Generating SQL for SQLite using Ollama, ChromaDB

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

Which LLM do you want to use?

- OpenAI via Vanna.AI (Recommended)

  Use Vanna.AI for free to generate your queries
- OpenAI

  Use OpenAI with your own API key
- Azure OpenAI

  If you have OpenAI models deployed on Azure
- [Selected] Ollama

  Use Ollama locally for free. Requires additional setup.
- Mistral via Mistral API

  If you have a Mistral API key
- Other LLM

  If you have a different LLM model

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

- Vanna Hosted Vector DB (Recommended)

  Use Vanna.AIs hosted vector database (pgvector) for free. This is usable across machines with no additional setup.
- [Selected] ChromaDB

  Use ChromaDBs open-source vector database for free locally. No additional setup is necessary -- all database files will be created and stored locally.
- Marqo

  Use Marqo locally for free. Requires additional setup. Or use their hosted option.
- Other VectorDB

Use any other vector database. Requires additional setup.

Setup

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

In [1]:
```
import warnings
import re

warnings.filterwarnings('ignore', category=DeprecationWarning, message='^Number of requested results')
# warnings.filterwarnings('ignore', category=DeprecationWarning, message=re.escape(r'^Some regex pattern')

import os

import re
from time import time

from vanna.ollama import Ollama
from vanna.chromadb.chromadb_vector import ChromaDB_VectorStore
```

In [2]:
```
class MyVanna(ChromaDB_VectorStore, Ollama):
    def __init__(self, config=None):
        ChromaDB_VectorStore.__init__(self, config=config)
        Ollama.__init__(self, config=config)
```

In [3]:
```
file_db = "~/Downloads/chinook.sqlite"
model_name = 'codegemma'

clean_and_train = True  # False
```

In [4]:
```
config = {
    'model': model_name,   # 'mistral' # "starcoder2"
}
vn = MyVanna(config=config)
```

In [5]:
```
hostname = os.uname().nodename
print("Hostname:", hostname)
```

```
Hostname: ducklover1
```

In [6]:

```
file_db = os.path.abspath(os.path.expanduser(file_db))
vn.connect_to_sqlite(file_db)
```

In [7]:

```
vn.run_sql_is_set
```

Out[7]:

```
True
```

In [8]:

```
def remove_collections(collection_name=None, ACCEPTED_TYPES = ["sql", "ddl", "documentation"]):
    if not collection_name:
        collections = ACCEPTED_TYPES
    elif isinstance(collection_name, str):
        collections = [collection_name]
    elif isinstance(collection_name, list):
        collections = collection_name
    else:
        print(f"\t{collection_name} is unknown: Skipped")
        return

    for c in collections:
        if not c in ACCEPTED_TYPES:
            print(f"\t{c} is unknown: Skipped")
            continue

        # print(f"vn.remove_collection('{c}')")
        vn.remove_collection(c)
```

In [9]:

```
def strip_brackets(ddl):
    """
    This function removes square brackets from table and column names in a DDL script.

    Args:
        ddl (str): The DDL script containing square brackets.

    Returns:
        str: The DDL script with square brackets removed.
    """
    # Use regular expressions to match and replace square brackets
    pattern = r"\[([^\]]+)]"  # Match any character except ] within square brackets
    return re.sub(pattern, r"\1", ddl)
```

In [10]:

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

Training

SQLite sample database

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

In [11]:

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

In [12]:

```
df_ddl
```

Out[12]:

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

In [13]:

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

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

```
Adding ddl: CREATE TABLE "albums"
(
    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    Title NVARCHAR(160)  NOT NULL,
    ArtistId INTEGER  NOT NULL,
    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE sqlite_sequence(name,seq)
Adding ddl: CREATE TABLE "artists"
(
    ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    Name NVARCHAR(120)
)
Adding ddl: CREATE TABLE "customers"
(
    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    FirstName NVARCHAR(40)  NOT NULL,
    LastName NVARCHAR(20)  NOT NULL,
    Company NVARCHAR(80),
    Address NVARCHAR(70),
    City NVARCHAR(40),
    State NVARCHAR(40),
    Country NVARCHAR(40),
    PostalCode NVARCHAR(10),
    Phone NVARCHAR(24),
    Fax NVARCHAR(24),
    Email NVARCHAR(60)  NOT NULL,
    SupportRepId INTEGER,
    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE "employees"
(
    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    LastName NVARCHAR(20)  NOT NULL,
    FirstName NVARCHAR(20)  NOT NULL,
    Title NVARCHAR(30),
    ReportsTo INTEGER,
    BirthDate DATETIME,
    HireDate DATETIME,
    Address NVARCHAR(70),
    City NVARCHAR(40),
    State NVARCHAR(40),
    Country NVARCHAR(40),
    PostalCode NVARCHAR(10),
    Phone NVARCHAR(24),
    Fax NVARCHAR(24),
    Email NVARCHAR(60),
    FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE "genres"
(
    GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    Name NVARCHAR(120)
)
Adding ddl: CREATE TABLE "invoices"
(
    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    CustomerId INTEGER  NOT NULL,
    InvoiceDate DATETIME  NOT NULL,
    BillingAddress NVARCHAR(70),
    BillingCity NVARCHAR(40),
    BillingState NVARCHAR(40),
    BillingCountry NVARCHAR(40),
    BillingPostalCode NVARCHAR(10),
    Total NUMERIC(10,2)  NOT NULL,
    FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE "invoice_items"
(
    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    InvoiceId INTEGER  NOT NULL,
    TrackId INTEGER  NOT NULL,
    UnitPrice NUMERIC(10,2)  NOT NULL,
    Quantity INTEGER  NOT NULL,
    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId)
                ON DELETE NO ACTION ON UPDATE NO ACTION,
    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE "media_types"
```

```
(
    MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    Name NVARCHAR(120)
)
Adding ddl: CREATE TABLE "playlists"
(
    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    Name NVARCHAR(120)
)
Adding ddl: CREATE TABLE "playlist_track"
(
    PlaylistId INTEGER  NOT NULL,
    TrackId INTEGER  NOT NULL,
    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),
    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId)
                ON DELETE NO ACTION ON UPDATE NO ACTION,
    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE TABLE "tracks"
(
    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    Name NVARCHAR(200)  NOT NULL,
    AlbumId INTEGER,
    MediaTypeId INTEGER  NOT NULL,
    GenreId INTEGER,
    Composer NVARCHAR(220),
    Milliseconds INTEGER  NOT NULL,
    Bytes INTEGER,
    UnitPrice NUMERIC(10,2)  NOT NULL,
    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId)
                ON DELETE NO ACTION ON UPDATE NO ACTION,
    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId)
                ON DELETE NO ACTION ON UPDATE NO ACTION,
    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId)
                ON DELETE NO ACTION ON UPDATE NO ACTION
)
Adding ddl: CREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)
Adding ddl: CREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)
Adding ddl: CREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)
Adding ddl: CREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)
Adding ddl: CREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)
Adding ddl: CREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)
Adding ddl: CREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)
Adding ddl: CREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)
Adding ddl: CREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)
Adding ddl: CREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)
Adding ddl: CREATE TABLE sqlite_stat1(tbl,idx,stat)
Adding documentation....
```

In [14]:

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

Out[14]:

	id	question	content	training_data_type
0	039f9d54-59f7-5f29-8c04-14dbc3e95671-ddl	None	CREATE TABLE "artists"\r\n(\r\n ArtistId IN...	ddl
1	0db84e3d-ef41-563c-803e-21c1b985dc19-ddl	None	CREATE TABLE "invoices"\r\n(\r\n InvoiceId ...	ddl
2	10cba811-ddba-5042-9e90-d764dfcd1629-ddl	None	CREATE INDEX IFK_InvoiceCustomerId ON "invoice...	ddl
3	2c711317-b93d-5f60-a728-cb1c6fcbc040-ddl	None	CREATE INDEX IFK_CustomerSupportRepId ON "cust...	ddl
4	37319c81-65f7-50ee-956b-795de244bee5-ddl	None	CREATE TABLE sqlite_stat1(tbl,idx,stat)	ddl
5	40bd77cd-e1de-5872-8693-624117ff413c-ddl	None	CREATE INDEX IFK_InvoiceLineInvoiceId ON "invo...	ddl
6	41130543-7164-562a-90a7-0fd0a409c154-ddl	None	CREATE TABLE "albums"\r\n(\r\n AlbumId INTE...	ddl
7	458debc8-8082-5450-a17a-66028bd55ace-ddl	None	CREATE TABLE "playlists"\r\n(\r\n PlaylistI...	ddl
8	4815f3fd-925b-53ce-9dfa-0e4285d5abd3-ddl	None	CREATE TABLE "invoice_items"\r\n(\r\n Invoi...	ddl
9	48d484e9-984c-58ff-b391-75521c69d486-ddl	None	CREATE INDEX IFK_PlaylistTrackTrackId ON "play...	ddl
10	551e1120-a6ee-554f-8b8a-ccf4f22d3636-ddl	None	CREATE INDEX IFK_AlbumArtistId ON "albums" (Ar...	ddl
11	5ff4911e-45c1-5a59-9566-243a9b6a3320-ddl	None	CREATE TABLE "employees"\r\n(\r\n EmployeeI...	ddl
12	65df0648-bf05-5f75-9365-c21f54b2302d-ddl	None	CREATE TABLE "media_types"\r\n(\r\n MediaTy...	ddl
13	6b585176-e66d-5b23-8d86-ca8a80e3af3d-ddl	None	CREATE INDEX IFK_EmployeeReportsTo ON "employe...	ddl
14	868758b8-e018-55e7-8cc3-75c0e6d211c8-ddl	None	CREATE INDEX IFK_TrackAlbumId ON "tracks" (Alb...	ddl
15	9ea4613d-c1be-5a77-ada9-c54ee3f0cab7-ddl	None	CREATE INDEX IFK_TrackMediaTypeId ON "tracks" ...	ddl
16	a9c9a852-608d-5ef2-aede-26ba098d83d1-	None	CREATE INDEX IFK_TrackGenreId ON "tracks" (Gen...	ddl

	id	question	content	training_data_type
	ddl			
17	b42cc9e1-9219-5a42-9a06-de906f76239e-ddl	None	CREATE TABLE "tracks"\r\n(\r\n TrackId INTE...	ddl
18	c387b9d2-5ff4-5a07-8364-f5dab45bb2a9-ddl	None	CREATE TABLE "genres"\r\n(\r\n GenreId INTE...	ddl
19	d654f328-dc36-549e-84c3-06ee0db7e0f7-ddl	None	CREATE TABLE "playlist_track"\r\n(\r\n Play...	ddl
20	d93f0d68-023d-5afb-8121-ba346699d318-ddl	None	CREATE TABLE "customers"\r\n(\r\n CustomerI...	ddl
21	e5879308-329e-543f-a693-0c14e2f9972e-ddl	None	CREATE INDEX IFK_InvoiceLineTrackId ON "invoic...	ddl
22	ea84418b-1a28-59b4-a1f4-2fb674208adc-ddl	None	CREATE TABLE sqlite_sequence(name,seq)	ddl
0	2b4dda0a-a6ac-5e34-8f76-e41c0734d55e-doc	None	In the chinook database invoice means order	documentation

Asking the AI

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

In [15]:

```
ts_start = time()
```

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

In [16]:

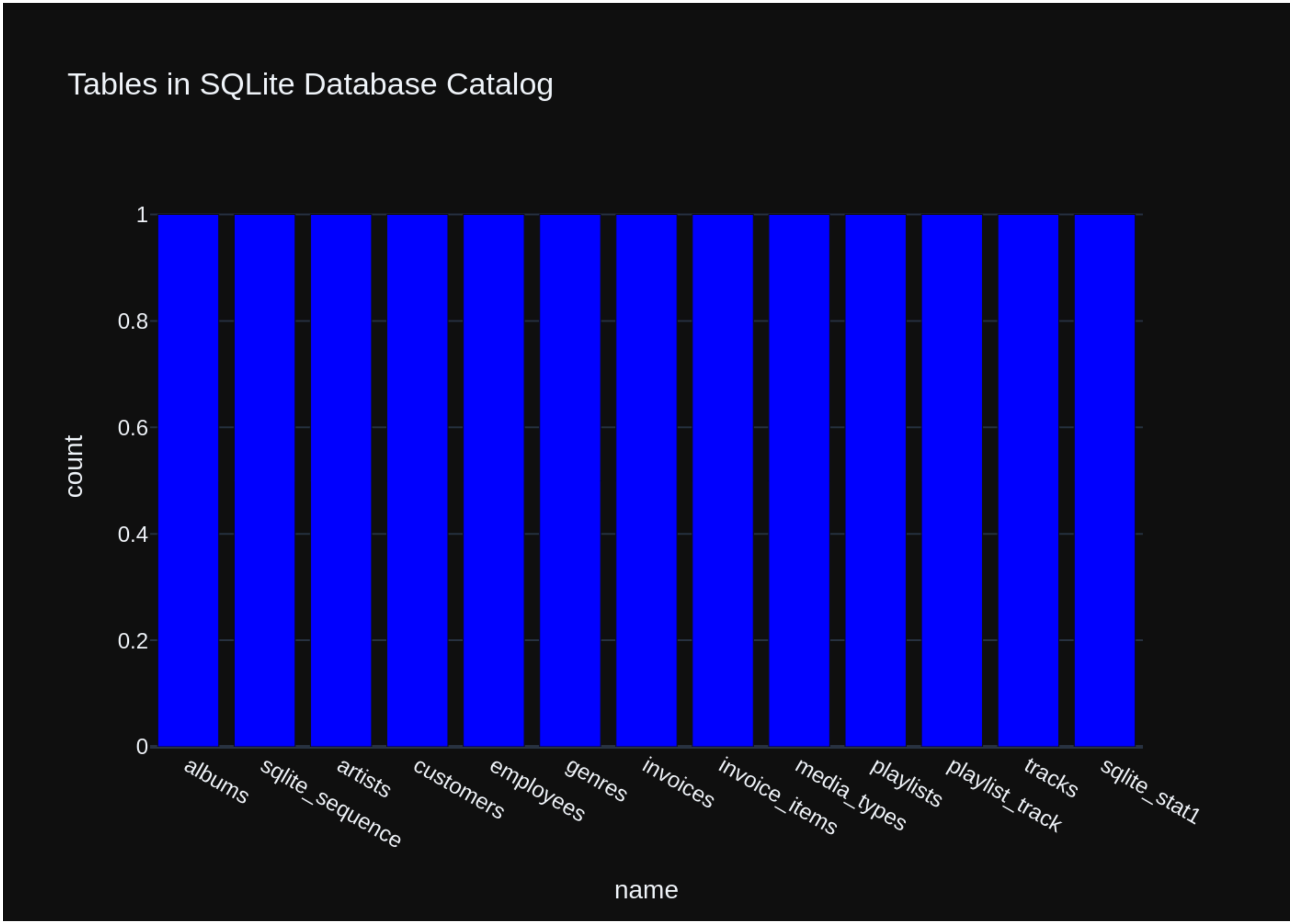
```
vn.ask(question="Can you list all tables in the SQLite database catalog?")
```

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

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE TABLE sqlite_sequence(na
me,seq)\n\nCREATE TABLE "playlists"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "genres"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT
NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOI
NCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER
NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n
Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums"
(AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres"
(GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "med
ia_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "media_type
s"\r\n(\r\n    MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCR
EATE TABLE "artists"\r\n(\r\n    ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR
(120)\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT
NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)
NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (Track
Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n    Playl
istId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY
(PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELET
E NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELE
TE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTO
INCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN
KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n==
=Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the p
rovided 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 ask
ed and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'co
ntent': 'Can you list all tables in the SQLite database catalog?'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE TABLE sqlite_sequence(na
me,seq)\n\nCREATE TABLE \"playlists\"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r
```

~~~
\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOINCREM
ENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY
KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId
INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT
NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENC
ES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFER
ENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeI
d) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE
TABLE \"media_types\"\r\n(\r\n    MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARC
HAR(120)\r\n)\n\nCREATE TABLE \"artists\"\r\n(\r\n    ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n    InvoiceLineId INTEGER PRIMA
RY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n
UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERE
NCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId)
REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"play
list_track\"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT
PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES \"playlists
\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES
\"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r
\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    Arti
stId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE N
O ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means orde
r\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query w
ithout any explanations for the question. \n2. If the provided context is almost sufficient but requires kn
owledge of a specific string in a particular column, please generate an intermediate SQL query to find the
distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provi
ded context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant ta
ble(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was
given before. \n"}, {"role": "user", "content": "Can you list all tables in the SQLite database catalog?"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:29:00.073933367Z', 'message': {'role': 'assistan
t', 'content': "```sql\nSELECT name FROM sqlite_master\nWHERE type = 'table';\n```"}, 'done_reason': 'sto
p', 'done': True, 'total_duration': 33314761756, 'load_duration': 2937374149, 'prompt_eval_count': 866, 'pr
ompt_eval_duration': 26657277000, 'eval_count': 19, 'eval_duration': 3674329000}
```sql
SELECT name FROM sqlite_master
WHERE type = 'table';
~~~
Output from LLM: ```sql
SELECT name FROM sqlite_master
WHERE type = 'table';
```
```

```
Extracted SQL: SELECT name FROM sqlite_master
WHERE type = 'table'
SELECT name FROM sqlite_master
WHERE type = 'table'
               name
0            albums
1   sqlite_sequence
2           artists
3         customers
4         employees
5            genres
6          invoices
7     invoice_items
8       media_types
9         playlists
10   playlist_track
11           tracks
12     sqlite_stat1
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: 'Can you list all tables in the SQLite database catalog?'\n\nThe
DataFrame was produced using this query: SELECT name FROM sqlite_master\nWHERE type = 'table'\n\nThe follow
ing is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n name    object\n
dtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results
of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the d
ataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the
code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:29:15.319116315Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='name', title='Tables in SQLit
e Database Catalog')\n\nfig.update_traces(marker_color='blue')\n\nfig.show()\n```"}, 'done_reason': 'stop',
'done': True, 'total_duration': 15213795324, 'load_duration': 44029894, 'prompt_eval_count': 158, 'prompt_e
val_duration': 5390176000, 'eval_count': 51, 'eval_duration': 9734817000}
```


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

Out[16]:
```
("SELECT name FROM sqlite_master\nWHERE type = 'table'",
                 name
 0             albums
 1    sqlite_sequence
 2            artists
 3          customers
 4          employees
 5             genres
 6           invoices
 7      invoice_items
 8        media_types
 9          playlists
 10    playlist_track
 11            tracks
 12      sqlite_stat1,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'name=%{x}<br>count=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': 'blue', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['albums', 'sqlite_sequence', 'artists', 'customers', 'employees',
                           'genres', 'invoices', 'invoice_items', 'media_types', 'playlists',
                           'playlist_track', 'tracks', 'sqlite_stat1'], dtype=object),
               'xaxis': 'x',
               'y': array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Tables in SQLite Database Catalog'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'name'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'count'}}}
 }))
```

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

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

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingA
ddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCount
ry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIG
N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r
\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\t
ON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t
\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER P
RIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  N
OT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State N
VARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n
Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (Sup
portRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREA
TE TABLE "employees"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVA
RCHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo I
NTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARC
HAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Pho
ne NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFEREN
CES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite_se
quence(name,seq)\n\nCREATE TABLE "playlists"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT N
ULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE TABLE "albums"\r\n
(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    A
rtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE
NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT NUL
L,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackI
d),\r\n    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDA
TE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPD
ATE NO ACTION\r\n)\n\nCREATE TABLE "media_types"\r\n(\r\n    MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT
NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice mea
ns order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL
query without any explanations for the question. \n2. If the provided context is almost sufficient but requ
ires knowledge of a specific string in a particular column, please generate an intermediate SQL query to fi
nd the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If th
e provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most rel
evant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly a
s it was given before. \n'}, {'role': 'user', 'content': 'Can you list all tables in the SQLite database ca
talog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {'rol
e': 'user', 'content': "which table stores customer's orders"}]
```

```
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    Billin
gAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCou
ntry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FORE
IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NO
T NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (Tra
ckId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n    Custom
erId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName N
VARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(4
0),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NV
ARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n
FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO A
CTION\r\n)\n\nCREATE TABLE \"employees\"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(3
0),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR
(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode N
VARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN
KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE \"playlists\"\r\n(\r\n    PlaylistId INTEGER PRI
MARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)
\n\nCREATE TABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NV
ARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES \"artist
s\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n
(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack
PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"media_types\"\r\n(\r\n    MediaTy
peId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n\n===Additional Context
\n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is s
ufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provide
d context is almost sufficient but requires knowledge of a specific string in a particular column, please g
enerate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a com
ment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be g
enerated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered befo
re, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "Can you lis
t all tables in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_m
aster\nWHERE type = 'table'"}, {"role": "user", "content": "which table stores customer's orders"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:29:58.05870737Z', 'message': {'role': 'assistan
t', 'content': 'invoices'}, 'done_reason': 'stop', 'done': True, 'total_duration': 42325721874, 'load_durat
ion': 588738, 'prompt_eval_count': 1136, 'prompt_eval_duration': 40729679000, 'eval_count': 8, 'eval_durati
on': 1459372000}
invoices
invoices
Couldn't run sql:  Execution failed on sql 'invoices': near "invoices": syntax error
```

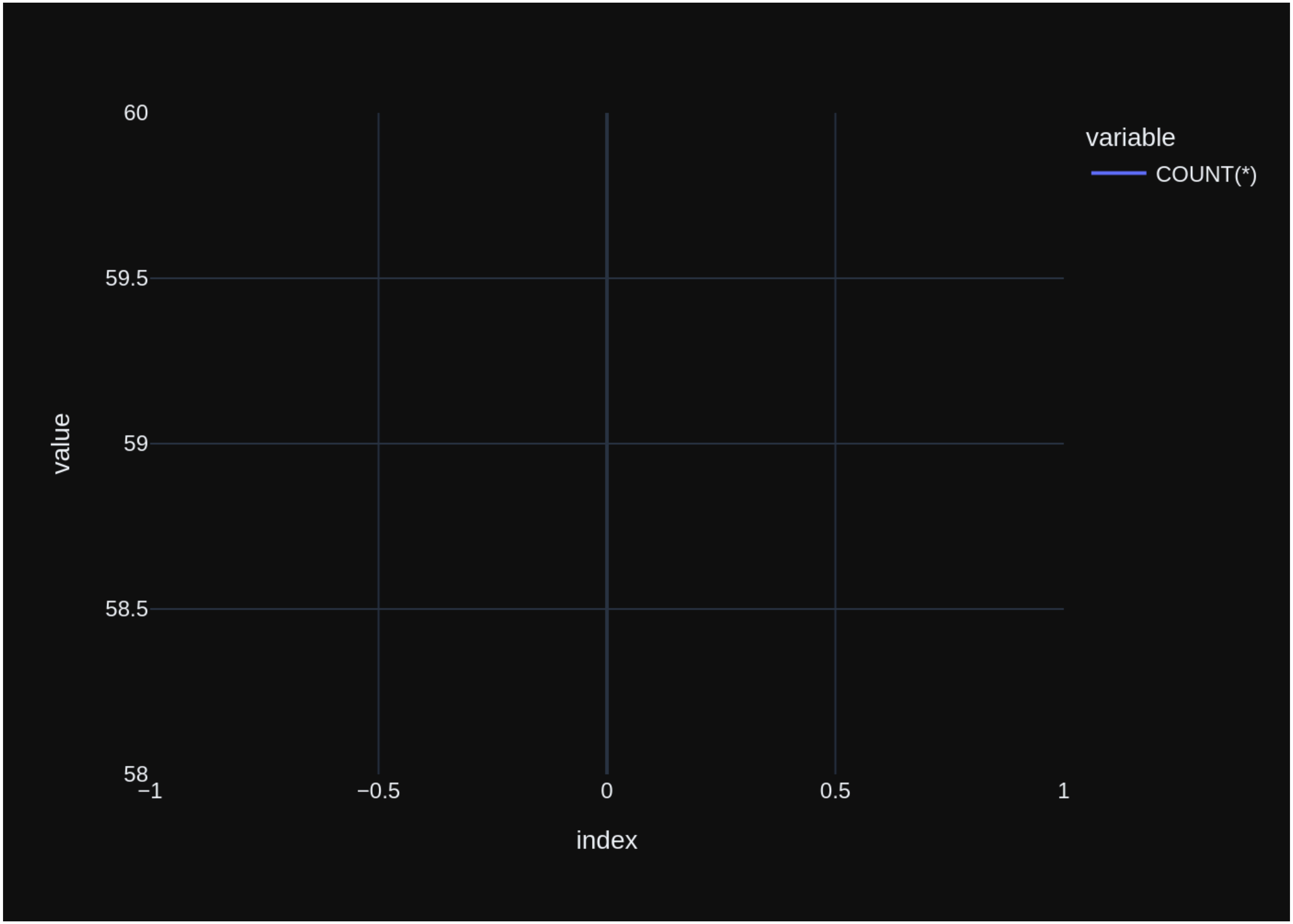
In [18]:

```
vn.ask(question="How many customers are there")
```

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

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingA
ddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCount
ry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIG
N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "customers"\r\n(\r
\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n
LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVAR
CHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Ph
one NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGE
R,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPD
ATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE TABLE "invoic
e_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NO
T NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER
NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON U
PDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON
UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE TAB
LE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT
NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n
\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items"
(TrackId)\n\nCREATE TABLE "employees"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r
\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(3
0),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR
(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode N
VARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN
KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n
\nCREATE TABLE "playlists"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name N
VARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response
Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explan
ations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a spec
ific string in a particular column, please generate an intermediate SQL query to find the distinct strings
in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is in
sufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If
the question has been asked and answered before, please repeat the answer exactly as it was given before.
\n'}, {'role': 'user', 'content': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'ass
istant', 'content': "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {'role': 'user', 'content': 'H
ow many customers are there'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    Billin
gAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCou
ntry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FORE
IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"customers
\"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NU
LL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n
City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(1
0),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    Support
RepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREAT
E TABLE \"invoice_items\"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Invo
iceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n
Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON D
ELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON \"invoice_items\"
(InvoiceId)\n\nCREATE TABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES
\"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLine
TrackId ON \"invoice_items\" (TrackId)\n\nCREATE TABLE \"employees\"\r\n(\r\n    EmployeeId INTEGER PRIMARY
KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NUL
L,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIM
E,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHA
R(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVA
RCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlists\"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCRE
MENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoic
e means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid
SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but
requires knowledge of a specific string in a particular column, please generate an intermediate SQL query t
o find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. I
f the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most
relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactl
y as it was given before. \n"}, {"role": "user", "content": "Can you list all tables in the SQLite database
catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {"rol
e": "user", "content": "How many customers are there"}]
Ollama Response:
```

```
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:30:32.046521154Z', 'message': {'role': 'assistan
t', 'content': 'SELECT COUNT(*) FROM customers'}, 'done_reason': 'stop', 'done': True, 'total_duration': 33
944019976, 'load_duration': 790749, 'prompt_eval_count': 870, 'prompt_eval_duration': 31543443000, 'eval_co
unt': 12, 'eval_duration': 2266579000}
SELECT COUNT(*) FROM customers
SELECT COUNT(*) FROM customers
   COUNT(*)
0        59
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: 'How many customers are there'\n\nThe DataFrame was produced usin
g this query: SELECT COUNT(*) FROM customers\n\nThe following is information about the resulting pandas Dat
aFrame 'df': \nRunning df.dtypes gives:\n COUNT(*)    int64\ndtype: object"}, {"role": "user", "content":
"Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pan
das dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only
Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:30:45.63727739Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.indicator(\n    df,\n    value='COUNT
(*)',\n    title='Number of Customers'\n)\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total
_duration': 13566794710, 'load_duration': 42813945, 'prompt_eval_count': 148, 'prompt_eval_duration': 51707
95000, 'eval_count': 45, 'eval_duration': 8303309000}
```


60
59.5
59
58.5
58
value
−1
−0.5
0
0.5
1
index
variable
COUNT(*)

Out[18]:

```
('SELECT COUNT(*) FROM customers',
    COUNT(*)
 0        59,
 Figure({
     'data': [{'hovertemplate': 'variable=COUNT(*)<br>index=%{x}<br>value=%{y}<extra></extra>',
               'legendgroup': 'COUNT(*)',
               'line': {'color': '#636efa', 'dash': 'solid'},
               'marker': {'symbol': 'circle'},
               'mode': 'lines',
               'name': 'COUNT(*)',
               'orientation': 'v',
               'showlegend': True,
               'type': 'scatter',
               'x': array([0]),
               'xaxis': 'x',
               'y': array([59]),
               '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 [ ]:

```

```

In [19]:

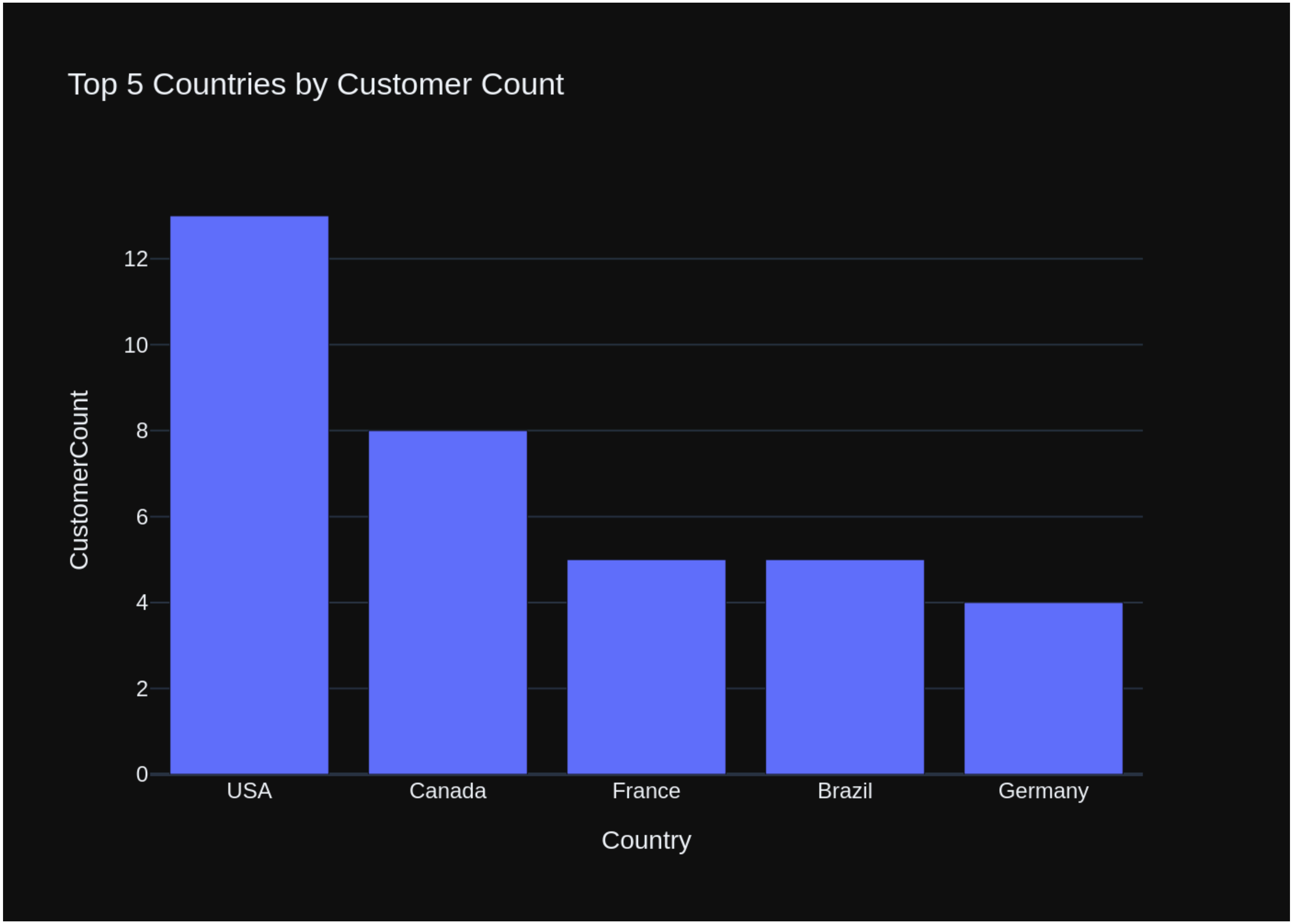
```
vn.ask(question="what are the top 5 countries that customers come from?")
```

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

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingA
ddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCount
ry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIG
N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Firs
tName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    A
ddress NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n
PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT
NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineI
d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  N
OT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (I
nvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN
KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE
TABLE "media_types"\r\n(\r\n    MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHA
R(120)\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "employe
es"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NU
LL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    Bi
rthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    S
tate NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(2
4),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES "employee
s" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n    A
lbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId IN
TEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n
TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FO
REIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO
N,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI
ON\r\n)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIM
ARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTy
peId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER
NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFE
RENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REF
ERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeI
d) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Addi
tional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provid
ed context is sufficient, please generate a valid SQL query without any explanations for the question. \n2.
If the provided context is almost sufficient but requires knowledge of a specific string in a particular co
lumn, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the qu
```

```
ery with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why
it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and
answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content':
'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'rol
e': 'user', 'content': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'assistant', 'c
ontent': "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {'role': 'user', 'content': 'what are the
top 5 countries that customers come from?'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    Billin
gAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCou
ntry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FORE
IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n
Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n
PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT
NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n    InvoiceL
ineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGE
R  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KE
Y (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    F
OREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n
\nCREATE TABLE \"media_types\"\r\n(\r\n    MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    N
ame NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE T
ABLE \"employees\"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARC
HAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INT
EGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHA
R(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone
NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES
\"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums
\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r
\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n    PlaylistId INTEG
ER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistI
d, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACT
ION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE \"tracks\"\r\n(\r
\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    Album
Id INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r
\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n
FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n
FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n
FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guide
lines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanation
s for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific
string in a particular column, please generate an intermediate SQL query to find the distinct strings in th
at column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insuffi
cient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the qu
estion has been asked and answered before, please repeat the answer exactly as it was given before. \n"},
{"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT
(*) FROM customers"}, {"role": "user", "content": "Can you list all tables in the SQLite database catalo
g?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {"role": "u
ser", "content": "what are the top 5 countries that customers come from?"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:31:40.028753571Z', 'message': {'role': 'assistan
t', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY Custo
merCount DESC\nLIMIT 5'}, 'done_reason': 'stop', 'done': True, 'total_duration': 54278295933, 'load_duratio
n': 650080, 'prompt_eval_count': 1338, 'prompt_eval_duration': 47571260000, 'eval_count': 32, 'eval_duratio
n': 6521851000}
SELECT Country, COUNT(*) AS CustomerCount
FROM customers
GROUP BY Country
ORDER BY CustomerCount DESC
LIMIT 5
SELECT Country, COUNT(*) AS CustomerCount
FROM customers
GROUP BY Country
ORDER BY CustomerCount DESC
LIMIT 5
   Country  CustomerCount
0      USA             13
1   Canada              8
2   France              5
3   Brazil              5
4  Germany              4
Ollama parameters:
```

```
model=codegemma: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 CustomerCount\nFROM customers\nGROUP BY
Country\nORDER BY CustomerCount DESC\nLIMIT 5\n\nThe following is information about the resulting pandas Da
taFrame 'df': \nRunning df.dtypes gives:\n Country          object\nCustomerCount     int64\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': 'codegemma:latest', 'created_at': '2024-06-15T19:31:54.7285304Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Country', y='CustomerCount',
title='Top 5 Countries by Customer Count')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'tota
l_duration': 14673241032, 'load_duration': 41983568, 'prompt_eval_count': 178, 'prompt_eval_duration': 6202
713000, 'eval_count': 45, 'eval_duration': 8339776000}
```


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

Out[19]:

