Generating SQL for SQLite using Ollama, ChromaDB

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

Which LLM do you want to use?

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

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

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

Setup

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

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

```
warnings.filterwarnings('ignore', category=DeprecationWarning, message='^Num
# warnings.filterwarnings('ignore', category=DeprecationWarning, message=re.

import os

import re
from time import time

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

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

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

clean_and_train = True  # False
```

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

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

```
Hostname: ducklover1
```

In [6]:
```
file_db = os.path.abspath(os.path.expanduser(file_db))
vn.connect_to_sqlite(file_db)
```

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

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

In [8]:
```
def remove_collections(collection_name=None, ACCEPTED_TYPES = ["sql", "ddl",
    if not collection_name:
        collections = ACCEPTED_TYPES
    elif isinstance(collection_name, str):
        collections = [collection_name]
    elif isinstance(collection_name, list):
        collections = collection_name
    else:
        print(f"\t{collection_name} is unknown: Skipped")
        return

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

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

In [9]:
```
def strip_brackets(ddl):
    """
    This function removes square brackets from table and column names in a D

    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 b
    return re.sub(pattern, r"\1", ddl)
```

In [10]:
```
if clean_and_train:
    remove_collections()
```

Training

SQLite sample database

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

In [11]:
```
df_ddl = vn.run_sql("SELECT type, sql FROM sqlite_master WHERE sql is not nu
```

In [12]:
```
df_ddl
```

Out[12]:

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

In [13]:

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

    # Sometimes you may want to add documentation about your business termin
    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" (SupportRep
Id)
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" (Invoic
eId)
Adding ddl: CREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)
Adding ddl: CREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (Track
Id)
Adding ddl: CREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)
Adding ddl: CREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)
Adding ddl: CREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)
Adding ddl: CREATE TABLE sqlite_stat1(tbl,idx,stat)
Adding documentation....
```

In [14]:
```
# show training data
training_data = vn.get_training_data()
training_data
```

Out[14]:

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

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

Asking the AI

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

In [15]:
```
ts_start = time()
```

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

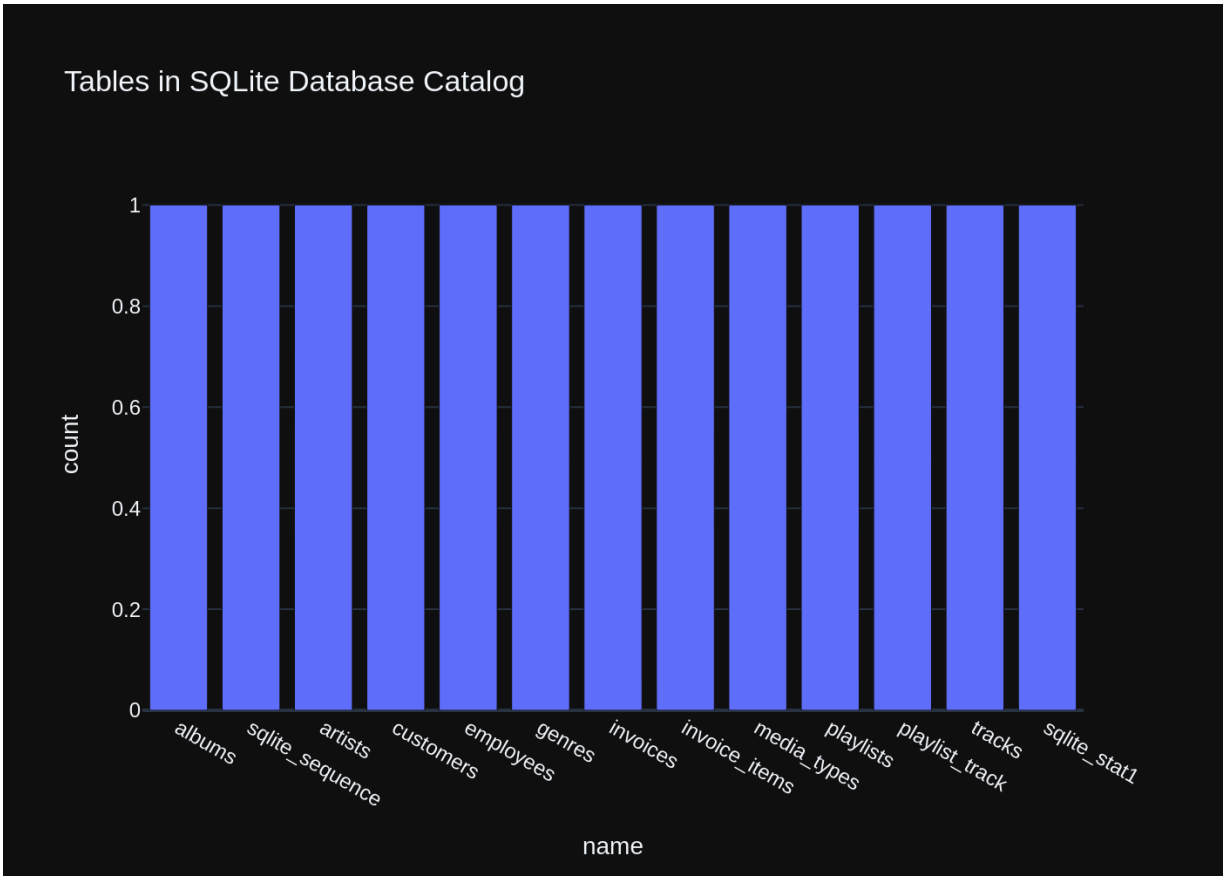
In [16]:
```
vn.ask(question="Can you list all tables in the SQLite database catalog?")
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please
help to generate a SQL query to answer the question. Your response should ON
LY be based on the given context and follow the response guidelines and form
at instructions. \n===Tables \nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCR
EATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE "playlists"\r\n(\r\n
PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR
(120)\r\n)\n\nCREATE TABLE "genres"\r\n(\r\n    GenreId INTEGER PRIMARY KEY
AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "trac
ks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    N
ame NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTE
GER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n
Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMER
IC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Genr
eId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaType
Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "me
dia_types"\r\n(\r\n    MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "artists"\r\n(\r\n    Arti
stId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r
\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMAR
Y KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    Tra
ckId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Qua
ntity INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES "invoice
s" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREI
GN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION O
N 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 P
K_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (Play
listId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON U
PDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "album
s"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Ti
tle NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREI
GN 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\nIn the chinook dat
abase invoice means order\n\n===Response Guidelines \n1. If the provided con
text is sufficient, please generate a valid SQL query without any explanatio
ns for the question. \n2. If the provided context is almost sufficient but r
equires knowledge of a specific string in a particular column, please genera
te an intermediate SQL query to find the distinct strings in that column. Pr
epend the query with a comment saying intermediate_sql \n3. If the provided
context is insufficient, please explain why it can\'t be generated. \n4. Ple
ase use the most relevant table(s). \n5. If the question has been asked and
answered before, please repeat the answer exactly as it was given before.
\n'}, {'role': 'user', 'content': 'Can you list all tables in the SQLite dat
abase catalog?'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE TABLE s
```

qlite_sequence(name,seq)\n\nCREATE TABLE \"playlists\"\r\n(\r\n   PlaylistI
d 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 AUTOINCR
EMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"tracks\"\r
\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name N
VARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER
NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Mill
iseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(1
0,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreI
d) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTy
peId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE
\"media_types\"\r\n(\r\n   MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NO
T NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"artists\"\r\n(\r\n
ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(12
0)\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 \"invo
ices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   F
OREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO AC
TION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n
PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONST
RAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KE
Y (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO AC
TION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tracks\"
(TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TAB
LE \"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 NUL
L,\r\n   FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nI
n the chinook database invoice means order\n\n===Response Guidelines \n1. If
the provided context is sufficient, please generate a valid SQL query withou
t any explanations for the question. \n2. If the provided context is almost
sufficient but requires knowledge of a specific string in a particular colum
n, please generate an intermediate SQL query to find the distinct strings in
that column. Prepend the query with a comment saying intermediate_sql \n3. I
f the provided context is insufficient, please explain why it can't be gener
ated. \n4. Please use the most relevant table(s). \n5. If the question has b
een asked and answered before, please repeat the answer exactly as it was gi
ven before. \n"}, {"role": "user", "content": "Can you list all tables in th
e SQLite database catalog?"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:32:48.982498268
Z', 'message': {'role': 'assistant', 'content': "```sql\nSELECT name FROM sq
lite_master WHERE type = 'table';\n```"}, 'done_reason': 'stop', 'done': Tru
e, 'total_duration': 37462553880, 'load_duration': 4751495996, 'prompt_eval_
count': 866, 'prompt_eval_duration': 29016681000, 'eval_count': 18, 'eval_du
ration': 3557832000}
LLM Response: ```sql
SELECT name FROM sqlite_master WHERE type = 'table';
```
Info: Output from LLM: ```sql
SELECT name FROM sqlite_master WHERE type = 'table';
```

```
Extracted SQL: SELECT name FROM sqlite_master WHERE type = 'table'
SELECT name FROM sqlite_master WHERE type = 'table'
              name
0           albums
1   sqlite_sequence
2          artists
3        customers
4        employees
5           genres
6         invoices
7     invoice_items
8       media_types
9         playlists
10   playlist_track
11          tracks
12     sqlite_stat1
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: 'Ca
n you list all tables in the SQLite database catalog?'\n\nThe DataFrame was
produced using this query: SELECT name FROM sqlite_master WHERE type = 'tabl
e'\n\nThe following is information about the resulting pandas DataFrame 'd
f': \nRunning df.dtypes gives:\n name    object\ndtype: object"}, {"role":
"user", "content": "Can you generate the Python plotly code to chart the res
ults of the dataframe? Assume the data is in a pandas dataframe called 'df'.
If there is only one value in the dataframe, use an Indicator. Respond with
only Python code. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:33:02.117758388
Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.ex
press as px\n\nfig = px.bar(df, x='name', title='Tables in SQLite Database C
atalog')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_d
uration': 13102639880, 'load_duration': 22239056, 'prompt_eval_count': 167,
'prompt_eval_duration': 5716657000, 'eval_count': 38, 'eval_duration': 72318
64000}
```


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

Out[16]:
```
("SELECT name FROM sqlite_master WHERE type = 'table'",
                 name
 0             albums
 1   sqlite_sequence
 2            artists
 3          customers
 4          employees
 5             genres
 6           invoices
 7     invoice_items
 8       media_types
 9         playlists
 10   playlist_track
 11           tracks
 12     sqlite_stat1,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'name=%{x}<br>count=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['albums', 'sqlite_sequence', 'artists', 'custome
rs', 'employees',
                           'genres', 'invoices', 'invoice_items', 'media_ty
pes', 'playlists',
                           'playlist_track', 'tracks', 'sqlite_stat1'], dty
pe=object),
               'xaxis': 'x',
               'y': array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Tables in SQLite Database Catalog'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'name'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'count'}}}
 }))
```

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

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

SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "employees"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE "playlists"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE TABLE "albums"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "media_types"\r\n(\r\n   MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': 'Can you list all tables

```
in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT
name FROM sqlite_master WHERE type = 'table'"}, {'role': 'user', 'content':
"which table stores customer's orders"}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGER P
RIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n
InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    B
illingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCou
ntry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMER
IC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\"
(CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE
TABLE \"invoice_items\"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOIN
CREMENT 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 (TrackI
d) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n    CustomerId INTEGER PRI
MARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r
\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    A
ddress NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r
\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVA
RCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n
SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES \"employe
es\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCR
EATE TABLE \"employees\"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVA
RCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r
\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR
(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NV
ARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n
Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo)
REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE
NO ACTION\r\n)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE \"p
laylists\"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)
\n\nCREATE TABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGE
R  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 \"playl
ist_track\"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER
NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, Track
Id),\r\n    FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackI
d) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION\r\n)\n\nCREATE TABLE \"media_types\"\r\n(\r\n    MediaTypeId INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n\n===A
dditional Context \n\nIn the chinook database invoice means order\n\n===Resp
onse Guidelines \n1. If the provided context is sufficient, please generate
```

```
a valid SQL query without any explanations for the question. \n2. If the pro
vided context is almost sufficient but requires knowledge of a specific stri
ng in a particular column, please generate an intermediate SQL query to find
the distinct strings in that column. Prepend the query with a comment saying
intermediate_sql \n3. If the provided context is insufficient, please explai
n why it can't be generated. \n4. Please use the most relevant table(s). \n
5. If the question has been asked and answered before, please repeat the ans
wer exactly as it was given before. \n"}, {"role": "user", "content": "Can y
ou list all tables in the SQLite database catalog?"}, {"role": "assistant",
"content": "SELECT name FROM sqlite_master WHERE type = 'table'"}, {"role":
"user", "content": "which table stores customer's orders"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:33:47.432825333
Z', 'message': {'role': 'assistant', 'content': "The table that stores custo
mer's orders is called invoices."}, 'done_reason': 'stop', 'done': True, 'to
tal_duration': 44890486955, 'load_duration': 22128789, 'prompt_eval_count':
1145, 'prompt_eval_duration': 42038440000, 'eval_count': 13, 'eval_duratio
n': 2594205000}
LLM Response: The table that stores customer's orders is called invoices.
The table that stores customer's orders is called invoices.
Couldn't run sql:  Execution failed on sql 'The table that stores customer's
orders is called invoices.': near "The": syntax error
```

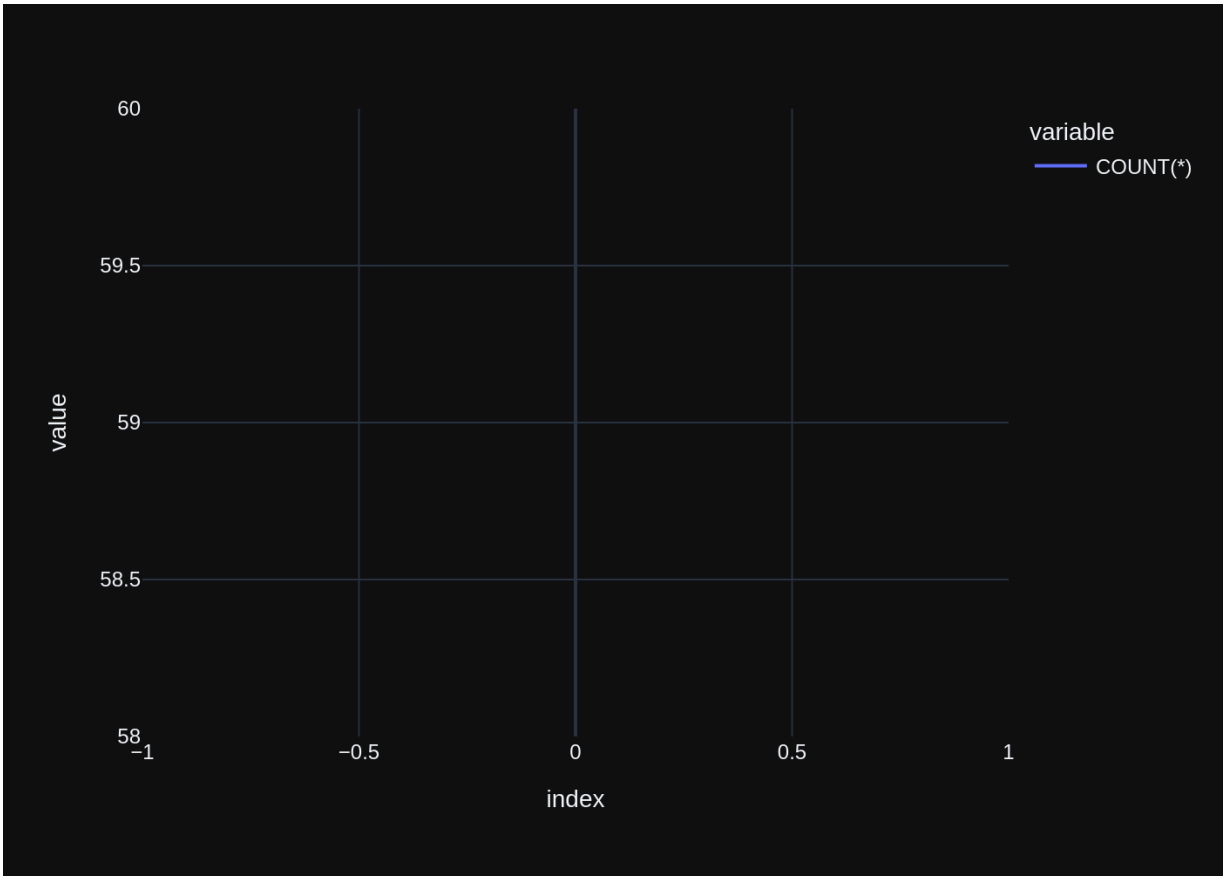
In [18]:
```
vn.ask(question="How many customers are there")
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please
help to generate a SQL query to answer the question. Your response should ON
LY be based on the given context and follow the response guidelines and form
at instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId
INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT N
ULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(7
0),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n
BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    T
otal NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "cu
stomers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n
\nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCRE
ATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREME
NT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCH
AR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR(7
0),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVAR
CHAR(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 I
NTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_
InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE TABLE "invoice_item
s"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPr
ice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREI
GN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO AC
TION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks"
(TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IND
EX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE TABLE
"albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOR
EIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACT
ION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invo
ice_items" (TrackId)\n\nCREATE TABLE "employees"\r\n(\r\n    EmployeeId INTE
GER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NU
LL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n
ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n
Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r
\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVA
RCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIG
N KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO A
CTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlists"\r\n(\r\n    Playl
istId 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 o
rder\n\n===Response Guidelines \n1. If the provided context is sufficient, p
lease generate a valid SQL query without any explanations for the question.
\n2. If the provided context is almost sufficient but requires knowledge of
a specific string in a particular column, please generate an intermediate SQ
L query to find the distinct strings in that column. Prepend the query with
a comment saying intermediate_sql \n3. If the provided context is insufficie
nt, please explain why it can\'t be generated. \n4. Please use the most rele
vant 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': 'Can you list all tables in the SQLite database catalog?'}, {'rol
e': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type = 'ta
ble'"}, {'role': 'user', 'content': 'How many customers are there'}]
Info: Ollama parameters:
model=codegemma:latest,
```

```
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGER P
RIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n
InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    B
illingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCou
ntry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMER
IC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\"
(CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE
INDEX IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TAB
LE \"customers\"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NO
T 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 NVARCHAR(40),\r\n    Country NVARCHAR(40),\r
\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHA
R(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n
FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustome
rId ON \"invoices\" (CustomerId)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n
InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId I
NTEGER  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 (Inv
oiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON U
PDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_
InvoiceLineInvoiceId ON \"invoice_items\" (InvoiceId)\n\nCREATE TABLE \"albu
ms\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOR
EIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO A
CTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"i
nvoice_items\" (TrackId)\n\nCREATE TABLE \"employees\"\r\n(\r\n    EmployeeI
d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)
NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(3
0),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    HireDate DAT
ETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State N
VARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r
\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(6
0),\r\n    FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlis
ts\"\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 databa
se invoice means order\n\n===Response Guidelines \n1. If the provided contex
t is sufficient, please generate a valid SQL query without any explanations
for the question. \n2. If the provided context is almost sufficient but requ
ires knowledge of a specific string in a particular column, please generate
an intermediate SQL query to find the distinct strings in that column. Prepe
nd the query with a comment saying intermediate_sql \n3. If the provided con
text 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 answ
ered before, please repeat the answer exactly as it was given before. \n"},
{"role": "user", "content": "Can you list all tables in the SQLite database
catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master
```

```
WHERE type = 'table'"}, {"role": "user", "content": "How many customers are
there"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:34:21.880782336
Z', 'message': {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM custom
ers'}, 'done_reason': 'stop', 'done': True, 'total_duration': 34405765267,
'load_duration': 21471816, 'prompt_eval_count': 1054, 'prompt_eval_duratio
n': 31856098000, 'eval_count': 12, 'eval_duration': 2328738000}
LLM Response: SELECT COUNT(*) FROM customers
SELECT COUNT(*) FROM customers
   COUNT(*)
0        59
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: 'Ho
w many customers are there'\n\nThe DataFrame was produced using this query:
SELECT COUNT(*) FROM customers\n\nThe following is information about the res
ulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n COUNT(*)    int64
\ndtype: object"}, {"role": "user", "content": "Can you generate the Python
plotly code to chart the results of the dataframe? Assume the data is in a p
andas dataframe called 'df'. If there is only one value in the dataframe, us
e an Indicator. Respond with only Python code. Do not answer with any explan
ations -- just the code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:34:37.976535947
Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.ex
press as px\n\nfig = px.indicator(\n    df,\n    value='COUNT(*)',\n    titl
e='Number of Customers',\n    gauge={'axis': {'visible': True}},\n)\n\nfig.s
how()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 1607510
3075, 'load_duration': 18877263, 'prompt_eval_count': 158, 'prompt_eval_dura
tion': 5202186000, 'eval_count': 56, 'eval_duration': 10810960000}
```



