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'  # '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 False:
    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
0	5698caa9-f558-5bc3-ab17-53fdf8d8e27a-sql	what are the top 5 countries that customers co...	SELECT Country, COUNT(*) AS TotalCustomers\nFR...	sql
1	ad7073e4-6bfb-593d-802a-8b0d388fafa4-sql	How many records are in table called customer	SELECT COUNT(*) FROM Customer	sql
2	c9165031-db41-5792-8000-bd3db552c193-sql	Show me a list of tables in the SQLite database	SELECT name FROM sqlite_master WHERE type='table'	sql
3	e676785b-a05d-541a-bcf4-52eda4106c03-sql	What person has a name of "John Doe"?	SELECT * FROM t_person WHERE name = 'John Doe'	sql
0	00e3136b-f688-5ae4-a2bb-3354635261fc-ddl	None	CREATE INDEX IFK_CustomerSupportRepId ON Custo...	ddl
1	04a3d9eb-fab7-5641-96a1-ae7924ae05e5-ddl	None	CREATE INDEX IFK_EmployeeReportsTo ON Employee...	ddl
2	11439f95-bfbc-530c-8b7f-91f9e45d2877-ddl	None	CREATE TABLE Track\n(\n TrackId INTEGER NO...	ddl
3	29f57d6d-552a-5cc2-8c8c-cb78918d1646-ddl	None	CREATE TABLE Genre\n(\n GenreId INTEGER NO...	ddl
4	33d9efb0-969c-59b8-95f1-04b3e0ddd2df-ddl	None	CREATE INDEX IFK_TrackAlbumId ON Track (AlbumId)	ddl
5	37bc3b25-2dc1-5160-bcae-baf85b93f023-ddl	None	CREATE TABLE InvoiceLine\n(\n InvoiceLineId...	ddl
6	4893f9e4-2275-53cb-89da-43de3063acf1-ddl	None	CREATE TABLE PlaylistTrack\n(\n PlaylistId ...	ddl
7	82dfbbaf-4fe6-5213-8c5a-2e765f459ca1-ddl	None	CREATE TABLE Artist\n(\n ArtistId INTEGER ...	ddl
8	8a3cbf5f-a4d9-5429-93de-e49ef72c5bd5-ddl	None	CREATE TABLE Customer\n(\n CustomerId INTEG...	ddl
9	93344862-5b4e-5a02-8c70-982b08c6badf-ddl	None	CREATE INDEX IFK_InvoiceLineInvoiceId ON Invoi...	ddl
10	96df68d6-cba8-5982-9b6e-4dd4e797683f-ddl	None	CREATE INDEX IFK_AlbumArtistId ON Album (Artis...	ddl
11	97594156-2183-5f7b-82f6-a6ae2d3e248c-ddl	None	CREATE INDEX IFK_InvoiceLineTrackId ON Invoice...	ddl
12	9f818206-3266-527b-8906-	None	CREATE TABLE Employee\n(\n EmployeeId	ddl

	id	question	content	training_data_type
	e9644c2414f4-ddl		INTEG...	
13	a4e8be30-d5dd-5a8a-aa99-64c74783e17d-ddl	None	\n CREATE TABLE IF NOT EXISTS t_person (\n ...	ddl
14	b586614d-f6de-5d5e-aa55-692d9fbbc325-ddl	None	CREATE TABLE Playlist\n(\n PlaylistId INTEG...	ddl
15	b5cb3609-158b-57bb-9124-1425c39eb7d6-ddl	None	CREATE INDEX IFK_TrackGenreId ON Track (GenreId)	ddl
16	bf1f06a1-8065-512c-bfe7-591d0e017d3e-ddl	None	CREATE TABLE Invoice\n(\n InvoiceId INTEGER...	ddl
17	c88034a5-9a78-5b9b-b864-d9b542ffff77-ddl	None	CREATE TABLE MediaType\n(\n MediaTypeId INT...	ddl
18	ca1e33d8-8948-52fa-8c3d-0b5f6b0cccdc-ddl	None	CREATE INDEX IFK_TrackMediaTypeId ON Track (Me...	ddl
19	cfe3064c-0442-5555-be9c-bbc8fa9542a8-ddl	None	CREATE INDEX IFK_InvoiceCustomerId ON Invoice ...	ddl
20	e9124f31-fca1-5503-b473-9b995624b92e-ddl	None	CREATE TABLE Album\n(\n AlbumId INTEGER NO...	ddl
21	f4b619e3-ec27-5093-a24f-938a38bd48a2-ddl	None	CREATE INDEX IFK_PlaylistTrackTrackId ON Playl...	ddl
0	51cf1d6d-7637-5b87-b9e7-31c577fbde59-doc	None	Our business defines OTIF score as the percent...	documentation

df_ddl = vn.run_sql("SELECT type, sql FROM sqlite_master WHERE sql is not null")df_ddlfor ddl in df_ddl['sql'].to_list(): ddl = strip_brackets(ddl) vn.train(ddl=ddl)

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 [10]:
```
ts_start = time()
```

In [11]:
```
vn.ask(question="Show me a list of tables in the SQLite database")
```

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

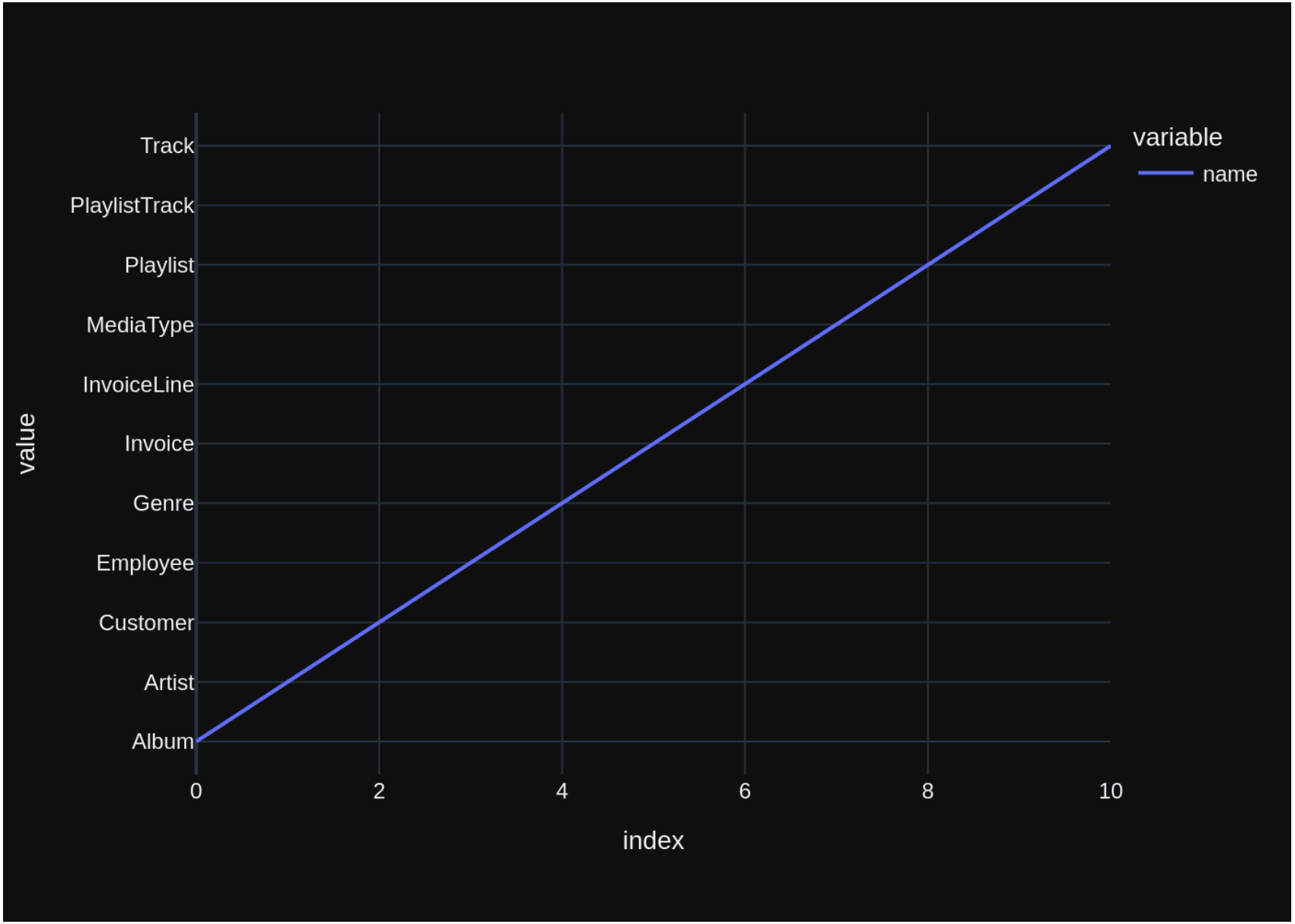
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE Playlist\n(\n   PlaylistId INTEGER NOT NULL,\n   Name NVARCHAR(120),\n   CONSTRAINT PK_Playlist PRIMARY KEY (PlaylistId)\n)\n\n\n   CREATE TABLE IF NOT EXISTS t_person (\n   id INT PRIMARY KEY,\n   name VARCHAR(100),\n   email text,\n   age INT\n)\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 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) 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 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 (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_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 PRIMARY 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 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 UPDATE NO ACTION\n)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'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 Co

```
untry, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT
5'}, {'role': 'user', 'content': 'What person has a name of "John Doe"?'}, {'role': 'assistant', 'content':
"SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': 'Show me a list of tables in
the SQLite database'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE Playlist\n(\n    PlaylistId INTEGER  NOT NULL,\n    Name NVAR
CHAR(120),\n    CONSTRAINT PK_Playlist PRIMARY KEY  (PlaylistId)\n)\n\n\n    CREATE TABLE IF NOT EXISTS t_p
erson (\n        id INT PRIMARY KEY,\n        name VARCHAR(100),\n        email text,\n        age INT\n
)\n\n\nCREATE TABLE InvoiceLine\n(\n    InvoiceLineId INTEGER  NOT NULL,\n    InvoiceId INTEGER  NOT NUL
L,\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 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 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 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 MediaType\n(\n    MediaTypeId INTEGER  NOT NULL,\n    Name NVA
RCHAR(120),\n    CONSTRAINT PK_MediaType PRIMARY KEY  (MediaTypeId)\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\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 TABLE Genre\n(\n    GenreId INTEGER
NOT NULL,\n    Name NVARCHAR(120),\n    CONSTRAINT PK_Genre PRIMARY KEY  (GenreId)\n)\n\nCREATE TABLE Invoi
ce\n(\n    InvoiceId INTEGER  NOT NULL,\n    CustomerId INTEGER  NOT NULL,\n    InvoiceDate DATETIME  NOT N
ULL,\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 (Customer
Id) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\n\n===Additional Context \n\nOur business defines OT
IF score as the percentage of orders that are delivered on time and in full\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. 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"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"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(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "What person has a name of \"John Doe\"?"}, {"role": "assistant", "content": "SELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content": "Show me a list of tables in the SQLite database"}]

Add of existing embedding ID: c9165031-db41-5792-8000-bd3db552c193-sql
Insert of existing embedding ID: c9165031-db41-5792-8000-bd3db552c193-sql

```
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:18:32.516417196Z', 'message': {'role': 'assistant',
'content': "SELECT name FROM sqlite_master WHERE type='table';"}, 'done_reason': 'stop', 'done': True, 'tot
al_duration': 78184281036, 'load_duration': 1583862687, 'prompt_eval_count': 1197, 'prompt_eval_duration':
74603838000, 'eval_count': 11, 'eval_duration': 1693920000}
SELECT name FROM sqlite_master WHERE type='table';
Output from LLM: SELECT name FROM sqlite_master WHERE type='table';
Extracted SQL: SELECT name FROM sqlite_master WHERE type='table'
SELECT name FROM sqlite_master WHERE type='table'
            name
0          Album
1         Artist
2       Customer
3       Employee
4          Genre
5        Invoice
6    InvoiceLine
7      MediaType
8       Playlist
9   PlaylistTrack
10          Track
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: 'Show me a list of tables in the SQLite database'\n\nThe DataFram
e was produced using this query: SELECT name FROM sqlite_master WHERE type='table'\n\nThe following is info
rmation about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n name    object\ndtype: obje
ct"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the data
frame? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, u
se an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:18:49.673915643Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nfig = px.bar(df, x='name', y='', title='SQLite Database Table
s')\nfig.update_layout(xaxis_title='Table Name', yaxis_title='')\nfig.show()\n```"}, 'done_reason': 'stop',
'done': True, 'total_duration': 17013979942, 'load_duration': 1311220, 'prompt_eval_count': 159, 'prompt_ev
al_duration': 9468008000, 'eval_count': 47, 'eval_duration': 7443695000}
```


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

Out[11]:
```
("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 [12]:
```
vn.ask(question="How many records are in table called customer")
```

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

[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE 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(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) 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 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 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   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\n\n   CREATE TABLE IF NOT EXISTS t_person (\n   id INT PRIMARY KEY,\n   name VARCHAR(100),\n   email text,\n   age INT\n   )\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 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 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 (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   PlaylistId INTEGER NOT NULL,\n   Name NVARCHAR(120),\n   CONSTRAINT PK_Playlist PRIMARY KEY (PlaylistId)\n)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and

```
answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content':
'How many records are in table called customer'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM Cu
stomer'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role':
'assistant', 'content': 'SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER
BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite data
base'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'us
er', 'content': 'What person has a name of "John Doe"?'}, {'role': 'assistant', 'content': "SELECT * FROM t
_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': 'How many records are in table called custom
er'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE 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\n\n    CREAT
E TABLE IF NOT EXISTS t_person (\n        id INT PRIMARY KEY,\n        name VARCHAR(100),\n        email te
xt,\n        age INT\n    )\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    FOR
EIGN 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  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 TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVARCHAR(200)  NOT NUL
```

L,\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 (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   PlaylistId INTEGER NOT NULL,\n   Name NVARCHAR(120),\n   CONSTRAINT PK_Playlist PRIMARY KEY (PlaylistId)\n)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "How many 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(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "Show me a list of tables in the SQLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "content": "What person has a name of \"John Doe\"?"}, {"role": "assistant", "content": "SELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content": "How many records are in table called customer"}]

Add of existing embedding ID: ad7073e4-6bfb-593d-802a-8b0d388fafa4-sql
Insert of existing embedding ID: ad7073e4-6bfb-593d-802a-8b0d388fafa4-sql

```
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:20:15.694314757Z', 'message': {'role': 'assistant',
'content': 'SELECT COUNT(*) FROM Customer'}, 'done_reason': 'stop', 'done': True, 'total_duration': 8536618
6382, 'load_duration': 543377, 'prompt_eval_count': 1348, 'prompt_eval_duration': 84180170000, 'eval_coun
t': 6, 'eval_duration': 853315000}
SELECT COUNT(*) FROM Customer
SELECT COUNT(*) FROM Customer
   COUNT(*)
0        59
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: 'How many 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."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:20:34.602474984Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = go.Figure(data=[g
o.Indicator(\n    title='Count of Customer Records',\n    mode='number+delta',\n    value=df.iloc[0, 0]
\n)])\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 18771724952, 'load_durat
ion': 687443, 'prompt_eval_count': 153, 'prompt_eval_duration': 9111023000, 'eval_count': 60, 'eval_duratio
n': 9566004000}
```


Count of Customer Records
59

Out[12]:
```
('SELECT COUNT(*) FROM Customer',
    COUNT(*)
 0        59,
 Figure({
     'data': [{'mode': 'number+delta', 'title': {'text': 'Count of Customer Records'}, 'type': 'indicato
r', 'value': 59}],
     'layout': {'template': '...'}
 }))
```

In [ ]:
```
```

In [13]:
```
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 4, updating n_results = 4
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

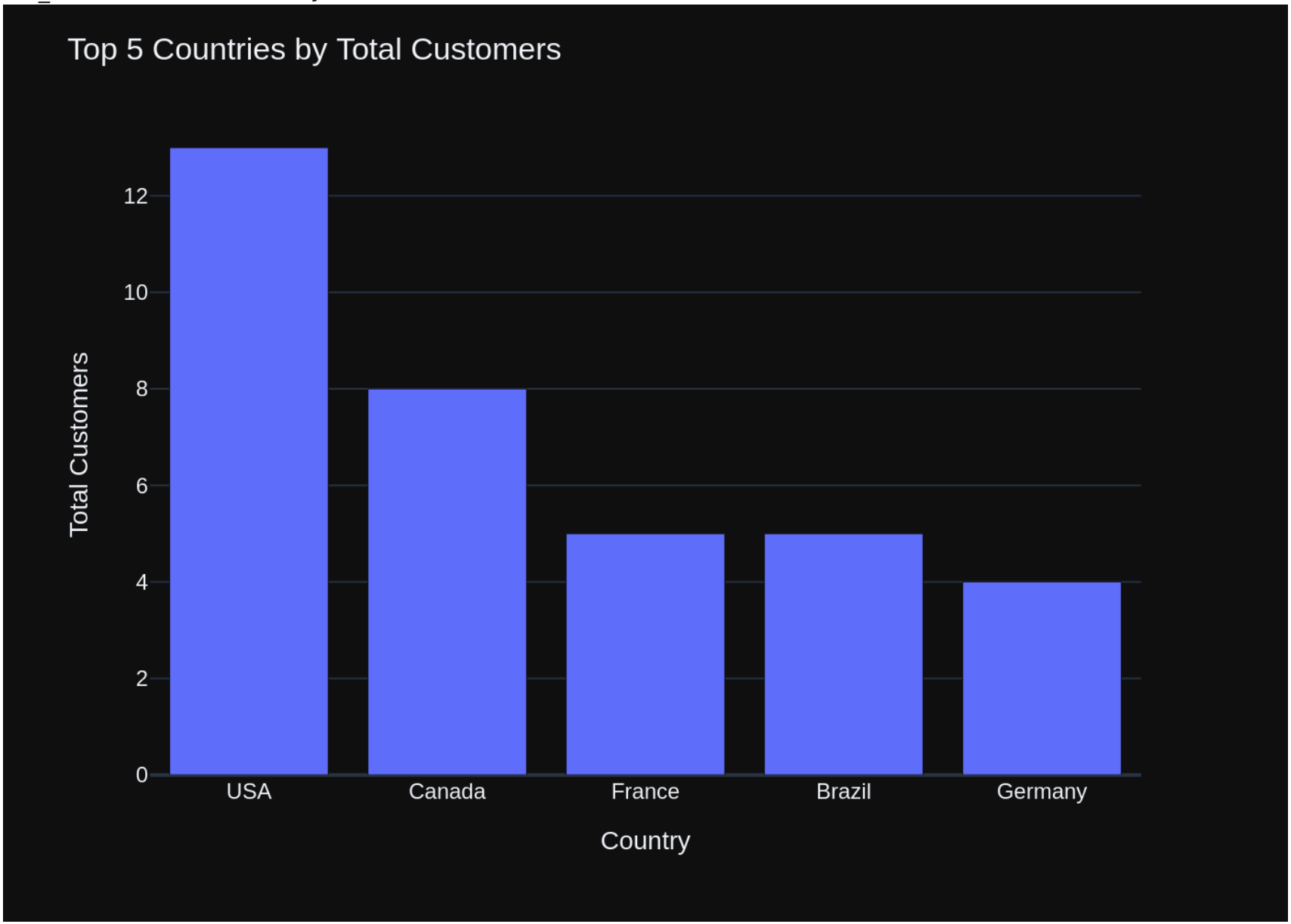
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE Customer\n(\n   CustomerId INTEGER NOT NULL,\n   FirstName NVARCHAR(40) NOT NULL,\n   LastName NVARCHAR(20) NOT NULL,\n   Company NVARCHAR(80),\n   Address NVARCHAR(70),\n   City NVARCHAR(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) 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 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 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   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 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   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 MediaType\n(\n   MediaTypeId INTEGER NOT NULL,\n   Name NVARCHAR(120),\n   CONSTRAINT PK_MediaType PRIMARY KEY (MediaTypeId)\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 INDEX IFK_CustomerSupportRepId 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   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 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 (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 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\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\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': 'w
hat are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country,
COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers 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': 'Show me a list of tables in the SQLite datab
ase'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'use
r', 'content': 'What person has a name of "John Doe"?'}, {'role': 'assistant', 'content': "SELECT * FROM t_
person WHERE name = 'John Doe'"}, {'role': 'user', 'content': 'what are the top 5 countries that customers
come from?'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and 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\n===Additional Context \n\nOur business defines OTIF score as the
percentage of orders that are delivered on time and in full\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": "w
hat are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country,
COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers 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": "Show me a list of tables in the SQLite datab
ase"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "use
r", "content": "What person has a name of \"John Doe\"?"}, {"role": "assistant", "content": "SELECT * FROM
t_person WHERE name = 'John Doe'"}, {"role": "user", "content": "what are the top 5 countries that customer
s come from?"}]
Add of existing embedding ID: 5698caa9-f558-5bc3-ab17-53fdf8d8e27a-sql
Insert of existing embedding ID: 5698caa9-f558-5bc3-ab17-53fdf8d8e27a-sql
```

```
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:22:09.499626388Z', 'message': {'role': 'assistant',
'content': 'SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCust
omers DESC\nLIMIT 5'}, 'done_reason': 'stop', 'done': True, 'total_duration': 94754905379, 'load_duration':
619715, 'prompt_eval_count': 1438, 'prompt_eval_duration': 90196403000, 'eval_count': 26, 'eval_duration':
4265919000}
SELECT Country, COUNT(*) AS TotalCustomers
FROM Customer
GROUP BY Country
ORDER BY TotalCustomers DESC
LIMIT 5
SELECT Country, COUNT(*) AS TotalCustomers
FROM Customer
GROUP BY Country
ORDER BY TotalCustomers DESC
LIMIT 5
   Country  TotalCustomers
0      USA              13
1   Canada               8
2   France               5
3   Brazil               5
4  Germany               4
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: 'what are the top 5 countries that customers come from?'\n\nThe D
ataFrame was produced using this query: SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY
Country\nORDER BY TotalCustomers DESC\nLIMIT 5\n\nThe following is information about the resulting pandas D
ataFrame 'df': \nRunning df.dtypes gives:\n Country           object\nTotalCustomers     int64\ndtype: obje
ct"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the data
frame? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, u
se an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:22:39.100457406Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nimport pandas as pd\n\nfig = px.bar(df, x='Country', y='Total
Customers')\nfig.update_layout(title='Top 5 Countries by Total Customers', xaxis_title='Country', yaxis_tit
le='Total Customers')\n\nif len(df) == 1:\n    fig = px.bar(df, x=df.index, y=df.values.tolist()[0], color_
discrete_sequence=['red'])\n    fig.update_layout(title='Single Country Data', xaxis_title='Index', yaxis_t
itle='Value')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 29459537038, 'lo
```

ad_duration': 931023, 'prompt_eval_count': 180, 'prompt_eval_duration': 10651134000, 'eval_count': 116, 'eval_duration': 18748247000}