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

More SQL questions

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

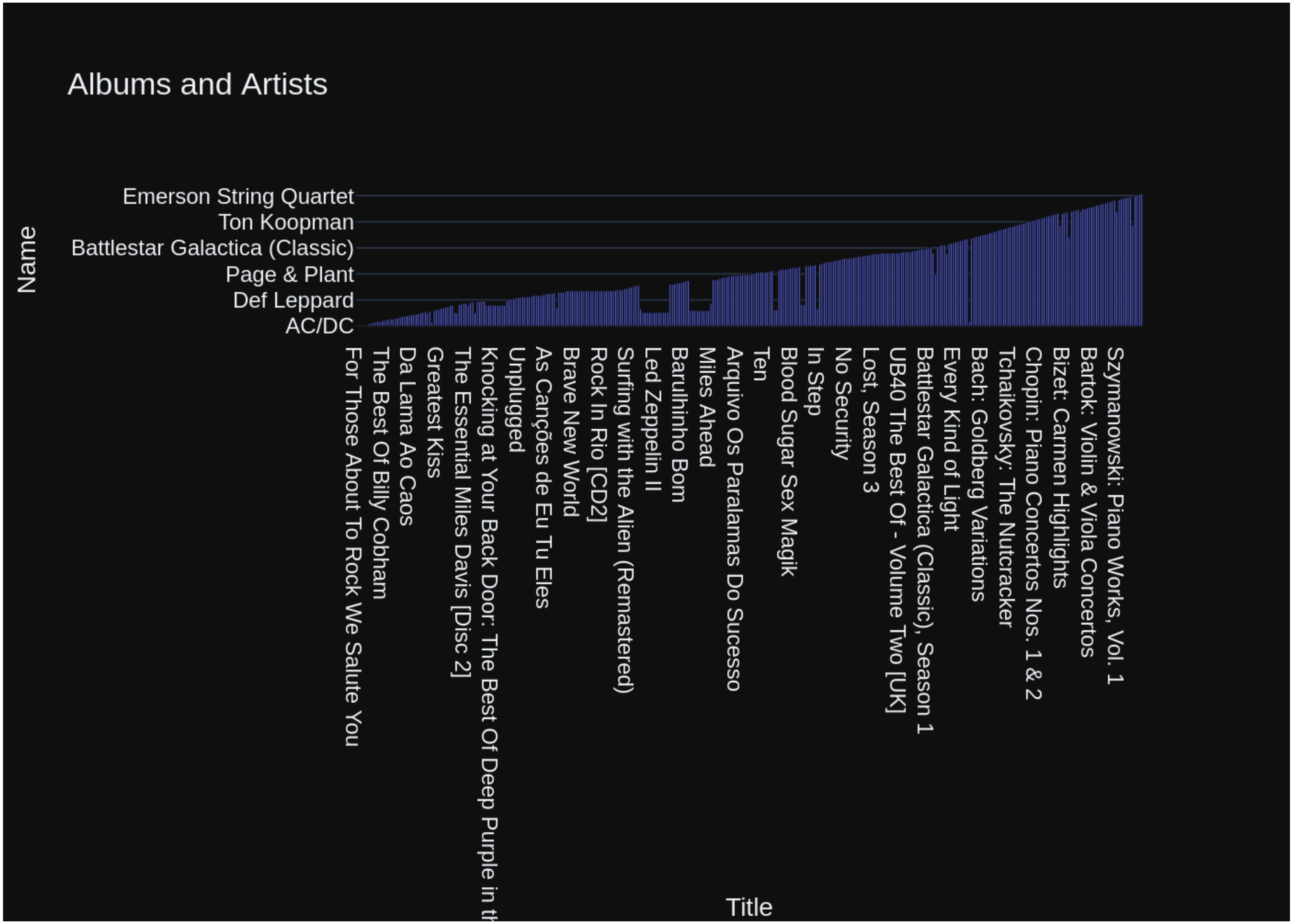
In [20]:

```
question = """
    List all albums and their corresponding artist names
"""

vn.ask(question=question)
```

```
Number of requested results 10 is greater than number of elements in index 3, updating n_results = 3
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "alb
ums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NUL
L,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t
\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY
KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId
INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT
NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENC
ES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFEREN
CES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) RE
FERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX I
FK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n    ArtistId INTEGER PRIMARY KEY AU
TOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)
\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE TABLE "playlists"\r\n(\r
\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE
"genres"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n
CREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\n\n===Additional Context \n\nIn the chinook d
atabase invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please ge
nerate a valid SQL query without any explanations for the question. \n2. If the provided context is almost
sufficient but requires knowledge of a specific string in a particular column, please generate an intermedi
ate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermed
iate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Plea
se use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat t
he answer exactly as it was given before. \n'}, {'role': 'user', 'content': 'Can you list all tables in the
SQLite database catalog?'}, {'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 CustomerCount\nFROM customers\nGROUP BY Country\nORDER
BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'as
sistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': '  \n    List all album
s and their corresponding artist names  \n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE
\"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT
NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER P
RIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"artists\"\r\n(\r\n    ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "Can you list all tables in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist names  \n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:32:30.671916716Z', 'message': {'role': 'assistant', 'content': 'SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId = artists.ArtistId'}, 'done_reason': 'stop', 'done': True, 'total_duration': 35851380592, 'load_duration': 620650, 'prompt_eval_count': 832, 'prompt_eval_duration': 29573828000, 'eval_count': 31, 'eval_duration': 6050501000}
SELECT albums.Title, artists.Name
FROM albums
JOIN artists ON albums.ArtistId = artists.ArtistId
SELECT albums.Title, artists.Name
FROM albums
JOIN artists ON albums.ArtistId = artists.ArtistId
                                            Title  \
0                 For Those About To Rock We Salute You
1                                     Balls to the Wall
2                                     Restless and Wild
3                                     Let There Be Rock
4                                              Big Ones
```

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

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

[347 rows x 2 columns]
Ollama parameters:
model=codegemma: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 albums.Title, artists.Name\nFROM albums\nJOIN ar
tists ON albums.ArtistId = artists.ArtistId\n\nThe following is information about the resulting pandas Data
Frame 'df': \nRunning df.dtypes gives:\n Title    object\nName     object\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': 'codegemma:latest', 'created_at': '2024-06-15T19:32:44.252424398Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Title', y='Name', title='Albu
ms and Artists')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 13553433825,
'load_duration': 42266386, 'prompt_eval_count': 176, 'prompt_eval_duration': 6105820000, 'eval_count': 40,
'eval_duration': 7354987000}
```


Albums and Artists
Name
Emerson String Quartet
Ton Koopman
Battlestar Galactica (Classic)
Page & Plant
Def Leppard
AC/DC
For Those About To Rock We Salute You
The Best Of Billy Cobham
Da Lama Ao Caos
Greatest Kiss
The Essential Miles Davis [Disc 2]
Knocking at Your Back Door: The Best Of Deep Purple in th
Unplugged
As Canções de Eu Tu Eles
Brave New World
Rock In Rio [CD2]
Surfing with the Alien (Remastered)
Led Zeppelin II
Barulhinho Bom
Miles Ahead
Arquivo Os Paralamas Do Sucesso
Ten
Blood Sugar Sex Magik
In Step
No Security
Lost, Season 3
UB40 The Best Of - Volume Two [UK]
Battlestar Galactica (Classic), Season 1
Every Kind of Light
Bach: Goldberg Variations
Tchaikovsky: The Nutcracker
Chopin: Piano Concertos Nos. 1 & 2
Bizet: Carmen Highlights
Bartok: Violin & Viola Concertos
Szymanowski: Piano Works, Vol. 1
Title

```
Out[20]: ('SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId = artists.ArtistId',
                                                           Title  \
          0                For Those About To Rock We Salute You   
          1                                    Balls to the Wall   
          2                                    Restless and Wild   
          3                                    Let There Be Rock   
          4                                             Big Ones   
          ..                                                 ...   
          342                             Respighi:Pines of Rome   
          343  Schubert: The Late String Quartets & String Qu...   
          344                                Monteverdi: L'Orfeo   
          345                              Mozart: Chamber Music   
          346  Koyaanisqatsi (Soundtrack from the Motion Pict...   

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

          [347 rows x 2 columns],
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'Title=%{x}<br>Name=%{y}<extra></extra>',
                        'legendgroup': '',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array(['For Those About To Rock We Salute You', 'Balls to the Wall',
                                    'Restless and Wild', ..., "Monteverdi: L'Orfeo",
                                    'Mozart: Chamber Music',
                                    'Koyaanisqatsi (Soundtrack from the Motion Picture)'], dtype=object),
                        'xaxis': 'x',
                        'y': array(['AC/DC', 'Accept', 'Accept', ...,
                                    'C. Monteverdi, Nigel Rogers - Chiaroscuro; London Baroque; London Cornett & Sa
         ckbu',
                                    'Nash Ensemble', 'Philip Glass Ensemble'], dtype=object),
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                         'legend': {'tracegroupgap': 0},
                         'template': '...',
                         'title': {'text': 'Albums and Artists'},
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Title'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}}}
          }))
```

In [21]:

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

vn.ask(question=question)
```

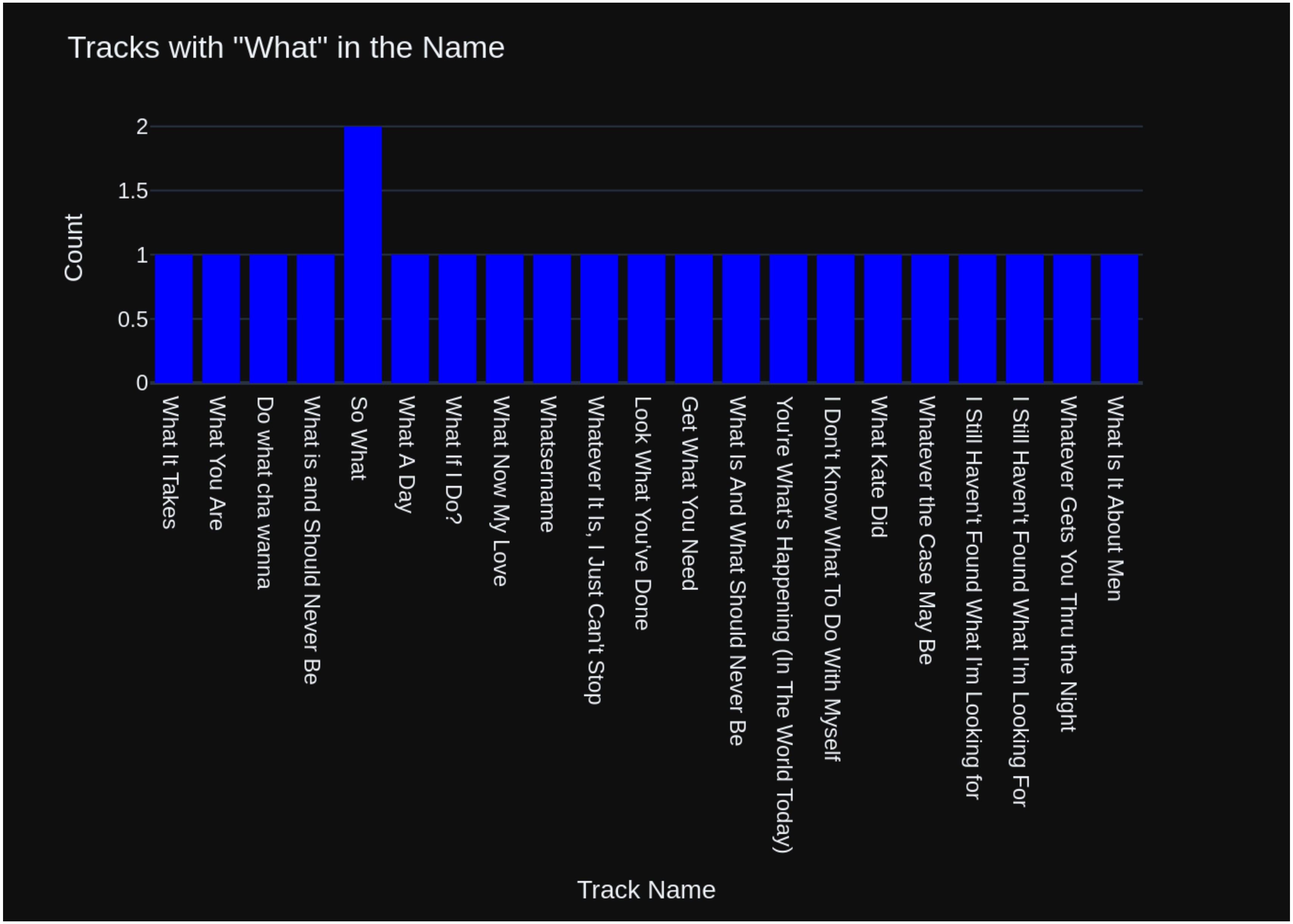
```
Number of requested results 10 is greater than number of elements in index 4, updating n_results = 4
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK_Pl
aylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMA
RY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTyp
eId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER 
NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFE
RENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REF
ERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeI
d) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IN
DEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n
\nCREATE TABLE "playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NUL
L,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) R
EFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Tra
ckId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IF
K_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)
\n\nCREATE TABLE "playlists"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name
NVARCHAR(120)\r\n)\n\nCREATE TABLE "genres"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means orde
r\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query w
ithout any explanations for the question. \n2. If the provided context is almost sufficient but requires kn
owledge of a specific string in a particular column, please generate an intermediate SQL query to find the
distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provi
ded context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant t
able(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it wa
s given before. \n'}, {'role': 'user', 'content': '  \n    List all albums and their corresponding artist n
ames  \n'}, {'role': 'assistant', 'content': 'SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists
ON albums.ArtistId = artists.ArtistId'}, {'role': 'user', 'content': 'Can you list all tables in the SQLite
database catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master\nWHERE type = 'tabl
e'"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assi
stant', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY C
ustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assist
ant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': '  \n    Find all tracks wi
th a name containing "What" (case-insensitive)\n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX IFK_
PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\n\nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    Med
iaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTE
GER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId)
REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreI
d) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Med
iaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n
CREATE INDEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON \"tracks\" (M
ediaTypeId)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INT
EGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY
(PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n
FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE INDEX IFK_AlbumArtistId ON
\"albums\" (ArtistId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n    GenreId INTEGER PRIMARY K
EY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook dat
abase invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please gene
rate a valid SQL query without any explanations for the question. \n2. If the provided context is almost su
fficient but requires knowledge of a specific string in a particular column, please generate an intermediat
e SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermedia
te_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please
use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the
answer exactly as it was given before. \n"}, {"role": "user", "content": "  \n    List all albums and their
corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT albums.Title, artists.Name\nFROM
albums\nJOIN artists ON albums.ArtistId = artists.ArtistId"}, {"role": "user", "content": "Can you list all
tables in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master
\nWHERE type = 'table'"}, {"role": "user", "content": "what are the top 5 countries that customers come fro
m?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY
Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "How many customers are ther
e"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "  \n
Find all tracks with a name containing \"What\" (case-insensitive)\n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:33:20.95116991Z', 'message': {'role': 'assistan
t', 'content': "SELECT * FROM tracks\nWHERE Name LIKE '%What%'"}, 'done_reason': 'stop', 'done': True, 'tot
al_duration': 36456425974, 'load_duration': 573053, 'prompt_eval_count': 928, 'prompt_eval_duration': 32686
393000, 'eval_count': 18, 'eval_duration': 3457499000}
SELECT * FROM tracks
WHERE Name LIKE '%What%'
SELECT * FROM tracks
WHERE Name LIKE '%What%'
    TrackId                                          Name  AlbumId  \
0        26                                 What It Takes        5
```

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

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

```
19            1        1                                                 U2
20            2        9                                               None
21            2        9  Delroy "Chris" Cooper, Donovan Jackson, Earl C...

    Milliseconds      Bytes  UnitPrice
0         310622   10144730       0.99
1         249391    5988186       0.99
2         274155    9018565       0.99
3         260675    8497116       0.99
4         564009   18360449       0.99
5         158275    5203430       0.99
6         302994    9929799       0.99
7         149995    4913383       0.99
8         252316    8244843       0.99
9         247222    8249453       0.99
10        230974    7517083       0.99
11        247719    8043765       0.99
12        287973    9369385       0.99
13        142027    4631104       0.99
14        189152    6162894       0.99
15        221387    7251478       0.99
16       2610250  484583988       1.99
17       2616410  183867185       1.99
18        353567   11542247       0.99
19        280764    9306737       0.99
20        215084    3499018       0.99
21        209573    3426106       0.99
Ollama parameters:
model=codegemma: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 * FROM tracks\nWHERE Name LIKE '%Wha
t%'\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n
TrackId           int64\nName             object\nAlbumId           int64\nMediaTypeId       int64\nGenreId
int64\nComposer         object\nMilliseconds      int64\nBytes             int64\nUnitPrice       float64\n
dtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results
of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the d
ataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the
code."}]
```

````
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:33:45.172453052Z', 'message': {'role': 'assistan
t', 'content': '```python\nimport plotly.express as px\n\nfig = px.histogram(df, x=\'Name\', nbins=10)\n\nf
ig.update_traces(marker_color=\'blue\')\n\nfig.update_layout(\n    title=\'Tracks with "What" in the Name
\',\n    xaxis_title=\'Track Name\',\n    yaxis_title=\'Count\'\n)\n\nfig.show()\n```'}, 'done_reason': 'st
op', 'done': True, 'total_duration': 24194490396, 'load_duration': 691910, 'prompt_eval_count': 215, 'promp
t_eval_duration': 7421025000, 'eval_count': 88, 'eval_duration': 16636370000}
````


Tracks with "What" in the Name
Count
2
1.5
1
0.5
0
What It Takes
What You Are
Do what cha wanna
What is and Should Never Be
So What
What A Day
What If I Do?
What Now My Love
Whatsername
Whatever It Is, I Just Can't Stop
Look What You've Done
Get What You Need
What Is And What Should Never Be
You're What's Happening (In The World Today)
I Don't Know What To Do With Myself
What Kate Did
Whatever the Case May Be
I Still Haven't Found What I'm Looking for
I Still Haven't Found What I'm Looking For
Whatever Gets You Thru the Night
What Is It About Men
Track Name

```
Out[21]: ("SELECT * FROM tracks\nWHERE 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  ,
Figure({
    'data': [{'alignmentgroup': 'True',
              'bingroup': 'x',
              'hovertemplate': 'Name=%{x}<br>count=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': 'blue', 'pattern': {'shape': ''}},
              'name': '',
              'nbinsx': 10,
              'offsetgroup': '',
              'orientation': 'v',
              'showlegend': False,
              'type': 'histogram',
              'x': array(['What It Takes', 'What You Are', 'Do what cha wanna',
                          'What is and Should Never Be', 'So What', 'What A Day', 'What If I Do?',
                          'What Now My Love', 'Whatsername', "Whatever It Is, I Just Can't Stop",
                          "Look What You've Done", 'Get What You Need',
                          'What Is And What Should Never Be',
                          "You're What's Happening (In The World Today)", 'So What',
                          "I Don't Know What To Do With Myself", 'What Kate Did',
                          'Whatever the Case May Be',
                          "I Still Haven't Found What I'm Looking for",
                          "I Still Haven't Found What I'm Looking For",
                          'Whatever Gets You Thru the Night', 'What Is It About Men'],
                         dtype=object),
              'xaxis': 'x',
              'yaxis': 'y'}],
    'layout': {'barmode': 'relative',
               'legend': {'tracegroupgap': 0},
               'margin': {'t': 60},
               'template': '...',
               'title': {'text': 'Tracks with "What" in the Name'},
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Track Name'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Count'}}}
}))
```

In [22]:
```
question = """
    Get the total number of invoices for each customer
"""

vn.ask(question=question)
```

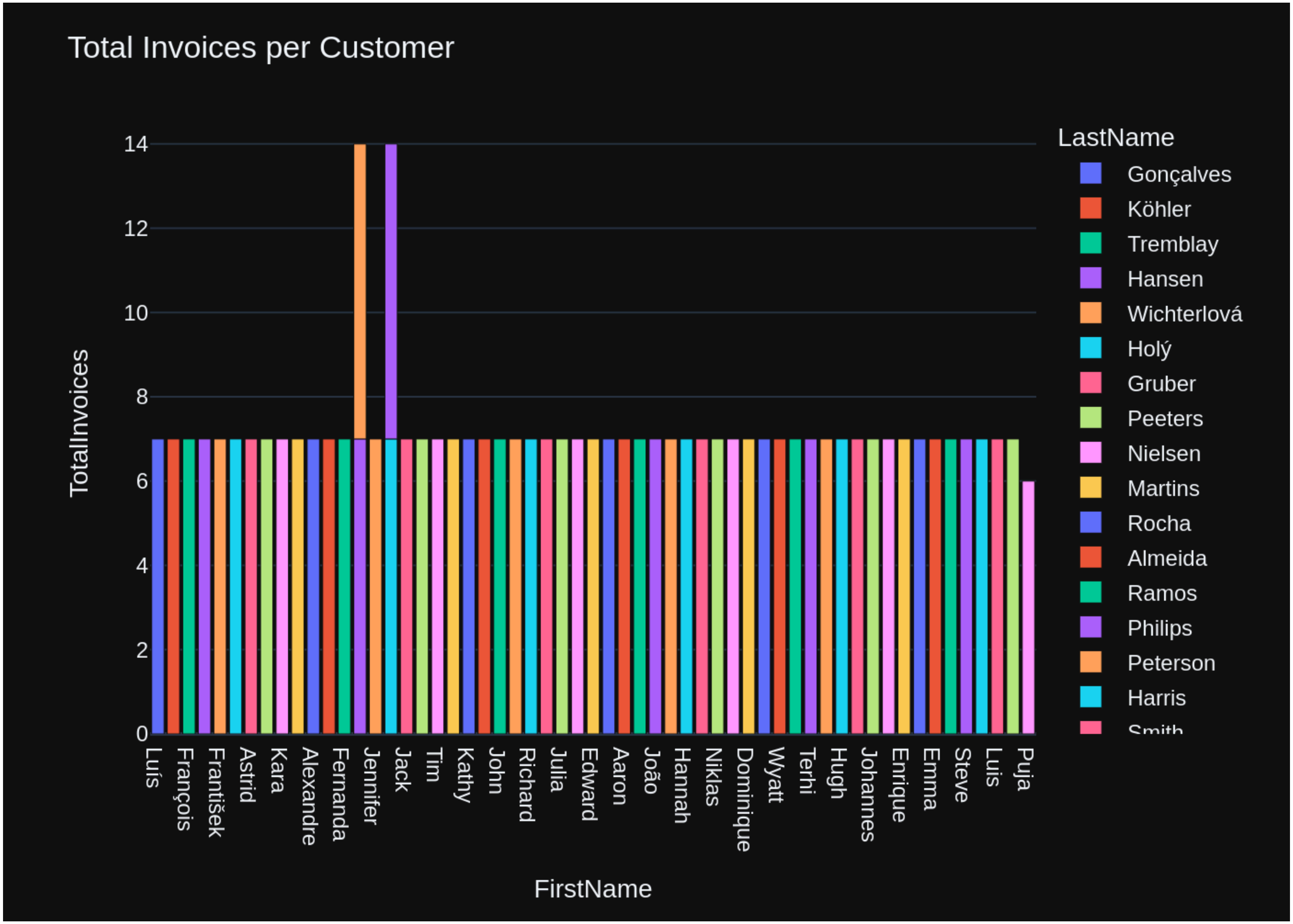
```
Number of requested results 10 is greater than number of elements in index 5, updating n_results = 5
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingA
ddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCount
ry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIG
N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId
ON "invoice_items" (InvoiceId)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KE
Y AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    Unit
Price NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES
"invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERE
NCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLin
eTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KE
Y AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NUL
L,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHA
R(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax N
VARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportR
epId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IN
DEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "employees"\r\n(\r\n    Employee
Id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVA
RCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n
HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n
Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r
\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DEL
ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\n
CREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR
(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n
Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMER
IC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DEL
ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means
order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL que
ry without any explanations for the question. \n2. If the provided context is almost sufficient but require
s knowledge of a specific string in a particular column, please generate an intermediate SQL query to find
the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the p
rovided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most 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': 'How many customers are there'}, {'role': 'assistan
t', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': 'what are the top 5 countries
that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nF
```

```
ROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n
List all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT album
s.Title, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId = artists.ArtistId'}, {'role': 'user',
'content': '  \n    Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assista
nt', 'content': "SELECT * FROM tracks\nWHERE Name LIKE '%What%'"}, {'role': 'user', 'content': 'Can you lis
t all tables in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_m
aster\nWHERE type = 'table'"}, {'role': 'user', 'content': '  \n    Get the total number of invoices for ea
ch customer\n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    Billin
gAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCou
ntry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FORE
IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineInv
oiceId ON \"invoice_items\" (InvoiceId)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n    InvoiceLineId INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r
\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId)
REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Tr
ackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX
IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE TABLE \"customers\"\r\n(\r\n    CustomerId
INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCH
AR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n
State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(2
4),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREI
GN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"employees
\"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NUL
L,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    Bir
thDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    St
ate NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r
\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES \"employees\"
(EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON
\"employees\" (ReportsTo)\n\nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT
NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NUL
L,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Byt
es INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\" (A
lbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres\"
(GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES \"me
dia_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context
\n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is s
ufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provide
d context is almost sufficient but requires knowledge of a specific string in a particular column, please g
enerate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a com
ment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be g
enerated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered befo
re, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "How many cu
stomers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user",
"content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SEL
ECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMI
T 5"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist names  \n"}, {"r
ole": "assistant", "content": "SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists ON albums.Artis
tId = artists.ArtistId"}, {"role": "user", "content": "  \n    Find all tracks with a name containing \"Wha
t\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks\nWHERE Name LIKE '%Wha
t%'"}, {"role": "user", "content": "Can you list all tables in the SQLite database catalog?"}, {"role": "as
sistant", "content": "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {"role": "user", "content": "
\n    Get the total number of invoices for each customer\n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:34:48.091990698Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM cus
tomers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName'},
'done_reason': 'stop', 'done': True, 'total_duration': 62808391740, 'load_duration': 719379, 'prompt_eval_c
ount': 1395, 'prompt_eval_duration': 49920338000, 'eval_count': 60, 'eval_duration': 12489366000}
SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId, c.FirstName, c.LastName
SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId, c.FirstName, c.LastName
    CustomerId  FirstName      LastName  TotalInvoices
0            1       Luís     Gonçalves              7
1            2     Leonie        Köhler              7
2            3   François      Tremblay              7
3            4      Bjørn        Hansen              7
4            5  František   Wichterlová              7
5            6     Helena          Holý              7
```

```
6            7     Astrid        Gruber              7
7            8       Daan       Peeters              7
8            9       Kara       Nielsen              7
9           10    Eduardo       Martins              7
10          11  Alexandre         Rocha              7
11          12    Roberto       Almeida              7
12          13   Fernanda         Ramos              7
13          14       Mark       Philips              7
14          15   Jennifer      Peterson              7
15          16      Frank        Harris              7
16          17       Jack         Smith              7
17          18   Michelle        Brooks              7
18          19        Tim         Goyer              7
19          20        Dan        Miller              7
20          21      Kathy         Chase              7
21          22    Heather       Leacock              7
22          23       John        Gordon              7
23          24      Frank       Ralston              7
24          25     Victor       Stevens              7
25          26    Richard    Cunningham              7
26          27    Patrick          Gray              7
27          28      Julia       Barnett              7
28          29     Robert         Brown              7
29          30     Edward       Francis              7
30          31     Martha          Silk              7
31          32      Aaron      Mitchell              7
32          33      Ellie      Sullivan              7
33          34       João     Fernandes              7
34          35   Madalena       Sampaio              7
35          36     Hannah     Schneider              7
36          37       Fynn    Zimmermann              7
37          38     Niklas      Schröder              7
38          39    Camille       Bernard              7
39          40  Dominique      Lefebvre              7
40          41       Marc        Dubois              7
41          42      Wyatt        Girard              7
42          43   Isabelle       Mercier              7
43          44      Terhi    Hämäläinen              7
44          45   Ladislav        Kovács              7
45          46       Hugh      O'Reilly              7
46          47      Lucas       Mancini              7
47          48   Johannes  Van der Berg              7
```

```
48          49  Stanisław        Wójcik              7
49          50    Enrique         Muñoz              7
50          51     Joakim     Johansson              7
51          52       Emma         Jones              7
52          53       Phil        Hughes              7
53          54      Steve        Murray              7
54          55       Mark        Taylor              7
55          56      Diego     Gutiérrez              7
56          57       Luis         Rojas              7
57          58      Manoj        Pareek              7
58          59       Puja    Srivastava              6
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Get the total number of invoices for each customer\n'\n
\nThe DataFrame was produced using this query: SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.Invoic
eId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Custome
rId, c.FirstName, c.LastName\n\nThe following is information about the resulting pandas DataFrame 'df': \nR
unning df.dtypes gives:\n CustomerId        int64\nFirstName        object\nLastName         object\nTotalI
nvoices     int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to
chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only
one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explana
tions -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:35:06.032439529Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='FirstName', y='TotalInvoice
s', color='LastName')\n\nfig.update_layout(title='Total Invoices per Customer')\n\nfig.show()\n```"}, 'done
_reason': 'stop', 'done': True, 'total_duration': 17914143288, 'load_duration': 717820, 'prompt_eval_coun
t': 219, 'prompt_eval_duration': 7627007000, 'eval_count': 54, 'eval_duration': 10151303000}
```


Total Invoices per Customer
TotalInvoices
0
2
4
6
8
10
12
14
FirstName
Luís
François
František
Astrid
Kara
Alexandre
Fernanda
Jennifer
Jack
Tim
Kathy
John
Richard
Julia
Edward
Aaron
João
Hannah
Niklas
Dominique
Wyatt
Terhi
Hugh
Johannes
Enrique
Emma
Steve
Luis
Puja
LastName
Gonçalves
Köhler
Tremblay
Hansen
Wichterlová
Holý
Gruber
Peeters
Nielsen
Martins
Rocha
Almeida
Ramos
Philips
Peterson
Harris
Smith

```
Out[22]: ('SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOI
N invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName',
     CustomerId  FirstName      LastName  TotalInvoices
 0            1       Luís     Gonçalves              7
 1            2     Leonie        Köhler              7
 2            3   François      Tremblay              7
 3            4      Bjørn        Hansen              7
 4            5  František   Wichterlová              7
 5            6     Helena          Holý              7
 6            7     Astrid        Gruber              7
 7            8       Daan       Peeters              7
 8            9       Kara       Nielsen              7
 9           10    Eduardo       Martins              7
 10          11  Alexandre         Rocha              7
 11          12    Roberto       Almeida              7
 12          13   Fernanda         Ramos              7
 13          14       Mark       Philips              7
 14          15   Jennifer      Peterson              7
 15          16      Frank        Harris              7
 16          17       Jack         Smith              7
 17          18   Michelle        Brooks              7
 18          19        Tim         Goyer              7
 19          20        Dan        Miller              7
 20          21      Kathy         Chase              7
 21          22    Heather       Leacock              7
 22          23       John        Gordon              7
 23          24      Frank       Ralston              7
 24          25     Victor       Stevens              7
 25          26    Richard    Cunningham              7
 26          27    Patrick          Gray              7
 27          28      Julia       Barnett              7
 28          29     Robert         Brown              7
 29          30     Edward       Francis              7
 30          31     Martha          Silk              7
 31          32      Aaron      Mitchell              7
 32          33      Ellie      Sullivan              7
 33          34       João     Fernandes              7
 34          35   Madalena       Sampaio              7
 35          36     Hannah     Schneider              7
 36          37       Fynn    Zimmermann              7
 37          38     Niklas      Schröder              7
 38          39    Camille       Bernard              7
 39          40  Dominique      Lefebvre              7
 40          41       Marc        Dubois              7
 41          42      Wyatt        Girard              7
 42          43   Isabelle       Mercier              7
 43          44      Terhi    Hämäläinen              7
 44          45   Ladislav        Kovács              7
 45          46       Hugh      O'Reilly              7
 46          47      Lucas       Mancini              7
 47          48   Johannes  Van der Berg              7
 48          49  Stanisław        Wójcik              7
 49          50    Enrique         Muñoz              7
 50          51     Joakim     Johansson              7
 51          52       Emma         Jones              7
 52          53       Phil        Hughes              7
 53          54      Steve        Murray              7
 54          55       Mark        Taylor              7
 55          56      Diego     Gutiérrez              7
 56          57       Luis         Rojas              7
 57          58      Manoj        Pareek              7
 58          59       Puja    Srivastava              6,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Gonçalves<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra
>',
               'legendgroup': 'Gonçalves',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': 'Gonçalves',
               'offsetgroup': 'Gonçalves',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Luís'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Köhler<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Köhler',
               'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
               'name': 'Köhler',
               'offsetgroup': 'Köhler',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Leonie'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Tremblay<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra
>',
               'legendgroup': 'Tremblay',
               'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
               'name': 'Tremblay',
               'offsetgroup': 'Tremblay',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['François'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Hansen<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Hansen',
               'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
               'name': 'Hansen',
               'offsetgroup': 'Hansen',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Bjørn'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Wichterlová<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extr
a>',
               'legendgroup': 'Wichterlová',
               'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
               'name': 'Wichterlová',
               'offsetgroup': 'Wichterlová',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['František'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Holý<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Holý',
               'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
               'name': 'Holý',
               'offsetgroup': 'Holý',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Helena'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Gruber<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Gruber',
               'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
               'name': 'Gruber',
               'offsetgroup': 'Gruber',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Astrid'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Peeters<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Peeters',
               'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
               'name': 'Peeters',
               'offsetgroup': 'Peeters',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Daan'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Nielsen<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Nielsen',
               'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
               'name': 'Nielsen',
               'offsetgroup': 'Nielsen',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Kara'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Martins<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Martins',
               'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
               'name': 'Martins',
               'offsetgroup': 'Martins',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Eduardo'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Rocha<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Rocha',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': 'Rocha',
               'offsetgroup': 'Rocha',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Alexandre'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Almeida<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Almeida',
               'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
               'name': 'Almeida',
               'offsetgroup': 'Almeida',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Roberto'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Ramos<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Ramos',
               'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
               'name': 'Ramos',
               'offsetgroup': 'Ramos',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Fernanda'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Philips<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Philips',
               'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
               'name': 'Philips',
               'offsetgroup': 'Philips',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Mark'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Peterson<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra
>',
               'legendgroup': 'Peterson',
               'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
               'name': 'Peterson',
               'offsetgroup': 'Peterson',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Jennifer'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Harris<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Harris',
               'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
               'name': 'Harris',
               'offsetgroup': 'Harris',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Frank'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Smith<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Smith',
               'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
               'name': 'Smith',
               'offsetgroup': 'Smith',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Jack'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Brooks<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Brooks',
               'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
               'name': 'Brooks',
               'offsetgroup': 'Brooks',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Michelle'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Goyer<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Goyer',
               'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
               'name': 'Goyer',
               'offsetgroup': 'Goyer',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Tim'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Miller<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Miller',
               'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
               'name': 'Miller',
               'offsetgroup': 'Miller',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Dan'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Chase<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Chase',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': 'Chase',
               'offsetgroup': 'Chase',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Kathy'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Leacock<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Leacock',
               'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
               'name': 'Leacock',
               'offsetgroup': 'Leacock',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Heather'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Gordon<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Gordon',
               'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
               'name': 'Gordon',
               'offsetgroup': 'Gordon',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['John'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Ralston<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Ralston',
               'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
               'name': 'Ralston',
               'offsetgroup': 'Ralston',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Frank'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Stevens<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Stevens',
               'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
               'name': 'Stevens',
               'offsetgroup': 'Stevens',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Victor'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Cunningham<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra
>',
               'legendgroup': 'Cunningham',
               'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
               'name': 'Cunningham',
               'offsetgroup': 'Cunningham',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Richard'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Gray<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Gray',
               'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
               'name': 'Gray',
               'offsetgroup': 'Gray',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Patrick'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Barnett<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Barnett',
               'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
               'name': 'Barnett',
               'offsetgroup': 'Barnett',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Julia'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Brown<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Brown',
               'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
               'name': 'Brown',
               'offsetgroup': 'Brown',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Robert'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Francis<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Francis',
               'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
               'name': 'Francis',
               'offsetgroup': 'Francis',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Edward'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Silk<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Silk',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': 'Silk',
               'offsetgroup': 'Silk',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Martha'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Mitchell<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra
```

```
>',
               'legendgroup': 'Mitchell',
               'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
               'name': 'Mitchell',
               'offsetgroup': 'Mitchell',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Aaron'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Sullivan<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra
>',
               'legendgroup': 'Sullivan',
               'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
               'name': 'Sullivan',
               'offsetgroup': 'Sullivan',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Ellie'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Fernandes<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra
>',
               'legendgroup': 'Fernandes',
               'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
               'name': 'Fernandes',
               'offsetgroup': 'Fernandes',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['João'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Sampaio<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Sampaio',
               'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
               'name': 'Sampaio',
               'offsetgroup': 'Sampaio',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Madalena'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Schneider<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra
>',
               'legendgroup': 'Schneider',
               'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
               'name': 'Schneider',
               'offsetgroup': 'Schneider',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Hannah'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Zimmermann<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra
>',
               'legendgroup': 'Zimmermann',
               'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
               'name': 'Zimmermann',
               'offsetgroup': 'Zimmermann',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Fynn'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Schröder<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra
>',
               'legendgroup': 'Schröder',
               'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
               'name': 'Schröder',
               'offsetgroup': 'Schröder',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Niklas'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Bernard<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Bernard',
               'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
               'name': 'Bernard',
               'offsetgroup': 'Bernard',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Camille'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Lefebvre<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra
>',
               'legendgroup': 'Lefebvre',
               'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
               'name': 'Lefebvre',
               'offsetgroup': 'Lefebvre',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Dominique'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Dubois<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Dubois',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': 'Dubois',
               'offsetgroup': 'Dubois',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Marc'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Girard<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Girard',
               'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
               'name': 'Girard',
               'offsetgroup': 'Girard',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Wyatt'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Mercier<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Mercier',
               'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
               'name': 'Mercier',
               'offsetgroup': 'Mercier',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Isabelle'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Hämäläinen<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra
>',
               'legendgroup': 'Hämäläinen',
               'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
               'name': 'Hämäläinen',
               'offsetgroup': 'Hämäläinen',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Terhi'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Kovács<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Kovács',
               'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
               'name': 'Kovács',
               'offsetgroup': 'Kovács',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Ladislav'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': "LastName=O'Reilly<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra
>",
               'legendgroup': "O'Reilly",
               'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
               'name': "O'Reilly",
               'offsetgroup': "O'Reilly",
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Hugh'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Mancini<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Mancini',
               'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
               'name': 'Mancini',
               'offsetgroup': 'Mancini',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Lucas'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Van der Berg<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></ext
ra>',
               'legendgroup': 'Van der Berg',
               'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
               'name': 'Van der Berg',
               'offsetgroup': 'Van der Berg',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Johannes'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Wójcik<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Wójcik',
               'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
               'name': 'Wójcik',
               'offsetgroup': 'Wójcik',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Stanisław'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Muñoz<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Muñoz',
               'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
               'name': 'Muñoz',
               'offsetgroup': 'Muñoz',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Enrique'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Johansson<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra
>',
               'legendgroup': 'Johansson',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': 'Johansson',
               'offsetgroup': 'Johansson',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Joakim'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Jones<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Jones',
               'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
               'name': 'Jones',
               'offsetgroup': 'Jones',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Emma'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Hughes<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Hughes',
               'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
               'name': 'Hughes',
               'offsetgroup': 'Hughes',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Phil'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Murray<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Murray',
               'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
               'name': 'Murray',
               'offsetgroup': 'Murray',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Steve'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Taylor<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Taylor',
               'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
               'name': 'Taylor',
               'offsetgroup': 'Taylor',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Mark'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Gutiérrez<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra
>',
               'legendgroup': 'Gutiérrez',
               'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
               'name': 'Gutiérrez',
               'offsetgroup': 'Gutiérrez',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Diego'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Rojas<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Rojas',
               'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
               'name': 'Rojas',
               'offsetgroup': 'Rojas',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Luis'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Pareek<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Pareek',
               'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
               'name': 'Pareek',
               'offsetgroup': 'Pareek',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Manoj'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Srivastava<br>FirstName=%{x}<br>TotalInvoices=%{y}<extra></extra
>',
               'legendgroup': 'Srivastava',
               'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
               'name': 'Srivastava',
               'offsetgroup': 'Srivastava',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Puja'], dtype=object),
               'xaxis': 'x',
               'y': array([6]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'title': {'text': 'LastName'}, 'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'title': {'text': 'Total Invoices per Customer'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'FirstName'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalInvoices'}}}
 }))
```