Out[18]:
```
('SELECT COUNT(*) FROM customers',
    COUNT(*)
 0        59,
 Figure({
     'data': [{'hovertemplate': 'variable=COUNT(*)<br>index=%{x}<br>value=%
{y}<extra></extra>',
               'legendgroup': 'COUNT(*)',
               'line': {'color': '#636efa', 'dash': 'solid'},
               'marker': {'symbol': 'circle'},
               'mode': 'lines',
               'name': 'COUNT(*)',
               'orientation': 'v',
               'showlegend': True,
               'type': 'scatter',
               'x': array([0]),
               'xaxis': 'x',
               'y': array([59]),
               'yaxis': 'y'}],
     'layout': {'legend': {'title': {'text': 'variable'}, 'tracegroupgap':
0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'index'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'value'}}}
 }))
```

In [ ]:

In [19]:
```
vn.ask(question="what are the top 5 countries that customers come from?")
```

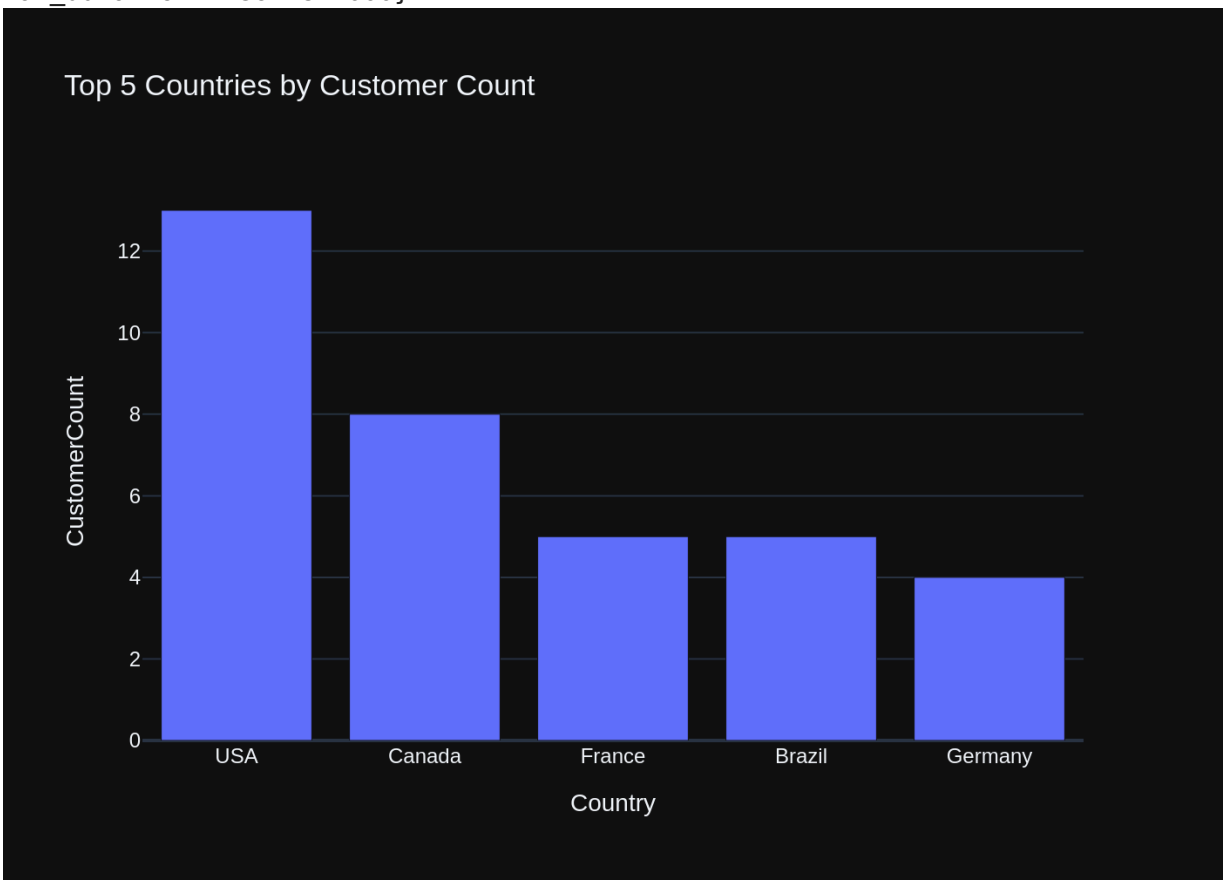
```
Number of requested results 10 is greater than number of elements in index
2, updating n_results = 2
Number of requested results 10 is greater than number of elements in index
1, updating n_results = 1
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please
help to generate a SQL query to answer the question. Your response should ON
LY be based on the given context and follow the response guidelines and form
at instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId
INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT N
ULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(7
0),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n
BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    T
otal NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "cu
stomers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n
\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName N
VARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR
(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NV
ARCHAR(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 I
NTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "inv
oice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT
NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NUL
L,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NU
LL,\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 "media_types"\r\n(\r\n    MediaTypeId INTEGER PRIMARY KE
Y AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK
_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "employe
es"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r
\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIM
E,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCH
AR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    Posta
lCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n
Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES "employees" (E
mployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TA
BLE "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 NUL
L,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_track"\r
\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r
\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n
FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELE
TE 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\nCRE
ATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE "tracks"\r\n(\r\n    Tra
ckId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)
NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n
GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER
NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r
\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE
NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genr
es" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIG
N KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE
NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chi
nook database invoice means order\n\n===Response Guidelines \n1. If the prov
ided context is sufficient, please generate a valid SQL query without any ex
```

planations for the question. \n2. If the provided context is almost sufficie
nt but requires knowledge of a specific string in a particular column, pleas
e generate an intermediate SQL query to find the distinct strings in that co
lumn. Prepend the query with a comment saying intermediate_sql \n3. If the p
rovided context is insufficient, please explain why it can\'t be generated.
\n4. Please use the most relevant table(s). \n5. If the question has been as
ked and answered before, please repeat the answer exactly as it was given be
fore. \n'}, {'role': 'user', 'content': 'How many customers are there'}, {'r
ole': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'u
ser', 'content': 'Can you list all tables in the SQLite database catalog?'},
{'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type
= 'table'"}, {'role': 'user', 'content': 'what are the top 5 countries that
customers come from?'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n   InvoiceId INTEGER P
RIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n
InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   B
illingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCou
ntry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMER
IC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES \"customers\"
(CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE
TABLE \"customers\"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AUTOINCREMENT
NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   LastName NVARCHAR
(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r
\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR
(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax
NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEG
ER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoi
ce_items\"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT N
ULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r
\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NUL
L,\r\n   FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t
\tON DELETE NO ACTION ON UPDATE NO 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 \"media_types\"\r\n(\r\n   MediaTypeId INTEGER PRIMAR
Y KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX
IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"e
mployees\"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NO
T NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDat
e DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   Ci
ty NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n
PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(2
4),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES \"e
mployees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE TABLE \"albums\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGE
R 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 \"playl
ist_track\"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER
NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, Track
Id),\r\n    FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackI
d) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION\r\n)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE \"trac
ks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INT
EGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n
Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMER
IC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\" (Albu
mId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Ge
nreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE
NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (Medi
aTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additi
onal Context \n\nIn the chinook database invoice means order\n\n===Response
Guidelines \n1. If the provided context is sufficient, please generate a val
id SQL query without any explanations for the question. \n2. If the provided
context is almost sufficient but requires knowledge of a specific string in
a particular column, please generate an intermediate SQL query to find the d
istinct strings in that column. Prepend the query with a comment saying inte
rmediate_sql \n3. If the provided context is insufficient, please explain wh
y it can't be generated. \n4. Please use the most relevant table(s). \n5. If
the question has been asked and answered before, please repeat the answer ex
actly as it was given before. \n"}, {"role": "user", "content": "How many cu
stomers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM
customers"}, {"role": "user", "content": "Can you list all tables in the SQL
ite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM
sqlite_master WHERE type = 'table'"}, {"role": "user", "content": "what are
the top 5 countries that customers come from?"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:35:35.435322534
Z', 'message': {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) as
CustomerCount FROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC
\nLIMIT 5'}, 'done_reason': 'stop', 'done': True, 'total_duration': 57349596
283, 'load_duration': 22795591, 'prompt_eval_count': 1347, 'prompt_eval_dura
tion': 50366736000, 'eval_count': 31, 'eval_duration': 6653491000}
LLM Response: SELECT Country, COUNT(*) as CustomerCount FROM customers
GROUP BY Country
ORDER BY CustomerCount DESC
LIMIT 5
SELECT Country, COUNT(*) as CustomerCount FROM customers
GROUP BY Country
ORDER BY CustomerCount DESC
LIMIT 5
   Country  CustomerCount
0      USA             13
1   Canada              8
2   France              5
3   Brazil              5
4  Germany              4
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
```

```
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: 'wh
at are the top 5 countries that customers come from?'\n\nThe DataFrame was p
roduced using this query: SELECT Country, COUNT(*) as CustomerCount FROM cus
tomers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5\n\nThe follow
ing is information about the resulting pandas DataFrame 'df': \nRunning df.d
types gives:\n Country          object\nCustomerCount     int64\ndtype: obje
ct"}, {"role": "user", "content": "Can you generate the Python plotly code t
o chart the results of the dataframe? Assume the data is in a pandas datafra
me called 'df'. If there is only one value in the dataframe, use an Indicato
r. Respond with only Python code. Do not answer with any explanations -- jus
t the code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:35:50.781186137
Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.ex
press as px\n\nfig = px.bar(df, x='Country', y='CustomerCount', title='Top 5
Countries by Customer Count')\nfig.show()\n```"}, 'done_reason': 'stop', 'do
ne': True, 'total_duration': 14946732708, 'load_duration': 24799698, 'prompt
_eval_count': 187, 'prompt_eval_duration': 6203713000, 'eval_count': 45, 'ev
al_duration': 8627871000}
```



Out[19]:
```
('SELECT Country, COUNT(*) as CustomerCount FROM customers\nGROUP BY Countr
y\nORDER BY CustomerCount DESC\nLIMIT 5',
    Country  CustomerCount
 0      USA             13
 1   Canada              8
 2   France              5
 3   Brazil              5
 4  Germany              4,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'Country=%{x}<br>CustomerCount=%{y}<extra>
</extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['USA', 'Canada', 'France', 'Brazil', 'Germany'],
dtype=object),
               'xaxis': 'x',
               'y': array([13,  8,  5,  5,  4]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Top 5 Countries by Customer Count'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Country'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'CustomerCount'}}}
 }))
```

More SQL questions

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

In [20]:
```
question = """
    List all albums and their corresponding artist names
"""

vn.ask(question=question)
```

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

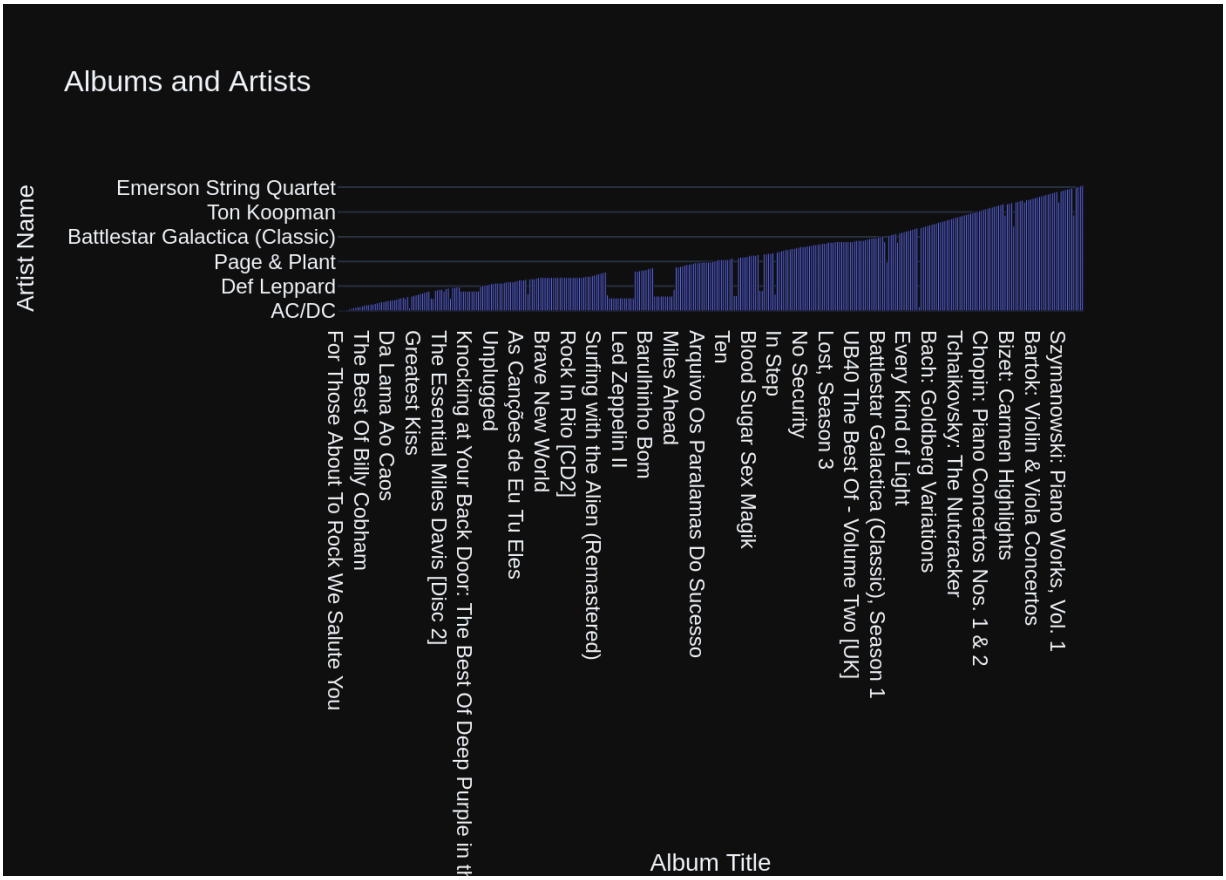
```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please
help to generate a SQL query to answer the question. Your response should ON
LY be based on the given context and follow the response guidelines and form
at instructions. \n===Tables \nCREATE INDEX IFK_AlbumArtistId ON "albums" (A
rtistId)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AU
TOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId
INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (Artis
tId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "t
racks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INT
EGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n
Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMER
IC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Genr
eId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaType
Id) \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    Ar
tistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)
\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX
IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE TABLE "play
lists"\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 I
NTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n
\nCREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\n\n===Additi
onal Context \n\nIn the chinook database invoice means order\n\n===Response
Guidelines \n1. If the provided context is sufficient, please generate a val
id SQL query without any explanations for the question. \n2. If the provided
context is almost sufficient but requires knowledge of a specific string in
a particular column, please generate an intermediate SQL query to find the d
istinct strings in that column. Prepend the query with a comment saying inte
rmediate_sql \n3. If the provided context is insufficient, please explain wh
y it can\'t be generated. \n4. Please use the most relevant table(s). \n5. I
f the question has been asked and answered before, please repeat the answer
exactly as it was given before. \n'}, {'role': 'user', 'content': 'Can you l
ist all tables in the SQLite database catalog?'}, {'role': 'assistant', 'con
tent': "SELECT name FROM sqlite_master WHERE type = 'table'"}, {'role': 'use
r', 'content': 'what are the top 5 countries that customers come from?'},
{'role': 'assistant', 'content': 'SELECT Country, COUNT(*) as CustomerCount
FROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'},
{'role': 'user', 'content': 'How many customers are there'}, {'role': 'assis
tant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'conte
nt': '  \n    List all albums and their corresponding artist names  \n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE INDEX IFK_AlbumArtistId ON \"albums\" (ArtistId)\n
\nCREATE TABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREM
ENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER
NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks
```