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

More SQL questions

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

In [14]:
```
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 4, updating n_results = 4
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

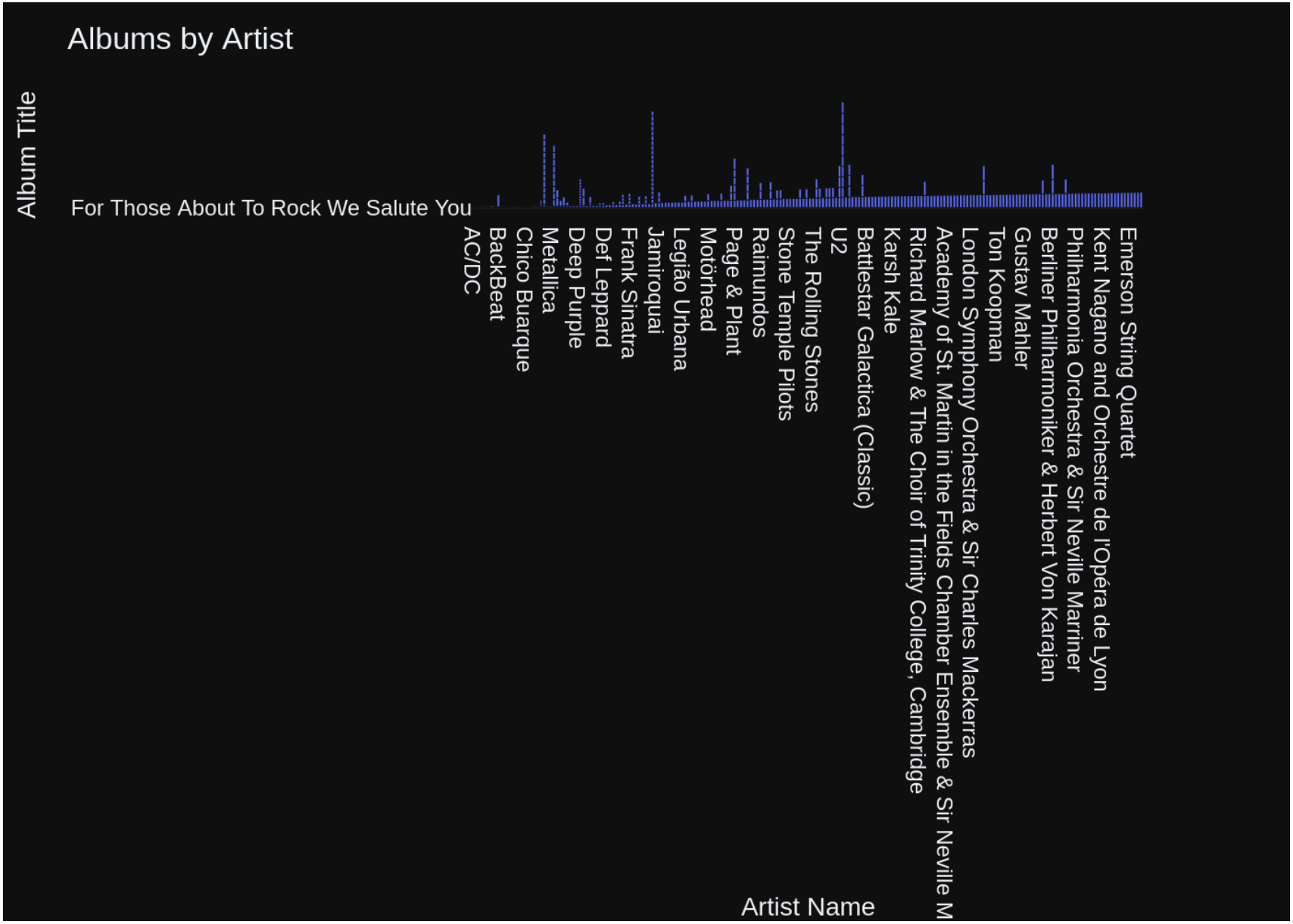
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_AlbumArtistId ON Album (ArtistId)\n\nCREATE 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\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 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 UPDATE NO ACTION\n)\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 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 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\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': 'What person has a name of "John Doe"?'}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': '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 corresponding artist names \n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo

```
rmat instructions. \n===Tables \nCREATE INDEX IFK_AlbumArtistId ON 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\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are del
ivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please gen
erate a valid SQL query without any explanations for the question. \n2. If the provided context is almost s
ufficient but requires knowledge of a specific string in a particular column, please generate an intermedia
te SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermedi
ate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please
use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the
answer exactly as it was given before. \n"}, {"role": "user", "content": "Show me a list of tables in the S
QLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"},
{"role": "user", "content": "What person has a name of \"John Doe\"?"}, {"role": "assistant", "content": "S
ELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content": "what are the top 5 countries
that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS TotalCustomers\n
FROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "How
many records are in table called customer"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM Custome
r"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist names  \n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:23:40.455618696Z', 'message': {'role': 'assistant',
'content': '```\nSELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.Artis
tId;\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 61222295438, 'load_duration': 1279711,
'prompt_eval_count': 897, 'prompt_eval_duration': 55347614000, 'eval_count': 34, 'eval_duration': 553951600
0}
```

```
SELECT a.Title, a.ArtistId, ar.Name
FROM Album a
```

```
JOIN Artist ar ON a.ArtistId = ar.ArtistId;
```
Output from LLM: ```
SELECT a.Title, a.ArtistId, ar.Name
FROM Album a
JOIN Artist ar ON a.ArtistId = ar.ArtistId;
```
Extracted SQL: SELECT a.Title, a.ArtistId, ar.Name
FROM Album a
JOIN Artist ar ON a.ArtistId = ar.ArtistId
SELECT a.Title, a.ArtistId, ar.Name
FROM Album a
JOIN Artist ar ON a.ArtistId = ar.ArtistId

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

[347 rows x 3 columns]
Ollama parameters:
model=llama3:latest,
```

```
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all albums and their corresponding artist names
\n'\n\nThe DataFrame was produced using this query: SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN
Artist ar ON a.ArtistId = ar.ArtistId\n\nThe following is information about the resulting pandas DataFrame
'df': \nRunning df.dtypes gives:\n Title       object\nArtistId     int64\nName        object\ndtype: objec
t"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataf
rame? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, us
e an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:24:01.402471198Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nimport pandas as pd\n\nfig = px.bar(df, x='Name', y='Title')
\nfig.update_layout(title='Albums by Artist', xaxis_title='Artist Name', yaxis_title='Album Title')\n\nfig.
show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 20774958592, 'load_duration': 306216
9, 'prompt_eval_count': 191, 'prompt_eval_duration': 11384319000, 'eval_count': 58, 'eval_duration': 923780
6000}
```


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

Out[14]: ('SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId',

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

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

```
                           'Nash Ensemble', 'Philip Glass Ensemble'], dtype=object),
                'xaxis': 'x',
                'y': array(['For Those About To Rock We Salute You', 'Balls to the Wall',
                           'Restless and Wild', ..., "Monteverdi: L'Orfeo",
                           'Mozart: Chamber Music',
                           'Koyaanisqatsi (Soundtrack from the Motion Picture)'], dtype=object),
                'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'title': {'text': 'Albums by Artist'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Artist Name'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Album Title'}}}
 }))
```

In [15]:
```
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 5, updating n_results = 5
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and 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\n===Additional Context \n\nOur business defines OTIF score as the percen
tage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided contex
t is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the p
rovided context is almost sufficient but requires knowledge of a specific string in a particular column, pl
ease generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with
a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't
be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered
before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': '  \n
List all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT a.Titl
e, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'conte
nt': 'What person has a name of "John Doe"?'}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHE
RE name = 'John Doe'"}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'r
ole': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'conte
nt': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Co
untry, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers 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'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
```

[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_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 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 UPDATE NO ACTION\n)\n\nCREATE 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_InvoiceLineTrackId 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 (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 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 TABLE Playlist\n(\n   PlaylistId INTEGER NOT NULL,\n   Name NVARCHAR(120),\n   CONSTRAINT PK_Playlist PRIMARY KEY (PlaylistId)\n)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "What person has a name of \"John Doe\"?"}, {"role": "assistant", "content": "SELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content": "Show me a list of tables in the SQLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"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 name containing \"What\" (case-insensitive)\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:25:02.657844591Z', 'message': {'role': 'assistant', 'content': "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%';"}, 'done_reason': 'stop', 'done': True, 'total_duration': 60980075046, 'load_duration': 625424, 'prompt_eval_count': 941, 'prompt_eval_duration': 58026296000, 'eval_count': 16, 'eval_duration': 2534942000}

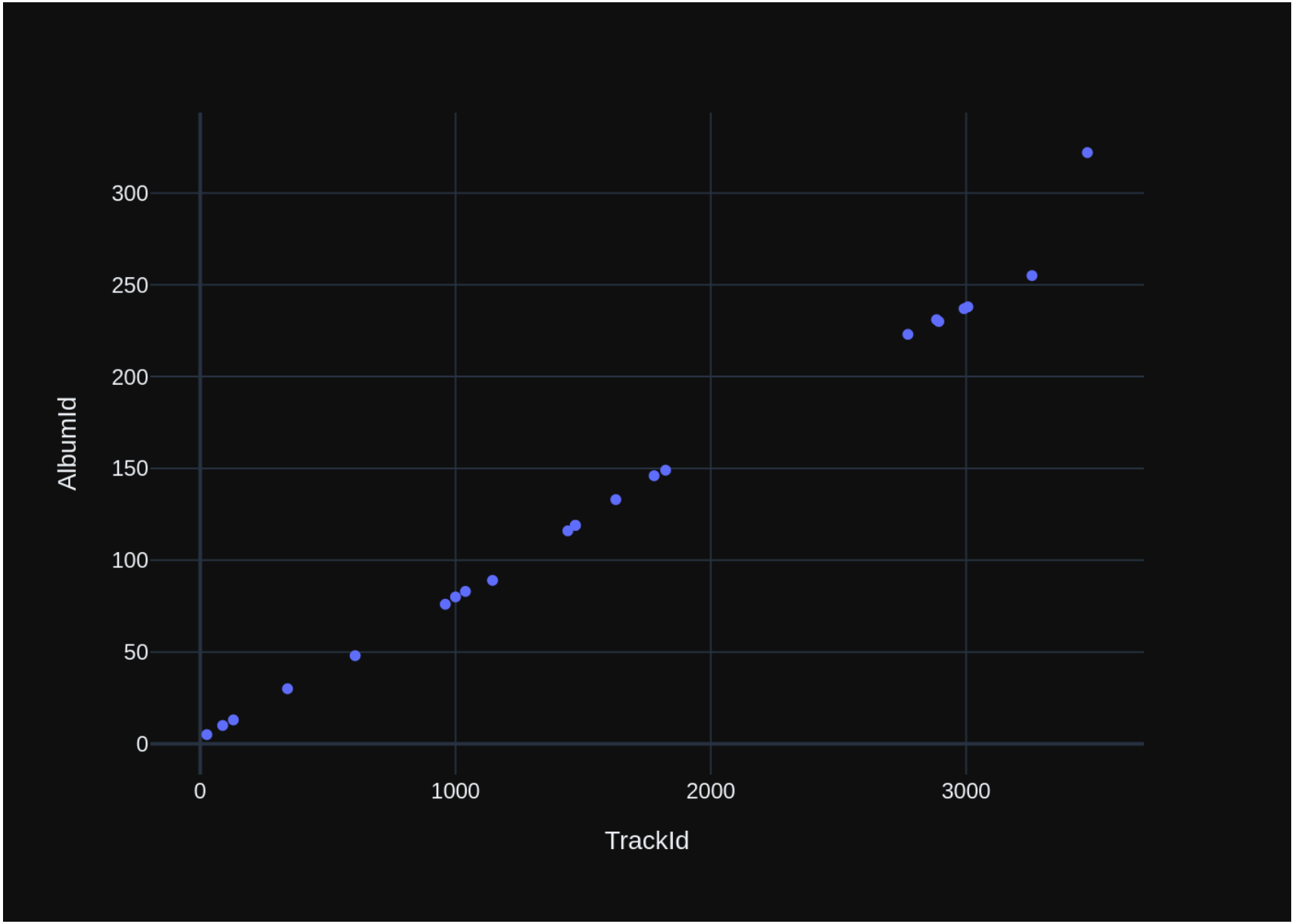
```
SELECT *
FROM Track
WHERE LOWER(Name) LIKE '%what%';
Output from LLM: SELECT *
FROM Track
WHERE LOWER(Name) LIKE '%what%';
Extracted SQL: SELECT *
FROM Track
WHERE LOWER(Name) LIKE '%what%'
SELECT *
FROM Track
WHERE LOWER(Name) LIKE '%what%'
    TrackId                                        Name  AlbumId  \
0        26                                What It Takes        5
1        88                                 What You Are       10
2       130                             Do what cha wanna       13
3       342                   What is and Should Never Be       30
4       607                                      So What       48
5       960                                   What A Day       76
6      1000                                What If I Do?       80
7      1039                              What Now My Love       83
8      1145                                   Whatsername       89
9      1440            Whatever It Is, I Just Can't Stop      116
10     1469                         Look What You've Done      119
11     1470                             Get What You Need      119
12     1628              What Is And What Should Never Be      133
13     1778  You're What's Happening (In The World Today)      146
14     1823                                      So What      149
15     2772          I Don't Know What To Do With Myself      223
16     2884                                What Kate Did      231
17     2893                     Whatever the Case May Be      230
18     2992    I Still Haven't Found What I'm Looking for      237
19     3007    I Still Haven't Found What I'm Looking For      238
20     3258              Whatever Gets You Thru the Night      255
21     3475                         What Is It About Men      322

    MediaTypeId  GenreId                                        Composer  \
0             1        1            Steven Tyler, Joe Perry, Desmond Child
1             1        1                         Audioslave/Chris Cornell
2             1        2                                      George Duke
3             1        1                       Jimmy Page/Robert Plant
4             1        2                                      Miles Davis
```

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

	Milliseconds	Bytes	UnitPrice
0	310622	10144730	0.99
1	249391	5988186	0.99
2	274155	9018565	0.99
3	260675	8497116	0.99
4	564009	18360449	0.99
5	158275	5203430	0.99
6	302994	9929799	0.99
7	149995	4913383	0.99
8	252316	8244843	0.99
9	247222	8249453	0.99
10	230974	7517083	0.99
11	247719	8043765	0.99
12	287973	9369385	0.99
13	142027	4631104	0.99
14	189152	6162894	0.99
15	221387	7251478	0.99
16	2610250	484583988	1.99
17	2616410	183867185	1.99
18	353567	11542247	0.99
19	280764	9306737	0.99
20	215084	3499018	0.99
21	209573	3426106	0.99

Ollama parameters:

```
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Find all tracks with a name containing \"What\" (case-in
sensitive)\n'\n\nThe DataFrame was produced using this query: SELECT * \nFROM Track \nWHERE LOWER(Name) LIK
E '%what%'\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes g
ives:\n TrackId           int64\nName             object\nAlbumId           int64\nMediaTypeId       int64
\nGenreId           int64\nComposer         object\nMilliseconds      int64\nBytes             int64\nUnitP
rice       float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code t
o chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is onl
y one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any expla
nations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:25:32.497452247Z', 'message': {'role': 'assistant',
'content': '```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = px.bar(df, x="Nam
e", y="Milliseconds")\nfig.update_layout(title=\'What tracks duration\')\nfig.show()\n\n# or if there is on
ly one value in the dataframe:\nfig = go.Figure(data=[go.Indicator(\n    mode = "number+delta",\n    value
= df[\'Milliseconds\'].mean(),\n    title.value = "Mean duration of What tracks"\n)])\nfig.show()\n```'},
'done_reason': 'stop', 'done': True, 'total_duration': 29585662912, 'load_duration': 3131245, 'prompt_eval_
count': 222, 'prompt_eval_duration': 13165248000, 'eval_count': 101, 'eval_duration': 16267784000}
```


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

Out[15]: ("SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'",

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

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

    Milliseconds      Bytes  UnitPrice
0         310622   10144730       0.99
1         249391    5988186       0.99
2         274155    9018565       0.99
3         260675    8497116       0.99
4         564009   18360449       0.99
5         158275    5203430       0.99
6         302994    9929799       0.99
7         149995    4913383       0.99
8         252316    8244843       0.99
9         247222    8249453       0.99
10        230974    7517083       0.99
11        247719    8043765       0.99
12        287973    9369385       0.99
13        142027    4631104       0.99
14        189152    6162894       0.99
15        221387    7251478       0.99
16       2610250  484583988       1.99
17       2616410  183867185       1.99
18        353567   11542247       0.99
19        280764    9306737       0.99
20        215084    3499018       0.99
21        209573    3426106       0.99  ,
Figure({
    'data': [{'hovertemplate': 'TrackId=%{x}<br>AlbumId=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'symbol': 'circle'},
              'mode': 'markers',
              'name': '',
              'orientation': 'v',
              'showlegend': False,
              'type': 'scatter',
              'x': array([  26,   88,  130,  342,  607,  960, 1000, 1039, 1145, 1440, 1469, 1470,
                         1628, 1778, 1823, 2772, 2884, 2893, 2992, 3007, 3258, 3475]),
              'xaxis': 'x',
```

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

In [16]:
```
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 6, updating n_results = 6
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_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(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_InvoiceLineInvoiceId ON InvoiceLine (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 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 INDEX IFK_InvoiceLineTrackId 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 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) 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 INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\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   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\n\n   CREATE TABLE IF NOT EXISTS t_person (\n   id INT PRIMARY KEY,\n   name VARCHAR(100),\n   email text,\n   age INT\n   )\n\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': 'How many 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(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"},

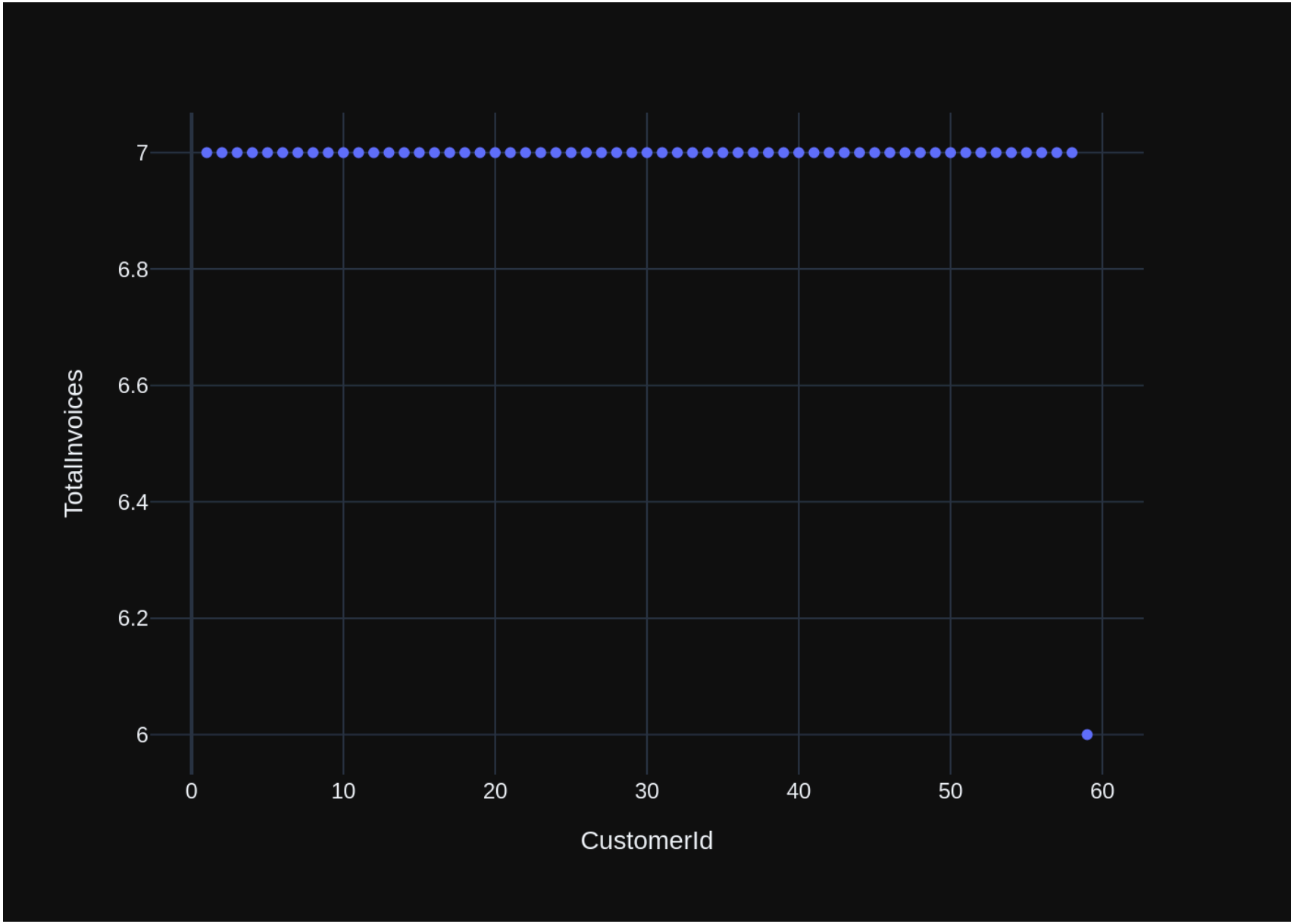
```
{'role': 'user', 'content': 'What person has a name of "John Doe"?'}, {'role': 'assistant', 'content': "SEL
ECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': 'Show me a list of tables in the
SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"},
{'role': 'user', 'content': '  \n    Get the total number of invoices for each customer\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_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\n\n    CREATE TABLE IF NOT EXISTS t_pers
on (\n        id INT PRIMARY KEY,\n        name VARCHAR(100),\n        email text,\n        age INT\n    )
\n\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are deli
vered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please gene
rate a valid SQL query without any explanations for the question. \n2. If the provided context is almost su
fficient but requires knowledge of a specific string in a particular column, please generate an intermediat
e SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermedia
te_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please
```

use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "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(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": "What person has a name of \"John Doe\"?"}, {"role": "assistant", "content": "SELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content": "Show me a list of tables in the SQLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "content": " \n   Get the total number of invoices for each customer\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:26:51.341702279Z', 'message': {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId'}, 'done_reason': 'stop', 'done': True, 'total_duration': 78707413996, 'load_duration': 629714, 'prompt_eval_count': 1209, 'prompt_eval_duration': 75226309000, 'eval_count': 19, 'eval_duration': 3049425000}
```
SELECT CustomerId, COUNT(*) AS TotalInvoices
FROM Invoice
GROUP BY CustomerId
```
```
SELECT CustomerId, COUNT(*) AS TotalInvoices
FROM Invoice
GROUP BY CustomerId
```

	CustomerId	TotalInvoices
0	1	7
1	2	7
2	3	7
3	4	7
4	5	7
5	6	7
6	7	7
7	8	7
8	9	7
9	10	7
10	11	7
11	12	7
12	13	7
13	14	7
14	15	7
15	16	7
16	17	7

17	18	7
18	19	7
19	20	7
20	21	7
21	22	7
22	23	7
23	24	7
24	25	7
25	26	7
26	27	7
27	28	7
28	29	7
29	30	7
30	31	7
31	32	7
32	33	7
33	34	7
34	35	7
35	36	7
36	37	7
37	38	7
38	39	7
39	40	7
40	41	7
41	42	7
42	43	7
43	44	7
44	45	7
45	46	7
46	47	7
47	48	7
48	49	7
49	50	7
50	51	7
51	52	7
52	53	7
53	54	7
54	55	7
55	56	7
56	57	7
57	58	7
58	59	6

