Generating SQL for SQLite using Ollama, ChromaDB

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

Which LLM do you want to use?

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

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

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

Use any other vector database. Requires additional setup.

Setup

!pip install 'vanna[chromadb]'

In [1]:
```
model_name = 'gpt-4o-mini'
file_db = "~/Downloads/chinook.sqlite"
```

In [2]:
```
from api_key_store import ApiKeyStore
s = ApiKeyStore()

openai_api_key = s.get_api_key(provider="OPENAI")
```

openai_api_key

In [3]:
```
from vanna.openai import OpenAI_Chat
from vanna.chromadb.chromadb_vector import ChromaDB_VectorStore
```

In [4]:
```
class MyVanna(ChromaDB_VectorStore, OpenAI_Chat):
    def __init__(self, config=None):
        ChromaDB_VectorStore.__init__(self, config=config)
        OpenAI_Chat.__init__(self, config=config)

config = {
    'api_key': openai_api_key,
    'model': model_name
}
vn = MyVanna(config=config)
```

Which database do you want to query?

- Postgres
- Microsoft SQL Server
- DuckDB
- Snowflake
- BigQuery

- [Selected] SQLite
- Other Database

  Use Vanna to generate queries for any SQL database

In [5]:
```
import os
import re
from time import time
```

In [6]:
```
# file_db = "./db/gpt3sql.sqlite"

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]:
```
clean_and_train = True  # False
```

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

```
Hostname: ducklover1
```

In [10]:
```
def remove_collections(collection_name=None, ACCEPTED_TYPES = ["sql", "ddl", "documentation"]):
    if not collection_name:
        collections = ACCEPTED_TYPES
    elif isinstance(collection_name, str):
        collections = [collection_name]
    elif isinstance(collection_name, list):
        collections = collection_name
    else:
        print(f"\t{collection_name} is unknown: Skipped")
        return

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

```
            continue

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

In [11]:
```
def strip_brackets(ddl):
    """
    This function removes square brackets from table and column names in a DDL script.

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

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

In [12]:
```
if clean_and_train:
    remove_collections()
```

Training

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

In [13]:
```
# show training data
training_data = vn.get_training_data()
training_data
```

Out[13]:

	id	question	content	training_data_type

In [14]:
```
df_ddl = vn.run_sql("SELECT type, sql FROM sqlite_master WHERE sql is not null")
```

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

Out[15]:

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

In [16]:
```
if clean_and_train:
    for ddl in df_ddl['sql'].to_list():
        ddl = strip_brackets(ddl)
        vn.train(ddl=ddl)

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

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

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

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

```
Adding ddl: CREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)
Adding ddl: CREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)
Adding ddl: CREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)
Adding ddl: CREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)
Adding ddl: CREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)
Adding ddl: CREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)
Adding ddl: CREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)
Adding ddl: CREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)
Adding ddl: CREATE TABLE sqlite_stat1(tbl,idx,stat)
Adding documentation....
```

In [ ]:

Asking the AI

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

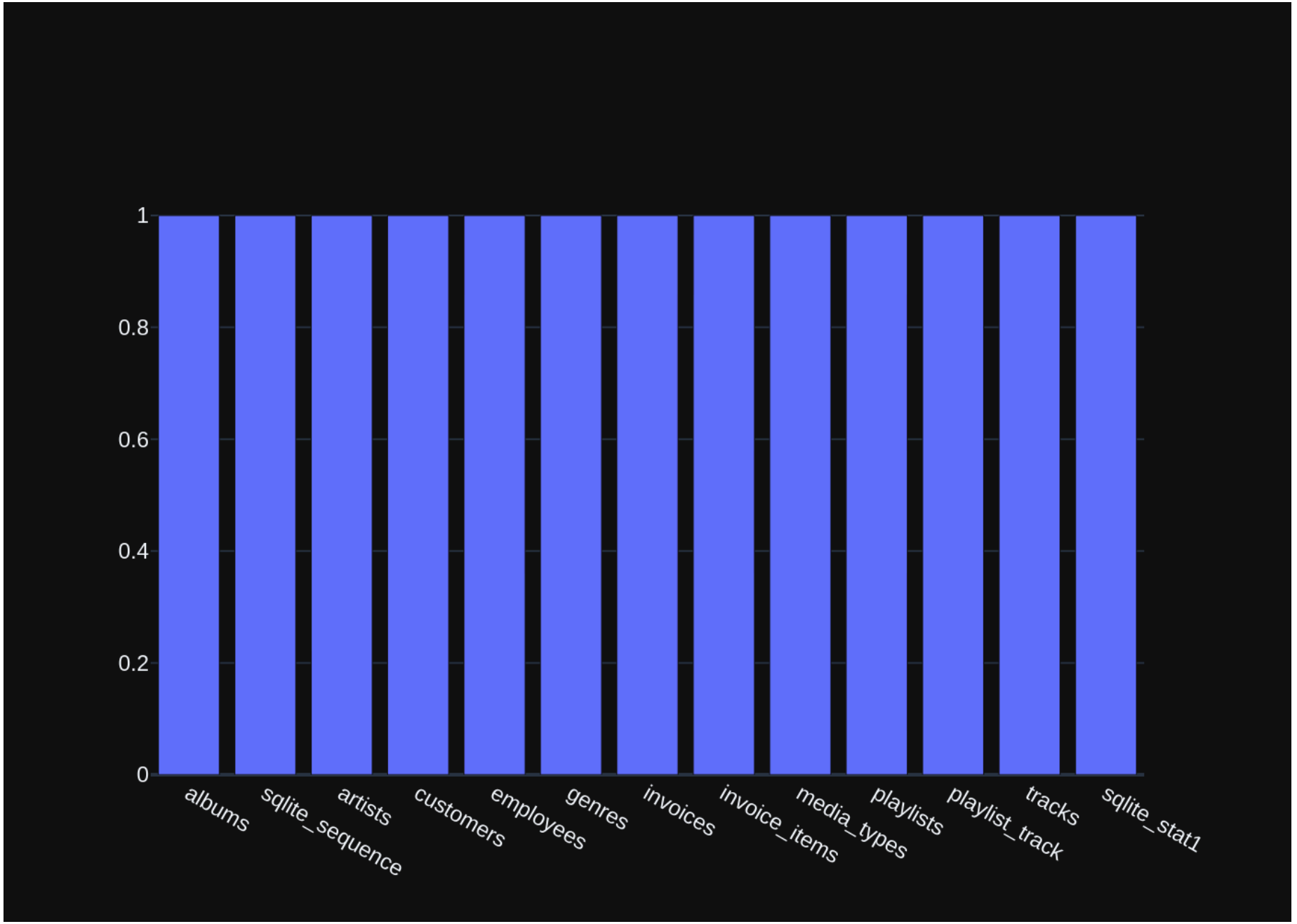
In [17]:
```
ts_start = time()
```

In [18]:
```
vn.ask(question="Show me a list of tables in the SQLite database")
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREA
TE TABLE "playlists"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r
\n)\n\nCREATE TABLE "media_types"\r\n(\r\n    MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NV
ARCHAR(120)\r\n)\n\nCREATE TABLE "artists"\r\n(\r\n    ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    N
ame NVARCHAR(120)\r\n)\n\nCREATE TABLE "genres"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT
NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT
NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE N
O ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT
NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    Genr
eId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    Unit
Price NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTIO
N ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPD
ATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION O
N UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Gu
idelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for
the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a par
ticular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the q
uery with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can
\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered befor
e, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': 'Show me a list of tab
les in the SQLite database'}]
Using model gpt-4o-mini for 652.0 tokens (approx)
LLM Response: ```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
Using model gpt-4o-mini for 168.0 tokens (approx)
```


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

Out[18]:
```
("SELECT name FROM sqlite_master WHERE type='table';",
                name
0             albums
1   sqlite_sequence
2            artists
3          customers
4          employees
5             genres
6           invoices
7      invoice_items
8        media_types
9          playlists
10    playlist_track
11            tracks
12     sqlite_stat1,
Figure({
    'data': [{'type': 'bar',
              'x': array(['albums', 'sqlite_sequence', 'artists', 'customers', 'employees',
                          'genres', 'invoices', 'invoice_items', 'media_types', 'playlists',
                          'playlist_track', 'tracks', 'sqlite_stat1'], dtype=object),
              'y': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]}],
    'layout': {'template': '...'}
}))
```

In [19]:
```
vn.ask(question="which table stores customer's orders")
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r
\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    Inv
oiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  N
OT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId)
REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFE
RENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n
CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVAR
CHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State
NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVA
RCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFER
ENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite_sequence
(name,seq)\n\nCREATE TABLE "playlists"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Nam
e NVARCHAR(120)\r\n)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\n\n===Additional Context \n\nIn the chinook databa
se 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 distinc
t strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is i
nsufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the ques
tion has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'us
er', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FR
OM sqlite_master WHERE type='table';"}, {'role': 'user', 'content': "which table stores customer's orders"}]
Using model gpt-4o-mini for 683.75 tokens (approx)
LLM Response: The "invoices" table stores customer's orders.
The "invoices" table stores customer's orders.
Couldn't run sql:  Execution failed on sql 'The "invoices" table stores customer's orders.': near "The": syntax erro
r
```

In [20]:
```
vn.ask(question="How many records are in table called customer")
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r
\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n    Custome
rId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(2
0)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARC
HAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR
(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES
"employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r
\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId I
NTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (I
nvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Tra
ckId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "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 UPDA
TE NO ACTION\r\n)\n\n\n===Additional Context \n\n===Response Guidelines \n1. If the provided context is sufficient,
please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost
sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL q
uery to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If t
he provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant ta
ble(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given be
fore. \n'}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant', 'c
ontent': "SELECT name FROM sqlite_master WHERE type='table';"}, {'role': 'user', 'content': 'How many records are in
table called customer'}]
Using model gpt-4o-mini for 690.25 tokens (approx)
LLM Response: SELECT COUNT(*) FROM customers;
Extracted SQL: SELECT COUNT(*) FROM customers;
SELECT COUNT(*) FROM customers;
   COUNT(*)
0        59
Using model gpt-4o-mini for 163.5 tokens (approx)
```


Total Customer Records
59

Out[20]:
```
('SELECT COUNT(*) FROM customers;',
   COUNT(*)
0        59,
Figure({
    'data': [{'mode': 'number', 'title': {'text': 'Total Customer Records'}, 'type': 'indicator', 'value': 59}],
    'layout': {'template': '...'}
}))
```

In [21]:
```
vn.ask(question="How many customers are there")
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r
\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "cu
stomers" (SupportRepId)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n
Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    Posta
lCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n
SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACT
ION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE TABLE "invoi
ce_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NUL
L,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r
\n    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n
FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE I
NDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_item
s" (TrackId)\n\n\n===Additional Context \n\n===Response Guidelines \n1. If the provided context is sufficient, pleas
e generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost suffi
cient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query
to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the pr
ovided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table
(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given befor
e. \n'}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assistant', 'conten
t': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite databas
e'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table';"}, {'role': 'user', 'conte
nt': 'How many customers are there'}]
Using model gpt-4o-mini for 707.25 tokens (approx)
LLM Response: SELECT COUNT(*) FROM customers;
Extracted SQL: SELECT COUNT(*) FROM customers;
SELECT COUNT(*) FROM customers;
   COUNT(*)
0        59
Using model gpt-4o-mini for 159.25 tokens (approx)
```


Total Customers
59

Out[21]:
```
('SELECT COUNT(*) FROM customers;',
   COUNT(*)
0        59,
Figure({
    'data': [{'mode': 'number', 'title': {'text': 'Total Customers'}, 'type': 'indicator', 'value': 59}],
    'layout': {'template': '...'}
}))
```

In [ ]:
```
```

In [22]:
```
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 3, updating n_results = 3
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r
\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n    Custome
rId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(2
0)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARC
HAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR
(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES
"employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r
\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId I
NTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (I
nvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Tra
ckId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "media_type
s"\r\n(\r\n    MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDE
X IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\n\n===Additiona
l 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 contex
t is almost sufficient but requires knowledge of a specific string in a particular column, please generate an interm
ediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_s
ql \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 w
as given before. \n'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content':
'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': 'How many records are in table called customer'},
{'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': 'Show me a list of
tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='tabl
e';"}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}]
Using model gpt-4o-mini for 729.75 tokens (approx)
LLM Response: intermediate_sql
SELECT DISTINCT Country FROM customers;
The LLM is not allowed to see the data in your database. Your question requires database introspection to generate t
he necessary SQL. Please set allow_llm_to_see_data=True to enable this.
Couldn't run sql:  Execution failed on sql 'The LLM is not allowed to see the data in your database. Your question r
equires database introspection to generate the necessary SQL. Please set allow_llm_to_see_data=True to enable thi
s.': near "The": syntax error
```

More SQL questions

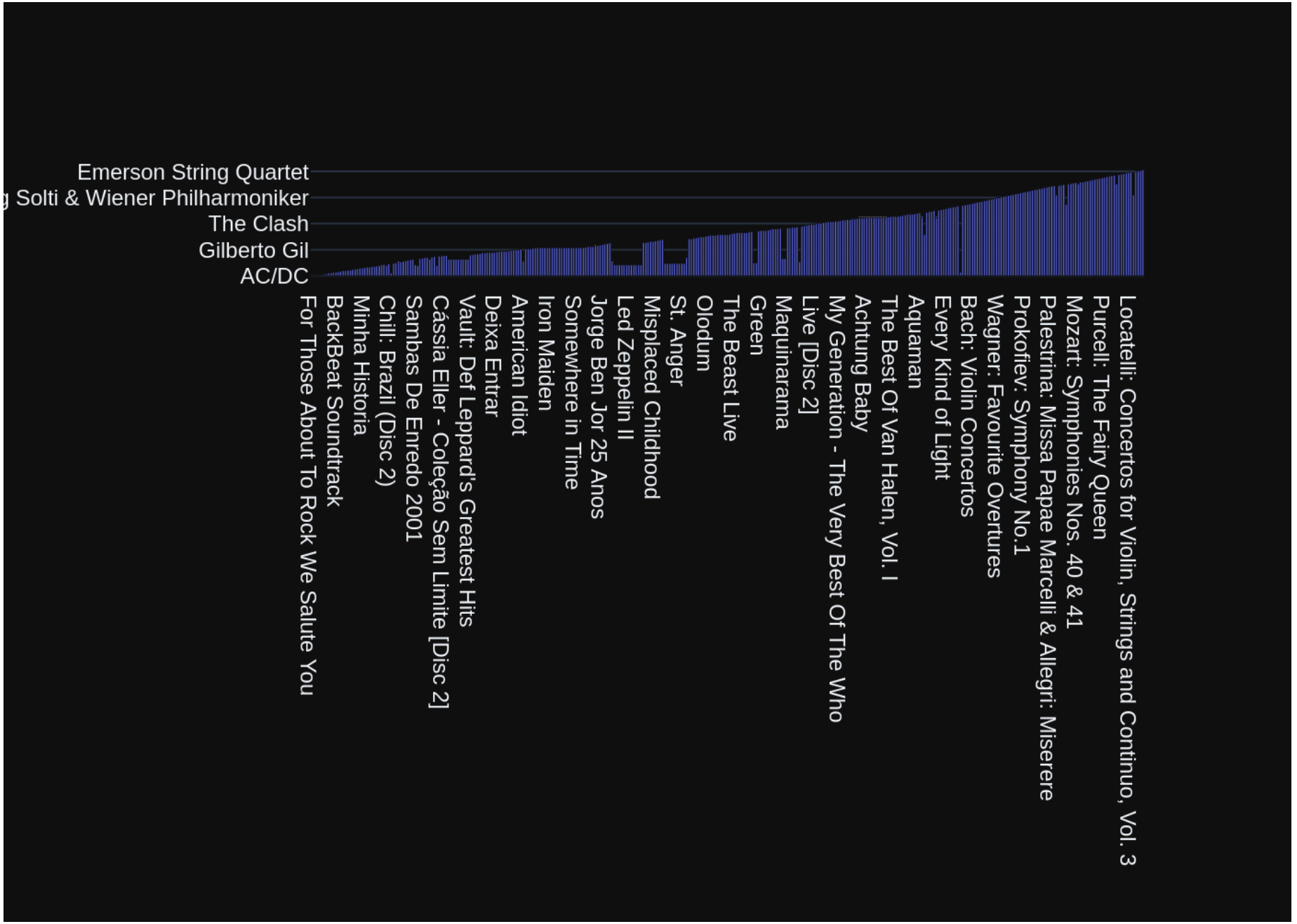
see `sample-sql-queries-sqlite-chinook.ipynb`

In [23]:
```
question = """
    List all albums and their corresponding artist names
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "albums"\r\n(\r\n
AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  N
OT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI
ON\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHA
R(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Compo
ser NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT
NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r
\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FO
REIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n    ArtistId INTEGER PRIMA
RY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)
\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE TABLE "playlists"\r\n(\r\n    Play
listId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "genres"\r\n(\r\n
GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_TrackMediaTyp
eId ON "tracks" (MediaTypeId)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Respo
nse Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanation
s for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in
a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend
the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it
can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered be
fore, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': 'Show me a list of
tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='tabl
e';"}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assistant', 'conten
t': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assis
tant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': '  \n    List all albums and their
corresponding artist names  \n'}]
Using model gpt-4o-mini for 689.5 tokens (approx)
LLM Response: SELECT albums.Title AS AlbumTitle, artists.Name AS ArtistName
FROM albums
JOIN artists ON albums.ArtistId = artists.ArtistId;
Extracted SQL: SELECT albums.Title AS AlbumTitle, artists.Name AS ArtistName
FROM albums
JOIN artists ON albums.ArtistId = artists.ArtistId;
SELECT albums.Title AS AlbumTitle, artists.Name AS ArtistName
FROM albums
JOIN artists ON albums.ArtistId = artists.ArtistId;
                                          AlbumTitle  \
0               For Those About To Rock We Salute You
1                                    Balls to the Wall
```

```
2                                   Restless and Wild
3                                   Let There Be Rock
4                                            Big Ones
..                                                ...
342                             Respighi:Pines of Rome
343  Schubert: The Late String Quartets & String Qu...
344                               Monteverdi: L'Orfeo
345                              Mozart: Chamber Music
346  Koyaanisqatsi (Soundtrack from the Motion Pict...

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

[347 rows x 2 columns]
Using model gpt-4o-mini for 197.75 tokens (approx)
```


Emerson String Quartet
g Solti & Wiener Philharmoniker
The Clash
Gilberto Gil
AC/DC
Locatelli: Concertos for Violin, Strings and Continuo, Vol. 3
Purcell: The Fairy Queen
Mozart: Symphonies Nos. 40 & 41
Palestrina: Missa Papae Marcelli & Allegri: Miserere
Prokofiev: Symphony No.1
Wagner: Favourite Overtures
Bach: Violin Concertos
Every Kind of Light
Aquaman
The Best Of Van Halen, Vol. I
Achtung Baby
My Generation - The Very Best Of The Who
Live [Disc 2]
Maquinarama
Green
The Beast Live
Olodum
St. Anger
Misplaced Childhood
Led Zeppelin II
Jorge Ben Jor 25 Anos
Somewhere in Time
Iron Maiden
American Idiot
Deixa Entrar
Vault: Def Leppard's Greatest Hits
Cássia Eller - Coleção Sem Limite [Disc 2]
Sambas De Enredo 2001
Chill: Brazil (Disc 2)
Minha Historia
BackBeat Soundtrack
For Those About To Rock We Salute You

Out[23]:
```
('SELECT albums.Title AS AlbumTitle, artists.Name AS ArtistName \nFROM albums \nJOIN artists ON albums.ArtistId =
artists.ArtistId;',
                                          AlbumTitle  \
 0               For Those About To Rock We Salute You
 1                                   Balls to the Wall
 2                                   Restless and Wild
 3                                   Let There Be Rock
 4                                         Big Ones
 ..                                             ...
 342                          Respighi:Pines of Rome
 343  Schubert: The Late String Quartets & String Qu...
 344                             Monteverdi: L'Orfeo
 345                           Mozart: Chamber Music
 346  Koyaanisqatsi (Soundtrack from the Motion Pict...