```
\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Na
me NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEG
ER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n
Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMER
IC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"albums\" (Albu
mId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Ge
nreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE
NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (Medi
aTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDE
X IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"artists\"\r\n
(\r\n    ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NV
ARCHAR(120)\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (GenreId)\n
\nCREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\n\nC
REATE TABLE \"playlists\"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOINC
REMENT 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 N
VARCHAR(120)\r\n)\n\nCREATE INDEX IFK_TrackMediaTypeId ON \"tracks\" (MediaT
ypeId)\n\n\n===Additional Context \n\nIn the chinook database invoice means
order\n\n===Response Guidelines \n1. If the provided context is sufficient,
please generate a valid SQL query without any explanations for the question.
\n2. If the provided context is almost sufficient but requires knowledge of
a specific string in a particular column, please generate an intermediate SQ
L query to find the distinct strings in that column. Prepend the query with
a comment saying intermediate_sql \n3. If the provided context is insufficie
nt, please explain why it can't be generated. \n4. Please use the most relev
ant table(s). \n5. If the question has been asked and answered before, pleas
e repeat the answer exactly as it was given before. \n"}, {"role": "user",
"content": "Can you list all tables in the SQLite database catalog?"}, {"rol
e": "assistant", "content": "SELECT name FROM sqlite_master WHERE type = 'ta
ble'"}, {"role": "user", "content": "what are the top 5 countries that custo
mers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT
(*) as CustomerCount FROM customers\nGROUP BY Country\nORDER BY CustomerCoun
t DESC\nLIMIT 5"}, {"role": "user", "content": "How many customers are ther
e"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"r
ole": "user", "content": "  \n    List all albums and their corresponding ar
tist names  \n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:36:31.425935751
Z', 'message': {'role': 'assistant', 'content': 'SELECT al.Title as AlbumTit
le, ar.Name as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId =
ar.ArtistId'}, 'done_reason': 'stop', 'done': True, 'total_duration': 405511
74830, 'load_duration': 23149878, 'prompt_eval_count': 840, 'prompt_eval_dur
ation': 31755465000, 'eval_count': 39, 'eval_duration': 8341505000}
LLM Response: SELECT al.Title as AlbumTitle, ar.Name as ArtistName
FROM albums al
JOIN artists ar ON al.ArtistId = ar.ArtistId
SELECT al.Title as AlbumTitle, ar.Name as ArtistName
FROM albums al
JOIN artists ar ON al.ArtistId = ar.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]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: '
\n    List all albums and their corresponding artist names  \n'\n\nThe DataF
rame was produced using this query: SELECT al.Title as AlbumTitle, ar.Name a
s ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.ArtistId\n
\nThe following is information about the resulting pandas DataFrame 'df': \n
Running df.dtypes gives:\n AlbumTitle    object\nArtistName    object\ndtyp
e: object"}, {"role": "user", "content": "Can you generate the Python plotly
code to chart the results of the dataframe? Assume the data is in a pandas d
ataframe called 'df'. If there is only one value in the dataframe, use an In
dicator. Respond with only Python code. Do not answer with any explanations
-- just the code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:36:50.497782656
Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.ex
press as px\n\nfig = px.bar(df, x='AlbumTitle', y='ArtistName', title='Album
s and Artists')\n\nfig.update_layout(xaxis_title='Album Title', yaxis_title
='Artist Name')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True,
'total_duration': 19049517071, 'load_duration': 20950152, 'prompt_eval_coun
t': 196, 'prompt_eval_duration': 6483082000, 'eval_count': 64, 'eval_duratio
n': 12501427000}
```


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

Out[20]:
```
('SELECT al.Title as AlbumTitle, ar.Name as ArtistName\nFROM albums al\nJOI
N artists ar ON al.ArtistId = ar.ArtistId',
                                              AlbumTitle  \
 0                For 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': [{'alignmentgroup': 'True',
               'hovertemplate': 'AlbumTitle=%{x}<br>ArtistName=%{y}<extra>
</extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['For Those About To Rock We Salute You', 'Balls
to the Wall',
                           'Restless and Wild', ..., "Monteverdi: L'Orfeo",
                           'Mozart: Chamber Music',
                           'Koyaanisqatsi (Soundtrack from the Motion Pictu
re)'], dtype=object),
               'xaxis': 'x',
               'y': array(['AC/DC', 'Accept', 'Accept', ...,
                           'C. Monteverdi, Nigel Rogers - Chiaroscuro; Lond
on Baroque; London Cornett & Sackbu',
                           'Nash Ensemble', 'Philip Glass Ensemble'], dtype
=object),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
```

```
                'template': '...',
                'title': {'text': 'Albums and Artists'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Album Title'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Artist Name'}}}
 }))
```

In [21]:
```
question = """
    Find all tracks with a name containing "What" (case-insensitive)
"""

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please
help to generate a SQL query to answer the question. Your response should ON
LY be based on the given context and follow the response guidelines and form
at instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON "tracks" (Ge
nreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackI
d)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGE
R,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Comp
oser NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTE
GER,\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 ACTIO
N,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DEL
ETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENC
ES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI
ON\r\n)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE IND
EX 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, TrackI
d),\r\n    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n
\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) RE
FERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId 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 I
NTEGER 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. I
f the provided context is almost sufficient but requires knowledge of a spec
ific string in a particular column, please generate an intermediate SQL quer
y to find the distinct strings in that column. Prepend the query with a comm
ent saying intermediate_sql \n3. If the provided context is insufficient, pl
ease explain why it can\'t be generated. \n4. Please use the most relevant t
able(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', 'conte
nt': '  \n    List all albums and their corresponding artist names  \n'},
{'role': 'assistant', 'content': 'SELECT al.Title as AlbumTitle, ar.Name as
ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.ArtistId'},
{'role': 'user', 'content': 'Can you list all tables in the SQLite database
catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master
WHERE type = 'table'"}, {'role': 'user', 'content': 'what are the top 5 coun
tries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT
Country, COUNT(*) as CustomerCount FROM customers\nGROUP BY Country\nORDER B
Y CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many custo
mers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM cus
tomers'}, {'role': 'user', 'content': '  \n    Find all tracks with a name c
ontaining "What" (case-insensitive)\n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
```

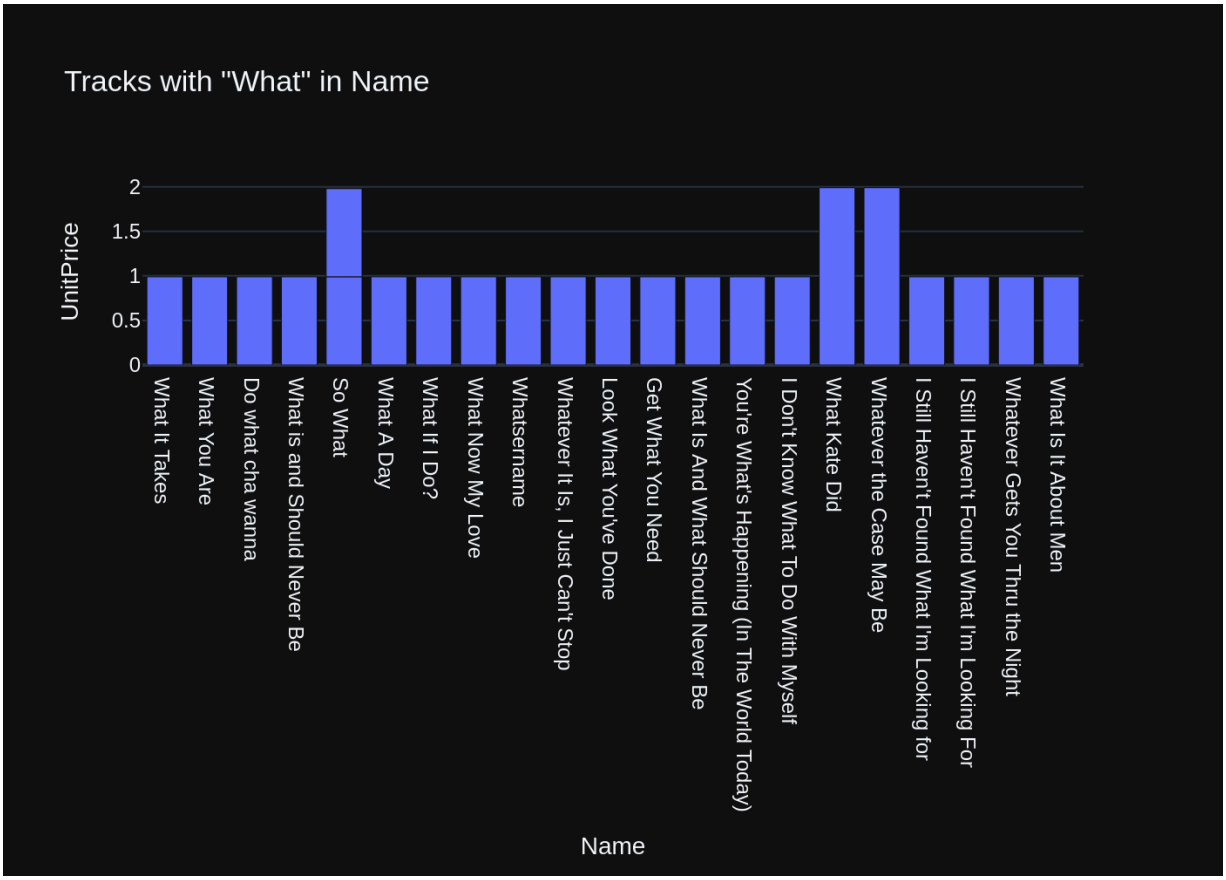
ons. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (GenreId)\n\n
CREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\n\nCRE
ATE TABLE \"tracks\"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT N
OT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n
MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARC
HAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n
UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES
\"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n
FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO A
CTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES \"med
ia_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r
\n)\n\nCREATE INDEX IFK_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 NO
T NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackI
d),\r\n   FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId)
REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackI
d)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABL
E \"playlists\"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT
NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n
GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(12
0)\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means
order\n\n===Response Guidelines \n1. If the provided context is sufficient,
please generate a valid SQL query without any explanations for the question.
\n2. If the provided context is almost sufficient but requires knowledge of
a specific string in a particular column, please generate an intermediate SQ
L query to find the distinct strings in that column. Prepend the query with
a comment saying intermediate_sql \n3. If the provided context is insufficie
nt, please explain why it can't be generated. \n4. Please use the most relev
ant table(s). \n5. If the question has been asked and answered before, pleas
e repeat the answer exactly as it was given before. \n"}, {"role": "user",
"content": " \n   List all albums and their corresponding artist names
\n"}, {"role": "assistant", "content": "SELECT al.Title as AlbumTitle, ar.Na
me as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.Artist
Id"}, {"role": "user", "content": "Can you list all tables in the SQLite dat
abase catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_
master WHERE type = 'table'"}, {"role": "user", "content": "what are the top
5 countries that customers come from?"}, {"role": "assistant", "content": "S
ELECT Country, COUNT(*) as CustomerCount FROM customers\nGROUP BY Country\nO
RDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "How many
customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FRO
M customers"}, {"role": "user", "content": " \n   Find all tracks with a n
ame containing \"What\" (case-insensitive)\n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:37:29.83023206Z',
'message': {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name
LIKE '%What%'"}, 'done_reason': 'stop', 'done': True, 'total_duration': 3909
1292355, 'load_duration': 18696710, 'prompt_eval_count': 944, 'prompt_eval_d
uration': 35100409000, 'eval_count': 17, 'eval_duration': 3429278000}
LLM Response: SELECT * FROM tracks WHERE Name LIKE '%What%'
SELECT * FROM tracks WHERE Name LIKE '%What%'
    TrackId   Name AlbumId \
0   26   What It Takes   5
1   88   What You Are   10

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

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

	Milliseconds	Bytes	UnitPrice
0	310622	10144730	0.99
1	249391	5988186	0.99
2	274155	9018565	0.99
3	260675	8497116	0.99
4	564009	18360449	0.99
5	158275	5203430	0.99
6	302994	9929799	0.99
7	149995	4913383	0.99
8	252316	8244843	0.99

```
9         247222    8249453       0.99
10        230974    7517083       0.99
11        247719    8043765       0.99
12        287973    9369385       0.99
13        142027    4631104       0.99
14        189152    6162894       0.99
15        221387    7251478       0.99
16       2610250  484583988       1.99
17       2616410  183867185       1.99
18        353567   11542247       0.99
19        280764    9306737       0.99
20        215084    3499018       0.99
21        209573    3426106       0.99
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: '
\n    Find all tracks with a name containing \"What\" (case-insensitive)
\n'\n\nThe DataFrame was produced using this query: SELECT * FROM tracks WHE
RE Name LIKE '%What%'\n\nThe following is information about the resulting pa
ndas DataFrame 'df': \nRunning df.dtypes gives:\n TrackId           int64\nN
ame             object\nAlbumId           int64\nMediaTypeId       int64\nGe
nreId           int64\nComposer         object\nMilliseconds      int64\nByt
es             int64\nUnitPrice       float64\ndtype: object"}, {"role": "us
er", "content": "Can you generate the Python plotly code to chart the result
s of the dataframe? Assume the data is in a pandas dataframe called 'df'. If
there is only one value in the dataframe, use an Indicator. Respond with onl
y Python code. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:37:46.249636056
Z', 'message': {'role': 'assistant', 'content': '```python\nimport plotly.ex
press as px\n\nfig = px.bar(df, x=\'Name\', y=\'UnitPrice\', title=\'Tracks
with "What" in Name\')\n\nfig.show()\n```'}, 'done_reason': 'stop', 'done':
True, 'total_duration': 16390360883, 'load_duration': 18314570, 'prompt_eval
_count': 224, 'prompt_eval_duration': 7771277000, 'eval_count': 44, 'eval_du
ration': 8554750000}
```


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

Out[21]: ("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	1	3	Culmer/Exal

```
t
 15            1        7                                                  Non
e
 16            3       19                                                  Non
e
 17            3       19                                                  Non
e
 18            1        1      Bono/Clayton, Adam/Mullen Jr., Larry/The Edg
e
 19            1        1                                                    U
2
 20            2        9                                                  Non
e
 21            2        9  Delroy "Chris" Cooper, Donovan Jackson, Earl
C...

     Milliseconds      Bytes  UnitPrice
 0         310622   10144730       0.99
 1         249391    5988186       0.99
 2         274155    9018565       0.99
 3         260675    8497116       0.99
 4         564009   18360449       0.99
 5         158275    5203430       0.99
 6         302994    9929799       0.99
 7         149995    4913383       0.99
 8         252316    8244843       0.99
 9         247222    8249453       0.99
 10        230974    7517083       0.99
 11        247719    8043765       0.99
 12        287973    9369385       0.99
 13        142027    4631104       0.99
 14        189152    6162894       0.99
 15        221387    7251478       0.99
 16       2610250  484583988       1.99
 17       2616410  183867185       1.99
 18        353567   11542247       0.99
 19        280764    9306737       0.99
 20        215084    3499018       0.99
 21        209573    3426106       0.99  ,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'Name=%{x}<br>UnitPrice=%{y}<extra></extra
>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['What It Takes', 'What You Are', 'Do what cha wa
nna',
                           '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 Kat
e Did',
                            'Whatever the Case May Be',
                            "I Still Haven't Found What I'm Looking for",
                            "I Still Haven't Found What I'm Looking For",
                            'Whatever Gets You Thru the Night', 'What Is It
About Men'],
                           dtype=object),
               'xaxis': 'x',
               'y': array([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]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Tracks with "What" in Name'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Name'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'UnitPrice'}}}
 }))
```

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

vn.ask(question=question)
```

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

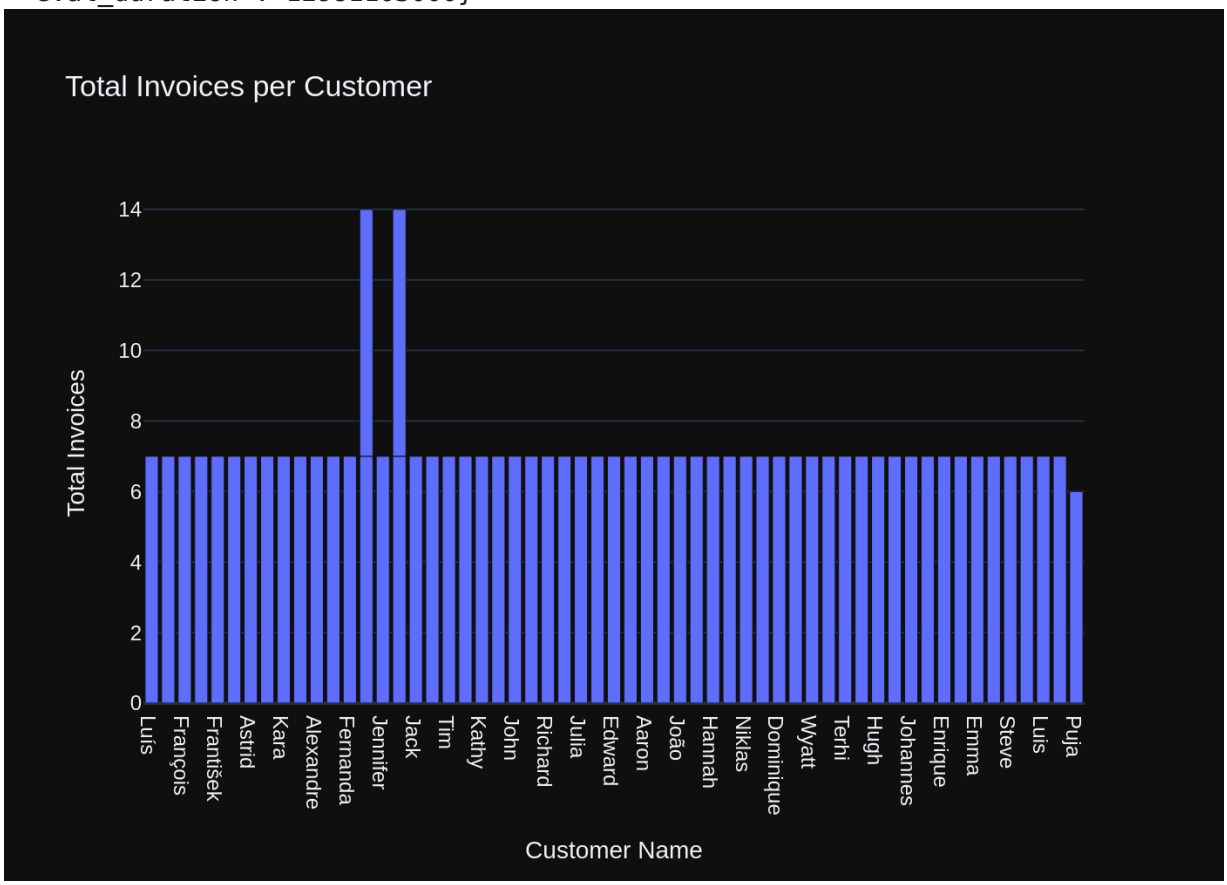
```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please
help to generate a SQL query to answer the question. Your response should ON
LY be based on the given context and follow the response guidelines and form
at instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId
INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT N
ULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(7
0),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n
BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    T
otal NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "cu
stomers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n
\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE IN
DEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE TABLE
"invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT
NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NU
LL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT N
ULL,\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_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nC
REATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCRE
MENT 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(7
0),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVAR
CHAR(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 I
NTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_
CustomerSupportRepId ON "customers" (SupportRepId)\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 DATETIM
E,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r\n    City NVARCH
AR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    Posta
lCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n
Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFERENCES "employees" (E
mployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IN
DEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE TABLE "track
s"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Na
me NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEG
ER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n
Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMER
IC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Genr
eId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaType
Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional
Context \n\nIn the chinook database invoice means order\n\n===Response Guide
lines \n1. If the provided context is sufficient, please generate a valid SQ
L query without any explanations for the question. \n2. If the provided cont
ext is almost sufficient but requires knowledge of a specific string in a pa
rticular column, please generate an intermediate SQL query to find the disti
nct strings in that column. Prepend the query with a comment saying intermed
iate_sql \n3. If the provided context is insufficient, please explain why it
can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the
question has been asked and answered before, please repeat the answer exactl
y as it was given before. \n'}, {'role': 'user', 'content': 'How many custom
```