```
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Get the total number of invoices for each customer\n'\n
\nThe DataFrame was produced using this query: SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\n
GROUP BY CustomerId\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning d
f.dtypes gives:\n CustomerId       int64\nTotalInvoices    int64\ndtype: object"}, {"role": "user", "conten
t": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a
pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with o
nly Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:27:10.686571718Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = go.Figure(data=[p
x.bar(df, x='CustomerId', y='TotalInvoices', title='Total Invoices by Customer')])\n\nfig.show()\n```"}, 'd
one_reason': 'stop', 'done': True, 'total_duration': 19187557055, 'load_duration': 978209, 'prompt_eval_cou
nt': 177, 'prompt_eval_duration': 10531083000, 'eval_count': 54, 'eval_duration': 8588786000}
```


7
6.8
6.6
6.4
6.2
6
TotalInvoices
0
10
20
30
40
50
60
CustomerId

Out[16]: ('SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY 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': [{'hovertemplate': 'CustomerId=%{x}<br>TotalInvoices=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'symbol': 'circle'},
              'mode': 'markers',
              'name': '',
              'orientation': 'v',
              'showlegend': False,
              'type': 'scatter',
              'x': array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
                          19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
                          37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
                          55, 56, 57, 58, 59]),
              'xaxis': 'x',
              'y': array([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': {'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 [17]:
```
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 7, updating n_results = 7
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and 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\n===Additional Cont
ext \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full
\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    Get the total number of invoices for each customer
```

\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP B
Y CustomerId'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'ro
le': 'assistant', 'content': 'SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\n
ORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': '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 corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELE
CT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId'}, {'role': 'use
r', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'ass
istant', 'content': "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content':
'What person has a name of "John Doe"?'}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE na
me = 'John Doe'"}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role':
'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': '
\n   Find the total number of invoices per country:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE Invoice\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

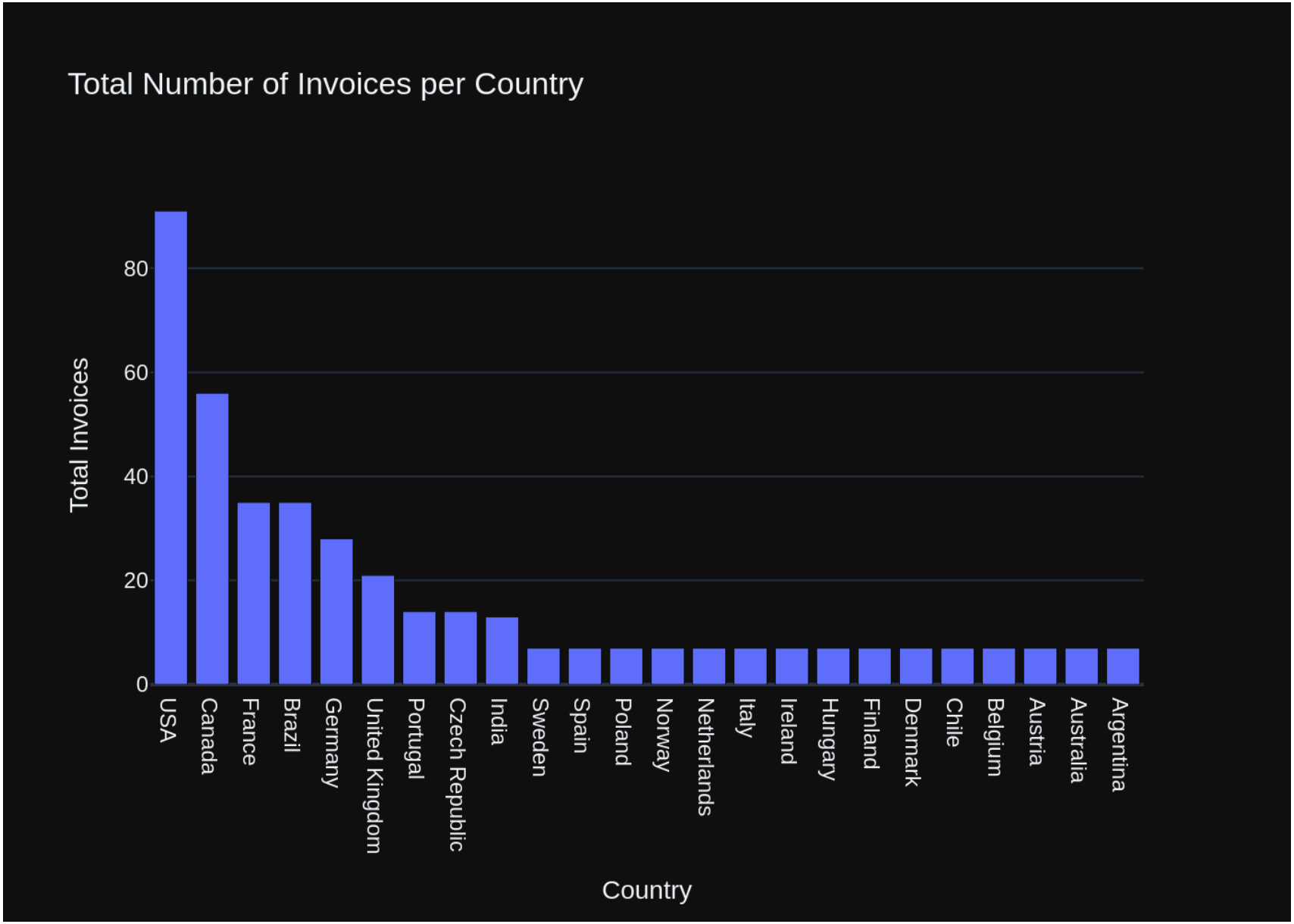
```
ETE NO ACTION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Track\n(\n    TrackId INTEGER  NOT NULL,\n    Name NVA
RCHAR(200)  NOT NULL,\n    AlbumId INTEGER,\n    MediaTypeId INTEGER  NOT NULL,\n    GenreId INTEGER,\n
Composer NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    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\n===Additional Cont
ext \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full
\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    Get the total number of invoices for each customer
\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP B
Y CustomerId"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"ro
le": "assistant", "content": "SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\n
ORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "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 corresponding artist names  \n"}, {"role": "assistant", "content": "SELE
CT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId"}, {"role": "use
r", "content": "  \n    Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "a
ssistant", "content": "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "conten
t": "What person has a name of \"John Doe\"?"}, {"role": "assistant", "content": "SELECT * FROM t_person WH
ERE name = 'John Doe'"}, {"role": "user", "content": "Show me a list of tables in the SQLite database"},
{"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "co
ntent": "  \n    Find the total number of invoices per country:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:28:50.621310533Z', 'message': {'role': 'assistant',
'content': '```\nSELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON
c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC;\n```'}, 'done_reason': 'sto
p', 'done': True, 'total_duration': 99797039031, 'load_duration': 572527, 'prompt_eval_count': 1456, 'promp
t_eval_duration': 91421492000, 'eval_count': 47, 'eval_duration': 7893056000}
```

```
SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
GROUP BY c.Country
```

```
ORDER BY TotalInvoices DESC;
```
Output from LLM: ```
```
SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
GROUP BY c.Country
ORDER BY TotalInvoices DESC;
```
Extracted SQL: SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
GROUP BY c.Country
ORDER BY TotalInvoices DESC
SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
GROUP BY c.Country
ORDER BY TotalInvoices DESC

	Country	TotalInvoices
0	USA	91
1	Canada	56
2	France	35
3	Brazil	35
4	Germany	28
5	United Kingdom	21
6	Portugal	14
7	Czech Republic	14
8	India	13
9	Sweden	7
10	Spain	7
11	Poland	7
12	Norway	7
13	Netherlands	7
14	Italy	7
15	Ireland	7
16	Hungary	7
17	Finland	7
18	Denmark	7
19	Chile	7
20	Belgium	7
21	Austria	7

```
22       Australia             7
23       Argentina             7
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Find the total number of invoices per country:\n'\n\nThe
DataFrame was produced using this query: SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Custom
er c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC\n\nThe
following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Country
object\nTotalInvoices     int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python
plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If
there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer wi
th any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:29:14.18934331Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nimport numpy as np\n\nfig = px.bar(df, x='Country', y='TotalI
nvoices', title='Total Number of Invoices per Country')\n\nfig.update_layout(xaxis_title='Country',\n
yaxis_title='Total Invoices',\n                  legend_title='Legend')\n\nfig.show()\n```"}, 'done_reaso
n': 'stop', 'done': True, 'total_duration': 23396539323, 'load_duration': 2508325, 'prompt_eval_count': 20
3, 'prompt_eval_duration': 12087476000, 'eval_count': 70, 'eval_duration': 11198655000}
```


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

Out[17]: ('SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC',

	Country	TotalInvoices
0	USA	91
1	Canada	56
2	France	35
3	Brazil	35
4	Germany	28
5	United Kingdom	21
6	Portugal	14
7	Czech Republic	14
8	India	13
9	Sweden	7
10	Spain	7
11	Poland	7
12	Norway	7
13	Netherlands	7
14	Italy	7
15	Ireland	7
16	Hungary	7
17	Finland	7
18	Denmark	7
19	Chile	7
20	Belgium	7
21	Austria	7
22	Australia	7
23	Argentina	7,

```
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'Country=%{x}<br>TotalInvoices=%{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', 'United Kingdom',
                          'Portugal', 'Czech Republic', 'India', 'Sweden', 'Spain', 'Poland',
                          'Norway', 'Netherlands', 'Italy', 'Ireland', 'Hungary', 'Finland',
                          'Denmark', 'Chile', 'Belgium', 'Austria', 'Australia', 'Argentina'],
```

```
                           dtype=object),
                  'xaxis': 'x',
                  'y': array([91, 56, 35, 35, 28, 21, 14, 14, 13,  7,  7,  7,  7,  7,  7,  7,  7,  7,
                              7,  7,  7,  7,  7,  7]),
                  'yaxis': 'y'}],
       'layout': {'barmode': 'relative',
                  'legend': {'title': {'text': 'Legend'}, 'tracegroupgap': 0},
                  'template': '...',
                  'title': {'text': 'Total Number of Invoices per Country'},
                  'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Country'}},
                  'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total Invoices'}}}
 }))
```

In [18]:
```
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 8, updating n_results = 8
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE 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 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 UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON InvoiceLine (TrackId)\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 (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_EmployeeReportsTo ON Employee (ReportsTo)\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(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) 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 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   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 INDEX IFK_CustomerSupportRepId ON Customer (SupportRepId)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role':

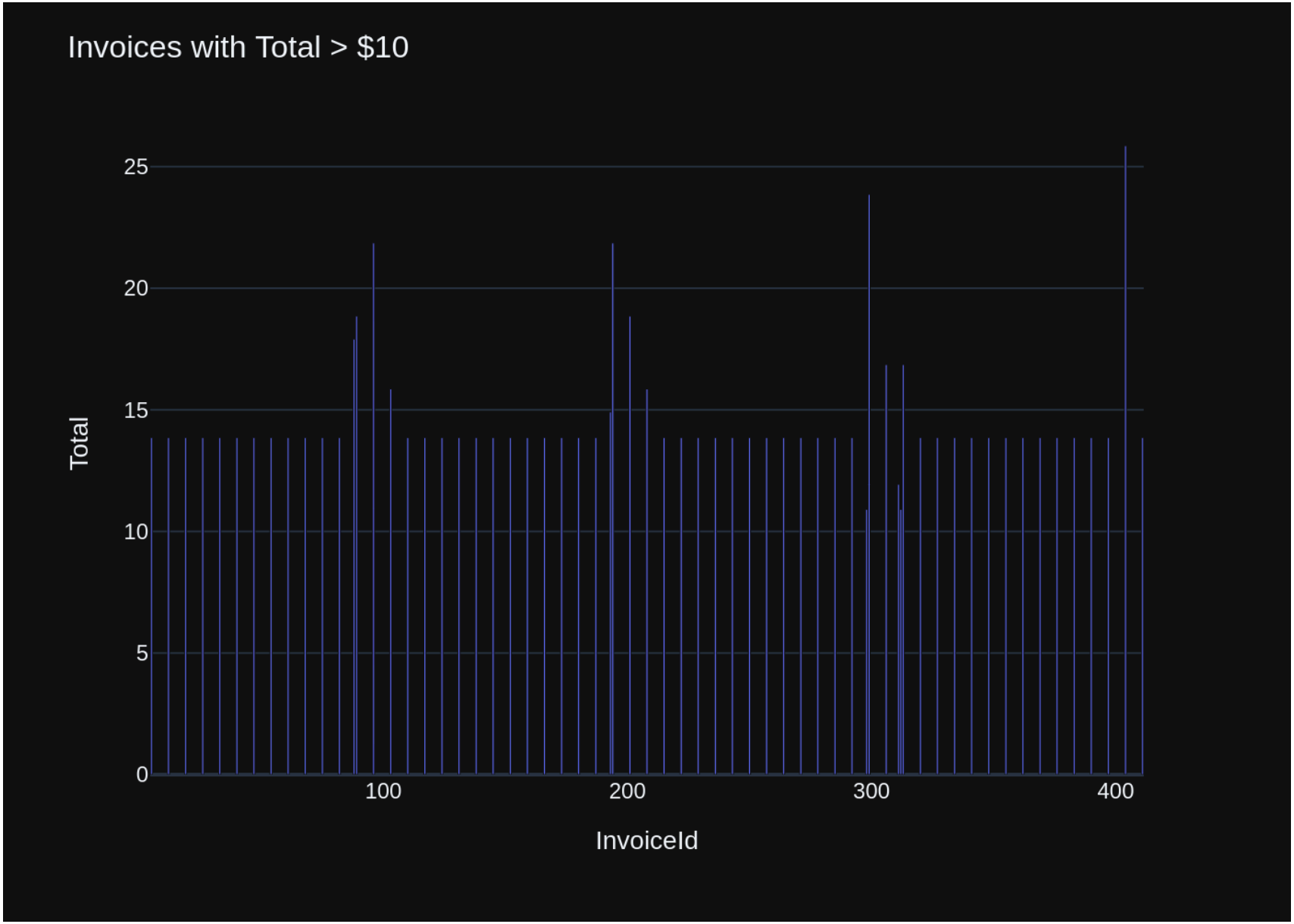
'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoi
ce i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC'}, {'role': 'user', 'c
ontent': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELEC
T Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers 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   List all albums and their c
orresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name\nFRO
M Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find all trac
ks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * \nFROM
Track \nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': 'Show me a list of tables in the SQL
ite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'ro
le': 'user', 'content': 'What person has a name of "John Doe"?'}, {'role': 'assistant', 'content': "SELECT
* FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': ' \n   List all invoices with a to
tal exceeding $10:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE 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\n===Additional Context
\n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\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": "  \n    Get the total number of invoices for each customer\n"},
{"role": "assistant", "content": "SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY Cust
omerId"}, {"role": "user", "content": "  \n    Find the total number of invoices per country:\n"}, {"role":
"assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoi
ce i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC"}, {"role": "user", "c
ontent": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELEC
T Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers 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    List all albums and their c
orresponding artist names  \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name\nFRO
M Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "  \n    Find all trac
ks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * \nFRO
M Track \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": "Show me a list of tables in the S
QLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"},
{"role": "user", "content": "What person has a name of \"John Doe\"?"}, {"role": "assistant", "content": "S
ELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content": "  \n    List all invoices wit
h a total exceeding $10:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:30:50.142335592Z', 'message': {'role': 'assistant',
'content': '```\nSELECT * \nFROM Invoice \nWHERE Total > 10.00\n```'}, 'done_reason': 'stop', 'done': True,
'total_duration': 95741819910, 'load_duration': 593281, 'prompt_eval_count': 1467, 'prompt_eval_duration':
92251687000, 'eval_count': 18, 'eval_duration': 2956871000}
```

```
SELECT *
FROM Invoice
WHERE Total > 10.00
```

```
```
Output from LLM: ```
SELECT *
FROM Invoice
WHERE Total > 10.00
```
Extracted SQL: SELECT *
FROM Invoice
WHERE Total > 10.00

SELECT *
FROM Invoice
WHERE Total > 10.00

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

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

[64 rows x 9 columns]
Ollama parameters:
```

```
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all invoices with a total exceeding $10:\n'\n\nThe
DataFrame was produced using this query: SELECT * \nFROM Invoice \nWHERE Total > 10.00\n\n\nThe following i
s information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n InvoiceId
int64\nCustomerId             int64\nInvoiceDate           object\nBillingAddress        object\nBillingCit
y           object\nBillingState          object\nBillingCountry        object\nBillingPostalCode     objec
t\nTotal                float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python
plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If
there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer wi
th any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:31:11.050785206Z', 'message': {'role': 'assistant',
'content': '```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = px.bar(df, x="Invo
iceId", y="Total")\nfig.update_layout(title=\'Invoices with Total > $10\')\nfig.show()\n```'}, 'done_reaso
n': 'stop', 'done': True, 'total_duration': 20737570152, 'load_duration': 2408317, 'prompt_eval_count': 21
2, 'prompt_eval_duration': 12575488000, 'eval_count': 51, 'eval_duration': 8095048000}
```


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

Out[18]:
```
('SELECT * \nFROM Invoice \nWHERE Total > 10.00\n',
    InvoiceId  CustomerId          InvoiceDate             BillingAddress  \
0           5          23  2009-01-11 00:00:00            69 Salem Street
1          12           2  2009-02-11 00:00:00    Theodor-Heuss-Straße 34
2          19          40  2009-03-14 00:00:00             8, Rue Hanovre
3          26          19  2009-04-14 00:00:00            1 Infinite Loop
4          33          57  2009-05-15 00:00:00            Calle Lira, 198
..        ...         ...                  ...                        ...
59        383          10  2013-08-12 00:00:00  Rua Dr. Falcão Filho, 155
60        390          48  2013-09-12 00:00:00      Lijnbaansgracht 120bg
61        397          27  2013-10-13 00:00:00            1033 N Park Ave
62        404           6  2013-11-13 00:00:00              Rilská 3174/6
63        411          44  2013-12-14 00:00:00            Porthaninkatu 9

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

[64 rows x 9 columns],
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'InvoiceId=%{x}<br>Total=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'v',
              'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
              'x': array([  5,  12,  19,  26,  33,  40,  47,  54,  61,  68,  75,  82,  88,  89,
                           96, 103, 110, 117, 124, 131, 138, 145, 152, 159, 166, 173, 180, 187,
                          193, 194, 201, 208, 215, 222, 229, 236, 243, 250, 257, 264, 271, 278,
```

```
                           285, 292, 298, 299, 306, 311, 312, 313, 320, 327, 334, 341, 348, 355,
                           362, 369, 376, 383, 390, 397, 404, 411]),
                'xaxis': 'x',
                'y': array([13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                           13.86, 13.86, 17.91, 18.86, 21.86, 15.86, 13.86, 13.86, 13.86, 13.86,
                           13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 14.91, 21.86,
                           18.86, 15.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                           13.86, 13.86, 13.86, 13.86, 10.91, 23.86, 16.86, 11.94, 10.91, 16.86,
                           13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                           13.86, 13.86, 25.86, 13.86]),
                'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'title': {'text': 'Invoices with Total > $10'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total'}}}
 }))
```

In [19]:

```
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 9, updating n_results = 9
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE 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 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   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 INDEX IFK_InvoiceLineInvoiceId ON InvoiceLine (InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineTrackId 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 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) 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 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   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 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 (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\n\n   CREATE TABLE IF NOT EXISTS t_person (\n   id INT PRIMARY KEY,\n   name VARCHAR(100),\n   email text,\n   age INT\n   )\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 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\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been