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

 [347 rows x 2 columns],
 Figure({
     'data': [{'text': array(['AC/DC', 'Accept', 'Accept', ...,
                              'C. Monteverdi, Nigel Rogers - Chiaroscuro; London Baroque; London Cornett & Sackb
u',
                              'Nash Ensemble', 'Philip Glass Ensemble'], dtype=object),
               '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),
```

```
                'y': array(['AC/DC', 'Accept', 'Accept', ...,
                            'C. Monteverdi, Nigel Rogers - Chiaroscuro; London Baroque; London Cornett & Sackbu',
                            'Nash Ensemble', 'Philip Glass Ensemble'], dtype=object)}],
     'layout': {'template': '...'}
 }))
```

In [24]:
```
question = """
    Find all tracks with a name containing "What" (case-insensitive)
"""

vn.ask(question=question)
```

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

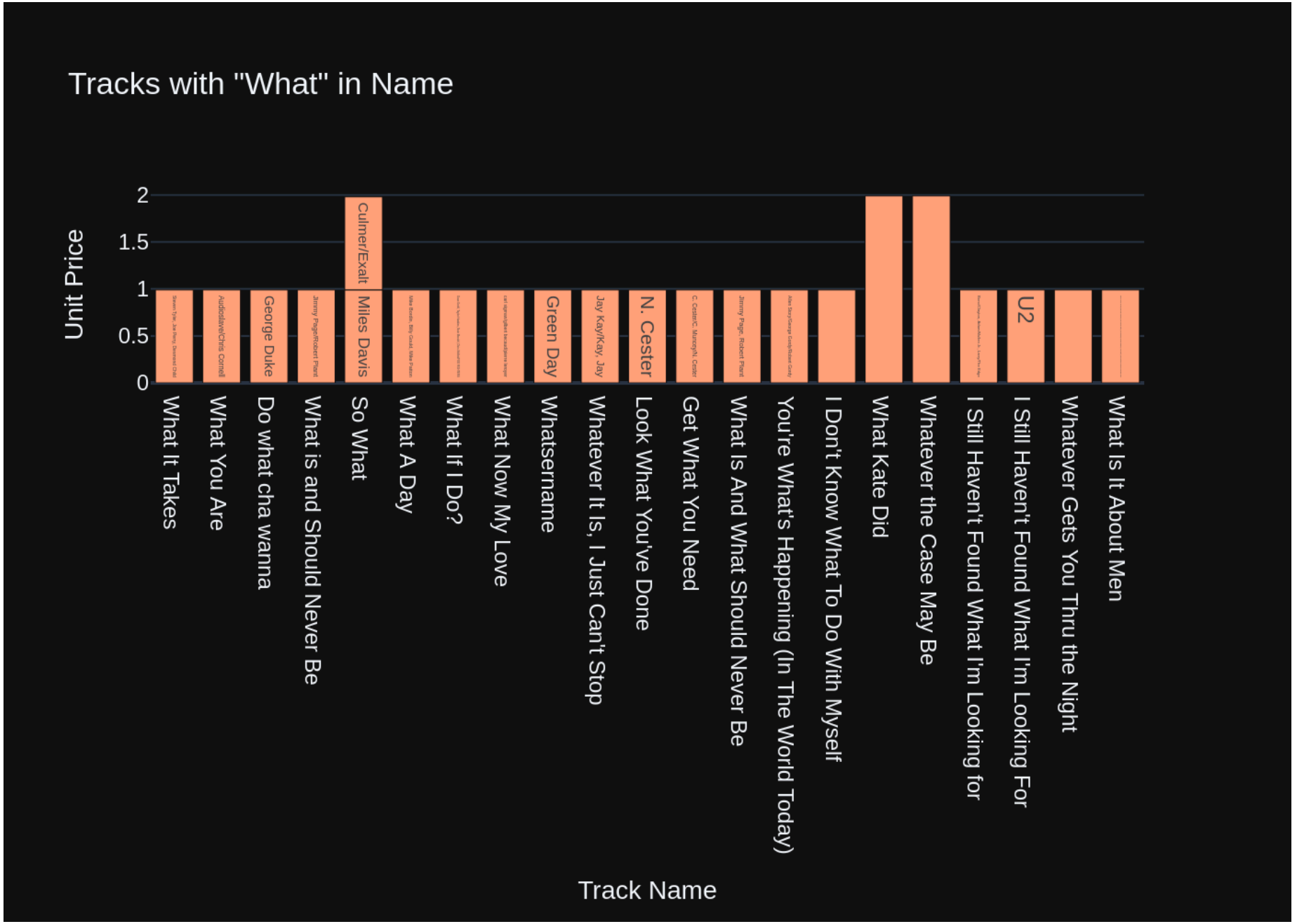
```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrac
kId ON "playlist_track" (TrackId)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT
NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    Genr
eId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    Unit
Price NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTIO
N ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPD
ATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION O
N UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDEX IFK_TrackMediaTypeId
ON "tracks" (MediaTypeId)\n\nCREATE TABLE "playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId
INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (Playli
stId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Track
Id) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLi
neTrackId ON "invoice_items" (TrackId)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "pla
ylists"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE
TABLE "genres"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n\n=
==Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided
context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the prov
ided context is almost sufficient but requires knowledge of a specific string in a particular column, please generat
e an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying int
ermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please u
se the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exa
ctly as it was given before. \n'}, {'role': 'user', 'content': '  \n    List all albums and their corresponding arti
st names  \n'}, {'role': 'assistant', 'content': 'SELECT albums.Title AS AlbumTitle, artists.Name AS ArtistName \nFR
OM albums \nJOIN artists ON albums.ArtistId = artists.ArtistId;'}, {'role': 'user', 'content': 'Show me a list of ta
bles in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='tabl
e';"}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT
(*) FROM customers;'}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assis
tant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': '  \n    Find all tracks with a na
me containing "What" (case-insensitive)\n'}]
Using model gpt-4o-mini for 759.25 tokens (approx)
LLM Response: SELECT * FROM tracks WHERE Name LIKE '%What%';
Extracted SQL: 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

Using model gpt-4o-mini for 223.5 tokens (approx)


Tracks with "What" in Name
Unit Price
2
1.5
1
0.5
0
Culmer/Exalt
Miles Davis
U2
Green Day
N. Cester
Jay Kay/Kay, Jay
Jimmy Page, Robert Plant
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[24]: ("SELECT * FROM tracks WHERE Name LIKE '%What%';",

	TrackId	Name	AlbumId	MediaTypeId	GenreId	Composer
0	26	What It Takes	5	1	1	Steven Tyler, Joe Perry, Desmond Child
1	88	What You Are	10	1	1	Audioslave/Chris Cornell
2	130	Do what cha wanna	13	1	2	George Duke
3	342	What is and Should Never Be	30	1	1	Jimmy Page/Robert Plant
4	607	So What	48	1	2	Miles Davis
5	960	What A Day	76	1	1	Mike Bordin, Billy Gould, Mike Patton
6	1000	What If I Do?	80	1	1	Dave Grohl, Taylor Hawkins, Nate Mendel, Chris...
7	1039	What Now My Love	83	1	12	carl sigman/gilbert becaud/pierre leroyer
8	1145	Whatsername	89	1	4	Green Day
9	1440	Whatever It Is, I Just Can't Stop	116	1	1	Jay Kay/Kay, Jay
10	1469	Look What You've Done	119	1	4	N. Cester
11	1470	Get What You Need	119	1	4	C. Cester/C. Muncey/N. Cester
12	1628	What Is And What Should Never Be	133	1	1	Jimmy Page, Robert Plant
13	1778	You're What's Happening (In The World Today)	146	1	14	Allen Story/George Gordy/Robert Gordy
14	1823	So What	149			
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			

```
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': [{'marker': {'color': 'lightsalmon'},
              'text': array(['Steven Tyler, Joe Perry, Desmond Child', 'Audioslave/Chris Cornell',
                             'George Duke', 'Jimmy Page/Robert Plant', 'Miles Davis',
                             'Mike Bordin, Billy Gould, Mike Patton',
                             'Dave Grohl, Taylor Hawkins, Nate Mendel, Chris Shiflett/FOO FIGHTERS',
                             'carl sigman/gilbert becaud/pierre leroyer', 'Green Day',
                             'Jay Kay/Kay, Jay', 'N. Cester', 'C. Cester/C. Muncey/N. Cester',
                             'Jimmy Page, Robert Plant', 'Allen Story/George Gordy/Robert Gordy',
```

```
                                'Culmer/Exalt', None, None, None,
                                'Bono/Clayton, Adam/Mullen Jr., Larry/The Edge', 'U2', None,
                                'Delroy "Chris" Cooper, Donovan Jackson, Earl Chinna Smith, Felix Howard, Gordon Wil
liams, Luke Smith, Paul Watson & Wilburn Squiddley Cole'],
                               dtype=object),
                'type': 'bar',
                'x': array(['What It Takes', 'What You Are', 'Do what cha wanna',
                            'What is and Should Never Be', 'So What', 'What A Day', 'What If I Do?',
                            'What Now My Love', 'Whatsername', "Whatever It Is, I Just Can't Stop",
                            "Look What You've Done", 'Get What You Need',
                            'What Is And What Should Never Be',
                            "You're What's Happening (In The World Today)", 'So What',
                            "I Don't Know What To Do With Myself", 'What Kate Did',
                            'Whatever the Case May Be',
                            "I Still Haven't Found What I'm Looking for",
                            "I Still Haven't Found What I'm Looking For",
                            'Whatever Gets You Thru the Night', 'What Is It About Men'],
                           dtype=object),
                'y': array([0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99,
                            0.99, 0.99, 0.99, 0.99, 1.99, 1.99, 0.99, 0.99, 0.99, 0.99])}],
     'layout': {'template': '...',
                'title': {'text': 'Tracks with "What" in Name'},
                'xaxis': {'title': {'text': 'Track Name'}},
                'yaxis': {'title': {'text': 'Unit Price'}}}
 }))
```

In [25]:
```
question = """
    Get the total number of invoices for each customer
"""

vn.ask(question=question)
```

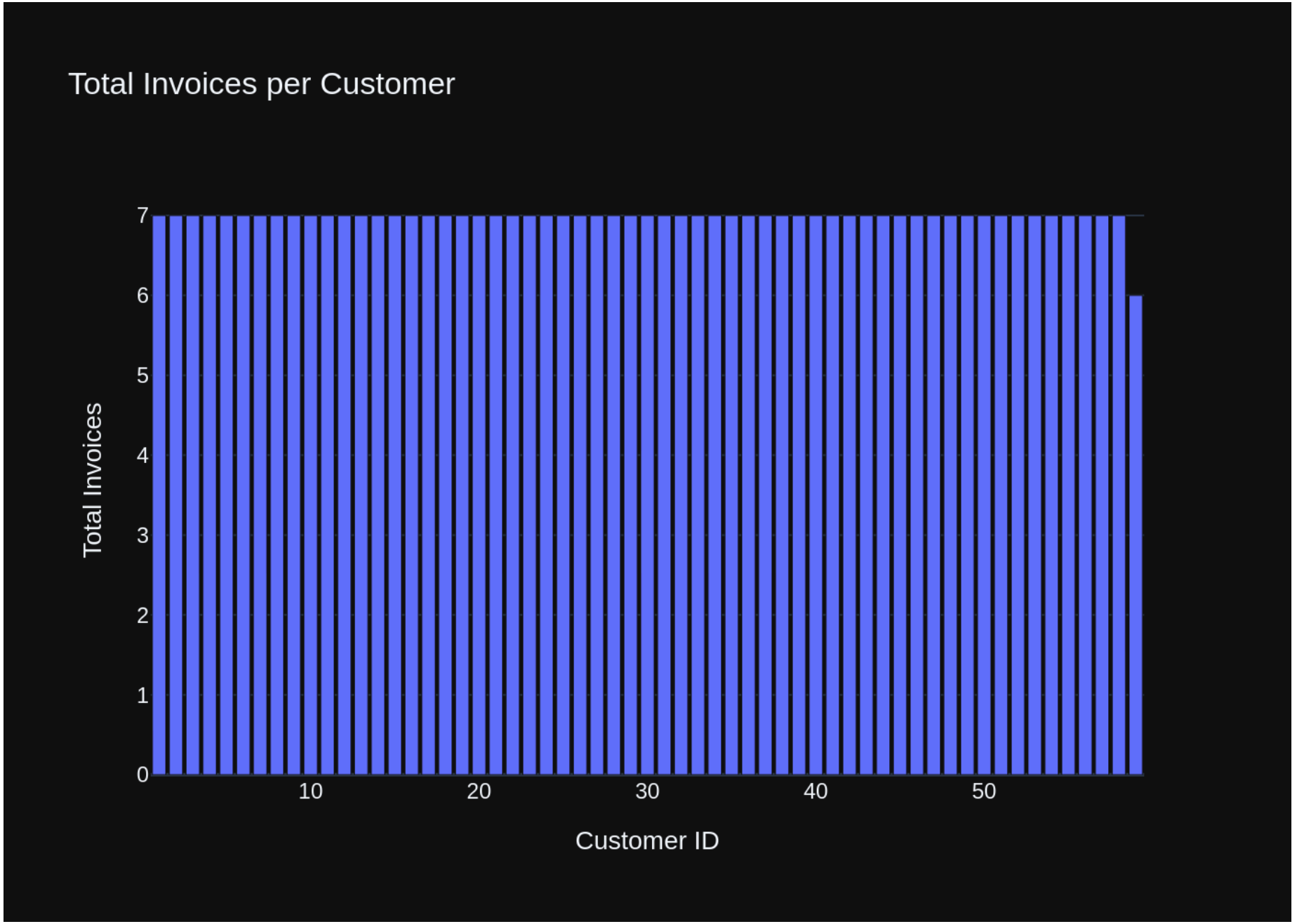
```
Number of requested results 10 is greater than number of elements in index 5, updating n_results = 5
Number of requested results 10 is greater than number of elements in index 1, updating n_results = 1
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r
\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoi
ces" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE TABLE "invoice_i
tems"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n
TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREI
GN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN
KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK
_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY
KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n
Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Count
ry NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NV
ARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (Employe
eId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (S
upportRepId)\n\n\n===Additional Context \n\n===Response Guidelines \n1. If the provided context is sufficient, pleas
e generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost suffi
cient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query
to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the pr
ovided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table
(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given befor
e. \n'}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assistant', 'conten
t': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assis
tant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': '  \n    List all albums and their
corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT albums.Title AS AlbumTitle, artists.Name
AS ArtistName \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 WHERE Name LIKE '%What%';"}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite datab
ase'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table';"}, {'role': 'user', 'con
tent': '  \n    Get the total number of invoices for each customer\n'}]
Using model gpt-4o-mini for 806.25 tokens (approx)
LLM Response: SELECT CustomerId, COUNT(*) AS TotalInvoices
FROM invoices
GROUP BY CustomerId;
Extracted SQL: SELECT CustomerId, COUNT(*) AS TotalInvoices
FROM invoices
GROUP BY CustomerId;
SELECT CustomerId, COUNT(*) AS TotalInvoices
```

```
FROM invoices
GROUP BY CustomerId;
```

	CustomerId	TotalInvoices
0	1	7
1	2	7
2	3	7
3	4	7
4	5	7
5	6	7
6	7	7
7	8	7
8	9	7
9	10	7
10	11	7
11	12	7
12	13	7
13	14	7
14	15	7
15	16	7
16	17	7
17	18	7
18	19	7
19	20	7
20	21	7
21	22	7
22	23	7
23	24	7
24	25	7
25	26	7
26	27	7
27	28	7
28	29	7
29	30	7
30	31	7
31	32	7
32	33	7
33	34	7
34	35	7
35	36	7
36	37	7
37	38	7

```
38          39              7
39          40              7
40          41              7
41          42              7
42          43              7
43          44              7
44          45              7
45          46              7
46          47              7
47          48              7
48          49              7
49          50              7
50          51              7
51          52              7
52          53              7
53          54              7
54          55              7
55          56              7
56          57              7
57          58              7
58          59              6
Using model gpt-4o-mini for 186.25 tokens (approx)
```


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

Out[25]: ('SELECT CustomerId, COUNT(*) AS TotalInvoices \nFROM invoices \nGROUP BY CustomerId;',

	CustomerId	TotalInvoices
0	1	7
1	2	7
2	3	7
3	4	7
4	5	7
5	6	7
6	7	7
7	8	7
8	9	7
9	10	7
10	11	7
11	12	7
12	13	7
13	14	7
14	15	7
15	16	7
16	17	7
17	18	7
18	19	7
19	20	7
20	21	7
21	22	7
22	23	7
23	24	7
24	25	7
25	26	7
26	27	7
27	28	7
28	29	7
29	30	7
30	31	7
31	32	7
32	33	7
33	34	7
34	35	7
35	36	7
36	37	7
37	38	7

```
38          39              7
39          40              7
40          41              7
41          42              7
42          43              7
43          44              7
44          45              7
45          46              7
46          47              7
47          48              7
48          49              7
49          50              7
50          51              7
51          52              7
52          53              7
53          54              7
54          55              7
55          56              7
56          57              7
57          58              7
58          59              6,
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'Customer ID=%{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([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
                          19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
                          37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
                          55, 56, 57, 58, 59]),
              'xaxis': 'x',
              'y': array([7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
                          7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
                          7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 6]),
              'yaxis': 'y'}],
```

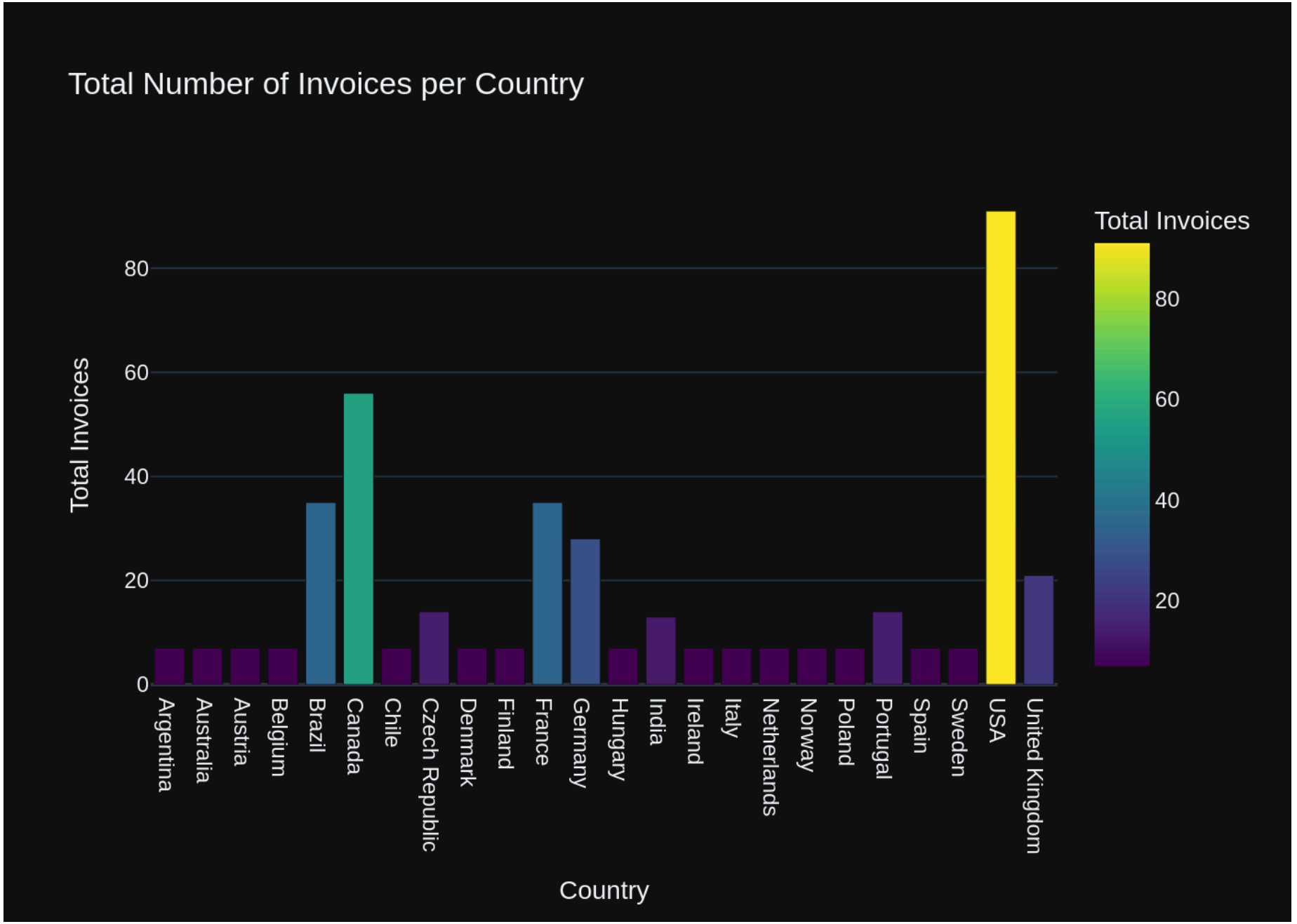
```
    'layout': {'barmode': 'relative',
               'legend': {'tracegroupgap': 0},
               'template': '...',
               'title': {'text': 'Total Invoices per Customer'},
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Customer ID'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total Invoices'}}}
})
```

In [26]:
```
question = """
    Find the total number of invoices per country:
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r
\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    Inv
oiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  N
OT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId)
REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFE
RENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId
ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE INDEX I
FK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE "employees"\r\n(\r\n    EmployeeId INTEGER PRIMAR
Y KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n
Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NV
ARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVAR
CHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (Reports
To) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_Emp
loyeeReportsTo ON "employees" (ReportsTo)\n\n\n===Additional Context \n\n===Response Guidelines \n1. If the provided
context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the prov
ided context is almost sufficient but requires knowledge of a specific string in a particular column, please generat
e an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying int
ermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please u
se the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exa
ctly as it was given before. \n'}, {'role': 'user', 'content': '  \n    Get the total number of invoices for each cu
stomer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices \nFROM invoices \nGROUP B
Y CustomerId;'}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assistant',
'content': 'SELECT COUNT(*) FROM customers;'}, {'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'}, {'role': 'assistant', 'content': 'SELECT albums.Title AS AlbumTitle, artist
s.Name AS ArtistName \nFROM albums \nJOIN artists ON albums.ArtistId = artists.ArtistId;'}, {'role': 'user', 'conten
t': '  \n    Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content':
"SELECT * FROM tracks WHERE Name LIKE '%What%';"}, {'role': 'user', 'content': 'Show me a list of tables in the SQLi
te database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table';"}, {'role': 'use
r', 'content': '  \n    Find the total number of invoices per country:\n'}]
Using model gpt-4o-mini for 846.25 tokens (approx)
LLM Response: SELECT BillingCountry, COUNT(*) AS TotalInvoices
FROM invoices
GROUP BY BillingCountry;
Extracted SQL: SELECT BillingCountry, COUNT(*) AS TotalInvoices
FROM invoices
```

```
GROUP BY BillingCountry;
SELECT BillingCountry, COUNT(*) AS TotalInvoices
FROM invoices
GROUP BY BillingCountry;
    BillingCountry  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
Using model gpt-4o-mini for 188.25 tokens (approx)
```


Total Number of Invoices per Country
Total Invoices
80
60
40
20
0
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
Total Invoices
80
60
40
20

```
Out[26]: ('SELECT BillingCountry, COUNT(*) AS TotalInvoices \nFROM invoices \nGROUP BY BillingCountry;',
    BillingCountry  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=%{marker.color}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': 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]),
                         'coloraxis': 'coloraxis',
                         'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'v',
              'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
```

