, ple
ase repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "  \n     Find the
top 3 customers who spent the most money overall:\n"}, {"role": "assistant", "content": "SELECT CustomerId,
SUM(Total) AS TotalSpent\nFROM Invoice\nGROUP BY CustomerId\nORDER BY TotalSpent DESC\nLIMIT 3"}, {"role":
"user", "content": "  \n     Find the customer with the most invoices \n"}, {"role": "assistant", "conten
t": "SELECT C.Email FROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.CustomerId,
C.FirstName, C.LastName, C.Email\nORDER BY COUNT(I.InvoiceId) DESC\nLIMIT 1"}, {"role": "user", "content":
"  \n    Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content":
"SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY UnitPrice DESC \nLIMIT 5"}, {"role": "user", "conten
t": "  \n     Identify artists who have albums with tracks appearing in multiple genres:\n"}, {"role": "ass
istant", "content": "SELECT A.Name, COUNT(DISTINCT T.GenreId) AS NumberOfGenres\nFROM Artist A\nJOIN Album
AS ALBUM ON A.ArtistId = ALBUM.ArtistId\nJOIN Track T ON ALBUM.AlbumId = T.AlbumId\nGROUP BY A.Name\nHAVING
COUNT(DISTINCT T.GenreId) > 1"}, {"role": "user", "content": "  \n    Get the total number of invoices for
each customer\n"}, {"role": "assistant", "content": "SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoic
es \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId"}, {"role": "u
ser", "content": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "conten
t": "SELECT * FROM Invoice WHERE Total > 10"}, {"role": "user", "content": "  \n    Find all invoices since
2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT DISTINCT InvoiceId, Total
\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'"}, {"role": "user", "content": "  \n    Find the total nu
mber of invoices per country:\n"}, {"role": "assistant", "content": "SELECT COUNT(DISTINCT InvoiceId), Bill
ingCountry\nFROM Invoice"}, {"role": "user", "content": "  \n    Get the average invoice total for each cus
tomer:\n"}, {"role": "assistant", "content": "SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM I
nvoice \nGROUP BY CustomerId"}, {"role": "user", "content": "  \n     Get all playlists containing at least
10 tracks and the total duration of those tracks:\n"}, {"role": "assistant", "content": "SELECT P.Name, SUM
(T.Milliseconds / 1000) AS TotalDuration\nFROM Playlist P \nJOIN PlaylistTrack PT ON P.PlaylistId = PT.Play
listId \nJOIN Track T ON PT.TrackId = T.TrackId \nGROUP BY P.Name \nHAVING COUNT(DISTINCT T.TrackId) >= 1
0"}, {"role": "user", "content": "  \nHint: album quantity is found in invoiceLine,           \nFind the to
p 5 customers who bought the most albums in total quantity (across all invoices):\n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T11:34:34.442335943Z', 'message': {'role': 'assistan
t', 'content': 'SELECT C.CustomerId, SUM(Quantity) AS TotalAlbums \nFROM Customer C \nJOIN Invoice I ON C.C
ustomerId = I.CustomerId \nJOIN InvoiceLine IL ON I.InvoiceId = IL.InvoiceId \nJOIN Track T ON IL.TrackId =
T.Trackid \nWHERE T.AlbumId IS NOT NULL \nGROUP BY C.CustomerId \nORDER BY TotalAlbums DESC LIMIT 5'}, 'don
e_reason': 'stop', 'done': True, 'total_duration': 86013813214, 'load_duration': 2545603207, 'prompt_eval_c
ount': 2043, 'prompt_eval_duration': 69044877000, 'eval_count': 82, 'eval_duration': 12820829000}
```