ers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM cust
omers'}, {'role': 'user', 'content': 'what are the top 5 countries that cust
omers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT
(*) as CustomerCount FROM customers\nGROUP BY Country\nORDER BY CustomerCoun
t DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   List all albums and
their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SE
LECT al.Title as AlbumTitle, ar.Name as ArtistName\nFROM albums al\nJOIN art
ists ar ON al.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n
Find all tracks with a name containing "What" (case-insensitive)\n'}, {'rol
e': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%Wha
t%'"}, {'role': 'user', 'content': 'Can you list all tables in the SQLite da
tabase catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite
_master WHERE type = 'table'"}, {'role': 'user', 'content': ' \n   Get the
total number of invoices for each customer\n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n   InvoiceId INTEGER P
RIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n
InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   B
illingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCou
ntry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMER
IC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES \"customers\"
(CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE
INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK
_InvoiceLineInvoiceId ON \"invoice_items\" (InvoiceId)\n\nCREATE TABLE \"inv
oice_items\"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT
NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NUL
L,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NU
LL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n
\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) RE
FERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO
N\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)
\n\nCREATE TABLE \"customers\"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AU
TOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   LastNa
me NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARC
HAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country
NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n
Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId I
NTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (Employee
Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK
_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"empl
oyees\"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r
\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NU
LL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DA
TETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City N
VARCHAR(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(2
4),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES \"e
mployees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE INDEX IFK_EmployeeReportsTo ON \"employees\" (ReportsTo)\n\nCREAT

```
E 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    Me
diaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHA
R(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 A
CTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES \"med
ia_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r
\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means ord
er\n\n===Response Guidelines \n1. If the provided context is sufficient, ple
ase generate a valid SQL query without any explanations for the question. \n
2. If the provided context is almost sufficient but requires knowledge of a
specific string in a particular column, please generate an intermediate SQL
query to find the distinct strings in that column. Prepend the query with a
comment saying intermediate_sql \n3. If the provided context is insufficien
t, please explain why it can't be generated. \n4. Please use the most releva
nt table(s). \n5. If the question has been asked and answered before, please
repeat the answer exactly as it was given before. \n"}, {"role": "user", "co
ntent": "How many customers are there"}, {"role": "assistant", "content": "S
ELECT COUNT(*) FROM customers"}, {"role": "user", "content": "what are the t
op 5 countries that customers come from?"}, {"role": "assistant", "content":
"SELECT Country, COUNT(*) as CustomerCount FROM customers\nGROUP BY Country
\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "  \n
List all albums and their corresponding artist names  \n"}, {"role": "assist
ant", "content": "SELECT al.Title as AlbumTitle, ar.Name as ArtistName\nFROM
albums al\nJOIN artists ar ON al.ArtistId = ar.ArtistId"}, {"role": "user",
"content": "  \n    Find all tracks with a name containing \"What\" (case-in
sensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks WHERE
Name LIKE '%What%'"}, {"role": "user", "content": "Can you list all tables i
n the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT n
ame FROM sqlite_master WHERE type = 'table'"}, {"role": "user", "content": "
\n    Get the total number of invoices for each customer\n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:38:51.210591241
Z', 'message': {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastN
ame, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i
ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, 'done_reason': 'sto
p', 'done': True, 'total_duration': 64852362683, 'load_duration': 22964083,
'prompt_eval_count': 1410, 'prompt_eval_duration': 53590551000, 'eval_coun
t': 48, 'eval_duration': 10572407000}
LLM Response: SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as TotalInv
oices
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as TotalInvoices
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
    FirstName      LastName  TotalInvoices
0        Luís     Gonçalves             7
1      Leonie        Köhler             7
2    François      Tremblay             7
3       Bjørn        Hansen             7
4   František   Wichterlová             7
```

```
5      Helena          Holý             7
6      Astrid        Gruber             7
7        Daan       Peeters             7
8        Kara       Nielsen             7
9     Eduardo       Martins             7
10  Alexandre         Rocha             7
11    Roberto       Almeida             7
12   Fernanda         Ramos             7
13       Mark       Philips             7
14   Jennifer      Peterson             7
15      Frank        Harris             7
16       Jack         Smith             7
17   Michelle        Brooks             7
18        Tim         Goyer             7
19        Dan        Miller             7
20      Kathy         Chase             7
21    Heather       Leacock             7
22       John        Gordon             7
23      Frank       Ralston             7
24     Victor       Stevens             7
25    Richard    Cunningham             7
26    Patrick          Gray             7
27      Julia       Barnett             7
28     Robert         Brown             7
29     Edward       Francis             7
30     Martha          Silk             7
31      Aaron      Mitchell             7
32      Ellie      Sullivan             7
33       João     Fernandes             7
34   Madalena       Sampaio             7
35     Hannah     Schneider             7
36       Fynn    Zimmermann             7
37     Niklas      Schröder             7
38    Camille       Bernard             7
39  Dominique      Lefebvre             7
40       Marc        Dubois             7
41      Wyatt        Girard             7
42   Isabelle       Mercier             7
43      Terhi    Hämäläinen             7
44   Ladislav        Kovács             7
45       Hugh      O'Reilly             7
46      Lucas       Mancini             7
47   Johannes  Van der Berg             7
48  Stanisław        Wójcik             7
49    Enrique         Muñoz             7
50     Joakim     Johansson             7
51       Emma         Jones             7
52       Phil        Hughes             7
53      Steve        Murray             7
54       Mark        Taylor             7
55      Diego     Gutiérrez             7
56       Luis         Rojas             7
57      Manoj        Pareek             7
58       Puja    Srivastava             6
Info: Ollama parameters:
model=codegemma:latest,
```

options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: '
\n   Get the total number of invoices for each customer\n'\n\nThe DataFrame
was produced using this query: SELECT c.FirstName, c.LastName, COUNT(i.Invoi
ceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId =
i.CustomerId\nGROUP BY c.CustomerId\n\nThe following is information about th
e resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n FirstName
object\nLastName   object\nTotalInvoices   int64\ndtype: object"},
{"role": "user", "content": "Can you generate the Python plotly code to char
t the results of the dataframe? Assume the data is in a pandas dataframe cal
led 'df'. If there is only one value in the dataframe, use an Indicator. Res
pond with only Python code. Do not answer with any explanations -- just the
code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:39:11.696501278
Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.ex
press as px\n\nfig = px.bar(df, x='FirstName', y='TotalInvoices', title='Tot
al Invoices per Customer')\n\nfig.update_layout(xaxis_title='Customer Name',
yaxis_title='Total Invoices')\n\nfig.show()\n```"}, 'done_reason': 'stop',
'done': True, 'total_duration': 20464364564, 'load_duration': 21953592, 'pro
mpt_eval_count': 210, 'prompt_eval_duration': 7411622000, 'eval_count': 66,
'eval_duration': 12981103000}



Out[22]: ('SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId',

	FirstName	LastName	TotalInvoices
0	Luís	Gonçalves	7
1	Leonie	Köhler	7
2	François	Tremblay	7
3	Bjørn	Hansen	7
4	František	Wichterlová	7
5	Helena	Holý	7
6	Astrid	Gruber	7
7	Daan	Peeters	7
8	Kara	Nielsen	7
9	Eduardo	Martins	7
10	Alexandre	Rocha	7
11	Roberto	Almeida	7
12	Fernanda	Ramos	7
13	Mark	Philips	7
14	Jennifer	Peterson	7
15	Frank	Harris	7
16	Jack	Smith	7
17	Michelle	Brooks	7
18	Tim	Goyer	7
19	Dan	Miller	7
20	Kathy	Chase	7
21	Heather	Leacock	7
22	John	Gordon	7
23	Frank	Ralston	7
24	Victor	Stevens	7
25	Richard	Cunningham	7
26	Patrick	Gray	7
27	Julia	Barnett	7
28	Robert	Brown	7
29	Edward	Francis	7
30	Martha	Silk	7
31	Aaron	Mitchell	7
32	Ellie	Sullivan	7
33	João	Fernandes	7
34	Madalena	Sampaio	7
35	Hannah	Schneider	7
36	Fynn	Zimmermann	7
37	Niklas	Schröder	7
38	Camille	Bernard	7
39	Dominique	Lefebvre	7
40	Marc	Dubois	7
41	Wyatt	Girard	7
42	Isabelle	Mercier	7
43	Terhi	Hämäläinen	7
44	Ladislav	Kovács	7
45	Hugh	O'Reilly	7
46	Lucas	Mancini	7
47	Johannes	Van der Berg	7
48	Stanisław	Wójcik	7
49	Enrique	Muñoz	7
50	Joakim	Johansson	7
51	Emma	Jones	7

```
 52       Phil        Hughes             7
 53      Steve        Murray             7
 54       Mark        Taylor             7
 55      Diego     Gutiérrez             7
 56       Luis         Rojas             7
 57      Manoj        Pareek             7
 58       Puja    Srivastava             6,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'FirstName=%{x}<br>TotalInvoices=%{y}<extra
></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Luís', 'Leonie', 'François', 'Bjørn', 'Františe
k', 'Helena', 'Astrid',
                           'Daan', 'Kara', 'Eduardo', 'Alexandre', 'Robert
o', 'Fernanda', 'Mark',
                           'Jennifer', 'Frank', 'Jack', 'Michelle', 'Tim',
'Dan', 'Kathy',
                           'Heather', 'John', 'Frank', 'Victor', 'Richard',
'Patrick', 'Julia',
                           'Robert', 'Edward', 'Martha', 'Aaron', 'Ellie',
'João', 'Madalena',
                           'Hannah', 'Fynn', 'Niklas', 'Camille', 'Dominiqu
e', 'Marc', 'Wyatt',
                           'Isabelle', 'Terhi', 'Ladislav', 'Hugh', 'Luca
s', 'Johannes',
                           'Stanisław', 'Enrique', 'Joakim', 'Emma', 'Phi
l', 'Steve', 'Mark',
                           'Diego', 'Luis', 'Manoj', 'Puja'], dtype=objec
t),
               'xaxis': 'x',
               'y': array([7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
7, 7, 7, 7, 7, 7, 7, 7,
                           7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
7, 7, 7, 7, 7, 7, 7, 7,
                           7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 6]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Total Invoices per Customer'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Customer Name'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Total Invoices'}}}
 }))
```

In [23]:
```
question = """
    Find the total number of invoices per country:
```

```
"""

vn.ask(question=question)
```

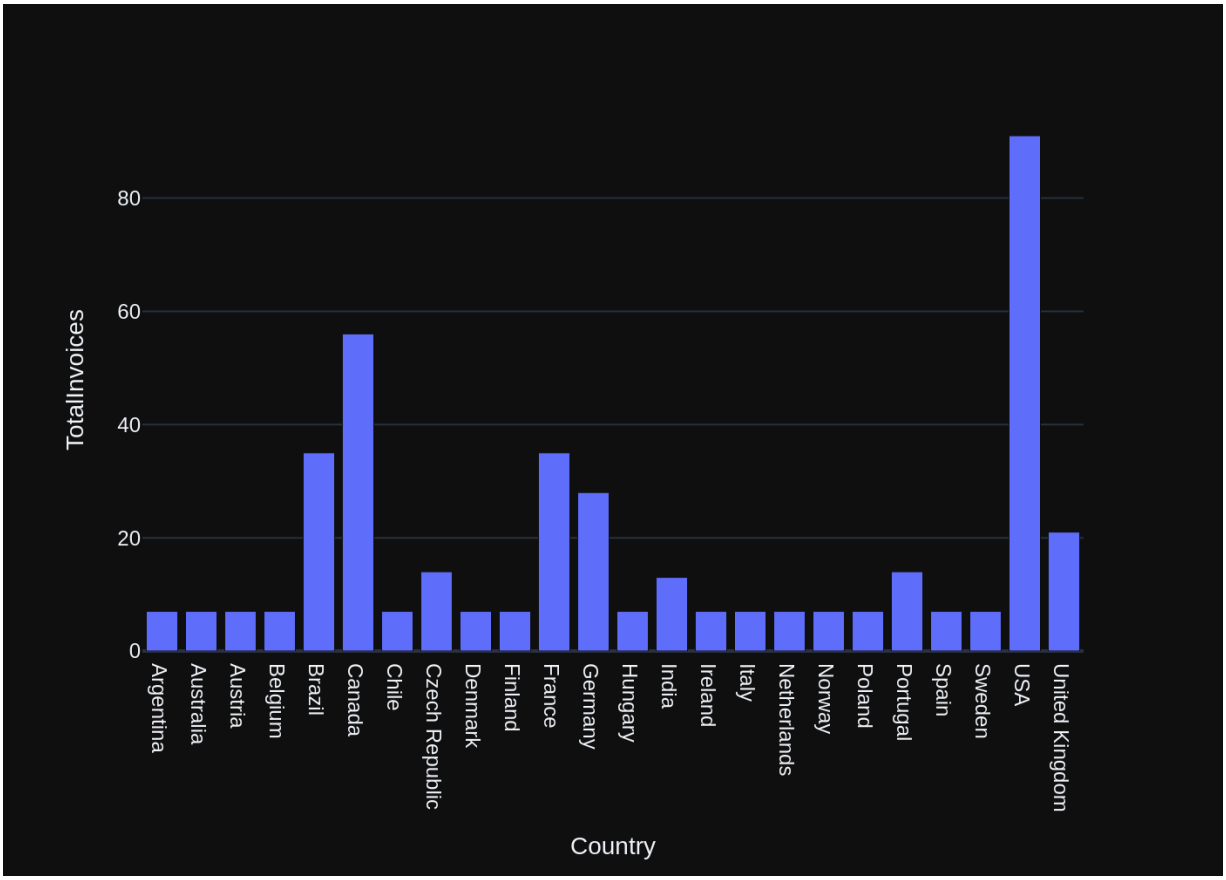
```
Number of requested results 10 is greater than number of elements in index
6, updating n_results = 6
Number of requested results 10 is greater than number of elements in index
1, updating n_results = 1
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please
help to generate a SQL query to answer the question. Your response should ON
LY be based on the given context and follow the response guidelines and form
at instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId
INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT N
ULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(7
0),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n
BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    T
otal NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "cu
stomers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n
\nCREATE TABLE "invoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY
AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId I
NTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity
INTEGER  NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES "invoices" (Inv
oiceId) \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 UPDAT
E NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (Custom
erId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceI
d)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCR
EATE TABLE "employees"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREM
ENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARC
HAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n
BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVARCHAR(70),\r
\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NVARCHAR
(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(24),\r\n    Fax
NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY (ReportsTo) REFE
RENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY
KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n
LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address
NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Co
untry NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(2
4),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NULL,\r\n    Sup
portRepId 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 "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 NUL
L,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n
TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(20
0)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NUL
L,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Millisecond
s INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  N
OT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\t
ON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERE
NCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n
FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReport
sTo ON "employees" (ReportsTo)\n\n\n===Additional Context \n\nIn the chinook
database invoice means order\n\n===Response Guidelines \n1. If the provided
context is sufficient, please generate a valid SQL query without any explana
tions for the question. \n2. If the provided context is almost sufficient bu
t requires knowledge of a specific string in a particular column, please gen
erate an intermediate SQL query to find the distinct strings in that column.
Prepend the query with a comment saying intermediate_sql \n3. If the provide
d 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 an
d answered before, please repeat the answer exactly as it was given before.
\n'}, {'role': 'user', 'content': ' \n   Get the total number of invoices
for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName,
c.LastName, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invo
ices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'us
er', 'content': 'what are the top 5 countries that customers come from?'},
{'role': 'assistant', 'content': 'SELECT Country, COUNT(*) as CustomerCount
FROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'},
{'role': 'user', 'content': 'How many customers are there'}, {'role': 'assis
tant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'conte
nt': ' \n   List all albums and their corresponding artist names \n'},
{'role': 'assistant', 'content': 'SELECT al.Title as AlbumTitle, ar.Name as
ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.ArtistId'},
{'role': 'user', 'content': ' \n   Find all tracks with a name containing
"What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FR
OM tracks WHERE Name LIKE '%What%'"}, {'role': 'user', 'content': 'Can you l
ist all tables in the SQLite database catalog?'}, {'role': 'assistant', 'con
tent': "SELECT name FROM sqlite_master WHERE type = 'table'"}, {'role': 'use
r', 'content': ' \n   Find the total number of invoices per country:\n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n   InvoiceId INTEGER P
RIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n
InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   B
illingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCou
ntry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMER
IC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES \"customers\"
(CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE
TABLE \"invoice_items\"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOIN
CREMENT 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 (TrackI
d) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerI
d)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON \"invoice_items\" (InvoiceId)
\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCR
EATE TABLE \"employees\"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVA
RCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r
\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR
(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NV
ARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n
Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo)
REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE
NO ACTION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n   CustomerId INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NUL
L,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n
Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r

\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVA
RCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n
SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES \"employe
es\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCR
EATE TABLE \"albums\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT
NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT
NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t
\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks\"\r
\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name N
VARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER
NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Mill
iseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(1
0,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreI
d) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTy
peId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX I
FK_EmployeeReportsTo ON \"employees\" (ReportsTo)\n\n\n===Additional Context
\n\nIn the chinook database invoice means order\n\n===Response Guidelines \n
1. If the provided context is sufficient, please generate a valid SQL query
without any explanations for the 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 strin
gs in that column. Prepend the query with a comment saying intermediate_sql
\n3. If the provided context is insufficient, please explain why it can't be
generated. \n4. Please use the most relevant table(s). \n5. If the question
has been asked and answered before, please repeat the answer exactly as it w
as given before. \n"}, {"role": "user", "content": " \n   Get the total nu
mber of invoices for each customer\n"}, {"role": "assistant", "content": "SE
LECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as TotalInvoices\nFROM cust
omers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Customer
Id"}, {"role": "user", "content": "what are the top 5 countries that custome
rs come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*)
as CustomerCount FROM customers\nGROUP BY Country\nORDER BY CustomerCount DE
SC\nLIMIT 5"}, {"role": "user", "content": "How many customers are there"},
{"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role":
"user", "content": " \n   List all albums and their corresponding artist n
ames \n"}, {"role": "assistant", "content": "SELECT al.Title as AlbumTitle,
ar.Name as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.A
rtistId"}, {"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": "Can you list all tables in the SQLite database catalog?"}, {"role": "as
sistant", "content": "SELECT name FROM sqlite_master WHERE type = 'table'"},
{"role": "user", "content": " \n   Find the total number of invoices per c
ountry:\n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:40:20.853090655
Z', 'message': {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.I
nvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerI
d = i.CustomerId\nGROUP BY c.Country'}, 'done_reason': 'stop', 'done': True,
'total_duration': 69039064143, 'load_duration': 22821650, 'prompt_eval_coun
t': 1552, 'prompt_eval_duration': 58571088000, 'eval_count': 44, 'eval_durat
ion': 9698855000}
LLM Response: SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices
FROM customers c