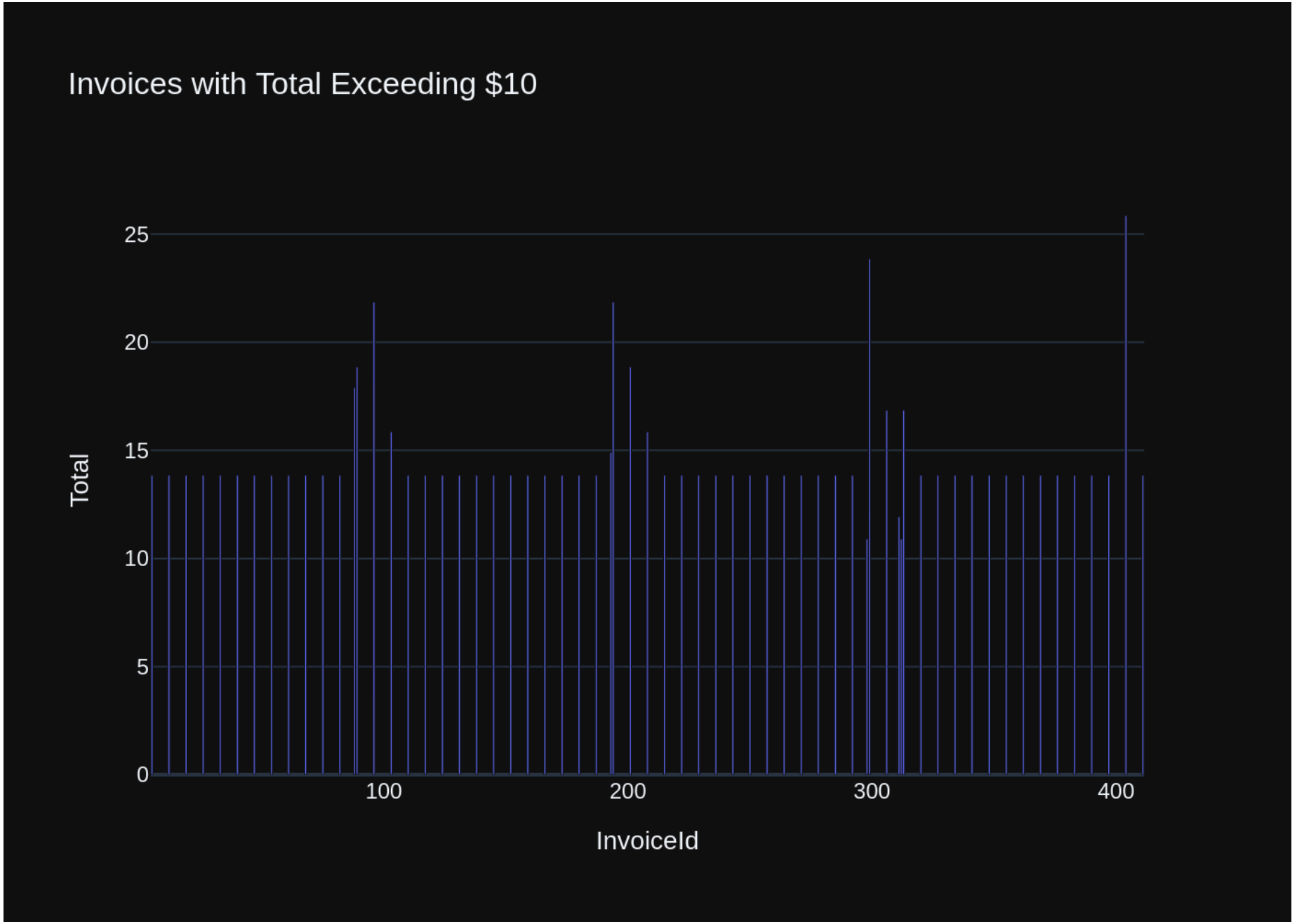
```
                    'x': array(['Argentina', 'Australia', 'Austria', 'Belgium', 'Brazil', 'Canada',
                                'Chile', 'Czech Republic', 'Denmark', 'Finland', 'France', 'Germany',
                                'Hungary', 'India', 'Ireland', 'Italy', 'Netherlands', 'Norway',
                                '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',
                    'coloraxis': {'colorbar': {'title': {'text': 'Total Invoices'}},
                                  'colorscale': [[0.0, '#440154'], [0.1111111111111111,
                                                 '#482878'], [0.2222222222222222,
                                                 '#3e4989'], [0.3333333333333333,
                                                 '#31688e'], [0.4444444444444444,
                                                 '#26828e'], [0.5555555555555556,
                                                 '#1f9e89'], [0.6666666666666666,
                                                 '#35b779'], [0.7777777777777778,
                                                 '#6ece58'], [0.8888888888888888,
                                                 '#b5de2b'], [1.0, '#fde725']]},
                    'legend': {'tracegroupgap': 0},
                    'template': '...',
                    'title': {'text': 'Total Number of Invoices per Country'},
                    'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Country'}},
                    'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total Invoices'}}}
          }))
```

In [27]:
```
question = """
    List all invoices with a total exceeding $10:
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT
NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT
NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE N
O ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE
TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NU
LL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n
BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total N
UMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACT
ION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE INDEX IFK
_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOI
NCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NUL
L,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGE
R,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON D
ELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO
ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELET
E NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\n===Response Guidelines \n1. If the provided cont
ext is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided
context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an
intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermed
iate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use th
e most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly
as it was given before. \n'}, {'role': 'user', 'content': '  \n    Get the total number of invoices for each custome
r\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices \nFROM invoices \nGROUP BY Cus
tomerId;'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per country:\n'}, {'role': 'assis
tant', 'content': 'SELECT BillingCountry, COUNT(*) AS TotalInvoices \nFROM invoices \nGROUP BY BillingCountry;'},
{'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assistant', 'content': 'SELE
CT COUNT(*) FROM customers;'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'c
ontent': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': '  \n    List all albums and their correspo
nding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT albums.Title AS AlbumTitle, artists.Name AS Artis
tName \nFROM albums \nJOIN artists ON albums.ArtistId = artists.ArtistId;'}, {'role': 'user', 'content': '  \n    Fi
nd all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM
tracks WHERE Name LIKE '%What%';"}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'},
{'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='table';"}, {'role': 'user', 'content':
'  \n    List all invoices with a total exceeding $10:\n'}]
Using model gpt-4o-mini for 881.75 tokens (approx)
LLM Response: SELECT * FROM invoices WHERE Total > 10;
Extracted SQL: SELECT * FROM invoices WHERE Total > 10;
SELECT * FROM invoices WHERE Total > 10;
```

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

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

[64 rows x 9 columns]
Using model gpt-4o-mini for 228.5 tokens (approx)
```


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

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

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

[64 rows x 9 columns],
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'InvoiceId=%{x}<br>Total=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'v',
              'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
              'x': array([  5,  12,  19,  26,  33,  40,  47,  54,  61,  68,  75,  82,  88,  89,
```

```
                            96, 103, 110, 117, 124, 131, 138, 145, 152, 159, 166, 173, 180, 187,
                           193, 194, 201, 208, 215, 222, 229, 236, 243, 250, 257, 264, 271, 278,
                           285, 292, 298, 299, 306, 311, 312, 313, 320, 327, 334, 341, 348, 355,
                           362, 369, 376, 383, 390, 397, 404, 411]),
                'xaxis': 'x',
                'y': array([13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                           13.86, 13.86, 17.91, 18.86, 21.86, 15.86, 13.86, 13.86, 13.86, 13.86,
                           13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 14.91, 21.86,
                           18.86, 15.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                           13.86, 13.86, 13.86, 13.86, 10.91, 23.86, 16.86, 11.94, 10.91, 16.86,
                           13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                           13.86, 13.86, 25.86, 13.86]),
                'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Invoices with Total Exceeding $10'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total'}}}
 }))
```

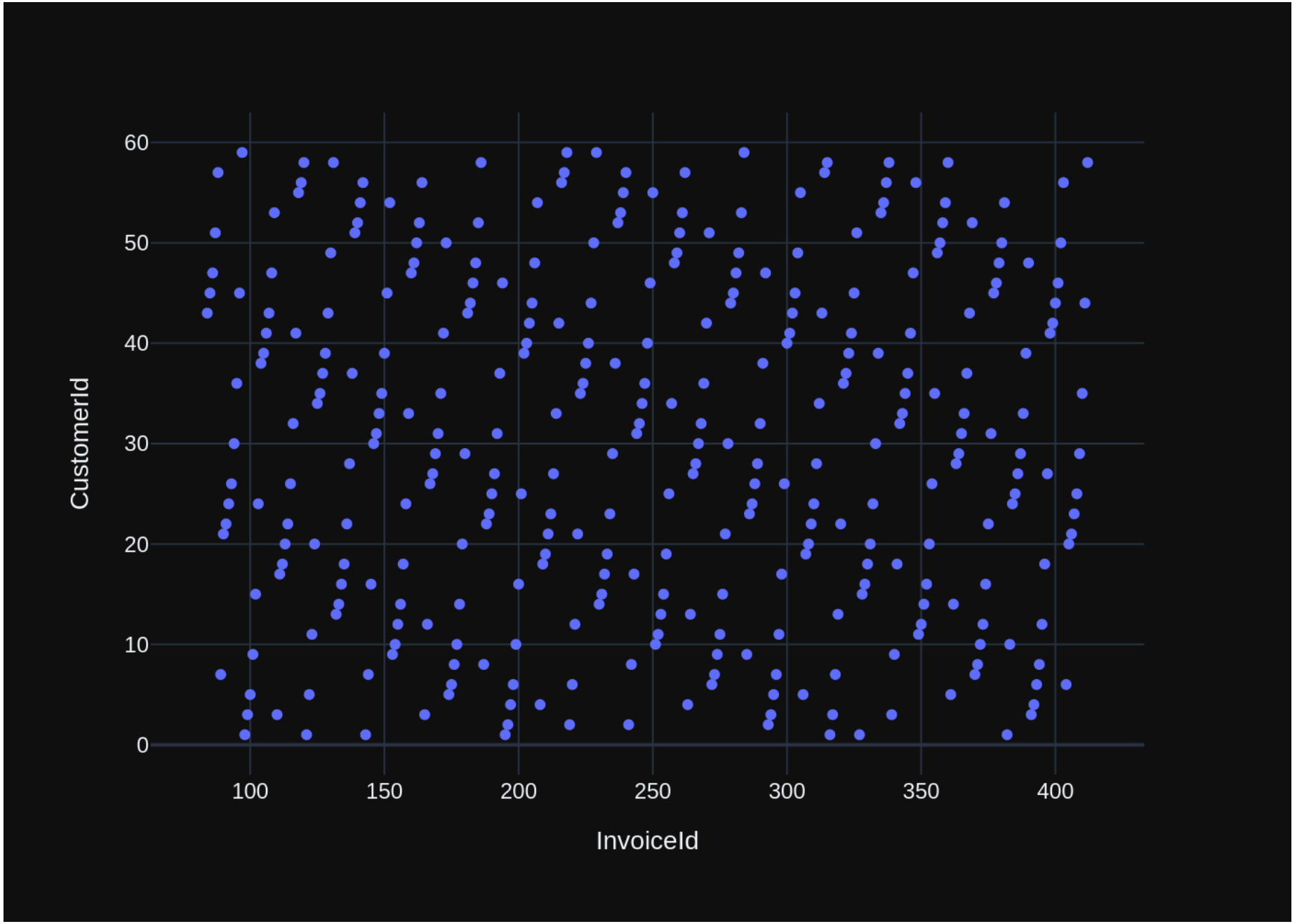
In [28]:
```
question = """
    Find all invoices since 2010 and the total amount invoiced:
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r
\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    Inv
oiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  N
OT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId)
REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFE
RENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoic
eId ON "invoice_items" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX
IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE "employees"\r\n(\r\n    EmployeeId INTEGER PRIMA
RY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n
Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NV
ARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVAR
CHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (Reports
To) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Cont
ext \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is suffic
ient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is a
lmost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate
SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3.
If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevan
t table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was give
n before. \n'}, {'role': 'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'as
sistant', 'content': 'SELECT * FROM invoices WHERE Total > 10;'}, {'role': 'user', 'content': '  \n    Find the tota
l number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(*) AS TotalInv
oices \nFROM invoices \nGROUP BY BillingCountry;'}, {'role': 'user', 'content': '  \n    Get the total number of inv
oices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices \nFROM i
nvoices \nGROUP BY CustomerId;'}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'ro
le': 'assistant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': 'How many customers are
there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': '  \n    L
ist all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT albums.Title AS
AlbumTitle, artists.Name AS ArtistName \nFROM albums \nJOIN artists ON albums.ArtistId = artists.ArtistId;'}, {'rol
e': 'user', 'content': '  \n    Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assi
stant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%';"}, {'role': 'user', 'content': 'Show me a list of
tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type='tabl
e';"}, {'role': 'user', 'content': '  \n    Find all invoices since 2010 and the total amount invoiced:\n'}]
Using model gpt-4o-mini for 904.0 tokens (approx)
LLM Response: SELECT *, Total
FROM invoices
```

```
WHERE InvoiceDate >= '2010-01-01';
Extracted SQL: SELECT *, Total
FROM invoices
WHERE InvoiceDate >= '2010-01-01';
SELECT *, Total
FROM invoices
WHERE InvoiceDate >= '2010-01-01';
     InvoiceId  CustomerId          InvoiceDate  \
0           84          43  2010-01-08 00:00:00
1           85          45  2010-01-08 00:00:00
2           86          47  2010-01-09 00:00:00
3           87          51  2010-01-10 00:00:00
4           88          57  2010-01-13 00:00:00
..         ...         ...                  ...
324        408          25  2013-12-05 00:00:00
325        409          29  2013-12-06 00:00:00
326        410          35  2013-12-09 00:00:00
327        411          44  2013-12-14 00:00:00
328        412          58  2013-12-22 00:00:00

                             BillingAddress BillingCity BillingState  \
0                          68, Rue Jouvence       Dijon         None
1                         Erzsébet krt. 58.    Budapest         None
2                      Via Degli Scipioni, 43        Rome           RM
3                                Celsiusg. 9   Stockholm         None
4                            Calle Lira, 198    Santiago         None
..                                       ...         ...          ...
324                     319 N. Frances Street     Madison           WI
325                    796 Dundas Street West     Toronto           ON
326  Rua dos Campeões Europeus de Viena, 4350       Porto         None
327                           Porthaninkatu 9    Helsinki         None
328                      12,Community Centre       Delhi         None

    BillingCountry BillingPostalCode  Total  Total
0           France             21000   1.98   1.98
1          Hungary            H-1073   1.98   1.98
2            Italy             00192   3.96   3.96
3           Sweden             11230   6.94   6.94
4            Chile              None  17.91  17.91
..             ...               ...    ...    ...
324            USA             53703   3.96   3.96
```

```
325         Canada          M6J 1V1   5.94   5.94
326       Portugal             None   8.91   8.91
327        Finland            00530  13.86  13.86
328          India           110017   1.99   1.99

[329 rows x 10 columns]
Using model gpt-4o-mini for 245.75 tokens (approx)
```


60
50
40
30
20
10
0
CustomerId
100
150
200
250
300
350
400
InvoiceId

Out[28]: ("SELECT *, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01';",

	InvoiceId	CustomerId	InvoiceDate	BillingAddress	BillingCity	BillingState	BillingCountry	BillingPostalCode	Total	Total
0	84	43	2010-01-08 00:00:00	68, Rue Jouvence	Dijon	None	France	21000	1.98	1.98
1	85	45	2010-01-08 00:00:00	Erzsébet krt. 58.	Budapest	None	Hungary	H-1073	1.98	1.98
2	86	47	2010-01-09 00:00:00	Via Degli Scipioni, 43	Rome	RM	Italy	00192	3.96	3.96
3	87	51	2010-01-10 00:00:00	Celsiusg. 9	Stockholm	None	Sweden	11230	6.94	6.94
4	88	57	2010-01-13 00:00:00	Calle Lira, 198	Santiago	None	Chile	None	17.91	17.91
..	...	...	...	...	...	...	...	...	...	...
324	408	25	2013-12-05 00:00:00	319 N. Frances Street	Madison	WI	USA	53703	3.96	3.96
325	409	29	2013-12-06 00:00:00	796 Dundas Street West	Toronto	ON	Canada	M6J 1V1	5.94	5.94
326	410	35	2013-12-09 00:00:00	Rua dos Campeões Europeus de Viena, 4350	Porto	None	Portugal	None	8.91	8.91
327	411	44	2013-12-14 00:00:00	Porthaninkatu 9	Helsinki	None	Finland	00530	13.86	13.86
328	412	58	2013-12-22 00:00:00	12,Community Centre	Delhi	None	India	110017	1.99	1.99

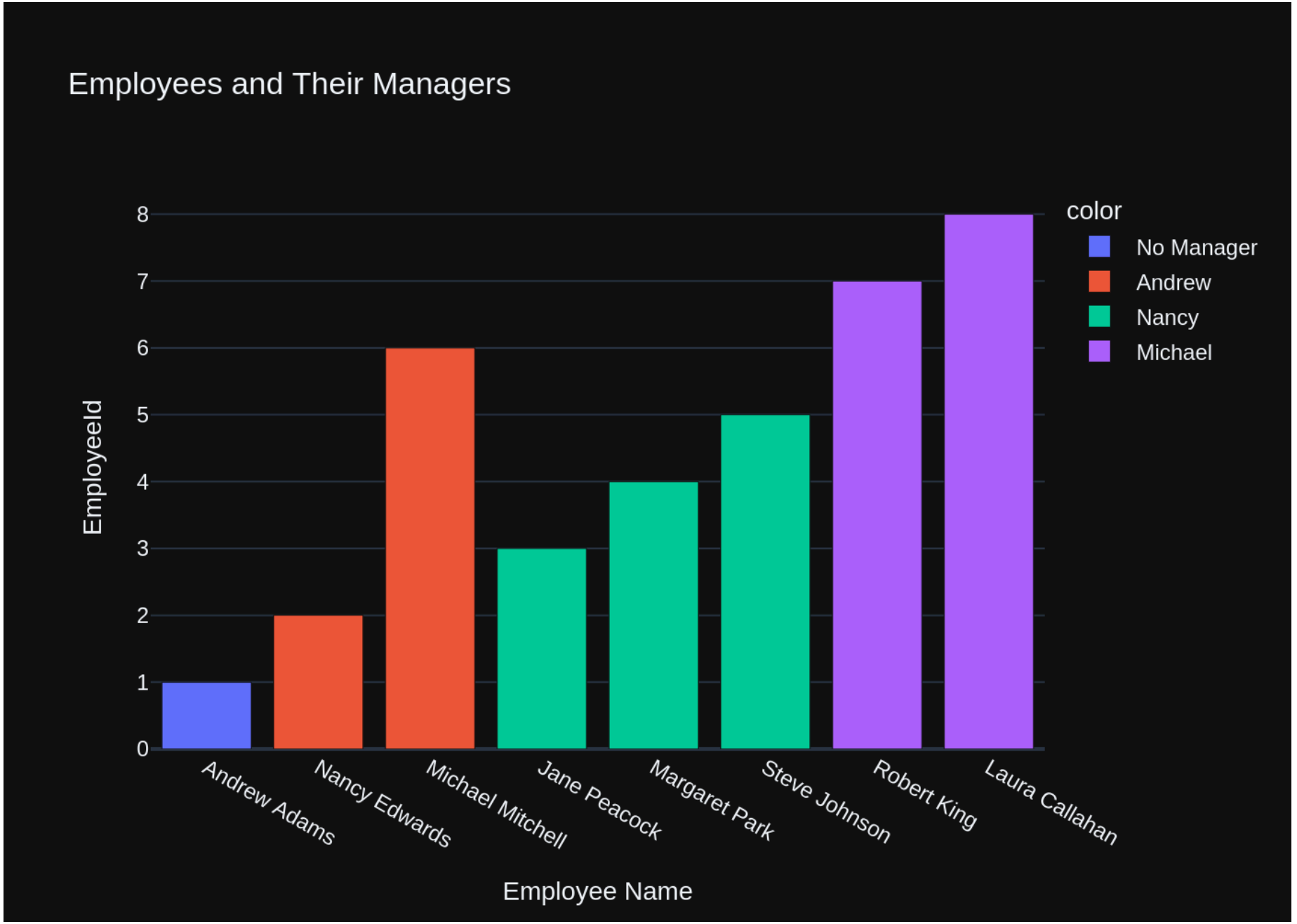
```
[329 rows x 10 columns],
Figure({
    'data': [{'hovertemplate': 'InvoiceId=%{x}<br>CustomerId=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'symbol': 'circle'},
              'mode': 'markers',
              'name': '',
              'orientation': 'v',
              'showlegend': False,
              'type': 'scatter',
              'x': array([ 84,  85,  86, ..., 410, 411, 412]),
              'xaxis': 'x',
              'y': array([43, 45, 47, ..., 35, 44, 58]),
              'yaxis': 'y'}],
    'layout': {'legend': {'tracegroupgap': 0},
               'margin': {'t': 60},
               'template': '...',
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceId'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}}}
})
```

In [29]:
```
question = """
    List all employees and their reporting manager's name (if any):
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE TABLE "employee
s"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n
FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r
\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Co
untry NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email
NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDA
TE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Addres
s NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode
NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    Suppor
tRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON
UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "invoic
es"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    In
voiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingSta
te NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,
2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPD
ATE NO ACTION\r\n)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\n\n===Additional Context \n\n===Response Guidelines
\n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the quest
ion. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular c
olumn, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with
a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be gene
rated. \n4. 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 all invoices since
2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT *, Total \nFROM invoices \nWHERE I
nvoiceDate >= '2010-01-01';"}, {'role': 'user', 'content': '  \n    Get the total number of invoices for each custom
er\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices \nFROM invoices \nGROUP BY Cu
stomerId;'}, {'role': 'user', 'content': '  \n    List all albums and their corresponding artist names  \n'}, {'rol
e': 'assistant', 'content': 'SELECT albums.Title AS AlbumTitle, artists.Name AS ArtistName \nFROM albums \nJOIN arti
sts ON albums.ArtistId = artists.ArtistId;'}, {'role': 'user', 'content': '  \n    Find the total number of invoices
per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(*) AS TotalInvoices \nFROM invoices
\nGROUP BY BillingCountry;'}, {'role': 'user', 'content': '  \n    List all invoices with a total exceeding $1
0:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10;'}, {'role': 'user', 'content': 'H
ow many records are in table called customer'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers;'},
{'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM
customers;'}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant',
'content': "SELECT name FROM sqlite_master WHERE type='table';"}, {'role': 'user', 'content': '  \n    Find all trac
ks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHER
E Name LIKE '%What%';"}, {'role': 'user', 'content': "  \n    List all employees and their reporting manager's name
(if any):\n"}]
```

```
Using model gpt-4o-mini for 946.5 tokens (approx)
LLM Response: SELECT e.EmployeeId, e.FirstName AS EmployeeFirstName, e.LastName AS EmployeeLastName,
       m.FirstName AS ManagerFirstName, m.LastName AS ManagerLastName
FROM employees e
LEFT JOIN employees m ON e.ReportsTo = m.EmployeeId;
Extracted SQL: SELECT e.EmployeeId, e.FirstName AS EmployeeFirstName, e.LastName AS EmployeeLastName,
       m.FirstName AS ManagerFirstName, m.LastName AS ManagerLastName
FROM employees e
LEFT JOIN employees m ON e.ReportsTo = m.EmployeeId;
SELECT e.EmployeeId, e.FirstName AS EmployeeFirstName, e.LastName AS EmployeeLastName,
       m.FirstName AS ManagerFirstName, m.LastName AS ManagerLastName
FROM employees e
LEFT JOIN employees m ON e.ReportsTo = m.EmployeeId;
   EmployeeId EmployeeFirstName EmployeeLastName ManagerFirstName  \
0           1            Andrew            Adams             None
1           2             Nancy          Edwards           Andrew
2           3              Jane          Peacock            Nancy
3           4          Margaret             Park            Nancy
4           5             Steve          Johnson            Nancy
5           6           Michael         Mitchell           Andrew
6           7            Robert             King          Michael
7           8             Laura         Callahan          Michael

  ManagerLastName
0            None
1           Adams
2         Edwards
3         Edwards
4         Edwards
5           Adams
6        Mitchell
7        Mitchell
Using model gpt-4o-mini for 249.5 tokens (approx)
```