```
LLM Response: SELECT C.CustomerId, SUM(Quantity) AS TotalAlbums
FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
JOIN InvoiceLine IL ON I.InvoiceId = IL.InvoiceId
JOIN Track T ON IL.TrackId = T.Trackid
WHERE T.AlbumId IS NOT NULL
GROUP BY C.CustomerId
ORDER BY TotalAlbums DESC LIMIT 5
SELECT C.CustomerId, SUM(Quantity) AS TotalAlbums
FROM Customer C
JOIN Invoice I ON C.CustomerId = I.CustomerId
JOIN InvoiceLine IL ON I.InvoiceId = IL.InvoiceId
JOIN Track T ON IL.TrackId = T.Trackid
WHERE T.AlbumId IS NOT NULL
GROUP BY C.CustomerId
ORDER BY TotalAlbums DESC LIMIT 5
   CustomerId  TotalAlbums
0          58           38
1          57           38
2          56           38
3          55           38
4          54           38
Info: Ollama parameters:
model=llama3.1: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 answers the question the user asked: '  \nHint: album quantity is found in invoiceLine,          \nFi
nd the top 5 customers who bought the most albums in total quantity (across all invoices):\n'\n\nThe DataFr
ame was produced using this query: SELECT C.CustomerId, SUM(Quantity) AS TotalAlbums \nFROM Customer C \nJO
IN Invoice I ON C.CustomerId = I.CustomerId \nJOIN InvoiceLine IL ON I.InvoiceId = IL.InvoiceId \nJOIN Trac
k T ON IL.TrackId = T.Trackid \nWHERE T.AlbumId IS NOT NULL \nGROUP BY C.CustomerId \nORDER BY TotalAlbums 
DESC LIMIT 5\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes
gives:\n CustomerId     int64\nTotalAlbums    int64\ndtype: object"}, {"role": "user", "content": "Can you 
generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataf
rame called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python c
ode. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T11:35:02.06488707Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nif len(df) == 1:\n    fig = px.bar(x=df['Customer
Id'], y=df['TotalAlbums'])\nelse:\n    fig = px.bar(df, x='CustomerID', y='TotalAlbums')\n    \nfig.update_
```

layout(title_text='Top Customers by Total Album Quantity',\n   xaxis_title_text='Customer ID',\n   yaxis_title_text='Total Album Quantity')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 27512508852, 'load_duration': 82361558, 'prompt_eval_count': 438, 'prompt_eval_duration': 13498909000, 'eval_count': 94, 'eval_duration': 13781616000}



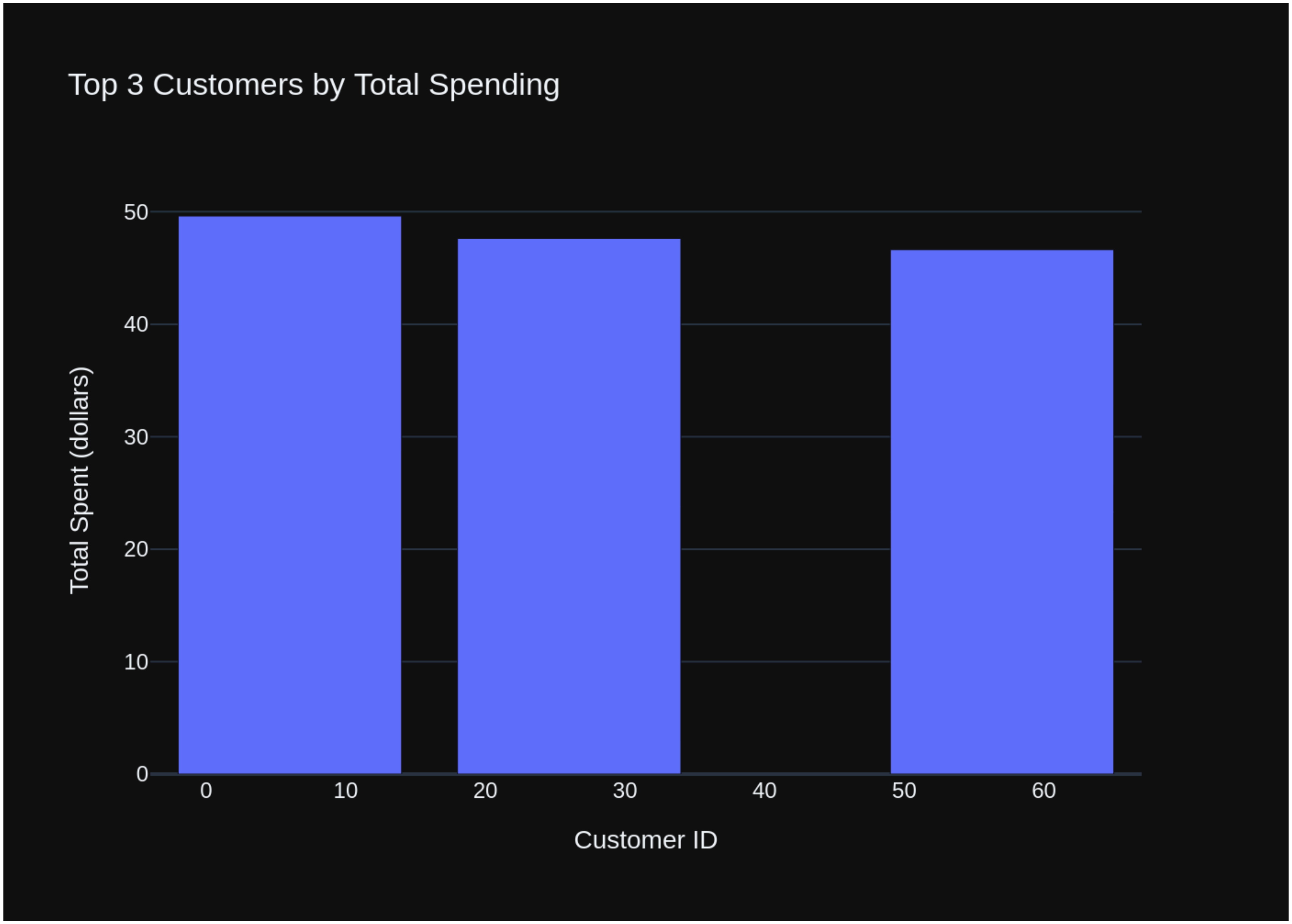
Out[40]:
```
('SELECT C.CustomerId, SUM(Quantity) AS TotalAlbums \nFROM Customer C \nJOIN Invoice I ON C.CustomerId =
I.CustomerId \nJOIN InvoiceLine IL ON I.InvoiceId = IL.InvoiceId \nJOIN Track T ON IL.TrackId = T.Trackid
\nWHERE T.AlbumId IS NOT NULL \nGROUP BY C.CustomerId \nORDER BY TotalAlbums DESC LIMIT 5',
    CustomerId  TotalAlbums
 0          58           38
 1          57           38
 2          56           38
 3          55           38
 4          54           38,
 Figure({
     'data': [{'hovertemplate': 'CustomerId=%{x}<br>TotalAlbums=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'symbol': 'circle'},
               'mode': 'markers',
               'name': '',
               'orientation': 'v',
               'showlegend': False,
               'type': 'scatter',
               'x': array([58, 57, 56, 55, 54]),
               'xaxis': 'x',
               'y': array([38, 38, 38, 38, 38]),
               'yaxis': 'y'}],
     'layout': {'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalAlbums'}}}
 }))
```

In [35]:
```python
question = """
     Find the top 3 customers who spent the most money overall:
"""