```
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.Country
SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.Country
           Country  TotalInvoices
0        Argentina              7
1        Australia              7
2          Austria              7
3          Belgium              7
4           Brazil             35
5           Canada             56
6            Chile              7
7   Czech Republic             14
8          Denmark              7
9          Finland              7
10          France             35
11         Germany             28
12         Hungary              7
13           India             13
14         Ireland              7
15           Italy              7
16     Netherlands              7
17          Norway              7
18          Poland              7
19        Portugal             14
20           Spain              7
21          Sweden              7
22             USA             91
23  United Kingdom             21
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: '
\n    Find the total number of invoices per country:\n'\n\nThe DataFrame was
produced using this query: SELECT c.Country, COUNT(i.InvoiceId) as TotalInvo
ices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROU
P BY c.Country\n\nThe following is information about the resulting pandas Da
taFrame 'df': \nRunning df.dtypes gives:\n Country          object\nTotalInv
oices     int64\ndtype: object"}, {"role": "user", "content": "Can you gener
ate the Python plotly code to chart the results of the dataframe? Assume the
data is in a pandas dataframe called 'df'. If there is only one value in the
dataframe, use an Indicator. Respond with only Python code. Do not answer wi
th any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:40:36.668879952
Z', 'message': {'role': 'assistant', 'content': "import plotly.express as px
\n\nfig = px.bar(df, x='Country', y='TotalInvoices', title='Total Invoices p
er Country')\n\nfig.show()\n</start_of_turn>"}, 'done_reason': 'stop', 'don
e': True, 'total_duration': 15794682591, 'load_duration': 21720770, 'prompt_
eval_count': 202, 'prompt_eval_duration': 6824709000, 'eval_count': 46, 'eva
l_duration': 8902095000}
```


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

Out[23]:
```
('SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\n
JOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country',
            Country  TotalInvoices
 0         Argentina             7
 1         Australia             7
 2           Austria             7
 3           Belgium             7
 4            Brazil            35
 5            Canada            56
 6             Chile             7
 7    Czech Republic            14
 8           Denmark             7
 9           Finland             7
 10           France            35
 11          Germany            28
 12          Hungary             7
 13            India            13
 14          Ireland             7
 15            Italy             7
 16      Netherlands             7
 17           Norway             7
 18           Poland             7
 19         Portugal            14
 20            Spain             7
 21           Sweden             7
 22              USA            91
 23   United Kingdom            21,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'Country=%{x}<br>TotalInvoices=%{y}<extra>
</extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Argentina', 'Australia', 'Austria', 'Belgium',
'Brazil', 'Canada',
                           'Chile', 'Czech Republic', 'Denmark', 'Finland',
'France', 'Germany',
                           'Hungary', 'India', 'Ireland', 'Italy', 'Netherl
ands', 'Norway',
                           'Poland', 'Portugal', 'Spain', 'Sweden', 'USA',
'United Kingdom'],
                          dtype=object),
               'xaxis': 'x',
               'y': array([ 7,  7,  7,  7, 35, 56,  7, 14,  7,  7, 35, 28,
7, 13,  7,  7,  7,  7,
                            7, 14,  7,  7, 91, 21]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
```

```
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Country'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'TotalInvoices'}}}
 }))
```

In [24]:
```
question = """
    List all invoices with a total exceeding $10:
"""

vn.ask(question=question)
```

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

SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_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 AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE TABLE "customers"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "employees"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Get the total n

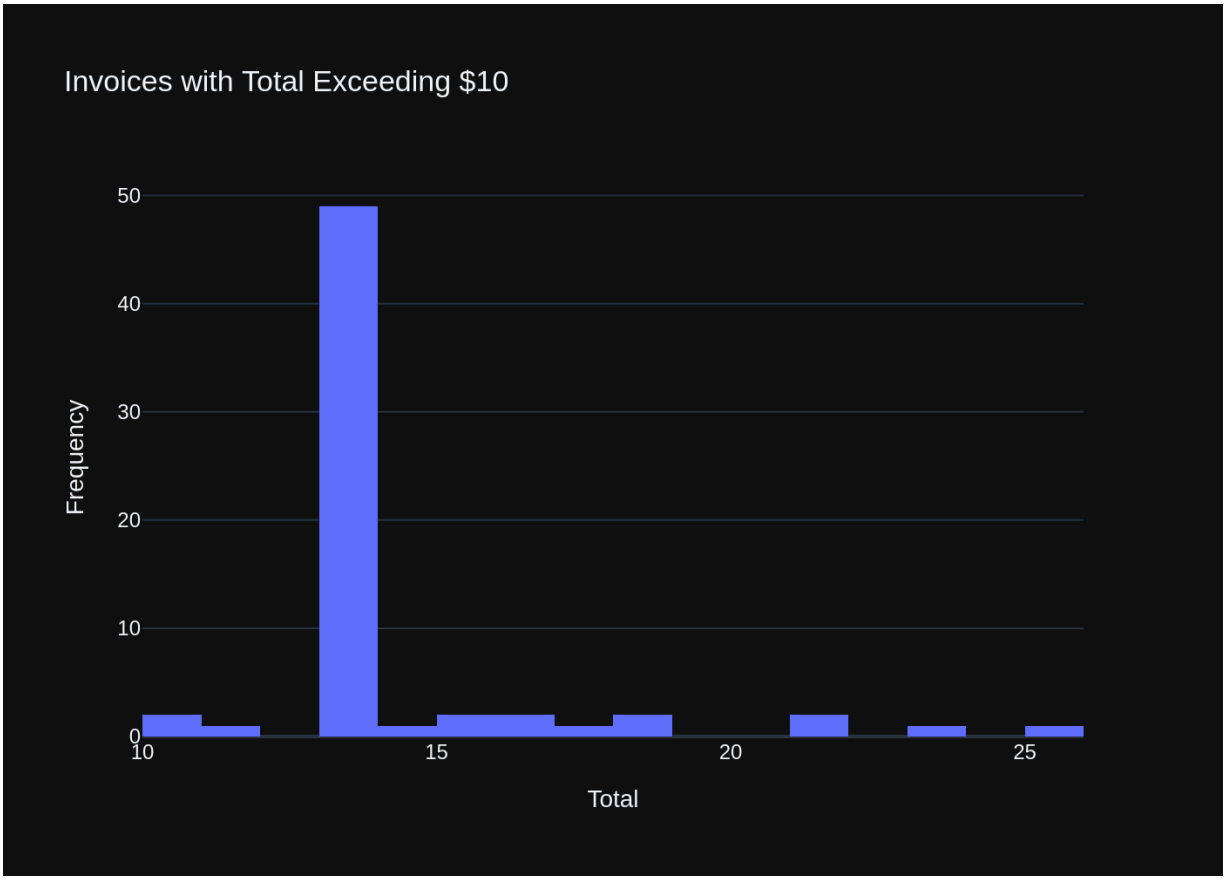
umber of invoices for each customer\n'}, {'role': 'assistant', 'content': 'S
ELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as TotalInvoices\nFROM cus
tomers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Custome
rId'}, {'role': 'user', 'content': ' \n   Find the total number of invoice
s per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUN
T(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.Cust
omerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': 'Ho
w many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT
(*) FROM customers'}, {'role': 'user', 'content': 'what are the top 5 countr
ies that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Co
untry, COUNT(*) as CustomerCount FROM customers\nGROUP BY Country\nORDER BY
CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   List all
albums and their corresponding artist names \n'}, {'role': 'assistant', 'co
ntent': 'SELECT al.Title as AlbumTitle, ar.Name as ArtistName\nFROM albums a
l\nJOIN artists ar ON al.ArtistId = ar.ArtistId'}, {'role': 'user', 'conten
t': ' \n   Find all tracks with a name containing "What" (case-insensitiv
e)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LI
KE '%What%'"}, {'role': 'user', 'content': 'Can you list all tables in the S
QLite database catalog?'}, {'role': 'assistant', 'content': "SELECT name FRO
M sqlite_master WHERE type = 'table'"}, {'role': 'user', 'content': ' \n
List all invoices with a total exceeding $10:\n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE TABLE \"invoice_items\"\r\n(\r\n   InvoiceLineId
INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NU
LL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NU
LL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERE
NCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO
N,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON D
ELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvo
iceId ON \"invoice_items\" (InvoiceId)\n\nCREATE TABLE \"invoices\"\r\n(\r\n
InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTE
GER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress
NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR
(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR
(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId)
REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE
NO ACTION\r\n)\n\nCREATE INDEX IFK_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 AUTOINCR
EMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGE
R,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Comp
oser NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTE
GER,\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 ACT
ION,\r\n   FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFER
ENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON \"employees\" (ReportsT
o)\n\nCREATE TABLE \"customers\"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY

AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   Last
Name NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVA
RCHAR(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(2
4),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   Sup
portRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES \"employees
\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREA
TE TABLE \"employees\"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREM
ENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARC
HAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n
BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r
\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR
(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax
NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFE
RENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO A
CTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (Suppor
tRepId)\n\n\n===Additional Context \n\nIn the chinook database invoice means
order\n\n===Response Guidelines \n1. If the provided context is sufficient,
please generate a valid SQL query without any explanations for the question.
\n2. If the provided context is almost sufficient but requires knowledge of
a specific string in a particular column, please generate an intermediate SQ
L query to find the distinct strings in that column. Prepend the query with
a comment saying intermediate_sql \n3. If the provided context is insufficie
nt, please explain why it can't be generated. \n4. Please use the most relev
ant table(s). \n5. If the question has been asked and answered before, pleas
e repeat the answer exactly as it was given before. \n"}, {"role": "user",
"content": " \n   Get the total number of invoices for each customer\n"},
{"role": "assistant", "content": "SELECT c.FirstName, c.LastName, COUNT(i.In
voiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId
= i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n
Find the total number of invoices per country:\n"}, {"role": "assistant", "c
ontent": "SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices\nFROM custom
ers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"},
{"role": "user", "content": "How many customers are there"}, {"role": "assis
tant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user", "conte
nt": "what are the top 5 countries that customers come from?"}, {"role": "as
sistant", "content": "SELECT Country, COUNT(*) as CustomerCount FROM custome
rs\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "use
r", "content": " \n   List all albums and their corresponding artist names
\n"}, {"role": "assistant", "content": "SELECT al.Title as AlbumTitle, ar.Na
me as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.Artist
Id"}, {"role": "user", "content": " \n   Find all tracks with a name conta
ining \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SEL
ECT * FROM tracks WHERE Name LIKE '%What%'"}, {"role": "user", "content": "C
an you list all tables in the SQLite database catalog?"}, {"role": "assistan
t", "content": "SELECT name FROM sqlite_master WHERE type = 'table'"}, {"rol
e": "user", "content": " \n   List all invoices with a total exceeding $1
0:\n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:41:40.974763899
Z', 'message': {'role': 'assistant', 'content': 'SELECT * FROM invoices WHER
E Total > 10'}, 'done_reason': 'stop', 'done': True, 'total_duration': 64195
970244, 'load_duration': 22235610, 'prompt_eval_count': 1584, 'prompt_eval_d
uration': 59590342000, 'eval_count': 17, 'eval_duration': 3608561000}
LLM Response: SELECT * FROM invoices WHERE Total > 10
SELECT * FROM invoices WHERE Total > 10

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

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

[64 rows x 9 columns]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: '
\n    List all invoices with a total exceeding $10:\n'\n\nThe DataFrame was
produced using this query: SELECT * FROM invoices WHERE Total > 10\n\nThe fo
llowing is information about the resulting pandas DataFrame 'df': \nRunning
df.dtypes gives:\n InvoiceId              int64\nCustomerId             int6
4\nInvoiceDate           object\nBillingAddress        object\nBillingCity
object\nBillingState          object\nBillingCountry        object\nBillingP
ostalCode     object\nTotal                float64\ndtype: object"}, {"rol
e": "user", "content": "Can you generate the Python plotly code to chart the
results of the dataframe? Assume the data is in a pandas dataframe called 'd
f'. If there is only one value in the dataframe, use an Indicator. Respond w
ith only Python code. Do not answer with any explanations -- just the cod
e."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:42:01.869868552
Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.ex
press as px\n\nfig = px.histogram(df, x='Total', nbins=20, title='Invoices w
ith Total Exceeding $10')\n\nfig.update_layout(xaxis_title='Total', yaxis_ti
tle='Frequency')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True,
'total_duration': 20868540209, 'load_duration': 21465644, 'prompt_eval_coun
t': 215, 'prompt_eval_duration': 7695754000, 'eval_count': 67, 'eval_duratio
n': 13014299000}
```


Invoices with Total Exceeding $10
Frequency
50
40
30
20
10
0
10
15
20
25
Total

Out[24]:
```
('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',
               'bingroup': 'x',
               'hovertemplate': 'Total=%{x}<br>count=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'nbinsx': 20,
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'type': 'histogram',
               'x': 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]),
               'xaxis': 'x',
               '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': {'t
ext': 'Total'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Frequency'}}}
 }))
```

In [25]:
```
question = """
    Find all invoices since 2010 and the total amount invoiced:
"""

vn.ask(question=question)
```

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

SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE 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 PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the

provided context is almost sufficient but requires knowledge of a specific s
tring in a particular column, please generate an intermediate SQL query to f
ind the distinct strings in that column. Prepend the query with a comment sa
ying intermediate_sql \n3. If the provided context is insufficient, please e
xplain why it can\'t be generated. \n4. Please use the most relevant table
(s). \n5. If the question has been asked and answered before, please repeat
the answer exactly as it was given before. \n'}, {'role': 'user', 'content':
' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assist
ant', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'use
r', 'content': ' \n   Get the total number of invoices for each customer
\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, COUN
T(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.Cust
omerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content':
' \n   Find the total number of invoices per country:\n'}, {'role': 'assis
tant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices\nFR
OM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.C
ountry'}, {'role': 'user', 'content': 'How many customers are there'}, {'rol
e': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'use
r', 'content': 'what are the top 5 countries that customers come from?'},
{'role': 'assistant', 'content': 'SELECT Country, COUNT(*) as CustomerCount
FROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'},
{'role': 'user', 'content': ' \n   List all albums and their corresponding
artist names \n'}, {'role': 'assistant', 'content': 'SELECT al.Title as Alb
umTitle, ar.Name as ArtistName\nFROM albums al\nJOIN artists ar ON al.Artist
Id = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find all tracks wit
h a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'co
ntent': "SELECT * FROM tracks WHERE Name LIKE '%What%'"}, {'role': 'user',
'content': 'Can you list all tables in the SQLite database catalog?'}, {'rol
e': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type = 'ta
ble'"}, {'role': 'user', 'content': ' \n   Find all invoices since 2010 an
d the total amount invoiced:\n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n   InvoiceId INTEGER P
RIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n
InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   B
illingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCou
ntry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMER
IC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES \"customers\"
(CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE
TABLE \"invoice_items\"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOIN
CREMENT 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 (TrackI
d) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON \"invoice_items\" (I
nvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)
\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCR
EATE TABLE \"employees\"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCR

EMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVA
RCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r
\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR
(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NV
ARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n
Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo)
REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE
NO ACTION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n   CustomerId INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NUL
L,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n
Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r
\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVA
RCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n
SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES \"employe
es\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCR
EATE 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 NVARC
HAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n
UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES
\"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n
FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO A
CTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES \"med
ia_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r
\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOI
NCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INT
EGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES \"artists\" (Artist
Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"p
laylist_track\"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INT
EGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId,
TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (Playlist
Id) \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\n\n===Additional Context \n\nIn the chinook database invoic
e means order\n\n===Response Guidelines \n1. If the provided context is suff
icient, please generate a valid SQL query without any explanations for the q
uestion. \n2. If the provided context is almost sufficient but requires know
ledge 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 qu
ery 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 m
ost relevant table(s). \n5. If the question has been asked and answered befo
re, please repeat the answer exactly as it was given before. \n"}, {"role":
"user", "content": " \n   List all invoices with a total exceeding $1
0:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total
> 10"}, {"role": "user", "content": " \n   Get the total number of invoice
s for each customer\n"}, {"role": "assistant", "content": "SELECT c.FirstNam
e, c.LastName, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN i
nvoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role":
"user", "content": " \n   Find the total number of invoices per countr
y:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceI
d) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.C
ustomerId\nGROUP BY c.Country"}, {"role": "user", "content": "How many custo
mers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM cus
tomers"}, {"role": "user", "content": "what are the top 5 countries that cus
tomers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT

```
(*) as CustomerCount FROM customers\nGROUP BY Country\nORDER BY CustomerCoun
t DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    List all albums and
their corresponding artist names  \n"}, {"role": "assistant", "content": "SE
LECT al.Title as AlbumTitle, ar.Name as ArtistName\nFROM albums al\nJOIN art
ists ar ON al.ArtistId = ar.ArtistId"}, {"role": "user", "content": "  \n
Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"ro
le": "assistant", "content": "SELECT * FROM tracks WHERE Name LIKE '%Wha
t%'"}, {"role": "user", "content": "Can you list all tables in the SQLite da
tabase catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite
_master WHERE type = 'table'"}, {"role": "user", "content": "  \n    Find al
l invoices since 2010 and the total amount invoiced:\n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:43:16.310165465
Z', 'message': {'role': 'assistant', 'content': "SELECT InvoiceDate, Total F
ROM invoices WHERE InvoiceDate >= '2010-01-01"}, 'done_reason': 'stop', 'don
e': True, 'total_duration': 74330952622, 'load_duration': 22772819, 'prompt_
eval_count': 1789, 'prompt_eval_duration': 66876665000, 'eval_count': 29, 'e
val_duration': 6367274000}
LLM Response: SELECT InvoiceDate, Total FROM invoices WHERE InvoiceDate >=
'2010-01-01
SELECT InvoiceDate, Total FROM invoices WHERE InvoiceDate >= '2010-01-01
Couldn't run sql:  Execution failed on sql 'SELECT InvoiceDate, Total FROM i
nvoices WHERE InvoiceDate >= '2010-01-01': unrecognized token: "'2010-01-01"
```

In [26]:

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

vn.ask(question=question)
```

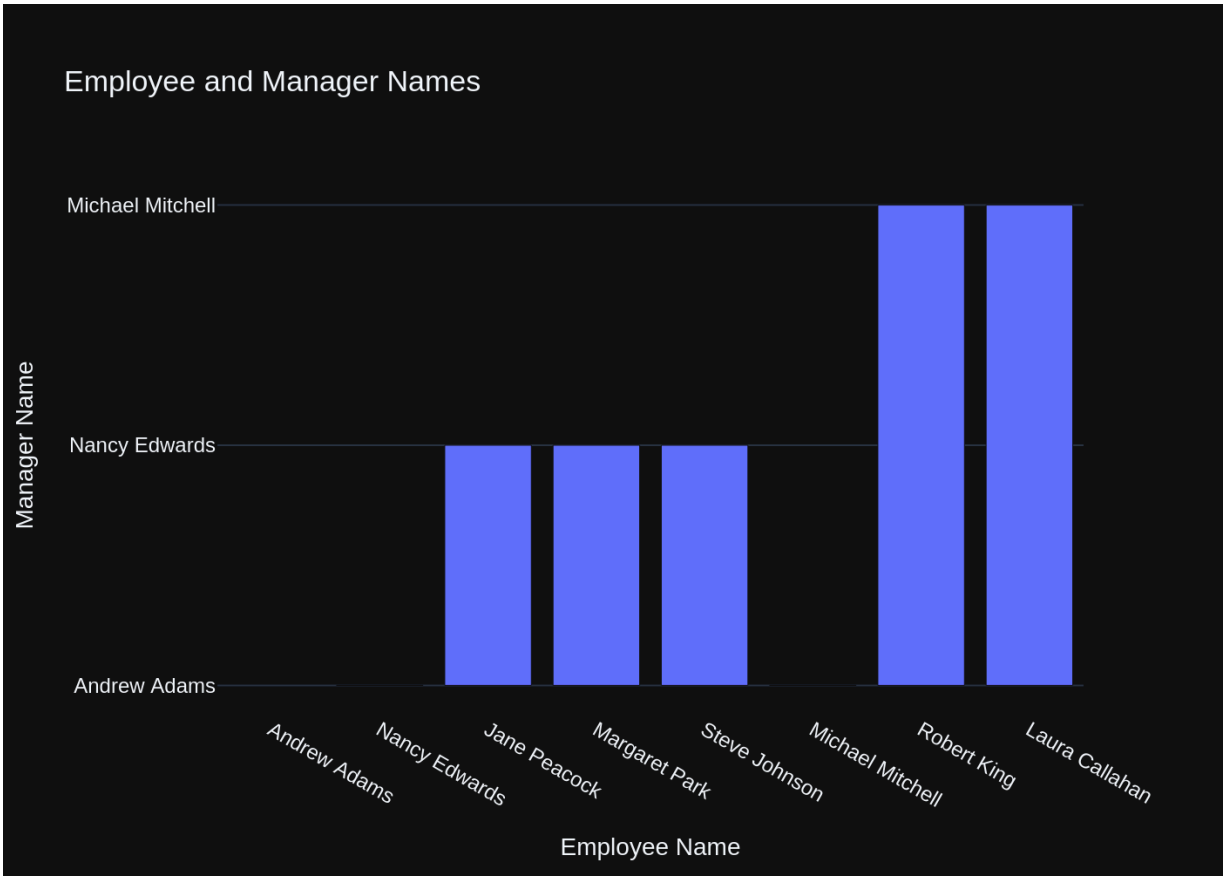
```
Number of requested results 10 is greater than number of elements in index
8, updating n_results = 8
Number of requested results 10 is greater than number of elements in index
1, updating n_results = 1
```

SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE TABLE "employees"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "artists"\r\n(\r\n   ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t

be generated. \n4. Please use the most relevant table(s). \n5. If the questi
on has been asked and answered before, please repeat the answer exactly as i
t was given before. \n'}, {'role': 'user', 'content': ' \n   Get the total
number of invoices for each customer\n'}, {'role': 'assistant', 'content':
'SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as TotalInvoices\nFROM c
ustomers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Custo
merId'}, {'role': 'user', 'content': 'what are the top 5 countries that cust
omers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT
(*) as CustomerCount FROM customers\nGROUP BY Country\nORDER BY CustomerCoun
t DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   List all albums and
their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SE
LECT al.Title as AlbumTitle, ar.Name as ArtistName\nFROM albums al\nJOIN art
ists ar ON al.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n
Find the total number of invoices per country:\n'}, {'role': 'assistant', 'c
ontent': 'SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices\nFROM custom
ers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'},
{'role': 'user', 'content': ' \n   List all invoices with a total exceedin
g $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE T
otal > 10'}, {'role': 'user', 'content': 'How many customers are there'},
{'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role':
'user', 'content': 'Can you list all tables in the SQLite database catalo
g?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE
type = 'table'"}, {'role': 'user', 'content': ' \n   Find all tracks with
a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'cont
ent': "SELECT * FROM tracks WHERE Name LIKE '%What%'"}, {'role': 'user', 'co
ntent': " \n   List all employees and their reporting manager's name (if a
ny):\n"}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE INDEX IFK_EmployeeReportsTo ON \"employees\" (Repo
rtsTo)\n\nCREATE TABLE \"employees\"\r\n(\r\n   EmployeeId INTEGER PRIMARY
KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   F
irstName 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(2
4),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY
(ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTIO
N 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(4
0) 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 N
VARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r
\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60)
NOT NULL,\r\n   SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REF
ERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (Suppo
rtRepId)\n\nCREATE TABLE \"invoices\"\r\n(\r\n   InvoiceId INTEGER PRIMARY
KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   Invo
iceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   Billi

ngCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCountry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"artists\"\r\n(\r\n ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL,\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) as CustomerCount FROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all albums and their corresponding artist names\n"}, {"role": "assistant", "content": "SELECT al.Title as AlbumTitle, ar.Name as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"role": "user", "content": " \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": "user", "content": "Can you list all tables in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type = 'table'"}, {"role": "user", "content": " \n Find all tracks with a name containing \"What\" (case-in

```
sensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks WHERE
Name LIKE '%What%'"}, {"role": "user", "content": "  \n    List all employee
s and their reporting manager's name (if any):\n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:44:31.617876089
Z', 'message': {'role': 'assistant', 'content': "SELECT e.FirstName || ' ' |
| e.LastName as EmployeeName, m.FirstName || ' ' || m.LastName as ManagerNam
e\nFROM employees e\nLEFT JOIN employees m ON e.ReportsTo = m.EmployeeId"},
'done_reason': 'stop', 'done': True, 'total_duration': 75265930718, 'load_du
ration': 23272293, 'prompt_eval_count': 1710, 'prompt_eval_duration': 623498
84000, 'eval_count': 53, 'eval_duration': 11827906000}
LLM Response: SELECT e.FirstName || ' ' || e.LastName as EmployeeName, m.Fir
stName || ' ' || m.LastName as ManagerName
FROM employees e
LEFT JOIN employees m ON e.ReportsTo = m.EmployeeId
SELECT e.FirstName || ' ' || e.LastName as EmployeeName, m.FirstName || ' '
|| m.LastName as ManagerName
FROM employees e
LEFT JOIN employees m ON e.ReportsTo = m.EmployeeId
       EmployeeName       ManagerName
0      Andrew Adams              None
1     Nancy Edwards      Andrew Adams
2      Jane Peacock     Nancy Edwards
3     Margaret Park     Nancy Edwards
4     Steve Johnson     Nancy Edwards
5  Michael Mitchell      Andrew Adams
6       Robert King  Michael Mitchell
7    Laura Callahan  Michael Mitchell
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: '
\n    List all employees and their reporting manager's name (if any):\n'\n\n
The DataFrame was produced using this query: SELECT e.FirstName || ' ' || e.
LastName as EmployeeName, m.FirstName || ' ' || m.LastName as ManagerName\nF
ROM employees e\nLEFT JOIN employees m ON e.ReportsTo = m.EmployeeId\n\nThe
following is information about the resulting pandas DataFrame 'df': \nRunnin
g df.dtypes gives:\n EmployeeName    object\nManagerName     object\ndtype:
object"}, {"role": "user", "content": "Can you generate the Python plotly co
de to chart the results of the dataframe? Assume the data is in a pandas dat
aframe called 'df'. If there is only one value in the dataframe, use an Indi
cator. Respond with only Python code. Do not answer with any explanations --
just the code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:44:53.446525937
Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.ex
press as px\n\nfig = px.bar(df, x='EmployeeName', y='ManagerName', title='Em
ployee and Manager Names')\n\nfig.update_layout(\n    xaxis_title='Employee
Name',\n    yaxis_title='Manager Name'\n)\n\nfig.show()\n```"}, 'done_reaso
n': 'stop', 'done': True, 'total_duration': 21801086631, 'load_duration': 17
252990, 'prompt_eval_count': 215, 'prompt_eval_duration': 7960147000, 'eval_
count': 70, 'eval_duration': 13734251000}
```


Employee and Manager Names
Michael Mitchell
Nancy Edwards
Andrew Adams
Manager Name
Andrew Adams
Nancy Edwards
Jane Peacock
Margaret Park
Steve Johnson
Michael Mitchell
Robert King
Laura Callahan
Employee Name

Out[26]:
```
("SELECT e.FirstName || ' ' || e.LastName as EmployeeName, m.FirstName || '
' || m.LastName as ManagerName\nFROM employees e\nLEFT JOIN employees m ON
e.ReportsTo = m.EmployeeId",
        EmployeeName       ManagerName
 0      Andrew Adams              None
 1     Nancy Edwards      Andrew Adams
 2      Jane Peacock     Nancy Edwards
 3     Margaret Park     Nancy Edwards
 4     Steve Johnson     Nancy Edwards
 5  Michael Mitchell      Andrew Adams
 6       Robert King  Michael Mitchell
 7    Laura Callahan  Michael Mitchell,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'EmployeeName=%{x}<br>ManagerName=%{y}<extr
a></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Andrew Adams', 'Nancy Edwards', 'Jane Peacock',
'Margaret Park',
                           'Steve Johnson', 'Michael Mitchell', 'Robert Kin
g', 'Laura Callahan'],
                          dtype=object),
               'xaxis': 'x',
               'y': array([None, 'Andrew Adams', 'Nancy Edwards', 'Nancy Ed
wards', 'Nancy Edwards',
                           'Andrew Adams', 'Michael Mitchell', 'Michael Mit
chell'], dtype=object),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Employee and Manager Names'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Employee Name'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Manager Name'}}}
 }))
```

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

vn.ask(question=question)
```

```
Number of requested results 10 is greater than number of elements in index
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 the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "customers"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE TABLE "employees"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT *

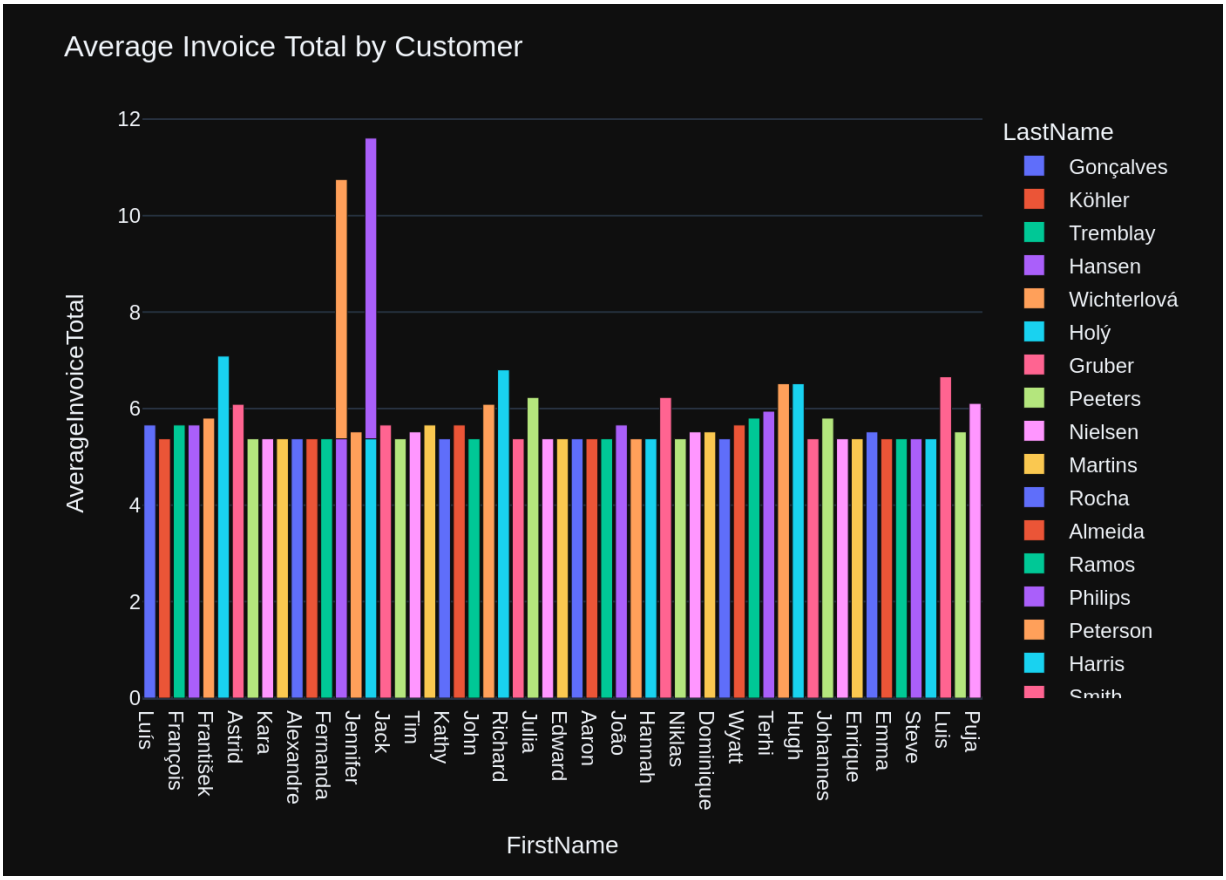
FROM invoices WHERE Total > 10'}, {'role': 'user', 'content': 'How many cust
omers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM cu
stomers'}, {'role': 'user', 'content': 'what are the top 5 countries that cu
stomers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUN
T(*) as CustomerCount FROM customers\nGROUP BY Country\nORDER BY CustomerCou
nt DESC\nLIMIT 5'}, {'role': 'user', 'content': " \n   List all employees
and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'con
tent': "SELECT e.FirstName || ' ' || e.LastName as EmployeeName, m.FirstName
|| ' ' || m.LastName as ManagerName\nFROM employees e\nLEFT JOIN employees m
ON e.ReportsTo = m.EmployeeId"}, {'role': 'user', 'content': ' \n   List a
ll albums and their corresponding artist names \n'}, {'role': 'assistant',
'content': 'SELECT al.Title as AlbumTitle, ar.Name as ArtistName\nFROM album
s al\nJOIN artists ar ON al.ArtistId = ar.ArtistId'}, {'role': 'user', 'cont
ent': ' \n   Find all tracks with a name containing "What" (case-insensiti
ve)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name L
IKE '%What%'"}, {'role': 'user', 'content': 'Can you list all tables in the
SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT name FR
OM sqlite_master WHERE type = 'table'"}, {'role': 'user', 'content': ' \n
Get the average invoice total for each customer:\n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n   InvoiceId INTEGER P
RIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n
InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   B
illingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCou
ntry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMER
IC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES \"customers\"
(CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE
INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK
_InvoiceLineInvoiceId ON \"invoice_items\" (InvoiceId)\n\nCREATE TABLE \"inv
oice_items\"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT
NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NUL
L,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NU
LL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n
\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) RE
FERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO
N\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)
\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\nCREATE INDEX IFK_CustomerSupp
ortRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"customers\"\r\n
(\r\n   CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   First
Name NVARCHAR(40) NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n
Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(4
0),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCod
e NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   E
mail NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r\n   FOREIGN KEY
(SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO AC
TION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON \"emp
loyees\" (ReportsTo)\n\nCREATE TABLE \"employees\"\r\n(\r\n   EmployeeId IN
TEGER 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 DATETIM
E,\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),\r\n
FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DEL
ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the
chinook database invoice means order\n\n===Response Guidelines \n1. If the p
rovided context is sufficient, please generate a valid SQL query without any
explanations for the question. \n2. If the provided context is almost suffic
ient 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.
\n4. Please use the most relevant table(s). \n5. If the question has been as
ked and answered before, please repeat the answer exactly as it was given be
fore. \n"}, {"role": "user", "content": " \n   Get the total number of inv
oices for each customer\n"}, {"role": "assistant", "content": "SELECT c.Firs
tName, c.LastName, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJO
IN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"rol
e": "user", "content": " \n   Find the total number of invoices per countr
y:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceI
d) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.C
ustomerId\nGROUP BY c.Country"}, {"role": "user", "content": " \n   List a
ll invoices with a total exceeding $10:\n"}, {"role": "assistant", "conten
t": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "content":
"How many customers are there"}, {"role": "assistant", "content": "SELECT CO
UNT(*) FROM customers"}, {"role": "user", "content": "what are the top 5 cou
ntries that customers come from?"}, {"role": "assistant", "content": "SELECT
Country, COUNT(*) as CustomerCount FROM customers\nGROUP BY Country\nORDER B
Y CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": " \n   List a
ll employees and their reporting manager's name (if any):\n"}, {"role": "ass
istant", "content": "SELECT e.FirstName || ' ' || e.LastName as EmployeeNam
e, m.FirstName || ' ' || m.LastName as ManagerName\nFROM employees e\nLEFT J
OIN employees m ON e.ReportsTo = m.EmployeeId"}, {"role": "user", "content":
" \n   List all albums and their corresponding artist names \n"}, {"rol
e": "assistant", "content": "SELECT al.Title as AlbumTitle, ar.Name as Artis
tName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.ArtistId"}, {"rol
e": "user", "content": " \n   Find all tracks with a name containing \"Wha
t\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM
tracks WHERE Name LIKE '%What%'"}, {"role": "user", "content": "Can you list
all tables in the SQLite database catalog?"}, {"role": "assistant", "conten
t": "SELECT name FROM sqlite_master WHERE type = 'table'"}, {"role": "user",
"content": " \n   Get the average invoice total for each customer:\n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:46:03.239297093
Z', 'message': {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastN
ame, AVG(i.Total) as AverageInvoiceTotal\nFROM customers c\nJOIN invoices i
ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, 'done_reason': 'sto
p', 'done': True, 'total_duration': 69694870440, 'load_duration': 22032222,
'prompt_eval_count': 1572, 'prompt_eval_duration': 58193636000, 'eval_coun
t': 48, 'eval_duration': 10269038000}
LLM Response: SELECT c.FirstName, c.LastName, AVG(i.Total) as AverageInvoice
Total
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId

```
SELECT c.FirstName, c.LastName, AVG(i.Total) as AverageInvoiceTotal
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
```

	FirstName	LastName	AverageInvoiceTotal
0	Luís	Gonçalves	5.660000
1	Leonie	Köhler	5.374286
2	François	Tremblay	5.660000
3	Bjørn	Hansen	5.660000
4	František	Wichterlová	5.802857
5	Helena	Holý	7.088571
6	Astrid	Gruber	6.088571
7	Daan	Peeters	5.374286
8	Kara	Nielsen	5.374286
9	Eduardo	Martins	5.374286
10	Alexandre	Rocha	5.374286
11	Roberto	Almeida	5.374286
12	Fernanda	Ramos	5.374286
13	Mark	Philips	5.374286
14	Jennifer	Peterson	5.517143
15	Frank	Harris	5.374286
16	Jack	Smith	5.660000
17	Michelle	Brooks	5.374286
18	Tim	Goyer	5.517143
19	Dan	Miller	5.660000
20	Kathy	Chase	5.374286
21	Heather	Leacock	5.660000
22	John	Gordon	5.374286
23	Frank	Ralston	6.231429
24	Victor	Stevens	6.088571
25	Richard	Cunningham	6.802857
26	Patrick	Gray	5.374286
27	Julia	Barnett	6.231429
28	Robert	Brown	5.374286
29	Edward	Francis	5.374286
30	Martha	Silk	5.374286
31	Aaron	Mitchell	5.374286
32	Ellie	Sullivan	5.374286
33	João	Fernandes	5.660000
34	Madalena	Sampaio	5.374286
35	Hannah	Schneider	5.374286
36	Fynn	Zimmermann	6.231429
37	Niklas	Schröder	5.374286
38	Camille	Bernard	5.517143
39	Dominique	Lefebvre	5.517143
40	Marc	Dubois	5.374286
41	Wyatt	Girard	5.660000
42	Isabelle	Mercier	5.802857
43	Terhi	Hämäläinen	5.945714
44	Ladislav	Kovács	6.517143
45	Hugh	O'Reilly	6.517143
46	Lucas	Mancini	5.374286
47	Johannes	Van der Berg	5.802857
48	Stanisław	Wójcik	5.374286
49	Enrique	Muñoz	5.374286
50	Joakim	Johansson	5.517143