Employees and Their Managers
EmployeeId
0
1
2
3
4
5
6
7
8
Andrew Adams
Nancy Edwards
Michael Mitchell
Jane Peacock
Margaret Park
Steve Johnson
Robert King
Laura Callahan
Employee Name
color
No Manager
Andrew
Nancy
Michael

Out[29]:
```
('SELECT e.EmployeeId, e.FirstName AS EmployeeFirstName, e.LastName AS EmployeeLastName, \n       m.FirstName AS M
anagerFirstName, m.LastName AS ManagerLastName\nFROM employees e\nLEFT JOIN employees m ON e.ReportsTo = m.Employe
eId;',
    EmployeeId EmployeeFirstName EmployeeLastName ManagerFirstName  \
 0           1            Andrew           Adams             None
 1           2             Nancy         Edwards           Andrew
 2           3              Jane         Peacock            Nancy
 3           4          Margaret            Park            Nancy
 4           5             Steve         Johnson            Nancy
 5           6           Michael        Mitchell           Andrew
 6           7            Robert            King          Michael
 7           8             Laura        Callahan          Michael

   ManagerLastName
 0            None
 1           Adams
 2         Edwards
 3         Edwards
 4         Edwards
 5           Adams
 6        Mitchell
 7        Mitchell  ,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'color=No Manager<br>Employee Name=%{x}<br>EmployeeId=%{y}<extra></extra>',
               'legendgroup': 'No Manager',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': 'No Manager',
               'offsetgroup': 'No Manager',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Andrew Adams'], dtype=object),
               'xaxis': 'x',
               'y': array([1]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'color=Andrew<br>Employee Name=%{x}<br>EmployeeId=%{y}<extra></extra>',
               'legendgroup': 'Andrew',
```

```
                  'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
                  'name': 'Andrew',
                  'offsetgroup': 'Andrew',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Nancy Edwards', 'Michael Mitchell'], dtype=object),
                  'xaxis': 'x',
                  'y': array([2, 6]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'color=Nancy<br>Employee Name=%{x}<br>EmployeeId=%{y}<extra></extra>',
                  'legendgroup': 'Nancy',
                  'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
                  'name': 'Nancy',
                  'offsetgroup': 'Nancy',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Jane Peacock', 'Margaret Park', 'Steve Johnson'], dtype=object),
                  'xaxis': 'x',
                  'y': array([3, 4, 5]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'color=Michael<br>Employee Name=%{x}<br>EmployeeId=%{y}<extra></extra>',
                  'legendgroup': 'Michael',
                  'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
                  'name': 'Michael',
                  'offsetgroup': 'Michael',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Robert King', 'Laura Callahan'], dtype=object),
                  'xaxis': 'x',
                  'y': array([7, 8]),
                  'yaxis': 'y'}],
      'layout': {'barmode': 'relative',
                 'legend': {'title': {'text': 'color'}, 'tracegroupgap': 0},
```

```
                    'template': '...',
                    'title': {'text': 'Employees and Their Managers'},
                    'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Employee Name'}},
                    'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'EmployeeId'}}}
  }))
```

In [30]:

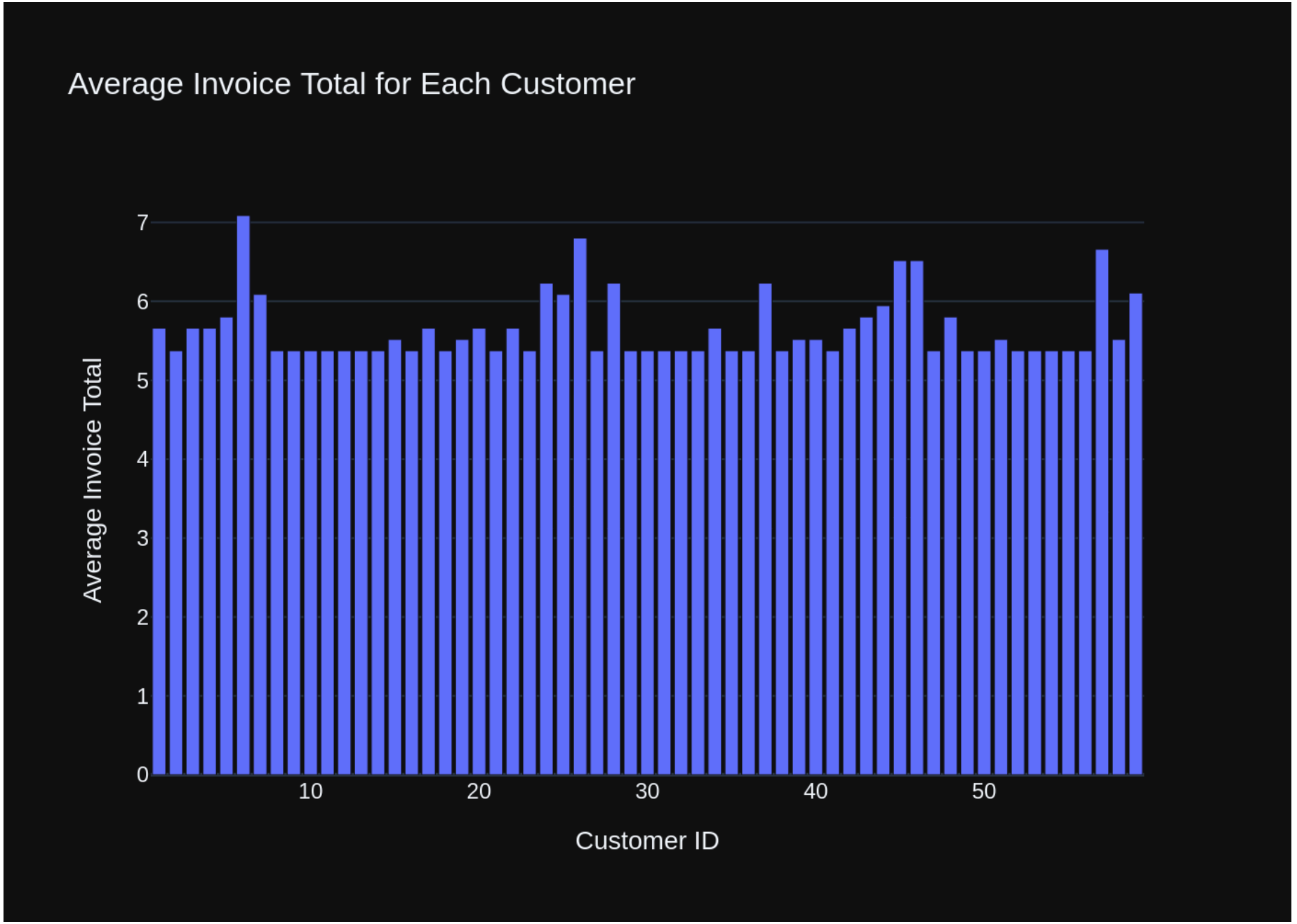
```
question = """
    Get the average invoice total for each customer:
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r
\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoi
ces" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE TABLE "invoice_i
tems"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n
TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREI
GN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN
KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK
_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE INDEX IFK_Cust
omerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KE
Y AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Co
mpany NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country
NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARC
HAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\n===Response Guidelines \n1. I
f the provided context is sufficient, please generate a valid SQL query without any explanations for the question.
\n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular colum
n, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a c
omment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generat
ed. \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 number of inv
oices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices \nFROM i
nvoices \nGROUP BY CustomerId;'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per countr
y:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(*) AS TotalInvoices \nFROM invoices \nGROUP B
Y BillingCountry;'}, {'role': 'user', 'content': '  \n    Find all invoices since 2010 and the total amount invoice
d:\n'}, {'role': 'assistant', 'content': "SELECT *, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01';"},
{'role': 'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'conte
nt': 'SELECT * FROM invoices WHERE Total > 10;'}, {'role': 'user', 'content': 'How many customers are there'}, {'rol
e': 'assistant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': 'How many records are in
table called customer'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'cont
ent': "  \n    List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content':
'SELECT e.EmployeeId, e.FirstName AS EmployeeFirstName, e.LastName AS EmployeeLastName, \n       m.FirstName AS Mana
gerFirstName, m.LastName AS ManagerLastName\nFROM employees e\nLEFT JOIN employees m ON e.ReportsTo = m.EmployeeI
d;'}, {'role': 'user', 'content': '  \n    List all albums and their corresponding artist names  \n'}, {'role': 'ass
istant', 'content': 'SELECT albums.Title AS AlbumTitle, artists.Name AS ArtistName \nFROM albums \nJOIN artists ON a
lbums.ArtistId = artists.ArtistId;'}, {'role': 'user', 'content': '  \n    Find all tracks with a name containing "W
hat" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%';"}, {'r
```

```
ole': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant', 'content': "SELEC
T name FROM sqlite_master WHERE type='table';"}, {'role': 'user', 'content': '  \n    Get the average invoice total
for each customer:\n'}]
Using model gpt-4o-mini for 1017.5 tokens (approx)
LLM Response: SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal
FROM invoices
GROUP BY CustomerId;
Extracted SQL: SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal
FROM invoices
GROUP BY CustomerId;
SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal
FROM invoices
GROUP BY CustomerId;
    CustomerId  AverageInvoiceTotal
0            1             5.660000
1            2             5.374286
2            3             5.660000
3            4             5.660000
4            5             5.802857
5            6             7.088571
6            7             6.088571
7            8             5.374286
8            9             5.374286
9           10             5.374286
10          11             5.374286
11          12             5.374286
12          13             5.374286
13          14             5.374286
14          15             5.517143
15          16             5.374286
16          17             5.660000
17          18             5.374286
18          19             5.517143
19          20             5.660000
20          21             5.374286
21          22             5.660000
22          23             5.374286
23          24             6.231429
24          25             6.088571
25          26             6.802857
26          27             5.374286
```

```
27          28          6.231429
28          29          5.374286
29          30          5.374286
30          31          5.374286
31          32          5.374286
32          33          5.374286
33          34          5.660000
34          35          5.374286
35          36          5.374286
36          37          6.231429
37          38          5.374286
38          39          5.517143
39          40          5.517143
40          41          5.374286
41          42          5.660000
42          43          5.802857
43          44          5.945714
44          45          6.517143
45          46          6.517143
46          47          5.374286
47          48          5.802857
48          49          5.374286
49          50          5.374286
50          51          5.517143
51          52          5.374286
52          53          5.374286
53          54          5.374286
54          55          5.374286
55          56          5.374286
56          57          6.660000
57          58          5.517143
58          59          6.106667
Using model gpt-4o-mini for 191.75 tokens (approx)
```


Average Invoice Total for Each Customer
Average Invoice Total
0
1
2
3
4
5
6
7
10
20
30
40
50
Customer ID

Out[30]: ('SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM invoices \nGROUP BY CustomerId;',

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

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

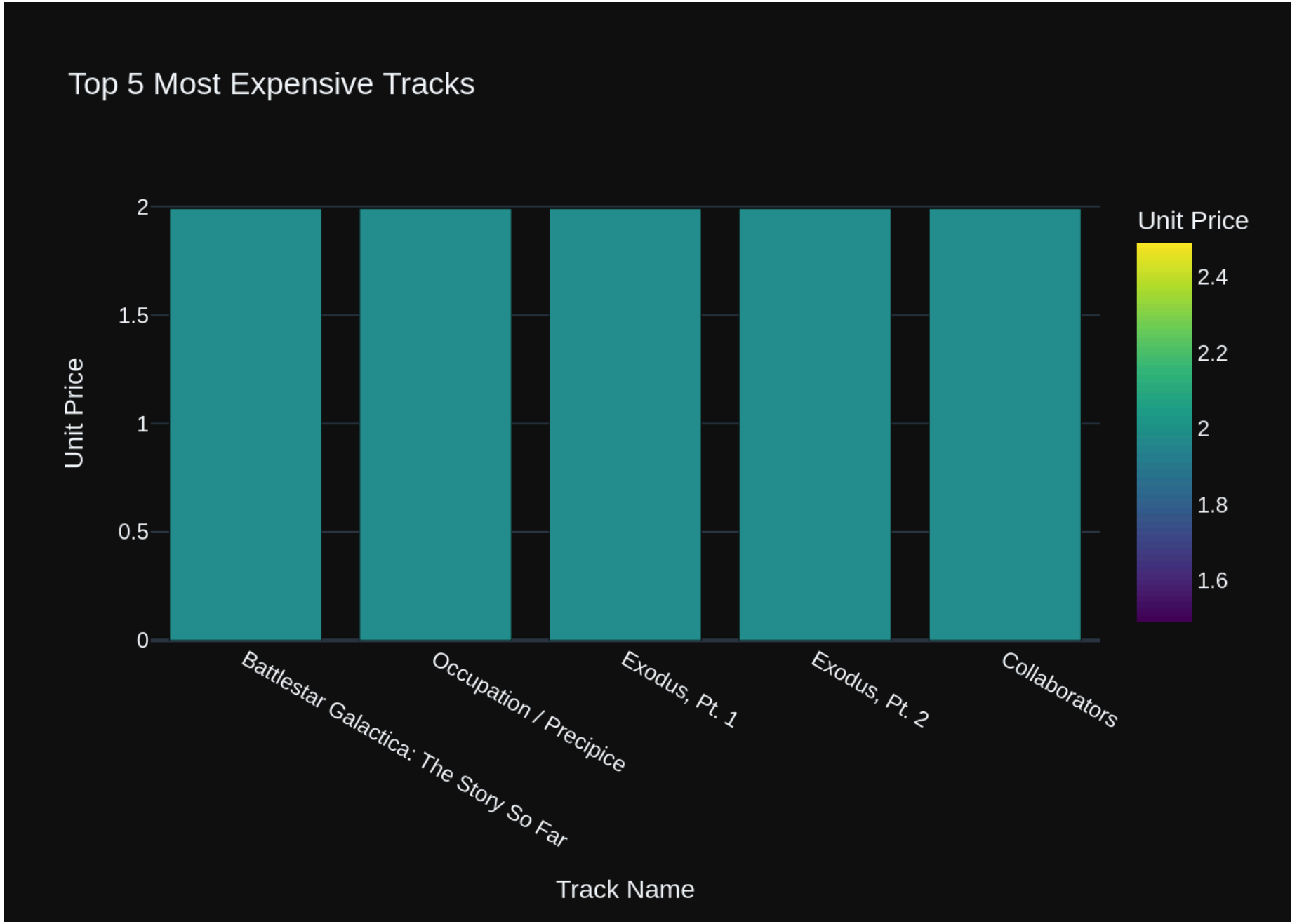
```
                            6.08857143, 6.80285714, 5.37428571, 6.23142857, 5.37428571, 5.37428571,
                            5.37428571, 5.37428571, 5.37428571, 5.66      , 5.37428571, 5.37428571,
                            6.23142857, 5.37428571, 5.51714286, 5.51714286, 5.37428571, 5.66      ,
                            5.80285714, 5.94571429, 6.51714286, 6.51714286, 5.37428571, 5.80285714,
                            5.37428571, 5.37428571, 5.51714286, 5.37428571, 5.37428571, 5.37428571,
                            5.37428571, 5.37428571, 6.66      , 5.51714286, 6.10666667]),
                'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Average Invoice Total for Each Customer'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Customer ID'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Average Invoice Total'}}}
 }))
```

In [31]:
```
question = """
    Find the top 5 most expensive tracks (based on unit price):
"""

vn.ask(question=question)
```

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

SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGE
R,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUME
RIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION\r\n)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (G
enreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE INDEX IFK_InvoiceLineTrackI
d ON "invoice_items" (TrackId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE TABLE "invoi
ce_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NUL
L,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r
\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n
FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE T
ABLE "playlist_track"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT
PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Playlis
tId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \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 say
ing intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. P
lease use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the ans
wer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containin
g "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%';"},
{'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'conte
nt': 'SELECT * FROM invoices WHERE Total > 10;'}, {'role': 'user', 'content': ' \n   List all albums and their cor
responding artist names \n'}, {'role': 'assistant', 'content': 'SELECT albums.Title AS AlbumTitle, artists.Name AS
ArtistName \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 *, Total
\nFROM invoices \nWHERE InvoiceDate >= '2010-01-01';"}, {'role': 'user', 'content': ' \n   Get the average invoice
total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal
\nFROM invoices \nGROUP BY CustomerId;'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per
country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(*) AS TotalInvoices \nFROM invoices \nG
ROUP BY BillingCountry;'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each customer
\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices \nFROM invoices \nGROUP BY Cust
omerId;'}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant', 'co
ntent': "SELECT name FROM sqlite_master WHERE type='table';"}, {'role': 'user', 'content': 'How many customers are t
here'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': 'How many r
ecords are in table called customer'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role':

```
'user', 'content': '  \n    Find the top 5 most expensive tracks (based on unit price):\n'}]
Using model gpt-4o-mini for 981.0 tokens (approx)
LLM Response: SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5;
Extracted SQL: SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5;
SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5;
   TrackId                                  Name  AlbumId  MediaTypeId  \
0     2819  Battlestar Galactica: The Story So Far      226            3
1     2820                 Occupation / Precipice      227            3
2     2821                         Exodus, Pt. 1      227            3
3     2822                         Exodus, Pt. 2      227            3
4     2823                         Collaborators      227            3

   GenreId Composer  Milliseconds       Bytes  UnitPrice
0       18     None       2622250   490750393       1.99
1       19     None       5286953  1054423946       1.99
2       19     None       2621708   475079441       1.99
3       19     None       2618000   466820021       1.99
4       19     None       2626626   483484911       1.99
Using model gpt-4o-mini for 224.0 tokens (approx)
```


Top 5 Most Expensive Tracks
Unit Price
2
1.5
1
0.5
0
Battlestar Galactica: The Story So Far
Occupation / Precipice
Exodus, Pt. 1
Exodus, Pt. 2
Collaborators
Track Name
Unit Price
2.4
2.2
2
1.8
1.6

```
Out[31]: ('SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5;',
             TrackId                                  Name  AlbumId  MediaTypeId  \
          0     2819  Battlestar Galactica: The Story So Far      226            3
          1     2820                 Occupation / Precipice      227            3
          2     2821                         Exodus, Pt. 1      227            3
          3     2822                         Exodus, Pt. 2      227            3
          4     2823                         Collaborators      227            3

             GenreId Composer  Milliseconds       Bytes  UnitPrice
          0       18     None       2622250   490750393       1.99
          1       19     None       5286953  1054423946       1.99
          2       19     None       2621708   475079441       1.99
          3       19     None       2618000   466820021       1.99
          4       19     None       2626626   483484911       1.99  ,
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'Track Name=%{x}<br>Unit Price=%{marker.color}<extra></extra>',
                        'legendgroup': '',
                        'marker': {'color': array([1.99, 1.99, 1.99, 1.99, 1.99]),
                                   'coloraxis': 'coloraxis',
                                   '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',
                         'coloraxis': {'colorbar': {'title': {'text': 'Unit Price'}},
                                       'colorscale': [[0.0, '#440154'], [0.1111111111111111,
                                                      '#482878'], [0.2222222222222222,
                                                      '#3e4989'], [0.3333333333333333,
                                                      '#31688e'], [0.4444444444444444,
                                                      '#26828e'], [0.5555555555555556,
                                                      '#1f9e89'], [0.6666666666666666,
```

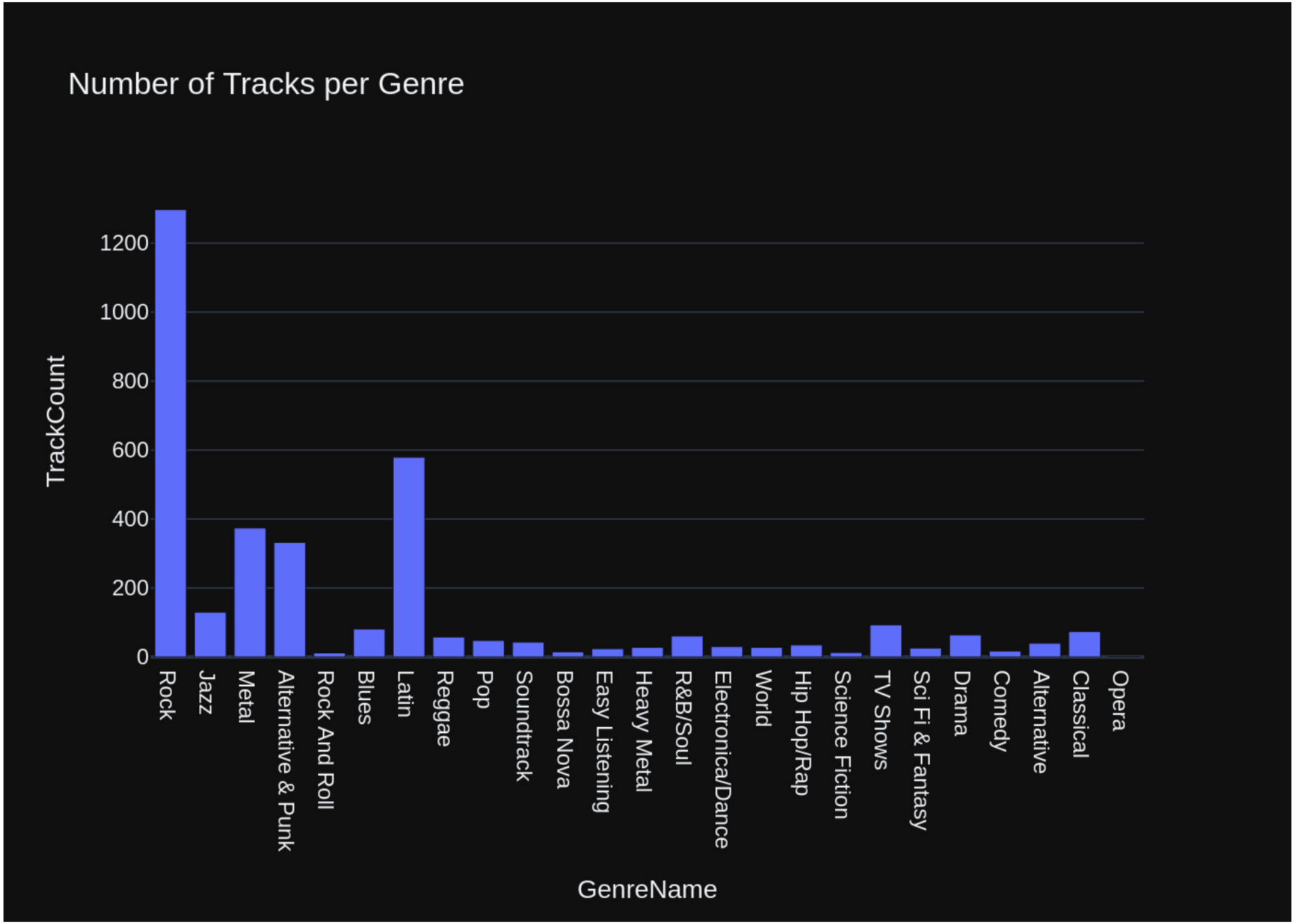
```
                                                '#35b779'], [0.7777777777777778,
                                                '#6ece58'], [0.8888888888888888,
                                                '#b5de2b'], [1.0, '#fde725']]},
                    'legend': {'tracegroupgap': 0},
                    'template': '...',
                    'title': {'text': 'Top 5 Most Expensive Tracks'},
                    'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Track Name'}},
                    'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Unit Price'}}}
          }))
```