vn.ask(question=question)
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n    
CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(70),\n    B
illingCity NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    BillingPost
alCode NVARCHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceI
d),\n    FOREIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO A
CTION\n)\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    InvoiceId INTEGER  NOT N
ULL,\n    TrackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quantity INTEGER  NOT NUL
L,\n    CONSTRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (InvoiceId) REFERENCES Inv
oice (InvoiceId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track
(TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Customer\n(\n    CustomerId INTE
GER  NOT NULL,\n    FirstName NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NV
ARCHAR(80),\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCH
AR(40),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(6
0)  NOT NULL,\n    SupportRepId INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIG
N KEY (SupportRepId) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nC
REATE INDEX IFK_CustomerSupportRepId ON Customer (SupportRepId)\n\nCREATE TABLE Track\n(\n    TrackId INTEG
ER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NUL
L,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEG
ER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN K
EY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (Ge
nreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTyp
eId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IF
K_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\n
\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CON
STRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playl
ist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track
(TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Employee\n(\n    EmployeeId INTE
GER  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    FirstName NVARCHAR(20)  NOT NULL,\n    Title NVAR
CHAR(30),\n    ReportsTo INTEGER,\n    BirthDate DATETIME,\n    HireDate DATETIME,\n    Address NVARCHAR(7
0),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(1
0),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60),\n    CONSTRAINT PK_Employee PR
IMARY KEY  (EmployeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO A
CTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON InvoiceLine (TrackId)\n\n===Response
Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explan
ations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a spec
ific string in a particular column, please generate an intermediate SQL query to find the distinct strings
in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is in
sufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If t
he question has been asked and answered before, please repeat the answer exactly as it was given before.
\n"}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive tracks (based on unit price):\n'},
{'role': 'assistant', 'content': 'SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY UnitPrice DESC \nLIM
IT 5'}, {'role': 'user', 'content': '  \n     Find the customer with the most invoices \n'}, {'role': 'assi
stant', 'content': 'SELECT C.Email FROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY
C.CustomerId, C.FirstName, C.LastName, C.Email\nORDER BY COUNT(I.InvoiceId) DESC\nLIMIT 1'}, {'role': 'use
r', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content':
'SELECT Country, COUNT(CustomerId) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOf
Customers DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Get the average invoice total for each cust
omer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM In
voice \nGROUP BY CustomerId'}, {'role': 'user', 'content': '  \n    Get the total number of invoices for ea
ch customer\n'}, {'role': 'assistant', 'content': 'SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices
\nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId'}, {'role': 'use
r', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content':
'SELECT * FROM Invoice WHERE Total > 10'}, {'role': 'user', 'content': '  \n    Find all invoices since 201
0 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT DISTINCT InvoiceId, Total\nF
ROM Invoice\nWHERE InvoiceDate >= '2010-01-01'"}, {'role': 'user', 'content': '  \n    Find the total numbe
r of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT COUNT(DISTINCT InvoiceId), Billing
Country\nFROM Invoice'}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'ro
le': 'assistant', 'content': 'SELECT COUNT(*) FROM customer'}, {'role': 'user', 'content': '  \n     List a
ll customers from Canada and their email addresses:\n'}, {'role': 'assistant', 'content': "SELECT Country,
Email FROM Customer WHERE Country = 'Canada'"}, {'role': 'user', 'content': '  \n     Find the top 3 custom
ers who spent the most money overall:\n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE TABLE Invoice\n(\n    InvoiceId INTEGER  NOT NULL,\n    CustomerId
INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT NULL,\n    BillingAddress NVARCHAR(70),\n    BillingCity
NVARCHAR(40),\n    BillingState NVARCHAR(40),\n    BillingCountry NVARCHAR(40),\n    BillingPostalCode NVAR
CHAR(10),\n    Total NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Invoice PRIMARY KEY  (InvoiceId),\n    FOR
EIGN KEY (CustomerId) REFERENCES Customer (CustomerId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n
CREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    InvoiceId INTEGER  NOT NULL,\n    Tr
ackId INTEGER  NOT NULL,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    Quantity INTEGER  NOT NULL,\n    CONS
TRAINT PK_InvoiceLine PRIMARY KEY  (InvoiceLineId),\n    FOREIGN KEY (InvoiceId) REFERENCES Invoice (Invoic
eId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId)
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Customer\n(\n    CustomerId INTEGER  NOT N
ULL,\n    FirstName NVARCHAR(40)  NOT NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    Company NVARCHAR(8
0),\n    Address NVARCHAR(70),\n    City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(4
0),\n    PostalCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60)
NOT NULL,\n    SupportRepId INTEGER,\n    CONSTRAINT PK_Customer PRIMARY KEY  (CustomerId),\n    FOREIGN KE
Y (SupportRepId) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREAT
E INDEX IFK_CustomerSupportRepId ON Customer (SupportRepId)\n\nCREATE TABLE Track\n(\n    TrackId INTEGER
NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n
GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n
UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (Album
Id) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) RE
FERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFE
RENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_Invoice
CustomerId ON Invoice (CustomerId)\n\nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\n\nCREATE
TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT P
K_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (Play
listId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackI
d) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Employee\n(\n    EmployeeId INTEGER  NO
T NULL,\n    LastName NVARCHAR(20)  NOT NULL,\n    FirstName NVARCHAR(20)  NOT NULL,\n    Title NVARCHAR(3
0),\n    ReportsTo INTEGER,\n    BirthDate DATETIME,\n    HireDate DATETIME,\n    Address NVARCHAR(70),\n
City NVARCHAR(40),\n    State NVARCHAR(40),\n    Country NVARCHAR(40),\n    PostalCode NVARCHAR(10),\n    P
hone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60),\n    CONSTRAINT PK_Employee PRIMARY KEY
(EmployeeId),\n    FOREIGN KEY (ReportsTo) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UP
DATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON InvoiceLine (TrackId)\n\n===Response Guidelines
\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for
the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string
in a particular column, please generate an intermediate SQL query to find the distinct strings in that colu
mn. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient,
please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question
has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role":
"user", "content": "  \n    Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assi
stant", "content": "SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY UnitPrice DESC \nLIMIT 5"}, {"rol
e": "user", "content": "  \n     Find the customer with the most invoices \n"}, {"role": "assistant", "cont
ent": "SELECT C.Email FROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId\nGROUP BY C.CustomerI
d, C.FirstName, C.LastName, C.Email\nORDER BY COUNT(I.InvoiceId) DESC\nLIMIT 1"}, {"role": "user", "conten
t": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Cou
ntry, COUNT(CustomerId) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers D
ESC\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 Invoice \nGRO
UP BY CustomerId"}, {"role": "user", "content": "  \n    Get the total number of invoices for each customer
\n"}, {"role": "assistant", "content": "SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvoices \nFROM Cus
tomer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId"}, {"role": "user", "conten
t": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT *
FROM Invoice WHERE Total > 10"}, {"role": "user", "content": "  \n    Find all invoices since 2010 and the
total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT DISTINCT InvoiceId, Total\nFROM Invoic
e\nWHERE InvoiceDate >= '2010-01-01'"}, {"role": "user", "content": "  \n    Find the total number of invoi
ces per country:\n"}, {"role": "assistant", "content": "SELECT COUNT(DISTINCT InvoiceId), BillingCountry\nF
```