In [23]:

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

vn.ask(question=question)
```

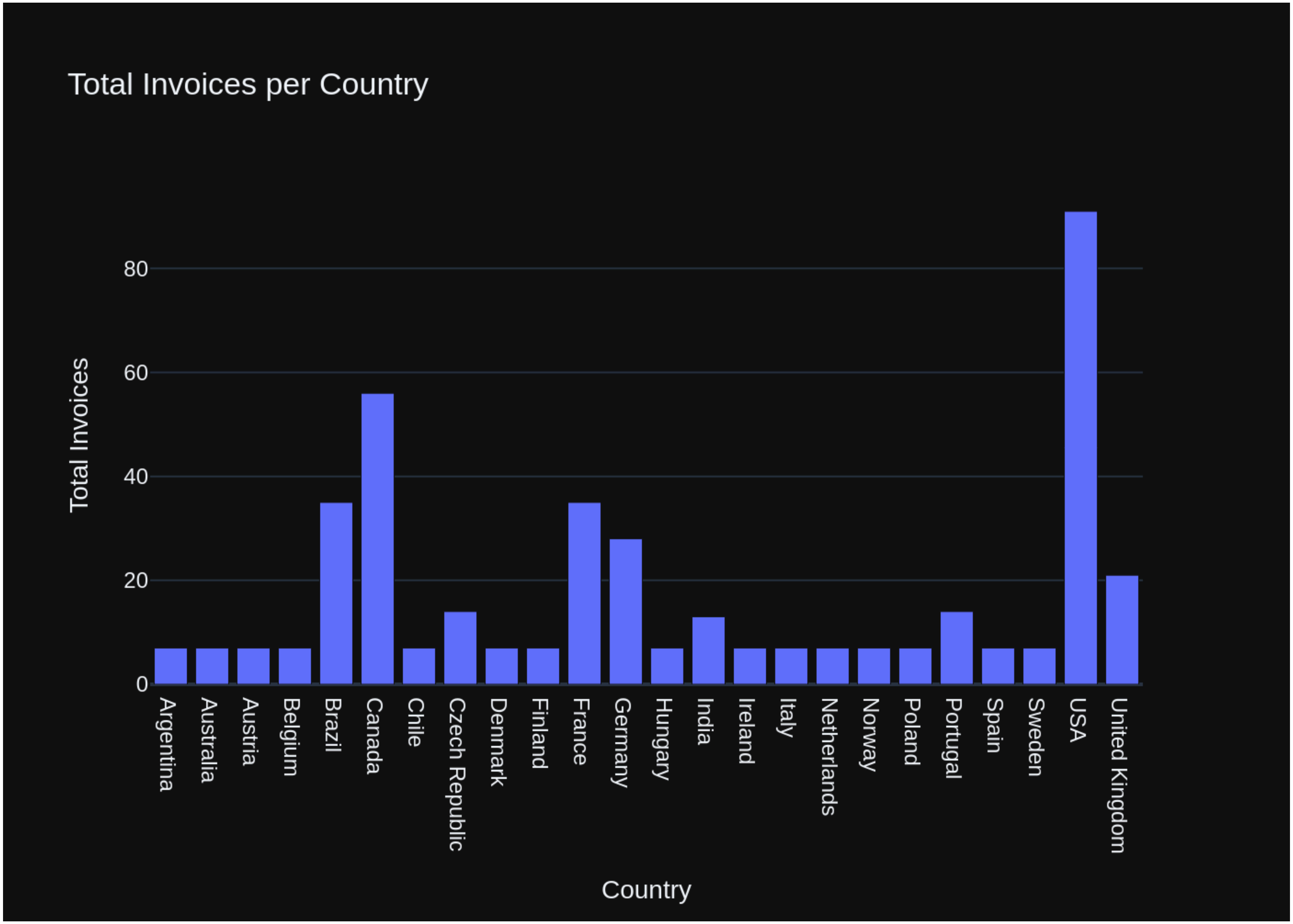
```
Number of requested results 10 is greater than number of elements in index 6, updating n_results = 6
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingA
ddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCount
ry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIG
N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r
\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\t
ON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t
\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (Custome
rId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceLin
eTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE "employees"\r\n(\r\n    EmployeeId INTEGER PRIMARY KE
Y AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NUL
L,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIM
E,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHA
R(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVA
RCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION O
N UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company
NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Coun
try NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n
Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "em
ployees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r
\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    Arti
stId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREM
ENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT
NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n
Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums"
(AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres"
(GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "med
ia_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeRe
portsTo ON "employees" (ReportsTo)\n\n\n===Additional Context \n\nIn the chinook database invoice means ord
er\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query
without any explanations for the question. \n2. If the provided context is almost sufficient but requires k
nowledge of a specific string in a particular column, please generate an intermediate SQL query to find the
distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provi
ded context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant t
able(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it wa
s given before. \n'}, {'role': 'user', 'content': '  \n    Get the total number of invoices for each custom
```

```
er\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId)
AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId,
c.FirstName, c.LastName'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come fr
om?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP B
Y Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are ther
e'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': '  \n
List all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT album
s.Title, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId = artists.ArtistId'}, {'role': 'user',
'content': '  \n    Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assista
nt', 'content': "SELECT * FROM tracks\nWHERE Name LIKE '%What%'"}, {'role': 'user', 'content': 'Can you lis
t all tables in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_m
aster\nWHERE type = 'table'"}, {'role': 'user', 'content': '  \n    Find the total number of invoices per c
ountry:\n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    Billin
gAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCou
ntry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FORE
IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NO
T NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (Tra
ckId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoi
ces\" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON \"invoice_items\" (InvoiceId)\n\nCREATE INDE
X IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE TABLE \"employees\"\r\n(\r\n    EmployeeI
d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVAR
CHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n
HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n
Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r
\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON D
ELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n    CustomerId INTEGER PRIMA
RY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT N
ULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARC
HAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax
NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (Support
RepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE
TABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)
NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTE
GER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n
MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds I
NTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (Album
Id) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Ge
nreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY
(MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE INDEX IFK_EmployeeReportsTo ON \"employees\" (ReportsTo)\n\n\n===Additional Context \n\nIn the c
hinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, p
lease generate a valid SQL query without any explanations for the question. \n2. If the provided context is
almost sufficient but requires knowledge of a specific string in a particular column, please generate an in
termediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying i
ntermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n
4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please r
epeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "  \n    Get the total n
umber of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, c.FirstNam
e, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.C
ustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName"}, {"role": "user", "content": "what are the top
5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS Cust
omerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "cont
ent": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"},
{"role": "user", "content": "  \n    List all albums and their corresponding artist names  \n"}, {"role":
"assistant", "content": "SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId =
artists.ArtistId"}, {"role": "user", "content": "  \n    Find all tracks with a name containing \"What\" (c
ase-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks\nWHERE Name LIKE '%What%'"},
{"role": "user", "content": "Can you list all tables in the SQLite database catalog?"}, {"role": "assistan
t", "content": "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {"role": "user", "content": "  \n
Find the total number of invoices per country:\n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:36:11.480515679Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON
c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, 'done_reason': 'stop', 'done': True, 'total_duration': 6
5179403216, 'load_duration': 682777, 'prompt_eval_count': 1549, 'prompt_eval_duration': 55541072000, 'eval_
count': 44, 'eval_duration': 9234270000}
SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.Country
SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices
```

```
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.Country
           Country  TotalInvoices
0        Argentina              7
1        Australia              7
2          Austria              7
3          Belgium              7
4           Brazil             35
5           Canada             56
6            Chile              7
7   Czech Republic             14
8          Denmark              7
9          Finland              7
10          France             35
11         Germany             28
12         Hungary              7
13           India             13
14         Ireland              7
15           Italy              7
16     Netherlands              7
17          Norway              7
18          Poland              7
19        Portugal             14
20           Spain              7
21          Sweden              7
22             USA             91
23  United Kingdom             21
Ollama parameters:
model=codegemma: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
ers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country\n\nThe following is information a
bout the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Country          object\nTotalInvoic
es     int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to char
t the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one
value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanation
s -- just the code."}]
```

```
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:36:31.503482867Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(\n    df,\n    x='Country',\n    y='TotalInvoices',\n    title='Total Invoices per Country',\n    labels={'Country': 'Country', 'TotalInvoices': 'Total Invoices'}\n)\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 19997134989, 'load_duration': 42919127, 'prompt_eval_count': 192, 'prompt_eval_duration': 6587792000, 'eval_count': 71, 'eval_duration': 13316569000}
```


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

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

In [24]:

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

vn.ask(question=question)
```

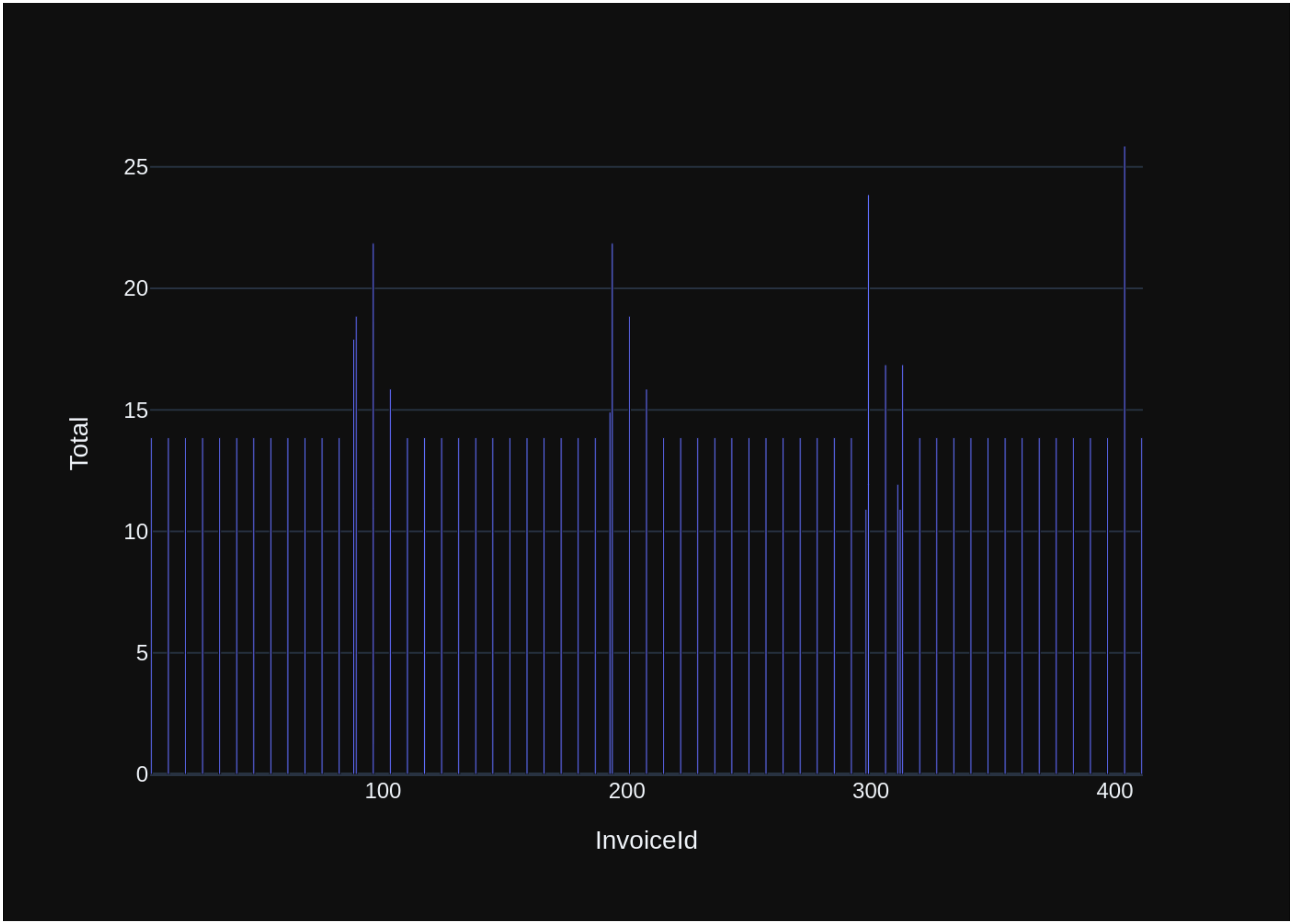
```
Number of requested results 10 is greater than number of elements in index 7, updating n_results = 7
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY
AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPr
ice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES
"invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERE
NCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLin
eInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY
KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n
BillingAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    Bill
ingCountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n
FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO
N\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE INDEX IFK_InvoiceCusto
merId ON "invoices" (CustomerId)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCRE
MENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT
NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n
Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums"
(AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres"
(GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "med
ia_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeRe
portsTo ON "employees" (ReportsTo)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY A
UTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n
Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n
Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r
\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENC
ES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "employee
s"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NUL
L,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    Bir
thDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    St
ate NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r
\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES "employees" (E
mployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON
"customers" (SupportRepId)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n==
=Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without
any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge
of a specific string in a particular column, please generate an intermediate SQL query to find the distinct
strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided 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 c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS Total
Invoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstN
```

```
ame, c.LastName'}, {'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 customers c\n
JOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': 'How many
customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user',
'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SEL
ECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMI
T 5'}, {'role': 'user', 'content': '  \n    List all albums and their corresponding artist names  \n'}, {'r
ole': 'assistant', 'content': 'SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists ON albums.Artis
tId = artists.ArtistId'}, {'role': 'user', 'content': '  \n    Find all tracks with a name containing "Wha
t" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks\nWHERE Name LIKE '%Wha
t%'"}, {'role': 'user', 'content': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'as
sistant', 'content': "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {'role': 'user', 'content': '
\n    List all invoices with a total exceeding $10:\n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoice_items\"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY K
EY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    Uni
tPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCE
S \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) RE
FERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_Invo
iceLineInvoiceId ON \"invoice_items\" (InvoiceId)\n\nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGE
R PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT
NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(4
0),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)
NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE IN
DEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEG
ER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n
MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds I
NTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (Album
Id) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Ge
nreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY
(MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE INDEX IFK_EmployeeReportsTo ON \"employees\" (ReportsTo)\n\nCREATE TABLE \"customers\"\r\n(\r\n
CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    Last
Name NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR
(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone
NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"employees\"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "  \n    Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName"}, {"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 customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId = artists.ArtistId"}, {"role": "user", "content": "  \n    Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks\nWHERE Name LIKE '%What%'"}, {"role": "user", "content": "Can you list all tables in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {"role": "user", "content": "  \n    List all invoices with a total exceeding $10:\n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:37:32.594492891Z', 'message': {'role': 'assistant', 'content': 'SELECT * FROM invoices\nWHERE Total > 10'}, 'done_reason': 'stop', 'done': True, 'total_duration': 60992181670, 'load_duration': 1833157, 'prompt_eval_count': 1581, 'prompt_eval_duration': 56917248000, 'eval_count': 18, 'eval_duration': 3629388000}
SELECT * FROM invoices
WHERE Total > 10
SELECT * FROM invoices
WHERE Total > 10
    InvoiceId  CustomerId          InvoiceDate             BillingAddress  \
0           5          23  2009-01-11 00:00:00            69 Salem Street
```

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

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

[64 rows x 9 columns]
Ollama parameters:
model=codegemma: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 * FROM invoices\nWHERE Total > 10\n\nThe following is infor
mation about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n InvoiceId              int64
\nCustomerId             int64\nInvoiceDate           object\nBillingAddress        object\nBillingCity
object\nBillingState          object\nBillingCountry        object\nBillingPostalCode     object\nTotal
float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the
results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value
in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- j
ust the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:37:46.277509719Z', 'message': {'role': 'assistan
```

t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='InvoiceId', y='Total')\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 13656469519, 'load_duration': 48350830, 'prompt_eval_count': 206, 'prompt_eval_duration': 7044713000, 'eval_count': 35, 'eval_duration': 6473314000}



Out[24]:

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

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

 [64 rows x 9 columns],
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'InvoiceId=%{x}<br>Total=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([  5,  12,  19,  26,  33,  40,  47,  54,  61,  68,  75,  82,  88,  89,
                            96, 103, 110, 117, 124, 131, 138, 145, 152, 159, 166, 173, 180, 187,
                           193, 194, 201, 208, 215, 222, 229, 236, 243, 250, 257, 264, 271, 278,
                          285, 292, 298, 299, 306, 311, 312, 313, 320, 327, 334, 341, 348, 355,
                          362, 369, 376, 383, 390, 397, 404, 411]),
              'xaxis': 'x',
              'y': array([13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                          13.86, 13.86, 17.91, 18.86, 21.86, 15.86, 13.86, 13.86, 13.86, 13.86,
                          13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 14.91, 21.86,
                          18.86, 15.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                          13.86, 13.86, 13.86, 13.86, 10.91, 23.86, 16.86, 11.94, 10.91, 16.86,
                          13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                          13.86, 13.86, 25.86, 13.86]),
              'yaxis': 'y'}],
    'layout': {'barmode': 'relative',
               'legend': {'tracegroupgap': 0},
               'margin': {'t': 60},
               'template': '...',
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceId'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total'}}}
}))
```

In [25]:

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

vn.ask(question=question)
```

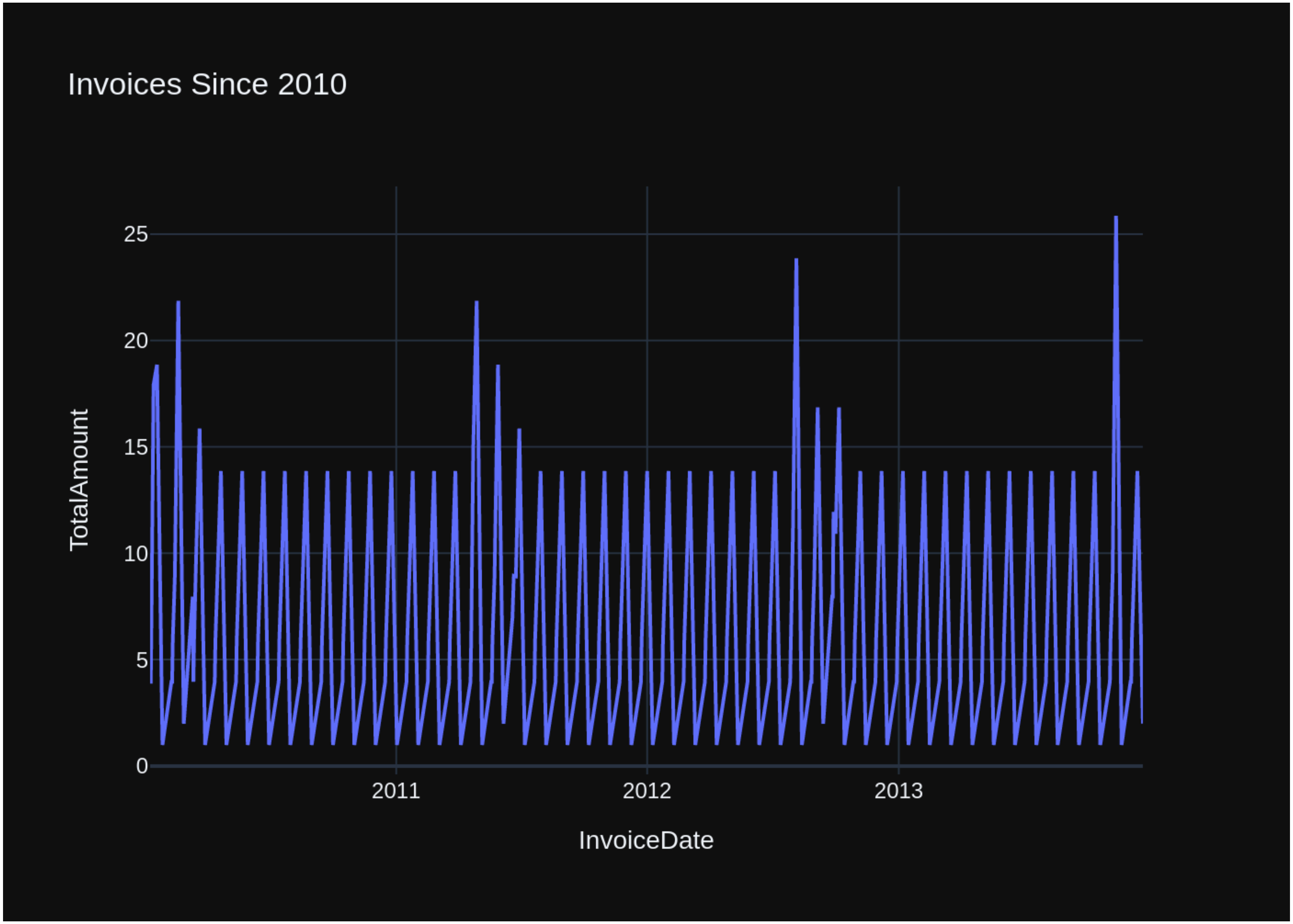
```
Number of requested results 10 is greater than number of elements in index 8, updating n_results = 8
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingA
ddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCount
ry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIG
N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r
\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\t
ON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t
\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items"
(InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLin
eTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE "employees"\r\n(\r\n    EmployeeId INTEGER PRIMARY KE
Y AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NUL
L,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIM
E,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHA
R(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVA
RCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION O
N UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company
NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Coun
try NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n
Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "em
ployees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n(\r
\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    Album
Id INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r
\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n
FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n
FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n
FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO A
CTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES
"artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_trac
k"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_Playlis
tTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Playlist
Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (Trac
kId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook dat
abase invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please gene
rate a valid SQL query without any explanations for the question. \n2. If the provided context is almost su
fficient but requires knowledge of a specific string in a particular column, please generate an intermediat
e SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermedia
```

```
te_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please
use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the
answer exactly as it was given before. \n'}, {'role': 'user', 'content': '  \n    List all invoices with a
total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices\nWHERE Total > 10'}, {'r
ole': 'user', 'content': '  \n    Get the total number of invoices for each customer\n'}, {'role': 'assista
nt', 'content': 'SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM cu
stomers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastNam
e'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per country:\n'}, {'role': 'ass
istant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices
i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': 'How many customers are
there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': 'w
hat are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country,
COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'rol
e': 'user', 'content': '  \n    Find all tracks with a name containing "What" (case-insensitive)\n'}, {'rol
e': 'assistant', 'content': "SELECT * FROM tracks\nWHERE Name LIKE '%What%'"}, {'role': 'user', 'content':
'  \n    List all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELE
CT albums.Title, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId = artists.ArtistId'}, {'role':
'user', 'content': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'assistant', 'conte
nt': "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {'role': 'user', 'content': '  \n    Find all
invoices since 2010 and the total amount invoiced:\n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    Billin
gAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCou
ntry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FORE
IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NO
T NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (Tra
ckId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON \"in
voice_items\" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDE
X IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE TABLE \"employees\"\r\n(\r\n    EmployeeI
d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVAR
CHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n
HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n
Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r
\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON D
ELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n    CustomerId INTEGER PRIMA
RY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT N
ULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARC
HAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax 
NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (Support
RepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE 
TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200) 
 NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Comp
oser NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(1
0,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON 
UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION 
ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON D
ELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY 
AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOR
EIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) 
\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT N
ULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId)
REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY 
(TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Addi
tional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provid
ed context is sufficient, please generate a valid SQL query without any explanations for the question. \n2.
If the provided context is almost sufficient but requires knowledge of a specific string in a particular co
lumn, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the qu
ery with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why
it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and 
answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content":
"  \n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM
invoices\nWHERE Total > 10"}, {"role": "user", "content": "  \n    Get the total number of invoices for eac
h customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.Inv
oiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Cust
omerId, c.FirstName, c.LastName"}, {"role": "user", "content": "  \n    Find the total number of invoices p
er country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nF
ROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"role": "user", "co
ntent": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customer
s"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assis
tant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY Cu
stomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    Find all tracks with a name containing 
\"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks\nWHERE Name LIKE '%
What%'"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist names  \n"},
{"role": "assistant", "content": "SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists ON albums.Ar
```

```
tistId = artists.ArtistId"}, {"role": "user", "content": "Can you list all tables in the SQLite database ca
talog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {"rol
e": "user", "content": "  \n    Find all invoices since 2010 and the total amount invoiced:\n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:39:00.194339895Z', 'message': {'role': 'assistan
t', 'content': "SELECT InvoiceDate, SUM(Total) AS TotalAmount\nFROM invoices\nWHERE InvoiceDate >= '2010-01
-01'\nGROUP BY InvoiceDate"}, 'done_reason': 'stop', 'done': True, 'total_duration': 73821663092, 'load_dur
ation': 634515, 'prompt_eval_count': 1787, 'prompt_eval_duration': 64304207000, 'eval_count': 43, 'eval_dur
ation': 9015575000}
SELECT InvoiceDate, SUM(Total) AS TotalAmount
FROM invoices
WHERE InvoiceDate >= '2010-01-01'
GROUP BY InvoiceDate
SELECT InvoiceDate, SUM(Total) AS TotalAmount
FROM invoices
WHERE InvoiceDate >= '2010-01-01'
GROUP BY InvoiceDate
             InvoiceDate  TotalAmount
0    2010-01-08 00:00:00         3.96
1    2010-01-09 00:00:00         3.96
2    2010-01-10 00:00:00         6.94
3    2010-01-13 00:00:00        17.91
4    2010-01-18 00:00:00        18.86
..                   ...          ...
277  2013-12-05 00:00:00         3.96
278  2013-12-06 00:00:00         5.94
279  2013-12-09 00:00:00         8.91
280  2013-12-14 00:00:00        13.86
281  2013-12-22 00:00:00         1.99

[282 rows x 2 columns]
Ollama parameters:
model=codegemma: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 InvoiceDate, SUM(Total) AS TotalAmount\nFROM 
invoices\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate\n\nThe following is information about the
resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n InvoiceDate     object\nTotalAmount    float6
4\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the resul
```

```
ts of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in th
e dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just t
he code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:39:16.031609823Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.line(df, x='InvoiceDate', y='TotalAmoun
t', title='Invoices Since 2010')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duratio
n': 15810026927, 'load_duration': 43574197, 'prompt_eval_count': 199, 'prompt_eval_duration': 7149711000,
'eval_count': 46, 'eval_duration': 8567456000}
```


Invoices Since 2010
TotalAmount
0
5
10
15
20
25
2011
2012
2013
InvoiceDate

```
Out[25]: ("SELECT InvoiceDate, SUM(Total) AS TotalAmount\nFROM invoices\nWHERE InvoiceDate >= '2010-01-01'\nGROUP B
         Y InvoiceDate",
                       InvoiceDate  TotalAmount
          0    2010-01-08 00:00:00         3.96
          1    2010-01-09 00:00:00         3.96
          2    2010-01-10 00:00:00         6.94
          3    2010-01-13 00:00:00        17.91
          4    2010-01-18 00:00:00        18.86
          ..                   ...          ...
          277  2013-12-05 00:00:00         3.96
          278  2013-12-06 00:00:00         5.94
          279  2013-12-09 00:00:00         8.91
          280  2013-12-14 00:00:00        13.86
          281  2013-12-22 00:00:00         1.99

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

```
In [26]: question = """
             List all employees and their reporting manager's name (if any):
         """
```

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

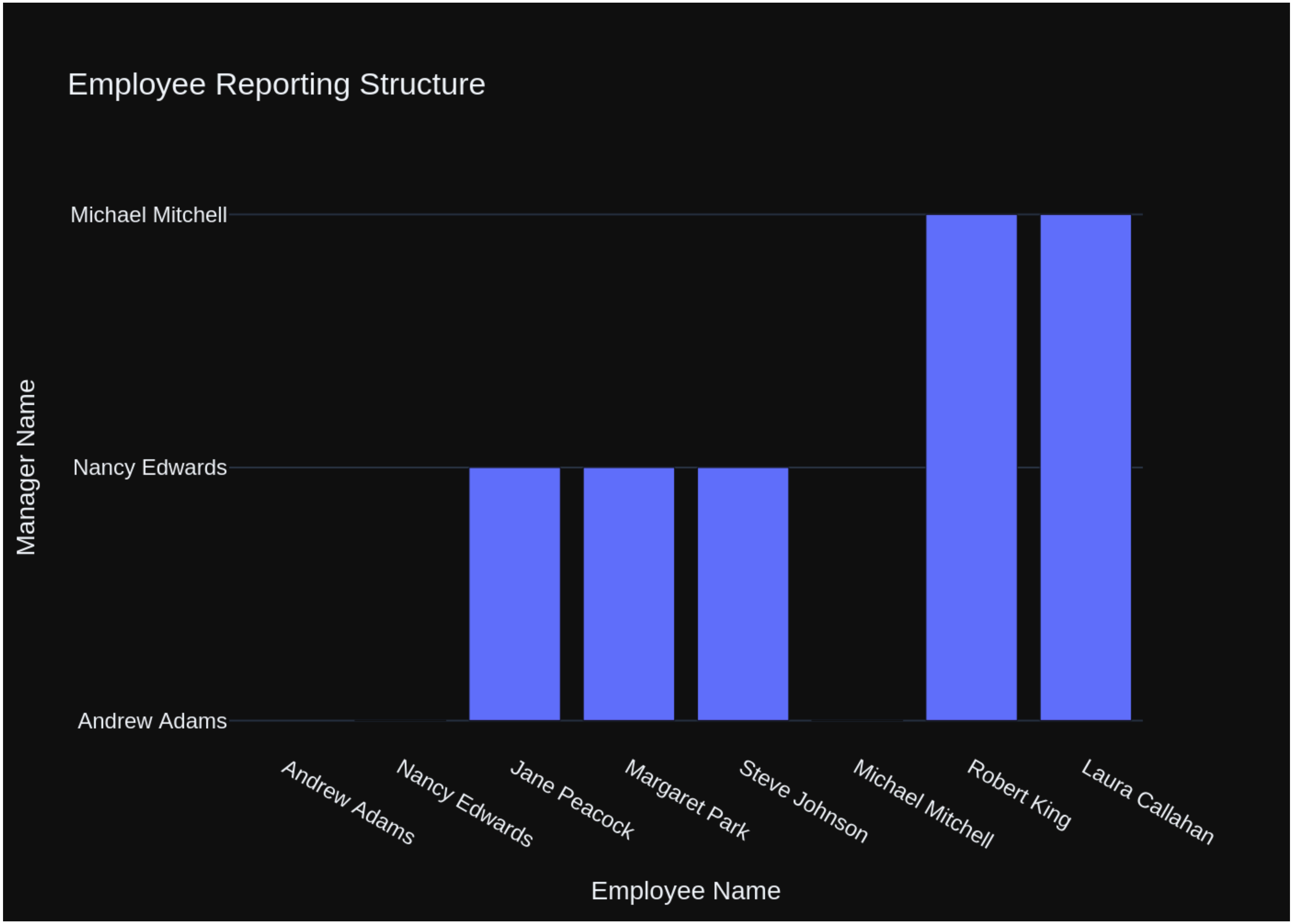
```
Number of requested results 10 is greater than number of elements in index 9, updating n_results = 9
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE TA
BLE "employees"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR
(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGE
R,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(4
0),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NV
ARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES
"employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r
\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NUL
L,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n
City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(1
0),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    Support
RepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO A
CTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCR
EATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId I
NTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    Billing
City NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPost
alCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "cust
omers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r
\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r
\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NUL
L,\r\n    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE
NO ACTION\r\n)\n\nCREATE TABLE "artists"\r\n(\r\n    ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r
\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NU
LL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    By
tes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (Al
bumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (G
enreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media
_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r
\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    Arti
stId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\n\n===Additional Context \n\nI
n the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is suffic
ient, please generate a valid SQL query without any explanations for the question. \n2. If the provided con
text is almost sufficient but requires knowledge of a specific string in a particular column, please genera
te an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment
saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be gener
ated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before,
please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': '  \n    Get the
```

```
total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.Fi
rstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId
= i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName'}, {'role': 'user', 'content': 'what are the
top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS
CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user',
'content': '  \n    Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant',
'content': "SELECT InvoiceDate, SUM(Total) AS TotalAmount\nFROM invoices\nWHERE InvoiceDate >= '2010-01-0
1'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': '  \n    List all albums and their corresponding ar
tist names  \n'}, {'role': 'assistant', 'content': 'SELECT albums.Title, artists.Name\nFROM albums\nJOIN ar
tists ON albums.ArtistId = artists.ArtistId'}, {'role': 'user', 'content': '  \n    Find the total number o
f invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS Tota
lInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role':
'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'conte
nt': 'SELECT * FROM invoices\nWHERE Total > 10'}, {'role': 'user', 'content': 'How many customers are ther
e'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': 'Can y
ou list all tables in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sq
lite_master\nWHERE type = 'table'"}, {'role': 'user', 'content': '  \n    Find all tracks with a name conta
ining "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks\nWHERE Name LIK
E '%What%'"}, {'role': 'user', 'content': "  \n    List all employees and their reporting manager's name (i
f any):\n"}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_EmployeeReportsTo ON \"employees\" (ReportsTo)\n\nCREATE
TABLE \"employees\"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVAR
CHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo IN
TEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCH
AR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phon
e NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENC
ES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"custom
ers\"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT
NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n
City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(1
0),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    Support
RepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n
\nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Custom
erId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    B
illingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    Billi
ngPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES
\"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice_i
tems\"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT
NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER
NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON
UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"artists\"\r\n(\r\n    ArtistId INTEGER PRIMARY KEY AUTOINCREMENT
NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY
AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INT
EGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NUL
L,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES
\"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENC
ES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) R
EFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABL
E \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NO
T NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\n\n===Add
itional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provi
ded context is sufficient, please generate a valid SQL query without any explanations for the question. \n
2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular
column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the
query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain w
hy it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked a
nd answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "conten
t": "  \n    Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELE
CT c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoi
ces i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName"}, {"role": "user", "c
ontent": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELEC
T Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT
5"}, {"role": "user", "content": "  \n    Find all invoices since 2010 and the total amount invoiced:\n"},
{"role": "assistant", "content": "SELECT InvoiceDate, SUM(Total) AS TotalAmount\nFROM invoices\nWHERE Invoi
ceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "content": "  \n    List all albums and th
eir corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT albums.Title, artists.Name\n
FROM albums\nJOIN artists ON albums.ArtistId = artists.ArtistId"}, {"role": "user", "content": "  \n    Fin
d the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT
(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY
c.Country"}, {"role": "user", "content": "  \n    List all invoices with a total exceeding $10:\n"}, {"rol
e": "assistant", "content": "SELECT * FROM invoices\nWHERE Total > 10"}, {"role": "user", "content": "How m
any customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "us
er", "content": "Can you list all tables in the SQLite database catalog?"}, {"role": "assistant", "conten
t": "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {"role": "user", "content": "  \n    Find all
```

```
tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * F
ROM tracks\nWHERE Name LIKE '%What%'"}, {"role": "user", "content": "  \n    List all employees and their r
eporting manager's name (if any):\n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:40:32.847394853Z', 'message': {'role': 'assistan
t', 'content': "SELECT e.FirstName || ' ' || e.LastName AS EmployeeName, m.FirstName || ' ' || m.LastName A
S ManagerName\nFROM employees e\nLEFT JOIN employees m ON e.ReportsTo = m.EmployeeId"}, 'done_reason': 'sto
p', 'done': True, 'total_duration': 76710363997, 'load_duration': 807122, 'prompt_eval_count': 1798, 'promp
t_eval_duration': 64956338000, 'eval_count': 53, 'eval_duration': 11168639000}
SELECT e.FirstName || ' ' || e.LastName AS EmployeeName, m.FirstName || ' ' || m.LastName AS ManagerName
FROM employees e
LEFT JOIN employees m ON e.ReportsTo = m.EmployeeId
SELECT e.FirstName || ' ' || e.LastName AS EmployeeName, m.FirstName || ' ' || m.LastName AS ManagerName
FROM employees e
LEFT JOIN employees m ON e.ReportsTo = m.EmployeeId
       EmployeeName       ManagerName
0      Andrew Adams              None
1     Nancy Edwards      Andrew Adams
2      Jane Peacock     Nancy Edwards
3     Margaret Park     Nancy Edwards
4     Steve Johnson     Nancy Edwards
5  Michael Mitchell      Andrew Adams
6       Robert King  Michael Mitchell
7    Laura Callahan  Michael Mitchell
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all employees and their reporting manager's name (i
f any):\n'\n\nThe DataFrame was produced using this query: SELECT e.FirstName || ' ' || e.LastName AS Emplo
yeeName, m.FirstName || ' ' || m.LastName AS ManagerName\nFROM employees e\nLEFT JOIN employees m ON e.Repo
rtsTo = m.EmployeeId\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning d
f.dtypes gives:\n EmployeeName    object\nManagerName     object\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': 'codegemma:latest', 'created_at': '2024-06-15T19:40:52.158022949Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='EmployeeName', y='ManagerNam
e', title='Employee Reporting Structure')\n\nfig.update_layout(xaxis_title='Employee Name', yaxis_title='Ma
```

```
nager Name')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 19286552822, 'loa
d_duration': 676647, 'prompt_eval_count': 205, 'prompt_eval_duration': 7146268000, 'eval_count': 64, 'eval_
duration': 12006049000}
```



```
Out[26]: ("SELECT e.FirstName || ' ' || e.LastName AS EmployeeName, m.FirstName || ' ' || m.LastName AS ManagerName
         \nFROM employees e\nLEFT JOIN employees m ON e.ReportsTo = m.EmployeeId",
                 EmployeeName       ManagerName
          0      Andrew Adams              None
          1     Nancy Edwards      Andrew Adams
          2      Jane Peacock     Nancy Edwards
          3     Margaret Park     Nancy Edwards
          4     Steve Johnson     Nancy Edwards
          5  Michael Mitchell      Andrew Adams
          6       Robert King  Michael Mitchell
          7    Laura Callahan  Michael Mitchell,
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'EmployeeName=%{x}<br>ManagerName=%{y}<extra></extra>',
                        'legendgroup': '',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array(['Andrew Adams', 'Nancy Edwards', 'Jane Peacock', 'Margaret Park',
                                    'Steve Johnson', 'Michael Mitchell', 'Robert King', 'Laura Callahan'],
                                   dtype=object),
                        'xaxis': 'x',
                        'y': array([None, 'Andrew Adams', 'Nancy Edwards', 'Nancy Edwards', 'Nancy Edwards',
                                    'Andrew Adams', 'Michael Mitchell', 'Michael Mitchell'], dtype=object),
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                         'legend': {'tracegroupgap': 0},
                         'template': '...',
                         'title': {'text': 'Employee Reporting Structure'},
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Employee Name'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Manager Name'}}}
          }))
```

```
In [27]: question = """
             Get the average invoice total for each customer:
         """
```