In [32]:
```
question = """
    List all genres and the number of tracks in each genre:
"""

vn.ask(question=question)
```

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

SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGE
R,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUME
RIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE TABLE "genres"\r\n(\r\n   GenreId IN
TEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON
"playlist_track" (TrackId)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "playlists"\r\n(\r
\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_Trac
kMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE TABLE "playlist_track"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r
\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREI
GN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FORE
IGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE
"albums"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n
ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTIO
N ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\n===Response Guidelines \n1. If the provided context is suf
ficient, 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 intermedi
ate 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 the top 5 most expensive tracks (based on unit pric
e):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5;'}, {'role': 'user',
'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SE
LECT albums.Title AS AlbumTitle, artists.Name AS ArtistName \nFROM albums \nJOIN artists ON albums.ArtistId = artist
s.ArtistId;'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)
\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%';"}, {'role': 'user', 'conten
t': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCou
ntry, COUNT(*) AS TotalInvoices \nFROM invoices \nGROUP BY BillingCountry;'}, {'role': 'user', 'content': ' \n   L
ist all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Tot
al > 10;'}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant', 'c
ontent': "SELECT name FROM sqlite_master WHERE type='table';"}, {'role': 'user', 'content': ' \n   Get the total n
umber of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoi
ces \nFROM invoices \nGROUP BY CustomerId;'}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 and
the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT *, Total \nFROM invoices \nWHERE InvoiceDat
e >= '2010-01-01';"}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assist
ant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': 'How many customers are there'},
{'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': ' \n   List all g

```
enres and the number of tracks in each genre:\n'}]
Using model gpt-4o-mini for 966.75 tokens (approx)
LLM Response: SELECT genres.Name AS GenreName, COUNT(tracks.TrackId) AS TrackCount
FROM genres
LEFT JOIN tracks ON genres.GenreId = tracks.GenreId
GROUP BY genres.GenreId;
Extracted SQL: SELECT genres.Name AS GenreName, COUNT(tracks.TrackId) AS TrackCount
FROM genres
LEFT JOIN tracks ON genres.GenreId = tracks.GenreId
GROUP BY genres.GenreId;
SELECT genres.Name AS GenreName, COUNT(tracks.TrackId) AS TrackCount
FROM genres
LEFT JOIN tracks ON genres.GenreId = tracks.GenreId
GROUP BY genres.GenreId;
             GenreName  TrackCount
0                 Rock        1297
1                 Jazz         130
2                Metal         374
3   Alternative & Punk         332
4        Rock And Roll          12
5                Blues          81
6                Latin         579
7               Reggae          58
8                  Pop          48
9           Soundtrack          43
10          Bossa Nova          15
11      Easy Listening          24
12         Heavy Metal          28
13            R&B/Soul          61
14   Electronica/Dance          30
15               World          28
16         Hip Hop/Rap          35
17     Science Fiction          13
18            TV Shows          93
19    Sci Fi & Fantasy          26
20               Drama          64
21              Comedy          17
22         Alternative          40
23           Classical          74
24               Opera           1
Using model gpt-4o-mini for 206.25 tokens (approx)
```


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

Out[32]:
```
('SELECT genres.Name AS GenreName, COUNT(tracks.TrackId) AS TrackCount \nFROM genres \nLEFT JOIN tracks ON genres.
GenreId = tracks.GenreId \nGROUP BY genres.GenreId;',
              GenreName  TrackCount
 0                  Rock        1297
 1                  Jazz         130
 2                 Metal         374
 3   Alternative & Punk         332
 4        Rock And Roll          12
 5                 Blues          81
 6                 Latin         579
 7                Reggae          58
 8                   Pop          48
 9             Soundtrack          43
 10             Bossa Nova          15
 11        Easy Listening          24
 12           Heavy Metal          28
 13              R&B/Soul          61
 14   Electronica/Dance            30
 15                World          28
 16           Hip Hop/Rap          35
 17     Science Fiction            13
 18              TV Shows          93
 19    Sci Fi & Fantasy            26
 20                Drama          64
 21               Comedy          17
 22           Alternative          40
 23             Classical          74
 24                 Opera           1,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'GenreName=%{x}<br>TrackCount=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Rock', 'Jazz', 'Metal', 'Alternative & Punk', 'Rock And Roll', 'Blues',
```

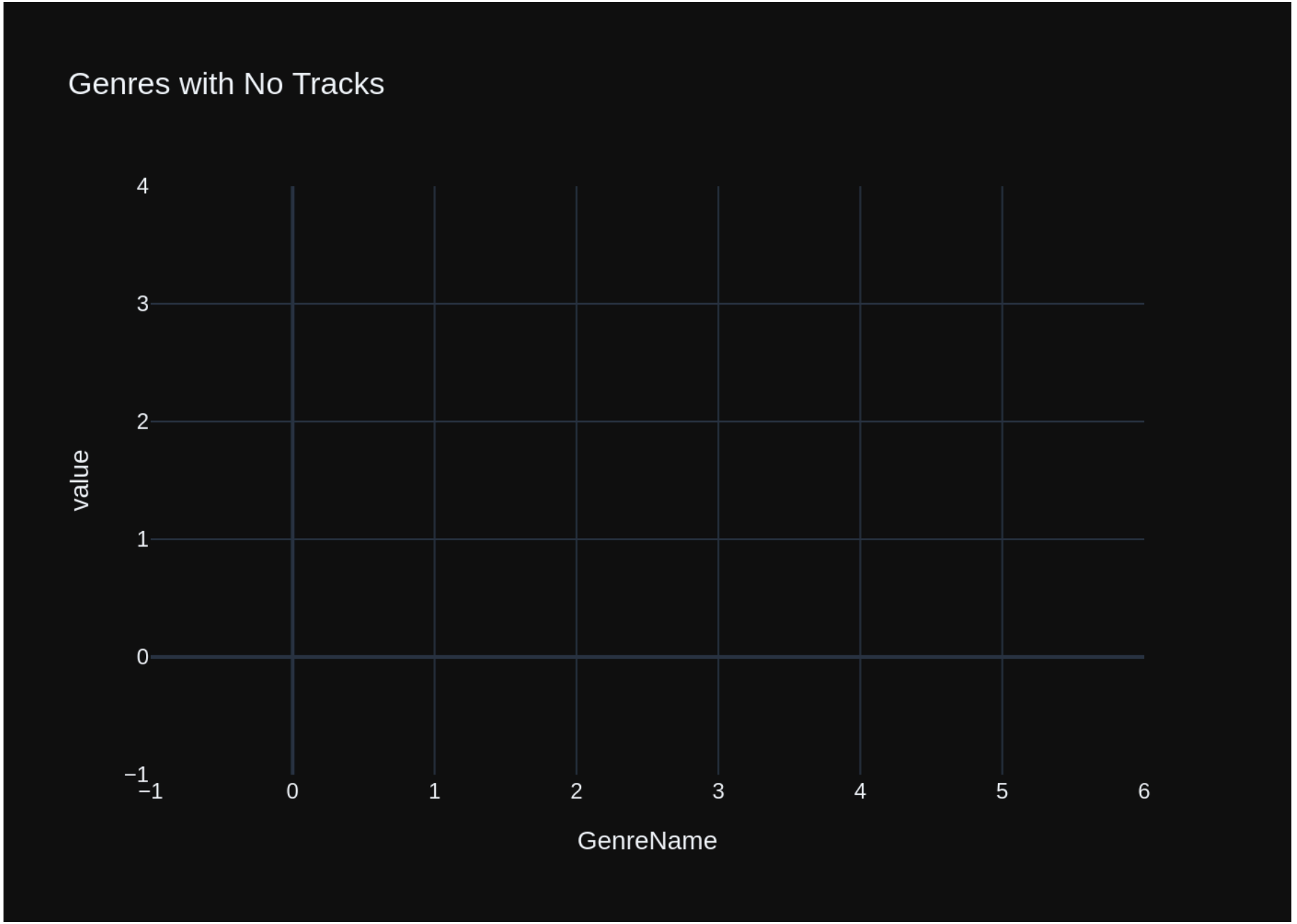
```
                            'Latin', 'Reggae', 'Pop', 'Soundtrack', 'Bossa Nova', 'Easy Listening',
                            'Heavy Metal', 'R&B/Soul', 'Electronica/Dance', 'World', 'Hip Hop/Rap',
                            'Science Fiction', 'TV Shows', 'Sci Fi & Fantasy', 'Drama', 'Comedy',
                            'Alternative', 'Classical', 'Opera'], dtype=object),
                'xaxis': 'x',
                'y': array([1297,  130,  374,  332,   12,   81,  579,   58,   48,   43,   15,   24,
                              28,   61,   30,   28,   35,   13,   93,   26,   64,   17,   40,   74,
                               1]),
                'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Number of Tracks per Genre'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'GenreName'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TrackCount'}}}
 }))
```

In [33]:
```
question = """
    Get all genres that do not have any tracks associated with them:
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE TABLE "tracks"\r\n(\r\n
TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n
MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  N
OT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "al
bums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (G
enreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types"
(MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "play
list_track" (TrackId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE INDEX IFK_TrackAlbumI
d ON "tracks" (AlbumId)\n\nCREATE TABLE "genres"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists"
(ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n    Playlis
tId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistI
d, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDA
TE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO AC
TION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\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 generate a va
lid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requi
res knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the dis
tinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context
is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the
question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role':
'user', 'content': '  \n    List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'cont
ent': 'SELECT genres.Name AS GenreName, COUNT(tracks.TrackId) AS TrackCount \nFROM genres \nLEFT JOIN tracks ON genr
es.GenreId = tracks.GenreId \nGROUP BY genres.GenreId;'}, {'role': 'user', 'content': '  \n    Find all tracks with
a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name L
IKE '%What%';"}, {'role': 'user', 'content': '  \n    List all albums and their corresponding artist names  \n'},
{'role': 'assistant', 'content': 'SELECT albums.Title AS AlbumTitle, artists.Name AS ArtistName \nFROM 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 * FROM tracks ORDER BY UnitPrice DESC LI
MIT 5;'}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assistant', 'con
tent': "SELECT name FROM sqlite_master WHERE type='table';"}, {'role': 'user', 'content': '  \n    Find all invoices
since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT *, Total \nFROM invoices \nW
HERE InvoiceDate >= '2010-01-01';"}, {'role': 'user', 'content': '  \n    List all invoices with a total exceeding
$10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10;'}, {'role': 'user', 'content':
'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'use
r', 'content': "  \n    List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant',
'content': 'SELECT e.EmployeeId, e.FirstName AS EmployeeFirstName, e.LastName AS EmployeeLastName, \n       m.FirstN
ame AS ManagerFirstName, m.LastName AS ManagerLastName\nFROM employees e\nLEFT JOIN employees m ON e.ReportsTo = m.E
```

```
mployeeId;'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per country:\n'}, {'role': 'ass
istant', 'content': 'SELECT BillingCountry, COUNT(*) AS TotalInvoices \nFROM invoices \nGROUP BY BillingCountry;'},
{'role': 'user', 'content': '  \n    Get all genres that do not have any tracks associated with them:\n'}]
Using model gpt-4o-mini for 1041.5 tokens (approx)
LLM Response: SELECT genres.Name AS GenreName
FROM genres
LEFT JOIN tracks ON genres.GenreId = tracks.GenreId
WHERE tracks.TrackId IS NULL;
Extracted SQL: SELECT genres.Name AS GenreName
FROM genres
LEFT JOIN tracks ON genres.GenreId = tracks.GenreId
WHERE tracks.TrackId IS NULL;
SELECT genres.Name AS GenreName
FROM genres
LEFT JOIN tracks ON genres.GenreId = tracks.GenreId
WHERE tracks.TrackId IS NULL;
Empty DataFrame
Columns: [GenreName]
Index: []
Using model gpt-4o-mini for 195.0 tokens (approx)
```


Genres with No Tracks
value
4
3
2
1
0
−1
−1
0
1
2
3
4
5
6
GenreName

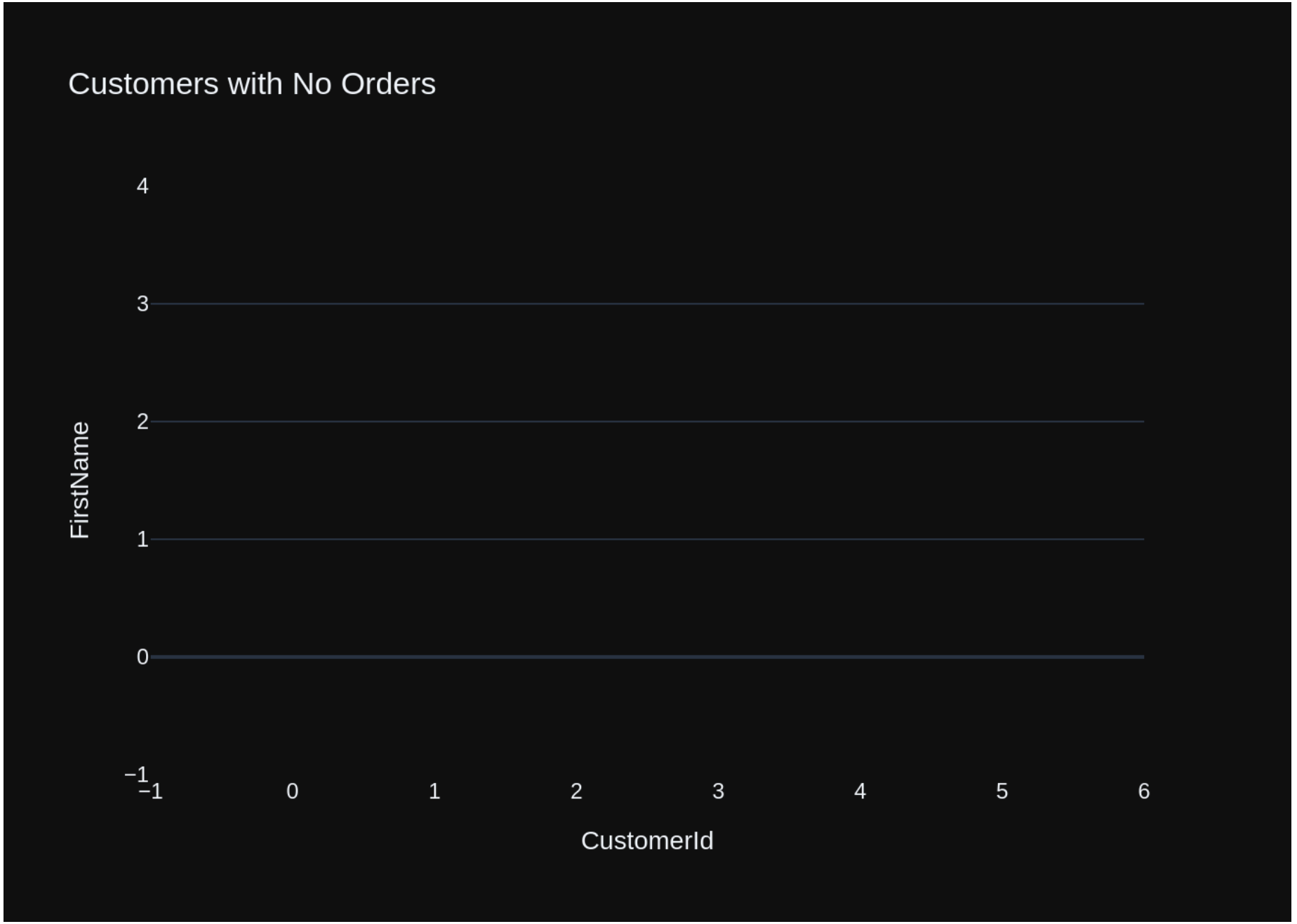
Out[33]:
```
('SELECT genres.Name AS GenreName \nFROM genres \nLEFT JOIN tracks ON genres.GenreId = tracks.GenreId \nWHERE trac
ks.TrackId IS NULL;',
 Empty DataFrame
 Columns: [GenreName]
 Index: [],
 Figure({
     'data': [],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Genres with No Tracks'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'GenreName'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'value'}}}
 }))
```

In [34]:
```
question = """
    List all customers who have not placed any orders:
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r
\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n    Custome
rId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(2
0)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARC
HAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR
(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES
"employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r
\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId I
NTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (I
nvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Tra
ckId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "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 UPDA
TE NO ACTION\r\n)\n\n\n===Additional Context \n\n===Response Guidelines \n1. If the provided context is sufficient,
please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost
sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL q
uery to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If t
he provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant ta
ble(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given be
fore. \n'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT CO
UNT(*) FROM customers;'}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'as
sistant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': '  \n    Get the total number o
f invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices \nF
ROM invoices \nGROUP BY CustomerId;'}, {'role': 'user', 'content': '  \n    List all invoices with a total exceeding
$10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10;'}, {'role': 'user', 'content':
'  \n    Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT
*, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01';"}, {'role': 'user', 'content': '  \n    Get the averag
e invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageIn
voiceTotal \nFROM invoices \nGROUP BY CustomerId;'}, {'role': 'user', 'content': '  \n    Find the total number of i
nvoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(*) AS TotalInvoices \nFROM
invoices \nGROUP BY BillingCountry;'}, {'role': 'user', 'content': "  \n    List all employees and their reporting m
anager's name (if any):\n"}, {'role': 'assistant', 'content': 'SELECT e.EmployeeId, e.FirstName AS EmployeeFirstNam
e, e.LastName AS EmployeeLastName, \n       m.FirstName AS ManagerFirstName, m.LastName AS ManagerLastName\nFROM emp
loyees e\nLEFT JOIN employees m ON e.ReportsTo = m.EmployeeId;'}, {'role': 'user', 'content': '  \n    List all albu
ms and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT albums.Title AS AlbumTitle,
artists.Name AS ArtistName \nFROM 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', 'conten
t': 'SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5;'}, {'role': 'user', 'content': '  \n    List all customer
s who have not placed any orders:\n'}]
Using model gpt-4o-mini for 1018.0 tokens (approx)
LLM Response: SELECT * FROM customers
WHERE CustomerId NOT IN (SELECT DISTINCT CustomerId FROM invoices);
Extracted SQL: SELECT * FROM customers
WHERE CustomerId NOT IN (SELECT DISTINCT CustomerId FROM invoices);
SELECT * FROM customers
WHERE CustomerId NOT IN (SELECT DISTINCT CustomerId FROM invoices);
Empty DataFrame
Columns: [CustomerId, FirstName, LastName, Company, Address, City, State, Country, PostalCode, Phone, Fax, Email, Su
pportRepId]
Index: []
Using model gpt-4o-mini for 252.25 tokens (approx)
```


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

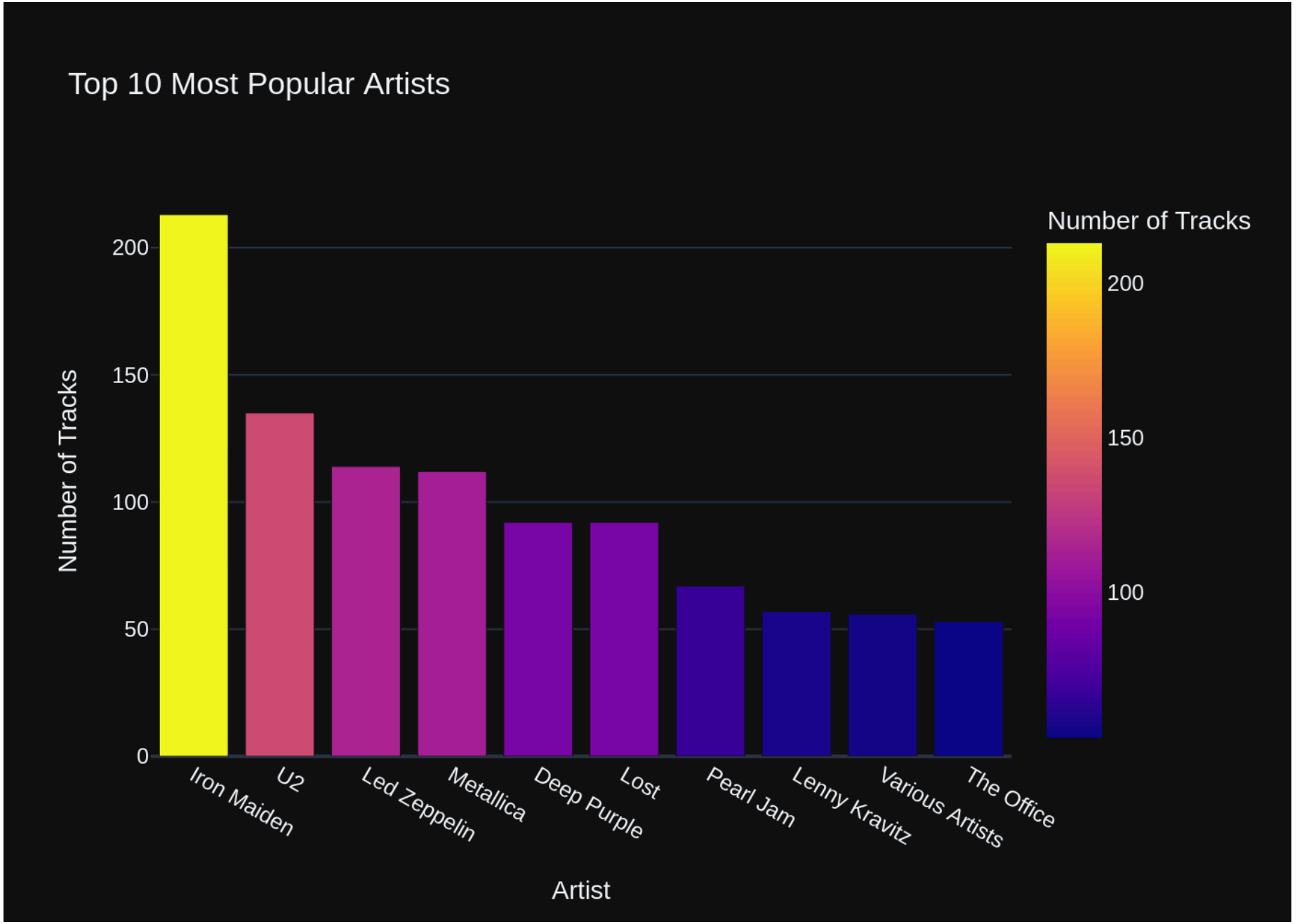
Out[34]:
```
('SELECT * FROM customers \nWHERE CustomerId NOT IN (SELECT DISTINCT CustomerId FROM invoices);',
 Empty DataFrame
 Columns: [CustomerId, FirstName, LastName, Company, Address, City, State, Country, PostalCode, Phone, Fax, Email,
SupportRepId]
 Index: [],
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'CustomerId=%{x}<br>FirstName=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([], dtype=object),
               'xaxis': 'x',
               'y': array([], dtype=object),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Customers with No Orders'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'FirstName'}}}
 }))
```