```
ROM Invoice"}, {"role": "user", "content": "How many records are in table called customer"}, {"role": "assi
stant", "content": "SELECT COUNT(*) FROM customer"}, {"role": "user", "content": "  \n     List all custome
rs from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SELECT Country, Email FROM
Customer WHERE Country = 'Canada'"}, {"role": "user", "content": "  \n     Find the top 3 customers who spe
nt the most money overall:\n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:41:42.920143083Z', 'message': {'role': 'assistan
t', 'content': '```sql\nSELECT CustomerId, SUM(Total) AS TotalSpent\nFROM Invoice\nGROUP BY CustomerId\nORD
ER BY TotalSpent DESC\nLIMIT 3;\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 73292162427,
'load_duration': 19009410, 'prompt_eval_count': 2044, 'prompt_eval_duration': 67481191000, 'eval_count': 3
7, 'eval_duration': 5735200000}
LLM Response: ```sql
SELECT CustomerId, SUM(Total) AS TotalSpent
FROM Invoice
GROUP BY CustomerId
ORDER BY TotalSpent DESC
LIMIT 3;
```
Info: Output from LLM: ```sql
SELECT CustomerId, SUM(Total) AS TotalSpent
FROM Invoice
GROUP BY CustomerId
ORDER BY TotalSpent DESC
LIMIT 3;
```
Extracted SQL: SELECT CustomerId, SUM(Total) AS TotalSpent
FROM Invoice
GROUP BY CustomerId
ORDER BY TotalSpent DESC
LIMIT 3
SELECT CustomerId, SUM(Total) AS TotalSpent
FROM Invoice
GROUP BY 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=llama3.1: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 answers the question the user asked: '  \n     Find the top 3 customers who spent the most money overa
ll:\n'\n\nThe DataFrame was produced using this query: SELECT CustomerId, SUM(Total) AS TotalSpent\nFROM In
voice\nGROUP BY CustomerId\nORDER BY TotalSpent DESC\nLIMIT 3\n\nThe following is information about the res
ulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId      int64\nTotalSpent    float64\ndt
ype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of
the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the data
frame, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the cod
e."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:42:16.647283641Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.graph_objects as go\n\nif len(df) == 1:\n    fig = go.Figure(data=
[go.Indicator(\n        mode = 'number+gauge',\n        value = df['TotalSpent'].values[0],\n        number
= {'suffix': '<b> dollars<b>'},\n        gauge = {'axistitle' : 'Total Spent', \n                 'barcolo
r'  : '#69A2B6'} \n    )])\nelse:\n    fig = go.Figure(data=[go.Bar(x=df['CustomerId'], y=df['TotalSpen
t'])])\n\nfig.update_layout(\n    title_text='Top 3 Customers by Total Spending',\n    xaxis_title='Custome
r ID',\n    yaxis_title='Total Spent (dollars)'\n)\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': Tru
e, 'total_duration': 33619358718, 'load_duration': 16575898, 'prompt_eval_count': 304, 'prompt_eval_duratio
n': 9260923000, 'eval_count': 165, 'eval_duration': 24285957000}
```


Top 3 Customers by Total Spending
Total Spent (dollars)
0
10
20
30
40
50
Customer ID
0
10
20
30
40
50
60

Out[35]:
```
('SELECT CustomerId, SUM(Total) AS TotalSpent\nFROM Invoice\nGROUP BY CustomerId\nORDER BY TotalSpent DESC
\nLIMIT 3',
    CustomerId  TotalSpent
 0           6       49.62
 1          26       47.62
 2          57       46.62,
 Figure({
     'data': [{'type': 'bar', 'x': array([ 6, 26, 57]), 'y': array([49.62, 47.62, 46.62])}],
     'layout': {'template': '...',
                'title': {'text': 'Top 3 Customers by Total Spending'},
                'xaxis': {'title': {'text': 'Customer ID'}},
                'yaxis': {'title': {'text': 'Total Spent (dollars)'}}}
 }))
```

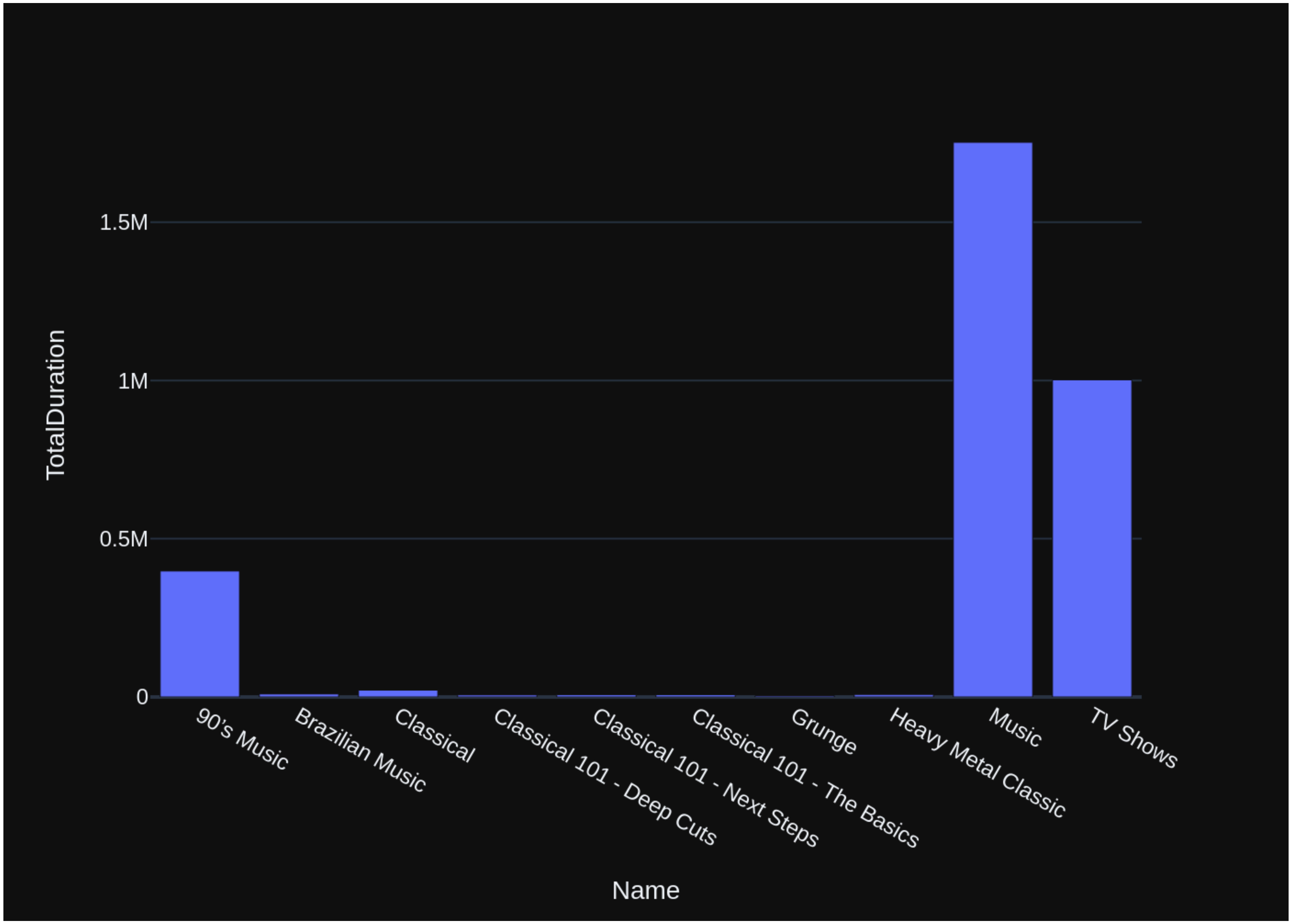
In [36]:
```python
question = """
     Get all playlists containing at least 10 tracks and the total duration of those tracks:
"""

vn.ask(question=question)
```

```
SQL Prompt: [{'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 guidel
ines and format instructions. \n===Tables \nCREATE INDEX IFK_PlaylistTrackTrackId ON PlaylistTrack (TrackI
d)\n\nCREATE TABLE Playlist\n(\n    PlaylistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT
PK_Playlist PRIMARY KEY  (PlaylistId)\n)\n\nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name
NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n
Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,
2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album
(AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (Genre
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (Med
iaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId
INTEGER  NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistI
d, TrackId),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UP
DATE NO ACTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION\n)\n\nCREATE INDEX IFK_TrackGenreId ON Track (GenreId)\n\nCREATE INDEX IFK_TrackAlbumId ON Track
(AlbumId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON Track (MediaTypeId)\n\nCREATE INDEX IFK_AlbumArtistId ON
Album (ArtistId)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NUL
L,\n    ArtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (Arti
stId) REFERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_Inv
oiceLineTrackId ON InvoiceLine (TrackId)\n\n===Response Guidelines \n1. If the provided context is sufficie
nt, please generate a valid SQL query without any explanations for the question. \n2. If the provided conte
xt is almost sufficient but requires knowledge of a specific string in a particular column, please generate
an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment say
ing intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generate
d. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, ple
ase repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': '  \n    List all g
enres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT G.Name, COUNT
(T.TrackId) AS NumberOfTracks \nFROM Genre G \nJOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name'},
{'role': 'user', 'content': '  \n    Find all tracks with a name containing "What" (case-insensitive)\n'},
{'role': 'assistant', 'content': "SELECT Name \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'u
ser', 'content': '  \n    Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assist
ant', 'content': 'SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY UnitPrice DESC \nLIMIT 5'}, {'role':
'user', 'content': '  \n    List all albums and their corresponding artist names  \n'}, {'role': 'assistan
t', 'content': 'SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistI
d'}, {'role': 'user', 'content': '  \n    Find all invoices since 2010 and the total amount invoiced:\n'},
{'role': 'assistant', 'content': "SELECT DISTINCT InvoiceId, Total\nFROM Invoice\nWHERE InvoiceDate >= '201
0-01-01'"}, {'role': 'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'rol
e': 'assistant', 'content': 'SELECT * FROM Invoice WHERE Total > 10'}, {'role': 'user', 'content': '  \n
Find the top 3 customers who spent the most money overall:\n'}, {'role': 'assistant', 'content': 'SELECT Cu
stomerId, SUM(Total) AS TotalSpent\nFROM Invoice\nGROUP BY CustomerId\nORDER BY TotalSpent DESC\nLIMIT 3'},
{'role': 'user', 'content': '  \n    Get the average invoice total for each customer:\n'}, {'role': 'assist
ant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerI
```