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

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

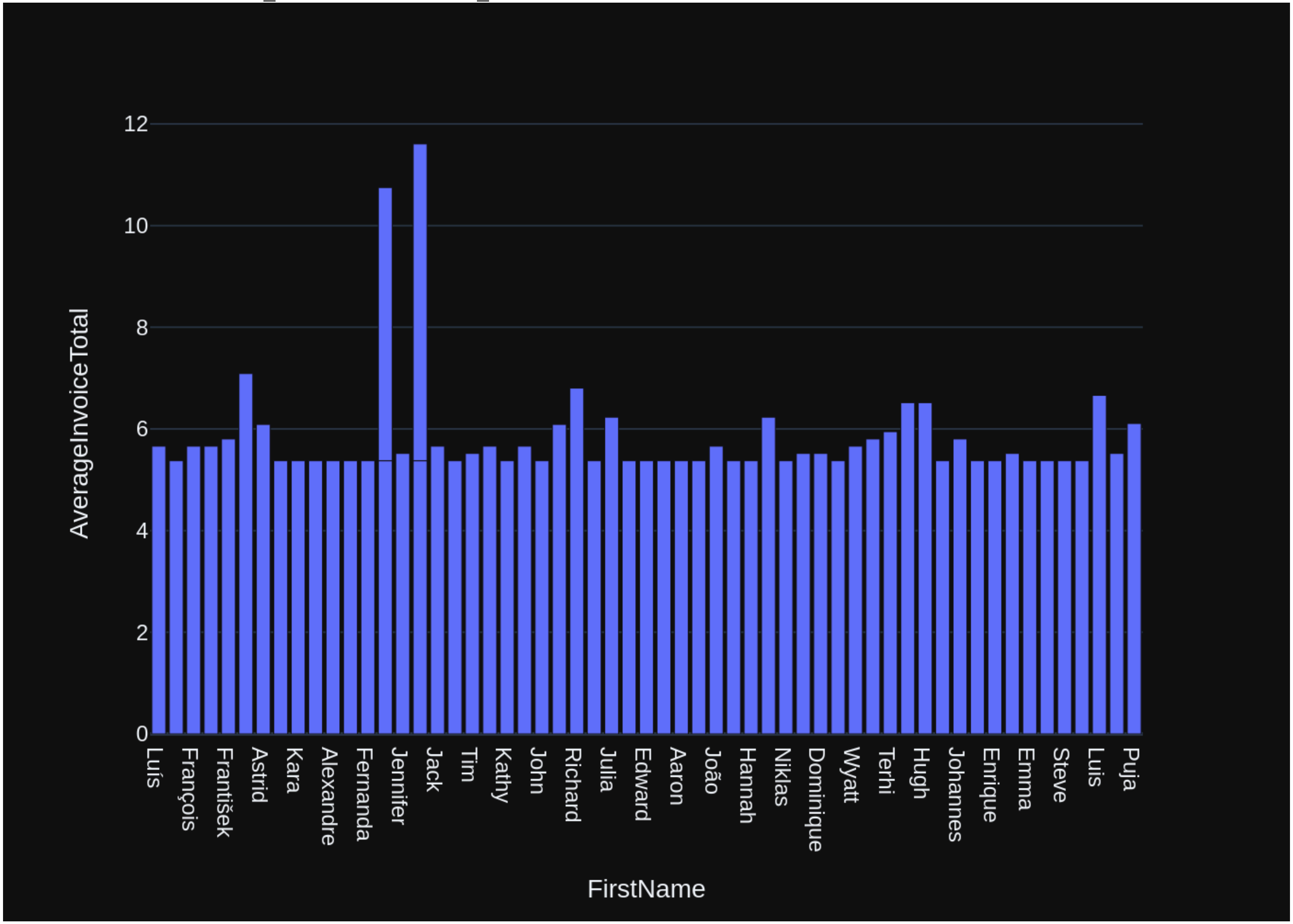
```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingA
ddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCount
ry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIG
N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId
ON "invoice_items" (InvoiceId)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KE
Y AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    Unit
Price NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES
"invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERE
NCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLin
eTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE INDEX IFK_Custom
erSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRI
MARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT
NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVAR
CHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fa
x NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (Suppo
rtRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE
INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE TABLE "employees"\r\n(\r\n    EmployeeId I
NTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHA
R(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    Hir
eDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Co
untry NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n
Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order
\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query 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 ta
ble(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was
given before. \n'}, {'role': 'user', 'content': '  \n    Get the total number of invoices for each customer
\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS
TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.F
irstName, c.LastName'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per countr
y:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM custo
mers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': '
\n    Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SE
LECT InvoiceDate, SUM(Total) AS TotalAmount\nFROM invoices\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY Inv
oiceDate"}, {'role': 'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'rol
```

```
e': 'assistant', 'content': 'SELECT * FROM invoices\nWHERE Total > 10'}, {'role': 'user', 'content': 'How m
any customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'us
er', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content':
'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\n
LIMIT 5'}, {'role': 'user', 'content': "  \n    List all employees and their reporting manager's name (if a
ny):\n"}, {'role': 'assistant', 'content': "SELECT e.FirstName || ' ' || e.LastName AS EmployeeName, m.Firs
tName || ' ' || m.LastName AS ManagerName\nFROM employees e\nLEFT JOIN employees m ON e.ReportsTo = m.Emplo
yeeId"}, {'role': 'user', 'content': '  \n    Find all tracks with a name containing "What" (case-insensiti
ve)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks\nWHERE Name LIKE '%What%'"}, {'role': 'use
r', 'content': '  \n    List all albums and their corresponding artist names  \n'}, {'role': 'assistant',
'content': 'SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId = artists.Artis
tId'}, {'role': 'user', 'content': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'as
sistant', 'content': "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {'role': 'user', 'content': '
\n    Get the average invoice total for each customer:\n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    Billin
gAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCou
ntry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FORE
IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineInv
oiceId ON \"invoice_items\" (InvoiceId)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n    InvoiceLineId INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r
\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId)
REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Tr
ackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX
IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE
INDEX IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"customers\"\r\n(\r\n    Cu
stomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastNa
me NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(4
0),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NV
ARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n
FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO A
CTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON \"employees\" (ReportsTo)\n\nCREATE TABLE \"employees
\"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NUL
L,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    Bir
thDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    St
ate NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r
\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES \"employees\"
(EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chi
nook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, ple
ase generate a valid SQL query without any explanations for the question. \n2. If the provided context is a
lmost sufficient but requires knowledge of a specific string in a particular column, please generate an int
ermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying in
termediate_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 repe
at the answer exactly as it was given before. \n"}, {"role": "user", "content": "  \n    Get the total numb
er of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, c.FirstName,
c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.Cust
omerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName"}, {"role": "user", "content": "  \n    Find the tot
al number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.Invoice
Id) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Countr
y"}, {"role": "user", "content": "  \n    Find all invoices since 2010 and the total amount invoiced:\n"},
{"role": "assistant", "content": "SELECT InvoiceDate, SUM(Total) AS TotalAmount\nFROM invoices\nWHERE Invoi
ceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "content": "  \n    List all invoices with
a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices\nWHERE Total > 10"},
{"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT
(*) FROM customers"}, {"role": "user", "content": "what are the top 5 countries that customers come fro
m?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY
Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    List all employees an
d their reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT e.FirstName || ' '
|| e.LastName AS EmployeeName, m.FirstName || ' ' || m.LastName AS ManagerName\nFROM employees e\nLEFT JOIN
employees m ON e.ReportsTo = m.EmployeeId"}, {"role": "user", "content": "  \n    Find all tracks with a na
me containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks\nWHER
E Name LIKE '%What%'"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist
names  \n"}, {"role": "assistant", "content": "SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists
ON albums.ArtistId = artists.ArtistId"}, {"role": "user", "content": "Can you list all tables in the SQLite
database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master\nWHERE type = 'tabl
e'"}, {"role": "user", "content": "  \n    Get the average invoice total for each customer:\n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:42:04.951048907Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.CustomerId, c.FirstName, c.LastName, AVG(i.Total) AS AverageInvoiceTotal\nFROM cus
tomers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName'},
'done_reason': 'stop', 'done': True, 'total_duration': 72698657981, 'load_duration': 708733, 'prompt_eval_c
ount': 1660, 'prompt_eval_duration': 59392843000, 'eval_count': 60, 'eval_duration': 12627796000}
SELECT c.CustomerId, c.FirstName, c.LastName, AVG(i.Total) AS AverageInvoiceTotal
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
```

```
GROUP BY c.CustomerId, c.FirstName, c.LastName
SELECT c.CustomerId, c.FirstName, c.LastName, AVG(i.Total) AS AverageInvoiceTotal
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId, c.FirstName, c.LastName
    CustomerId  FirstName      LastName  AverageInvoiceTotal
0            1       Luís     Gonçalves             5.660000
1            2     Leonie        Köhler             5.374286
2            3   François      Tremblay             5.660000
3            4      Bjørn        Hansen             5.660000
4            5  František   Wichterlová             5.802857
5            6     Helena          Holý             7.088571
6            7     Astrid        Gruber             6.088571
7            8       Daan       Peeters             5.374286
8            9       Kara       Nielsen             5.374286
9           10    Eduardo       Martins             5.374286
10          11  Alexandre         Rocha             5.374286
11          12    Roberto       Almeida             5.374286
12          13   Fernanda         Ramos             5.374286
13          14       Mark       Philips             5.374286
14          15   Jennifer      Peterson             5.517143
15          16      Frank        Harris             5.374286
16          17       Jack         Smith             5.660000
17          18   Michelle        Brooks             5.374286
18          19        Tim         Goyer             5.517143
19          20        Dan        Miller             5.660000
20          21      Kathy         Chase             5.374286
21          22    Heather       Leacock             5.660000
22          23       John        Gordon             5.374286
23          24      Frank       Ralston             6.231429
24          25     Victor       Stevens             6.088571
25          26    Richard    Cunningham             6.802857
26          27    Patrick          Gray             5.374286
27          28      Julia       Barnett             6.231429
28          29     Robert         Brown             5.374286
29          30     Edward       Francis             5.374286
30          31     Martha          Silk             5.374286
31          32      Aaron      Mitchell             5.374286
32          33      Ellie      Sullivan             5.374286
33          34       João     Fernandes             5.660000
34          35   Madalena       Sampaio             5.374286
35          36     Hannah     Schneider             5.374286
```

```
36          37       Fynn    Zimmermann             6.231429
37          38     Niklas      Schröder             5.374286
38          39    Camille       Bernard             5.517143
39          40  Dominique      Lefebvre             5.517143
40          41       Marc        Dubois             5.374286
41          42      Wyatt        Girard             5.660000
42          43   Isabelle       Mercier             5.802857
43          44      Terhi    Hämäläinen             5.945714
44          45   Ladislav        Kovács             6.517143
45          46       Hugh      O'Reilly             6.517143
46          47      Lucas       Mancini             5.374286
47          48   Johannes  Van der Berg             5.802857
48          49  Stanisław        Wójcik             5.374286
49          50    Enrique         Muñoz             5.374286
50          51     Joakim     Johansson             5.517143
51          52       Emma         Jones             5.374286
52          53       Phil        Hughes             5.374286
53          54      Steve        Murray             5.374286
54          55       Mark        Taylor             5.374286
55          56      Diego     Gutiérrez             5.374286
56          57       Luis         Rojas             6.660000
57          58      Manoj        Pareek             5.517143
58          59       Puja    Srivastava             6.106667
Ollama parameters:
model=codegemma: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, c.FirstName, c.LastName, AVG(i.Total) AS A
verageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerI
d, c.FirstName, c.LastName\n\nThe following is information about the resulting pandas DataFrame 'df': \nRun
ning df.dtypes gives:\n CustomerId               int64\nFirstName               object\nLastName
object\nAverageInvoiceTotal    float64\ndtype: object"}, {"role": "user", "content": "Can you generate the
Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called
'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not
answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:42:28.234735824Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='FirstName', y='AverageInvoice
Total', hover_name='LastName')\n\nif len(df) == 1:\n    fig.add_trace(px.indicator(value=df['AverageInvoice
```

```
Total'].values[0], title='Average Invoice Total'))\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': Tru
e, 'total_duration': 23255839300, 'load_duration': 662445, 'prompt_eval_count': 220, 'prompt_eval_duratio
n': 7632444000, 'eval_count': 82, 'eval_duration': 15492201000}
```



```
Out[27]: ('SELECT c.CustomerId, c.FirstName, c.LastName, AVG(i.Total) AS AverageInvoiceTotal\nFROM customers c\nJOI
         N invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName',
             CustomerId  FirstName      LastName  AverageInvoiceTotal
         0            1       Luís     Gonçalves             5.660000
         1            2     Leonie        Köhler             5.374286
         2            3   François      Tremblay             5.660000
         3            4      Bjørn        Hansen             5.660000
         4            5  František   Wichterlová             5.802857
         5            6     Helena          Holý             7.088571
         6            7     Astrid        Gruber             6.088571
         7            8       Daan       Peeters             5.374286
         8            9       Kara       Nielsen             5.374286
         9           10    Eduardo       Martins             5.374286
         10          11  Alexandre         Rocha             5.374286
         11          12    Roberto       Almeida             5.374286
         12          13   Fernanda         Ramos             5.374286
         13          14       Mark       Philips             5.374286
         14          15   Jennifer      Peterson             5.517143
         15          16      Frank        Harris             5.374286
         16          17       Jack         Smith             5.660000
         17          18   Michelle        Brooks             5.374286
         18          19        Tim         Goyer             5.517143
         19          20        Dan        Miller             5.660000
         20          21      Kathy         Chase             5.374286
         21          22    Heather       Leacock             5.660000
         22          23       John        Gordon             5.374286
         23          24      Frank       Ralston             6.231429
         24          25     Victor       Stevens             6.088571
         25          26    Richard    Cunningham             6.802857
         26          27    Patrick          Gray             5.374286
         27          28      Julia       Barnett             6.231429
         28          29     Robert         Brown             5.374286
         29          30     Edward       Francis             5.374286
         30          31     Martha          Silk             5.374286
         31          32      Aaron      Mitchell             5.374286
         32          33      Ellie      Sullivan             5.374286
         33          34       João     Fernandes             5.660000
         34          35   Madalena       Sampaio             5.374286
         35          36     Hannah     Schneider             5.374286
         36          37       Fynn    Zimmermann             6.231429
         37          38     Niklas      Schröder             5.374286
         38          39    Camille       Bernard             5.517143
```

```
39          40  Dominique      Lefebvre             5.517143
40          41       Marc        Dubois             5.374286
41          42      Wyatt        Girard             5.660000
42          43   Isabelle       Mercier             5.802857
43          44      Terhi    Hämäläinen             5.945714
44          45   Ladislav        Kovács             6.517143
45          46       Hugh      O'Reilly             6.517143
46          47      Lucas       Mancini             5.374286
47          48   Johannes  Van der Berg             5.802857
48          49  Stanisław        Wójcik             5.374286
49          50    Enrique         Muñoz             5.374286
50          51     Joakim     Johansson             5.517143
51          52       Emma         Jones             5.374286
52          53       Phil        Hughes             5.374286
53          54      Steve        Murray             5.374286
54          55       Mark        Taylor             5.374286
55          56      Diego     Gutiérrez             5.374286
56          57       Luis         Rojas             6.660000
57          58      Manoj        Pareek             5.517143
58          59       Puja    Srivastava             6.106667,
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': ('<b>%{hovertext}</b><br><br>Fir' ... 'voiceTotal=%{y}<extra></extra>'),
              'hovertext': array(['Gonçalves', 'Köhler', 'Tremblay', 'Hansen', 'Wichterlová', 'Holý',
                                  'Gruber', 'Peeters', 'Nielsen', 'Martins', 'Rocha', 'Almeida', 'Ramos',
                                  'Philips', 'Peterson', 'Harris', 'Smith', 'Brooks', 'Goyer', 'Miller',
                                  'Chase', 'Leacock', 'Gordon', 'Ralston', 'Stevens', 'Cunningham',
                                  'Gray', 'Barnett', 'Brown', 'Francis', 'Silk', 'Mitchell', 'Sullivan',
                                  'Fernandes', 'Sampaio', 'Schneider', 'Zimmermann', 'Schröder',
                                  'Bernard', 'Lefebvre', 'Dubois', 'Girard', 'Mercier', 'Hämäläinen',
                                  'Kovács', "O'Reilly", 'Mancini', 'Van der Berg', 'Wójcik', 'Muñoz',
                                  'Johansson', 'Jones', 'Hughes', 'Murray', 'Taylor', 'Gutiérrez',
                                  'Rojas', 'Pareek', 'Srivastava'], dtype=object),
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'v',
              'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
              'x': array(['Luís', 'Leonie', 'François', 'Bjørn', 'František', 'Helena', 'Astrid',
                          'Daan', 'Kara', 'Eduardo', 'Alexandre', 'Roberto', 'Fernanda', 'Mark',
                          'Jennifer', 'Frank', 'Jack', 'Michelle', 'Tim', 'Dan', 'Kathy',
                          'Heather', 'John', 'Frank', 'Victor', 'Richard', 'Patrick', 'Julia',
                          'Robert', 'Edward', 'Martha', 'Aaron', 'Ellie', 'João', 'Madalena',
                          'Hannah', 'Fynn', 'Niklas', 'Camille', 'Dominique', 'Marc', 'Wyatt',
                          'Isabelle', 'Terhi', 'Ladislav', 'Hugh', 'Lucas', 'Johannes',
                          'Stanisław', 'Enrique', 'Joakim', 'Emma', 'Phil', 'Steve', 'Mark',
                          'Diego', 'Luis', 'Manoj', 'Puja'], dtype=object),
              'xaxis': 'x',
              'y': array([5.66      , 5.37428571, 5.66      , 5.66      , 5.80285714, 7.08857143,
                          6.08857143, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571,
                          5.37428571, 5.37428571, 5.51714286, 5.37428571, 5.66      , 5.37428571,
                          5.51714286, 5.66      , 5.37428571, 5.66      , 5.37428571, 6.23142857,
                          6.08857143, 6.80285714, 5.37428571, 6.23142857, 5.37428571, 5.37428571,
                          5.37428571, 5.37428571, 5.37428571, 5.66      , 5.37428571, 5.37428571,
                          6.23142857, 5.37428571, 5.51714286, 5.51714286, 5.37428571, 5.66      ,
                          5.80285714, 5.94571429, 6.51714286, 6.51714286, 5.37428571, 5.80285714,
                          5.37428571, 5.37428571, 5.51714286, 5.37428571, 5.37428571, 5.37428571,
                          5.37428571, 5.37428571, 6.66      , 5.51714286, 6.10666667]),
              'yaxis': 'y'}],
    'layout': {'barmode': 'relative',
               'legend': {'tracegroupgap': 0},
               'margin': {'t': 60},
               'template': '...',
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'FirstName'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'AverageInvoiceTotal'}}}
}))
```

In [28]:

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

vn.ask(question=question)
```

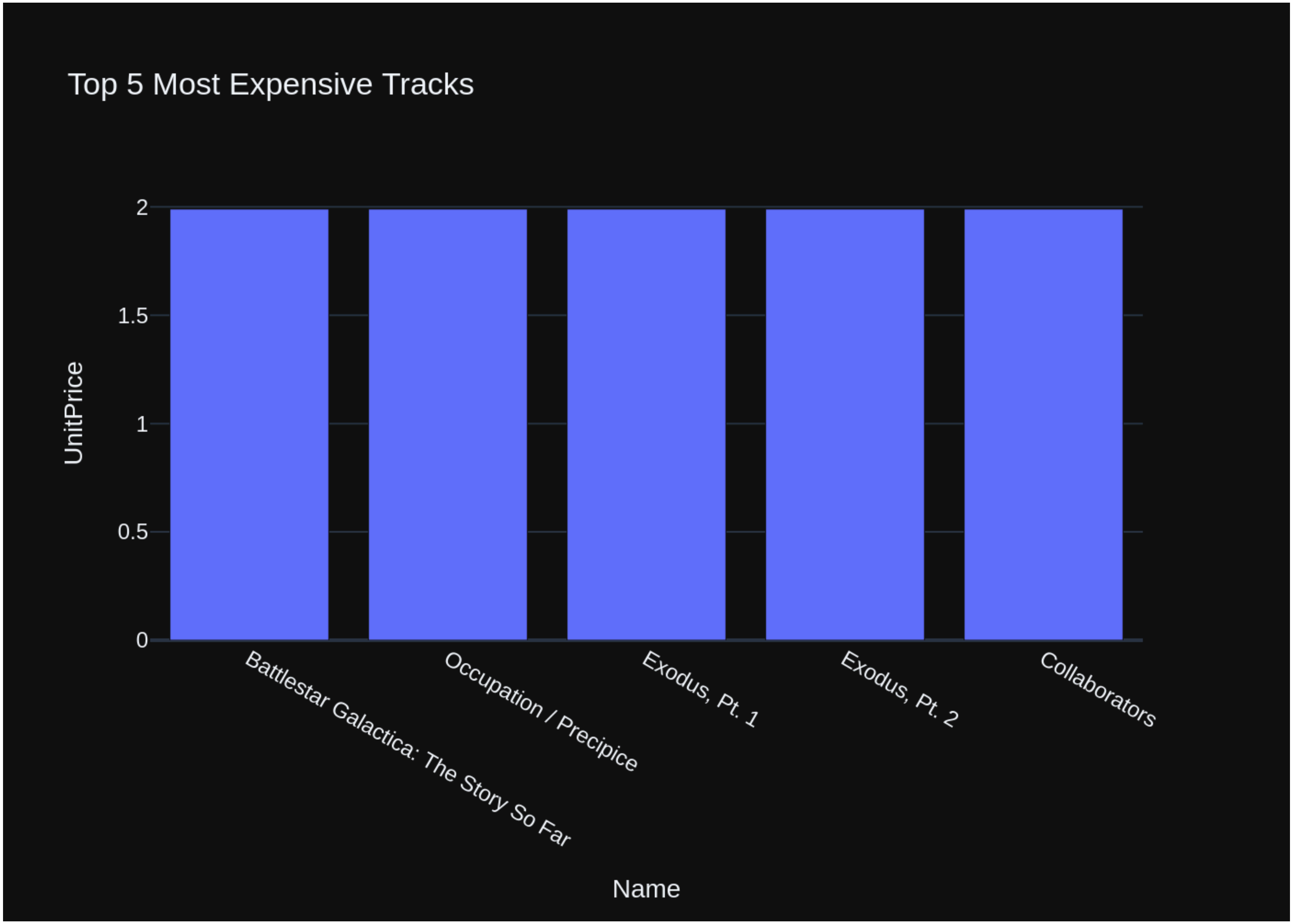
```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NU
LL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    By
tes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (Al
bumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (G
enreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media
_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackAlbumId
ON "tracks" (AlbumId)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK_PlaylistTr
ackTrackId ON "playlist_track" (TrackId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackI
d)\n\nCREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE TABLE "invoice_items"\r\n(\r\n
InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackI
d INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FO
REIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r
\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r
\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT
NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistI
d) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY
(TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDE
X IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY
AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOR
EIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n
\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If
the provided context is sufficient, please generate a valid SQL query without any explanations for the ques
tion. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a pa
rticular column, please generate an intermediate SQL query to find the distinct strings in that column. Pre
pend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please
explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has be
en asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'use
r', 'content': '  \n    Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'ass
istant', 'content': "SELECT * FROM tracks\nWHERE Name LIKE '%What%'"}, {'role': 'user', 'content': '  \n
List all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT album
s.Title, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId = artists.ArtistId'}, {'role': 'user',
'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SE
LECT * FROM invoices\nWHERE Total > 10'}, {'role': 'user', 'content': '  \n    Get the average invoice tota
l for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.FirstName, c.LastName, A
VG(i.Total) AS AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP
BY c.CustomerId, c.FirstName, c.LastName'}, {'role': 'user', 'content': '  \n    Find all invoices since 20
10 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceDate, SUM(Total) AS
TotalAmount\nFROM invoices\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'co
ntent': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT
```

```
Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT
5'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per country:\n'}, {'role': 'ass
istant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices
i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': '  \n    Get the total
number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.FirstNam
e, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.C
ustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName'}, {'role': 'user', 'content': 'How many customer
s are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'conten
t': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT na
me FROM sqlite_master\nWHERE type = 'table'"}, {'role': 'user', 'content': '  \n    Find the top 5 most exp
ensive tracks (based on unit price):\n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREM
ENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT
NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n
Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\"
(AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres
\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES
\"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_Trac
kAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX I
FK_PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"invoice
_items\" (TrackId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE TABLE \"invoi
ce_items\"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER
NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEG
ER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTIO
N ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO AC
TION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n    PlaylistId INTEGER  NOT NUL
L,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackI
d),\r\n    FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UP
DATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON
UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"albums
\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r
\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice
means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid S
QL query without any explanations for the question. \n2. If the provided context is almost sufficient but r
```

```
equires knowledge of a specific string in a particular column, please generate an intermediate SQL query to
find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If
the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most re
levant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly
as it was given before. \n"}, {"role": "user", "content": "  \n    Find all tracks with a name containing
\"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks\nWHERE Name LIKE '%
What%'"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist names  \n"},
{"role": "assistant", "content": "SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists ON albums.Ar
tistId = artists.ArtistId"}, {"role": "user", "content": "  \n    List all invoices with a total exceeding
$10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices\nWHERE Total > 10"}, {"role": "user", "c
ontent": "  \n    Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "S
ELECT c.CustomerId, c.FirstName, c.LastName, AVG(i.Total) AS AverageInvoiceTotal\nFROM customers c\nJOIN in
voices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName"}, {"role": "user",
"content": "  \n    Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant",
"content": "SELECT InvoiceDate, SUM(Total) AS TotalAmount\nFROM invoices\nWHERE InvoiceDate >= '2010-01-0
1'\nGROUP BY InvoiceDate"}, {"role": "user", "content": "what are the top 5 countries that customers come f
rom?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP
BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    Find the total num
ber of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS
TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"ro
le": "user", "content": "  \n    Get the total number of invoices for each customer\n"}, {"role": "assistan
t", "content": "SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM cus
tomers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName"},
{"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT
(*) FROM customers"}, {"role": "user", "content": "Can you list all tables in the SQLite database catalo
g?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {"role": "u
ser", "content": "  \n    Find the top 5 most expensive tracks (based on unit price):\n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:43:30.0961559Z', 'message': {'role': 'assistan
t', 'content': 'SELECT Name, UnitPrice\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'}, 'done_reason': 'st
op', 'done': True, 'total_duration': 61754956710, 'load_duration': 688102, 'prompt_eval_count': 1556, 'prom
pt_eval_duration': 55958128000, 'eval_count': 25, 'eval_duration': 5123630000}
SELECT Name, UnitPrice
FROM tracks
ORDER BY UnitPrice DESC
LIMIT 5
SELECT Name, UnitPrice
FROM tracks
ORDER BY UnitPrice DESC
LIMIT 5
                                     Name  UnitPrice
0  Battlestar Galactica: The Story So Far       1.99
```

```
1                  Occupation / Precipice       1.99
2                           Exodus, Pt. 1       1.99
3                           Exodus, Pt. 2       1.99
4                           Collaborators       1.99
Ollama parameters:
model=codegemma: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 Name, UnitPrice\nFROM tracks\nORDER BY UnitPr
ice DESC\nLIMIT 5\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.d
types gives:\n Name          object\nUnitPrice    float64\ndtype: object"}, {"role": "user", "content": "Ca
n you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas 
dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Pyt
hon code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:43:44.27134082Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Name', y='UnitPrice', title
='Top 5 Most Expensive Tracks')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duratio
n': 14148492565, 'load_duration': 627186, 'prompt_eval_count': 177, 'prompt_eval_duration': 6032723000, 'ev
al_count': 43, 'eval_duration': 7984407000}
```


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

Out[28]:

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

In [29]:

```python
question = """
    List all genres and the number of tracks in each genre:
"""

vn.ask(question=question)
```

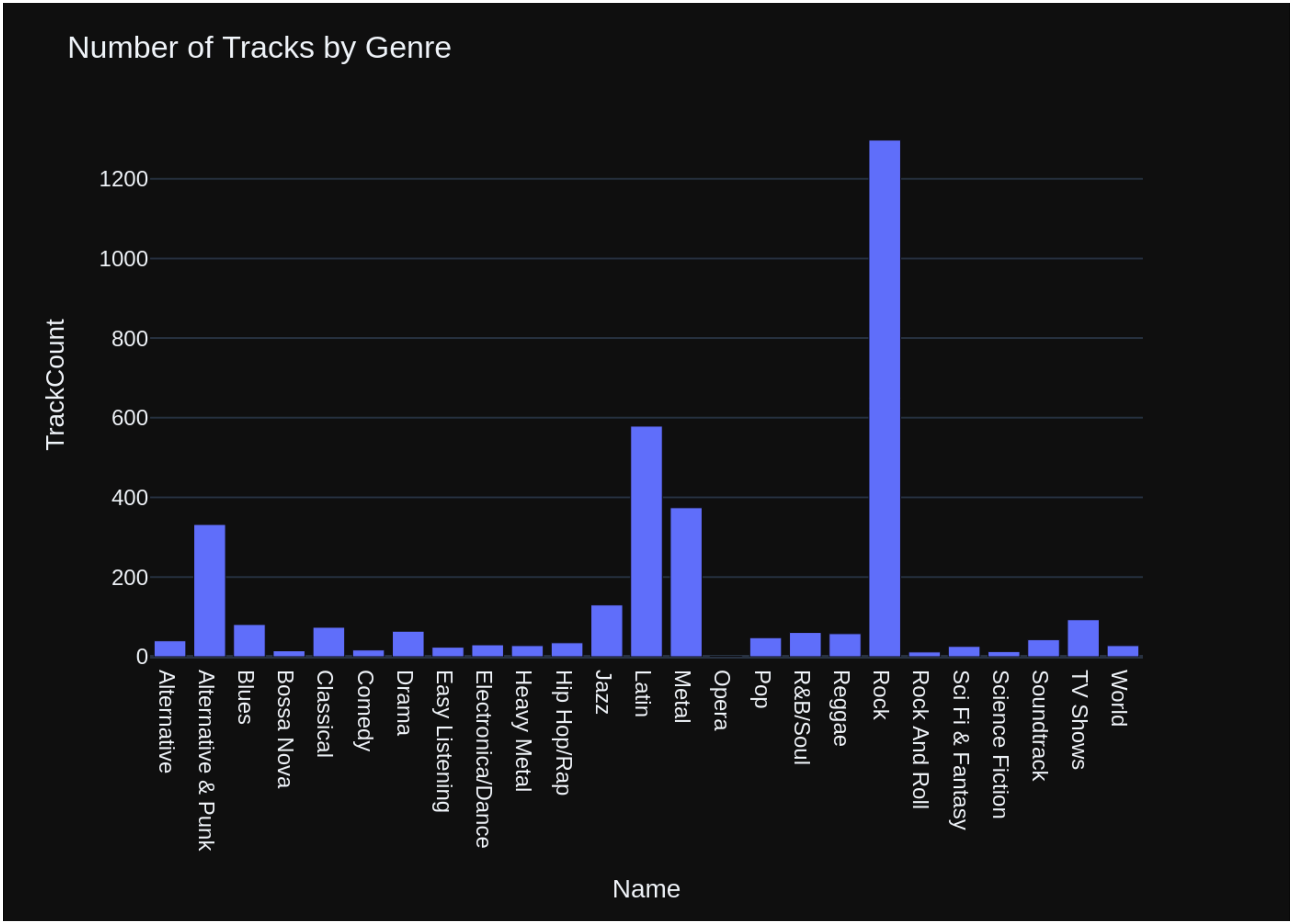
```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'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 fo
rmat instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NU
LL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    By
tes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (Al
bumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (G
enreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media
_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackGenreId
ON "tracks" (GenreId)\n\nCREATE TABLE "genres"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NU
LL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n
\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "playlists"\r\n(\r\n    PlaylistId IN
TEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_TrackMediaType
Id ON "tracks" (MediaTypeId)\n\nCREATE TABLE "playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n
TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FO
REIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO
N,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI
ON\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Tit
le NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "ar
tists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON
"albums" (ArtistId)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Respon
se Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any exp
lanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a s
pecific string in a particular column, please generate an intermediate SQL query to find the distinct strin
gs in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is
insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. I
f 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 Name, UnitPrice\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'},
{'role': 'user', 'content': '  \n    List all albums and their corresponding artist names  \n'}, {'role':
'assistant', 'content': 'SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId =
artists.ArtistId'}, {'role': 'user', 'content': '  \n    Find all tracks with a name containing "What" (cas
e-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks\nWHERE Name LIKE '%What%'"}, {'r
ole': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant',
'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerC
ount DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per countr
y:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM custo
mers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': '
\n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM in
voices\nWHERE Total > 10'}, {'role': 'user', 'content': 'Can you list all tables in the SQLite database cat
alog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {'role':
'user', 'content': '  \n    Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assi
```

```
stant', 'content': "SELECT InvoiceDate, SUM(Total) AS TotalAmount\nFROM invoices\nWHERE InvoiceDate >= '201
0-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': '  \n    Get the total number of invoices for
each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.
InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.C
ustomerId, c.FirstName, c.LastName'}, {'role': 'user', 'content': 'How many customers are there'}, {'role':
'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': '  \n    List all ge
nres and the number of tracks in each genre:\n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREM
ENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT
NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n
Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\"
(AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres
\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES
\"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_Trac
kGenreId ON \"tracks\" (GenreId)\n\nCREATE TABLE \"genres\"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOINC
REMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_trac
k\" (TrackId)\n\nCREATE INDEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"playlists\"\r\n(\r
\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX
IFK_TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n    PlaylistId
INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (Playl
istId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE
NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOI
NCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN
KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCR
EATE INDEX IFK_AlbumArtistId ON \"albums\" (ArtistId)\n\n\n===Additional Context \n\nIn the chinook databas
e invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate
a valid SQL query without any explanations for the question. \n2. If the provided context is almost suffici
ent but requires knowledge of a specific string in a particular column, please generate an intermediate SQL
query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql
\n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the
most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer e
xactly as it was given before. \n"}, {"role": "user", "content": "  \n    Find the top 5 most expensive tra
cks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT Name, UnitPrice\nFROM tracks\nORDE
R BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    List all albums and their correspondin
```