In [35]:
```
question = """
    Get the top 10 most popular artists (based on the number of tracks):
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGE
R,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUME
RIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks"
(AlbumId)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGE
R PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n
FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREAT
E TABLE "artists"\r\n(\r\n    ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n
\nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON "track
s" (MediaTypeId)\n\nCREATE TABLE "playlists"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Gu
idelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for
the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a par
ticular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the q
uery with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can
\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered befor
e, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': '  \n    Find the top
5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks ORDER BY
UnitPrice DESC LIMIT 5;'}, {'role': 'user', 'content': '  \n    List all genres and the number of tracks in each gen
re:\n'}, {'role': 'assistant', 'content': 'SELECT genres.Name AS GenreName, COUNT(tracks.TrackId) AS TrackCount \nFR
OM genres \nLEFT JOIN tracks ON genres.GenreId = tracks.GenreId \nGROUP BY genres.GenreId;'}, {'role': 'user', 'cont
ent': '  \n    List all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT
albums.Title AS AlbumTitle, artists.Name AS ArtistName \nFROM albums \nJOIN artists ON albums.ArtistId = artists.Art
istId;'}, {'role': 'user', 'content': '  \n    Find all tracks with a name containing "What" (case-insensitive)\n'},
{'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%';"}, {'role': 'user', 'content': '
\n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WH
ERE Total > 10;'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SE
LECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': '  \n    Get the average invoice total for each custome
r:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM invoices \nGRO
UP BY CustomerId;'}, {'role': 'user', 'content': 'How many records are in table called customer'}, {'role': 'assista
nt', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': '  \n    Get the total number of inv
oices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices \nFROM i
nvoices \nGROUP BY CustomerId;'}, {'role': 'user', 'content': '  \n    Find the total number of invoices per countr
y:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(*) AS TotalInvoices \nFROM invoices \nGROUP B
Y BillingCountry;'}, {'role': 'user', 'content': '  \n    Get the top 10 most popular artists (based on the number o
f tracks):\n'}]
```

```
Using model gpt-4o-mini for 933.75 tokens (approx)
LLM Response: SELECT artists.Name AS ArtistName, COUNT(tracks.TrackId) AS TrackCount
FROM artists
JOIN albums ON artists.ArtistId = albums.ArtistId
JOIN tracks ON albums.AlbumId = tracks.AlbumId
GROUP BY artists.ArtistId
ORDER BY TrackCount DESC
LIMIT 10;
Extracted SQL: SELECT artists.Name AS ArtistName, COUNT(tracks.TrackId) AS TrackCount
FROM artists
JOIN albums ON artists.ArtistId = albums.ArtistId
JOIN tracks ON albums.AlbumId = tracks.AlbumId
GROUP BY artists.ArtistId
ORDER BY TrackCount DESC
LIMIT 10;
SELECT artists.Name AS ArtistName, COUNT(tracks.TrackId) AS TrackCount
FROM artists
JOIN albums ON artists.ArtistId = albums.ArtistId
JOIN tracks ON albums.AlbumId = tracks.AlbumId
GROUP BY artists.ArtistId
ORDER BY TrackCount DESC
LIMIT 10;
        ArtistName  TrackCount
0      Iron Maiden         213
1               U2         135
2     Led Zeppelin         114
3        Metallica         112
4      Deep Purple          92
5             Lost          92
6        Pearl Jam          67
7    Lenny Kravitz          57
8  Various Artists          56
9       The Office          53
Using model gpt-4o-mini for 231.25 tokens (approx)
```


Top 10 Most Popular Artists
Number of Tracks
200
150
100
50
0
Number of Tracks
Iron Maiden
U2
Led Zeppelin
Metallica
Deep Purple
Lost
Pearl Jam
Lenny Kravitz
Various Artists
The Office
Artist

Out[35]:
```
('SELECT artists.Name AS ArtistName, COUNT(tracks.TrackId) AS TrackCount \nFROM artists \nJOIN albums ON artists.A
rtistId = albums.ArtistId \nJOIN tracks ON albums.AlbumId = tracks.AlbumId \nGROUP BY artists.ArtistId \nORDER BY
TrackCount DESC \nLIMIT 10;',
         ArtistName  TrackCount
 0      Iron Maiden         213
 1               U2         135
 2     Led Zeppelin         114
 3        Metallica         112
 4      Deep Purple          92
 5             Lost          92
 6        Pearl Jam          67
 7    Lenny Kravitz          57
 8  Various Artists          56
 9       The Office          53,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'Artist=%{x}<br>Number of Tracks=%{marker.color}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': array([213, 135, 114, 112,  92,  92,  67,  57,  56,  53]),
                          'coloraxis': 'coloraxis',
                          'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Iron Maiden', 'U2', 'Led Zeppelin', 'Metallica', 'Deep Purple', 'Lost',
                           'Pearl Jam', 'Lenny Kravitz', 'Various Artists', 'The Office'],
                          dtype=object),
               'xaxis': 'x',
               'y': array([213, 135, 114, 112,  92,  92,  67,  57,  56,  53]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'coloraxis': {'colorbar': {'title': {'text': 'Number of Tracks'}},
                              'colorscale': [[0.0, '#0d0887'], [0.1111111111111111,
                                             '#46039f'], [0.2222222222222222,
                                             '#7201a8'], [0.3333333333333333,
                                             '#9c179e'], [0.4444444444444444,
                                             '#bd3786'], [0.5555555555555556,
```

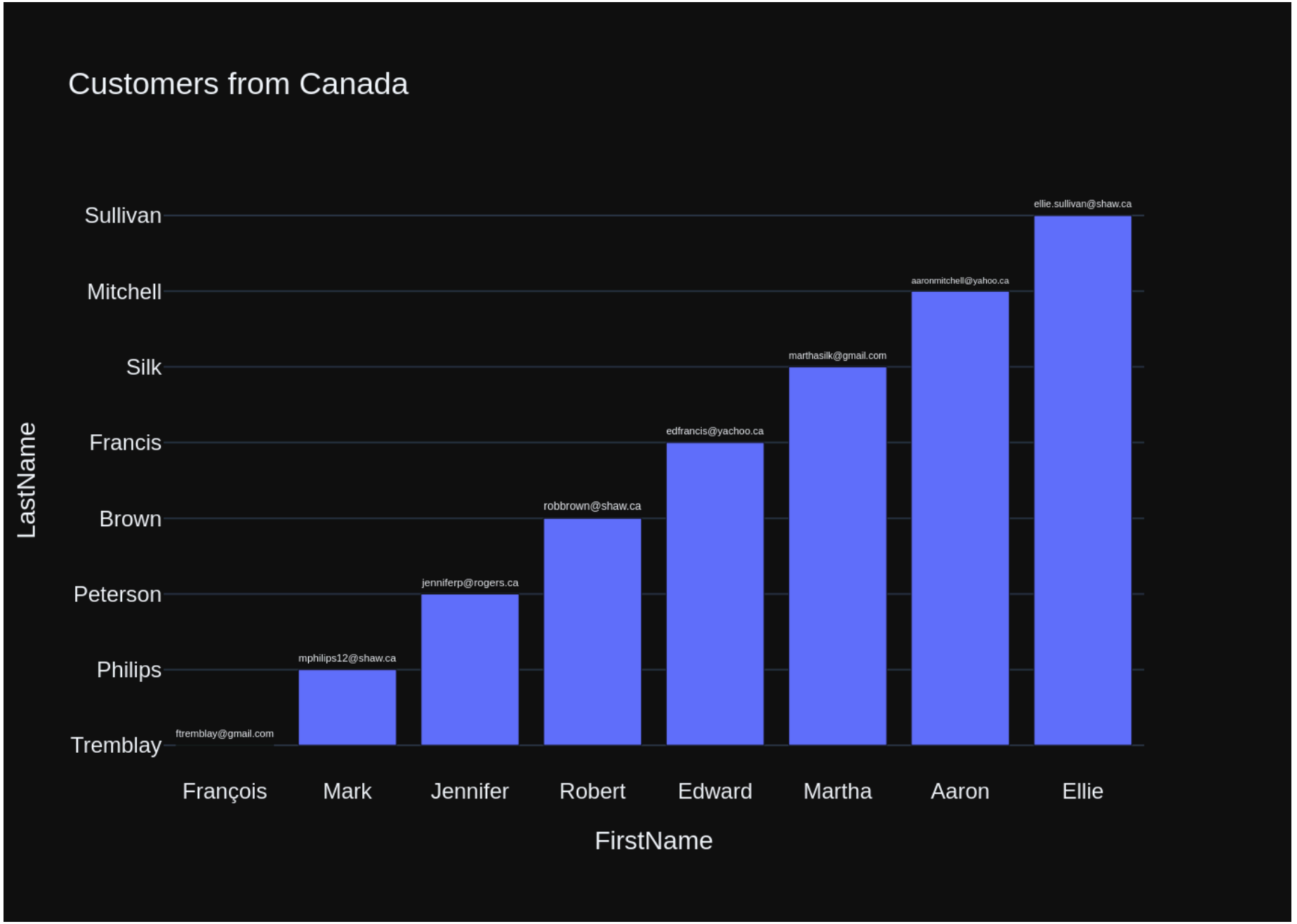
```
                                                '#d8576b'], [0.6666666666666666,
                                                '#ed7953'], [0.7777777777777778,
                                                '#fb9f3a'], [0.8888888888888888,
                                                '#fdca26'], [1.0, '#f0f921']]},
                    'legend': {'tracegroupgap': 0},
                    'template': '...',
                    'title': {'text': 'Top 10 Most Popular Artists'},
                    'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Artist'}},
                    'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Number of Tracks'}}}
          }))
```

In [36]:
```
question = """
     List all customers from Canada and their email addresses:
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "cust
omers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r
\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHA
R(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR
(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY
(SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TA
BLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NUL
L,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n
BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total N
UMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACT
ION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE TABLE "emplo
yees"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n
FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r
\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Co
untry NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email
NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDA
TE NO ACTION\r\n)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\n\n===Additional Context \n\n===Response Guidelines \n
1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the questio
n. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular col
umn, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a
comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be genera
ted. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please rep
eat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': 'How many customers are there'},
{'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': 'How many records a
re in table called customer'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user',
'content': '  \n    Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT
CustomerId, COUNT(*) AS TotalInvoices \nFROM invoices \nGROUP BY CustomerId;'}, {'role': 'user', 'content': '  \n
Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT
(*) AS TotalInvoices \nFROM invoices \nGROUP BY BillingCountry;'}, {'role': 'user', 'content': "  \n    List all emp
loyees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': 'SELECT e.EmployeeId, e.Fi
rstName AS EmployeeFirstName, e.LastName AS EmployeeLastName, \n       m.FirstName AS ManagerFirstName, m.LastName A
S ManagerLastName\nFROM employees e\nLEFT JOIN employees m ON e.ReportsTo = m.EmployeeId;'}, {'role': 'user', 'conte
nt': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM inv
oices WHERE Total > 10;'}, {'role': 'user', 'content': '  \n    Get the average invoice total for each custome
r:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM invoices \nGRO
UP BY CustomerId;'}, {'role': 'user', 'content': '  \n    Find all invoices since 2010 and the total amount invoice
d:\n'}, {'role': 'assistant', 'content': "SELECT *, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01';"},
{'role': 'user', 'content': '  \n    Get the top 10 most popular artists (based on the number of tracks):\n'}, {'rol
e': 'assistant', 'content': 'SELECT artists.Name AS ArtistName, COUNT(tracks.TrackId) AS TrackCount \nFROM artists
```

```
\nJOIN albums ON artists.ArtistId = albums.ArtistId \nJOIN tracks ON albums.AlbumId = tracks.AlbumId \nGROUP BY arti
sts.ArtistId \nORDER BY TrackCount DESC \nLIMIT 10;'}, {'role': 'user', 'content': '  \n    List all albums and thei
r corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT albums.Title AS AlbumTitle, artists.Nam
e AS ArtistName \nFROM albums \nJOIN artists ON albums.ArtistId = artists.ArtistId;'}, {'role': 'user', 'content': '
\n     List all customers from Canada and their email addresses:\n'}]
Using model gpt-4o-mini for 1082.75 tokens (approx)
LLM Response: SELECT FirstName, LastName, Email
FROM customers
WHERE Country = 'Canada';
Extracted SQL: SELECT FirstName, LastName, Email
FROM customers
WHERE Country = 'Canada';
SELECT FirstName, LastName, Email
FROM customers
WHERE Country = 'Canada';
  FirstName  LastName                  Email
0  François  Tremblay     ftremblay@gmail.com
1      Mark   Philips      mphilips12@shaw.ca
2  Jennifer  Peterson     jenniferp@rogers.ca
3    Robert     Brown        robbrown@shaw.ca
4    Edward   Francis     edfrancis@yachoo.ca
5    Martha      Silk    marthasilk@gmail.com
6     Aaron  Mitchell  aaronmitchell@yahoo.ca
7     Ellie  Sullivan  ellie.sullivan@shaw.ca
Using model gpt-4o-mini for 190.5 tokens (approx)
```


Customers from Canada
LastName
Sullivan
Mitchell
Silk
Francis
Brown
Peterson
Philips
Tremblay
ellie.sullivan@shaw.ca
aaronmitchell@yahoo.ca
marthasilk@gmail.com
edfrancis@yachoo.ca
robbrown@shaw.ca
jenniferp@rogers.ca
mphilips12@shaw.ca
ftremblay@gmail.com
François
Mark
Jennifer
Robert
Edward
Martha
Aaron
Ellie
FirstName

Out[36]:
```
("SELECT FirstName, LastName, Email \nFROM customers \nWHERE Country = 'Canada';",
  FirstName  LastName                  Email
0  François  Tremblay     ftremblay@gmail.com
1      Mark   Philips      mphilips12@shaw.ca
2  Jennifer  Peterson     jenniferp@rogers.ca
3    Robert     Brown        robbrown@shaw.ca
4    Edward   Francis     edfrancis@yachoo.ca
5    Martha      Silk    marthasilk@gmail.com
6     Aaron  Mitchell  aaronmitchell@yahoo.ca
7     Ellie  Sullivan  ellie.sullivan@shaw.ca,
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'FirstName=%{x}<br>LastName=%{y}<br>Email=%{text}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'v',
              'showlegend': False,
              'text': array(['ftremblay@gmail.com', 'mphilips12@shaw.ca', 'jenniferp@rogers.ca',
                             'robbrown@shaw.ca', 'edfrancis@yachoo.ca', 'marthasilk@gmail.com',
                             'aaronmitchell@yahoo.ca', 'ellie.sullivan@shaw.ca'], dtype=object),
              'textposition': 'outside',
              'texttemplate': '%{text}',
              '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'}],
    'layout': {'barmode': 'relative',
               'legend': {'tracegroupgap': 0},
               'template': '...',
               'title': {'text': 'Customers from Canada'},
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'FirstName'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'LastName'}}}
}))
```

In [37]:
```
question = """
     Find the customer with the most invoices
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r
\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoi
ces" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE TABLE "invoice_i
tems"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n
TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREI
GN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN
KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK
_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY
KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n
Company NVARCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Count
ry NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NV
ARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (Employe
eId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (S
upportRepId)\n\n\n===Additional Context \n\n===Response Guidelines \n1. If the provided context is sufficient, pleas
e generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost suffi
cient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query
to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the pr
ovided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table
(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given befor
e. \n'}, {'role': 'user', 'content': '  \n    Get the total number of invoices for each customer\n'}, {'role': 'assi
stant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices \nFROM invoices \nGROUP BY CustomerId;'}, {'role':
'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SEL
ECT * FROM invoices WHERE Total > 10;'}, {'role': 'user', 'content': '  \n    Find all invoices since 2010 and the t
otal amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT *, Total \nFROM invoices \nWHERE InvoiceDate >=
'2010-01-01';"}, {'role': 'user', 'content': '  \n    Find the total number of invoices per country:\n'}, {'role':
'assistant', 'content': 'SELECT BillingCountry, COUNT(*) AS TotalInvoices \nFROM invoices \nGROUP BY BillingCountr
y;'}, {'role': 'user', 'content': '  \n    Get the average invoice total for each customer:\n'}, {'role': 'assistan
t', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM invoices \nGROUP BY CustomerId;'}, {'rol
e': 'user', 'content': '  \n    Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistan
t', 'content': 'SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5;'}, {'role': 'user', 'content': 'How many custo
mers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content':
'How many records are in table called customer'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customer
s;'}, {'role': 'user', 'content': '  \n     List all customers from Canada and their email addresses:\n'}, {'role':
'assistant', 'content': "SELECT FirstName, LastName, Email \nFROM customers \nWHERE Country = 'Canada';"}, {'role':
'user', 'content': "  \n    List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistan
t', 'content': 'SELECT e.EmployeeId, e.FirstName AS EmployeeFirstName, e.LastName AS EmployeeLastName, \n       m.Fi
```

```
rstName AS ManagerFirstName, m.LastName AS ManagerLastName\nFROM employees e\nLEFT JOIN employees m ON e.ReportsTo =
m.EmployeeId;'}, {'role': 'user', 'content': '  \n     Find the customer with the most invoices \n'}]
Using model gpt-4o-mini for 1006.5 tokens (approx)
LLM Response: SELECT CustomerId, COUNT(*) AS InvoiceCount
FROM invoices
GROUP BY CustomerId
ORDER BY InvoiceCount DESC
LIMIT 1;
Extracted SQL: SELECT CustomerId, COUNT(*) AS InvoiceCount
FROM invoices
GROUP BY CustomerId
ORDER BY InvoiceCount DESC
LIMIT 1;
SELECT CustomerId, COUNT(*) AS InvoiceCount
FROM invoices
GROUP BY CustomerId
ORDER BY InvoiceCount DESC
LIMIT 1;
   CustomerId  InvoiceCount
0           1             7
Using model gpt-4o-mini for 192.75 tokens (approx)
```


Customer ID: 1
7
▲7

Out[37]:
```
('SELECT CustomerId, COUNT(*) AS InvoiceCount \nFROM invoices \nGROUP BY CustomerId \nORDER BY InvoiceCount DESC
\nLIMIT 1;',
    CustomerId  InvoiceCount
 0           1             7,
 Figure({
     'data': [{'delta': {'reference': 0},
               'mode': 'number+delta',
               'title': {'text': 'Customer ID: 1'},
               'type': 'indicator',
               'value': 7}],
     'layout': {'template': '...'}
 }))
```

In [ ]:
```
```

Advanced SQL questions

In [38]:
```
question = """
     Find the customer who bought the most albums in total quantity (across all invoices):
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGE
R,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUME
RIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r
\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE
NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTIO
N ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES
"artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albu
ms" (ArtistId)\n\nCREATE 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\nCREATE IN
DEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n    ArtistId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \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 particular column, please
generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment say
ing intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. P
lease use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the ans
wer exactly as it was given before. \n'}, {'role': 'user', 'content': '  \n     Find the customer with the most invo
ices \n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS InvoiceCount \nFROM invoices \nGROUP BY
CustomerId \nORDER BY InvoiceCount DESC \nLIMIT 1;'}, {'role': 'user', 'content': '  \n    Get the total number of i
nvoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS TotalInvoices \nFROM
invoices \nGROUP BY CustomerId;'}, {'role': 'user', 'content': '  \n    Get the top 10 most popular artists (based o
n the number of tracks):\n'}, {'role': 'assistant', 'content': 'SELECT artists.Name AS ArtistName, COUNT(tracks.Trac
kId) AS TrackCount \nFROM artists \nJOIN albums ON artists.ArtistId = albums.ArtistId \nJOIN tracks ON albums.AlbumI
d = tracks.AlbumId \nGROUP BY artists.ArtistId \nORDER BY TrackCount DESC \nLIMIT 10;'}, {'role': 'user', 'content':
'  \n    Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT
*, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01';"}, {'role': 'user', 'content': '  \n    Find the top 5
most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks ORDER BY Un
itPrice DESC LIMIT 5;'}, {'role': 'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'r
ole': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10;'}, {'role': 'user', 'content': '  \n    Get
the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS
AverageInvoiceTotal \nFROM invoices \nGROUP BY CustomerId;'}, {'role': 'user', 'content': '  \n    Find the total nu
mber of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(*) AS TotalInvoice
s \nFROM invoices \nGROUP BY BillingCountry;'}, {'role': 'user', 'content': '  \n    List all albums and their corre
```

```
sponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT albums.Title AS AlbumTitle, artists.Name AS Ar
tistName \nFROM albums \nJOIN artists ON albums.ArtistId = artists.ArtistId;'}, {'role': 'user', 'content': '  \n
List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT genres.Name AS
GenreName, COUNT(tracks.TrackId) AS TrackCount \nFROM genres \nLEFT JOIN tracks ON genres.GenreId = tracks.GenreId
\nGROUP BY genres.GenreId;'}, {'role': 'user', 'content': '  \n     Find the customer who bought the most albums in
total quantity (across all invoices): \n'}]
Using model gpt-4o-mini for 1106.5 tokens (approx)
LLM Response: SELECT Invoice.CustomerId, SUM(InvoiceItems.Quantity) AS TotalAlbums
FROM invoice_items AS InvoiceItems
JOIN invoices AS Invoice ON InvoiceItems.InvoiceId = Invoice.InvoiceId
GROUP BY Invoice.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1;
Extracted SQL: SELECT Invoice.CustomerId, SUM(InvoiceItems.Quantity) AS TotalAlbums
FROM invoice_items AS InvoiceItems
JOIN invoices AS Invoice ON InvoiceItems.InvoiceId = Invoice.InvoiceId
GROUP BY Invoice.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1;
SELECT Invoice.CustomerId, SUM(InvoiceItems.Quantity) AS TotalAlbums
FROM invoice_items AS InvoiceItems
JOIN invoices AS Invoice ON InvoiceItems.InvoiceId = Invoice.InvoiceId
GROUP BY Invoice.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1;
   CustomerId  TotalAlbums
0           1           38
Using model gpt-4o-mini for 234.75 tokens (approx)
```