```
d'}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assistant', 'c
ontent': 'SELECT COUNT(*) FROM customer'}, {'role': 'user', 'content': '  \n    Get the total number of inv
oices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT C.CustomerId, COUNT(I.InvoiceId) AS T
otalInvoices \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId'},
{'role': 'user', 'content': '  \n     Get all playlists containing at least 10 tracks and the total duratio
n of those tracks:\n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_PlaylistTrackTrackId ON PlaylistTrack (TrackId)\n\nCREATE
TABLE Playlist\n(\n    PlaylistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Playlist P
RIMARY KEY  (PlaylistId)\n)\n\nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(20
0)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Composer
NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT N
ULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId)
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n\t\t
ON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaTypeId)
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER
NOT NULL,\n    TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackI
d),\n    FOREIGN KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO A
CTION,\n    FOREIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\n)\n\nCREATE INDEX IFK_TrackGenreId ON Track (GenreId)\n\nCREATE INDEX IFK_TrackAlbumId ON Track (AlbumId)
\n\nCREATE INDEX IFK_TrackMediaTypeId ON Track (MediaTypeId)\n\nCREATE INDEX IFK_AlbumArtistId ON Album (Ar
tistId)\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    A
rtistId INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REF
ERENCES Artist (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceLineT
rackId ON InvoiceLine (TrackId)\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 alm
ost sufficient but requires knowledge of a specific string in a particular column, please generate an inter
mediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying inte
rmediate_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 repea
t the answer exactly as it was given before. \n"}, {"role": "user", "content": "  \n    List all genres and
the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT G.Name, COUNT(T.TrackId)
AS NumberOfTracks \nFROM Genre G \nJOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name"}, {"role": "use
r", "content": "  \n    Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "a
ssistant", "content": "SELECT Name \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "cont
ent": "  \n    Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "cont
```

```
ent": "SELECT Track.Name, UnitPrice \nFROM Track \nORDER BY UnitPrice DESC \nLIMIT 5"}, {"role": "user", "c
ontent": "  \n    List all albums and their corresponding artist names  \n"}, {"role": "assistant", "conten
t": "SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId = ART.ArtistId"}, {"rol
e": "user", "content": "  \n    Find all invoices since 2010 and the total amount invoiced:\n"}, {"role":
"assistant", "content": "SELECT DISTINCT InvoiceId, Total\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-0
1'"}, {"role": "user", "content": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "ass
istant", "content": "SELECT * FROM Invoice WHERE Total > 10"}, {"role": "user", "content": "  \n     Find t
he top 3 customers who spent the most money overall:\n"}, {"role": "assistant", "content": "SELECT Customer
Id, SUM(Total) AS TotalSpent\nFROM Invoice\nGROUP BY CustomerId\nORDER BY TotalSpent DESC\nLIMIT 3"}, {"rol
e": "user", "content": "  \n    Get the average invoice total for each customer:\n"}, {"role": "assistant",
"content": "SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM Invoice \nGROUP BY CustomerId"},
{"role": "user", "content": "How many records are in table called customer"}, {"role": "assistant", "conten
t": "SELECT COUNT(*) FROM customer"}, {"role": "user", "content": "  \n    Get the total number of invoices
for each customer\n"}, {"role": "assistant", "content": "SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalIn
voices \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId"}, {"rol
e": "user", "content": "  \n     Get all playlists containing at least 10 tracks and the total duration of
those tracks:\n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:43:23.619931224Z', 'message': {'role': 'assistan
t', 'content': 'SELECT P.Name, SUM(T.Milliseconds / 1000) AS TotalDuration\nFROM Playlist P \nJOIN Playlist
Track PT ON P.PlaylistId = PT.PlaylistId \nJOIN Track T ON PT.TrackId = T.TrackId \nGROUP BY P.Name \nHAVIN
G COUNT(DISTINCT T.TrackId) >= 10'}, 'done_reason': 'stop', 'done': True, 'total_duration': 66881196285, 'l
oad_duration': 14905909, 'prompt_eval_count': 1700, 'prompt_eval_duration': 55050643000, 'eval_count': 67,
'eval_duration': 10199685000}
LLM Response: SELECT P.Name, SUM(T.Milliseconds / 1000) AS TotalDuration
FROM Playlist P
JOIN PlaylistTrack PT ON P.PlaylistId = PT.PlaylistId
JOIN Track T ON PT.TrackId = T.TrackId
GROUP BY P.Name
HAVING COUNT(DISTINCT T.TrackId) >= 10
SELECT P.Name, SUM(T.Milliseconds / 1000) AS TotalDuration
FROM Playlist P
JOIN PlaylistTrack PT ON P.PlaylistId = PT.PlaylistId
JOIN Track T ON PT.TrackId = T.TrackId
GROUP BY P.Name
HAVING COUNT(DISTINCT T.TrackId) >= 10
                         Name  TotalDuration
0                  90’s Music         397970
1             Brazilian Music           9464
2                   Classical          21736
3    Classical 101 - Deep Cuts           6742
4   Classical 101 - Next Steps           7565
```

```
5  Classical 101 - The Basics           7429
6                      Grunge           4114
7         Heavy Metal Classic           8189
8                       Music        1752098
9                    TV Shows        1001974
Info: Ollama parameters:
model=llama3.1: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 answers the question the user asked: '  \n     Get all playlists containing at least 10 tracks and the
total duration of those tracks:\n'\n\nThe DataFrame was produced using this query: SELECT P.Name, SUM(T.Mil
liseconds / 1000) AS TotalDuration\nFROM Playlist P \nJOIN PlaylistTrack PT ON P.PlaylistId = PT.PlaylistId
\nJOIN Track T ON PT.TrackId = T.TrackId \nGROUP BY P.Name \nHAVING COUNT(DISTINCT T.TrackId) >= 10\n\nThe
following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name
object\nTotalDuration     int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python
plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If
there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer wi
th any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:43:39.712770325Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Name', y='TotalDuration')\nfi
g.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 15958713633, 'load_duration': 16608
996, 'prompt_eval_count': 376, 'prompt_eval_duration': 11467153000, 'eval_count': 31, 'eval_duration': 4414
851000}
```


TotalDuration
1.5M
1M
0.5M
0
90's Music
Brazilian Music
Classical
Classical 101 - Deep Cuts
Classical 101 - Next Steps
Classical 101 - The Basics
Grunge
Heavy Metal Classic
Music
TV Shows
Name