```
g artist names  \n"}, {"role": "assistant", "content": "SELECT albums.Title, artists.Name\nFROM albums\nJOI
N artists ON albums.ArtistId = artists.ArtistId"}, {"role": "user", "content": "  \n    Find all tracks wit
h a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks
\nWHERE Name LIKE '%What%'"}, {"role": "user", "content": "what are the top 5 countries that customers come
from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP
BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    Find the total num
ber of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS
TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"ro
le": "user", "content": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant",
"content": "SELECT * FROM invoices\nWHERE Total > 10"}, {"role": "user", "content": "Can you list all table
s in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master\nWHER
E type = 'table'"}, {"role": "user", "content": "  \n    Find all invoices since 2010 and the total amount
invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceDate, SUM(Total) AS TotalAmount\nFROM invoic
es\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "content": "  \n    Get the
total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, c.Fi
rstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId
= i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName"}, {"role": "user", "content": "How many cus
tomers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user", "c
ontent": "  \n    List all genres and the number of tracks in each genre:\n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:44:46.196478939Z', 'message': {'role': 'assistan
t', 'content': 'SELECT g.Name, COUNT(t.TrackId) AS TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId =
t.GenreId\nGROUP BY g.Name'}, 'done_reason': 'stop', 'done': True, 'total_duration': 61828704896, 'load_dur
ation': 715559, 'prompt_eval_count': 1446, 'prompt_eval_duration': 51613831000, 'eval_count': 46, 'eval_dur
ation': 9544091000}
SELECT g.Name, COUNT(t.TrackId) AS TrackCount
FROM genres g
JOIN tracks t ON g.GenreId = t.GenreId
GROUP BY g.Name
SELECT g.Name, COUNT(t.TrackId) AS TrackCount
FROM genres g
JOIN tracks t ON g.GenreId = t.GenreId
GROUP BY g.Name
                  Name  TrackCount
0          Alternative          40
1   Alternative & Punk         332
2                Blues          81
3           Bossa Nova          15
4            Classical          74
5               Comedy          17
6                Drama          64
7       Easy Listening          24
```

```
8    Electronica/Dance          30
9          Heavy Metal          28
10         Hip Hop/Rap          35
11                Jazz         130
12               Latin         579
13               Metal         374
14               Opera           1
15                 Pop          48
16            R&B/Soul          61
17              Reggae          58
18                Rock        1297
19       Rock And Roll          12
20    Sci Fi & Fantasy          26
21     Science Fiction          13
22          Soundtrack          43
23            TV Shows          93
24               World          28
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    List all genres and the number of tracks in each genr
e:\n'\n\nThe DataFrame was produced using this query: SELECT g.Name, COUNT(t.TrackId) AS TrackCount\nFROM g
enres g\nJOIN tracks t ON g.GenreId = t.GenreId\nGROUP BY g.Name\n\nThe following is information about the
resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name          object\nTrackCount     int64\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': 'codegemma:latest', 'created_at': '2024-06-15T19:45:02.627897651Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Name', y='TrackCount')\n\nfi
g.update_layout(title='Number of Tracks by Genre')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': Tru
e, 'total_duration': 16403862802, 'load_duration': 679922, 'prompt_eval_count': 197, 'prompt_eval_duratio
n': 6944437000, 'eval_count': 50, 'eval_duration': 9325402000}
```


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

```
Out[29]:  ('SELECT g.Name, COUNT(t.TrackId) AS TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.GenreId\nGR
          OUP BY g.Name',
                             Name  TrackCount
           0          Alternative          40
           1   Alternative & Punk         332
           2                Blues          81
           3           Bossa Nova          15
           4            Classical          74
           5               Comedy          17
           6                Drama          64
           7       Easy Listening          24
           8    Electronica/Dance          30
           9          Heavy Metal          28
           10         Hip Hop/Rap          35
           11                Jazz         130
           12               Latin         579
           13               Metal         374
           14               Opera           1
           15                 Pop          48
           16            R&B/Soul          61
           17              Reggae          58
           18                Rock        1297
           19       Rock And Roll          12
           20     Sci Fi & Fantasy         26
           21      Science Fiction         13
           22          Soundtrack          43
           23            TV Shows          93
           24               World          28,
           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(['Alternative', 'Alternative & Punk', 'Blues', 'Bossa Nova', 'Classical',
                                     'Comedy', 'Drama', 'Easy Listening', 'Electronica/Dance', 'Heavy Metal',
                                     'Hip Hop/Rap', 'Jazz', 'Latin', 'Metal', 'Opera', 'Pop', 'R&B/Soul',
                                     'Reggae', 'Rock', 'Rock And Roll', 'Sci Fi & Fantasy',
                                     'Science Fiction', 'Soundtrack', 'TV Shows', 'World'], dtype=object),
                         'xaxis': 'x',
                         'y': array([  40,  332,   81,   15,   74,   17,   64,   24,   30,   28,   35,  130,
                                      579,  374,    1,   48,   61,   58, 1297,   12,   26,   13,   43,   93,
                                       28]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          '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 [30]:

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

vn.ask(question=question)
```

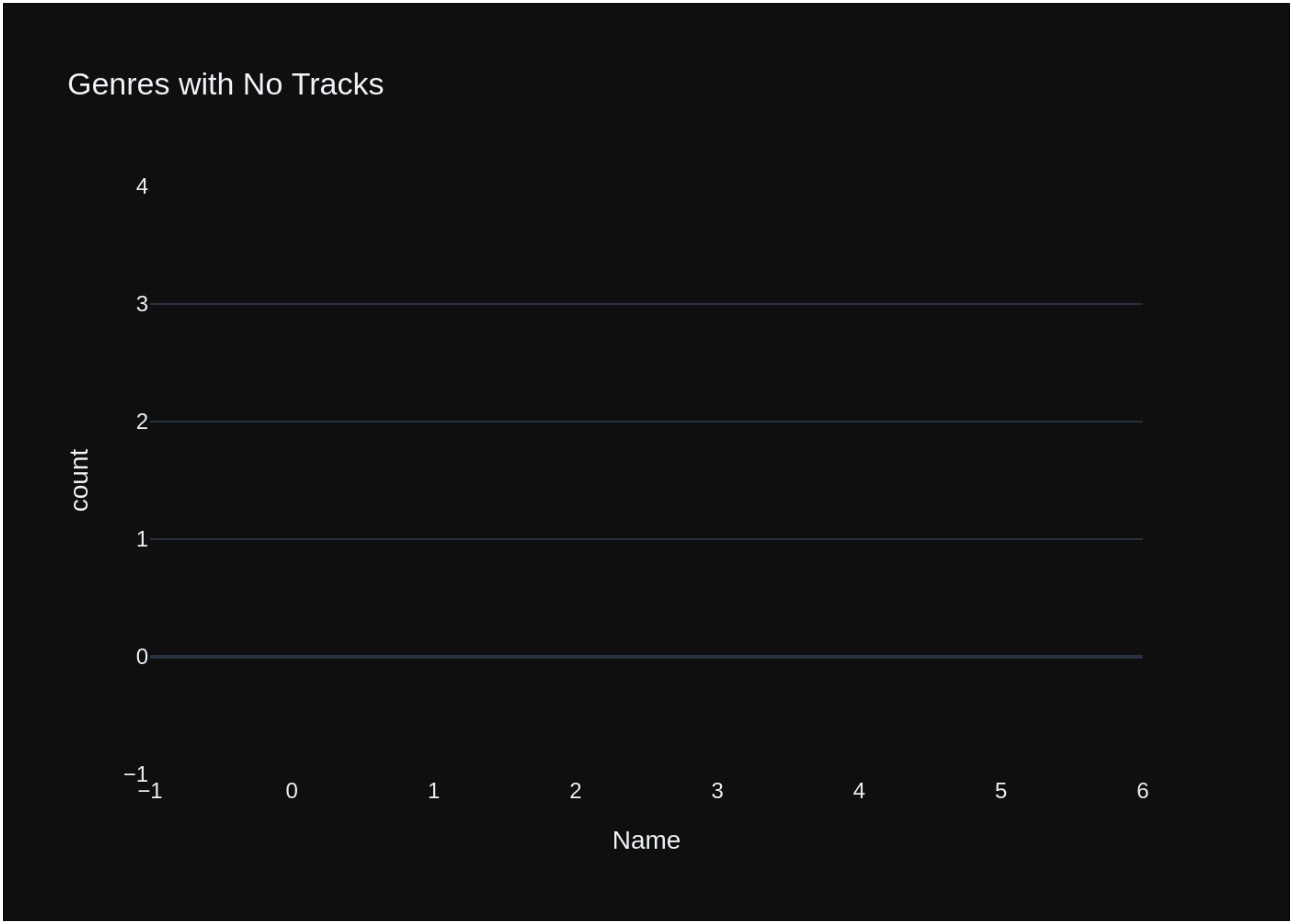
```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE TABLE "track
s"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n
AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(22
0),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NUL
L,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI
ON,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON
UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE INDE
X IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\n
CREATE TABLE "genres"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR
(120)\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES
"artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_trac
k"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_Playlis
tTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Playlist
Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (Trac
kId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (Ar
tistId)\n\nCREATE TABLE "playlists"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Re
sponse 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 st
rings 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). \n
5. If the question has been asked and answered before, please repeat the answer exactly as it was given bef
ore. \n'}, {'role': 'user', 'content': '  \n    List all genres and the number of tracks in each genr
e:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.TrackId) AS TrackCount\nFROM genres g\nJOI
N tracks t ON g.GenreId = t.GenreId\nGROUP BY g.Name'}, {'role': 'user', 'content': '  \n    Find all track
s with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tra
cks\nWHERE Name LIKE '%What%'"}, {'role': 'user', 'content': '  \n    List all albums and their correspondi
ng artist names  \n'}, {'role': 'assistant', 'content': 'SELECT albums.Title, artists.Name\nFROM albums\nJO
IN artists ON albums.ArtistId = artists.ArtistId'}, {'role': 'user', 'content': '  \n    Find the top 5 mos
t expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT Name, UnitPrice\nFR
OM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': 'Can you list all tables in the
SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master\nWHERE type =
'table'"}, {'role': 'user', 'content': '  \n    Find all invoices since 2010 and the total amount invoice
d:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceDate, SUM(Total) AS TotalAmount\nFROM invoices\nWHE
RE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': '  \n    List all invoi
ces with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices\nWHERE Total
> 10'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'as
```

```
sistant', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY
CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': "  \n    List all employees and their reporting
manager's name (if any):\n"}, {'role': 'assistant', 'content': "SELECT e.FirstName || ' ' || e.LastName AS
EmployeeName, m.FirstName || ' ' || m.LastName AS ManagerName\nFROM employees e\nLEFT JOIN employees m ON
e.ReportsTo = m.EmployeeId"}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assist
ant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': '  \n    Get all genres tha
t do not have any tracks associated with them:\n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE TABLE \"tr
acks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NUL
L,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NV
ARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  N
OT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE
NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDA
TE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE N
O ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_track\" (TrackId)
\n\nCREATE INDEX IFK_TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE INDEX IFK_TrackAlbumId ON \"tra
cks\" (AlbumId)\n\nCREATE TABLE \"genres\"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN K
EY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCRE
ATE TABLE \"playlist_track\"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r
\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFER
ENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Track
Id) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IF
K_AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n    PlaylistId INTEGER PRIMA
RY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook
database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please g
enerate a valid SQL query without any explanations for the question. \n2. If the provided context is almost
sufficient but requires knowledge of a specific string in a particular column, please generate an intermedi
ate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermed
iate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Pleas
e use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat th
e answer exactly as it was given before. \n"}, {"role": "user", "content": "  \n    List all genres and the
number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.TrackId) AS T
rackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"role": "user", "cont
```

```
ent": "  \n    Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistan
t", "content": "SELECT * FROM tracks\nWHERE Name LIKE '%What%'"}, {"role": "user", "content": "  \n    List
all albums and their corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT albums.Titl
e, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId = artists.ArtistId"}, {"role": "user", "conte
nt": "  \n    Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "conte
nt": "SELECT Name, UnitPrice\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content":
"Can you list all tables in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name F
ROM sqlite_master\nWHERE type = 'table'"}, {"role": "user", "content": "  \n    Find all invoices since 201
0 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceDate, SUM(Total) AS T
otalAmount\nFROM invoices\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "con
tent": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT
* FROM invoices\nWHERE Total > 10"}, {"role": "user", "content": "what are the top 5 countries that custome
rs come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM customer
s\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    List all
employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT e.Firs
tName || ' ' || e.LastName AS EmployeeName, m.FirstName || ' ' || m.LastName AS ManagerName\nFROM employees
e\nLEFT JOIN employees m ON e.ReportsTo = m.EmployeeId"}, {"role": "user", "content": "How many customers a
re there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content":
"  \n    Get all genres that do not have any tracks associated with them:\n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:46:03.068778736Z', 'message': {'role': 'assistan
t', 'content': 'SELECT g.Name\nFROM genres g\nLEFT JOIN tracks t ON g.GenreId = t.GenreId\nWHERE t.TrackId
IS NULL'}, 'done_reason': 'stop', 'done': True, 'total_duration': 60272659414, 'load_duration': 622328, 'pr
ompt_eval_count': 1450, 'prompt_eval_duration': 51738744000, 'eval_count': 38, 'eval_duration': 7859573000}
SELECT g.Name
FROM genres g
LEFT JOIN tracks t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
SELECT g.Name
FROM genres g
LEFT JOIN tracks t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
Empty DataFrame
Columns: [Name]
Index: []
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Get all genres that do not have any tracks associated wi
```

```
th them:\n'\n\nThe DataFrame was produced using this query: SELECT g.Name\nFROM genres g\nLEFT JOIN tracks
t ON g.GenreId = t.GenreId\nWHERE t.TrackId IS NULL\n\nThe following is information about the resulting pan
das DataFrame 'df': \nRunning df.dtypes gives:\n Name    object\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': 'codegemma:latest', 'created_at': '2024-06-15T19:46:18.876493705Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Name', title='Genres with No
Tracks')\n\nfig.update_traces(marker_color='red')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': Tru
e, 'total_duration': 15805503347, 'load_duration': 755559, 'prompt_eval_count': 183, 'prompt_eval_duratio
n': 6311203000, 'eval_count': 50, 'eval_duration': 9363019000}
```


Genres with No Tracks
count
4
3
2
1
0
−1
−1
0
1
2
3
4
5
6
Name

```
Out[30]: ('SELECT g.Name\nFROM genres g\nLEFT JOIN tracks t ON g.GenreId = t.GenreId\nWHERE t.TrackId IS NULL',
          Empty DataFrame
          Columns: [Name]
          Index: [],
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'Name=%{x}<br>count=%{y}<extra></extra>',
                        'legendgroup': '',
                        'marker': {'color': 'red', 'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array([], dtype=object),
                        'xaxis': 'x',
                        'y': array([], dtype=int64),
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                         'legend': {'tracegroupgap': 0},
                         'template': '...',
                         'title': {'text': 'Genres with No Tracks'},
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'count'}}}
          }))
```

```
In [31]: question = """
             List all customers who have not placed any orders:
         """

         vn.ask(question=question)
```

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

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the 
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingA
ddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCount
ry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIG
N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Firs
tName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    A
ddress NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n 
PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT
NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineI
d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  N
OT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (I
nvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN
KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE 
TABLE "employees"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCH
AR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTE
GER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR
(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone 
NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES
"employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_tra
ck"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_Playli
stTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Playlis
tId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (Tra
ckId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INT
EGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT
NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "playl
ists"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n
CREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR 
(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n 
Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMER
IC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION 
ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DEL
ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\n
\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If 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 bee
n asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'use
r', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content':
'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\n
LIMIT 5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'S
ELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': '  \n    Get the total number of invoices for
each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.
InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.C
ustomerId, c.FirstName, c.LastName'}, {'role': 'user', 'content': '  \n    Find the total number of invoice
s per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices
\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user',
'content': '  \n    Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content':
'SELECT c.CustomerId, c.FirstName, c.LastName, AVG(i.Total) AS AverageInvoiceTotal\nFROM customers c\nJOIN
invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName'}, {'role': 'use
r', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content':
'SELECT * FROM invoices\nWHERE Total > 10'}, {'role': 'user', 'content': '  \n    Find all invoices since 2
010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceDate, SUM(Total) AS
TotalAmount\nFROM invoices\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'co
ntent': "  \n    List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant',
'content': "SELECT e.FirstName || ' ' || e.LastName AS EmployeeName, m.FirstName || ' ' || m.LastName AS Ma
nagerName\nFROM employees e\nLEFT JOIN employees m ON e.ReportsTo = m.EmployeeId"}, {'role': 'user', 'conte
nt': '  \n    List all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content':
'SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId = artists.ArtistId'}, {'ro
le': 'user', 'content': '  \n    Find the top 5 most expensive tracks (based on unit price):\n'}, {'role':
'assistant', 'content': 'SELECT Name, UnitPrice\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role':
'user', 'content': '  \n    List all customers who have not placed any orders:\n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    Billin
gAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCou
ntry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FORE
IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n
Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n
PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT
NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n    InvoiceL
ineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGE
R  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KE
Y (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    F
OREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n
\nCREATE TABLE \"employees\"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Last
Name NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    Rep
ortsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    Ci
ty NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r
\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsT
o) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TAB
LE \"playlist_track\"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CO
NSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES
\"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) RE
FERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums
\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r
\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (Supp
ortRepId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER
NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n
Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\"
(AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres
\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES
\"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_Invo
iceCustomerId ON \"invoices\" (CustomerId)\n\n\n===Additional Context \n\nIn the chinook database invoice m
eans order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQ
L query without any explanations for the question. \n2. If the provided context is almost sufficient but re
quires knowledge of a specific string in a particular column, please generate an intermediate SQL query to
find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If
the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most re
levant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly
as it was given before. \n"}, {"role": "user", "content": "what are the top 5 countries that customers come
from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP
BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "How many customers are the
re"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "  \n
Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.Customer
Id, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.Cu
stomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName"}, {"role": "user", "content": "
```

```
\n    Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Countr
y, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nG
ROUP BY c.Country"}, {"role": "user", "content": "  \n    Get the average invoice total for each custome
r:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, c.FirstName, c.LastName, AVG(i.Total) AS Ave
rageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId,
c.FirstName, c.LastName"}, {"role": "user", "content": "  \n    List all invoices with a total exceeding $1
0:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices\nWHERE Total > 10"}, {"role": "user", "con
tent": "  \n    Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "con
tent": "SELECT InvoiceDate, SUM(Total) AS TotalAmount\nFROM invoices\nWHERE InvoiceDate >= '2010-01-01'\nGR
OUP BY InvoiceDate"}, {"role": "user", "content": "  \n    List all employees and their reporting manager's
name (if any):\n"}, {"role": "assistant", "content": "SELECT e.FirstName || ' ' || e.LastName AS EmployeeNa
me, m.FirstName || ' ' || m.LastName AS ManagerName\nFROM employees e\nLEFT JOIN employees m ON e.ReportsTo
= m.EmployeeId"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist names
\n"}, {"role": "assistant", "content": "SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists ON alb
ums.ArtistId = artists.ArtistId"}, {"role": "user", "content": "  \n    Find the top 5 most expensive track
s (based on unit price):\n"}, {"role": "assistant", "content": "SELECT Name, UnitPrice\nFROM tracks\nORDER
BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    List all customers who have not placed a
ny orders:\n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:47:34.980711241Z', 'message': {'role': 'assistan
t', 'content': 'SELECT * FROM customers\nWHERE CustomerId'}, 'done_reason': 'stop', 'done': True, 'total_du
ration': 76008212340, 'load_duration': 765264, 'prompt_eval_count': 2030, 'prompt_eval_duration': 735305100
00, 'eval_count': 9, 'eval_duration': 1815757000}
SELECT * FROM customers
WHERE CustomerId
SELECT * FROM customers
WHERE CustomerId
    CustomerId  FirstName      LastName  \
0            1       Luís     Gonçalves
1            2     Leonie        Köhler
2            3   François      Tremblay
3            4      Bjørn        Hansen
4            5  František   Wichterlová
5            6     Helena          Holý
6            7     Astrid        Gruber
7            8       Daan       Peeters
8            9       Kara       Nielsen
9           10    Eduardo       Martins
10          11  Alexandre         Rocha
11          12    Roberto       Almeida
12          13   Fernanda         Ramos
13          14       Mark       Philips
```

```
14          15   Jennifer      Peterson
15          16      Frank        Harris
16          17       Jack         Smith
17          18   Michelle        Brooks
18          19        Tim         Goyer
19          20        Dan        Miller
20          21      Kathy         Chase
21          22    Heather       Leacock
22          23       John        Gordon
23          24      Frank       Ralston
24          25     Victor       Stevens
25          26    Richard    Cunningham
26          27    Patrick          Gray
27          28      Julia       Barnett
28          29     Robert         Brown
29          30     Edward       Francis
30          31     Martha          Silk
31          32      Aaron      Mitchell
32          33      Ellie      Sullivan
33          34       João     Fernandes
34          35   Madalena       Sampaio
35          36     Hannah     Schneider
36          37       Fynn    Zimmermann
37          38     Niklas      Schröder
38          39    Camille       Bernard
39          40  Dominique      Lefebvre
40          41       Marc        Dubois
41          42      Wyatt        Girard
42          43   Isabelle       Mercier
43          44      Terhi    Hämäläinen
44          45   Ladislav        Kovács
45          46       Hugh      O'Reilly
46          47      Lucas       Mancini
47          48   Johannes  Van der Berg
48          49  Stanisław        Wójcik
49          50    Enrique         Muñoz
50          51     Joakim     Johansson
51          52       Emma         Jones
52          53       Phil        Hughes
53          54      Steve        Murray
54          55       Mark        Taylor
55          56      Diego     Gutiérrez
```

```
56          57       Luis         Rojas
57          58      Manoj        Pareek
58          59       Puja    Srivastava

                                             Company  \
0   Embraer - Empresa Brasileira de Aeronáutica S.A.
1                                               None
2                                               None
3                                               None
4                                   JetBrains s.r.o.
5                                               None
6                                               None
7                                               None
8                                               None
9                                   Woodstock Discos
10                              Banco do Brasil S.A.
11                                            Riotur
12                                              None
13                                             Telus
14                                     Rogers Canada
15                                       Google Inc.
16                             Microsoft Corporation
17                                              None
18                                        Apple Inc.
19                                              None
20                                              None
21                                              None
22                                              None
23                                              None
24                                              None
25                                              None
26                                              None
27                                              None
28                                              None
29                                              None
30                                              None
31                                              None
32                                              None
33                                              None
34                                              None
35                                              None
36                                              None
```

```
37                                              None
38                                              None
39                                              None
40                                              None
41                                              None
42                                              None
43                                              None
44                                              None
45                                              None
46                                              None
47                                              None
48                                              None
49                                              None
50                                              None
51                                              None
52                                              None
53                                              None
54                                              None
55                                              None
56                                              None
57                                              None
58                                              None

                                     Address                 City   State  \
0            Av. Brigadeiro Faria Lima, 2170  São José dos Campos      SP   
1                    Theodor-Heuss-Straße 34            Stuttgart    None   
2                          1498 rue Bélanger             Montréal      QC   
3                           Ullevålsveien 14                 Oslo    None   
4                              Klanova 9/506               Prague    None   
5                              Rilská 3174/6               Prague    None   
6       Rotenturmstraße 4, 1010 Innere Stadt               Vienne    None   
7                            Grétrystraat 63             Brussels    None   
8                        Sønder Boulevard 51           Copenhagen    None   
9                  Rua Dr. Falcão Filho, 155            São Paulo      SP   
10                        Av. Paulista, 2022            São Paulo      SP   
11                          Praça Pio X, 119       Rio de Janeiro      RJ   
12                              Qe 7 Bloco G             Brasília      DF   
13                            8210 111 ST NW             Edmonton      AB   
14                       700 W Pender Street            Vancouver      BC   
15                 1600 Amphitheatre Parkway        Mountain View      CA   
16                           1 Microsoft Way              Redmond      WA   
17                              627 Broadway             New York      NY   
```

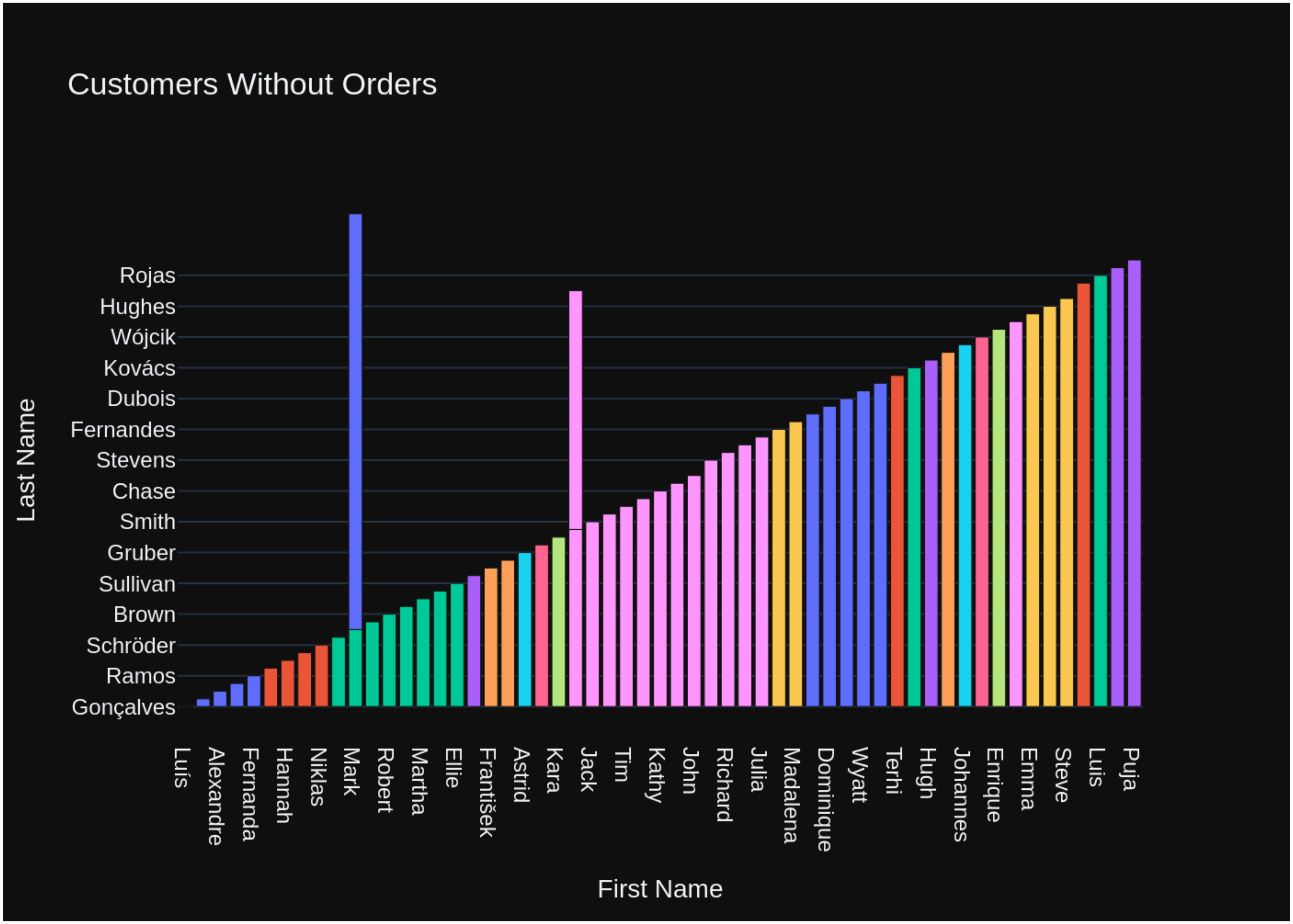
```
18                           1 Infinite Loop            Cupertino      CA
19                      541 Del Medio Avenue        Mountain View      CA
20                          801 W 4th Street                 Reno      NV
21                          120 S Orange Ave              Orlando      FL
22                           69 Salem Street               Boston      MA
23                     162 E Superior Street              Chicago      IL
24                     319 N. Frances Street              Madison      WI
25                       2211 W Berry Street           Fort Worth      TX
26                           1033 N Park Ave               Tucson      AZ
27                               302 S 700 E       Salt Lake City      UT
28                    796 Dundas Street West              Toronto      ON
29                          230 Elgin Street               Ottawa      ON
30                     194A Chain Lake Drive              Halifax      NS
31                        696 Osborne Street             Winnipeg      MB
32                            5112 48 Street          Yellowknife      NT
33                        Rua da Assunção 53               Lisbon    None
34  Rua dos Campeões Europeus de Viena, 4350                Porto    None
35                        Tauentzienstraße 8               Berlin    None
36                          Berger Straße 10            Frankfurt    None
37                       Barbarossastraße 19               Berlin    None
38                             4, Rue Milton                Paris    None
39                            8, Rue Hanovre                Paris    None
40                       11, Place Bellecour                 Lyon    None
41                    9, Place Louis Barthou             Bordeaux    None
42                          68, Rue Jouvence                Dijon    None
43                           Porthaninkatu 9             Helsinki    None
44                         Erzsébet krt. 58.             Budapest    None
45                          3 Chatham Street               Dublin  Dublin
46                    Via Degli Scipioni, 43                 Rome      RM
47                     Lijnbaansgracht 120bg            Amsterdam      VV
48                              Ordynacka 10               Warsaw    None
49                        C/ San Bernardo 85               Madrid    None
50                               Celsiusg. 9            Stockholm    None
51                         202 Hoxton Street               London    None
52                              113 Lupus St               London    None
53                            110 Raeburn Pl           Edinburgh     None
54                         421 Bourke Street               Sidney     NSW
55                        307 Macacha Güemes         Buenos Aires    None
56                           Calle Lira, 198             Santiago    None
57                       12,Community Centre                Delhi    None
58                         3,Raj Bhavan Road            Bangalore    None
```

```
           Country  PostalCode               Phone                 Fax  \
0           Brazil   12227-000  +55 (12) 3923-5555  +55 (12) 3923-5566   
1          Germany       70174    +49 0711 2842222                None   
2           Canada     H2G 1A7   +1 (514) 721-4711                None   
3           Norway        0171     +47 22 44 22 22                None   
4   Czech Republic       14700    +420 2 4172 5555    +420 2 4172 5555   
5   Czech Republic       14300    +420 2 4177 0449                None   
6          Austria        1010      +43 01 5134505                None   
7          Belgium        1000    +32 02 219 03 03                None   
8          Denmark        1720      +453 3331 9991                None   
9           Brazil   01007-010  +55 (11) 3033-5446  +55 (11) 3033-4564   
10          Brazil   01310-200  +55 (11) 3055-3278  +55 (11) 3055-8131   
11          Brazil   20040-020  +55 (21) 2271-7000  +55 (21) 2271-7070   
12          Brazil   71020-677  +55 (61) 3363-5547  +55 (61) 3363-7855   
13          Canada     T6G 2C7   +1 (780) 434-4554   +1 (780) 434-5565   
14          Canada     V6C 1G8   +1 (604) 688-2255   +1 (604) 688-8756   
15             USA  94043-1351   +1 (650) 253-0000   +1 (650) 253-0000   
16             USA  98052-8300   +1 (425) 882-8080   +1 (425) 882-8081   
17             USA  10012-2612   +1 (212) 221-3546   +1 (212) 221-4679   
18             USA       95014   +1 (408) 996-1010   +1 (408) 996-1011   
19             USA   94040-111   +1 (650) 644-3358                None   
20             USA       89503   +1 (775) 223-7665                None   
21             USA       32801   +1 (407) 999-7788                None   
22             USA        2113   +1 (617) 522-1333                None   
23             USA       60611   +1 (312) 332-3232                None   
24             USA       53703   +1 (608) 257-0597                None   
25             USA       76110   +1 (817) 924-7272                None   
26             USA       85719   +1 (520) 622-4200                None   
27             USA       84102   +1 (801) 531-7272                None   
28          Canada     M6J 1V1   +1 (416) 363-8888                None   
29          Canada     K2P 1L7   +1 (613) 234-3322                None   
30          Canada     B3S 1C5   +1 (902) 450-0450                None   
31          Canada     R3L 2B9   +1 (204) 452-6452                None   
32          Canada     X1A 1N6   +1 (867) 920-2233                None   
33        Portugal        None  +351 (213) 466-111                None   
34        Portugal        None  +351 (225) 022-448                None   
35         Germany       10789    +49 030 26550280                None   
36         Germany       60316    +49 069 40598889                None   
37         Germany       10779     +49 030 2141444                None   
38          France       75009  +33 01 49 70 65 65                None   
39          France       75002  +33 01 47 42 71 71                None   
40          France       69002  +33 04 78 30 30 30                None   
```

```
41          France       33000   +33 05 56 96 96 96                None
42          France       21000   +33 03 80 73 66 99                None
43         Finland       00530     +358 09 870 2000                None
44         Hungary      H-1073                 None                None
45         Ireland        None      +353 01 6792424                None
46           Italy       00192      +39 06 39733434                None
47     Netherlands        1016      +31 020 6223130                None
48          Poland      00-358     +48 22 828 37 39                None
49           Spain       28015      +34 914 454 454                None
50          Sweden       11230     +46 08-651 52 52                None
51  United Kingdom      N1 5LH    +44 020 7707 0707                None
52  United Kingdom    SW1V 3EN    +44 020 7976 5722                None
53  United Kingdom     EH4 1HH    +44 0131 315 3300                None
54       Australia        2010   +61 (02) 9332 3633                None
55       Argentina        1106  +54 (0)11 4311 4333                None
56           Chile        None    +56 (0)2 635 4444                None
57           India      110017    +91 0124 39883988                None
58           India      560001     +91 080 22289999                None

                            Email  SupportRepId
0            luisg@embraer.com.br             3
1           leonekohler@surfeu.de             5
2             ftremblay@gmail.com             3
3           bjorn.hansen@yahoo.no             4
4        frantisekw@jetbrains.com             4
5                 hholy@gmail.com             5
6          astrid.gruber@apple.at             5
7           daan_peeters@apple.be             4
8           kara.nielsen@jubii.dk             4
9        eduardo@woodstock.com.br             4
10               alero@uol.com.br             5
11  roberto.almeida@riotur.gov.br             3
12       fernadaramos4@uol.com.br             4
13             mphilips12@shaw.ca             5
14            jenniferp@rogers.ca             3
15             fharris@google.com             4
16        jacksmith@microsoft.com             5
17              michelleb@aol.com             3
18               tgoyer@apple.com             3
19            dmiller@comcast.com             4
20            kachase@hotmail.com             5
21             hleacock@gmail.com             4
```

```
22         johngordon22@yahoo.com             4
23             fralston@gmail.com             3
24             vstevens@yahoo.com             5
25       ricunningham@hotmail.com             4
26           patrick.gray@aol.com             4
27            jubarnett@gmail.com             5
28               robbrown@shaw.ca             3
29            edfrancis@yachoo.ca             3
30           marthasilk@gmail.com             5
31         aaronmitchell@yahoo.ca             4
32         ellie.sullivan@shaw.ca             3
33            jfernandes@yahoo.pt             4
34              masampaio@sapo.pt             4
35      hannah.schneider@yahoo.de             5
36           fzimmermann@yahoo.de             3
37            nschroder@surfeu.de             3
38       camille.bernard@yahoo.fr             4
39    dominiquelefebvre@gmail.com             4
40        marc.dubois@hotmail.com             5
41          wyatt.girard@yahoo.fr             3
42      isabelle_mercier@apple.fr             3
43      terhi.hamalainen@apple.fi             3
44       ladislav_kovacs@apple.hu             3
45           hughoreilly@apple.ie             3
46         lucas.mancini@yahoo.it             5
47        johavanderberg@yahoo.nl             5
48         stanisław.wójcik@wp.pl             4
49         enrique_munoz@yahoo.es             5
50      joakim.johansson@yahoo.se             5
51         emma_jones@hotmail.com             3
52          phil.hughes@gmail.com             3
53          steve.murray@yahoo.uk             5
54           mark.taylor@yahoo.au             4
55       diego.gutierrez@yahoo.ar             4
56             luisrojas@yahoo.cl             5
57        manoj.pareek@rediff.com             3
58       puja_srivastava@yahoo.in             3
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
```

```
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: '  \n    List all customers who have not placed any orders:\n'\n\nThe DataFrame was produced using this query: SELECT * FROM customers\nWHERE CustomerId\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId       int64\nFirstName       object\nLastName        object\nCompany         object\nAddress         object\nCity            object\nState           object\nCountry         object\nPostalCode      object\nPhone           object\nFax             object\nEmail           object\nSupportRepId     int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:47:57.753873044Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='FirstName', y='LastName', color='Country',\n             labels={'FirstName': 'First Name', 'LastName': 'Last Name', 'Country': 'Country'},\n             title='Customers Without Orders')\n\nfig.update_layout(showlegend=False)\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 22745824406, 'load_duration': 776752, 'prompt_eval_count': 217, 'prompt_eval_duration': 7353925000, 'eval_count': 81, 'eval_duration': 15261597000}
```