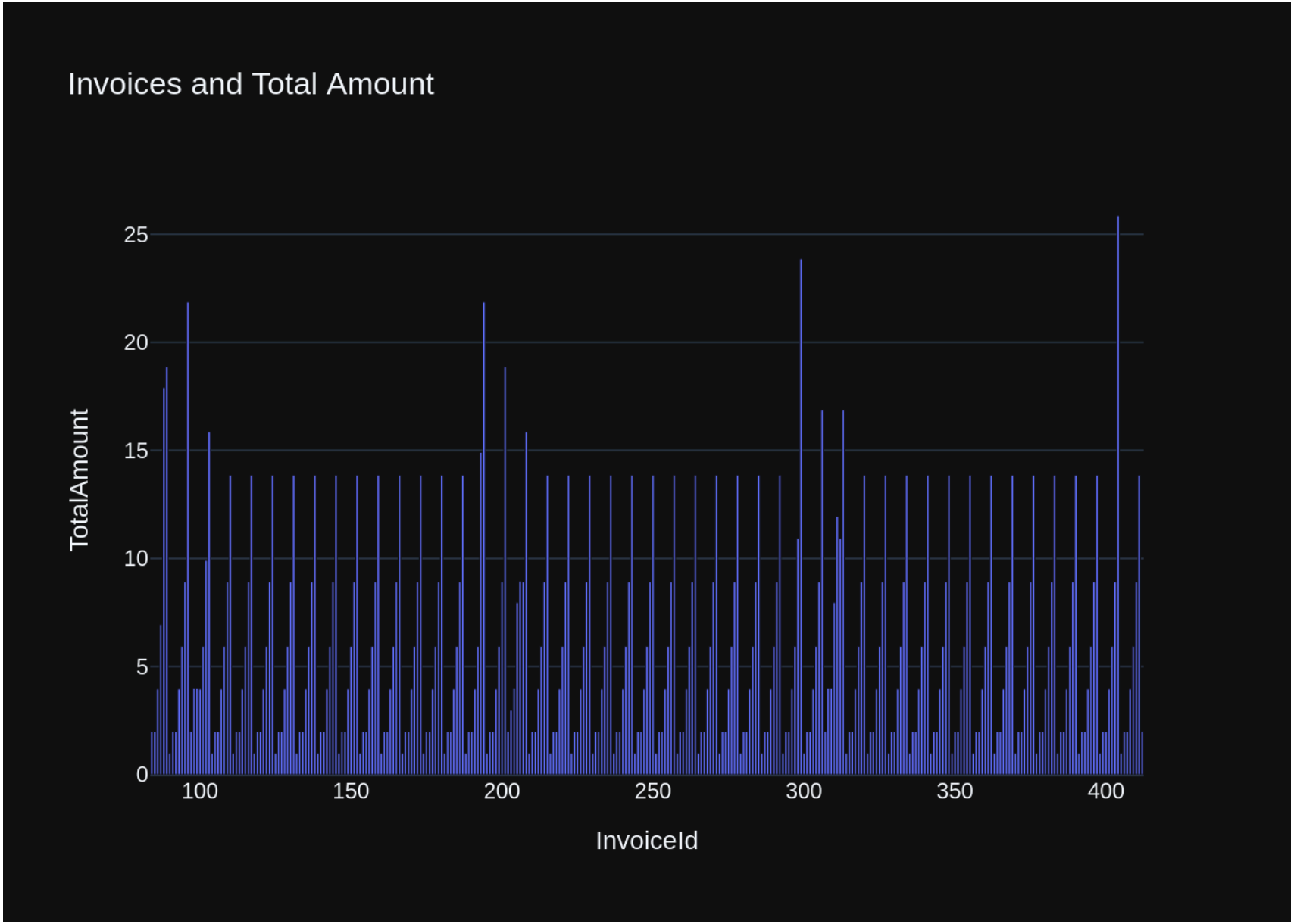
asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM Invoice \nWHERE Total > 10.00\n'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC'}, {'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(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': 'What person has a name of "John Doe"?'}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': '\n   Find all invoices since 2010 and the total amount invoiced:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and 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   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 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   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 INDEX IFK_InvoiceLineInvoiceId ON InvoiceLine (InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineTrackId 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 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) NOT NULL,\n   SupportRepId 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\n\n   CREATE TABLE IF NOT EXISTS t_person (\n
id INT PRIMARY KEY,\n   name VARCHAR(100),\n   email text,\n   age INT\n   )\n\n\nCREATE TA
BLE 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 (Playli
stId) \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\n===Additional Context \n\nOur business defines OTIF s
core as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If t
he provided context is sufficient, please generate a valid SQL query without any explanations for the quest
ion. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a par
ticular column, please generate an intermediate SQL query to find the distinct strings in that column. Prep
end the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please e
xplain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been
asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user",
"content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SE
LECT * \nFROM Invoice \nWHERE Total > 10.00\n"}, {"role": "user", "content": " \n   Get the total number
of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) AS TotalIn
voices\nFROM Invoice\nGROUP BY CustomerId"}, {"role": "user", "content": " \n   Find the total number of
invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalI
nvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY Total
Invoices DESC"}, {"role": "user", "content": "How many records are in table called customer"}, {"role": "as
sistant", "content": "SELECT COUNT(*) FROM Customer"}, {"role": "user", "content": "what are the top 5 coun
tries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS TotalCusto
mers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content":
" \n   List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELE
CT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId"}, {"role": "use
r", "content": " \n   Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "a
ssistant", "content": "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "conten
t": "What person has a name of \"John Doe\"?"}, {"role": "assistant", "content": "SELECT * FROM t_person WH
ERE name = 'John Doe'"}, {"role": "user", "content": "Show me a list of tables in the SQLite database"},
{"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "co

```
ntent": "  \n    Find all invoices since 2010 and the total amount invoiced:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:33:00.18029492Z', 'message': {'role': 'assistant',
'content': "```\nSELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-0
1'\nGROUP BY InvoiceId;\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 109002217118, 'load_
duration': 615768, 'prompt_eval_count': 1615, 'prompt_eval_duration': 102351183000, 'eval_count': 36, 'eval
_duration': 6028390000}
```
SELECT InvoiceId, SUM(Total) AS TotalAmount
FROM Invoice
WHERE InvoiceDate >= '2010-01-01'
GROUP BY InvoiceId;
```
Output from LLM: ```
SELECT InvoiceId, SUM(Total) AS TotalAmount
FROM Invoice
WHERE InvoiceDate >= '2010-01-01'
GROUP BY InvoiceId;
```
Extracted SQL: SELECT InvoiceId, SUM(Total) AS TotalAmount
FROM Invoice
WHERE InvoiceDate >= '2010-01-01'
GROUP BY InvoiceId
SELECT InvoiceId, SUM(Total) AS TotalAmount
FROM Invoice
WHERE InvoiceDate >= '2010-01-01'
GROUP BY InvoiceId
     InvoiceId TotalAmount
0   84   1.98
1   85   1.98
2   86   3.96
3   87   6.94
4   88   17.91
..   ..
324   408   3.96
325   409   5.94
326   410   8.91
327   411   13.86
328   412   1.99

[329 rows x 2 columns]
Ollama parameters:
```
```

```
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Find all invoices since 2010 and the total amount invoic
ed:\n'\n\nThe DataFrame was produced using this query: SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM In
voice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId\n\nThe following is information about the resu
lting pandas DataFrame 'df': \nRunning df.dtypes gives:\n InvoiceId        int64\nTotalAmount    float64\nd
type: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results o
f the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the da
taframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the c
ode."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:33:19.680442946Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = px.bar(df, x='Invo
iceId', y='TotalAmount', title='Invoices and Total Amount')\nfig.show()\n```"}, 'done_reason': 'stop', 'don
e': True, 'total_duration': 19346890253, 'load_duration': 960815, 'prompt_eval_count': 197, 'prompt_eval_du
ration': 11763856000, 'eval_count': 47, 'eval_duration': 7440224000}
```


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

Out[19]:
```
("SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY I
nvoiceId",
      InvoiceId  TotalAmount
 0           84         1.98
 1           85         1.98
 2           86         3.96
 3           87         6.94
 4           88        17.91
 ..         ...          ...
 324        408         3.96
 325        409         5.94
 326        410         8.91
 327        411        13.86
 328        412         1.99

 [329 rows x 2 columns],
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'InvoiceId=%{x}<br>TotalAmount=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([ 84,  85,  86, ..., 410, 411, 412]),
               'xaxis': 'x',
               'y': array([ 1.98,  1.98,  3.96, ...,  8.91, 13.86,  1.99]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Invoices and Total Amount'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalAmount'}}}
 }))
```

In [20]:
```
question = """
    List all employees and their reporting manager's name (if any):
"""
```

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

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

[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\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   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 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(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) 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\n\n   CREATE TABLE IF NOT EXISTS t_person (\n   id INT PRIMARY KEY,\n   name VARCHAR(100),\n   email text,\n   age INT\n   )\n\n\nCREATE INDEX IFK_CustomerSupportRepId 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   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 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   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 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 (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 INTEGER NOT NULL,\n   Name NVARCHAR(120),\n   CONSTRAINT PK_Artist PRIMARY KEY (ArtistId)\n)\n\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS TotalCusto

mers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': 'What person has a name of "John Doe"?'}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM Invoice \nWHERE Total > 10.00\n'}, {'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 name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': " \n   List all employees and their reporting manager's name (if any):\n"}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\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   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 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(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) 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\n\n   CREATE TABLE IF NOT EXISTS t_person (\n   id INT PRIMARY KEY,\n   name VARCHAR(100),\n   email text,\n   age INT\n   )\n\n\nCREATE INDEX IFK_CustomerSupportRepId 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   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 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   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 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 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 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\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders th
at are delivered on time and in full\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": "what are the top 5 coun
tries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS TotalCusto
mers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content":
" \n   Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT C
ustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId"}, {"role": "user", "content": "
\n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SE
LECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY Invoic
eId"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"role": "a
ssistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice
i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC"}, {"role": "user", "cont
ent": " \n   List all albums and their corresponding artist names \n"}, {"role": "assistant", "content":
"SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId"}, {"role":
"user", "content": "What person has a name of \"John Doe\"?"}, {"role": "assistant", "content": "SELECT * F
ROM t_person WHERE name = 'John Doe'"}, {"role": "user", "content": " \n   List all invoices with a total
exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM Invoice \nWHERE Total > 10.00\n"},
{"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 * \nFROM Track \nWHERE LOW
ER(Name) LIKE '%what%'"}, {"role": "user", "content": "Show me a list of tables in the SQLite database"},
{"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "co

```
ntent": "  \n    List all employees and their reporting manager's name (if any):\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:35:08.744020446Z', 'message': {'role': 'assistant',
'content': '```\nSELECT e.LastName, e.FirstName, m.LastName, m.FirstName\nFROM Employee e\nLEFT JOIN Employ
ee m ON e.ReportsTo = m.EmployeeId;\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 10891756
6303, 'load_duration': 657470, 'prompt_eval_count': 1615, 'prompt_eval_duration': 102338556000, 'eval_coun
t': 35, 'eval_duration': 5848746000}
```
SELECT e.LastName, e.FirstName, m.LastName, m.FirstName
FROM Employee e
LEFT JOIN Employee m ON e.ReportsTo = m.EmployeeId;
```
Output from LLM: ```
SELECT e.LastName, e.FirstName, m.LastName, m.FirstName
FROM Employee e
LEFT JOIN Employee m ON e.ReportsTo = m.EmployeeId;
```
Extracted SQL: SELECT e.LastName, e.FirstName, m.LastName, m.FirstName
FROM Employee e
LEFT JOIN Employee m ON e.ReportsTo = m.EmployeeId
SELECT e.LastName, e.FirstName, m.LastName, m.FirstName
FROM Employee e
LEFT JOIN Employee m ON e.ReportsTo = m.EmployeeId
   LastName FirstName LastName FirstName
0   Adams   Andrew   None   None
1   Edwards   Nancy   Adams   Andrew
2   Peacock   Jane   Edwards   Nancy
3   Park Margaret   Edwards   Nancy
4   Johnson   Steve   Edwards   Nancy
5 Mitchell   Michael   Adams   Andrew
6   King   Robert Mitchell   Michael
7 Callahan   Laura Mitchell   Michael
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: ' \n   List all employees and their reporting manager's name (i
f any):\n'\n\nThe DataFrame was produced using this query: SELECT e.LastName, e.FirstName, m.LastName, m.Fi
rstName\nFROM Employee e\nLEFT JOIN Employee m ON e.ReportsTo = m.EmployeeId\n\nThe following is informatio
n about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n LastName   object\nFirstName
```
```

```
object\nLastName     object\nFirstName    object\ndtype: object"}, {"role": "user", "content": "Can you gen
erate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas datafram
e called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python cod
e. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:35:42.373386374Z', 'message': {'role': 'assistant',
'content': '```\nimport plotly.express as px\nimport numpy as np\n\nfig = px.bar(df, x=\'LastName\', y=\'Fi
rstName\', color=\'LastName\')\nfig.update_layout(title=\'Employee Reporting Managers\', xaxis_title=\'Last
Name\', yaxis_title=\'First Name\')\n\nif df.shape[0] == 1:\n    fig = px.scatter([df.iloc[0]], x=[\'Report
ing Manager\'], y=[f\'{df.iloc[0]["FirstName"]} {df.iloc[0]["LastName"]}\'], color_discrete_sequence=[\'blu
e\'])\n    fig.update_layout(title=\'Single Employee Reporting Manager\', xaxis_title=\'Employee ID\')\n\nf
ig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 33468582371, 'load_duration': 7300
15, 'prompt_eval_count': 204, 'prompt_eval_duration': 12099752000, 'eval_count': 132, 'eval_duration': 2122
5257000}
Couldn't run plotly code:  The truth value of a Series is ambiguous. Use a.empty, a.bool(), a.item(), a.any
() or a.all().
```

```
Traceback (most recent call last):
  File "/home/papagame/anaconda3/envs/vanna/lib/python3.11/site-packages/vanna/base/base.py", line 1998, in
get_plotly_figure
    exec(plotly_code, globals(), ldict)
  File "<string>", line 4, in <module>
  File "/home/papagame/anaconda3/envs/vanna/lib/python3.11/site-packages/plotly/express/_chart_types.py", l
ine 373, in bar
    return make_figure(
           ^^^^^^^^^^^^
  File "/home/papagame/anaconda3/envs/vanna/lib/python3.11/site-packages/plotly/express/_core.py", line 209
0, in make_figure
    args = build_dataframe(args, constructor)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/papagame/anaconda3/envs/vanna/lib/python3.11/site-packages/plotly/express/_core.py", line 149
2, in build_dataframe
    df_output, wide_id_vars = process_args_into_dataframe(
                              ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/papagame/anaconda3/envs/vanna/lib/python3.11/site-packages/plotly/express/_core.py", line 122
8, in process_args_into_dataframe
    df_output[col_name] = to_unindexed_series(
                          ^^^^^^^^^^^^^^^^^^^^
  File "/home/papagame/anaconda3/envs/vanna/lib/python3.11/site-packages/plotly/express/_core.py", line 107
6, in to_unindexed_series
    return pd.Series(x, name=name).reset_index(drop=True)
           ^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/papagame/anaconda3/envs/vanna/lib/python3.11/site-packages/pandas/core/series.py", line 584,
in __init__
    data = sanitize_array(data, index, dtype, copy)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/papagame/anaconda3/envs/vanna/lib/python3.11/site-packages/pandas/core/construction.py", line
633, in sanitize_array
    return sanitize_array(
           ^^^^^^^^^^^^^^
  File "/home/papagame/anaconda3/envs/vanna/lib/python3.11/site-packages/pandas/core/construction.py", line
606, in sanitize_array
    subarr = maybe_infer_to_datetimelike(data)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/papagame/anaconda3/envs/vanna/lib/python3.11/site-packages/pandas/core/dtypes/cast.py", line
1182, in maybe_infer_to_datetimelike
    raise ValueError(value.ndim)  # pragma: no cover
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
ValueError: 2
```

```
During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/home/papagame/anaconda3/envs/vanna/lib/python3.11/site-packages/vanna/base/base.py", line 1674, in
ask
    fig = self.get_plotly_figure(plotly_code=plotly_code, df=df)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/papagame/anaconda3/envs/vanna/lib/python3.11/site-packages/vanna/base/base.py", line 2015, in
get_plotly_figure
    elif len(categorical_cols) >= 1 and df[categorical_cols[0]].nunique() < 10:
                                        ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/papagame/anaconda3/envs/vanna/lib/python3.11/site-packages/pandas/core/generic.py", line 157
7, in __nonzero__
    raise ValueError(
ValueError: The truth value of a Series is ambiguous. Use a.empty, a.bool(), a.item(), a.any() or a.all().
```

In [21]:
```
question = """
    Get the average invoice total for each customer:
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE 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 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   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 INDEX IFK_InvoiceLineTrackId 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   LastName NVARCHAR(20) NOT NULL,\n   Company NVARCHAR(80),\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) 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\n\n   CREATE TABLE IF NOT EXISTS t_person (\n   id INT PRIMARY KEY,\n   name VARCHAR(100),\n   email text,\n   age INT\n   )\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 (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\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC'}, {'role':

'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'conte
nt': 'SELECT * \nFROM Invoice \nWHERE Total > 10.00\n'}, {'role': 'user', 'content': 'what are the top 5 co
untries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS TotalCus
tomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers 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 employees and their reporting manager's name (i
f any):\n"}, {'role': 'assistant', 'content': 'SELECT e.LastName, e.FirstName, m.LastName, m.FirstName\nFRO
M Employee e\nLEFT JOIN Employee m ON e.ReportsTo = m.EmployeeId'}, {'role': 'user', 'content': ' \n   Fi
nd all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELEC
T * \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, a.ArtistId, a
r.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': 'What perso
n has a name of "John Doe"?'}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John
Doe'"}, {'role': 'user', 'content': ' \n   Get the average invoice total for each customer:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and 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\n\n   CREATE T
ABLE IF NOT EXISTS t_person (\n   id INT PRIMARY KEY,\n   name VARCHAR(100),\n   email tex
t,\n   age INT\n   )\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

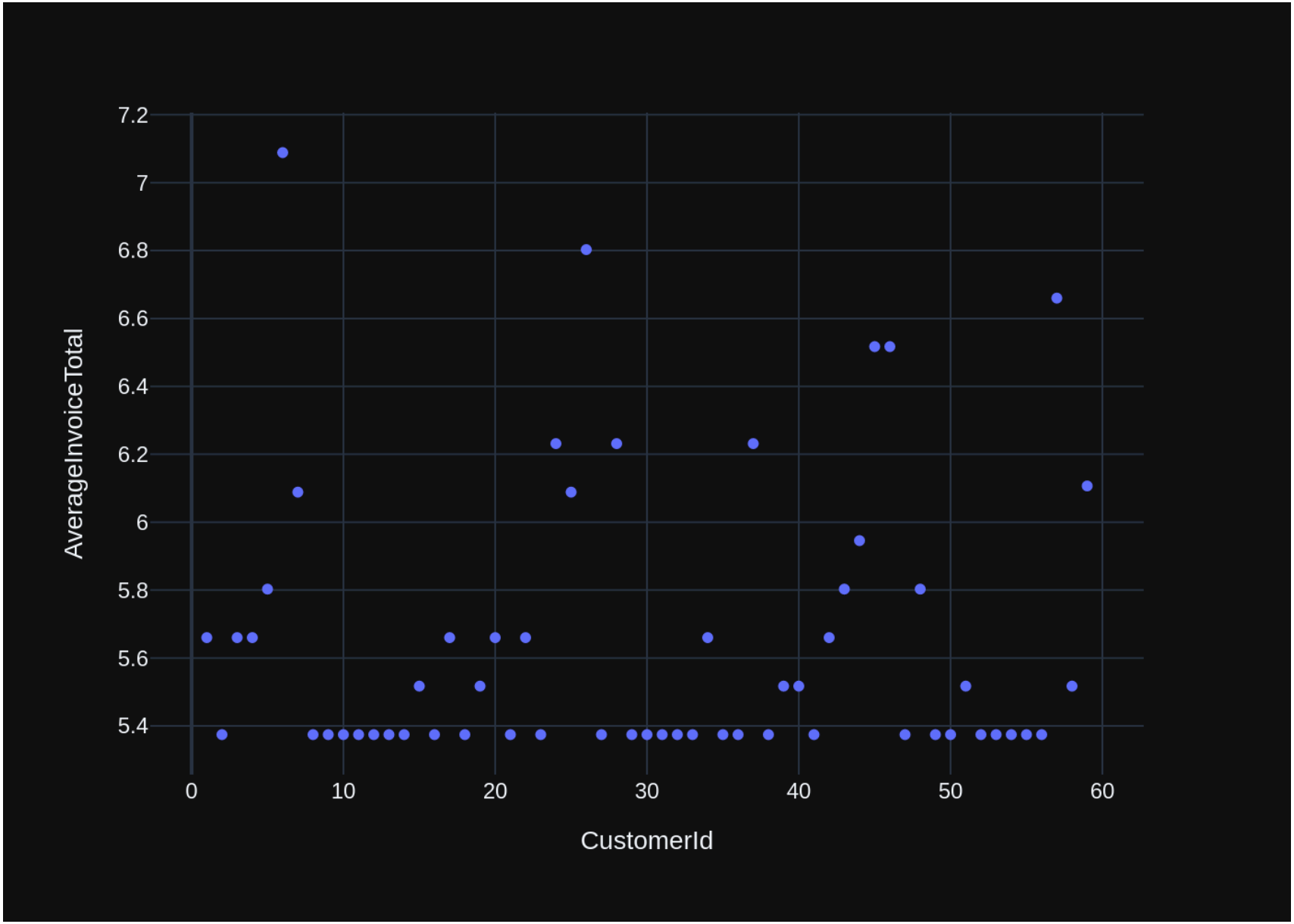
```
NVARCHAR(220),\n    Milliseconds INTEGER  NOT NULL,\n    Bytes INTEGER,\n    UnitPrice NUMERIC(10,2)  NOT N
ULL,\n    CONSTRAINT PK_Track PRIMARY KEY  (TrackId),\n    FOREIGN KEY (AlbumId) REFERENCES Album (AlbumId)
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\n    FOREIGN KEY (GenreId) REFERENCES 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\n\n===Additional Context \n\nOur business defines OTIF s
core as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If t
he provided context is sufficient, please generate a valid SQL query without any explanations for the quest
ion. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a par
ticular column, please generate an intermediate SQL query to find the distinct strings in that column. Prep
end the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please e
xplain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has 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", "conten
t": "SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId"}, {"role": "user", "c
ontent": "  \n    Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "c
ontent": "SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGRO
UP BY InvoiceId"}, {"role": "user", "content": "  \n    Find the total number of invoices per country:\n"},
{"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJ
OIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC"}, {"role":
"user", "content": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "conte
nt": "SELECT * \nFROM Invoice \nWHERE Total > 10.00\n"}, {"role": "user", "content": "what are the top 5 co
untries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS TotalCus
tomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers 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 employees and their reporting manager's name (i
f any):\n"}, {"role": "assistant", "content": "SELECT e.LastName, e.FirstName, m.LastName, m.FirstName\nFRO
M Employee e\nLEFT JOIN Employee m ON e.ReportsTo = m.EmployeeId"}, {"role": "user", "content": "  \n    Fi
nd all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SEL
ECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": "  \n    List all album
s and their corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId,
ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "What pers
on has a name of \"John Doe\"?"}, {"role": "assistant", "content": "SELECT * FROM t_person WHERE name = 'Jo
hn Doe'"}, {"role": "user", "content": "  \n    Get the average invoice total for each customer:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:37:21.488655751Z', 'message': {'role': 'assistant',
'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM Customer c\nJOIN Invoice i ON c.
CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, 'done_reason': 'stop', 'done': True, 'total_duration':
99068303245, 'load_duration': 695880, 'prompt_eval_count': 1465, 'prompt_eval_duration': 92361394000, 'eval
_count': 36, 'eval_duration': 5982494000}
SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
```

```
GROUP BY c.CustomerId
SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
```

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

```
36          37             6.231429
37          38             5.374286
38          39             5.517143
39          40             5.517143
40          41             5.374286
41          42             5.660000
42          43             5.802857
43          44             5.945714
44          45             6.517143
45          46             6.517143
46          47             5.374286
47          48             5.802857
48          49             5.374286
49          50             5.374286
50          51             5.517143
51          52             5.374286
52          53             5.374286
53          54             5.374286
54          55             5.374286
55          56             5.374286
56          57             6.660000
57          58             5.517143
58          59             6.106667
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Get the average invoice total for each customer:\n'\n\nT
he DataFrame was produced using this query: SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM
Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\n\nThe following is inform
ation about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId               int6
4\nAverageInvoiceTotal    float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Pytho
n plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'.
If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer
with any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:37:42.744364273Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = px.bar(df, x='Cust
omerId', y='AverageInvoiceTotal')\n\nif len(df) == 1:\n    fig.update_layout(title='Indicator')\nelse:\n
fig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 21090435783, 'load_duration': 281
```

1705, 'prompt_eval_count': 197, 'prompt_eval_duration': 11738172000, 'eval_count': 58, 'eval_duration': 9201811000}



Out[21]: ('SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId',

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