Customer ID: 1
Total Albums Bought
38

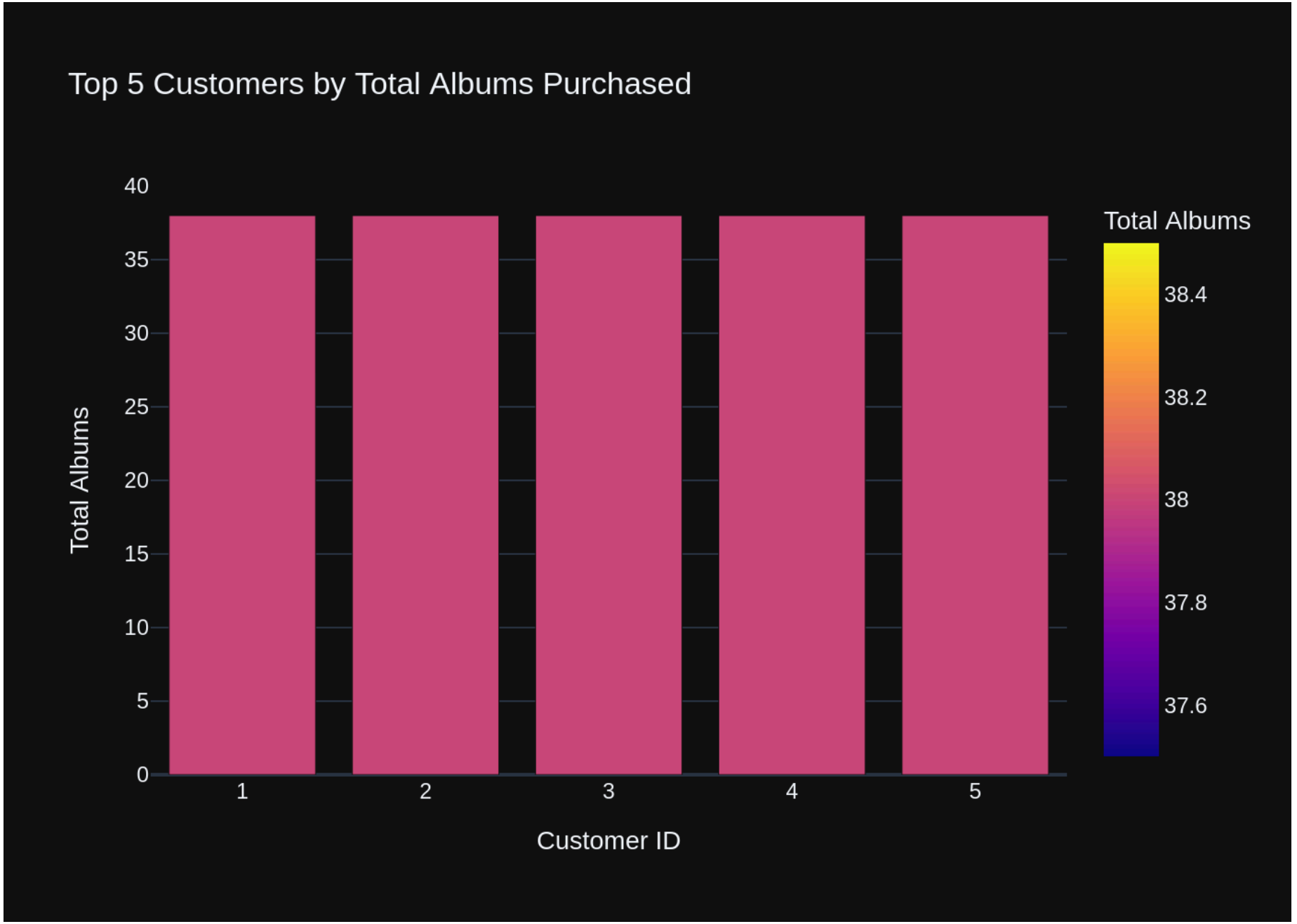
Out[38]:
```
('SELECT Invoice.CustomerId, SUM(InvoiceItems.Quantity) AS TotalAlbums \nFROM invoice_items AS InvoiceItems \nJOIN
invoices AS Invoice ON InvoiceItems.InvoiceId = Invoice.InvoiceId \nGROUP BY Invoice.CustomerId \nORDER BY TotalAl
bums DESC \nLIMIT 1;',
    CustomerId  TotalAlbums
 0           1           38,
 Figure({
     'data': [{'mode': 'number',
               'title': {'text': 'Customer ID: 1<br>Total Albums Bought'},
               'type': 'indicator',
               'value': 38}],
     'layout': {'template': '...'}
 }))
```

In [39]:
```
question = """
     Find the top 5 customer who bought the most albums in total quantity (across all invoices):
"""

vn.ask(question=question)
```

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

SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGE
R,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUME
RIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title
NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES "artists" (Arti
stId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId
INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r
\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCE
S "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "t
racks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums"
(ArtistId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE INDEX IFK_InvoiceCustomerId
ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE INDEX I
FK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n   ArtistId INTEGER PRIMARY KEY AUTOINCREME
NT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\n===Response Guidelines \n1. If the provid
ed context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the p
rovided context is almost sufficient but requires knowledge of a specific string in a particular column, please gene
rate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying
intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Pleas
e use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer
exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Find the customer who bought the most al
bums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT Invoice.CustomerId, SUM
(InvoiceItems.Quantity) AS TotalAlbums \nFROM invoice_items AS InvoiceItems \nJOIN invoices AS Invoice ON InvoiceIte
ms.InvoiceId = Invoice.InvoiceId \nGROUP BY Invoice.CustomerId \nORDER BY TotalAlbums DESC \nLIMIT 1;'}, {'role': 'u
ser', 'content': ' \n   Get the top 10 most popular artists (based on the number of tracks):\n'}, {'role': 'assist
ant', 'content': 'SELECT artists.Name AS ArtistName, COUNT(tracks.TrackId) AS TrackCount \nFROM artists \nJOIN album
s ON artists.ArtistId = albums.ArtistId \nJOIN tracks ON albums.AlbumId = tracks.AlbumId \nGROUP BY artists.ArtistId
\nORDER BY TrackCount DESC \nLIMIT 10;'}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks
(based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT
5;'}, {'role': 'user', 'content': ' \n   Find the customer with the most invoices \n'}, {'role': 'assistant', 'co
ntent': 'SELECT CustomerId, COUNT(*) AS InvoiceCount \nFROM invoices \nGROUP BY CustomerId \nORDER BY InvoiceCount D
ESC \nLIMIT 1;'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role':
'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10;'}, {'role': 'user', 'content': ' \n   Get the to
tal number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS Total
Invoices \nFROM invoices \nGROUP BY CustomerId;'}, {'role': 'user', 'content': ' \n   Get the average invoice tota
l for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nF
ROM invoices \nGROUP BY CustomerId;'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding

```
artist names  \n'}, {'role': 'assistant', 'content': 'SELECT albums.Title AS AlbumTitle, artists.Name AS ArtistName
\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 *, Total \nFROM inv
oices \nWHERE InvoiceDate >= '2010-01-01';"}, {'role': 'user', 'content': '  \n    Find the total number of invoices
per country:\n'}, {'role': 'assistant', 'content': 'SELECT BillingCountry, COUNT(*) AS TotalInvoices \nFROM invoices
\nGROUP BY BillingCountry;'}, {'role': 'user', 'content': '  \n     Find the top 5 customer who bought the most albu
ms in total quantity (across all invoices):\n'}]
Using model gpt-4o-mini for 1136.25 tokens (approx)
LLM Response: SELECT Invoice.CustomerId, SUM(InvoiceItems.Quantity) AS TotalAlbums
FROM invoice_items AS InvoiceItems
JOIN invoices AS Invoice ON InvoiceItems.InvoiceId = Invoice.InvoiceId
GROUP BY Invoice.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5;
Extracted SQL: SELECT Invoice.CustomerId, SUM(InvoiceItems.Quantity) AS TotalAlbums
FROM invoice_items AS InvoiceItems
JOIN invoices AS Invoice ON InvoiceItems.InvoiceId = Invoice.InvoiceId
GROUP BY Invoice.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5;
SELECT Invoice.CustomerId, SUM(InvoiceItems.Quantity) AS TotalAlbums
FROM invoice_items AS InvoiceItems
JOIN invoices AS Invoice ON InvoiceItems.InvoiceId = Invoice.InvoiceId
GROUP BY Invoice.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5;
   CustomerId  TotalAlbums
0           1           38
1           2           38
2           3           38
3           4           38
4           5           38
Using model gpt-4o-mini for 236.0 tokens (approx)
```


Top 5 Customers by Total Albums Purchased
Total Albums
40
35
30
25
20
15
10
5
0
1
2
3
4
5
Customer ID
Total Albums
38.4
38.2
38
37.8
37.6

Out[39]:
```
('SELECT Invoice.CustomerId, SUM(InvoiceItems.Quantity) AS TotalAlbums \nFROM invoice_items AS InvoiceItems \nJOIN
invoices AS Invoice ON InvoiceItems.InvoiceId = Invoice.InvoiceId \nGROUP BY Invoice.CustomerId \nORDER BY TotalAl
bums DESC \nLIMIT 5;',
    CustomerId  TotalAlbums
 0           1           38
 1           2           38
 2           3           38
 3           4           38
 4           5           38,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'Customer ID=%{x}<br>Total Albums=%{marker.color}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': array([38, 38, 38, 38, 38]), 'coloraxis': 'coloraxis', 'pattern': {'shape':
''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([1, 2, 3, 4, 5]),
               'xaxis': 'x',
               'y': array([38, 38, 38, 38, 38]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'coloraxis': {'colorbar': {'title': {'text': 'Total Albums'}},
                              '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': 'Top 5 Customers by Total Albums Purchased'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Customer ID'}},
```

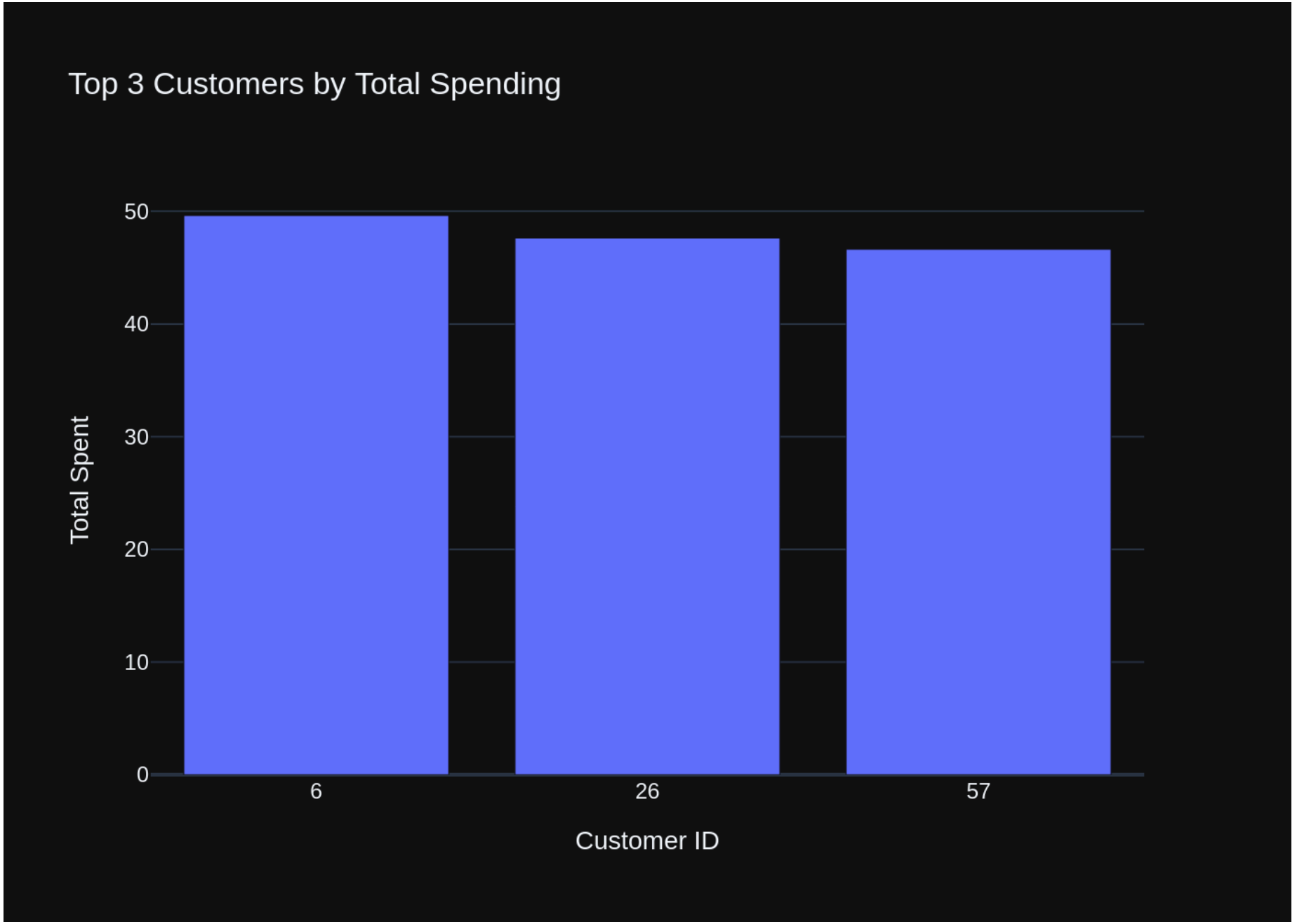
```
                    'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total Albums'}}}
  }))
```

In [40]:
```
question = """
     Find the top 3 customers who spent the most money overall:
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r
\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    Billing
PostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "customer
s" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "cu
stomers" (SupportRepId)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT
NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT
NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE N
O ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREME
NT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHA
R(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(4
0),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  N
OT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\t
ON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\nCRE
ATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items"
(TrackId)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\n\n===Additional Context \n\n===Response Guidelines \n1. If t
he 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, ple
ase generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment
saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n
4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the
answer exactly as it was given before. \n'}, {'role': 'user', 'content': '  \n     Find the top 5 customer who bough
t the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT Invoice.Cus
tomerId, SUM(InvoiceItems.Quantity) AS TotalAlbums \nFROM invoice_items AS InvoiceItems \nJOIN invoices AS Invoice O
N InvoiceItems.InvoiceId = Invoice.InvoiceId \nGROUP BY Invoice.CustomerId \nORDER BY TotalAlbums DESC \nLIMIT 5;'},
{'role': 'user', 'content': '  \n    Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assi
stant', 'content': 'SELECT * FROM tracks ORDER BY UnitPrice DESC LIMIT 5;'}, {'role': 'user', 'content': '  \n     F
ind the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'co
ntent': 'SELECT Invoice.CustomerId, SUM(InvoiceItems.Quantity) AS TotalAlbums \nFROM invoice_items AS InvoiceItems
\nJOIN invoices AS Invoice ON InvoiceItems.InvoiceId = Invoice.InvoiceId \nGROUP BY Invoice.CustomerId \nORDER BY To
talAlbums DESC \nLIMIT 1;'}, {'role': 'user', 'content': '  \n     Find the customer with the most invoices \n'},
{'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(*) AS InvoiceCount \nFROM invoices \nGROUP BY CustomerId
\nORDER BY InvoiceCount DESC \nLIMIT 1;'}, {'role': 'user', 'content': '  \n    Get the average invoice total for ea
ch customer:\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, AVG(Total) AS AverageInvoiceTotal \nFROM invo
ices \nGROUP BY CustomerId;'}, {'role': 'user', 'content': '  \n    Get the top 10 most popular artists (based on th
e number of tracks):\n'}, {'role': 'assistant', 'content': 'SELECT artists.Name AS ArtistName, COUNT(tracks.TrackId)
AS TrackCount \nFROM artists \nJOIN albums ON artists.ArtistId = albums.ArtistId \nJOIN tracks ON albums.AlbumId = t
racks.AlbumId \nGROUP BY artists.ArtistId \nORDER BY TrackCount DESC \nLIMIT 10;'}, {'role': 'user', 'content': '
```

```
\n    Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, C
OUNT(*) AS TotalInvoices \nFROM invoices \nGROUP BY CustomerId;'}, {'role': 'user', 'content': 'How many customers a
re there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers;'}, {'role': 'user', 'content': '  \n
List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE To
tal > 10;'}, {'role': 'user', 'content': '  \n    Find all invoices since 2010 and the total amount invoiced:\n'},
{'role': 'assistant', 'content': "SELECT *, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01';"}, {'role':
'user', 'content': '  \n     Find the top 3 customers who spent the most money overall:\n'}]
Using model gpt-4o-mini for 1146.75 tokens (approx)
LLM Response: SELECT CustomerId, SUM(Total) AS TotalSpent
FROM invoices
GROUP BY CustomerId
ORDER BY TotalSpent DESC
LIMIT 3;
Extracted SQL: SELECT CustomerId, SUM(Total) AS TotalSpent
FROM invoices
GROUP BY CustomerId
ORDER BY TotalSpent DESC
LIMIT 3;
SELECT CustomerId, SUM(Total) AS TotalSpent
FROM invoices
GROUP BY CustomerId
ORDER BY TotalSpent DESC
LIMIT 3;
   CustomerId  TotalSpent
0           6       49.62
1          26       47.62
2          57       46.62
Using model gpt-4o-mini for 196.5 tokens (approx)
```


Top 3 Customers by Total Spending
Total Spent
50
40
30
20
10
0
6
26
57
Customer ID

Out[40]:
```
('SELECT CustomerId, SUM(Total) AS TotalSpent \nFROM invoices \nGROUP BY CustomerId \nORDER BY TotalSpent DESC \nL
IMIT 3;',
    CustomerId  TotalSpent
 0           6       49.62
 1          26       47.62
 2          57       46.62,
 Figure({
     'data': [{'type': 'bar', 'x': array(['6', '26', '57'], dtype=object), 'y': array([49.62, 47.62, 46.62])}],
     'layout': {'template': '...',
                'title': {'text': 'Top 3 Customers by Total Spending'},
                'xaxis': {'title': {'text': 'Customer ID'}},
                'yaxis': {'title': {'text': 'Total Spent'}}}
 }))
```