```
Out[36]: ('SELECT P.Name, SUM(T.Milliseconds / 1000) AS TotalDuration\nFROM Playlist P \nJOIN PlaylistTrack PT ON
         P.PlaylistId = PT.PlaylistId \nJOIN Track T ON PT.TrackId = T.TrackId \nGROUP BY P.Name \nHAVING COUNT(DIS
         TINCT T.TrackId) >= 10',
                                Name  TotalDuration
          0                  90’s Music         397970
          1             Brazilian Music           9464
          2                   Classical          21736
          3   Classical 101 - Deep Cuts           6742
          4  Classical 101 - Next Steps           7565
          5  Classical 101 - The Basics           7429
          6                      Grunge           4114
          7         Heavy Metal Classic           8189
          8                       Music        1752098
          9                    TV Shows        1001974,
          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',
                                    'TV Shows'], dtype=object),
                        'xaxis': 'x',
                        'y': array([ 397970,    9464,   21736,    6742,    7565,    7429,    4114,    8189,
                                    1752098, 1001974]),
                        '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': 'TotalDuration'}}}
          }))
```

In [37]:
```
question = """
     Identify artists who have albums with tracks appearing in multiple genres:
"""

vn.ask(question=question)
```

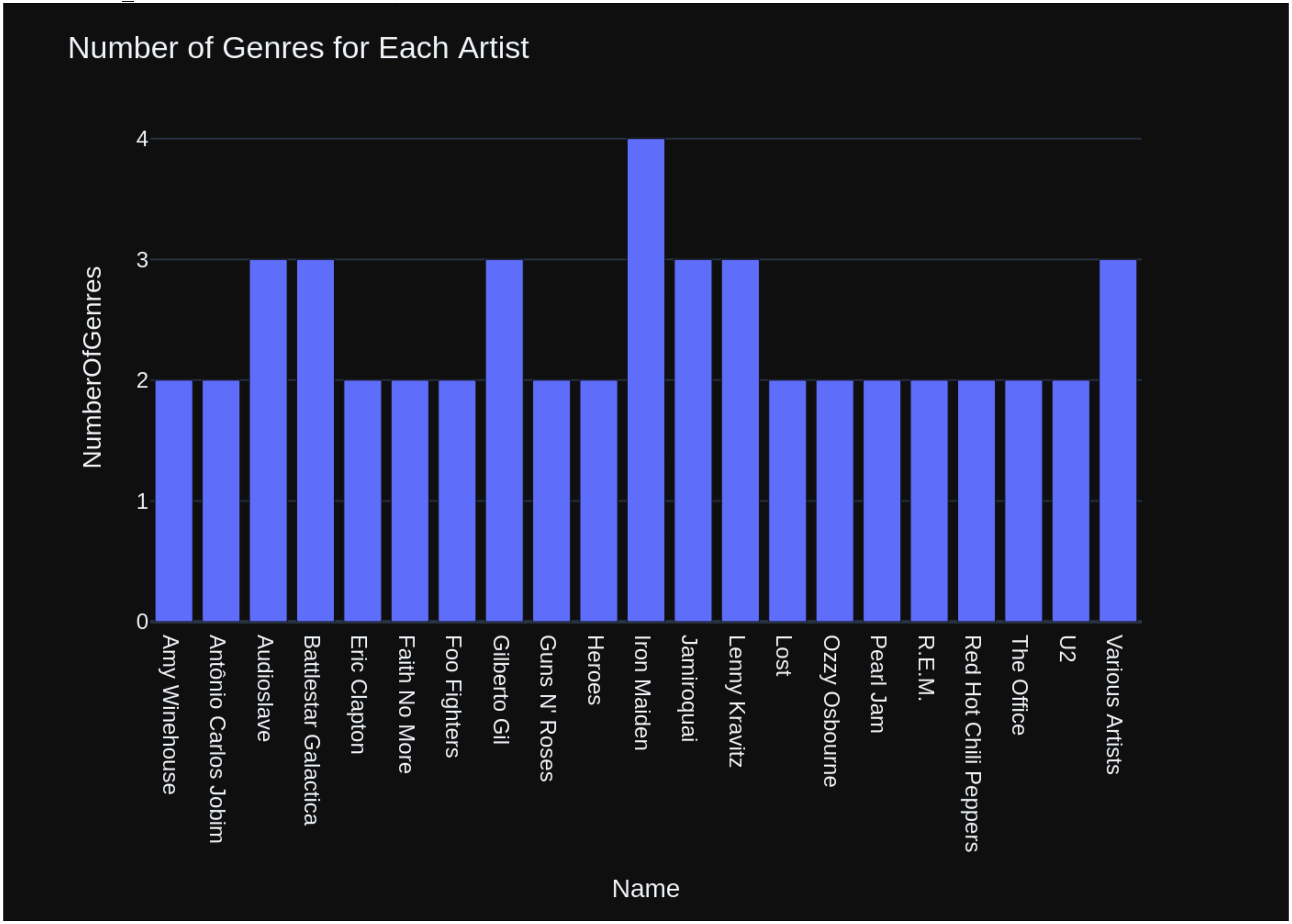
```
SQL Prompt: [{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to
answer the question. Your response should ONLY be based on the given context and follow the response guidel
ines and format instructions. \n===Tables \nCREATE INDEX IFK_AlbumArtistId ON Album (ArtistId)\n\nCREATE TA
BLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n
MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER
NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY
(TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\n    FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n
FOREIGN KEY (MediaTypeId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\n)\n\nCREATE INDEX IFK_TrackGenreId ON Track (GenreId)\n\nCREATE INDEX IFK_TrackAlbumId ON Track (AlbumId)
\n\nCREATE TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId
INTEGER  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES
Artist (ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_TrackMediaTypeId ON
Track (MediaTypeId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON PlaylistTrack (TrackId)\n\nCREATE TABLE Art
ist\n(\n    ArtistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Artist PRIMARY KEY  (Ar
tistId)\n)\n\nCREATE TABLE Genre\n(\n    GenreId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAIN
T PK_Genre PRIMARY KEY  (GenreId)\n)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n
TrackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIG
N KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOR
EIGN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Respon
se Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any exp
lanations 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 strin
gs in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is
insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If
the question has been asked and answered before, please repeat the answer exactly as it was given before.
\n"}, {'role': 'user', 'content': '  \n    List all albums and their corresponding artist names  \n'}, {'ro
le': 'assistant', 'content': 'SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId
= ART.ArtistId'}, {'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.TrackId) AS NumberOfTracks \nFROM Genre G
\nJOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name'}, {'role': 'user', 'content': '  \n     Get all
playlists containing at least 10 tracks and the total duration of those tracks:\n'}, {'role': 'assistant',
'content': 'SELECT P.Name, SUM(T.Milliseconds / 1000) AS TotalDuration\nFROM Playlist P \nJOIN PlaylistTrac
k PT ON P.PlaylistId = PT.PlaylistId \nJOIN Track T ON PT.TrackId = T.TrackId \nGROUP BY P.Name \nHAVING CO
UNT(DISTINCT T.TrackId) >= 10'}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive tracks
(based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT Track.Name, UnitPrice \nFROM Track \nO
RDER BY UnitPrice DESC \nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find all tracks with a name contai
ning "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT Name \nFROM Track \nWHERE LOWE
R(Name) LIKE '%what%'"}, {'role': 'user', 'content': '  \n    List all invoices with a total exceeding $1
0:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM Invoice WHERE Total > 10'}, {'role': 'user', 'conte
nt': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Co
untry, COUNT(CustomerId) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers
```