```
39          40             5.517143
40          41             5.374286
41          42             5.660000
42          43             5.802857
43          44             5.945714
44          45             6.517143
45          46             6.517143
46          47             5.374286
47          48             5.802857
48          49             5.374286
49          50             5.374286
50          51             5.517143
51          52             5.374286
52          53             5.374286
53          54             5.374286
54          55             5.374286
55          56             5.374286
56          57             6.660000
57          58             5.517143
58          59             6.106667,
Figure({
    'data': [{'hovertemplate': 'CustomerId=%{x}<br>AverageInvoiceTotal=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'symbol': 'circle'},
              'mode': 'markers',
              'name': '',
              'orientation': 'v',
              'showlegend': False,
              'type': 'scatter',
              'x': array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
                          19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
                          37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
                          55, 56, 57, 58, 59]),
              'xaxis': 'x',
              'y': array([5.66      , 5.37428571, 5.66      , 5.66      , 5.80285714, 7.08857143,
                          6.08857143, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571,
                          5.37428571, 5.37428571, 5.51714286, 5.37428571, 5.66      , 5.37428571,
                          5.51714286, 5.66      , 5.37428571, 5.66      , 5.37428571, 6.23142857,
                          6.08857143, 6.80285714, 5.37428571, 6.23142857, 5.37428571, 5.37428571,
                          5.37428571, 5.37428571, 5.37428571, 5.66      , 5.37428571, 5.37428571,
                          6.23142857, 5.37428571, 5.51714286, 5.51714286, 5.37428571, 5.66      ,
                          5.80285714, 5.94571429, 6.51714286, 6.51714286, 5.37428571, 5.80285714,
```

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

In [22]:
```
question = """
    Find the top 5 most expensive tracks (based on unit price):
"""

vn.ask(question=question)
```

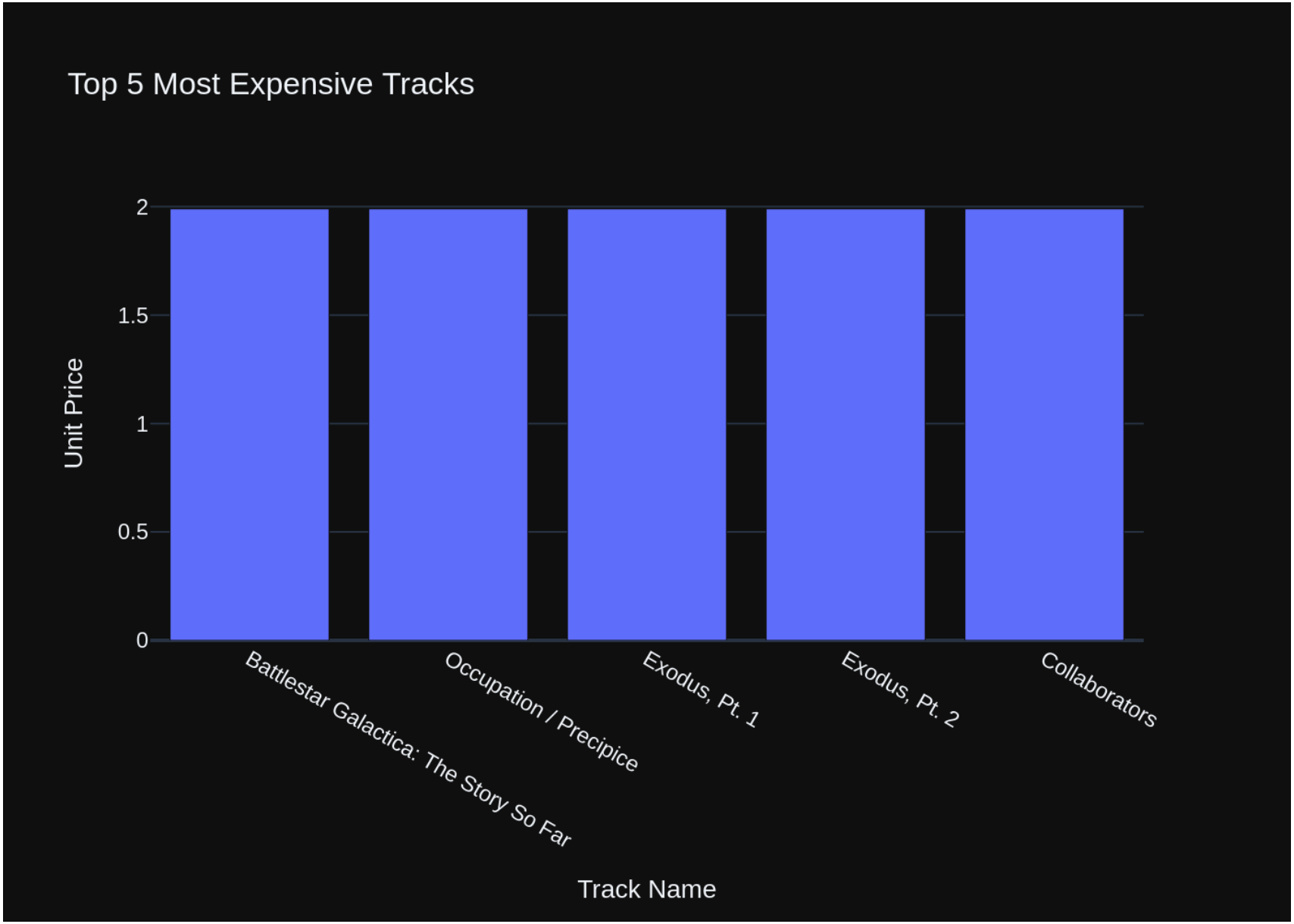
```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE 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 (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_TrackGenreId ON Track (GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON PlaylistTrack (TrackId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON InvoiceLine (TrackId)\n\nCREATE INDEX IFK_TrackMediaTypeId 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   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 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 (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 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\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM Invoice \nWHERE Total > 10.00\n'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM 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 InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\n

```
WHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {'role': 'user', 'content': '  \n    Find the tota
l number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceI
d) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nOR
DER BY TotalInvoices DESC'}, {'role': 'user', 'content': '  \n    Get the total number of invoices for each
customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice
\nGROUP BY CustomerId'}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'},
{'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'co
ntent': 'How many records are in table called customer'}, {'role': 'assistant', 'content': 'SELECT COUNT(*)
FROM Customer'}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive tracks (based on unit p
rice):\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE 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\n===Additional Context \n\nOur business defines OTIF score a
s the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the pro
vided context is sufficient, please generate a valid SQL query without any explanations for the question.
\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particul
ar column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend t
```

he query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explai
n why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been aske
d and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "con
tent": " \n   Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistan
t", "content": "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": "
\n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM
Invoice \nWHERE Total > 10.00\n"}, {"role": "user", "content": " \n   List all albums and their correspon
ding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name\nFROM Album
a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "what are the top 5 countries
that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS TotalCustomers\n
FROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n
Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerI
d, AVG(i.Total) AS AverageInvoiceTotal\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGRO
UP BY c.CustomerId"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amoun
t invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoic
e\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {"role": "user", "content": " \n   Find the t
otal number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.Invoi
ceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country
\nORDER BY TotalInvoices DESC"}, {"role": "user", "content": " \n   Get the total number of invoices for
each customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Inv
oice\nGROUP BY CustomerId"}, {"role": "user", "content": "Show me a list of tables in the SQLite databas
e"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "use
r", "content": "How many records are in table called customer"}, {"role": "assistant", "content": "SELECT C
OUNT(*) FROM Customer"}, {"role": "user", "content": " \n   Find the top 5 most expensive tracks (based o
n unit price):\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:39:11.178476337Z', 'message': {'role': 'assistant',
'content': '```\nSELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLIMIT 5;\n
```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 88318172662, 'load_duration': 632597, 'prompt
_eval_count': 1318, 'prompt_eval_duration': 82436796000, 'eval_count': 31, 'eval_duration': 5153066000}
```
SELECT t.TrackId, t.Name, t.UnitPrice
FROM Track t
ORDER BY t.UnitPrice DESC
LIMIT 5;
```
Output from LLM: ```
SELECT t.TrackId, t.Name, t.UnitPrice
FROM Track t
ORDER BY t.UnitPrice DESC
LIMIT 5;
```

```
Extracted SQL: SELECT t.TrackId, t.Name, t.UnitPrice
FROM Track t
ORDER BY t.UnitPrice DESC
LIMIT 5
SELECT t.TrackId, t.Name, t.UnitPrice
FROM Track t
ORDER BY t.UnitPrice DESC
LIMIT 5
   TrackId                                  Name  UnitPrice
0     2819  Battlestar Galactica: The Story So Far       1.99
1     2820                  Occupation / Precipice       1.99
2     2821                           Exodus, Pt. 1       1.99
3     2822                           Exodus, Pt. 2       1.99
4     2823                           Collaborators       1.99
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Find the top 5 most expensive tracks (based on unit pric
e):\n'\n\nThe DataFrame was produced using this query: SELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t
\nORDER BY t.UnitPrice DESC\nLIMIT 5\n\nThe following is information about the resulting pandas DataFrame
'df': \nRunning df.dtypes gives:\n TrackId        int64\nName          object\nUnitPrice    float64\ndtype:
object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the
dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the datafram
e, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the cod
e."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:39:34.55373884Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = go.Figure(data=[g
o.Bar(x=df['Name'], y=df['UnitPrice'])])\nfig.update_layout(title='Top 5 Most Expensive Tracks', xaxis_titl
e='Track Name', yaxis_title='Unit Price')\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_d
uration': 23254404712, 'load_duration': 776161, 'prompt_eval_count': 198, 'prompt_eval_duration': 117523450
00, 'eval_count': 71, 'eval_duration': 11353074000}
```


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

Out[22]:
```
('SELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLIMIT 5',
   TrackId                                  Name  UnitPrice
0     2819  Battlestar Galactica: The Story So Far       1.99
1     2820                 Occupation / Precipice       1.99
2     2821                          Exodus, Pt. 1       1.99
3     2822                          Exodus, Pt. 2       1.99
4     2823                          Collaborators       1.99,
Figure({
    'data': [{'type': 'bar',
              'x': array(['Battlestar Galactica: The Story So Far', 'Occupation / Precipice',
                          'Exodus, Pt. 1', 'Exodus, Pt. 2', 'Collaborators'], dtype=object),
              'y': array([1.99, 1.99, 1.99, 1.99, 1.99])}],
    'layout': {'template': '...',
               'title': {'text': 'Top 5 Most Expensive Tracks'},
               'xaxis': {'title': {'text': 'Track Name'}},
               'yaxis': {'title': {'text': 'Unit Price'}}}
})
```

In [23]:
```
question = """
    List all genres and the number of tracks in each genre:
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE 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 (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 TABLE Genre\n(\n   GenreId INTEGER NOT NULL,\n   Name NVARCHAR(120),\n   CONSTRAINT PK_Genre 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\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 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 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 Playlist\n(\n   PlaylistId INTEGER NOT NULL,\n   Name NVARCHAR(120),\n   CONSTRAINT PK_Playlist PRIMARY KEY (PlaylistId)\n)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqli

te_master WHERE type='table'"}, {'role': 'user', 'content': ' \n   Get the total number of invoices for e
ach customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invo
ice\nGROUP BY CustomerId'}, {'role': 'user', 'content': 'How many records are in table called customer'},
{'role': 'assistant', 'content': 'SELECT COUNT(*) FROM Customer'}, {'role': 'user', 'content': ' \n   Get
the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AV
G(i.Total) AS AverageInvoiceTotal\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY
c.CustomerId'}, {'role': 'user', 'content': ' \n   List all genres and the number of tracks in each genr
e:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE 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
\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered
on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a
valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficien
t but requires knowledge of a specific string in a particular column, please generate an intermediate SQL q
uery to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql
\n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the
most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer e
xactly as it was given before. \n"}, {"role": "user", "content": " \n   List all albums and their corresp
onding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name\nFROM Albu

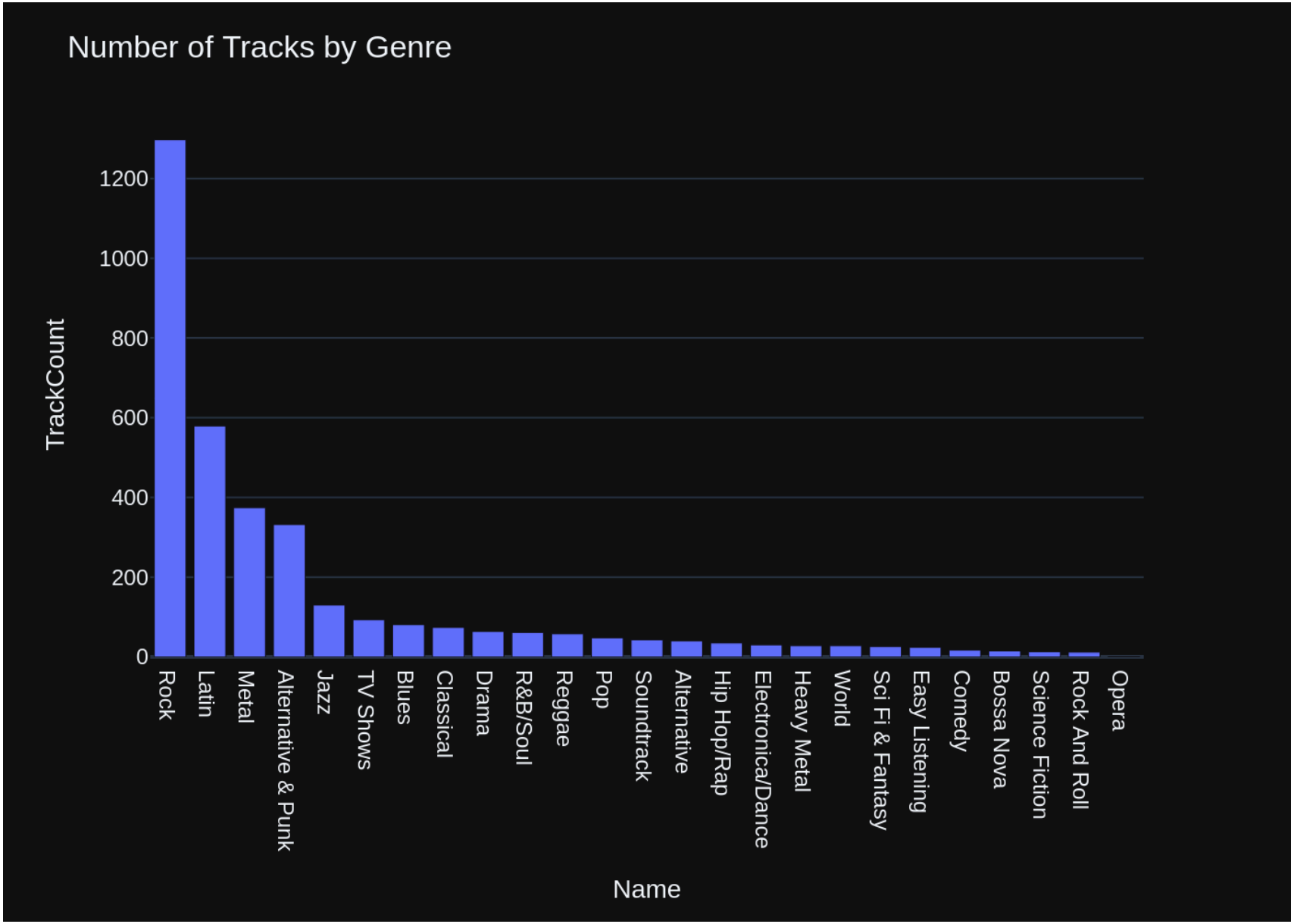
m a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT *\nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {"role": "user", "content": "Show me a list of tables in the SQLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "content": " \n   Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId"}, {"role": "user", "content": "How many records are in table called customer"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM Customer"}, {"role": "user", "content": " \n   Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   List all genres and the number of tracks in each genre:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:41:03.279938765Z', 'message': {'role': 'assistant', 'content': '```\nSELECT g.GenreId, g.Name, COUNT(t.TrackId) AS TrackCount\nFROM Genre g\nJOIN Track t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId, g.Name\nORDER BY TrackCount DESC;\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 88576605082, 'load_duration': 701773, 'prompt_eval_count': 1259, 'prompt_eval_duration': 78521738000, 'eval_count': 56, 'eval_duration': 9329560000}
```
SELECT g.GenreId, g.Name, COUNT(t.TrackId) AS TrackCount
FROM Genre g
JOIN Track t ON g.GenreId = t.GenreId
GROUP BY g.GenreId, g.Name
ORDER BY TrackCount DESC;
```
Output from LLM: ```
SELECT g.GenreId, g.Name, COUNT(t.TrackId) AS TrackCount
FROM Genre g
JOIN Track t ON g.GenreId = t.GenreId
GROUP BY g.GenreId, g.Name
ORDER BY TrackCount DESC;
```
Extracted SQL: SELECT g.GenreId, g.Name, COUNT(t.TrackId) AS TrackCount
```

```
FROM Genre g
JOIN Track t ON g.GenreId = t.GenreId
GROUP BY g.GenreId, g.Name
ORDER BY TrackCount DESC
SELECT g.GenreId, g.Name, COUNT(t.TrackId) AS TrackCount
FROM Genre g
JOIN Track t ON g.GenreId = t.GenreId
GROUP BY g.GenreId, g.Name
ORDER BY TrackCount DESC
```

	GenreId	Name	TrackCount
0	1	Rock	1297
1	7	Latin	579
2	3	Metal	374
3	4	Alternative & Punk	332
4	2	Jazz	130
5	19	TV Shows	93
6	6	Blues	81
7	24	Classical	74
8	21	Drama	64
9	14	R&B/Soul	61
10	8	Reggae	58
11	9	Pop	48
12	10	Soundtrack	43
13	23	Alternative	40
14	17	Hip Hop/Rap	35
15	15	Electronica/Dance	30
16	13	Heavy Metal	28
17	16	World	28
18	20	Sci Fi & Fantasy	26
19	12	Easy Listening	24
20	22	Comedy	17
21	11	Bossa Nova	15
22	18	Science Fiction	13
23	5	Rock And Roll	12
24	25	Opera	1

```
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all genres and the number of tracks in each genr
```

e:\n'\n\nThe DataFrame was produced using this query: SELECT g.GenreId, g.Name, COUNT(t.TrackId) AS TrackCo
unt\nFROM Genre g\nJOIN Track t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId, g.Name\nORDER BY TrackCount D
ESC\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n
GenreId   int64\nName   object\nTrackCount   int64\ndtype: object"}, {"role": "user", "conten
t": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a
pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with o
nly Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:41:23.248493377Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nfig = px.bar(df, x='Name', y='TrackCount')\nfig.update_layout
(title='Number of Tracks by Genre')\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duratio
n': 19831465123, 'load_duration': 2836769, 'prompt_eval_count': 221, 'prompt_eval_duration': 13199716000,
'eval_count': 41, 'eval_duration': 6476447000}


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

Out[23]: ('SELECT g.GenreId, g.Name, COUNT(t.TrackId) AS TrackCount\nFROM Genre g\nJOIN Track t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId, g.Name\nORDER BY TrackCount DESC',

	GenreId	Name	TrackCount
0	1	Rock	1297
1	7	Latin	579
2	3	Metal	374
3	4	Alternative & Punk	332
4	2	Jazz	130
5	19	TV Shows	93
6	6	Blues	81
7	24	Classical	74
8	21	Drama	64
9	14	R&B/Soul	61
10	8	Reggae	58
11	9	Pop	48
12	10	Soundtrack	43
13	23	Alternative	40
14	17	Hip Hop/Rap	35
15	15	Electronica/Dance	30
16	13	Heavy Metal	28
17	16	World	28
18	20	Sci Fi & Fantasy	26
19	12	Easy Listening	24
20	22	Comedy	17
21	11	Bossa Nova	15
22	18	Science Fiction	13
23	5	Rock And Roll	12
24	25	Opera	1,

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

```
                            'Heavy Metal', 'World', 'Sci Fi & Fantasy', 'Easy Listening', 'Comedy',
                            'Bossa Nova', 'Science Fiction', 'Rock And Roll', 'Opera'], dtype=object),
                 'xaxis': 'x',
                 'y': array([1297,  579,  374,  332,  130,   93,   81,   74,   64,   61,   58,   48,
                              43,   40,   35,   30,   28,   28,   26,   24,   17,   15,   13,   12,
                               1]),
                 'yaxis': 'y'}],
      'layout': {'barmode': 'relative',
                 'legend': {'tracegroupgap': 0},
                 'margin': {'t': 60},
                 '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': 'TrackCount'}}}
 }))
```

In [24]:

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

vn.ask(question=question)
```