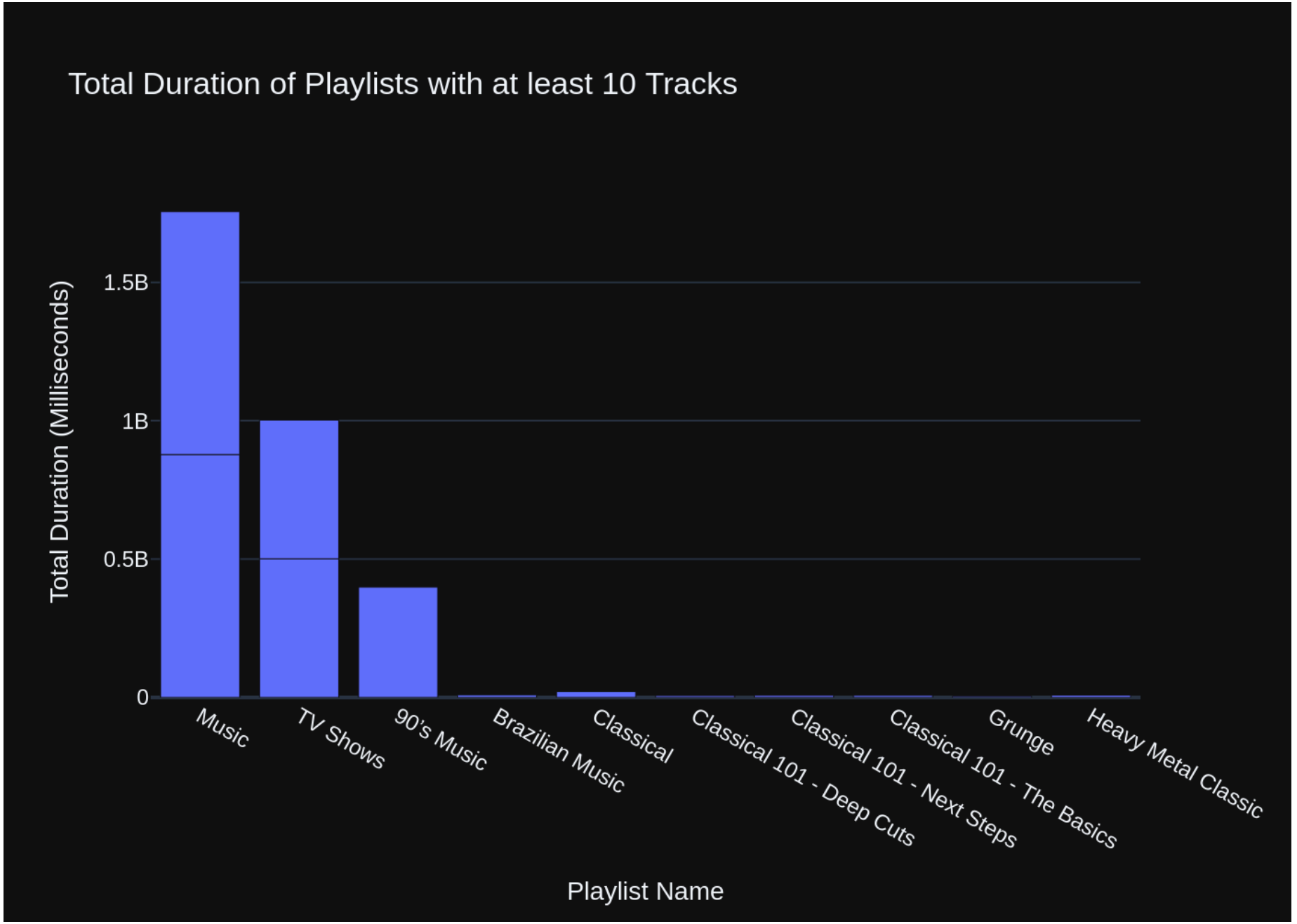
In [41]:
```
question = """
     Get all playlists containing at least 10 tracks and the total duration of those tracks:
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE TABLE "play
lists"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE T
ABLE "playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT
 PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Playlis
tId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY
 AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT
 NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INT
EGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE
 NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DE
LETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK_T
rackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE INDEX IFK
_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT
 NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REF
ERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn
the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, pleas
e generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost suffi
cient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query
to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the pr
ovided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table
(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given befor
e. \n'}, {'role': 'user', 'content': '  \n    List all genres and the number of tracks in each genre:\n'}, {'role':
'assistant', 'content': 'SELECT genres.Name AS GenreName, COUNT(tracks.TrackId) AS TrackCount \nFROM genres \nLEFT J
OIN tracks ON genres.GenreId = tracks.GenreId \nGROUP BY genres.GenreId;'}, {'role': 'user', 'content': '  \n    Get
the top 10 most popular artists (based on the number of tracks):\n'}, {'role': 'assistant', 'content': 'SELECT artis
ts.Name AS ArtistName, COUNT(tracks.TrackId) AS TrackCount \nFROM artists \nJOIN albums ON artists.ArtistId = album
s.ArtistId \nJOIN tracks ON albums.AlbumId = tracks.AlbumId \nGROUP BY artists.ArtistId \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 WHERE Name LIKE '%What%';"}, {'role': 'user', 'conten
t': '  \n     Find the top 5 customer who bought the most albums in total quantity (across all invoices):\n'}, {'rol
e': 'assistant', 'content': 'SELECT Invoice.CustomerId, SUM(InvoiceItems.Quantity) AS TotalAlbums \nFROM invoice_ite
ms AS InvoiceItems \nJOIN invoices AS Invoice ON InvoiceItems.InvoiceId = Invoice.InvoiceId \nGROUP BY Invoice.Custo
merId \nORDER BY TotalAlbums DESC \nLIMIT 5;'}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive t
racks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks ORDER BY UnitPrice DESC LIM
IT 5;'}, {'role': 'user', 'content': '  \n     Find the customer who bought the most albums in total quantity (acros
s all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT Invoice.CustomerId, SUM(InvoiceItems.Quantity) AS To
talAlbums \nFROM invoice_items AS InvoiceItems \nJOIN invoices AS Invoice ON InvoiceItems.InvoiceId = Invoice.Invoic
eId \nGROUP BY Invoice.CustomerId \nORDER BY TotalAlbums DESC \nLIMIT 1;'}, {'role': 'user', 'content': '  \n    Lis
```

```
t all albums and their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT albums.Title AS Al
bumTitle, artists.Name AS ArtistName \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 *, Total \nFROM invoices \nWHERE InvoiceDate >= '2010-01-01';"}, {'role': 'user', 'content': '
\n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WH
ERE Total > 10;'}, {'role': 'user', 'content': 'Show me a list of tables in the SQLite database'}, {'role': 'assista
nt', 'content': "SELECT name FROM sqlite_master WHERE type='table';"}, {'role': 'user', 'content': '  \n     Get all
playlists containing at least 10 tracks and the total duration of those tracks:\n'}]
Using model gpt-4o-mini for 1174.5 tokens (approx)
LLM Response: SELECT playlists.PlaylistId, playlists.Name, COUNT(playlist_track.TrackId) AS TrackCount, SUM(tracks.M
illiseconds) AS TotalDuration
FROM playlists
JOIN playlist_track ON playlists.PlaylistId = playlist_track.PlaylistId
JOIN tracks ON playlist_track.TrackId = tracks.TrackId
GROUP BY playlists.PlaylistId
HAVING TrackCount >= 10;
Extracted SQL: SELECT playlists.PlaylistId, playlists.Name, COUNT(playlist_track.TrackId) AS TrackCount, SUM(tracks.
Milliseconds) AS TotalDuration
FROM playlists
JOIN playlist_track ON playlists.PlaylistId = playlist_track.PlaylistId
JOIN tracks ON playlist_track.TrackId = tracks.TrackId
GROUP BY playlists.PlaylistId
HAVING TrackCount >= 10;
SELECT playlists.PlaylistId, playlists.Name, COUNT(playlist_track.TrackId) AS TrackCount, SUM(tracks.Milliseconds) A
S TotalDuration
FROM playlists
JOIN playlist_track ON playlists.PlaylistId = playlist_track.PlaylistId
JOIN tracks ON playlist_track.TrackId = tracks.TrackId
GROUP BY playlists.PlaylistId
HAVING TrackCount >= 10;
    PlaylistId                        Name  TrackCount  TotalDuration
0            1                       Music        3290      877683083
1            3                   TV Shows         213      501094957
2            5                  90’s Music        1477      398705153
3            8                       Music        3290      877683083
4           10                   TV Shows         213      501094957
5           11            Brazilian Music          39        9486559
6           12                   Classical          75       21770592
7           13   Classical 101 - Deep Cuts          25        6755730
8           14  Classical 101 - Next Steps          25        7575051
9           15  Classical 101 - The Basics          25        7439811
```

```
10          16                      Grunge          15        4122018
11          17         Heavy Metal Classic          26        8206312
Using model gpt-4o-mini for 271.25 tokens (approx)
```


Total Duration of Playlists with at least 10 Tracks
Total Duration (Milliseconds)
1.5B
1B
0.5B
0
Music
TV Shows
90's Music
Brazilian Music
Classical
Classical 101 - Deep Cuts
Classical 101 - Next Steps
Classical 101 - The Basics
Grunge
Heavy Metal Classic
Playlist Name

Out[41]:
```
('SELECT playlists.PlaylistId, playlists.Name, COUNT(playlist_track.TrackId) AS TrackCount, SUM(tracks.Millisecond
s) AS TotalDuration \nFROM playlists \nJOIN playlist_track ON playlists.PlaylistId = playlist_track.PlaylistId \nJ
OIN tracks ON playlist_track.TrackId = tracks.TrackId \nGROUP BY playlists.PlaylistId \nHAVING TrackCount >= 10;',
     PlaylistId                        Name  TrackCount  TotalDuration
 0            1                       Music        3290      877683083
 1            3                   TV Shows         213      501094957
 2            5                  90’s Music        1477      398705153
 3            8                       Music        3290      877683083
 4           10                   TV Shows         213      501094957
 5           11              Brazilian Music          39        9486559
 6           12                    Classical          75       21770592
 7           13   Classical 101 - Deep Cuts          25        6755730
 8           14  Classical 101 - Next Steps          25        7575051
 9           15  Classical 101 - The Basics          25        7439811
 10          16                      Grunge          15        4122018
 11          17         Heavy Metal Classic          26        8206312,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'Playlist Name=%{x}<br>Total Duration (Milliseconds)=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Music', 'TV Shows', '90’s Music', 'Music', 'TV Shows',
                           'Brazilian Music', 'Classical', 'Classical 101 - Deep Cuts',
                           'Classical 101 - Next Steps', 'Classical 101 - The Basics', 'Grunge',
                           'Heavy Metal Classic'], dtype=object),
               'xaxis': 'x',
               'y': array([877683083, 501094957, 398705153, 877683083, 501094957,   9486559,
                            21770592,   6755730,   7575051,   7439811,   4122018,   8206312]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Total Duration of Playlists with at least 10 Tracks'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Playlist Name'}},
```

```
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total Duration (Millisecond
s)'}}}
 }))
```

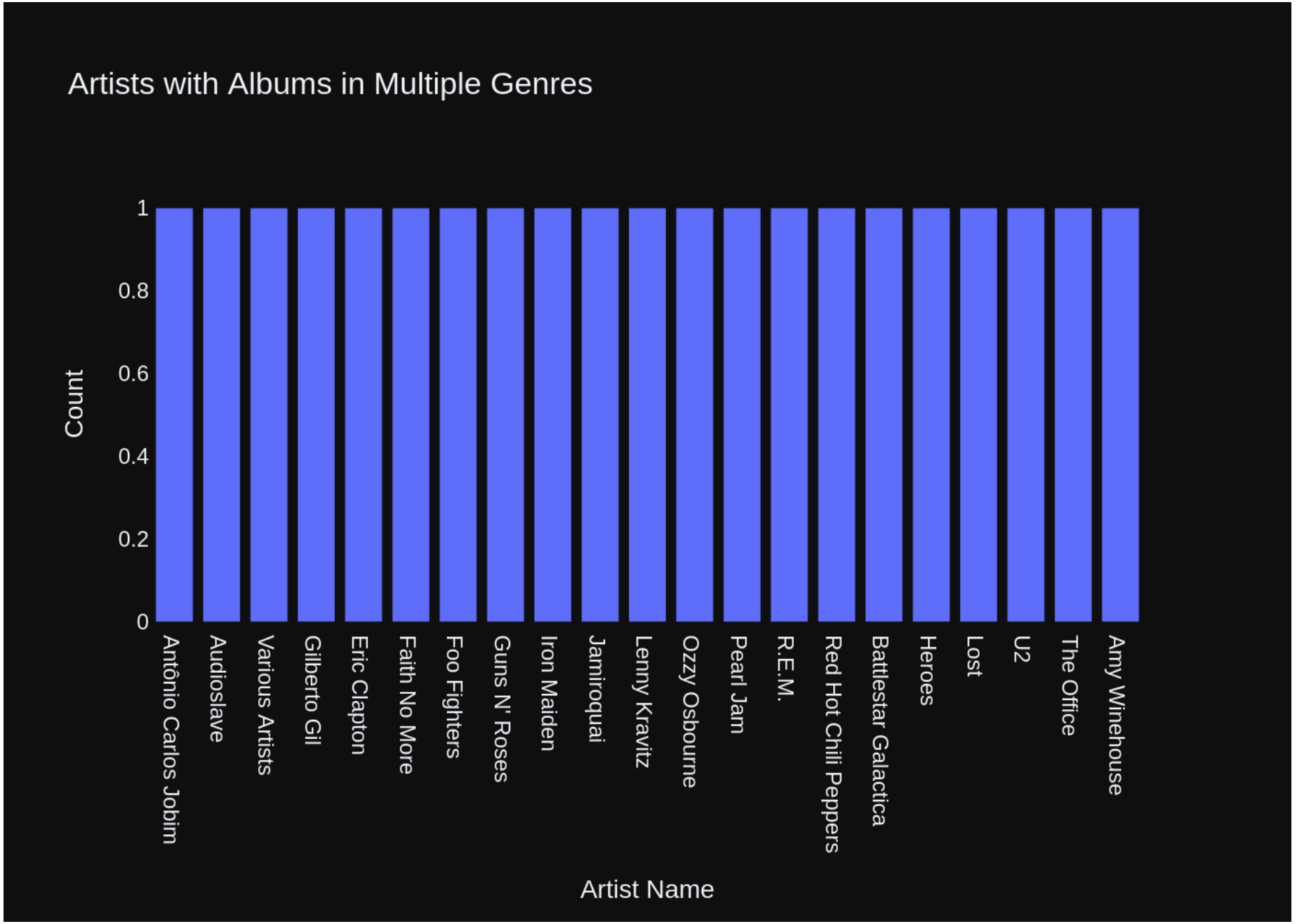
In [42]:
```
question = """
     Identify artists who have albums with tracks appearing in multiple genres:
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer t
he question. Your response should ONLY be based on the given context and follow the response guidelines and format i
nstructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGE
R,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUME
RIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks"
(GenreId)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGE
R PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n
FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREAT
E INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE TABLE "genres"\r\n(\r\n    GenreId INTEGER PRIMARY
KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_tra
ck" (TrackId)\n\nCREATE TABLE "artists"\r\n(\r\n    ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name
NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guideli
nes \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the q
uestion. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particul
ar column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query
with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be
generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, plea
se repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': '  \n    Get the top 10 most
popular artists (based on the number of tracks):\n'}, {'role': 'assistant', 'content': 'SELECT artists.Name AS Artis
tName, COUNT(tracks.TrackId) AS TrackCount \nFROM artists \nJOIN albums ON artists.ArtistId = albums.ArtistId \nJOIN
tracks ON albums.AlbumId = tracks.AlbumId \nGROUP BY artists.ArtistId \nORDER BY TrackCount DESC \nLIMIT 10;'}, {'ro
le': 'user', 'content': '  \n    List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant',
'content': 'SELECT genres.Name AS GenreName, COUNT(tracks.TrackId) AS TrackCount \nFROM genres \nLEFT JOIN tracks ON
genres.GenreId = tracks.GenreId \nGROUP BY genres.GenreId;'}, {'role': 'user', 'content': '  \n    List all albums a
nd their corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT albums.Title AS AlbumTitle, arti
sts.Name AS ArtistName \nFROM albums \nJOIN artists ON albums.ArtistId = artists.ArtistId;'}, {'role': 'user', 'cont
ent': '  \n     Get all playlists containing at least 10 tracks and the total duration of those tracks:\n'}, {'rol
e': 'assistant', 'content': 'SELECT playlists.PlaylistId, playlists.Name, COUNT(playlist_track.TrackId) AS TrackCoun
t, SUM(tracks.Milliseconds) AS TotalDuration \nFROM playlists \nJOIN playlist_track ON playlists.PlaylistId = playli
st_track.PlaylistId \nJOIN tracks ON playlist_track.TrackId = tracks.TrackId \nGROUP BY playlists.PlaylistId \nHAVIN
G TrackCount >= 10;'}, {'role': 'user', 'content': '  \n     Find the customer who bought the most albums in total q
uantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT Invoice.CustomerId, SUM(InvoiceItems.Q
uantity) AS TotalAlbums \nFROM invoice_items AS InvoiceItems \nJOIN invoices AS Invoice ON InvoiceItems.InvoiceId =
Invoice.InvoiceId \nGROUP BY Invoice.CustomerId \nORDER BY TotalAlbums DESC \nLIMIT 1;'}, {'role': 'user', 'conten
t': '  \n     Find the top 5 customer who bought the most albums in total quantity (across all invoices):\n'}, {'rol
e': 'assistant', 'content': 'SELECT Invoice.CustomerId, SUM(InvoiceItems.Quantity) AS TotalAlbums \nFROM invoice_ite
ms AS InvoiceItems \nJOIN invoices AS Invoice ON InvoiceItems.InvoiceId = Invoice.InvoiceId \nGROUP BY Invoice.Custo
```

```
merId \nORDER BY TotalAlbums DESC \nLIMIT 5;'}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive t
racks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks ORDER BY UnitPrice DESC LIM
IT 5;'}, {'role': 'user', 'content': '  \n    Find all tracks with a name containing "What" (case-insensitive)\n'},
{'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%';"}, {'role': 'user', 'content': 'Sho
w me a list of tables in the SQLite database'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHE
RE type='table';"}, {'role': 'user', 'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'role':
'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10;'}, {'role': 'user', 'content': '  \n     Identify
artists who have albums with tracks appearing in multiple genres:\n'}]
Using model gpt-4o-mini for 1175.5 tokens (approx)
LLM Response: SELECT DISTINCT artists.Name AS ArtistName
FROM artists
JOIN albums ON artists.ArtistId = albums.ArtistId
JOIN tracks ON albums.AlbumId = tracks.AlbumId
GROUP BY artists.ArtistId
HAVING COUNT(DISTINCT tracks.GenreId) > 1;
Extracted SQL: SELECT DISTINCT artists.Name AS ArtistName
FROM artists
JOIN albums ON artists.ArtistId = albums.ArtistId
JOIN tracks ON albums.AlbumId = tracks.AlbumId
GROUP BY artists.ArtistId
HAVING COUNT(DISTINCT tracks.GenreId) > 1;
SELECT DISTINCT artists.Name AS ArtistName
FROM artists
JOIN albums ON artists.ArtistId = albums.ArtistId
JOIN tracks ON albums.AlbumId = tracks.AlbumId
GROUP BY artists.ArtistId
HAVING COUNT(DISTINCT tracks.GenreId) > 1;
               ArtistName
0    Antônio Carlos Jobim
1              Audioslave
2         Various Artists
3            Gilberto Gil
4            Eric Clapton
5           Faith No More
6            Foo Fighters
7           Guns N' Roses
8             Iron Maiden
9              Jamiroquai
10          Lenny Kravitz
11          Ozzy Osbourne
12              Pearl Jam
```

```
13                R.E.M.
14  Red Hot Chili Peppers
15   Battlestar Galactica
16                Heroes
17                  Lost
18                    U2
19            The Office
20          Amy Winehouse
Using model gpt-4o-mini for 222.5 tokens (approx)
```


Artists with Albums in Multiple Genres
Count
1
0.8
0.6
0.4
0.2
0
Antônio Carlos Jobim
Audioslave
Various Artists
Gilberto Gil
Eric Clapton
Faith No More
Foo Fighters
Guns N' Roses
Iron Maiden
Jamiroquai
Lenny Kravitz
Ozzy Osbourne
Pearl Jam
R.E.M.
Red Hot Chili Peppers
Battlestar Galactica
Heroes
Lost
U2
The Office
Amy Winehouse
Artist Name

Out[42]:
```
('SELECT DISTINCT artists.Name AS ArtistName \nFROM artists \nJOIN albums ON artists.ArtistId = albums.ArtistId \n
JOIN tracks ON albums.AlbumId = tracks.AlbumId \nGROUP BY artists.ArtistId \nHAVING COUNT(DISTINCT tracks.GenreId)
> 1;',
                ArtistName
 0    Antônio Carlos Jobim
 1              Audioslave
 2         Various Artists
 3            Gilberto Gil
 4            Eric Clapton
 5           Faith No More
 6            Foo Fighters
 7           Guns N' Roses
 8             Iron Maiden
 9              Jamiroquai
 10          Lenny Kravitz
 11          Ozzy Osbourne
 12              Pearl Jam
 13                 R.E.M.
 14  Red Hot Chili Peppers
 15   Battlestar Galactica
 16                 Heroes
 17                   Lost
 18                     U2
 19             The Office
 20          Amy Winehouse,
 Figure({
     'data': [{'type': 'bar',
               'x': array(['Antônio Carlos Jobim', 'Audioslave', 'Various Artists', 'Gilberto Gil',
                           'Eric Clapton', 'Faith No More', 'Foo Fighters', "Guns N' Roses",
                           'Iron Maiden', 'Jamiroquai', 'Lenny Kravitz', 'Ozzy Osbourne',
                           'Pearl Jam', 'R.E.M.', 'Red Hot Chili Peppers', 'Battlestar Galactica',
                           'Heroes', 'Lost', 'U2', 'The Office', 'Amy Winehouse'], dtype=object),
               'y': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]}],
     'layout': {'template': '...',
                'title': {'text': 'Artists with Albums in Multiple Genres'},
                'xaxis': {'title': {'text': 'Artist Name'}},
                'yaxis': {'showgrid': False, 'title': {'text': 'Count'}, 'zeroline': False}}
 }))
```

Check completion time

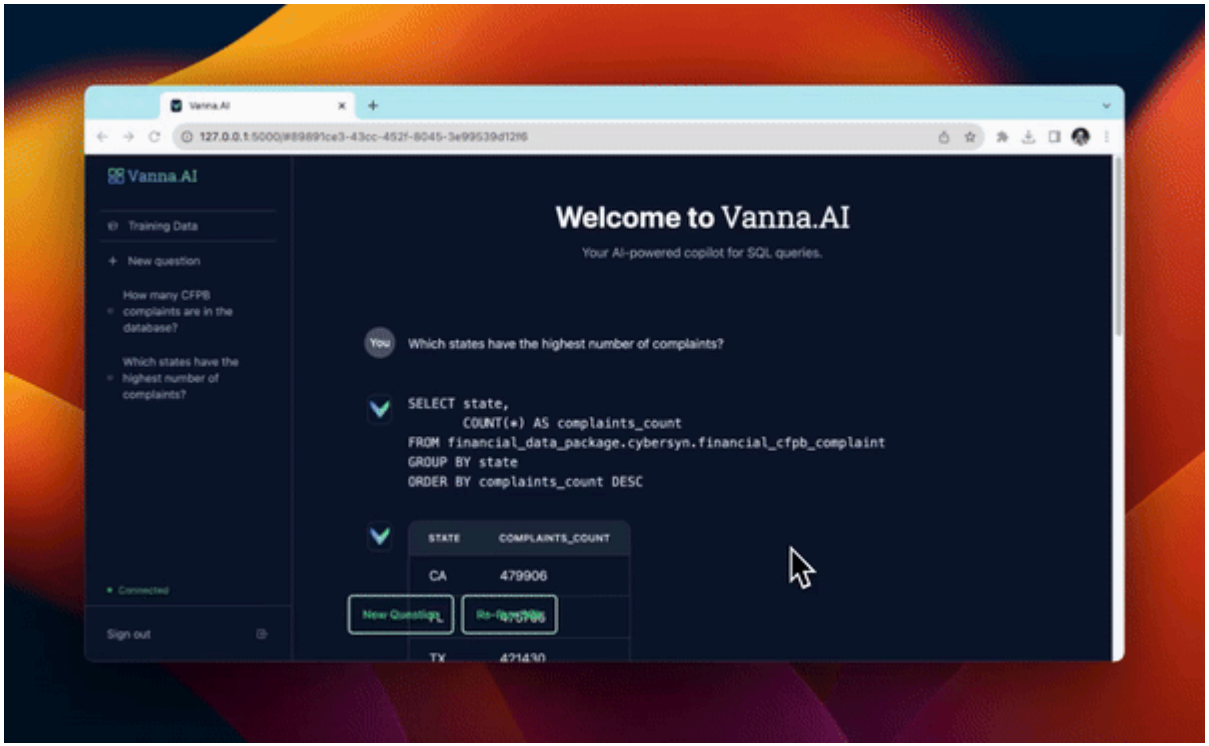
In [43]:
```
ts_stop = time()

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 'gpt-4o-mini' LLM took : 73.75 sec
```

In [44]:
```
from datetime import datetime
print(datetime.now())
```

```
2024-07-21 20:59:35.225156
```

Launch the User Interface



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

Next Steps

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

- Streamlit app
- Flask app
- Slackbot