```
51       Emma         Jones             5.374286
52       Phil        Hughes             5.374286
53      Steve        Murray             5.374286
54       Mark        Taylor             5.374286
55      Diego     Gutiérrez             5.374286
56       Luis         Rojas             6.660000
57      Manoj        Pareek             5.517143
58       Puja    Srivastava             6.106667
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: '
\n    Get the average invoice total for each customer:\n'\n\nThe DataFrame w
as produced using this query: SELECT c.FirstName, c.LastName, AVG(i.Total) a
s AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId =
i.CustomerId\nGROUP BY c.CustomerId\n\nThe following is information about th
e resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n FirstName
object\nLastName                object\nAverageInvoiceTotal    float64\ndtyp
e: object"}, {"role": "user", "content": "Can you generate the Python plotly
code to chart the results of the dataframe? Assume the data is in a pandas d
ataframe called 'df'. If there is only one value in the dataframe, use an In
dicator. Respond with only Python code. Do not answer with any explanations
-- just the code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:46:21.057063517
Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.ex
press as px\n\nfig = px.bar(df, x='FirstName', y='AverageInvoiceTotal', colo
r='LastName')\n\nfig.update_layout(title='Average Invoice Total by Custome
r')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_durati
on': 17796126499, 'load_duration': 29457566, 'prompt_eval_count': 211, 'prom
pt_eval_duration': 7038810000, 'eval_count': 55, 'eval_duration': 1059422900
0}
```


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

Out[27]: ('SELECT c.FirstName, c.LastName, AVG(i.Total) as AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId',

	FirstName	LastName	AverageInvoiceTotal
0	Luís	Gonçalves	5.660000
1	Leonie	Köhler	5.374286
2	François	Tremblay	5.660000
3	Bjørn	Hansen	5.660000
4	František	Wichterlová	5.802857
5	Helena	Holý	7.088571
6	Astrid	Gruber	6.088571
7	Daan	Peeters	5.374286
8	Kara	Nielsen	5.374286
9	Eduardo	Martins	5.374286
10	Alexandre	Rocha	5.374286
11	Roberto	Almeida	5.374286
12	Fernanda	Ramos	5.374286
13	Mark	Philips	5.374286
14	Jennifer	Peterson	5.517143
15	Frank	Harris	5.374286
16	Jack	Smith	5.660000
17	Michelle	Brooks	5.374286
18	Tim	Goyer	5.517143
19	Dan	Miller	5.660000
20	Kathy	Chase	5.374286
21	Heather	Leacock	5.660000
22	John	Gordon	5.374286
23	Frank	Ralston	6.231429
24	Victor	Stevens	6.088571
25	Richard	Cunningham	6.802857
26	Patrick	Gray	5.374286
27	Julia	Barnett	6.231429
28	Robert	Brown	5.374286
29	Edward	Francis	5.374286
30	Martha	Silk	5.374286
31	Aaron	Mitchell	5.374286
32	Ellie	Sullivan	5.374286
33	João	Fernandes	5.660000
34	Madalena	Sampaio	5.374286
35	Hannah	Schneider	5.374286
36	Fynn	Zimmermann	6.231429
37	Niklas	Schröder	5.374286
38	Camille	Bernard	5.517143
39	Dominique	Lefebvre	5.517143
40	Marc	Dubois	5.374286
41	Wyatt	Girard	5.660000
42	Isabelle	Mercier	5.802857
43	Terhi	Hämäläinen	5.945714
44	Ladislav	Kovács	6.517143
45	Hugh	O'Reilly	6.517143
46	Lucas	Mancini	5.374286
47	Johannes	Van der Berg	5.802857
48	Stanisław	Wójcik	5.374286
49	Enrique	Muñoz	5.374286
50	Joakim	Johansson	5.517143
51	Emma	Jones	5.374286

```
 52       Phil        Hughes             5.374286
 53      Steve        Murray             5.374286
 54       Mark        Taylor             5.374286
 55      Diego     Gutiérrez             5.374286
 56       Luis         Rojas             6.660000
 57      Manoj        Pareek             5.517143
 58       Puja    Srivastava             6.106667,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Gonçalves<br>FirstName=%{x}<br>Av
erageInvoiceTotal=%{y}<extra></extra>',
               'legendgroup': 'Gonçalves',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': 'Gonçalves',
               'offsetgroup': 'Gonçalves',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Luís'], dtype=object),
               'xaxis': 'x',
               'y': array([5.66]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Köhler<br>FirstName=%{x}<br>Avera
geInvoiceTotal=%{y}<extra></extra>',
               'legendgroup': 'Köhler',
               'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
               'name': 'Köhler',
               'offsetgroup': 'Köhler',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Leonie'], dtype=object),
               'xaxis': 'x',
               'y': array([5.37428571]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Tremblay<br>FirstName=%{x}<br>Ave
rageInvoiceTotal=%{y}<extra></extra>',
               'legendgroup': 'Tremblay',
               'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
               'name': 'Tremblay',
               'offsetgroup': 'Tremblay',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['François'], dtype=object),
               'xaxis': 'x',
               'y': array([5.66]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Hansen<br>FirstName=%{x}<br>Avera
geInvoiceTotal=%{y}<extra></extra>',
```

```
                    'legendgroup': 'Hansen',
                    'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
                    'name': 'Hansen',
                    'offsetgroup': 'Hansen',
                    'orientation': 'v',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['Bjørn'], dtype=object),
                    'xaxis': 'x',
                    'y': array([5.66]),
                    'yaxis': 'y'},
                   {'alignmentgroup': 'True',
                    'hovertemplate': ('LastName=Wichterlová<br>FirstN' ... 'voic
eTotal=%{y}<extra></extra>'),
                    'legendgroup': 'Wichterlová',
                    'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
                    'name': 'Wichterlová',
                    'offsetgroup': 'Wichterlová',
                    'orientation': 'v',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['František'], dtype=object),
                    'xaxis': 'x',
                    'y': array([5.80285714]),
                    'yaxis': 'y'},
                   {'alignmentgroup': 'True',
                    'hovertemplate': 'LastName=Holý<br>FirstName=%{x}<br>Average
InvoiceTotal=%{y}<extra></extra>',
                    'legendgroup': 'Holý',
                    'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
                    'name': 'Holý',
                    'offsetgroup': 'Holý',
                    'orientation': 'v',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['Helena'], dtype=object),
                    'xaxis': 'x',
                    'y': array([7.08857143]),
                    'yaxis': 'y'},
                   {'alignmentgroup': 'True',
                    'hovertemplate': 'LastName=Gruber<br>FirstName=%{x}<br>Avera
geInvoiceTotal=%{y}<extra></extra>',
                    'legendgroup': 'Gruber',
                    'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
                    'name': 'Gruber',
                    'offsetgroup': 'Gruber',
                    'orientation': 'v',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['Astrid'], dtype=object),
                    'xaxis': 'x',
                    'y': array([6.08857143]),
```

```
                'yaxis': 'y'},
               {'alignmentgroup': 'True',
                'hovertemplate': 'LastName=Peeters<br>FirstName=%{x}<br>Aver
ageInvoiceTotal=%{y}<extra></extra>',
                'legendgroup': 'Peeters',
                'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
                'name': 'Peeters',
                'offsetgroup': 'Peeters',
                'orientation': 'v',
                'showlegend': True,
                'textposition': 'auto',
                'type': 'bar',
                'x': array(['Daan'], dtype=object),
                'xaxis': 'x',
                'y': array([5.37428571]),
                'yaxis': 'y'},
               {'alignmentgroup': 'True',
                'hovertemplate': 'LastName=Nielsen<br>FirstName=%{x}<br>Aver
ageInvoiceTotal=%{y}<extra></extra>',
                'legendgroup': 'Nielsen',
                'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
                'name': 'Nielsen',
                'offsetgroup': 'Nielsen',
                'orientation': 'v',
                'showlegend': True,
                'textposition': 'auto',
                'type': 'bar',
                'x': array(['Kara'], dtype=object),
                'xaxis': 'x',
                'y': array([5.37428571]),
                'yaxis': 'y'},
               {'alignmentgroup': 'True',
                'hovertemplate': 'LastName=Martins<br>FirstName=%{x}<br>Aver
ageInvoiceTotal=%{y}<extra></extra>',
                'legendgroup': 'Martins',
                'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
                'name': 'Martins',
                'offsetgroup': 'Martins',
                'orientation': 'v',
                'showlegend': True,
                'textposition': 'auto',
                'type': 'bar',
                'x': array(['Eduardo'], dtype=object),
                'xaxis': 'x',
                'y': array([5.37428571]),
                'yaxis': 'y'},
               {'alignmentgroup': 'True',
                'hovertemplate': 'LastName=Rocha<br>FirstName=%{x}<br>Averag
eInvoiceTotal=%{y}<extra></extra>',
                'legendgroup': 'Rocha',
                'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                'name': 'Rocha',
                'offsetgroup': 'Rocha',
                'orientation': 'v',
                'showlegend': True,
                'textposition': 'auto',
```

```
                  'type': 'bar',
                  'x': array(['Alexandre'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.37428571]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Almeida<br>FirstName=%{x}<br>Aver
ageInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Almeida',
                  'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
                  'name': 'Almeida',
                  'offsetgroup': 'Almeida',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Roberto'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.37428571]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Ramos<br>FirstName=%{x}<br>Averag
eInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Ramos',
                  'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
                  'name': 'Ramos',
                  'offsetgroup': 'Ramos',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Fernanda'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.37428571]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Philips<br>FirstName=%{x}<br>Aver
ageInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Philips',
                  'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
                  'name': 'Philips',
                  'offsetgroup': 'Philips',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Mark'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.37428571]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Peterson<br>FirstName=%{x}<br>Ave
rageInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Peterson',
                  'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
                  'name': 'Peterson',
```

```
                    'offsetgroup': 'Peterson',
                    'orientation': 'v',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['Jennifer'], dtype=object),
                    'xaxis': 'x',
                    'y': array([5.51714286]),
                    'yaxis': 'y'},
                   {'alignmentgroup': 'True',
                    'hovertemplate': 'LastName=Harris<br>FirstName=%{x}<br>Avera
geInvoiceTotal=%{y}<extra></extra>',
                    'legendgroup': 'Harris',
                    'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
                    'name': 'Harris',
                    'offsetgroup': 'Harris',
                    'orientation': 'v',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['Frank'], dtype=object),
                    'xaxis': 'x',
                    'y': array([5.37428571]),
                    'yaxis': 'y'},
                   {'alignmentgroup': 'True',
                    'hovertemplate': 'LastName=Smith<br>FirstName=%{x}<br>Averag
eInvoiceTotal=%{y}<extra></extra>',
                    'legendgroup': 'Smith',
                    'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
                    'name': 'Smith',
                    'offsetgroup': 'Smith',
                    'orientation': 'v',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['Jack'], dtype=object),
                    'xaxis': 'x',
                    'y': array([5.66]),
                    'yaxis': 'y'},
                   {'alignmentgroup': 'True',
                    'hovertemplate': 'LastName=Brooks<br>FirstName=%{x}<br>Avera
geInvoiceTotal=%{y}<extra></extra>',
                    'legendgroup': 'Brooks',
                    'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
                    'name': 'Brooks',
                    'offsetgroup': 'Brooks',
                    'orientation': 'v',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['Michelle'], dtype=object),
                    'xaxis': 'x',
                    'y': array([5.37428571]),
                    'yaxis': 'y'},
                   {'alignmentgroup': 'True',
                    'hovertemplate': 'LastName=Goyer<br>FirstName=%{x}<br>Averag
```

```
eInvoiceTotal=%{y}<extra></extra>',
                'legendgroup': 'Goyer',
                'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
                'name': 'Goyer',
                'offsetgroup': 'Goyer',
                'orientation': 'v',
                'showlegend': True,
                'textposition': 'auto',
                'type': 'bar',
                'x': array(['Tim'], dtype=object),
                'xaxis': 'x',
                'y': array([5.51714286]),
                'yaxis': 'y'},
               {'alignmentgroup': 'True',
                'hovertemplate': 'LastName=Miller<br>FirstName=%{x}<br>Avera
geInvoiceTotal=%{y}<extra></extra>',
                'legendgroup': 'Miller',
                'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
                'name': 'Miller',
                'offsetgroup': 'Miller',
                'orientation': 'v',
                'showlegend': True,
                'textposition': 'auto',
                'type': 'bar',
                'x': array(['Dan'], dtype=object),
                'xaxis': 'x',
                'y': array([5.66]),
                'yaxis': 'y'},
               {'alignmentgroup': 'True',
                'hovertemplate': 'LastName=Chase<br>FirstName=%{x}<br>Averag
eInvoiceTotal=%{y}<extra></extra>',
                'legendgroup': 'Chase',
                'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                'name': 'Chase',
                'offsetgroup': 'Chase',
                'orientation': 'v',
                'showlegend': True,
                'textposition': 'auto',
                'type': 'bar',
                'x': array(['Kathy'], dtype=object),
                'xaxis': 'x',
                'y': array([5.37428571]),
                'yaxis': 'y'},
               {'alignmentgroup': 'True',
                'hovertemplate': 'LastName=Leacock<br>FirstName=%{x}<br>Aver
ageInvoiceTotal=%{y}<extra></extra>',
                'legendgroup': 'Leacock',
                'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
                'name': 'Leacock',
                'offsetgroup': 'Leacock',
                'orientation': 'v',
                'showlegend': True,
                'textposition': 'auto',
                'type': 'bar',
                'x': array(['Heather'], dtype=object),
                'xaxis': 'x',
```

```
                 'y': array([5.66]),
                 'yaxis': 'y'},
                {'alignmentgroup': 'True',
                 'hovertemplate': 'LastName=Gordon<br>FirstName=%{x}<br>Avera
geInvoiceTotal=%{y}<extra></extra>',
                 'legendgroup': 'Gordon',
                 'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
                 'name': 'Gordon',
                 'offsetgroup': 'Gordon',
                 'orientation': 'v',
                 'showlegend': True,
                 'textposition': 'auto',
                 'type': 'bar',
                 'x': array(['John'], dtype=object),
                 'xaxis': 'x',
                 'y': array([5.37428571]),
                 'yaxis': 'y'},
                {'alignmentgroup': 'True',
                 'hovertemplate': 'LastName=Ralston<br>FirstName=%{x}<br>Aver
ageInvoiceTotal=%{y}<extra></extra>',
                 'legendgroup': 'Ralston',
                 'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
                 'name': 'Ralston',
                 'offsetgroup': 'Ralston',
                 'orientation': 'v',
                 'showlegend': True,
                 'textposition': 'auto',
                 'type': 'bar',
                 'x': array(['Frank'], dtype=object),
                 'xaxis': 'x',
                 'y': array([6.23142857]),
                 'yaxis': 'y'},
                {'alignmentgroup': 'True',
                 'hovertemplate': 'LastName=Stevens<br>FirstName=%{x}<br>Aver
ageInvoiceTotal=%{y}<extra></extra>',
                 'legendgroup': 'Stevens',
                 'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
                 'name': 'Stevens',
                 'offsetgroup': 'Stevens',
                 'orientation': 'v',
                 'showlegend': True,
                 'textposition': 'auto',
                 'type': 'bar',
                 'x': array(['Victor'], dtype=object),
                 'xaxis': 'x',
                 'y': array([6.08857143]),
                 'yaxis': 'y'},
                {'alignmentgroup': 'True',
                 'hovertemplate': 'LastName=Cunningham<br>FirstName=%{x}<br>A
verageInvoiceTotal=%{y}<extra></extra>',
                 'legendgroup': 'Cunningham',
                 'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
                 'name': 'Cunningham',
                 'offsetgroup': 'Cunningham',
                 'orientation': 'v',
                 'showlegend': True,
```

```
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Richard'], dtype=object),
                  'xaxis': 'x',
                  'y': array([6.80285714]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Gray<br>FirstName=%{x}<br>Average
InvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Gray',
                  'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
                  'name': 'Gray',
                  'offsetgroup': 'Gray',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Patrick'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.37428571]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Barnett<br>FirstName=%{x}<br>Aver
ageInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Barnett',
                  'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
                  'name': 'Barnett',
                  'offsetgroup': 'Barnett',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Julia'], dtype=object),
                  'xaxis': 'x',
                  'y': array([6.23142857]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Brown<br>FirstName=%{x}<br>Averag
eInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Brown',
                  'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
                  'name': 'Brown',
                  'offsetgroup': 'Brown',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Robert'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.37428571]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Francis<br>FirstName=%{x}<br>Aver
ageInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Francis',
                  'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
```

```
                    'name': 'Francis',
                    'offsetgroup': 'Francis',
                    'orientation': 'v',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['Edward'], dtype=object),
                    'xaxis': 'x',
                    'y': array([5.37428571]),
                    'yaxis': 'y'},
                   {'alignmentgroup': 'True',
                    'hovertemplate': 'LastName=Silk<br>FirstName=%{x}<br>Average
InvoiceTotal=%{y}<extra></extra>',
                    'legendgroup': 'Silk',
                    'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                    'name': 'Silk',
                    'offsetgroup': 'Silk',
                    'orientation': 'v',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['Martha'], dtype=object),
                    'xaxis': 'x',
                    'y': array([5.37428571]),
                    'yaxis': 'y'},
                   {'alignmentgroup': 'True',
                    'hovertemplate': 'LastName=Mitchell<br>FirstName=%{x}<br>Ave
rageInvoiceTotal=%{y}<extra></extra>',
                    'legendgroup': 'Mitchell',
                    'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
                    'name': 'Mitchell',
                    'offsetgroup': 'Mitchell',
                    'orientation': 'v',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['Aaron'], dtype=object),
                    'xaxis': 'x',
                    'y': array([5.37428571]),
                    'yaxis': 'y'},
                   {'alignmentgroup': 'True',
                    'hovertemplate': 'LastName=Sullivan<br>FirstName=%{x}<br>Ave
rageInvoiceTotal=%{y}<extra></extra>',
                    'legendgroup': 'Sullivan',
                    'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
                    'name': 'Sullivan',
                    'offsetgroup': 'Sullivan',
                    'orientation': 'v',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['Ellie'], dtype=object),
                    'xaxis': 'x',
                    'y': array([5.37428571]),
                    'yaxis': 'y'},
                   {'alignmentgroup': 'True',
```