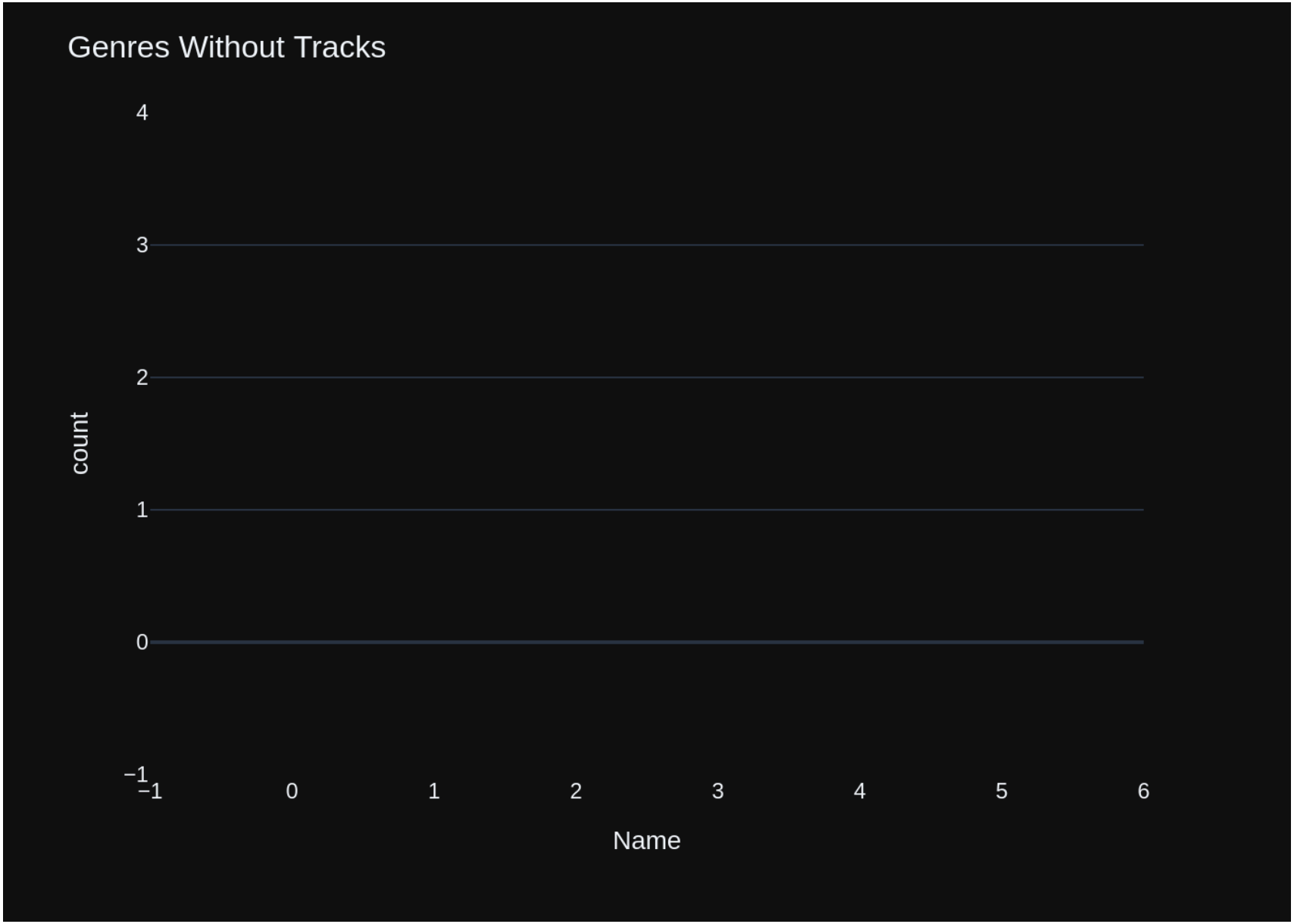
```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON 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 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 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   GenreId INTEGER NOT NULL,\n   Name NVARCHAR(120),\n   CONSTRAINT PK_Genre PRIMARY KEY (GenreId)\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_AlbumArtistId ON Album (ArtistId)\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 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 Artist\n(\n   ArtistId INTEGER NOT NULL,\n   Name NVARCHAR(120),\n   CONSTRAINT PK_Artist PRIMARY KEY (ArtistId)\n)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': ' \n   List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.GenreId, g.Name, COUNT(t.TrackId) AS TrackCount \nFROM Genre g\nJOIN Track t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId, g.Name\nORDER BY TrackCount DESC'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': 'What person has a name of "John Doe"?'}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find all invoices

```
since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceId, SUM(Tota
l) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {'role': 'user',
'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SE
LECT * \nFROM Invoice \nWHERE Total > 10.00\n'}, {'role': 'user', 'content': '  \n    Get the average invoi
ce total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS Ave
rageInvoiceTotal\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'},
{'role': 'user', 'content': '  \n    Get all genres that do not have any tracks associated with them:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON 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\n===Additio
nal Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and
in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL q
uery without any explanations for the question. \n2. If the provided context is almost sufficient but requi
res knowledge of a specific string in a particular column, please generate an intermediate SQL query to fin
d the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the
provided context is insufficient, please explain why it can't be generated. \n4. Please use the most releva
nt table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as i
t 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.GenreId, g.Name, COUNT(t.TrackId) AS TrackCount
\nFROM Genre g\nJOIN Track t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId, g.Name\nORDER BY TrackCount DES
```

C"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names \n"}, {"rol
e": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistI
d = ar.ArtistId"}, {"role": "user", "content": " \n   Find all tracks with a name containing \"What\" (ca
se-insensitive)\n"}, {"role": "assistant", "content": "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%wha
t%'"}, {"role": "user", "content": " \n   Find the top 5 most expensive tracks (based on unit pric
e):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.
UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "Show me a list of tables in the SQLite database"},
{"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "co
ntent": "What person has a name of \"John Doe\"?"}, {"role": "assistant", "content": "SELECT * FROM t_perso
n WHERE name = 'John Doe'"}, {"role": "user", "content": "what are the top 5 countries that customers come
from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP
BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find all invoices
since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceId, SUM(Tota
l) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {"role": "user",
"content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SE
LECT * \nFROM Invoice \nWHERE Total > 10.00\n"}, {"role": "user", "content": " \n   Get the average invoi
ce total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS Ave
rageInvoiceTotal\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"},
{"role": "user", "content": " \n   Get all genres that do not have any tracks associated with them:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:42:51.42585316Z', 'message': {'role': 'assistant',
'content': 'SELECT g.GenreId, g.Name\nFROM Genre g\nWHERE NOT EXISTS (\n   SELECT 1 \n   FROM Track t \n
WHERE t.GenreId = g.GenreId\n)'}, 'done_reason': 'stop', 'done': True, 'total_duration': 88027651618, 'load
_duration': 653931, 'prompt_eval_count': 1289, 'prompt_eval_duration': 80490421000, 'eval_count': 41, 'eval
_duration': 6806805000}
```
SELECT g.GenreId, g.Name
FROM Genre g
WHERE NOT EXISTS (
    SELECT 1
    FROM Track t
    WHERE t.GenreId = g.GenreId
)
SELECT g.GenreId, g.Name
FROM Genre g
WHERE NOT EXISTS (
    SELECT 1
    FROM Track t
    WHERE t.GenreId = g.GenreId
)
Empty DataFrame
Columns: [GenreId, Name]
Index: []
```

```
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Get all genres that do not have any tracks associated wi
th them:\n'\n\nThe DataFrame was produced using this query: SELECT g.GenreId, g.Name\nFROM Genre g\nWHERE N
OT EXISTS (\n    SELECT 1 \n    FROM Track t \n    WHERE t.GenreId = g.GenreId\n)\n\nThe following is infor
mation about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n GenreId    object\nName
object\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the
results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value
in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- j
ust the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:43:10.953697325Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = px.bar(df, x='Nam
e', y=None)\nfig.update_layout(title='Genres Without Tracks')\nfig.show()\n```"}, 'done_reason': 'stop', 'd
one': True, 'total_duration': 19525705792, 'load_duration': 661275, 'prompt_eval_count': 203, 'prompt_eval_
duration': 12117278000, 'eval_count': 46, 'eval_duration': 7271711000}
```


Genres Without Tracks
4
3
2
1
0
−1
count
−1
0
1
2
3
4
5
6
Name

Out[24]:
```
('SELECT g.GenreId, g.Name\nFROM Genre g\nWHERE NOT EXISTS (\n    SELECT 1 \n    FROM Track t \n    WHERE
t.GenreId = g.GenreId\n)',
 Empty DataFrame
 Columns: [GenreId, Name]
 Index: [],
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'Name=%{x}<br>count=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([], dtype=object),
               'xaxis': 'x',
               'y': array([], dtype=int64),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'title': {'text': 'Genres Without Tracks'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'count'}}}
 }))
```

In [25]:
```
question = """
    List all customers who have not placed any orders:
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE 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(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) 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 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 UPDATE NO ACTION\n)\n\n\n   CREATE TABLE IF NOT EXISTS t_person (\n   id INT PRIMARY KEY,\n   name VARCHAR(100),\n   email text,\n   age INT\n   )\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 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   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 INDEX IFK_CustomerSupportRepId 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   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 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 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 (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\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\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': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'conten
t': 'SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers D
ESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Get the total number of invoices for each customer
\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP B
Y CustomerId'}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'ass
istant', 'content': 'SELECT COUNT(*) FROM Customer'}, {'role': 'user', 'content': '  \n    Find the total n
umber of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId)
AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER
BY TotalInvoices DESC'}, {'role': 'user', 'content': '  \n    Get the average invoice total for each custom
er:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM C
ustomer 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 *
\nFROM Invoice \nWHERE Total > 10.00\n'}, {'role': 'user', 'content': '  \n    Find all invoices since 2010
and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceId, SUM(Total) AS Total
Amount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {'role': 'user', 'content':
"  \n    List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'conten
t': 'SELECT e.LastName, e.FirstName, m.LastName, m.FirstName\nFROM Employee e\nLEFT JOIN Employee m ON e.Re
portsTo = m.EmployeeId'}, {'role': 'user', 'content': '  \n    List all albums and their corresponding arti
st names  \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN A
rtist ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': 'What person has a name of "John Do
e"?'}, {'role': 'assistant', 'content': "SELECT * FROM t_person WHERE name = 'John Doe'"}, {'role': 'user',
'content': '  \n    List all customers who have not placed any orders:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE 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\n\n   CREATE TABLE IF NOT EXISTS t_person (\n   id INT PRIM
ARY KEY,\n   name VARCHAR(100),\n   email text,\n   age INT\n   )\n\n\nCREATE TABLE Invoice
Line\n(\n   InvoiceLineId INTEGER NOT NULL,\n   InvoiceId INTEGER NOT NULL,\n   TrackId INTEGER NOT N
ULL,\n   UnitPrice NUMERIC(10,2) NOT NULL,\n   Quantity INTEGER NOT NULL,\n   CONSTRAINT PK_InvoiceLin
e 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 ACT
ION ON UPDATE NO ACTION\n)\n\nCREATE TABLE Employee\n(\n   EmployeeId INTEGER NOT NULL,\n   LastName NVA
RCHAR(20) NOT NULL,\n   FirstName NVARCHAR(20) NOT NULL,\n   Title NVARCHAR(30),\n   ReportsTo INTEGE
R,\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(60),\n   CONSTRAINT PK_Employee PRIMARY KEY (EmployeeId),\n   FORE
IGN KEY (ReportsTo) REFERENCES Employee (EmployeeId) \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\n)\n\nCR
EATE INDEX IFK_CustomerSupportRepId ON Customer (SupportRepId)\n\nCREATE TABLE PlaylistTrack\n(\n   Playli
stId INTEGER NOT NULL,\n   TrackId INTEGER NOT NULL,\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (Play
listId, 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 UPD
ATE NO ACTION\n)\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
oiceCustomerId ON Invoice (CustomerId)\n\nCREATE TABLE Track\n(\n   TrackId INTEGER NOT NULL,\n   Name N
VARCHAR(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\n\n===Additional Context \n\nOur business defi
nes OTIF score as the percentage of orders that are delivered on time and in full\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": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "conten
t": "SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers D
ESC\nLIMIT 5"}, {"role": "user", "content": " \n   Get the total number of invoices for each customer
\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP B
Y CustomerId"}, {"role": "user", "content": "How many records are in table called customer"}, {"role": "ass
istant", "content": "SELECT COUNT(*) FROM Customer"}, {"role": "user", "content": " \n   Find the total n
umber of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId)
AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER
BY TotalInvoices DESC"}, {"role": "user", "content": " \n   Get the average invoice total for each custom

```
er:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM C
ustomer 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 *
\nFROM Invoice \nWHERE Total > 10.00\n"}, {"role": "user", "content": "  \n    Find all invoices since 2010
and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceId, SUM(Total) AS Total
Amount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {"role": "user", "content":
"  \n    List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "conten
t": "SELECT e.LastName, e.FirstName, m.LastName, m.FirstName\nFROM Employee e\nLEFT JOIN Employee m ON e.Re
portsTo = m.EmployeeId"}, {"role": "user", "content": "  \n    List all albums and their corresponding arti
st names  \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN A
rtist ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "What person has a name of \"John Doe
\"?"}, {"role": "assistant", "content": "SELECT * FROM t_person WHERE name = 'John Doe'"}, {"role": "user",
"content": "  \n    List all customers who have not placed any orders:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:45:12.194962401Z', 'message': {'role': 'assistant',
'content': "Note that the provided tables do not contain information about orders, so it's not possible to
list all customers who have not placed any orders. The question would need additional context or tables to
answer accurately."}, 'done_reason': 'stop', 'done': True, 'total_duration': 121053197706, 'load_duration':
653783, 'prompt_eval_count': 1775, 'prompt_eval_duration': 113368610000, 'eval_count': 41, 'eval_duration':
6969541000}
Note that the provided tables do not contain information about orders, so it's not possible to list all cus
tomers who have not placed any orders. The question would need additional context or tables to answer accur
ately.
Note that the provided tables do not contain information about orders, so it's not possible to list all cus
tomers who have not placed any orders. The question would need additional context or tables to answer accur
ately.
Couldn't run sql:  Execution failed on sql 'Note that the provided tables do not contain information about
orders, so it's not possible to list all customers who have not placed any orders. The question would need
additional context or tables to answer accurately.': near "Note": syntax error
```

In [26]:
```
question = """
    Get the top 10 most popular artists (based on the number of tracks):
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE 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 (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_AlbumArtistId ON Album (ArtistId)\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 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_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 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\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.GenreId, g.Name, COUNT(t.TrackId) AS TrackCount\nFROM Genre g\nJOIN Track t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId, g.Name\nORDER BY TrackCount DESC'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': ' \n   Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM Customer 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': 'as

sistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice
i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC'}, {'role': 'user', 'cont
ent': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT
* \nFROM Invoice \nWHERE Total > 10.00\n'}, {'role': 'user', 'content': 'How many records are in table call
ed customer'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM Customer'}, {'role': 'user', 'conten
t': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM
sqlite_master WHERE type='table'"}, {'role': 'user', 'content': ' \n   Get the top 10 most popular artist
s (based on the number of tracks):\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and 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\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are del
ivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please gen
erate a valid SQL query without any explanations for the question. \n2. If the provided context is almost s
ufficient but requires knowledge of a specific string in a particular column, please generate an intermedia
te SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermedi
ate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please
use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the
answer exactly as it was given before. \n"}, {"role": "user", "content": " \n   Find the top 5 most expen
sive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitP

rice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n   List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.GenreId, g.Name, COUNT(t.TrackId) AS TrackCount\nFROM Genre g\nJOIN Track t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId, g.Name\nORDER BY TrackCount DESC"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": " \n   Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM Customer 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 c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM Invoice \nWHERE Total > 10.00\n"}, {"role": "user", "content": "How many records are in table called customer"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM Customer"}, {"role": "user", "content": "Show me a list of tables in the SQLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type='table'"}, {"role": "user", "content": " \n   Get the top 10 most popular artists (based on the number of tracks):\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:46:39.064839506Z', 'message': {'role': 'assistant', 'content': '```\nSELECT ar.Name, COUNT(t.TrackId) AS TrackCount\nFROM Artist ar\nJOIN Track t ON ar.ArtistId = t.ArtistId\nGROUP BY ar.Name\nORDER BY TrackCount DESC\nLIMIT 10;\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 86783172381, 'load_duration': 742151, 'prompt_eval_count': 1242, 'prompt_eval_duration': 77715181000, 'eval_count': 50, 'eval_duration': 8333034000}
```
SELECT ar.Name, COUNT(t.TrackId) AS TrackCount
FROM Artist ar
JOIN Track t ON ar.ArtistId = t.ArtistId
GROUP BY ar.Name
ORDER BY TrackCount DESC
LIMIT 10;
```
Output from LLM: ```
SELECT ar.Name, COUNT(t.TrackId) AS TrackCount
FROM Artist ar
JOIN Track t ON ar.ArtistId = t.ArtistId
GROUP BY ar.Name
ORDER BY TrackCount DESC
LIMIT 10;

```
Extracted SQL: SELECT ar.Name, COUNT(t.TrackId) AS TrackCount
FROM Artist ar
JOIN Track t ON ar.ArtistId = t.ArtistId
GROUP BY ar.Name
ORDER BY TrackCount DESC
LIMIT 10
SELECT ar.Name, COUNT(t.TrackId) AS TrackCount
FROM Artist ar
JOIN Track t ON ar.ArtistId = t.ArtistId
GROUP BY ar.Name
ORDER BY TrackCount DESC
LIMIT 10
Couldn't run sql:  Execution failed on sql 'SELECT ar.Name, COUNT(t.TrackId) AS TrackCount
FROM Artist ar
JOIN Track t ON ar.ArtistId = t.ArtistId
GROUP BY ar.Name
ORDER BY TrackCount DESC
LIMIT 10': no such column: t.ArtistId
```

In [27]:
```
question = """
     List all customers from Canada and their email addresses:
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE 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(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) 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\n\n   CREATE TABLE IF NOT EXISTS t_person (\n   id INT PRIMARY KEY,\n   name VARCHAR(100),\n   email text,\n   age INT\n   )\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   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\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   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   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 INDEX IFK_InvoiceLineTrackId ON InvoiceLine (TrackId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON InvoiceLine (InvoiceId)\n\nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC'}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM Customer'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerI

d, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId'}, {'role': 'user', 'content': " \n   Lis
t all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': 'SELECT
e.LastName, e.FirstName, m.LastName, m.FirstName\nFROM Employee e\nLEFT JOIN Employee m ON e.ReportsTo = m.
EmployeeId'}, {'role': 'user', 'content': ' \n   Get the average invoice total for each customer:\n'},
{'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM Customer 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', 'content': 'SELECT * \nFROM Invoic
e \nWHERE Total > 10.00\n'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the tota
l amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM
Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {'role': 'user', 'content': ' \n   Find
the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.Trac
kId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': '
\n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT
a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user',
'content': ' \n   List all customers from Canada and their email addresses:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE 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\n\n   CREATE TABLE IF NOT EXISTS t_person (\n   id INT PRIMARY KE
Y,\n   name VARCHAR(100),\n   email text,\n   age INT\n   )\n\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_InvoiceCustomerId ON Invoice (CustomerId)\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 InvoiceLine\n(\n   In
voiceLineId INTEGER NOT NULL,\n   InvoiceId INTEGER NOT NULL,\n   TrackId INTEGER NOT NULL,\n   UnitP

```
rice 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_InvoiceLineTrackId ON InvoiceLine (TrackId)\n\nCREATE INDEX IFK_InvoiceLin
eInvoiceId ON InvoiceLine (InvoiceId)\n\nCREATE INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\n\n\n==
=Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on t
ime and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a val
id SQL query without any explanations for the question. \n2. If the provided context is almost sufficient b
ut requires knowledge of a specific string in a particular column, please generate an intermediate SQL quer
y to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n
3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the m
ost relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer ex
actly as it was given before. \n"}, {"role": "user", "content": "what are the top 5 countries that customer
s come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer
\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    Find the
total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.Invo
iceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country
\nORDER BY TotalInvoices DESC"}, {"role": "user", "content": "How many records are in table called custome
r"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM Customer"}, {"role": "user", "content": "  \n
Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT CustomerI
d, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId"}, {"role": "user", "content": "  \n    Lis
t all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT
e.LastName, e.FirstName, m.LastName, m.FirstName\nFROM Employee e\nLEFT JOIN Employee m ON e.ReportsTo = m.
EmployeeId"}, {"role": "user", "content": "  \n    Get the average invoice total for each customer:\n"},
{"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM Customer 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", "content": "SELECT * \nFROM Invoic
e \nWHERE Total > 10.00\n"}, {"role": "user", "content": "  \n    Find all invoices since 2010 and the tota
l amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM
Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {"role": "user", "content": "  \n    Find
the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT t.Trac
kId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "
\n    List all albums and their corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT
a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId"}, {"role": "user",
"content": "  \n     List all customers from Canada and their email addresses:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:48:15.29178545Z', 'message': {'role': 'assistant',
'content': "```\nSELECT c.Email, c.Country, c.CustomerId\nFROM Customer c\nWHERE c.Country = 'Canada'\n``
`"}, 'done_reason': 'stop', 'done': True, 'total_duration': 96082375392, 'load_duration': 2811461, 'prompt_
eval_count': 1443, 'prompt_eval_duration': 91048150000, 'eval_count': 26, 'eval_duration': 4294985000}
```

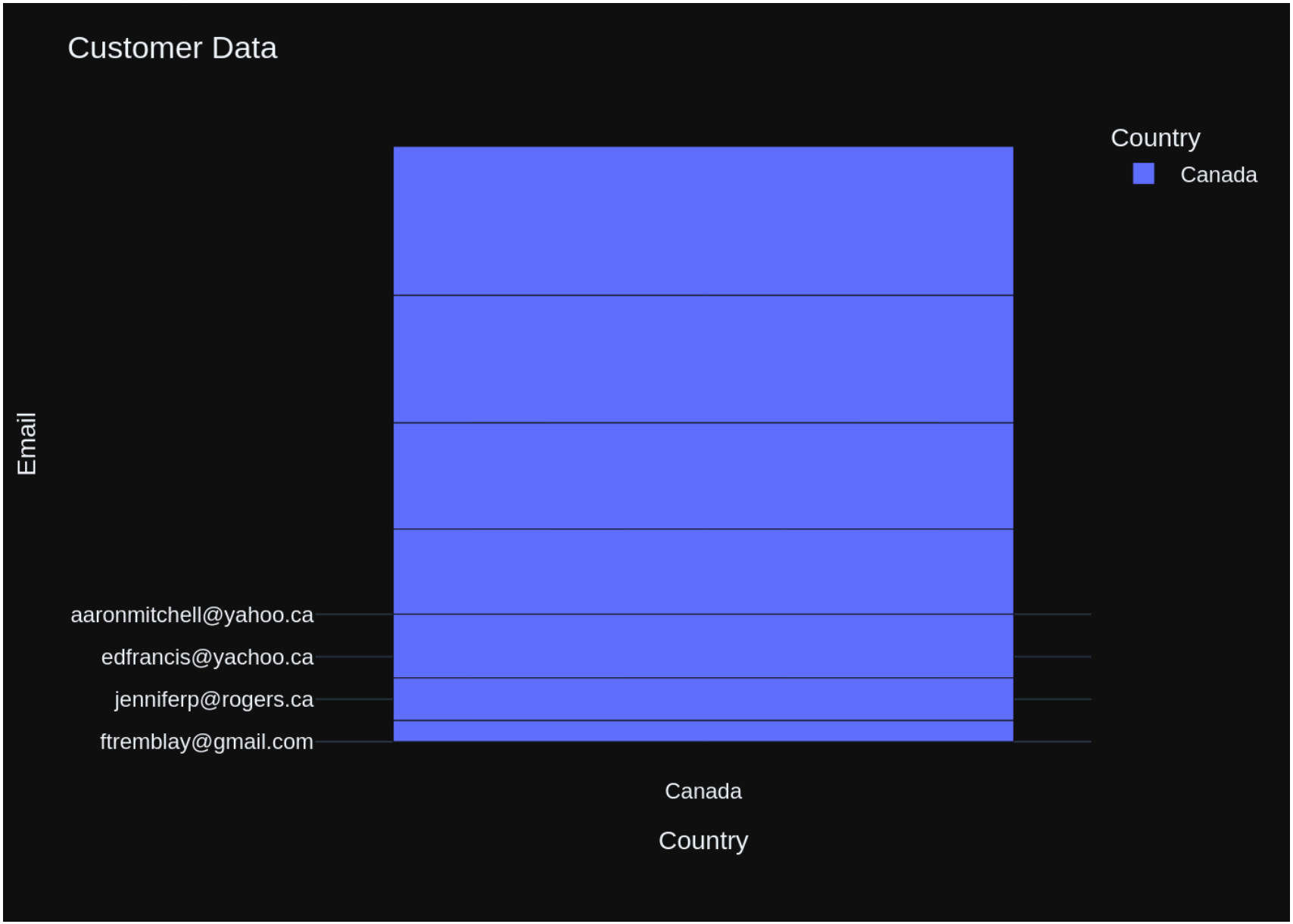
SELECT c.Email, c.Country, c.CustomerId

```
FROM Customer c
WHERE c.Country = 'Canada'
```
Output from LLM: ```
```
SELECT c.Email, c.Country, c.CustomerId
FROM Customer c
WHERE c.Country = 'Canada'
```
Extracted SQL: SELECT c.Email, c.Country, c.CustomerId
FROM Customer c
WHERE c.Country = 'Canada'