Customers Without Orders
Last Name
Rojas
Hughes
Wójcik
Kovács
Dubois
Fernandes
Stevens
Chase
Smith
Gruber
Sullivan
Brown
Schröder
Ramos
Gonçalves
Luís
Alexandre
Fernanda
Hannah
Niklas
Mark
Robert
Martha
Ellie
František
Astrid
Kara
Jack
Tim
Kathy
John
Richard
Julia
Madalena
Dominique
Wyatt
Terhi
Hugh
Johannes
Enrique
Emma
Steve
Luis
Puja
First Name

```
Out[31]:  ('SELECT * FROM customers\nWHERE CustomerId',
               CustomerId  FirstName      LastName  \
           0            1       Luís     Gonçalves
           1            2     Leonie        Köhler
           2            3   François      Tremblay
           3            4      Bjørn        Hansen
           4            5  František   Wichterlová
           5            6     Helena          Holý
           6            7     Astrid        Gruber
           7            8       Daan       Peeters
           8            9       Kara       Nielsen
           9           10    Eduardo       Martins
           10          11  Alexandre         Rocha
           11          12    Roberto       Almeida
           12          13   Fernanda         Ramos
           13          14       Mark       Philips
           14          15   Jennifer      Peterson
           15          16      Frank        Harris
           16          17       Jack         Smith
           17          18   Michelle        Brooks
           18          19        Tim         Goyer
           19          20        Dan        Miller
           20          21      Kathy         Chase
           21          22    Heather       Leacock
           22          23       John        Gordon
           23          24      Frank       Ralston
           24          25     Victor       Stevens
           25          26    Richard    Cunningham
           26          27    Patrick          Gray
           27          28      Julia       Barnett
           28          29     Robert         Brown
           29          30     Edward       Francis
           30          31     Martha          Silk
           31          32      Aaron      Mitchell
           32          33      Ellie      Sullivan
           33          34       João     Fernandes
           34          35   Madalena       Sampaio
           35          36     Hannah     Schneider
           36          37       Fynn    Zimmermann
           37          38     Niklas      Schröder
           38          39    Camille       Bernard
           39          40  Dominique      Lefebvre
```

```
40          41       Marc        Dubois
41          42      Wyatt        Girard
42          43   Isabelle       Mercier
43          44      Terhi    Hämäläinen
44          45   Ladislav        Kovács
45          46       Hugh      O'Reilly
46          47      Lucas       Mancini
47          48   Johannes  Van der Berg
48          49  Stanisław        Wójcik
49          50    Enrique         Muñoz
50          51     Joakim     Johansson
51          52       Emma         Jones
52          53       Phil        Hughes
53          54      Steve        Murray
54          55       Mark        Taylor
55          56      Diego     Gutiérrez
56          57       Luis         Rojas
57          58      Manoj        Pareek
58          59       Puja    Srivastava

                                             Company  \
0   Embraer - Empresa Brasileira de Aeronáutica S.A.
1                                               None
2                                               None
3                                               None
4                                   JetBrains s.r.o.
5                                               None
6                                               None
7                                               None
8                                               None
9                                   Woodstock Discos
10                              Banco do Brasil S.A.
11                                            Riotur
12                                              None
13                                             Telus
14                                     Rogers Canada
15                                       Google Inc.
16                             Microsoft Corporation
17                                              None
18                                        Apple Inc.
19                                              None
20                                              None
```

```
21                                              None
22                                              None
23                                              None
24                                              None
25                                              None
26                                              None
27                                              None
28                                              None
29                                              None
30                                              None
31                                              None
32                                              None
33                                              None
34                                              None
35                                              None
36                                              None
37                                              None
38                                              None
39                                              None
40                                              None
41                                              None
42                                              None
43                                              None
44                                              None
45                                              None
46                                              None
47                                              None
48                                              None
49                                              None
50                                              None
51                                              None
52                                              None
53                                              None
54                                              None
55                                              None
56                                              None
57                                              None
58                                              None

                                     Address                 City   State  \
0            Av. Brigadeiro Faria Lima, 2170  São José dos Campos      SP
1                    Theodor-Heuss-Straße 34            Stuttgart    None
```

```
2                          1498 rue Bélanger             Montréal      QC
3                           Ullevålsveien 14                 Oslo    None
4                              Klanova 9/506               Prague    None
5                              Rilská 3174/6               Prague    None
6       Rotenturmstraße 4, 1010 Innere Stadt               Vienne    None
7                            Grétrystraat 63             Brussels    None
8                        Sønder Boulevard 51           Copenhagen    None
9                  Rua Dr. Falcão Filho, 155            São Paulo      SP
10                        Av. Paulista, 2022            São Paulo      SP
11                          Praça Pio X, 119       Rio de Janeiro      RJ
12                              Qe 7 Bloco G             Brasília      DF
13                            8210 111 ST NW             Edmonton      AB
14                       700 W Pender Street            Vancouver      BC
15                 1600 Amphitheatre Parkway        Mountain View      CA
16                           1 Microsoft Way              Redmond      WA
17                              627 Broadway             New York      NY
18                           1 Infinite Loop            Cupertino      CA
19                      541 Del Medio Avenue        Mountain View      CA
20                          801 W 4th Street                 Reno      NV
21                          120 S Orange Ave              Orlando      FL
22                           69 Salem Street               Boston      MA
23                     162 E Superior Street              Chicago      IL
24                     319 N. Frances Street              Madison      WI
25                       2211 W Berry Street           Fort Worth      TX
26                           1033 N Park Ave               Tucson      AZ
27                               302 S 700 E       Salt Lake City      UT
28                    796 Dundas Street West              Toronto      ON
29                          230 Elgin Street               Ottawa      ON
30                     194A Chain Lake Drive              Halifax      NS
31                        696 Osborne Street             Winnipeg      MB
32                            5112 48 Street          Yellowknife      NT
33                        Rua da Assunção 53               Lisbon    None
34  Rua dos Campeões Europeus de Viena, 4350                Porto    None
35                        Tauentzienstraße 8               Berlin    None
36                          Berger Straße 10            Frankfurt    None
37                       Barbarossastraße 19               Berlin    None
38                             4, Rue Milton                Paris    None
39                            8, Rue Hanovre                Paris    None
40                       11, Place Bellecour                 Lyon    None
41                    9, Place Louis Barthou             Bordeaux    None
42                         68, Rue Jouvence                 Dijon    None
43                           Porthaninkatu 9             Helsinki    None
```

```
44                         Erzsébet krt. 58.             Budapest    None
45                          3 Chatham Street               Dublin  Dublin
46                    Via Degli Scipioni, 43                 Rome      RM
47                     Lijnbaansgracht 120bg            Amsterdam      VV
48                              Ordynacka 10               Warsaw    None
49                        C/ San Bernardo 85               Madrid    None
50                               Celsiusg. 9            Stockholm    None
51                         202 Hoxton Street               London    None
52                              113 Lupus St               London    None
53                            110 Raeburn Pl           Edinburgh     None
54                         421 Bourke Street               Sidney     NSW
55                        307 Macacha Güemes         Buenos Aires    None
56                           Calle Lira, 198             Santiago    None
57                       12,Community Centre                Delhi    None
58                         3,Raj Bhavan Road            Bangalore    None

           Country  PostalCode                Phone                 Fax  \
0           Brazil   12227-000   +55 (12) 3923-5555  +55 (12) 3923-5566   
1          Germany       70174     +49 0711 2842222                None   
2           Canada     H2G 1A7    +1 (514) 721-4711                None   
3           Norway        0171      +47 22 44 22 22                None   
4   Czech Republic       14700     +420 2 4172 5555    +420 2 4172 5555   
5   Czech Republic       14300     +420 2 4177 0449                None   
6          Austria        1010       +43 01 5134505                None   
7          Belgium        1000     +32 02 219 03 03                None   
8          Denmark        1720       +453 3331 9991                None   
9           Brazil   01007-010   +55 (11) 3033-5446  +55 (11) 3033-4564   
10          Brazil   01310-200   +55 (11) 3055-3278  +55 (11) 3055-8131   
11          Brazil   20040-020   +55 (21) 2271-7000  +55 (21) 2271-7070   
12          Brazil   71020-677   +55 (61) 3363-5547  +55 (61) 3363-7855   
13          Canada     T6G 2C7    +1 (780) 434-4554   +1 (780) 434-5565   
14          Canada     V6C 1G8    +1 (604) 688-2255   +1 (604) 688-8756   
15             USA  94043-1351    +1 (650) 253-0000   +1 (650) 253-0000   
16             USA  98052-8300    +1 (425) 882-8080   +1 (425) 882-8081   
17             USA  10012-2612    +1 (212) 221-3546   +1 (212) 221-4679   
18             USA       95014    +1 (408) 996-1010   +1 (408) 996-1011   
19             USA   94040-111    +1 (650) 644-3358                None   
20             USA       89503    +1 (775) 223-7665                None   
21             USA       32801    +1 (407) 999-7788                None   
22             USA        2113    +1 (617) 522-1333                None   
23             USA       60611    +1 (312) 332-3232                None   
24             USA       53703    +1 (608) 257-0597                None   
```

```
25             USA       76110    +1 (817) 924-7272                None
26             USA       85719    +1 (520) 622-4200                None
27             USA       84102    +1 (801) 531-7272                None
28          Canada     M6J 1V1    +1 (416) 363-8888                None
29          Canada     K2P 1L7    +1 (613) 234-3322                None
30          Canada     B3S 1C5    +1 (902) 450-0450                None
31          Canada     R3L 2B9    +1 (204) 452-6452                None
32          Canada     X1A 1N6    +1 (867) 920-2233                None
33        Portugal        None   +351 (213) 466-111                None
34        Portugal        None   +351 (225) 022-448                None
35         Germany       10789     +49 030 26550280                None
36         Germany       60316     +49 069 40598889                None
37         Germany       10779      +49 030 2141444                None
38          France       75009   +33 01 49 70 65 65                None
39          France       75002   +33 01 47 42 71 71                None
40          France       69002   +33 04 78 30 30 30                None
41          France       33000   +33 05 56 96 96 96                None
42          France       21000   +33 03 80 73 66 99                None
43         Finland       00530     +358 09 870 2000                None
44         Hungary      H-1073                 None                None
45         Ireland        None      +353 01 6792424                None
46           Italy       00192      +39 06 39733434                None
47     Netherlands        1016      +31 020 6223130                None
48          Poland      00-358     +48 22 828 37 39                None
49           Spain       28015      +34 914 454 454                None
50          Sweden       11230     +46 08-651 52 52                None
51  United Kingdom      N1 5LH    +44 020 7707 0707                None
52  United Kingdom    SW1V 3EN    +44 020 7976 5722                None
53  United Kingdom     EH4 1HH    +44 0131 315 3300                None
54       Australia        2010   +61 (02) 9332 3633                None
55       Argentina        1106  +54 (0)11 4311 4333                None
56           Chile        None    +56 (0)2 635 4444                None
57           India      110017    +91 0124 39883988                None
58           India      560001     +91 080 22289999                None

                            Email  SupportRepId
0            luisg@embraer.com.br             3
1           leonekohler@surfeu.de             5
2             ftremblay@gmail.com             3
3           bjorn.hansen@yahoo.no             4
4        frantisekw@jetbrains.com             4
5                 hholy@gmail.com             5
```

```
6          astrid.gruber@apple.at             5
7           daan_peeters@apple.be             4
8           kara.nielsen@jubii.dk             4
9        eduardo@woodstock.com.br             4
10              alero@uol.com.br              5
11  roberto.almeida@riotur.gov.br             3
12       fernadaramos4@uol.com.br             4
13             mphilips12@shaw.ca             5
14            jenniferp@rogers.ca             3
15             fharris@google.com             4
16        jacksmith@microsoft.com             5
17              michelleb@aol.com             3
18               tgoyer@apple.com             3
19            dmiller@comcast.com             4
20            kachase@hotmail.com             5
21             hleacock@gmail.com             4
22          johngordon22@yahoo.com            4
23             fralston@gmail.com             3
24             vstevens@yahoo.com             5
25       ricunningham@hotmail.com             4
26           patrick.gray@aol.com             4
27            jubarnett@gmail.com             5
28               robbrown@shaw.ca             3
29            edfrancis@yachoo.ca             3
30           marthasilk@gmail.com             5
31          aaronmitchell@yahoo.ca            4
32          ellie.sullivan@shaw.ca            3
33            jfernandes@yahoo.pt             4
34              masampaio@sapo.pt             4
35      hannah.schneider@yahoo.de             5
36           fzimmermann@yahoo.de             3
37            nschroder@surfeu.de             3
38       camille.bernard@yahoo.fr             4
39    dominiquelefebvre@gmail.com             4
40        marc.dubois@hotmail.com             5
41           wyatt.girard@yahoo.fr            3
42      isabelle_mercier@apple.fr             3
43      terhi.hamalainen@apple.fi             3
44       ladislav_kovacs@apple.hu             3
45           hughoreilly@apple.ie             3
46          lucas.mancini@yahoo.it            5
47        johavanderberg@yahoo.nl             5
```

```
48         stanisław.wójcik@wp.pl             4
49         enrique_munoz@yahoo.es             5
50      joakim.johansson@yahoo.se             5
51         emma_jones@hotmail.com             3
52          phil.hughes@gmail.com             3
53          steve.murray@yahoo.uk             5
54           mark.taylor@yahoo.au             4
55       diego.gutierrez@yahoo.ar             4
56             luisrojas@yahoo.cl             5
57        manoj.pareek@rediff.com             3
58       puja_srivastava@yahoo.in             3  ,
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'Country=Brazil<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
              'legendgroup': 'Brazil',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': 'Brazil',
              'offsetgroup': 'Brazil',
              'orientation': 'v',
              'showlegend': True,
              'textposition': 'auto',
              'type': 'bar',
              'x': array(['Luís', 'Eduardo', 'Alexandre', 'Roberto', 'Fernanda'], dtype=object),
              'xaxis': 'x',
              'y': array(['Gonçalves', 'Martins', 'Rocha', 'Almeida', 'Ramos'], dtype=object),
              'yaxis': 'y'},
             {'alignmentgroup': 'True',
              'hovertemplate': 'Country=Germany<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
              'legendgroup': 'Germany',
              'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
              'name': 'Germany',
              'offsetgroup': 'Germany',
              'orientation': 'v',
              'showlegend': True,
              'textposition': 'auto',
              'type': 'bar',
              'x': array(['Leonie', 'Hannah', 'Fynn', 'Niklas'], dtype=object),
              'xaxis': 'x',
              'y': array(['Köhler', 'Schneider', 'Zimmermann', 'Schröder'], dtype=object),
              'yaxis': 'y'},
             {'alignmentgroup': 'True',
              'hovertemplate': 'Country=Canada<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
                                  'legendgroup': 'Canada',
                                  'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
                                  'name': 'Canada',
                                  'offsetgroup': 'Canada',
                                  'orientation': 'v',
                                  'showlegend': True,
                                  'textposition': 'auto',
                                  'type': 'bar',
                                  'x': array(['François', 'Mark', 'Jennifer', 'Robert', 'Edward', 'Martha', 'Aaron',
                                              'Ellie'], dtype=object),
                                  'xaxis': 'x',
                                  'y': array(['Tremblay', 'Philips', 'Peterson', 'Brown', 'Francis', 'Silk',
                                              'Mitchell', 'Sullivan'], dtype=object),
                                  'yaxis': 'y'},
                                 {'alignmentgroup': 'True',
                                  'hovertemplate': 'Country=Norway<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
                                  'legendgroup': 'Norway',
                                  'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
                                  'name': 'Norway',
                                  'offsetgroup': 'Norway',
                                  'orientation': 'v',
                                  'showlegend': True,
                                  'textposition': 'auto',
                                  'type': 'bar',
                                  'x': array(['Bjørn'], dtype=object),
                                  'xaxis': 'x',
                                  'y': array(['Hansen'], dtype=object),
                                  'yaxis': 'y'},
                                 {'alignmentgroup': 'True',
                                  'hovertemplate': 'Country=Czech Republic<br>First Name=%{x}<br>Last Name=%{y}<extra></extra
                  >',
                                  'legendgroup': 'Czech Republic',
                                  'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
                                  'name': 'Czech Republic',
                                  'offsetgroup': 'Czech Republic',
                                  'orientation': 'v',
                                  'showlegend': True,
                                  'textposition': 'auto',
                                  'type': 'bar',
                                  'x': array(['František', 'Helena'], dtype=object),
                                  'xaxis': 'x',
                                  'y': array(['Wichterlová', 'Holý'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Country=Austria<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
 'legendgroup': 'Austria',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'Austria',
 'offsetgroup': 'Austria',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['Astrid'], dtype=object),
 'xaxis': 'x',
 'y': array(['Gruber'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Country=Belgium<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
 'legendgroup': 'Belgium',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'Belgium',
 'offsetgroup': 'Belgium',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['Daan'], dtype=object),
 'xaxis': 'x',
 'y': array(['Peeters'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Country=Denmark<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
 'legendgroup': 'Denmark',
 'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
 'name': 'Denmark',
 'offsetgroup': 'Denmark',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['Kara'], dtype=object),
 'xaxis': 'x',
 'y': array(['Nielsen'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Country=USA<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
 'legendgroup': 'USA',
 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
 'name': 'USA',
 'offsetgroup': 'USA',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['Frank', 'Jack', 'Michelle', 'Tim', 'Dan', 'Kathy', 'Heather', 'John',
             'Frank', 'Victor', 'Richard', 'Patrick', 'Julia'], dtype=object),
 'xaxis': 'x',
 'y': array(['Harris', 'Smith', 'Brooks', 'Goyer', 'Miller', 'Chase', 'Leacock',
             'Gordon', 'Ralston', 'Stevens', 'Cunningham', 'Gray', 'Barnett'],
            dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Country=Portugal<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
 'legendgroup': 'Portugal',
 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
 'name': 'Portugal',
 'offsetgroup': 'Portugal',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['João', 'Madalena'], dtype=object),
 'xaxis': 'x',
 'y': array(['Fernandes', 'Sampaio'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Country=France<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
 'legendgroup': 'France',
 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
 'name': 'France',
 'offsetgroup': 'France',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['Camille', 'Dominique', 'Marc', 'Wyatt', 'Isabelle'], dtype=object),
 'xaxis': 'x',
 'y': array(['Bernard', 'Lefebvre', 'Dubois', 'Girard', 'Mercier'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Country=Finland<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
 'legendgroup': 'Finland',
 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
 'name': 'Finland',
 'offsetgroup': 'Finland',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['Terhi'], dtype=object),
 'xaxis': 'x',
 'y': array(['Hämäläinen'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Country=Hungary<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
 'legendgroup': 'Hungary',
 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
 'name': 'Hungary',
 'offsetgroup': 'Hungary',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['Ladislav'], dtype=object),
 'xaxis': 'x',
 'y': array(['Kovács'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Country=Ireland<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
 'legendgroup': 'Ireland',
 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
 'name': 'Ireland',
 'offsetgroup': 'Ireland',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['Hugh'], dtype=object),
 'xaxis': 'x',
 'y': array(["O'Reilly"], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Country=Italy<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
 'legendgroup': 'Italy',
 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
 'name': 'Italy',
 'offsetgroup': 'Italy',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['Lucas'], dtype=object),
 'xaxis': 'x',
 'y': array(['Mancini'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Country=Netherlands<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
 'legendgroup': 'Netherlands',
 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
 'name': 'Netherlands',
 'offsetgroup': 'Netherlands',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
 'x': array(['Johannes'], dtype=object),
 'xaxis': 'x',
 'y': array(['Van der Berg'], dtype=object),
 'yaxis': 'y'},
{'alignmentgroup': 'True',
 'hovertemplate': 'Country=Poland<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
 'legendgroup': 'Poland',
 'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
 'name': 'Poland',
 'offsetgroup': 'Poland',
 'orientation': 'v',
 'showlegend': True,
 'textposition': 'auto',
 'type': 'bar',
                             'x': array(['Stanisław'], dtype=object),
                             'xaxis': 'x',
                             'y': array(['Wójcik'], dtype=object),
                             'yaxis': 'y'},
                            {'alignmentgroup': 'True',
                             'hovertemplate': 'Country=Spain<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
                             'legendgroup': 'Spain',
                             'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
                             'name': 'Spain',
                             'offsetgroup': 'Spain',
                             'orientation': 'v',
                             'showlegend': True,
                             'textposition': 'auto',
                             'type': 'bar',
                             'x': array(['Enrique'], dtype=object),
                             'xaxis': 'x',
                             'y': array(['Muñoz'], dtype=object),
                             'yaxis': 'y'},
                            {'alignmentgroup': 'True',
                             'hovertemplate': 'Country=Sweden<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
                             'legendgroup': 'Sweden',
                             'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
                             'name': 'Sweden',
                             'offsetgroup': 'Sweden',
                             'orientation': 'v',
                             'showlegend': True,
                             'textposition': 'auto',
                             'type': 'bar',
                             'x': array(['Joakim'], dtype=object),
                             'xaxis': 'x',
                             'y': array(['Johansson'], dtype=object),
                             'yaxis': 'y'},
                            {'alignmentgroup': 'True',
                             'hovertemplate': 'Country=United Kingdom<br>First Name=%{x}<br>Last Name=%{y}<extra></extra
              >',
                             'legendgroup': 'United Kingdom',
                             'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
                             'name': 'United Kingdom',
                             'offsetgroup': 'United Kingdom',
                             'orientation': 'v',
                             'showlegend': True,
                             'textposition': 'auto',
                             'type': 'bar',
                             'x': array(['Emma', 'Phil', 'Steve'], dtype=object),
                             'xaxis': 'x',
                             'y': array(['Jones', 'Hughes', 'Murray'], dtype=object),
                             'yaxis': 'y'},
                            {'alignmentgroup': 'True',
                             'hovertemplate': 'Country=Australia<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
                             'legendgroup': 'Australia',
                             'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                             'name': 'Australia',
                             'offsetgroup': 'Australia',
                             'orientation': 'v',
                             'showlegend': True,
                             'textposition': 'auto',
                             'type': 'bar',
                             'x': array(['Mark'], dtype=object),
                             'xaxis': 'x',
                             'y': array(['Taylor'], dtype=object),
                             'yaxis': 'y'},
                            {'alignmentgroup': 'True',
                             'hovertemplate': 'Country=Argentina<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
                             'legendgroup': 'Argentina',
                             'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
                             'name': 'Argentina',
                             'offsetgroup': 'Argentina',
                             'orientation': 'v',
                             'showlegend': True,
                             'textposition': 'auto',
                             'type': 'bar',
                             'x': array(['Diego'], dtype=object),
                             'xaxis': 'x',
                             'y': array(['Gutiérrez'], dtype=object),
                             'yaxis': 'y'},
                            {'alignmentgroup': 'True',
                             'hovertemplate': 'Country=Chile<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
                             'legendgroup': 'Chile',
                             'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
                             'name': 'Chile',
                             'offsetgroup': 'Chile',
                             'orientation': 'v',
                             'showlegend': True,
                             'textposition': 'auto',
                             'type': 'bar',
                             'x': array(['Luis'], dtype=object),
                             'xaxis': 'x',
                             'y': array(['Rojas'], dtype=object),
                             'yaxis': 'y'},
                            {'alignmentgroup': 'True',
                             'hovertemplate': 'Country=India<br>First Name=%{x}<br>Last Name=%{y}<extra></extra>',
                             'legendgroup': 'India',
                             'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
                             'name': 'India',
                             'offsetgroup': 'India',
                             'orientation': 'v',
                             'showlegend': True,
                             'textposition': 'auto',
                             'type': 'bar',
                             'x': array(['Manoj', 'Puja'], dtype=object),
                             'xaxis': 'x',
                             'y': array(['Pareek', 'Srivastava'], dtype=object),
                             'yaxis': 'y'}],
                   'layout': {'barmode': 'relative',
                              'legend': {'title': {'text': 'Country'}, 'tracegroupgap': 0},
                              'showlegend': False,
                              'template': '...',
                              'title': {'text': 'Customers Without Orders'},
                              'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'First Name'}},
                              'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Last Name'}}}
               }))
```

In [32]:

```
question = """
    There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums
    Can you find the top 10 most popular artists based on the number of tracks
"""

vn.ask(question=question)
```

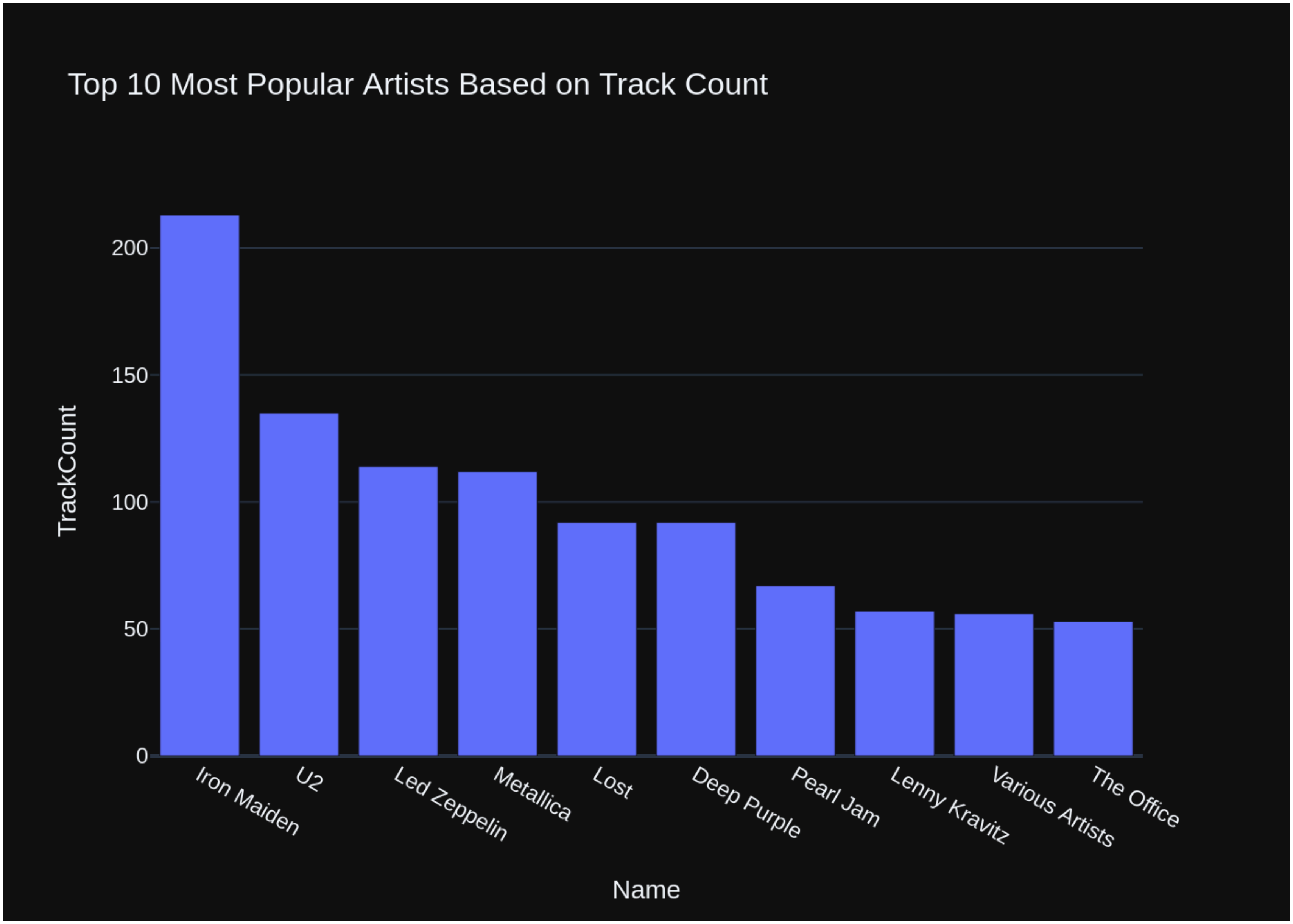
```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NU
LL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    By
tes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (Al
bumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (G
enreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media
_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r
\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    Arti
stId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "artists"\r\n(\r\n    ArtistId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\n
CREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "playlists"\r\n(\r\n    PlaylistId INTE
GER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "genres"\r\n(\r\n
GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "playlis
t_track"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_P
laylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Pl
aylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks"
(TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks"
(GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\n\n===Additional Context
\n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is s
ufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provide
d context is almost sufficient but requires knowledge of a specific string in a particular column, please g
enerate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a com
ment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be
generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered bef
ore, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': '  \n    Li
st all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT albums.T
itle, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId = artists.ArtistId'}, {'role': 'user', 'co
ntent': '  \n    List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'conten
t': 'SELECT g.Name, COUNT(t.TrackId) AS TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.GenreId\n
GROUP BY g.Name'}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive tracks (based on unit
price):\n'}, {'role': 'assistant', 'content': 'SELECT Name, UnitPrice\nFROM tracks\nORDER BY UnitPrice DESC
\nLIMIT 5'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'rol
e': 'assistant', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nO
RDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find all tracks with a name con
taining "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks\nWHERE Name L
IKE '%What%'"}, {'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 customers c\n
JOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': 'Can you
list all tables in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlit
```

```
e_master\nWHERE type = 'table'"}, {'role': 'user', 'content': '  \n    Get the total number of invoices for
each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.
InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.C
ustomerId, c.FirstName, c.LastName'}, {'role': 'user', 'content': '  \n    Get the average invoice total fo
r each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.FirstName, c.LastName, AVG
(i.Total) AS AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP B
Y c.CustomerId, c.FirstName, c.LastName'}, {'role': 'user', 'content': '  \n    List all invoices with a to
tal exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices\nWHERE Total > 10'}, {'rol
e': 'user', 'content': ' \n    There are 3 tables: artists, albums and tracks, where albums and artists are
linked by ArtistId, albums and tracks are linked by AlbumId,\n    Can you find the top 10 most popular arti
sts based on the number of tracks\n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREM
ENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT
NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n
Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\"
(AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres
\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES
\"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums
\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r
\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"artists\"\r\n(\r\n    ArtistId INTEGER PRIMAR
Y KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums
\" (ArtistId)\n\nCREATE INDEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"playlists\"\r\n(\r
\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE
\"genres\"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n
\nCREATE TABLE \"playlist_track\"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NUL
L,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) R
EFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (T
rackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDE
X IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_track\"
(TrackId)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guideli
nes \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations
for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific st
ring in a particular column, please generate an intermediate SQL query to find the distinct strings in that
column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficie
```

```
nt, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the quest
ion has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"ro
le": "user", "content": "  \n    List all albums and their corresponding artist names  \n"}, {"role": "assi
stant", "content": "SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId = artis
ts.ArtistId"}, {"role": "user", "content": "  \n    List all genres and the number of tracks in each genr
e:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.TrackId) AS TrackCount\nFROM genres g\nJOI
N tracks t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"role": "user", "content": "  \n    Find the top 5
most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT Name, UnitPrice
\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "what are the top 5 countrie
s that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS CustomerCount
\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "
\n    Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "conten
t": "SELECT * FROM tracks\nWHERE Name LIKE '%What%'"}, {"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 customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"},
{"role": "user", "content": "Can you list all tables in the SQLite database catalog?"}, {"role": "assistan
t", "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 c.Customer
Id, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.Cu
stomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName"}, {"role": "user", "content": "
\n    Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.Cust
omerId, c.FirstName, c.LastName, AVG(i.Total) AS AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON
c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName"}, {"role": "user", "content":
"  \n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM
invoices\nWHERE Total > 10"}, {"role": "user", "content": " \n    There are 3 tables: artists, albums and t
racks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n    Can y
ou find the top 10 most popular artists based on the number of tracks\n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:49:09.572113645Z', 'message': {'role': 'assistan
t', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artists a\nJOIN albums al ON a.ArtistId
= al.ArtistId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMIT 1
0'}, 'done_reason': 'stop', 'done': True, 'total_duration': 71658791615, 'load_duration': 1065349, 'prompt_
eval_count': 1554, 'prompt_eval_duration': 56055314000, 'eval_count': 71, 'eval_duration': 14926444000}
SELECT a.Name, COUNT(t.TrackId) AS TrackCount
FROM artists a
JOIN albums al ON a.ArtistId = al.ArtistId
JOIN tracks t ON al.AlbumId = t.AlbumId
GROUP BY a.Name
ORDER BY TrackCount DESC
LIMIT 10
SELECT a.Name, COUNT(t.TrackId) AS TrackCount
FROM artists a
```

```
JOIN albums al ON a.ArtistId = al.ArtistId
JOIN tracks t ON al.AlbumId = t.AlbumId
GROUP BY a.Name
ORDER BY TrackCount DESC
LIMIT 10
              Name  TrackCount
0      Iron Maiden         213
1               U2         135
2     Led Zeppelin         114
3        Metallica         112
4             Lost          92
5      Deep Purple          92
6        Pearl Jam          67
7    Lenny Kravitz          57
8  Various Artists          56
9       The Office          53
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query 
that answers the question the user asked: ' \n    There are 3 tables: artists, albums and tracks, where alb
ums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n    Can you find the top 
10 most popular artists based on the number of tracks\n'\n\nThe DataFrame was produced using this query: SE
LECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artists a\nJOIN albums al ON a.ArtistId = al.ArtistId\nJO
IN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMIT 10\n\nThe following
is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name          object
\nTrackCount     int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly co
de to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is
only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any ex
planations -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:49:27.844031184Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Name', y='TrackCount', title
='Top 10 Most Popular Artists Based on Track Count')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': T
rue, 'total_duration': 18243824446, 'load_duration': 695926, 'prompt_eval_count': 260, 'prompt_eval_duratio
n': 8971244000, 'eval_count': 49, 'eval_duration': 9142455000}
```


Top 10 Most Popular Artists Based on Track Count
TrackCount
0
50
100
150
200
Iron Maiden
U2
Led Zeppelin
Metallica
Lost
Deep Purple
Pearl Jam
Lenny Kravitz
Various Artists
The Office
Name

Out[32]:

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

In [33]:

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

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

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

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCRE
ATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName N
VARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address
NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    Post
alCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NUL
L,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n
\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER P
RIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NU
LL,\r\n    BillingAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(4
0),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)
NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION O
N UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE TABLE "e
mployees"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)
NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n
BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n
State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(2
4),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES "employee
s" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r
\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n
TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n
FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO
N,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI
ON\r\n)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE "playlist_track"\r\n(\r\n    PlaylistId I
NTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (Playli
stId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO AC
TION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO A
CTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE T
ABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  N
OT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database
invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a
valid SQL query without any explanations for the question. \n2. If the provided context is almost 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 th
e most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer
exactly as it was given before. \n'}, {'role': 'user', 'content': 'what are the top 5 countries that custom
ers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM custome
rs\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    List all
customers who have not placed any orders:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM customers\nW
```

```
HERE CustomerId'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'cont
ent': 'SELECT COUNT(*) FROM customers'}, {'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 customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'use
r', 'content': '  \n    Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'cont
ent': 'SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c
\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName'}, {'rol
e': 'user', 'content': "  \n    List all employees and their reporting manager's name (if any):\n"}, {'rol
e': 'assistant', 'content': "SELECT e.FirstName || ' ' || e.LastName AS EmployeeName, m.FirstName || ' ' ||
m.LastName AS ManagerName\nFROM employees e\nLEFT JOIN employees m ON e.ReportsTo = m.EmployeeId"}, {'rol
e': 'user', 'content': '  \n    Get the average invoice total for each customer:\n'}, {'role': 'assistant',
'content': 'SELECT c.CustomerId, c.FirstName, c.LastName, AVG(i.Total) AS AverageInvoiceTotal\nFROM custome
rs c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName'}, {'r
ole': 'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant',
'content': 'SELECT * FROM invoices\nWHERE Total > 10'}, {'role': 'user', 'content': '  \n    Find all invoi
ces since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceDate, SU
M(Total) AS TotalAmount\nFROM invoices\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role':
'user', 'content': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'assistant', 'conte
nt': "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {'role': 'user', 'content': '  \n     List al
l customers from Canada and their email addresses:\n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nC
REATE TABLE \"customers\"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstNa
me NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Addr
ess NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    
PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT
NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) 
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INT
EGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  
NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR
(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)
NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE TABLE
\"employees\"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(2
0)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGE
R,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(4
0),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NV
ARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES
\"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice_i
tems\"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT
NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER
NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON
UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE \"playlist_track\"\r\n
(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack
PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON \"employee
s\" (ReportsTo)\n\nCREATE TABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) R
EFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Co
ntext \n\nIn the chinook database invoice means order\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": "what ar
e the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT
(*) AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "u
ser", "content": "  \n    List all customers who have not placed any orders:\n"}, {"role": "assistant", "co
ntent": "SELECT * FROM customers\nWHERE CustomerId"}, {"role": "user", "content": "How many customers are t
here"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "
\n    Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Countr
y, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nG
ROUP BY c.Country"}, {"role": "user", "content": "  \n    Get the total number of invoices for each custome
r\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId) A
S TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId,
c.FirstName, c.LastName"}, {"role": "user", "content": "  \n    List all employees and their reporting mana
ger's name (if any):\n"}, {"role": "assistant", "content": "SELECT e.FirstName || ' ' || e.LastName AS Empl
oyeeName, m.FirstName || ' ' || m.LastName AS ManagerName\nFROM employees e\nLEFT JOIN employees m ON e.Rep
ortsTo = m.EmployeeId"}, {"role": "user", "content": "  \n    Get the average invoice total for each custom
er:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, c.FirstName, c.LastName, AVG(i.Total) AS Av
erageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId,
c.FirstName, c.LastName"}, {"role": "user", "content": "  \n    List all invoices with a total exceeding $1
0:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices\nWHERE Total > 10"}, {"role": "user", "con
tent": "  \n    Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "con
tent": "SELECT InvoiceDate, SUM(Total) AS TotalAmount\nFROM invoices\nWHERE InvoiceDate >= '2010-01-01'\nGR
```