```
DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n     Find the customer with the most invoices \n'}, {'rol
e': 'assistant', 'content': 'SELECT C.Email FROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId
\nGROUP BY C.CustomerId, C.FirstName, C.LastName, C.Email\nORDER BY COUNT(I.InvoiceId) DESC\nLIMIT 1'}, {'r
ole': 'user', 'content': '  \n     Find the top 3 customers who spent the most money overall:\n'}, {'role':
'assistant', 'content': 'SELECT CustomerId, SUM(Total) AS TotalSpent\nFROM Invoice\nGROUP BY CustomerId\nOR
DER BY TotalSpent DESC\nLIMIT 3'}, {'role': 'user', 'content': '  \n    Get the total number of invoices fo
r each customer\n'}, {'role': 'assistant', 'content': 'SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvo
ices \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId'}, {'role':
'user', 'content': '  \n     Identify artists who have albums with tracks appearing in multiple genre
s:\n'}]
Info: Ollama parameters:
model=llama3.1: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 fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_AlbumArtistId ON Album (ArtistId)\n\nCREATE TABLE Track\n
(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId
INTEGER  NOT NULL,\n    GenreId INTEGER,\n    Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NUL
L,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT NULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (Track
Id),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n
FOREIGN KEY (GenreId) REFERENCES Genre (GenreId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIG
N KEY (MediaTypeId) REFERENCES MediaType (MediaTypeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n
CREATE INDEX IFK_TrackGenreId ON Track (GenreId)\n\nCREATE INDEX IFK_TrackAlbumId ON Track (AlbumId)\n\nCRE
ATE TABLE Album\n(\n    AlbumId INTEGER  NOT NULL,\n    Title NVARCHAR(160)  NOT NULL,\n    ArtistId INTEGE
R  NOT NULL,\n    CONSTRAINT PK_Album PRIMARY KEY  (AlbumId),\n    FOREIGN KEY (ArtistId) REFERENCES Artist
(ArtistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_TrackMediaTypeId ON Track
(MediaTypeId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON PlaylistTrack (TrackId)\n\nCREATE TABLE Artist\n
(\n    ArtistId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Artist PRIMARY KEY  (ArtistI
d)\n)\n\nCREATE TABLE Genre\n(\n    GenreId INTEGER  NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_
Genre PRIMARY KEY  (GenreId)\n)\n\nCREATE TABLE PlaylistTrack\n(\n    PlaylistId INTEGER  NOT NULL,\n    Tr
ackId INTEGER  NOT NULL,\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\n    FOREIGN
KEY (PlaylistId) REFERENCES Playlist (PlaylistId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREI
GN KEY (TrackId) REFERENCES Track (TrackId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n===Response
Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explan
ations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a spec
ific string in a particular column, please generate an intermediate SQL query to find the distinct strings
in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is in
sufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If t
he question has been asked and answered before, please repeat the answer exactly as it was given before.
\n"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist names  \n"}, {"ro
```

```
le": "assistant", "content": "SELECT A.Title, ART.Name \nFROM Album AS A \nJOIN Artist AS ART ON A.ArtistId
= ART.ArtistId"}, {"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.TrackId) AS NumberOfTracks \nFROM Genre G
\nJOIN Track T ON G.GenreId = T.GenreId \nGROUP BY G.Name"}, {"role": "user", "content": "  \n     Get all
playlists containing at least 10 tracks and the total duration of those tracks:\n"}, {"role": "assistant",
"content": "SELECT P.Name, SUM(T.Milliseconds / 1000) AS TotalDuration\nFROM Playlist P \nJOIN PlaylistTrac
k PT ON P.PlaylistId = PT.PlaylistId \nJOIN Track T ON PT.TrackId = T.TrackId \nGROUP BY P.Name \nHAVING CO
UNT(DISTINCT T.TrackId) >= 10"}, {"role": "user", "content": "  \n    Find the top 5 most expensive tracks
(based on unit price):\n"}, {"role": "assistant", "content": "SELECT Track.Name, UnitPrice \nFROM Track \nO
RDER BY UnitPrice DESC \nLIMIT 5"}, {"role": "user", "content": "  \n    Find all tracks with a name contai
ning \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT Name \nFROM Track \nWHERE LO
WER(Name) LIKE '%what%'"}, {"role": "user", "content": "  \n    List all invoices with a total exceeding $1
0:\n"}, {"role": "assistant", "content": "SELECT * FROM Invoice WHERE Total > 10"}, {"role": "user", "conte
nt": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Co
untry, COUNT(CustomerId) AS NumberOfCustomers\nFROM Customer\nGROUP BY Country\nORDER BY NumberOfCustomers
DESC\nLIMIT 5"}, {"role": "user", "content": "  \n     Find the customer with the most invoices \n"}, {"rol
e": "assistant", "content": "SELECT C.Email FROM Customer C\nJOIN Invoice I ON C.CustomerId = I.CustomerId
\nGROUP BY C.CustomerId, C.FirstName, C.LastName, C.Email\nORDER BY COUNT(I.InvoiceId) DESC\nLIMIT 1"}, {"r
ole": "user", "content": "  \n     Find the top 3 customers who spent the most money overall:\n"}, {"role":
"assistant", "content": "SELECT CustomerId, SUM(Total) AS TotalSpent\nFROM Invoice\nGROUP BY CustomerId\nOR
DER BY TotalSpent DESC\nLIMIT 3"}, {"role": "user", "content": "  \n    Get the total number of invoices fo
r each customer\n"}, {"role": "assistant", "content": "SELECT C.CustomerId, COUNT(I.InvoiceId) AS TotalInvo
ices \nFROM Customer C \nJOIN Invoice I ON C.CustomerId = I.CustomerId \nGROUP BY C.CustomerId"}, {"role":
"user", "content": "  \n     Identify artists who have albums with tracks appearing in multiple genre
s:\n"}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:44:52.501370299Z', 'message': {'role': 'assistan
t', 'content': 'SELECT A.Name, COUNT(DISTINCT T.GenreId) AS NumberOfGenres\nFROM Artist A\nJOIN Album AS AL
BUM ON A.ArtistId = ALBUM.ArtistId\nJOIN Track T ON ALBUM.AlbumId = T.AlbumId\nGROUP BY A.Name\nHAVING COUN
T(DISTINCT T.GenreId) > 1'}, 'done_reason': 'stop', 'done': True, 'total_duration': 72609890081, 'load_dura
tion': 15483975, 'prompt_eval_count': 1832, 'prompt_eval_duration': 59765315000, 'eval_count': 73, 'eval_du
ration': 11129785000}
LLM Response: SELECT A.Name, COUNT(DISTINCT T.GenreId) AS NumberOfGenres
FROM Artist A
JOIN Album AS ALBUM ON A.ArtistId = ALBUM.ArtistId
JOIN Track T ON ALBUM.AlbumId = T.AlbumId
GROUP BY A.Name
HAVING COUNT(DISTINCT T.GenreId) > 1
SELECT A.Name, COUNT(DISTINCT T.GenreId) AS NumberOfGenres
FROM Artist A
JOIN Album AS ALBUM ON A.ArtistId = ALBUM.ArtistId
JOIN Track T ON ALBUM.AlbumId = T.AlbumId
```