SELECT c.Email, c.Country, c.CustomerId
FROM Customer c
WHERE c.Country = 'Canada'

```
                    Email Country  CustomerId
0     ftremblay@gmail.com  Canada           3
1      mphilips12@shaw.ca  Canada          14
2     jenniferp@rogers.ca  Canada          15
3        robbrown@shaw.ca  Canada          29
4     edfrancis@yachoo.ca  Canada          30
5    marthasilk@gmail.com  Canada          31
6  aaronmitchell@yahoo.ca  Canada          32
7  ellie.sullivan@shaw.ca  Canada          33
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n     List all customers from Canada and their email addresse
s:\n'\n\nThe DataFrame was produced using this query: SELECT c.Email, c.Country, c.CustomerId\nFROM Custome
r c\nWHERE c.Country = 'Canada'\n\n\nThe following is information about the resulting pandas DataFrame 'd
f': \nRunning df.dtypes gives:\n Email         object\nCountry       object\nCustomerId     int64\ndtype: o
bject"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the d
ataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the datafram
e, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the cod
e."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:48:35.727822834Z', 'message': {'role': 'assistant',
'content': '```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = px.bar(df, x="Coun
```

try", y="Email", color="Country")\n\nfig.update_layout(title_text=\'Customer Data\', xaxis_title=\'Country\')\n\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 20293216356, 'load_duration': 2658946, 'prompt_eval_count': 187, 'prompt_eval_duration': 11131144000, 'eval_count': 56, 'eval_duration': 9012384000}



Out[27]: ("SELECT c.Email, c.Country, c.CustomerId\nFROM Customer c\nWHERE c.Country = 'Canada'\n",

	Email	Country	CustomerId
0	ftremblay@gmail.com	Canada	3
1	mphilips12@shaw.ca	Canada	14
2	jenniferp@rogers.ca	Canada	15
3	robbrown@shaw.ca	Canada	29
4	edfrancis@yachoo.ca	Canada	30
5	marthasilk@gmail.com	Canada	31
6	aaronmitchell@yahoo.ca	Canada	32
7	ellie.sullivan@shaw.ca	Canada	33,

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

In [28]:
```
question = """
     Find the customer with the most invoices
"""

vn.ask(question=question)
```

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

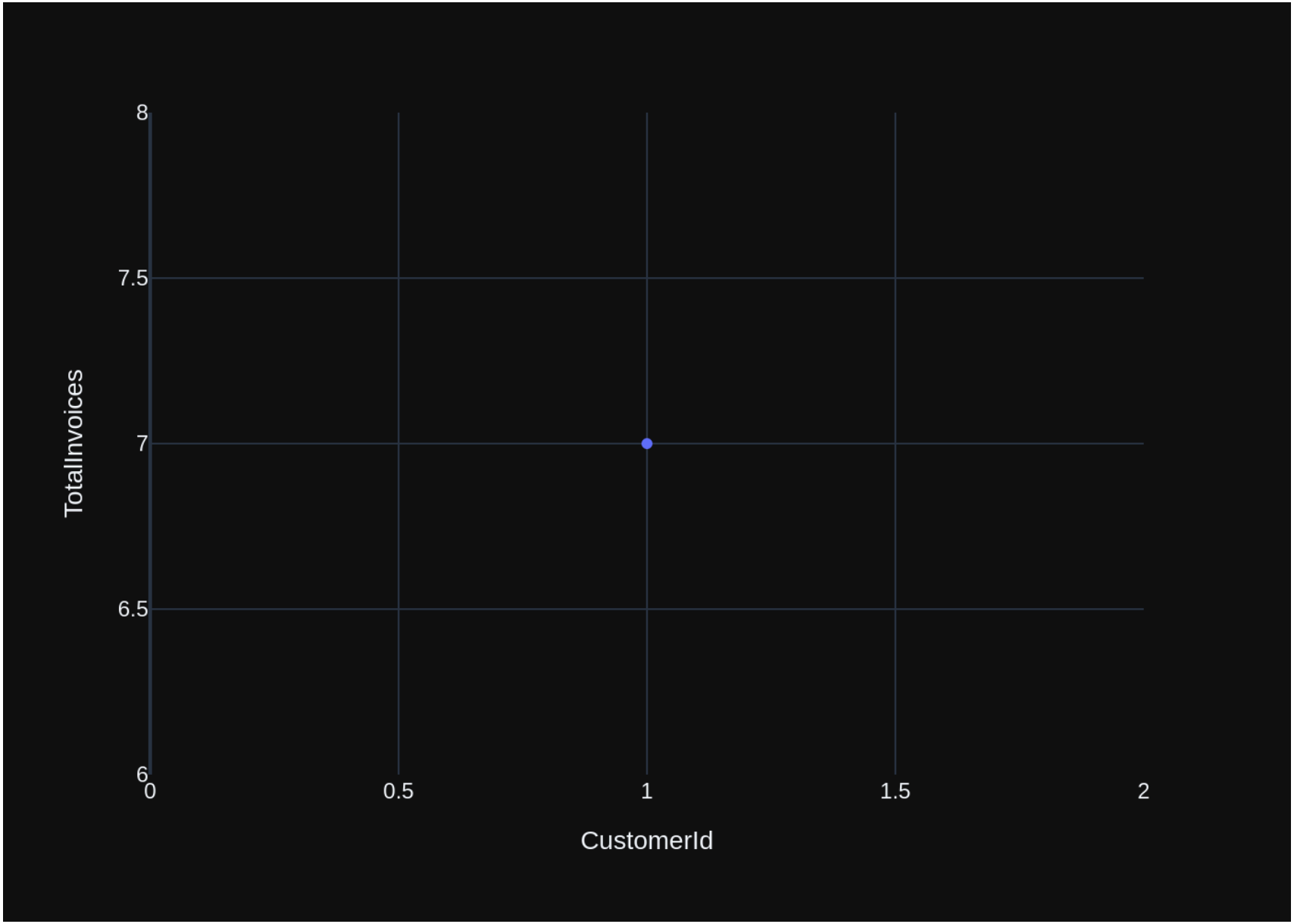
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_InvoiceCustomerId ON Invoice (CustomerId)\n\nCREATE 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(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_InvoiceLineInvoiceId ON InvoiceLine (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 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 INDEX IFK_InvoiceLineTrackId 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 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) 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\n\n   CREATE TABLE IF NOT EXISTS t_person (\n   id INT PRIMARY KEY,\n   name VARCHAR(100),\n   email text,\n   age INT\n   )\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   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 INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM Invoice \nWHERE Total > 10.00\n'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content':

"SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY Inv
oiceId"}, {'role': 'user', 'content': ' \n   Get the average invoice total for each customer:\n'}, {'rol
e': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM Customer c\nJOI
N Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n   F
ind the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.
TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'conten
t': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Cou
ntry, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5'},
{'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 customers from Canada
and their email addresses:\n'}, {'role': 'assistant', 'content': "SELECT c.Email, c.Country, c.CustomerId\n
FROM Customer c\nWHERE c.Country = 'Canada'\n"}, {'role': 'user', 'content': " \n   List all employees an
d their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': 'SELECT e.LastName, e.Firs
tName, m.LastName, m.FirstName\nFROM Employee e\nLEFT JOIN Employee m ON e.ReportsTo = m.EmployeeId'}, {'ro
le': 'user', 'content': ' \n   Find the customer with the most invoices \n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE 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\n\n   CREATE TABLE IF NOT EXISTS t_person (\n   id INT PRIMARY
KEY,\n   name VARCHAR(100),\n   email text,\n   age INT\n   )\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    Post
alCode NVARCHAR(10),\n    Phone NVARCHAR(24),\n    Fax NVARCHAR(24),\n    Email NVARCHAR(60),\n    CONSTRAI
NT 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_EmployeeReportsTo ON Employee (ReportsT
o)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are deli
vered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please gene
rate a valid SQL query without any explanations for the question. \n2. If the provided context is almost su
fficient but requires knowledge of a specific string in a particular column, please generate an intermediat
e SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermedia
te_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please
use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the
answer exactly as it was given before. \n"}, {"role": "user", "content": "  \n    Get the total number of i
nvoices for each customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) AS TotalInvoic
es\nFROM Invoice\nGROUP BY CustomerId"}, {"role": "user", "content": "  \n    Find the total number of invo
ices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoi
ces\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvo
ices DESC"}, {"role": "user", "content": "  \n    List all invoices with a total exceeding $10:\n"}, {"rol
e": "assistant", "content": "SELECT * \nFROM Invoice \nWHERE Total > 10.00\n"}, {"role": "user", "content":
"  \n    Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content":
"SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY Inv
oiceId"}, {"role": "user", "content": "  \n    Get the average invoice total for each customer:\n"}, {"rol
e": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM Customer c\nJOI
N Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": "  \n    F
ind the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT t.
TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "conten
t": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Cou
ntry, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5"},
{"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 customers from Canada
and their email addresses:\n"}, {"role": "assistant", "content": "SELECT c.Email, c.Country, c.CustomerId\n
FROM Customer c\nWHERE c.Country = 'Canada'\n"}, {"role": "user", "content": "  \n    List all employees an
d their reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT e.LastName, e.Firs
tName, m.LastName, m.FirstName\nFROM Employee e\nLEFT JOIN Employee m ON e.ReportsTo = m.EmployeeId"}, {"ro
le": "user", "content": "  \n     Find the customer with the most invoices \n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:50:19.045657441Z', 'message': {'role': 'assistant',
'content': '```\nSELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i
ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1;\n```'}, 'done_
reason': 'stop', 'done': True, 'total_duration': 103188218615, 'load_duration': 1172377, 'prompt_eval_coun
t': 1482, 'prompt_eval_duration': 93499444000, 'eval_count': 53, 'eval_duration': 8937497000}
```

```
SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalInvoices DESC
LIMIT 1;
```
Output from LLM: ```
```
SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalInvoices DESC
LIMIT 1;
```
Extracted SQL: SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalInvoices DESC
LIMIT 1
SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalInvoices DESC
LIMIT 1
   CustomerId TotalInvoices
0   1   7
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: ' \n   Find the customer with the most invoices \n'\n\nThe Dat
aFrame was produced using this query: SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Custom
er c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLI
MIT 1\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes give
s:\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 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."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:50:53.667546154Z', 'message': {'role': 'assistant',
'content': '```\nimport plotly.express as px\nimport plotly.graph_objects as go\n\nfig = go.Figure(data=[g
o.Bar(x=\'CustomerId\', y=\'TotalInvoices\', text_auto=True)] if df.shape[0] > 1 else [\n   go.Indicator
(\n   mode="number",\n   value=df[\'TotalInvoices\'].values[0],\n   number={\'font\': {\'siz
e\': 40}, \'digits\': \'0\'},\n   title_text="Customer with Most Invoices"\n   )\n])\n\nfig.update_la
yout(title=\'Customers with Most Invoices\', xaxis_title=\'Customer ID\', yaxis_title=\'Number of Invoices
\')\n\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 34475828022, 'load_duratio
n': 2745013, 'prompt_eval_count': 207, 'prompt_eval_duration': 12337570000, 'eval_count': 136, 'eval_durati
on': 21981602000}


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

Out[28]:
```
('SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerI
d = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1',
    CustomerId  TotalInvoices
 0           1             7,
 Figure({
     'data': [{'hovertemplate': 'CustomerId=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'symbol': 'circle'},
               'mode': 'markers',
               'name': '',
               'orientation': 'v',
               'showlegend': False,
               'type': 'scatter',
               'x': array([1]),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'}],
     'layout': {'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalInvoices'}}}
 }))
```

In [ ]:

Advanced SQL questions

In [29]:
```
question = """
     Find the customer who bought the most albums in total quantity (across all invoices):
"""

vn.ask(question=question)
```

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

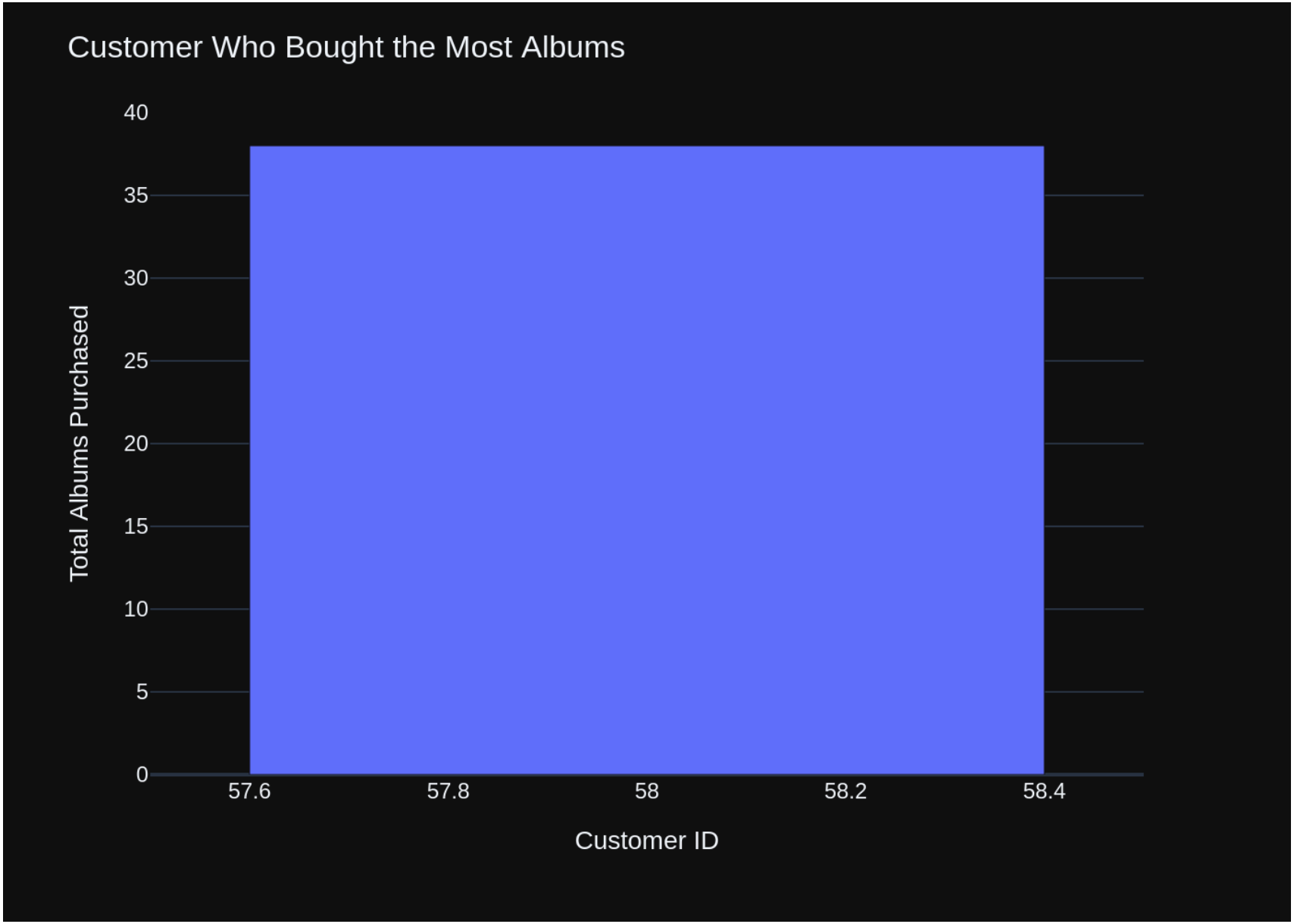
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE Track\n(\n   TrackId INTEGER NOT NULL,\n   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 (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 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_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   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 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(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_TrackAlbumId ON Track (AlbumId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON InvoiceLine (InvoiceId)\n\nCREATE INDEX IFK_InvoiceLineTrackId 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 (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\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': ' \n   Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on un

it price):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDE
R BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the total number of invoices pe
r country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFR
OM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DE
SC'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'ass
istant', 'content': 'SELECT * \nFROM Invoice \nWHERE Total > 10.00\n'}, {'role': 'user', 'content': ' \n
Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerI
d, AVG(i.Total) AS AverageInvoiceTotal\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGRO
UP BY c.CustomerId'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist n
ames \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artis
t ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   List all genres and the number of
tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.GenreId, g.Name, COUNT(t.TrackId) AS
TrackCount\nFROM Genre g\nJOIN Track t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId, g.Name\nORDER BY Track
Count DESC'}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assis
tant', 'content': 'SELECT COUNT(*) FROM Customer'}, {'role': 'user', 'content': ' \n   Find the customer
who bought the most albums in total quantity (across all invoices): \n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE 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\n===Additional Context \n\nOur b
usiness defines OTIF score as the percentage of orders that are delivered on time and in full\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   Find the customer with the most invoices \n"}, {"role": "assist
ant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i
ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "us
er", "content": " \n   Get the total number of invoices for each customer\n"}, {"role": "assistant", "con
tent": "SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId"}, {"role": "user",
"content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant",
"content": "SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nG
ROUP BY InvoiceId"}, {"role": "user", "content": " \n   Find the top 5 most expensive tracks (based on un
it price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDE
R BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find the total number of invoices pe
r country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFR
OM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DE
SC"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "ass
istant", "content": "SELECT * \nFROM Invoice \nWHERE Total > 10.00\n"}, {"role": "user", "content": " \n
Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerI
d, AVG(i.Total) AS AverageInvoiceTotal\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGRO
UP BY c.CustomerId"}, {"role": "user", "content": " \n   List all albums and their corresponding artist n
ames \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artis
t ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   List all genres and the number of
tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.GenreId, g.Name, COUNT(t.TrackId) AS
TrackCount\nFROM Genre g\nJOIN Track t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId, g.Name\nORDER BY Track
Count DESC"}, {"role": "user", "content": "How many records are in table called customer"}, {"role": "assis
tant", "content": "SELECT COUNT(*) FROM Customer"}, {"role": "user", "content": " \n   Find the customer
who bought the most albums in total quantity (across all invoices): \n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:52:41.664448883Z', 'message': {'role': 'assistant',
'content': '```\nSELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums\nFROM Customer c\nJOIN Invoice i ON
c.CustomerId = i.CustomerId\nJOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId\nGROUP BY c.CustomerId\nORDE
R BY TotalAlbums DESC\nLIMIT 1;\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 10782144586

```
6, 'load_duration': 685743, 'prompt_eval_count': 1516, 'prompt_eval_duration': 95906778000, 'eval_count': 6
6, 'eval_duration': 11173055000}
```
```
SELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1;
```
Output from LLM: ```
```
SELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1;
```
Extracted SQL: SELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1
SELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1
   CustomerId TotalAlbums
0   58   38
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query

that answers the question the user asked: ' \n   Find the customer who bought the most albums in total q
uantity (across all invoices): \n'\n\nThe DataFrame was produced using this query: SELECT c.CustomerId, SUM
(il.Quantity) AS TotalAlbums\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nJOIN InvoiceL
ine il ON i.InvoiceId = il.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n\nThe foll
owing 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 plot
ly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If the
re is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with
any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:53:06.470494544Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nimport numpy as np\n\nfig = px.bar(df, x='CustomerId', y='Tot
alAlbums')\n\nfig.update_layout(title='Customer Who Bought the Most Albums',\n   yaxis_titl
e='Total Albums Purchased',\n   xaxis_title='Customer ID')\nfig.show()\n```"}, 'done_reaso
n': 'stop', 'done': True, 'total_duration': 24636650367, 'load_duration': 2802278, 'prompt_eval_count': 23
3, 'prompt_eval_duration': 13921191000, 'eval_count': 66, 'eval_duration': 10564852000}


Customer Who Bought the Most Albums
Total Albums Purchased
40
35
30
25
20
15
10
5
0
57.6
57.8
58
58.2
58.4
Customer ID

Out[29]:
```
('SELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums\nFROM Customer c\nJOIN Invoice i ON c.CustomerId =
i.CustomerId\nJOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbu
ms DESC\nLIMIT 1',
    CustomerId  TotalAlbums
 0          58           38,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'CustomerId=%{x}<br>TotalAlbums=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([58]),
               'xaxis': 'x',
               'y': array([38]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'title': {'text': 'Customer Who Bought the Most Albums'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Customer ID'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total Albums Purchase
d'}}}
 }))
```

In [30]:
```
question = """
     Find the top 5 customer who bought the most albums in total quantity (across all invoices):
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE 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 (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 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_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   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 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 UPDATE NO ACTION\n)\n\nCREATE INDEX IFK_InvoiceCustomerId 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 (InvoiceId)\n\nCREATE TABLE Artist\n(\n   ArtistId INTEGER NOT NULL,\n   Name NVARCHAR(120),\n   CONSTRAINT PK_Artist PRIMARY KEY (ArtistId)\n)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': ' \n   Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nJOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer

\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId'}, {'role': 'user', 'content': ' \n   Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM 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 c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM Invoice \nWHERE Total > 10.00\n'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find the top 5 customer who bought the most albums in total quantity (across all invoices):\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and 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 (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 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_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   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 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 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 (InvoiceId)\n\nCREATE TABLE Artist\n(\n   ArtistId INTEGER NOT NULL,\n   Name NVARCHAR(120),\n   CONSTRAINT PK_Artist PRIMARY KEY (ArtistId)\n)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n   Find the customer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nJOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n   Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n   Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId"}, {"role": "user", "content": " \n   Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM 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 c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \nFROM Invoice \nWHERE Total > 10.00\n"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   Find the top 5 customer who bought the most albums in total quantity (across all invoices):\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:54:52.74495965Z', 'message': {'role': 'assistant', 'content': '```\nSELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nJOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, 'done_reason': 'stop', 'done': True, 'total_duration': 106122314269, 'loa

```
d_duration': 1006377, 'prompt_eval_count': 1497, 'prompt_eval_duration': 94569573000, 'eval_count': 64, 'ev
al_duration': 10834720000}
```
```
SELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5
```
```
SELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5
Couldn't run sql:  Execution failed on sql '```
SELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5': unrecognized token: "```
SELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5"
```