```
OUP BY InvoiceDate"}, {"role": "user", "content": "Can you list all tables in the SQLite database catalo
g?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {"role": "u
ser", "content": "  \n     List all customers from Canada and their email addresses:\n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:50:38.999277776Z', 'message': {'role': 'assistan
t', 'content': "SELECT FirstName, LastName, Email\nFROM customers\nWHERE Country = 'Canada"}, 'done_reaso
n': 'stop', 'done': True, 'total_duration': 71063207289, 'load_duration': 718147, 'prompt_eval_count': 181
6, 'prompt_eval_duration': 65863815000, 'eval_count': 22, 'eval_duration': 4524976000}
SELECT FirstName, LastName, Email
FROM customers
WHERE Country = 'Canada
SELECT FirstName, LastName, Email
FROM customers
WHERE Country = 'Canada
Couldn't run sql:  Execution failed on sql 'SELECT FirstName, LastName, Email
FROM customers
WHERE Country = 'Canada': unrecognized token: "'Canada"
```

In [34]:

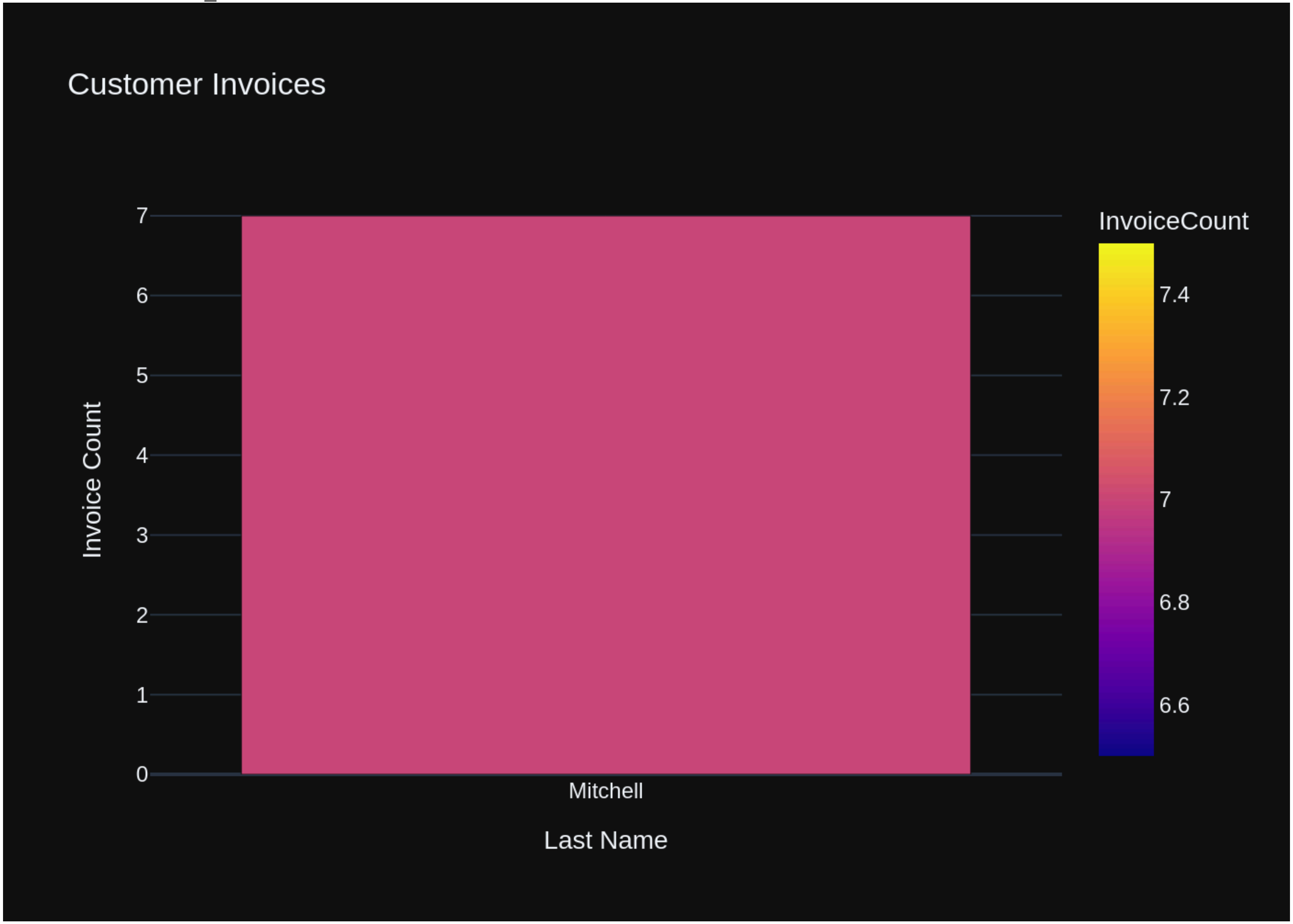
```python
question = """
     Find the customer with the most invoices
"""

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 fo
rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingA
ddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCount
ry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIG
N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId
ON "invoice_items" (InvoiceId)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KE
Y AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    Unit
Price NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES
"invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERE
NCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLin
eTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KE
Y AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NUL
L,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHA
R(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax N
VARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportR
epId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IN
DEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "employees"\r\n(\r\n    Employee
Id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVA
RCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n
HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n
Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r
\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DEL
ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\n
CREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR
(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n
Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMER
IC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DEL
ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means
order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL que
ry without any explanations for the question. \n2. If the provided context is almost sufficient but require
s knowledge of a specific string in a particular column, please generate an intermediate SQL query to find
the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the p
rovided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most 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    Get the total number of invoices for each cu
stomer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.Invoice
Id) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Customer
```

```
Id, c.FirstName, c.LastName'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per c
ountry:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM
customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'conten
t': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT *
FROM invoices\nWHERE Total > 10'}, {'role': 'user', 'content': '  \n    Get the average invoice total for e
ach customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.FirstName, c.LastName, AVG(i.To
tal) AS AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.C
ustomerId, c.FirstName, c.LastName'}, {'role': 'user', 'content': '  \n    Find all invoices since 2010 and
the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceDate, SUM(Total) AS TotalAm
ount\nFROM invoices\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content':
'  \n    List all customers who have not placed any orders:\n'}, {'role': 'assistant', 'content': 'SELECT *
FROM customers\nWHERE CustomerId'}, {'role': 'user', 'content': 'what are the top 5 countries that customer
s come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers
\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find the t
op 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT Name, UnitP
rice\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are
there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': "
\n    List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content':
"SELECT e.FirstName || ' ' || e.LastName AS EmployeeName, m.FirstName || ' ' || m.LastName AS ManagerName\n
FROM employees e\nLEFT JOIN employees m ON e.ReportsTo = m.EmployeeId"}, {'role': 'user', 'content': '  \n
Find the customer with the most invoices \n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    Billin
gAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCou
ntry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FORE
IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineInv
oiceId ON \"invoice_items\" (InvoiceId)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n    InvoiceLineId INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r
\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId)
REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Tr
ackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX
IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE TABLE \"customers\"\r\n(\r\n    CustomerId
INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCH
AR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n
State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(2
4),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREI
GN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"employees
\"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NUL
L,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    Bir
thDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    St
ate NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r
\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES \"employees\"
(EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON
\"employees\" (ReportsTo)\n\nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT
NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NUL
L,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Byt
es INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\" (A
lbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres\"
(GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES \"me
dia_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context
\n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is s
ufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provide
d context is almost sufficient but requires knowledge of a specific string in a particular column, please g
enerate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a com
ment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be g
enerated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered befo
re, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "  \n    Get
the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId,
c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.Custom
erId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName"}, {"role": "user", "content": "  \n
Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COU
NT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP B
Y c.Country"}, {"role": "user", "content": "  \n    List all invoices with a total exceeding $10:\n"}, {"ro
le": "assistant", "content": "SELECT * FROM invoices\nWHERE Total > 10"}, {"role": "user", "content": "  \n
Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerI
d, c.FirstName, c.LastName, AVG(i.Total) AS AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.Cus
tomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName"}, {"role": "user", "content": "  \n
Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT I
nvoiceDate, SUM(Total) AS TotalAmount\nFROM invoices\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDa
te"}, {"role": "user", "content": "  \n    List all customers who have not placed any orders:\n"}, {"role":
"assistant", "content": "SELECT * FROM customers\nWHERE CustomerId"}, {"role": "user", "content": "what are
the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*)
AS CustomerCount\nFROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "use
r", "content": "  \n    Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistan
t", "content": "SELECT Name, UnitPrice\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user",
"content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customer
s"}, {"role": "user", "content": "  \n    List all employees and their reporting manager's name (if an
y):\n"}, {"role": "assistant", "content": "SELECT e.FirstName || ' ' || e.LastName AS EmployeeName, m.First
Name || ' ' || m.LastName AS ManagerName\nFROM employees e\nLEFT JOIN employees m ON e.ReportsTo = m.Employ
eeId"}, {"role": "user", "content": "  \n     Find the customer with the most invoices \n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:51:57.360553039Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.FirstName, c.LastName, COUNT(*) AS InvoiceCount\nFROM customers c\nJOIN invoices i
ON c.CustomerId = i.CustomerId\nGROUP BY c.FirstName, c.LastName\nORDER BY InvoiceCount DESC\nLIMIT 1'}, 'd
one_reason': 'stop', 'done': True, 'total_duration': 78320902892, 'load_duration': 740082, 'prompt_eval_cou
nt': 1817, 'prompt_eval_duration': 65569402000, 'eval_count': 57, 'eval_duration': 12080663000}
SELECT c.FirstName, c.LastName, COUNT(*) AS InvoiceCount
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName
ORDER BY InvoiceCount DESC
LIMIT 1
SELECT c.FirstName, c.LastName, COUNT(*) AS InvoiceCount
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName
ORDER BY InvoiceCount DESC
LIMIT 1
  FirstName  LastName  InvoiceCount
0     Aaron  Mitchell             7
Ollama parameters:
model=codegemma: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.FirstName, c.LastName, COUNT(*) AS InvoiceCount\nFROM custom
ers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.FirstName, c.LastName\nORDER BY InvoiceCo
unt DESC\nLIMIT 1\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.d
types gives:\n FirstName       object\nLastName        object\nInvoiceCount     int64\ndtype: object"}, {"r
ole": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? As
sume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Ind
icator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:52:18.515999509Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='LastName', y='InvoiceCount',
```

```
color='InvoiceCount', title='Customer Invoices')\n\nfig.update_layout(\n    xaxis_title='Last Name',\n    y
axis_title='Invoice Count'\n)\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration':
21128426932, 'load_duration': 665607, 'prompt_eval_count': 208, 'prompt_eval_duration': 7212081000, 'eval_c
ount': 73, 'eval_duration': 13785719000}
```



Out[34]:

```
('SELECT c.FirstName, c.LastName, COUNT(*) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.Custome
rId = i.CustomerId\nGROUP BY c.FirstName, c.LastName\nORDER BY InvoiceCount DESC\nLIMIT 1',
   FirstName  LastName  InvoiceCount
 0     Aaron  Mitchell             7,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'LastName=%{x}<br>InvoiceCount=%{marker.color}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': array([7]), 'coloraxis': 'coloraxis', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Mitchell'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'coloraxis': {'colorbar': {'title': {'text': 'InvoiceCount'}},
                              'colorscale': [[0.0, '#0d0887'], [0.1111111111111111,
                                             '#46039f'], [0.2222222222222222,
                                             '#7201a8'], [0.3333333333333333,
                                             '#9c179e'], [0.4444444444444444,
                                             '#bd3786'], [0.5555555555555556,
                                             '#d8576b'], [0.6666666666666666,
                                             '#ed7953'], [0.7777777777777778,
                                             '#fb9f3a'], [0.8888888888888888,
                                             '#fdca26'], [1.0, '#f0f921']]},
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Customer Invoices'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Last Name'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Invoice Count'}}}
 }))
```

In [ ]:

Advanced SQL questions

In [35]:

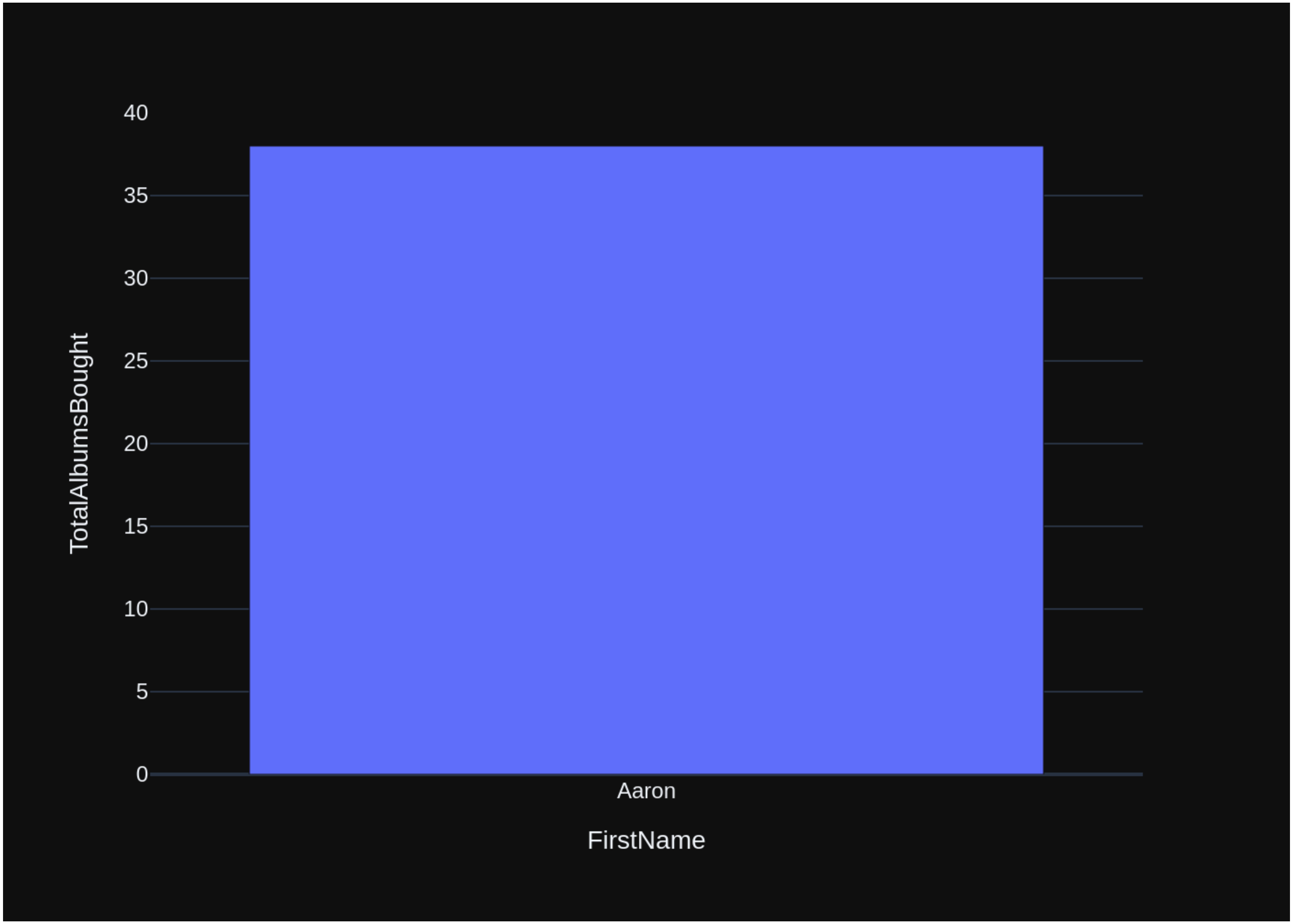
```
question = """
     Find the customer who bought the most albums in total quantity (across all invoices):
"""

vn.ask(question=question)
```

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

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NU
LL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    By
tes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (Al
bumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (G
enreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media
_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_item
s"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NUL
L,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT
NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDA
TE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) R
EFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_Alb
umArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTO
INCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    Bill
ingAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingC
ountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FO
REIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE INDEX IFK_InvoiceLineIn
voiceId ON "invoice_items" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\n
CREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n    ArtistId INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the ch
inook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, pl
ease generate a valid SQL query without any explanations for the question. \n2. If the provided context is
almost sufficient but requires knowledge of a specific string in a particular column, please generate an in
termediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying i
ntermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n
4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please r
epeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': '  \n     Find the custo
mer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, COUNT(*)
AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.FirstName, c.
LastName\nORDER BY InvoiceCount DESC\nLIMIT 1'}, {'role': 'user', 'content': '  \n    Get the total number
of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.FirstName, c.La
stName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.Customer
Id\nGROUP BY c.CustomerId, c.FirstName, c.LastName'}, {'role': 'user', 'content': ' \n    There are 3 table
s: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linke
d by AlbumId,\n    Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role':
'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artists a\nJOIN albums al ON
a.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DES
```

```
C\nLIMIT 10'}, {'role': 'user', 'content': '  \n    Find all invoices since 2010 and the total amount invoi
ced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceDate, SUM(Total) AS TotalAmount\nFROM invoices\nW
HERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': '  \n    Find the tot
al number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.Invoice
Id) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Countr
y'}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive tracks (based on unit price):\n'},
{'role': 'assistant', 'content': 'SELECT Name, UnitPrice\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'},
{'role': 'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistan
t', 'content': 'SELECT * FROM invoices\nWHERE Total > 10'}, {'role': 'user', 'content': '  \n    Get the av
erage invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, c.FirstN
ame, c.LastName, AVG(i.Total) AS AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId =
i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName'}, {'role': 'user', 'content': '  \n    List a
ll albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT albums.Titl
e, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId = artists.ArtistId'}, {'role': 'user', 'conte
nt': '  \n    List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content':
'SELECT g.Name, COUNT(t.TrackId) AS TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.GenreId\nGROU
P BY g.Name'}, {'role': 'user', 'content': '  \n     Find the customer who bought the most albums in total
quantity (across all invoices): \n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREM
ENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT
NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n
Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\"
(AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres
\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES
\"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoic
e_items\"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  N
OT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGE
R  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTI
ON ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (Ar
tistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IN
DEX IFK_AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGER P
RIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NU
LL,\r\n    BillingAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(4
0),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)
NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE IN
DEX IFK_InvoiceLineInvoiceId ON \"invoice_items\" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"in
voices\" (CustomerId)\n\nCREATE INDEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"artists
\"\r\n(\r\n    ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n\n===
Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the pr
ovided context is sufficient, please generate a valid SQL query without any explanations for the question.
\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particul
ar column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend t
he query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explai
n why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been aske
d and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "con
tent": "  \n     Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.
FirstName, c.LastName, COUNT(*) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.Cust
omerId\nGROUP BY c.FirstName, c.LastName\nORDER BY InvoiceCount DESC\nLIMIT 1"}, {"role": "user", "conten
t": "  \n    Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELE
CT c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoi
ces i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName"}, {"role": "user", "c
ontent": " \n    There are 3 tables: artists, albums and tracks, where albums and artists are linked by Art
istId, albums and tracks are linked by AlbumId,\n    Can you find the top 10 most popular artists based on
the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TrackCount\n
FROM artists a\nJOIN albums al ON a.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP
BY a.Name\nORDER BY TrackCount DESC\nLIMIT 10"}, {"role": "user", "content": "  \n    Find all invoices sin
ce 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceDate, SUM(Tota
l) AS TotalAmount\nFROM invoices\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "use
r", "content": "  \n    Find the total number of invoices per country:\n"}, {"role": "assistant", "conten
t": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.Customer
Id = i.CustomerId\nGROUP BY c.Country"}, {"role": "user", "content": "  \n    Find the top 5 most expensive
tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT Name, UnitPrice\nFROM tracks\nO
RDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    List all invoices with a total exce
eding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices\nWHERE Total > 10"}, {"role": "use
r", "content": "  \n    Get the average invoice total for each customer:\n"}, {"role": "assistant", "conten
t": "SELECT c.CustomerId, c.FirstName, c.LastName, AVG(i.Total) AS AverageInvoiceTotal\nFROM customers c\nJ
OIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName"}, {"role":
"user", "content": "  \n    List all albums and their corresponding artist names  \n"}, {"role": "assistan
t", "content": "SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId = artists.A
rtistId"}, {"role": "user", "content": "  \n    List all genres and the number of tracks in each genr
e:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.TrackId) AS TrackCount\nFROM genres g\nJOI
N tracks t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"role": "user", "content": "  \n     Find the cust
omer who bought the most albums in total quantity (across all invoices): \n"}]
Ollama Response:
```

```
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:53:46.151623376Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.FirstName, c.LastName, SUM(ii.Quantity) AS TotalAlbumsBought\nFROM customers c\nJO
IN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tra
cks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.FirstName, c.LastName
\nORDER BY TotalAlbumsBought DESC\nLIMIT 1'}, 'done_reason': 'stop', 'done': True, 'total_duration': 875164
05600, 'load_duration': 738231, 'prompt_eval_count': 1772, 'prompt_eval_duration': 64122200000, 'eval_coun
t': 107, 'eval_duration': 22735118000}
SELECT c.FirstName, c.LastName, SUM(ii.Quantity) AS TotalAlbumsBought
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId
JOIN tracks t ON ii.TrackId = t.TrackId
JOIN albums al ON t.AlbumId = al.AlbumId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalAlbumsBought DESC
LIMIT 1
SELECT c.FirstName, c.LastName, SUM(ii.Quantity) AS TotalAlbumsBought
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId
JOIN tracks t ON ii.TrackId = t.TrackId
JOIN albums al ON t.AlbumId = al.AlbumId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalAlbumsBought DESC
LIMIT 1
  FirstName  LastName  TotalAlbumsBought
0     Aaron  Mitchell                 38
Ollama parameters:
model=codegemma: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.FirstName, c.La
stName, SUM(ii.Quantity) AS TotalAlbumsBought\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.Custom
erId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN al
bums al ON t.AlbumId = al.AlbumId\nGROUP BY c.FirstName, c.LastName\nORDER BY TotalAlbumsBought DESC\nLIMIT
1\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n F
irstName            object\nLastName             object\nTotalAlbumsBought     int64\ndtype: object"}, {"ro
le": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Ass
ume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indi
cator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:54:23.848097737Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='FirstName', y='TotalAlbumsBou
ght', title='Customers with the Most Albums Bought')\n\nfig.update_layout(\n    xaxis_title='Customer Nam
e',\n    yaxis_title='Total Albums Bought',\n    plot_bgcolor='rgba(0, 0, 0, 0)',\n    paper_bgcolor='rgba
(0, 0, 0, 0)',\n)\n\nif len(df) == 1:\n    fig.add_trace(px.indicator(df, value='TotalAlbumsBought', title
='Total Albums Bought'))\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 37669
810576, 'load_duration': 635850, 'prompt_eval_count': 268, 'prompt_eval_duration': 9394658000, 'eval_coun
t': 147, 'eval_duration': 28143190000}
```


40
35
30
25
20
15
10
5
0
TotalAlbumsBought
Aaron
FirstName

Out[35]:

```
('SELECT c.FirstName, c.LastName, SUM(ii.Quantity) AS TotalAlbumsBought\nFROM customers c\nJOIN invoices i
ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.T
rackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.FirstName, c.LastName\nORDER BY T
otalAlbumsBought DESC\nLIMIT 1',
   FirstName  LastName  TotalAlbumsBought
 0     Aaron  Mitchell                 38,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'FirstName=%{x}<br>TotalAlbumsBought=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Aaron'], dtype=object),
               'xaxis': 'x',
               'y': array([38]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'FirstName'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalAlbumsBought'}}}
 }))
```

In [36]:

```
question = """
    Hint: album quantity is found in invoice_items,

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

vn.ask(question=question)
```

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

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY
AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPr
ice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES
"invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERE
NCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n
(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    Al
bumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(22
0),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NUL
L,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI
ON,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON
UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NU
LL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId)
REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_Al
bumArtistId ON "albums" (ArtistId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)
\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE "invoices"\r\n(\r\n
InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDa
te DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingS
tate NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total
NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DEL
ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\n
CREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n    ArtistId INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the ch
inook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, pl
ease generate a valid SQL query without any explanations for the question. \n2. If the provided context is
almost sufficient but requires knowledge of a specific string in a particular column, please generate an in
termediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying i
ntermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n
4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please r
epeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': '  \n     Find the custo
mer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'conten
t': 'SELECT c.FirstName, c.LastName, SUM(ii.Quantity) AS TotalAlbumsBought\nFROM customers c\nJOIN invoices
i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.
TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.FirstName, c.LastName\nORDER BY T
otalAlbumsBought DESC\nLIMIT 1'}, {'role': 'user', 'content': '  \n     Find the customer with the most inv
oices \n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, COUNT(*) AS InvoiceCount\nFRO
M customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.FirstName, c.LastName\nORDER BY I
nvoiceCount DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n    There are 3 tables: artists, albums and tr
acks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n    Can yo
u find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'content':
```

```
'SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artists a\nJOIN albums al ON a.ArtistId = al.ArtistId
\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMIT 10'}, {'role':
'user', 'content': '  \n    Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assi
stant', 'content': 'SELECT Name, UnitPrice\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'use
r', 'content': '  \n    Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'cont
ent': 'SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c
\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName'}, {'rol
e': 'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'c
ontent': 'SELECT * FROM invoices\nWHERE Total > 10'}, {'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 customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'},
{'role': 'user', 'content': '  \n    Get the average invoice total for each customer:\n'}, {'role': 'assist
ant', 'content': 'SELECT c.CustomerId, c.FirstName, c.LastName, AVG(i.Total) AS AverageInvoiceTotal\nFROM c
ustomers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastNam
e'}, {'role': 'user', 'content': '  \n    List all albums and their corresponding artist names  \n'}, {'rol
e': 'assistant', 'content': 'SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistI
d = artists.ArtistId'}, {'role': 'user', 'content': '  \n    Find all invoices since 2010 and the total amo
unt invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceDate, SUM(Total) AS TotalAmount\nFROM in
voices\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': '  \n    Hin
t: album quantity is found in invoice_items, \n    \n    Find the top 5 customers who bought the most album
s in total quantity (across all invoices):\n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoice_items\"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY K
EY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    Uni
tPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCE
S \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) RE
FERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks
\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n
AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(22
0),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NUL
L,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO AC
TION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTI
ON ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (Ar
tistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IN
```

```
DEX IFK_AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON \"invoice_items
\" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE TABLE \"invo
ices\"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NUL
L,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(4
0),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR
(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (Custo
merId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invo
ices\" (CustomerId)\n\nCREATE INDEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"artists\"\r
\n(\r\n    ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n\n===Addi
tional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provid
ed context is sufficient, please generate a valid SQL query without any explanations for the question. \n2.
If the provided context is almost sufficient but requires knowledge of a specific string in a particular co
lumn, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the qu
ery with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why
it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and
answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content":
"  \n     Find the customer who bought the most albums in total quantity (across all invoices): \n"}, {"rol
e": "assistant", "content": "SELECT c.FirstName, c.LastName, SUM(ii.Quantity) AS TotalAlbumsBought\nFROM cu
stomers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.Invoic
eId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.FirstNam
e, c.LastName\nORDER BY TotalAlbumsBought DESC\nLIMIT 1"}, {"role": "user", "content": "  \n     Find the c
ustomer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastName, COUN
T(*) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.FirstNam
e, c.LastName\nORDER BY InvoiceCount DESC\nLIMIT 1"}, {"role": "user", "content": " \n    There are 3 table
s: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linke
d by AlbumId,\n    Can you find the top 10 most popular artists based on the number of tracks\n"}, {"role":
"assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artists a\nJOIN albums al ON
a.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DES
C\nLIMIT 10"}, {"role": "user", "content": "  \n    Find the top 5 most expensive tracks (based on unit pri
ce):\n"}, {"role": "assistant", "content": "SELECT Name, UnitPrice\nFROM tracks\nORDER BY UnitPrice DESC\nL
IMIT 5"}, {"role": "user", "content": "  \n    Get the total number of invoices for each customer\n"}, {"ro
le": "assistant", "content": "SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvo
ices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName,
c.LastName"}, {"role": "user", "content": "  \n    List all invoices with a total exceeding $10:\n"}, {"rol
e": "assistant", "content": "SELECT * FROM invoices\nWHERE Total > 10"}, {"role": "user", "content": "  \n
Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COU
NT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP B
Y c.Country"}, {"role": "user", "content": "  \n    Get the average invoice total for each customer:\n"},
{"role": "assistant", "content": "SELECT c.CustomerId, c.FirstName, c.LastName, AVG(i.Total) AS AverageInvo
iceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstN
ame, c.LastName"}, {"role": "user", "content": "  \n    List all albums and their corresponding artist name
s  \n"}, {"role": "assistant", "content": "SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists ON
```

```
albums.ArtistId = artists.ArtistId"}, {"role": "user", "content": "  \n    Find all invoices since 2010 and
the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceDate, SUM(Total) AS TotalAm
ount\nFROM invoices\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "content":
"  \n    Hint: album quantity is found in invoice_items, \n    \n    Find the top 5 customers who bought th
e most albums in total quantity (across all invoices):\n"}]
```

```
Ollama Response:
```

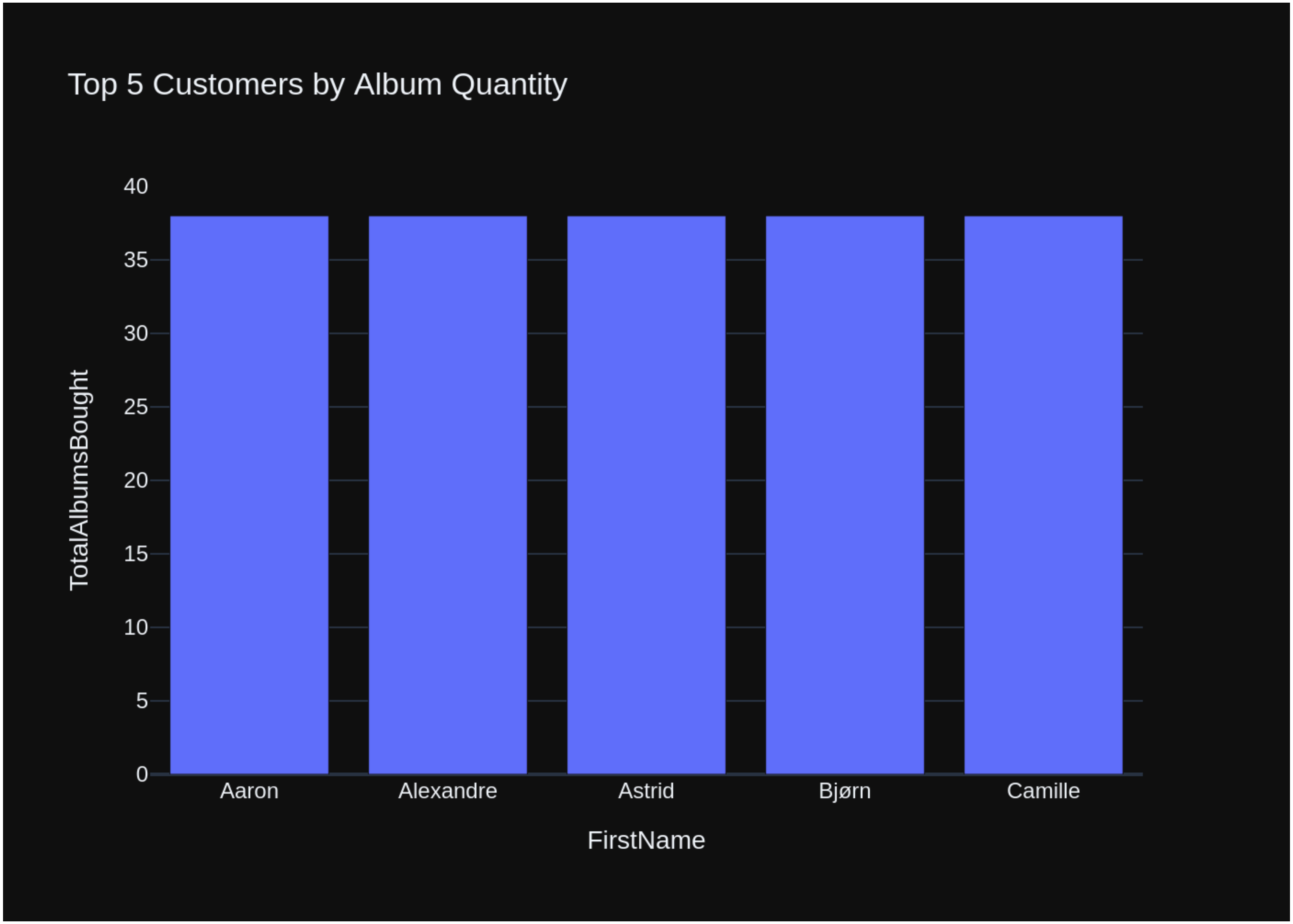
```
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:55:54.915516962Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.FirstName, c.LastName, SUM(ii.Quantity) AS TotalAlbumsBought\nFROM customers c\nJO
IN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tra
cks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.FirstName, c.LastName
\nORDER BY TotalAlbumsBought DESC\nLIMIT 5'}, 'done_reason': 'stop', 'done': True, 'total_duration': 909209
73330, 'load_duration': 809858, 'prompt_eval_count': 1856, 'prompt_eval_duration': 67391082000, 'eval_coun
t': 107, 'eval_duration': 22861935000}
```

```
SELECT c.FirstName, c.LastName, SUM(ii.Quantity) AS TotalAlbumsBought
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId
JOIN tracks t ON ii.TrackId = t.TrackId
JOIN albums al ON t.AlbumId = al.AlbumId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalAlbumsBought DESC
LIMIT 5
```

```
SELECT c.FirstName, c.LastName, SUM(ii.Quantity) AS TotalAlbumsBought
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId
JOIN tracks t ON ii.TrackId = t.TrackId
JOIN albums al ON t.AlbumId = al.AlbumId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalAlbumsBought DESC
LIMIT 5
```

```
   FirstName  LastName  TotalAlbumsBought
0      Aaron  Mitchell                 38
1  Alexandre     Rocha                 38
2     Astrid    Gruber                 38
3      Bjørn    Hansen                 38
4    Camille   Bernard                 38
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
```

```
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n    Hint: album quantity is found in invoice_items, \n    \n
Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'\n\nThe Data
Frame was produced using this query: SELECT c.FirstName, c.LastName, SUM(ii.Quantity) AS TotalAlbumsBought
\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId =
ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY
c.FirstName, c.LastName\nORDER BY TotalAlbumsBought DESC\nLIMIT 5\n\nThe following is information about the
resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n FirstName            object\nLastName
object\nTotalAlbumsBought     int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Pyt
hon plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'd
f'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not an
swer with any explanations -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:56:13.463922315Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='FirstName', y='TotalAlbumsBou
ght', title='Top 5 Customers by Album Quantity')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True,
'total_duration': 18522136478, 'load_duration': 686087, 'prompt_eval_count': 286, 'prompt_eval_duration': 9
740910000, 'eval_count': 46, 'eval_duration': 8648787000}
```


Top 5 Customers by Album Quantity
40
35
30
25
20
15
10
5
0
TotalAlbumsBought
Aaron
Alexandre
Astrid
Bjørn
Camille
FirstName

```
Out[36]: ('SELECT c.FirstName, c.LastName, SUM(ii.Quantity) AS TotalAlbumsBought\nFROM customers c\nJOIN invoices i
         ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.T
         rackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.FirstName, c.LastName\nORDER BY T
         otalAlbumsBought DESC\nLIMIT 5',
             FirstName  LastName  TotalAlbumsBought
          0      Aaron  Mitchell                 38
          1  Alexandre     Rocha                 38
          2     Astrid    Gruber                 38
          3      Bjørn    Hansen                 38
          4   Camille   Bernard                 38,
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'FirstName=%{x}<br>TotalAlbumsBought=%{y}<extra></extra>',
                        'legendgroup': '',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array(['Aaron', 'Alexandre', 'Astrid', 'Bjørn', 'Camille'], dtype=object),
                        'xaxis': 'x',
                        'y': array([38, 38, 38, 38, 38]),
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                         'legend': {'tracegroupgap': 0},
                         'template': '...',
                         'title': {'text': 'Top 5 Customers by Album Quantity'},
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'FirstName'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalAlbumsBought'}}}
          }))
```

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

In [37]:
```
question = """
     Find the top 5 customers who spent the most money overall,