```
GROUP BY A.Name
HAVING COUNT(DISTINCT T.GenreId) > 1
                     Name  NumberOfGenres
0           Amy Winehouse               2
1     Antônio Carlos Jobim              2
2              Audioslave               3
3    Battlestar Galactica               3
4            Eric Clapton               2
5           Faith No More               2
6            Foo Fighters               2
7            Gilberto Gil               3
8           Guns N' Roses               2
9                  Heroes               2
10            Iron Maiden               4
11             Jamiroquai               3
12          Lenny Kravitz               3
13                   Lost               2
14          Ozzy Osbourne               2
15              Pearl Jam               2
16                 R.E.M.               2
17  Red Hot Chili Peppers               2
18             The Office               2
19                     U2               2
20        Various Artists               3
Info: Ollama parameters:
model=llama3.1: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 answers the question the user asked: '  \n     Identify artists who have albums with tracks appearing
in multiple genres:\n'\n\nThe DataFrame was produced using this query: SELECT A.Name, COUNT(DISTINCT T.Genr
eId) AS NumberOfGenres\nFROM Artist A\nJOIN Album AS ALBUM ON A.ArtistId = ALBUM.ArtistId\nJOIN Track T ON
ALBUM.AlbumId = T.AlbumId\nGROUP BY A.Name\nHAVING COUNT(DISTINCT T.GenreId) > 1\n\nThe following is inform
ation about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name              object\nNum
berOfGenres     int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly cod
e to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is
only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any ex
planations -- just the code."}]
Info: Ollama Response:
{'model': 'llama3.1:latest', 'created_at': '2024-07-24T05:45:17.205040957Z', 'message': {'role': 'assistan
t', 'content': '```python\nimport plotly.express as px\n\nfig = px.bar(df, x=\'Name\', y=\'NumberOfGenres
```

```
\')\nif df[\'NumberOfGenres\'].nunique() == 1:\n    fig.update_layout(title_text=f\'Number of Genres for Al
l Artists: {df["NumberOfGenres"].iloc[0]}\')\n\nelse:\n    fig.update_layout(title_text=\'Number of Genres
for Each Artist\')\n\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 2458716716
5, 'load_duration': 80240551, 'prompt_eval_count': 378, 'prompt_eval_duration': 11552357000, 'eval_count':
88, 'eval_duration': 12892878000}
```



```
Out[37]: ('SELECT A.Name, COUNT(DISTINCT T.GenreId) AS NumberOfGenres\nFROM Artist A\nJOIN Album AS ALBUM ON A.Arti
         stId = ALBUM.ArtistId\nJOIN Track T ON ALBUM.AlbumId = T.AlbumId\nGROUP BY A.Name\nHAVING COUNT(DISTINCT
         T.GenreId) > 1',
                              Name  NumberOfGenres
          0          Amy Winehouse               2
          1    Antônio Carlos Jobim               2
          2             Audioslave               3
          3    Battlestar Galactica               3
          4           Eric Clapton               2
          5          Faith No More               2
          6           Foo Fighters               2
          7           Gilberto Gil               3
          8          Guns N' Roses               2
          9                 Heroes               2
          10           Iron Maiden               4
          11            Jamiroquai               3
          12          Lenny Kravitz               3
          13                  Lost               2
          14          Ozzy Osbourne               2
          15             Pearl Jam               2
          16                R.E.M.               2
          17  Red Hot Chili Peppers               2
          18            The Office               2
          19                    U2               2
          20       Various Artists               3,
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'Name=%{x}<br>NumberOfGenres=%{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([2, 2, 3, 3, 2, 2, 2, 3, 2, 2, 4, 3, 3, 2, 2, 2, 2, 2, 2, 2, 3]),
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                         'legend': {'tracegroupgap': 0},
                         'margin': {'t': 60},
                         'template': '...',
                         'title': {'text': 'Number of Genres for Each Artist'},
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'NumberOfGenres'}}}
          }))
```

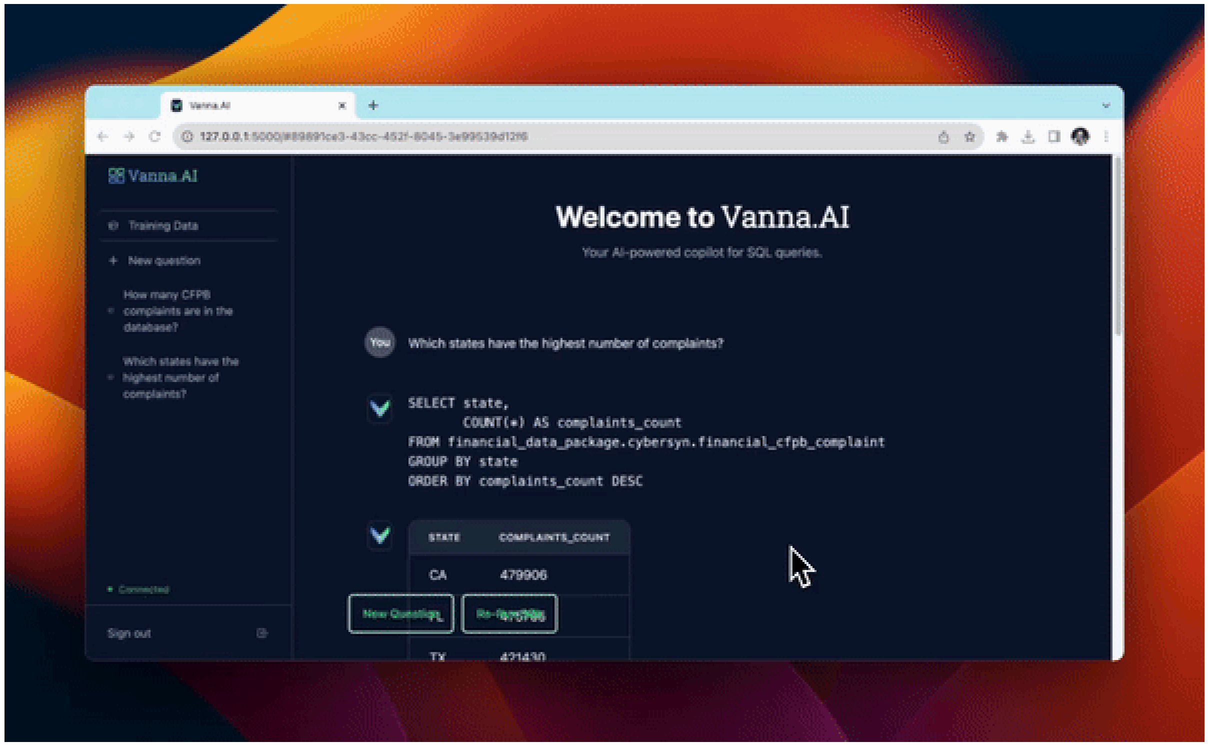
Check completion time

In [38]:
```
ts_stop = time()

elapsed_time = ts_stop - ts_start
print(f"elapsed_time : {elapsed_time} sec")
```

```
elapsed_time : 1915.2621357440948 sec
```

In [ ]:

Launch the User Interface


Vanna.AI
127.0.0.1:5000/#89891ce3-43cc-452f-8045-3e99539d12f6
Vanna.AI
Training Data
New question
How many CFPB complaints are in the database?
Which states have the highest number of complaints?
Connected
Sign out
New Question
Re-Run SQL

Welcome to Vanna.AI

Your AI-powered copilot for SQL queries.

You 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