In [31]:
```
question = """
     Find the top 3 customers who spent the most money overall:
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE Invoice\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 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   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 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(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) 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 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 (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 INDEX IFK_EmployeeReportsTo ON Employee (ReportsTo)\n\n\n   CREATE TABLE IF NOT EXISTS t_person (\n   id INT PRIMARY KEY,\n   name VARCHAR(100),\n   email text,\n   age INT\n   )\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 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 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   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\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the quest

ion has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'ro
le': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role':
'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLI
MIT 5'}, {'role': 'user', 'content': ' \n   Find the customer who bought the most albums in total quanti
ty (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(il.Quantity) AS
TotalAlbums\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nJOIN InvoiceLine il ON i.Invoi
ceId = il.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'conten
t': ' \n   Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.Cus
tomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerI
d\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': 'what are the
top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS
TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user',
'content': ' \n   Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content':
'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM Customer c\nJOIN Invoice i ON c.CustomerId
= i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n   List all invoices with a tota
l exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \nFROM Invoice \nWHERE Total > 10.00\n'},
{'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assi
stant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId'}, {'ro
le': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant',
'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.Cus
tomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC'}, {'role': 'user', 'content': '
\n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SE
LECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY Invoic
eId"}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assistant',
'content': 'SELECT COUNT(*) FROM Customer'}, {'role': 'user', 'content': ' \n   Find the top 3 customers
who spent the most money overall:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE Invoice\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\n\n   C
REATE TABLE IF NOT EXISTS t_person (\n   id INT PRIMARY KEY,\n   name VARCHAR(100),\n   emai
l text,\n   age INT\n   )\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\nCREATE TA
BLE Employee\n(\n   EmployeeId INTEGER NOT NULL,\n   LastName NVARCHAR(20) NOT NULL,\n   FirstName NVA
RCHAR(20) NOT NULL,\n   Title NVARCHAR(30),\n   ReportsTo INTEGER,\n   BirthDate DATETIME,\n   HireDat
e DATETIME,\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),\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\n===Additional Context \n\nOur business d
efines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guideli
nes \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations
for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific st
ring in a particular column, please generate an intermediate SQL query to find the distinct strings in that
column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficie
nt, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the quest
ion has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"ro
le": "user", "content": " \n   Find the top 5 most expensive tracks (based on unit price):\n"}, {"role":
"assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLI
MIT 5"}, {"role": "user", "content": " \n   Find the customer who bought the most albums in total quanti
ty (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(il.Quantity) AS
TotalAlbums\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nJOIN InvoiceLine il ON i.Invoi
ceId = il.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "conten
t": " \n   Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.Cus
tomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerI
d\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": "what are the
top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS

TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM 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", "content": "SELECT * \nFROM Invoice \nWHERE Total > 10.00\n"}, {"role": "user", "content": " \n   Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {"role": "user", "content": "How many records are in table called customer"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM Customer"}, {"role": "user", "content": " \n   Find the top 3 customers who spent the most money overall:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:56:55.597656563Z', 'message': {'role': 'assistant', 'content': '```\nSELECT c.CustomerId, SUM(il.Total) AS TotalSpent\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nJOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 3'}, 'done_reason': 'stop', 'done': True, 'total_duration': 122797999290, 'load_duration': 1107688, 'prompt_eval_count': 1741, 'prompt_eval_duration': 111093822000, 'eval_count': 64, 'eval_duration': 10947067000}
```
SELECT c.CustomerId, SUM(il.Total) AS TotalSpent
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 3
```
```
SELECT c.CustomerId, SUM(il.Total) AS TotalSpent
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 3
Couldn't run sql:  Execution failed on sql '```
SELECT c.CustomerId, SUM(il.Total) AS TotalSpent
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
```

```
JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 3': unrecognized token: "```
SELECT c.CustomerId, SUM(il.Total) AS TotalSpent
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 3"
```

In [32]:
```
question = """
     Get all playlists containing at least 10 tracks and the total duration of those tracks:
"""

vn.ask(question=question)
```

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

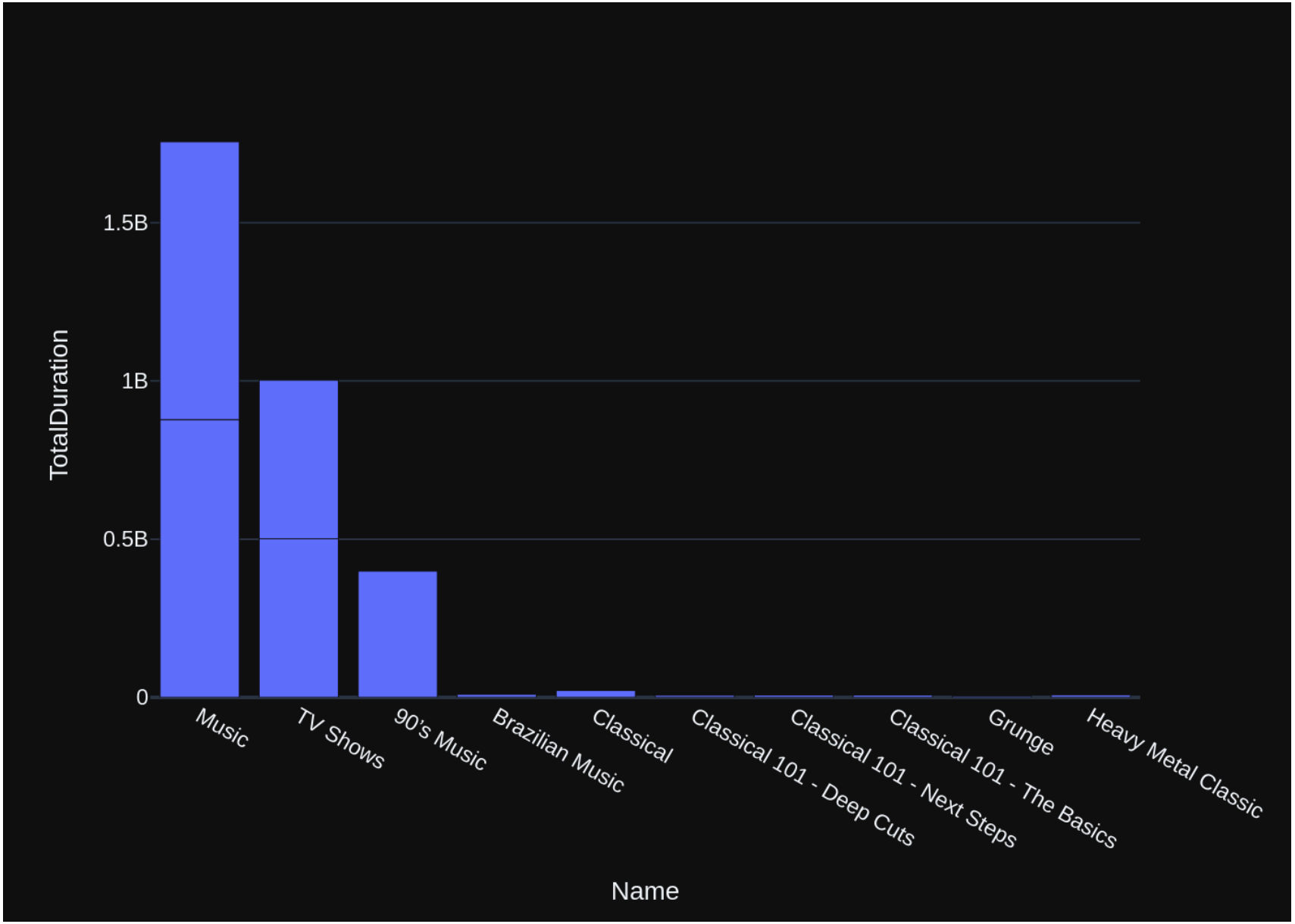
[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_PlaylistTrackTrackId ON PlaylistTrack (TrackId)\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 (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 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 (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_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 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_InvoiceLineTrackId ON InvoiceLine (TrackId)\n\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': ' \n   List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.GenreId, g.Name, COUNT(t.TrackId) AS TrackCount\nFROM Genre g\nJOIN Track t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId, g.Name\nORDER BY TrackCount DESC'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': ' \n   Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nJOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {'role': 'user', 'content': 'Show me a

list of tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master W
HERE type='table'"}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each custome
r\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP
BY CustomerId'}, {'role': 'user', 'content': ' \n   Find the customer with the most invoices \n'}, {'rol
e': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOI
N Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'},
{'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistan
t', 'content': 'SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY Total
Customers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Get all playlists containing at least 10 t
racks and the total duration of those tracks:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and 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\n===Additional Context \n\nOur business defines OTIF score as the perce
ntage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided conte
xt is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the
provided context is almost sufficient but requires knowledge of a specific string in a particular column, p
lease generate an intermediate SQL query to find the distinct strings in that column. Prepend the query wit
h a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it ca
n't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answe
red before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "

\n   List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.GenreId, g.Name, COUNT(t.TrackId) AS TrackCount\nFROM Genre g\nJOIN Track t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId, g.Name\nORDER BY TrackCount DESC"}, {"role": "user", "content": " \n   Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": " \n   Find the customer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId \nJOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceId, SUM(Total) AS TotalAmount\nFROM Invoice\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceId"}, {"role": "user", "content": "Show me a list of tables in the SQLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master \nWHERE type='table'"}, {"role": "user", "content": " \n   Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(*) AS TotalInvoices\nFROM Invoice\nGROUP BY CustomerId"}, {"role": "user", "content": " \n   Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Get all playlists containing at least 10 tracks and the total duration of those tracks:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:58:30.173912667Z', 'message': {'role': 'assistant', 'content': '```\nSELECT p.PlaylistId, p.Name, SUM(t.Milliseconds) AS TotalDuration\nFROM Playlist p\nJOIN PlaylistTrack pt ON p.PlaylistId = pt.PlaylistId\nJOIN Track t ON pt.TrackId = t.TrackId\nGROUP BY p.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10;\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 94531804425, 'load_duration': 720427, 'prompt_eval_count': 1296, 'prompt_eval_duration': 81285441000, 'eval_count': 74, 'eval_duration': 12511767000}
```
SELECT p.PlaylistId, p.Name, SUM(t.Milliseconds) AS TotalDuration
FROM Playlist p
JOIN PlaylistTrack pt ON p.PlaylistId = pt.PlaylistId
JOIN Track t ON pt.TrackId = t.TrackId
GROUP BY p.PlaylistId, p.Name
HAVING COUNT(pt.TrackId) >= 10;
```
Output from LLM: ```
SELECT p.PlaylistId, p.Name, SUM(t.Milliseconds) AS TotalDuration

```
FROM Playlist p
JOIN PlaylistTrack pt ON p.PlaylistId = pt.PlaylistId
JOIN Track t ON pt.TrackId = t.TrackId
GROUP BY p.PlaylistId, p.Name
HAVING COUNT(pt.TrackId) >= 10;
```
```
Extracted SQL: SELECT p.PlaylistId, p.Name, SUM(t.Milliseconds) AS TotalDuration
FROM Playlist p
JOIN PlaylistTrack pt ON p.PlaylistId = pt.PlaylistId
JOIN Track t ON pt.TrackId = t.TrackId
GROUP BY p.PlaylistId, p.Name
HAVING COUNT(pt.TrackId) >= 10
SELECT p.PlaylistId, p.Name, SUM(t.Milliseconds) AS TotalDuration
FROM Playlist p
JOIN PlaylistTrack pt ON p.PlaylistId = pt.PlaylistId
JOIN Track t ON pt.TrackId = t.TrackId
GROUP BY p.PlaylistId, p.Name
HAVING COUNT(pt.TrackId) >= 10
    PlaylistId                         Name  TotalDuration
0            1                        Music      877683083
1            3                    TV Shows      501094957
2            5                  90’s Music      398705153
3            8                        Music      877683083
4           10                    TV Shows      501094957
5           11               Brazilian Music        9486559
6           12                     Classical       21770592
7           13   Classical 101 - Deep Cuts        6755730
8           14  Classical 101 - Next Steps        7575051
9           15  Classical 101 - The Basics        7439811
10          16                       Grunge        4122018
11          17         Heavy Metal Classic        8206312
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n     Get all playlists containing at least 10 tracks and the
total duration of those tracks:\n'\n\nThe DataFrame was produced using this query: SELECT p.PlaylistId, p.N
ame, SUM(t.Milliseconds) 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.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >=
10\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n
```

PlaylistId int64\nName object\nTotalDuration int64\ndtype: object"}, {"role": "use
r", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the d
ata is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Re
spond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-08T23:58:50.094451626Z', 'message': {'role': 'assistant',
'content': "```\nimport plotly.express as px\nfig = px.bar(df, x='Name', y='TotalDuration')\nfig.show()\n``
`"}, 'done_reason': 'stop', 'done': True, 'total_duration': 19746486503, 'load_duration': 2890874, 'prompt_
eval_count': 249, 'prompt_eval_duration': 14868091000, 'eval_count': 30, 'eval_duration': 4721925000}


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

Out[32]:
```
('SELECT p.PlaylistId, p.Name, SUM(t.Milliseconds) AS TotalDuration\nFROM Playlist p\nJOIN PlaylistTrack p
t ON p.PlaylistId = pt.PlaylistId\nJOIN Track t ON pt.TrackId = t.TrackId\nGROUP BY p.PlaylistId, p.Name\n
HAVING COUNT(pt.TrackId) >= 10',
     PlaylistId                        Name  TotalDuration
 0            1                       Music      877683083
 1            3                   TV Shows      501094957
 2            5                  90’s Music      398705153
 3            8                       Music      877683083
 4           10                   TV Shows      501094957
 5           11              Brazilian Music        9486559
 6           12                    Classical       21770592
 7           13   Classical 101 - Deep Cuts        6755730
 8           14  Classical 101 - Next Steps        7575051
 9           15  Classical 101 - The Basics        7439811
 10          16                      Grunge        4122018
 11          17         Heavy Metal Classic        8206312,
 Figure({
     'data': [{'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(['Music', 'TV Shows', '90’s Music', 'Music', 'TV Shows',
                           'Brazilian Music', 'Classical', 'Classical 101 - Deep Cuts',
                           'Classical 101 - Next Steps', 'Classical 101 - The Basics', 'Grunge',
                           'Heavy Metal Classic'], dtype=object),
               'xaxis': 'x',
               'y': array([877683083, 501094957, 398705153, 877683083, 501094957,   9486559,
                            21770592,   6755730,   7575051,   7439811,   4122018,   8206312]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'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 [33]:
```
question = """
     Identify artists who have albums with tracks appearing in multiple genres:
"""

vn.ask(question=question)
```

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

[{'role': 'system', 'content': "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE 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 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 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 Artist\n(\n   ArtistId INTEGER NOT NULL,\n   Name NVARCHAR(120),\n   CONSTRAINT PK_Artist PRIMARY KEY (ArtistId)\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   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\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.GenreId, g.Name, COUNT(t.TrackId) AS TrackCount\nFROM Genre g\nJOIN Track t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId, g.Name\nORDER BY TrackCount DESC'}, {'role': 'user', 'content': ' \n   Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nJOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'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.PlaylistId, p.Name, SUM(t.Milliseconds) AS TotalDuration\nFROM Playlist p\nJOIN PlaylistTrack pt ON p.PlaylistId = pt.PlaylistId\nJOIN Track t ON pt.TrackId = t.TrackId\nGROUP BY p.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10'}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'conte

nt': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table'"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC'}, {'role': 'user', 'content': ' \n   Identify artists who have albums with tracks appearing in multiple genres:\n'}]
Ollama parameters:
model=llama3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_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 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 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 Artist\n(\n   ArtistId INTEGER NOT NULL,\n   Name NVARCHAR(120),\n   CONSTRAINT PK_Artist PRIMARY KEY (ArtistId)\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   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\n\n===Additional Context \n\nOur business defines OTIF score as the percentage of orders that are delivered on time and in full\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 fin

d the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name\nFROM Album a\nJOIN Artist ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.GenreId, g.Name, COUNT(t.TrackId) AS TrackCount\nFROM Genre g\nJOIN Track t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId, g.Name\nORDER BY TrackCount DESC"}, {"role": "user", "content": " \n   Find the customer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(il.Quantity) AS TotalAlbums\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nJOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"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.PlaylistId, p.Name, SUM(t.Milliseconds) AS TotalDuration\nFROM Playlist p\nJOIN PlaylistTrack pt ON p.PlaylistId = pt.PlaylistId\nJOIN Track t ON pt.TrackId = t.TrackId\nGROUP BY p.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10"}, {"role": "user", "content": " \n   Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM Track t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * \nFROM Track \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": "Show me a list of tables in the SQLite database"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master\nWHERE type='table'"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS TotalCustomers\nFROM Customer\nGROUP BY Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM Customer c\nJOIN Invoice i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\nORDER BY TotalInvoices DESC"}, {"role": "user", "content": " \n   Identify artists who have albums with tracks appearing in multiple genres:\n"}]
Ollama Response:
{'model': 'llama3:latest', 'created_at': '2024-06-09T00:00:34.944194974Z', 'message': {'role': 'assistant', 'content': '```\nSELECT a.ArtistId, a.Name \nFROM Artist a \nJOIN Album alb ON a.ArtistId = alb.ArtistId \nJOIN Track t ON alb.AlbumId = t.AlbumId \nWHERE t.GenreId IN (\n SELECT GenreId \n FROM Genre \n GROUP BY GenreId \n HAVING COUNT(DISTINCT TrackId) > 1\n)\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 104680346449, 'load_duration': 930681, 'prompt_eval_count': 1430, 'prompt_eval_duration': 90042563000, 'eval_count': 82, 'eval_duration': 13900924000}
```
SELECT a.ArtistId, a.Name
FROM Artist a
JOIN Album alb ON a.ArtistId = alb.ArtistId
```

```
JOIN Track t ON alb.AlbumId = t.AlbumId
WHERE t.GenreId IN (
  SELECT GenreId
  FROM Genre
  GROUP BY GenreId
  HAVING COUNT(DISTINCT TrackId) > 1
)
```
Output from LLM: ```
```
SELECT a.ArtistId, a.Name
FROM Artist a
JOIN Album alb ON a.ArtistId = alb.ArtistId
JOIN Track t ON alb.AlbumId = t.AlbumId
WHERE t.GenreId IN (
  SELECT GenreId
  FROM Genre
  GROUP BY GenreId
  HAVING COUNT(DISTINCT TrackId) > 1
)
```
Extracted SQL: SELECT a.ArtistId, a.Name
```
FROM Artist a
JOIN Album alb ON a.ArtistId = alb.ArtistId
JOIN Track t ON alb.AlbumId = t.AlbumId
WHERE t.GenreId IN (
  SELECT GenreId
  FROM Genre
  GROUP BY GenreId
  HAVING COUNT(DISTINCT TrackId) > 1
)

SELECT a.ArtistId, a.Name
FROM Artist a
JOIN Album alb ON a.ArtistId = alb.ArtistId
JOIN Track t ON alb.AlbumId = t.AlbumId
WHERE t.GenreId IN (
  SELECT GenreId
  FROM Genre
  GROUP BY GenreId
  HAVING COUNT(DISTINCT TrackId) > 1
)
```

```
Couldn't run sql:  Execution failed on sql 'SELECT a.ArtistId, a.Name
FROM Artist a
JOIN Album alb ON a.ArtistId = alb.ArtistId
JOIN Track t ON alb.AlbumId = t.AlbumId
WHERE t.GenreId IN (
  SELECT GenreId
  FROM Genre
  GROUP BY GenreId
  HAVING COUNT(DISTINCT TrackId) > 1
)
': misuse of aggregate: COUNT()
```

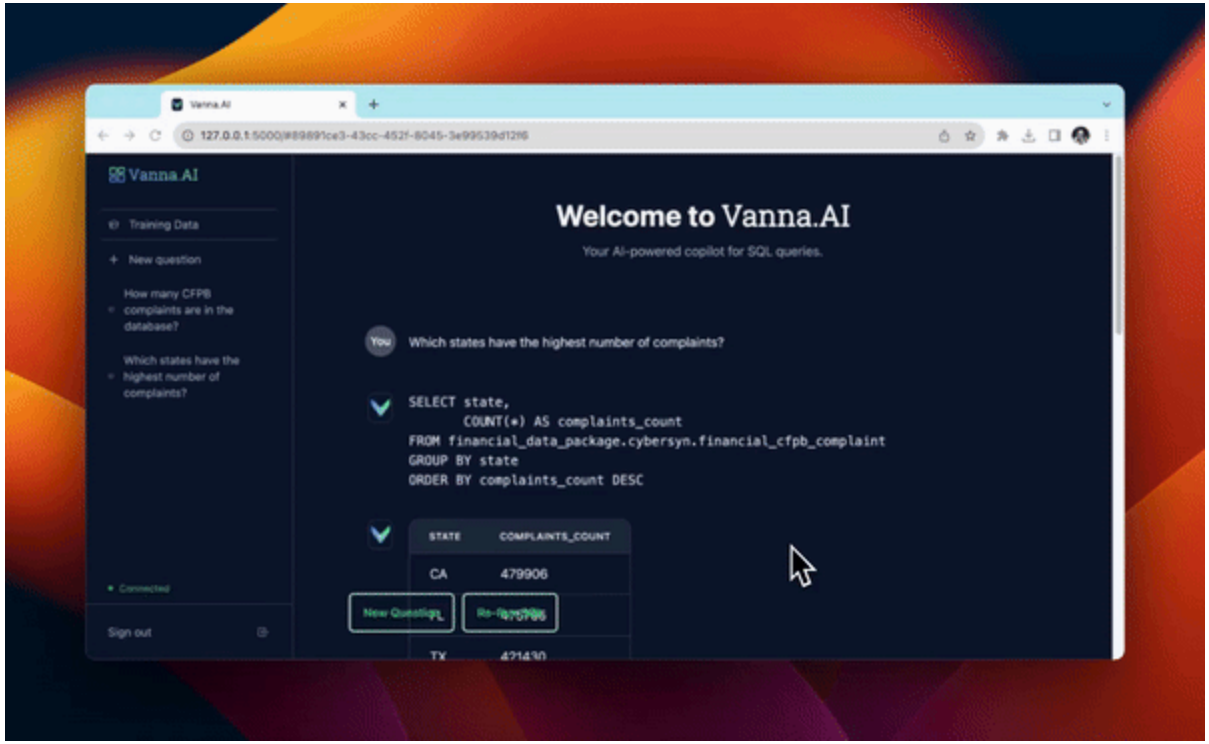
Check completion time

In [34]:
```
ts_stop = time()

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

```
elapsed_time : 2600.9133360385895 sec
```

In [ ]:
```

```

Launch the User Interface



from vanna.flask import VannaFlaskApp app = VannaFlaskApp(vn) app.run()

Next Steps

Using Vanna via Jupyter notebooks is great for getting started but check out additional customizable interfaces like the

- Streamlit app
- Flask app
- Slackbot