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

vn.ask(question=question)
```

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

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingA
ddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCount
ry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIG
N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r
\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\t
ON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t
\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items"
(InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLin
eTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KE
Y AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NUL
L,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHA
R(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax N
VARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportR
epId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TA
BLE "employees"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR
(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGE
R,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(4
0),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NV
ARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES
"employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n
(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    Al
bumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(22
0),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NUL
L,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI
ON,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON
UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    Tra
ckId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREI
GN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r
\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r
\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\n\n===Additional Context \n\nIn the
chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient,
please generate a valid SQL query without any explanations for the question. \n2. If the provided context 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.
\n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please
```

```
repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': '  \n    Hint: album qu
antity is found in invoice_items, \n    \n    Find the top 5 customers who bought the most albums in total
quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, SUM
(ii.Quantity) AS TotalAlbumsBought\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN
invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON
t.AlbumId = al.AlbumId\nGROUP BY c.FirstName, c.LastName\nORDER BY TotalAlbumsBought DESC\nLIMIT 5'}, {'rol
e': 'user', 'content': '  \n     Find the customer who bought the most albums in total quantity (across all
invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, SUM(ii.Quantity) AS Tota
lAlbumsBought\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON
i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.Album
Id\nGROUP BY c.FirstName, c.LastName\nORDER BY TotalAlbumsBought DESC\nLIMIT 1'}, {'role': 'user', 'conten
t': '  \n     Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.Fir
stName, c.LastName, COUNT(*) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.Custome
rId\nGROUP BY c.FirstName, c.LastName\nORDER BY InvoiceCount DESC\nLIMIT 1'}, {'role': 'user', 'content': '
\n    Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.Cust
omerId, c.FirstName, c.LastName, AVG(i.Total) AS AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON
c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName'}, {'role': 'user', 'content':
'  \n    Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT
c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices
i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName'}, {'role': 'user', 'conte
nt': '  \n    Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT
c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.Custo
merId\nGROUP BY c.Country'}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive tracks (bas
ed on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT Name, UnitPrice\nFROM tracks\nORDER BY Uni
tPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    List all invoices with a total exceeding $1
0:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices\nWHERE Total > 10'}, {'role': 'user', 'con
tent': '  \n    Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'con
tent': "SELECT InvoiceDate, SUM(Total) AS TotalAmount\nFROM invoices\nWHERE InvoiceDate >= '2010-01-01'\nGR
OUP BY InvoiceDate"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come fro
m?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) AS CustomerCount\nFROM customers\nGROUP BY
Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n     Find the top 5 custo
mers who spent the most money overall, \n     \n     Hint: order total can be found on invoices table, calc
ulation using invoice_items detail table is unnecessary \n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    Billin
gAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCou
ntry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FORE
IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NO
T NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (Tra
ckId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON \"in
voice_items\" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDE
X IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE TABLE \"customers\"\r\n(\r\n    CustomerI
d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVAR
CHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n
State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(2
4),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREI
GN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE TABLE \"employees\"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n
ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n
City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(1
0),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (Repo
rtsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE
TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)
NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Comp
oser NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(1
0,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON
UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON D
ELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n    PlaylistId INTEGER
NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, Tr
ackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION O
N UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTIO
N ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON \"employees\" (ReportsTo)\n\n\n===Addit
ional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provide
d context is sufficient, please generate a valid SQL query without any explanations for the question. \n2.
If the provided context is almost sufficient but requires knowledge of a specific string in a particular co
lumn, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the qu
ery with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why
it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and
answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content":
"  \n    Hint: album quantity is found in invoice_items, \n    \n    Find the top 5 customers who bought th
e most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.Firs
tName, c.LastName, SUM(ii.Quantity) AS TotalAlbumsBought\nFROM customers c\nJOIN invoices i ON c.CustomerId
```

```
= i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackI
d\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.FirstName, c.LastName\nORDER BY TotalAlbumsBought D
ESC\nLIMIT 5"}, {"role": "user", "content": "  \n     Find the customer who bought the most albums in total
quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastName, SUM
(ii.Quantity) AS TotalAlbumsBought\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN
invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON
t.AlbumId = al.AlbumId\nGROUP BY c.FirstName, c.LastName\nORDER BY TotalAlbumsBought DESC\nLIMIT 1"}, {"rol
e": "user", "content": "  \n     Find the customer with the most invoices \n"}, {"role": "assistant", "cont
ent": "SELECT c.FirstName, c.LastName, COUNT(*) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.Cus
tomerId = i.CustomerId\nGROUP BY c.FirstName, c.LastName\nORDER BY InvoiceCount DESC\nLIMIT 1"}, {"role":
"user", "content": "  \n    Get the average invoice total for each customer:\n"}, {"role": "assistant", "co
ntent": "SELECT c.CustomerId, c.FirstName, c.LastName, AVG(i.Total) AS AverageInvoiceTotal\nFROM customers
c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName"}, {"rol
e": "user", "content": "  \n    Get the total number of invoices for each customer\n"}, {"role": "assistan
t", "content": "SELECT c.CustomerId, c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\nFROM cus
tomers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId, c.FirstName, c.LastName"},
{"role": "user", "content": "  \n    Find the total number of invoices per country:\n"}, {"role": "assistan
t", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices\nFROM customers c\nJOIN invoices i ON
c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"role": "user", "content": "  \n    Find the top 5 most
expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT Name, UnitPrice\nFROM
tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    List all invoices with a t
otal exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices\nWHERE Total > 10"}, {"ro
le": "user", "content": "  \n    Find all invoices since 2010 and the total amount invoiced:\n"}, {"role":
"assistant", "content": "SELECT InvoiceDate, SUM(Total) AS TotalAmount\nFROM invoices\nWHERE InvoiceDate >=
'2010-01-01'\nGROUP BY InvoiceDate"}, {"role": "user", "content": "what are the top 5 countries that custom
ers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) AS CustomerCount\nFROM custome
rs\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "  \n     Find th
e top 5 customers who spent the most money overall, \n     \n     Hint: order total can be found on invoice
s table, calculation using invoice_items detail table is unnecessary \n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:57:31.787339888Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.FirstName, c.LastName\n<start_of_invoice>'}, 'done_reason': 'stop', 'done': True,
'total_duration': 78175565836, 'load_duration': 1194012, 'prompt_eval_count': 2030, 'prompt_eval_duration':
73926174000, 'eval_count': 17, 'eval_duration': 3552234000}
SELECT c.FirstName, c.LastName
<start_of_invoice>
SELECT c.FirstName, c.LastName
<start_of_invoice>
Couldn't run sql:  Execution failed on sql 'SELECT c.FirstName, c.LastName
<start_of_invoice>': incomplete input
```

In [38]:

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

vn.ask(question=question)
```

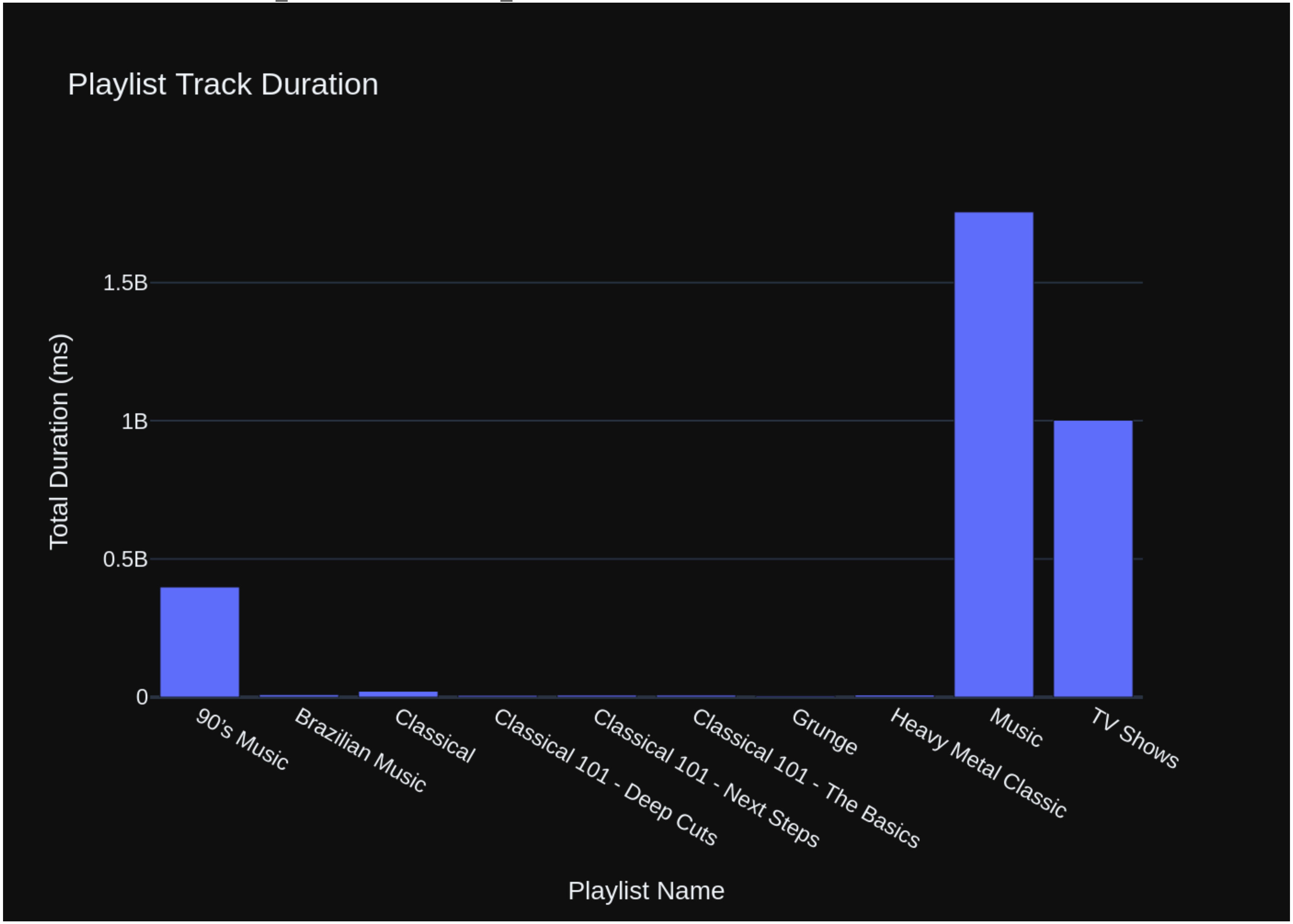
```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCRE
ATE TABLE "playlists"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCH
AR(120)\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTE
GER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (P
laylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FORE
IGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCRE
ATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(20
0)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    
Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMER
IC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DEL
ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE IND
EX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n
\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGE
R PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NU
LL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION\r\n)\n\nCREATE TABLE "genres"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    
Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Re
sponse 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 st
rings 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). \n
5. If the question has been asked and answered before, please repeat the answer exactly as it was given bef
ore. \n'}, {'role': 'user', 'content': '  \n    List all genres and the number of tracks in each genr
e:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.TrackId) AS TrackCount\nFROM genres g\nJOI
N tracks t ON g.GenreId = t.GenreId\nGROUP BY g.Name'}, {'role': 'user', 'content': ' \n    There are 3 tab
les: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are lin
ked by AlbumId,\n    Can you find the top 10 most popular artists based on the number of tracks\n'}, {'rol
e': 'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artists a\nJOIN albums al
ON a.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount
DESC\nLIMIT 10'}, {'role': 'user', 'content': '  \n    Find all tracks with a name containing "What" (case-
insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks\nWHERE Name LIKE '%What%'"}, {'rol
e': 'user', 'content': '  \n     Find the customer who bought the most albums in total quantity (across all
invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, SUM(ii.Quantity) AS Tota
lAlbumsBought\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON
i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.Album
Id\nGROUP BY c.FirstName, c.LastName\nORDER BY TotalAlbumsBought DESC\nLIMIT 1'}, {'role': 'user', 'conten
t': '  \n    Hint: album quantity is found in invoice_items, \n    \n    Find the top 5 customers who bough
```

```
t the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.
FirstName, c.LastName, SUM(ii.Quantity) AS TotalAlbumsBought\nFROM customers c\nJOIN invoices i ON c.Custom
erId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.T
rackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.FirstName, c.LastName\nORDER BY TotalAlbumsBou
ght DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    List all albums and their corresponding artist na
mes  \n'}, {'role': 'assistant', 'content': 'SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists O
N albums.ArtistId = artists.ArtistId'}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive 
tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT Name, UnitPrice\nFROM tracks\nO
RDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find all invoices since 2010 and th
e total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT InvoiceDate, SUM(Total) AS TotalAmou
nt\nFROM invoices\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {'role': 'user', 'content': '
\n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM in
voices\nWHERE Total > 10'}, {'role': 'user', 'content': 'Can you list all tables in the SQLite database cat
alog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {'role':
'user', 'content': '  \n     Get all playlists containing at least 10 tracks and the total duration of thos
e tracks:\n'}]
Ollama parameters:
model=codegemma: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 \"playlist_track\" (TrackId)\n\nC
REATE TABLE \"playlists\"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NV
ARCHAR(120)\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackI
d INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN 
KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n
FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVA
RCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGE
R,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitP
rice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELET
E NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DE
LETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (Gen
reId)\n\nCREATE INDEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON \"tr
acks\" (MediaTypeId)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"albums
\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r
\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n    GenreId INTEGER PRIMARY 
KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook da
```

```
tabase invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please gen
erate a valid SQL query without any explanations for the question. \n2. If the provided context is almost s
ufficient but requires knowledge of a specific string in a particular column, please generate an intermedia
te SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermedi
ate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please
use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the
answer exactly as it was given before. \n"}, {"role": "user", "content": "  \n    List all genres and the n
umber of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.TrackId) AS Tr
ackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"role": "user", "conte
nt": " \n    There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistI
d, albums and tracks are linked by AlbumId,\n    Can you find the top 10 most popular artists based on the
number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM
artists a\nJOIN albums al ON a.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY a.
Name\nORDER BY TrackCount DESC\nLIMIT 10"}, {"role": "user", "content": "  \n    Find all tracks with a nam
e containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks\nWHERE
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.FirstName, c.LastNam
e, SUM(ii.Quantity) AS TotalAlbumsBought\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId
\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums
al ON t.AlbumId = al.AlbumId\nGROUP BY c.FirstName, c.LastName\nORDER BY TotalAlbumsBought DESC\nLIMIT 1"},
{"role": "user", "content": "  \n    Hint: album quantity is found in invoice_items, \n    \n    Find the t
op 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistan
t", "content": "SELECT c.FirstName, c.LastName, SUM(ii.Quantity) AS TotalAlbumsBought\nFROM customers c\nJO
IN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tra
cks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.FirstName, c.LastName
\nORDER BY TotalAlbumsBought DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    List all albums and thei
r corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT albums.Title, artists.Name\nFR
OM albums\nJOIN artists ON albums.ArtistId = artists.ArtistId"}, {"role": "user", "content": "  \n    Find
the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT Name,
UnitPrice\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    Find all i
nvoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT InvoiceDat
e, SUM(Total) AS TotalAmount\nFROM invoices\nWHERE InvoiceDate >= '2010-01-01'\nGROUP BY InvoiceDate"}, {"r
ole": "user", "content": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "assistant",
"content": "SELECT * FROM invoices\nWHERE Total > 10"}, {"role": "user", "content": "Can you list all table
s in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master\nWHER
E type = 'table'"}, {"role": "user", "content": "  \n     Get all playlists containing at least 10 tracks a
nd the total duration of those tracks:\n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:58:51.453013152Z', 'message': {'role': 'assistan
t', 'content': 'SELECT p.Name, COUNT(pt.TrackId) AS TrackCount, SUM(t.Milliseconds) AS TotalDuration\nFROM
playlists p\nJOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId\nJOIN tracks t ON pt.TrackId = t.TrackI
d\nGROUP BY p.Name\nHAVING TrackCount >= 10'}, 'done_reason': 'stop', 'done': True, 'total_duration': 79620
```

```
811616, 'load_duration': 626263, 'prompt_eval_count': 1728, 'prompt_eval_duration': 61893265000, 'eval_coun
t': 81, 'eval_duration': 17141607000}
SELECT p.Name, COUNT(pt.TrackId) AS TrackCount, SUM(t.Milliseconds) AS TotalDuration
FROM playlists p
JOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId
JOIN tracks t ON pt.TrackId = t.TrackId
GROUP BY p.Name
HAVING TrackCount >= 10
SELECT p.Name, COUNT(pt.TrackId) AS TrackCount, SUM(t.Milliseconds) AS TotalDuration
FROM playlists p
JOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId
JOIN tracks t ON pt.TrackId = t.TrackId
GROUP BY p.Name
HAVING TrackCount >= 10
                         Name  TrackCount  TotalDuration
0                  90’s Music        1477      398705153
1             Brazilian Music          39        9486559
2                   Classical          75       21770592
3   Classical 101 - Deep Cuts          25        6755730
4  Classical 101 - Next Steps          25        7575051
5  Classical 101 - The Basics          25        7439811
6                      Grunge          15        4122018
7         Heavy Metal Classic          26        8206312
8                       Music        6580     1755366166
9                    TV Shows         426     1002189914
Ollama parameters:
model=codegemma: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.Name, COUNT(pt.
TrackId) AS TrackCount, SUM(t.Milliseconds) AS TotalDuration\nFROM playlists p\nJOIN playlist_track pt ON
p.PlaylistId = pt.PlaylistId\nJOIN tracks t ON pt.TrackId = t.TrackId\nGROUP BY p.Name\nHAVING TrackCount >
= 10\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes give
s:\n Name             object\nTrackCount        int64\nTotalDuration     int64\ndtype: object"}, {"role":
"user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume t
he data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicato
r. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T19:59:21.59289396Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Name', y='TotalDuration', hov
er_name='TrackCount',\n             title='Playlist Track Duration')\n\nfig.update_layout(xaxis_title='Play
list Name', yaxis_title='Total Duration (ms)')\n\nif len(df) == 1:\n    fig.add_trace(px.indicator(value=df
['TotalDuration'].iloc[0], title='Total Duration'))\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': Tr
ue, 'total_duration': 30111702984, 'load_duration': 44004942, 'prompt_eval_count': 245, 'prompt_eval_durati
on': 8686637000, 'eval_count': 112, 'eval_duration': 21331790000}
```



Out[38]:

```
('SELECT p.Name, COUNT(pt.TrackId) AS TrackCount, SUM(t.Milliseconds) AS TotalDuration\nFROM playlists p\n
JOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId\nJOIN tracks t ON pt.TrackId = t.TrackId\nGROUP BY
p.Name\nHAVING TrackCount >= 10',
                           Name  TrackCount  TotalDuration
 0                   90’s Music        1477      398705153
 1              Brazilian Music          39        9486559
 2                    Classical          75       21770592
 3    Classical 101 - Deep Cuts          25        6755730
 4   Classical 101 - Next Steps          25        7575051
 5   Classical 101 - The Basics          25        7439811
 6                       Grunge          15        4122018
 7          Heavy Metal Classic          26        8206312
 8                        Music        6580     1755366166
 9                     TV Shows         426     1002189914,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': '<b>%{hovertext}</b><br><br>Name=%{x}<br>TotalDuration=%{y}<extra></extra
>',
               'hovertext': array([1477.,   39.,   75.,   25.,   25.,   25.,   15.,   26., 6580.,  426.]),
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['90’s Music', 'Brazilian Music', 'Classical',
                           'Classical 101 - Deep Cuts', 'Classical 101 - Next Steps',
                           'Classical 101 - The Basics', 'Grunge', 'Heavy Metal Classic', 'Music',
                           'TV Shows'], dtype=object),
               'xaxis': 'x',
               'y': array([ 398705153,    9486559,   21770592,    6755730,    7575051,    7439811,
                              4122018,    8206312, 1755366166, 1002189914]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Playlist Track Duration'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Playlist Name'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total Duration (ms)'}}}
 }))
```

```
In [39]: question = """
              Identify artists who have albums with tracks appearing in multiple genres:


         """

         vn.ask(question=question)
```

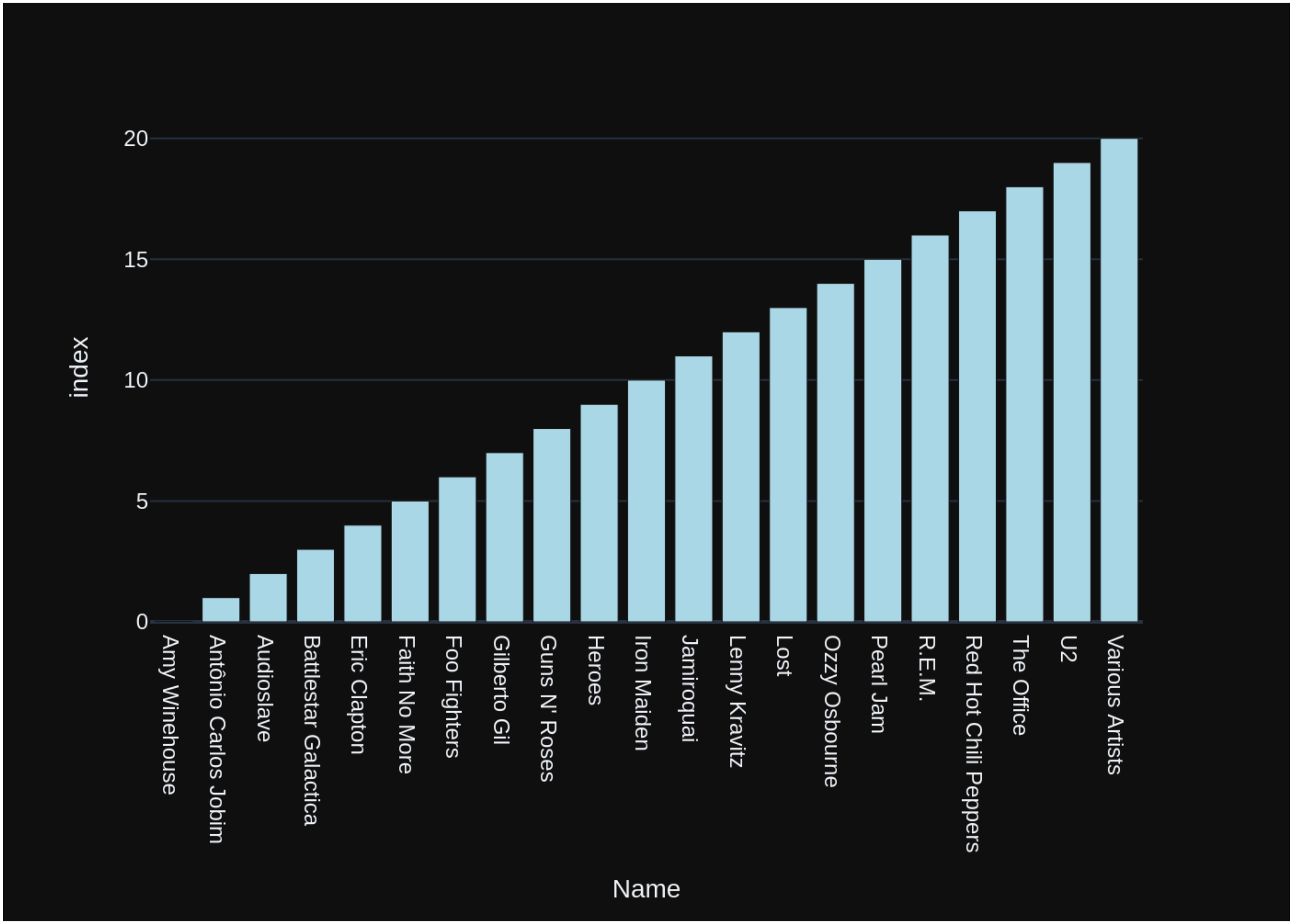
```
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NU
LL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    By
tes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (Al
bumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (G
enreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media
_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistI
d ON "albums" (ArtistId)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK_TrackAl
bumId ON "tracks" (AlbumId)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT
NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (Arti
stId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX
IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE TABLE "genres"\r\n(\r\n    GenreId INTEGER PRIMARY
KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "pla
ylist_track" (TrackId)\n\nCREATE TABLE "artists"\r\n(\r\n    ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT
NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT N
ULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackI
d),\r\n    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDA
TE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPD
ATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\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. I
f the question has been asked and answered before, please repeat the answer exactly as it was given before.
\n'}, {'role': 'user', 'content': ' \n    There are 3 tables: artists, albums and tracks, where albums and
artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n    Can you find the top 10 most
popular artists based on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT a.Name, COUNT
(t.TrackId) AS TrackCount\nFROM artists a\nJOIN albums al ON a.ArtistId = al.ArtistId\nJOIN tracks t ON al.
AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMIT 10'}, {'role': 'user', 'content': '
\n    List all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT
albums.Title, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId = artists.ArtistId'}, {'role': 'us
er', 'content': '  \n    List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant',
'content': 'SELECT g.Name, COUNT(t.TrackId) AS TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.Ge
nreId\nGROUP BY g.Name'}, {'role': 'user', 'content': '  \n    Hint: album quantity is found in invoice_ite
ms, \n    \n    Find the top 5 customers who bought the most albums in total quantity (across all invoice
s):\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, SUM(ii.Quantity) AS TotalAlbumsB
ought\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.Invoice
Id = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP
BY c.FirstName, c.LastName\nORDER BY TotalAlbumsBought DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n
```

```
Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assis
tant', 'content': 'SELECT c.FirstName, c.LastName, SUM(ii.Quantity) AS TotalAlbumsBought\nFROM customers c
\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN
tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.FirstName, c.LastN
ame\nORDER BY TotalAlbumsBought DESC\nLIMIT 1'}, {'role': 'user', 'content': '  \n     Get all playlists co
ntaining at least 10 tracks and the total duration of those tracks:\n'}, {'role': 'assistant', 'content':
'SELECT p.Name, COUNT(pt.TrackId) AS TrackCount, SUM(t.Milliseconds) AS TotalDuration\nFROM playlists p\nJO
IN playlist_track pt ON p.PlaylistId = pt.PlaylistId\nJOIN tracks t ON pt.TrackId = t.TrackId\nGROUP BY p.N
ame\nHAVING TrackCount >= 10'}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive tracks
(based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT Name, UnitPrice\nFROM tracks\nORDER BY
UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find all tracks with a name containing "Wha
t" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks\nWHERE Name LIKE '%Wha
t%'"}, {'role': 'user', 'content': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'as
sistant', 'content': "SELECT name FROM sqlite_master\nWHERE type = 'table'"}, {'role': 'user', 'content': '
\n     Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName,
c.LastName, COUNT(*) AS InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGRO
UP BY c.FirstName, c.LastName\nORDER BY InvoiceCount DESC\nLIMIT 1'}, {'role': 'user', 'content': '  \n
Identify artists who have albums with tracks appearing in multiple genres:\n\n\n'}]
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the
question. Your response should ONLY be based on the given context and follow the response guidelines and fo
rmat instructions. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREM
ENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT
NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n
Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\"
(AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES \"genres
\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES
\"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_Albu
mArtistId ON \"albums\" (ArtistId)\n\nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX
IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY
AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOR
EIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE INDEX IFK_TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE TABLE \"genres\"\r\n(\r\n    Ge
nreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_Playli
stTrackTrackId ON \"playlist_track\" (TrackId)\n\nCREATE TABLE \"artists\"\r\n(\r\n    ArtistId INTEGER PRI
MARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n
PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY
KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t
\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoi
ce means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a vali
d SQL query without any explanations for the question. \n2. If the provided context is almost sufficient bu
t requires knowledge of a specific string in a particular column, please generate an intermediate SQL query
to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3.
If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most
relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactl
y as it was given before. \n"}, {"role": "user", "content": " \n    There are 3 tables: artists, albums and
tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n    Can
you find the top 10 most popular artists based on the number of tracks\n"}, {"role": "assistant", "conten
t": "SELECT a.Name, COUNT(t.TrackId) AS TrackCount\nFROM artists a\nJOIN albums al ON a.ArtistId = al.Artis
tId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TrackCount DESC\nLIMIT 10"}, {"rol
e": "user", "content": "  \n    List all albums and their corresponding artist names  \n"}, {"role": "assis
tant", "content": "SELECT albums.Title, artists.Name\nFROM albums\nJOIN artists ON albums.ArtistId = artist
s.ArtistId"}, {"role": "user", "content": "  \n    List all genres and the number of tracks in each genr
e:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.TrackId) AS TrackCount\nFROM genres g\nJOI
N tracks t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"role": "user", "content": "  \n    Hint: album qu
antity is found in invoice_items, \n    \n    Find the top 5 customers who bought the most albums in total
quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastName, SUM
(ii.Quantity) AS TotalAlbumsBought\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN
invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON
t.AlbumId = al.AlbumId\nGROUP BY c.FirstName, c.LastName\nORDER BY TotalAlbumsBought DESC\nLIMIT 5"}, {"rol
e": "user", "content": "  \n     Find the customer who bought the most albums in total quantity (across all
invoices): \n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastName, SUM(ii.Quantity) AS Tota
lAlbumsBought\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON
i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.Album
Id\nGROUP BY c.FirstName, c.LastName\nORDER BY TotalAlbumsBought DESC\nLIMIT 1"}, {"role": "user", "conten
t": "  \n     Get all playlists containing at least 10 tracks and the total duration of those tracks:\n"},
{"role": "assistant", "content": "SELECT p.Name, COUNT(pt.TrackId) AS TrackCount, SUM(t.Milliseconds) AS To
talDuration\nFROM playlists p\nJOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId\nJOIN tracks t ON pt.
TrackId = t.TrackId\nGROUP BY p.Name\nHAVING TrackCount >= 10"}, {"role": "user", "content": "  \n    Find
the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT Name,
UnitPrice\nFROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    Find all t
racks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FR
OM tracks\nWHERE Name LIKE '%What%'"}, {"role": "user", "content": "Can you list all tables in the SQLite d
atabase catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master\nWHERE type = 'tabl
e'"}, {"role": "user", "content": "  \n     Find the customer with the most invoices \n"}, {"role": "assist
ant", "content": "SELECT c.FirstName, c.LastName, COUNT(*) AS InvoiceCount\nFROM customers c\nJOIN invoices
i ON c.CustomerId = i.CustomerId\nGROUP BY c.FirstName, c.LastName\nORDER BY InvoiceCount DESC\nLIMIT 1"},
{"role": "user", "content": "  \n     Identify artists who have albums with tracks appearing in multiple ge
nres:\n\n\n"}]
```

```
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T20:00:43.635734764Z', 'message': {'role': 'assistan
t', 'content': 'SELECT a.Name\nFROM artists a\nJOIN albums al ON a.ArtistId = al.ArtistId\nJOIN tracks t ON
al.AlbumId = t.AlbumId\nJOIN genres g ON t.GenreId = g.GenreId\nGROUP BY a.Name\nHAVING COUNT(DISTINCT g.Ge
nreId) > 1'}, 'done_reason': 'stop', 'done': True, 'total_duration': 81943055052, 'load_duration': 624455,
'prompt_eval_count': 1793, 'prompt_eval_duration': 64879510000, 'eval_count': 78, 'eval_duration': 16478580
000}
SELECT a.Name
FROM artists a
JOIN albums al ON a.ArtistId = al.ArtistId
JOIN tracks t ON al.AlbumId = t.AlbumId
JOIN genres g ON t.GenreId = g.GenreId
GROUP BY a.Name
HAVING COUNT(DISTINCT g.GenreId) > 1
SELECT a.Name
FROM artists a
JOIN albums al ON a.ArtistId = al.ArtistId
JOIN tracks t ON al.AlbumId = t.AlbumId
JOIN genres g ON t.GenreId = g.GenreId
GROUP BY a.Name
HAVING COUNT(DISTINCT g.GenreId) > 1
                     Name
0           Amy Winehouse
1    Antônio Carlos Jobim
2              Audioslave
3    Battlestar Galactica
4            Eric Clapton
5           Faith No More
6            Foo Fighters
7            Gilberto Gil
8           Guns N' Roses
9                  Heroes
10            Iron Maiden
11             Jamiroquai
12          Lenny Kravitz
13                   Lost
14          Ozzy Osbourne
15              Pearl Jam
16                 R.E.M.
17  Red Hot Chili Peppers
18             The Office
19                     U2
```

```
20        Various Artists
Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '  \n     Identify artists who have albums with tracks appearing
in multiple genres:\n\n\n'\n\nThe DataFrame was produced using this query: SELECT a.Name\nFROM artists a\nJ
OIN albums al ON a.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nJOIN genres g ON t.Genr
eId = g.GenreId\nGROUP BY a.Name\nHAVING COUNT(DISTINCT g.GenreId) > 1\n\nThe following is information abou
t the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name    object\ndtype: object"}, {"rol
e": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assu
me the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indic
ator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-06-15T20:01:00.724681466Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Name', y=df.index)\nfig.updat
e_traces(marker_color='lightblue')\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duratio
n': 17061278142, 'load_duration': 41935127, 'prompt_eval_count': 222, 'prompt_eval_duration': 7604117000,
'eval_count': 50, 'eval_duration': 9364729000}
```


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

```
Out[39]: ('SELECT a.Name\nFROM artists a\nJOIN albums al ON a.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId =
         t.AlbumId\nJOIN genres g ON t.GenreId = g.GenreId\nGROUP BY a.Name\nHAVING COUNT(DISTINCT g.GenreId) > 1',
                                 Name
          0           Amy Winehouse
          1    Antônio Carlos Jobim
          2              Audioslave
          3    Battlestar Galactica
          4            Eric Clapton
          5           Faith No More
          6            Foo Fighters
          7            Gilberto Gil
          8           Guns N' Roses
          9                  Heroes
          10            Iron Maiden
          11             Jamiroquai
          12          Lenny Kravitz
          13                   Lost
          14          Ozzy Osbourne
          15              Pearl Jam
          16                 R.E.M.
          17  Red Hot Chili Peppers
          18             The Office
          19                     U2
          20        Various Artists,
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'Name=%{x}<br>index=%{y}<extra></extra>',
                        'legendgroup': '',
                        'marker': {'color': 'lightblue', 'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array(['Amy Winehouse', 'Antônio Carlos Jobim', 'Audioslave',
                                    'Battlestar Galactica', 'Eric Clapton', 'Faith No More', 'Foo Fighters',
                                    'Gilberto Gil', "Guns N' Roses", 'Heroes', 'Iron Maiden', 'Jamiroquai',
                                    'Lenny Kravitz', 'Lost', 'Ozzy Osbourne', 'Pearl Jam', 'R.E.M.',
                                    'Red Hot Chili Peppers', 'The Office', 'U2', 'Various Artists'],
                                   dtype=object),
                        'xaxis': 'x',
                        'y': array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
                                    18, 19, 20]),
                        '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': 'index'}}}
          }))
```

Check completion time

In [ ]:

In [43]:

```
ts_stop = time()

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

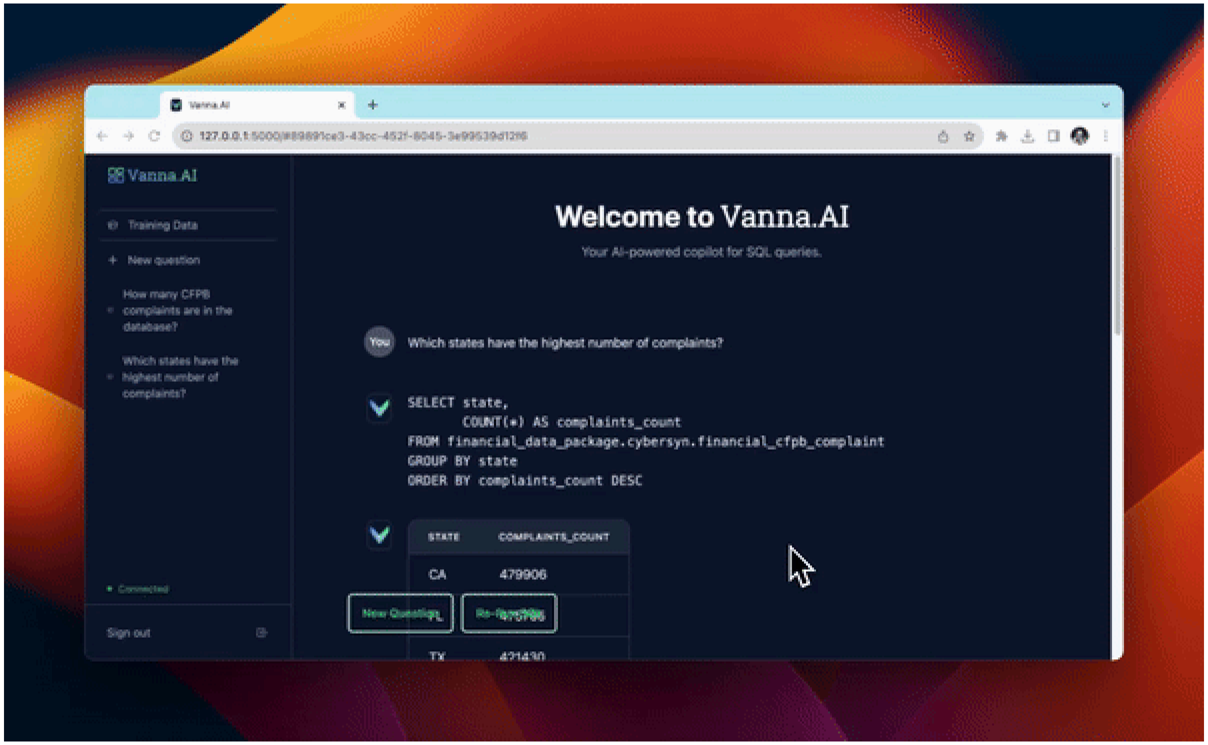
```
test running on 'ducklover1' with 'codegemma' LLM took : 2117.36 sec
```

In [42]:

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

```
2024-06-15 16:03:41.744951
```

Launch the User Interface



How many CFPB complaints are in the database?

Which states have the highest number of complaints?

Welcome to Vanna.AI

Your AI-powered copilot for SQL queries.

Which states have the highest number of complaints?

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

STATE	COMPLAINTS_COUNT
CA	479906
FL	[illegible]
TX	421430

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

Next Steps

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

- Streamlit app
- Flask app
- Slackbot