```
                'hovertemplate': 'LastName=Fernandes<br>FirstName=%{x}<br>Av
erageInvoiceTotal=%{y}<extra></extra>',
                'legendgroup': 'Fernandes',
                'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
                'name': 'Fernandes',
                'offsetgroup': 'Fernandes',
                'orientation': 'v',
                'showlegend': True,
                'textposition': 'auto',
                'type': 'bar',
                'x': array(['João'], dtype=object),
                'xaxis': 'x',
                'y': array([5.66]),
                'yaxis': 'y'},
               {'alignmentgroup': 'True',
                'hovertemplate': 'LastName=Sampaio<br>FirstName=%{x}<br>Aver
ageInvoiceTotal=%{y}<extra></extra>',
                'legendgroup': 'Sampaio',
                'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
                'name': 'Sampaio',
                'offsetgroup': 'Sampaio',
                'orientation': 'v',
                'showlegend': True,
                'textposition': 'auto',
                'type': 'bar',
                'x': array(['Madalena'], dtype=object),
                'xaxis': 'x',
                'y': array([5.37428571]),
                'yaxis': 'y'},
               {'alignmentgroup': 'True',
                'hovertemplate': 'LastName=Schneider<br>FirstName=%{x}<br>Av
erageInvoiceTotal=%{y}<extra></extra>',
                'legendgroup': 'Schneider',
                'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
                'name': 'Schneider',
                'offsetgroup': 'Schneider',
                'orientation': 'v',
                'showlegend': True,
                'textposition': 'auto',
                'type': 'bar',
                'x': array(['Hannah'], dtype=object),
                'xaxis': 'x',
                'y': array([5.37428571]),
                'yaxis': 'y'},
               {'alignmentgroup': 'True',
                'hovertemplate': 'LastName=Zimmermann<br>FirstName=%{x}<br>A
verageInvoiceTotal=%{y}<extra></extra>',
                'legendgroup': 'Zimmermann',
                'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
                'name': 'Zimmermann',
                'offsetgroup': 'Zimmermann',
                'orientation': 'v',
                'showlegend': True,
                'textposition': 'auto',
                'type': 'bar',
                'x': array(['Fynn'], dtype=object),
```

```
                  'xaxis': 'x',
                  'y': array([6.23142857]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Schröder<br>FirstName=%{x}<br>Ave
rageInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Schröder',
                  'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
                  'name': 'Schröder',
                  'offsetgroup': 'Schröder',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Niklas'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.37428571]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Bernard<br>FirstName=%{x}<br>Aver
ageInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Bernard',
                  'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
                  'name': 'Bernard',
                  'offsetgroup': 'Bernard',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Camille'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.51714286]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Lefebvre<br>FirstName=%{x}<br>Ave
rageInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Lefebvre',
                  'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
                  'name': 'Lefebvre',
                  'offsetgroup': 'Lefebvre',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Dominique'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.51714286]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Dubois<br>FirstName=%{x}<br>Avera
geInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Dubois',
                  'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                  'name': 'Dubois',
                  'offsetgroup': 'Dubois',
                  'orientation': 'v',
```

```
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Marc'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.37428571]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Girard<br>FirstName=%{x}<br>Avera
geInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Girard',
                  'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
                  'name': 'Girard',
                  'offsetgroup': 'Girard',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Wyatt'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.66]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Mercier<br>FirstName=%{x}<br>Aver
ageInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Mercier',
                  'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
                  'name': 'Mercier',
                  'offsetgroup': 'Mercier',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Isabelle'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.80285714]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Hämäläinen<br>FirstName=%{x}<br>A
verageInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Hämäläinen',
                  'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
                  'name': 'Hämäläinen',
                  'offsetgroup': 'Hämäläinen',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Terhi'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.94571429]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Kovács<br>FirstName=%{x}<br>Avera
geInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Kovács',
```

```
                    'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
                    'name': 'Kovács',
                    'offsetgroup': 'Kovács',
                    'orientation': 'v',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['Ladislav'], dtype=object),
                    'xaxis': 'x',
                    'y': array([6.51714286]),
                    'yaxis': 'y'},
                   {'alignmentgroup': 'True',
                    'hovertemplate': "LastName=O'Reilly<br>FirstName=%{x}<br>Ave
rageInvoiceTotal=%{y}<extra></extra>",
                    'legendgroup': "O'Reilly",
                    'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
                    'name': "O'Reilly",
                    'offsetgroup': "O'Reilly",
                    'orientation': 'v',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['Hugh'], dtype=object),
                    'xaxis': 'x',
                    'y': array([6.51714286]),
                    'yaxis': 'y'},
                   {'alignmentgroup': 'True',
                    'hovertemplate': 'LastName=Mancini<br>FirstName=%{x}<br>Aver
ageInvoiceTotal=%{y}<extra></extra>',
                    'legendgroup': 'Mancini',
                    'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
                    'name': 'Mancini',
                    'offsetgroup': 'Mancini',
                    'orientation': 'v',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['Lucas'], dtype=object),
                    'xaxis': 'x',
                    'y': array([5.37428571]),
                    'yaxis': 'y'},
                   {'alignmentgroup': 'True',
                    'hovertemplate': ('LastName=Van der Berg<br>First' ... 'voic
eTotal=%{y}<extra></extra>'),
                    'legendgroup': 'Van der Berg',
                    'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
                    'name': 'Van der Berg',
                    'offsetgroup': 'Van der Berg',
                    'orientation': 'v',
                    'showlegend': True,
                    'textposition': 'auto',
                    'type': 'bar',
                    'x': array(['Johannes'], dtype=object),
                    'xaxis': 'x',
                    'y': array([5.80285714]),
                    'yaxis': 'y'},
```

```
                {'alignmentgroup': 'True',
                 'hovertemplate': 'LastName=Wójcik<br>FirstName=%{x}<br>Avera
geInvoiceTotal=%{y}<extra></extra>',
                 'legendgroup': 'Wójcik',
                 'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
                 'name': 'Wójcik',
                 'offsetgroup': 'Wójcik',
                 'orientation': 'v',
                 'showlegend': True,
                 'textposition': 'auto',
                 'type': 'bar',
                 'x': array(['Stanisław'], dtype=object),
                 'xaxis': 'x',
                 'y': array([5.37428571]),
                 'yaxis': 'y'},
                {'alignmentgroup': 'True',
                 'hovertemplate': 'LastName=Muñoz<br>FirstName=%{x}<br>Averag
eInvoiceTotal=%{y}<extra></extra>',
                 'legendgroup': 'Muñoz',
                 'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
                 'name': 'Muñoz',
                 'offsetgroup': 'Muñoz',
                 'orientation': 'v',
                 'showlegend': True,
                 'textposition': 'auto',
                 'type': 'bar',
                 'x': array(['Enrique'], dtype=object),
                 'xaxis': 'x',
                 'y': array([5.37428571]),
                 'yaxis': 'y'},
                {'alignmentgroup': 'True',
                 'hovertemplate': 'LastName=Johansson<br>FirstName=%{x}<br>Av
erageInvoiceTotal=%{y}<extra></extra>',
                 'legendgroup': 'Johansson',
                 'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                 'name': 'Johansson',
                 'offsetgroup': 'Johansson',
                 'orientation': 'v',
                 'showlegend': True,
                 'textposition': 'auto',
                 'type': 'bar',
                 'x': array(['Joakim'], dtype=object),
                 'xaxis': 'x',
                 'y': array([5.51714286]),
                 'yaxis': 'y'},
                {'alignmentgroup': 'True',
                 'hovertemplate': 'LastName=Jones<br>FirstName=%{x}<br>Averag
eInvoiceTotal=%{y}<extra></extra>',
                 'legendgroup': 'Jones',
                 'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
                 'name': 'Jones',
                 'offsetgroup': 'Jones',
                 'orientation': 'v',
                 'showlegend': True,
                 'textposition': 'auto',
                 'type': 'bar',
```

```
                  'x': array(['Emma'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.37428571]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Hughes<br>FirstName=%{x}<br>Avera
geInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Hughes',
                  'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
                  'name': 'Hughes',
                  'offsetgroup': 'Hughes',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Phil'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.37428571]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Murray<br>FirstName=%{x}<br>Avera
geInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Murray',
                  'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
                  'name': 'Murray',
                  'offsetgroup': 'Murray',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Steve'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.37428571]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Taylor<br>FirstName=%{x}<br>Avera
geInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Taylor',
                  'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
                  'name': 'Taylor',
                  'offsetgroup': 'Taylor',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Mark'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.37428571]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Gutiérrez<br>FirstName=%{x}<br>Av
erageInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Gutiérrez',
                  'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
                  'name': 'Gutiérrez',
                  'offsetgroup': 'Gutiérrez',
```

```
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Diego'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.37428571]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Rojas<br>FirstName=%{x}<br>Averag
eInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Rojas',
                  'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
                  'name': 'Rojas',
                  'offsetgroup': 'Rojas',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Luis'], dtype=object),
                  'xaxis': 'x',
                  'y': array([6.66]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Pareek<br>FirstName=%{x}<br>Avera
geInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Pareek',
                  'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
                  'name': 'Pareek',
                  'offsetgroup': 'Pareek',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Manoj'], dtype=object),
                  'xaxis': 'x',
                  'y': array([5.51714286]),
                  'yaxis': 'y'},
                 {'alignmentgroup': 'True',
                  'hovertemplate': 'LastName=Srivastava<br>FirstName=%{x}<br>A
verageInvoiceTotal=%{y}<extra></extra>',
                  'legendgroup': 'Srivastava',
                  'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
                  'name': 'Srivastava',
                  'offsetgroup': 'Srivastava',
                  'orientation': 'v',
                  'showlegend': True,
                  'textposition': 'auto',
                  'type': 'bar',
                  'x': array(['Puja'], dtype=object),
                  'xaxis': 'x',
                  'y': array([6.10666667]),
                  'yaxis': 'y'}],
      'layout': {'barmode': 'relative',
                 'legend': {'title': {'text': 'LastName'}, 'tracegroupgap':
0},
```

```
                'margin': {'t': 60},
                'template': '...',
                'title': {'text': 'Average Invoice Total by Customer'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'FirstName'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'AverageInvoiceTotal'}}}
 }))
```

In [28]:

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

vn.ask(question=question)
```

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

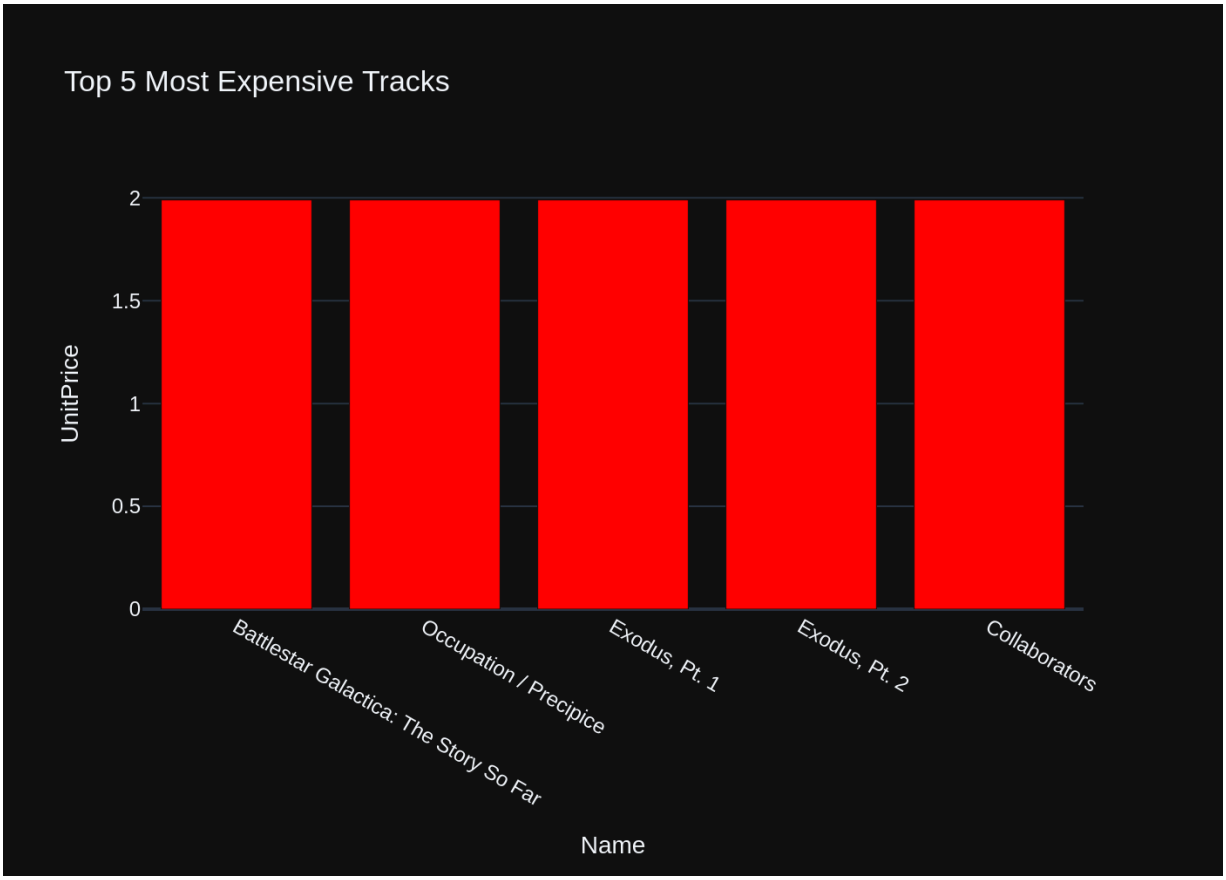
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER 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 INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "albums"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   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 WHERE Total > 10'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT al.Title as AlbumTitle, ar.Name as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, AVG(i.Total) as AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) as Customer

Count FROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT
5'}, {'role': 'user', 'content': ' \n   Find the total number of invoices
per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT
(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.Custo
merId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': '
\n   Get the total number of invoices for each customer\n'}, {'role': 'assi
stant', 'content': 'SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as To
talInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerI
d\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': 'Can you list all ta
bles in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SE
LECT name FROM sqlite_master WHERE type = 'table'"}, {'role': 'user', 'conte
nt': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELE
CT COUNT(*) FROM customers'}, {'role': 'user', 'content': " \n   List all
employees and their reporting manager's name (if any):\n"}, {'role': 'assist
ant', 'content': "SELECT e.FirstName || ' ' || e.LastName as EmployeeName,
m.FirstName || ' ' || m.LastName as ManagerName\nFROM employees e\nLEFT JOIN
employees m ON e.ReportsTo = m.EmployeeId"}, {'role': 'user', 'content': '
\n   Find the top 5 most expensive tracks (based on unit price):\n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n   TrackId INTEGER PRIMA
RY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   A
lbumId 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 NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY
(AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPD
ATE 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 (MediaTy
peId) 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\" (Albu
mId)\n\nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDE
X IFK_PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\n\nCREATE INDEX I
FK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE INDEX IFK_Tra
ckMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE TABLE \"invoice_items
\"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPr
ice 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
TABLE \"playlist_track\"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   Tr
ackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (Pl
aylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES \"playlists\"
(PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN
KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON
UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums\" (Artis
tId)\n\nCREATE TABLE \"albums\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTO
INCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId IN
TEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES \"artists\" (Artis
tId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional

```
Context \n\nIn the chinook database invoice means order\n\n===Response Guide
lines \n1. If the provided context is sufficient, please generate a valid SQ
L query without any explanations for the question. \n2. If the provided cont
ext is almost sufficient but requires knowledge of a specific string in a pa
rticular column, please generate an intermediate SQL query to find the disti
nct strings in that column. Prepend the query with a comment saying intermed
iate_sql \n3. If the provided context is insufficient, please explain why it
can't be generated. \n4. Please use the most relevant table(s). \n5. If the
question has been asked and answered before, please repeat the answer exactl
y as it was given before. \n"}, {"role": "user", "content": "  \n    Find al
l tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "a
ssistant", "content": "SELECT * FROM tracks WHERE Name LIKE '%What%'"}, {"ro
le": "user", "content": "  \n    List all invoices with a total exceeding $1
0:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total
> 10"}, {"role": "user", "content": "  \n    List all albums and their corre
sponding artist names  \n"}, {"role": "assistant", "content": "SELECT al.Tit
le as AlbumTitle, ar.Name as ArtistName\nFROM albums al\nJOIN artists ar ON
al.ArtistId = ar.ArtistId"}, {"role": "user", "content": "  \n    Get the av
erage invoice total for each customer:\n"}, {"role": "assistant", "content":
"SELECT c.FirstName, c.LastName, AVG(i.Total) as AverageInvoiceTotal\nFROM c
ustomers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Custo
merId"}, {"role": "user", "content": "what are the top 5 countries that cust
omers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT
(*) as CustomerCount FROM customers\nGROUP BY Country\nORDER BY CustomerCoun
t DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    Find the total numbe
r of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.C
ountry, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices
i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"role": "user", "co
ntent": "  \n    Get the total number of invoices for each customer\n"}, {"r
ole": "assistant", "content": "SELECT c.FirstName, c.LastName, COUNT(i.Invoi
ceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId =
i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": "Can you
list all tables in the SQLite database catalog?"}, {"role": "assistant", "co
ntent": "SELECT name FROM sqlite_master WHERE type = 'table'"}, {"role": "us
er", "content": "How many customers are there"}, {"role": "assistant", "cont
ent": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "  \n
List all employees and their reporting manager's name (if any):\n"}, {"rol
e": "assistant", "content": "SELECT e.FirstName || ' ' || e.LastName as Empl
oyeeName, m.FirstName || ' ' || m.LastName as ManagerName\nFROM employees e
\nLEFT JOIN employees m ON e.ReportsTo = m.EmployeeId"}, {"role": "user", "c
ontent": "  \n    Find the top 5 most expensive tracks (based on unit pric
e):\n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:47:22.351894515
Z', 'message': {'role': 'assistant', 'content': 'SELECT * FROM tracks\nORDER
BY UnitPrice DESC\nLIMIT 5'}, 'done_reason': 'stop', 'done': True, 'total_du
ration': 61031746157, 'load_duration': 17026603, 'prompt_eval_count': 1554,
'prompt_eval_duration': 55334428000, 'eval_count': 21, 'eval_duration': 4353
643000}
LLM Response: SELECT * FROM tracks
ORDER BY UnitPrice DESC
LIMIT 5
SELECT * FROM tracks
ORDER BY UnitPrice DESC
LIMIT 5
   TrackId                                     Name  AlbumId  MediaTypeId  \
```

```
0     2819  Battlestar Galactica: The Story So Far      226           3
1     2820                  Occupation / Precipice      227           3
2     2821                          Exodus, Pt. 1       227           3
3     2822                          Exodus, Pt. 2       227           3
4     2823                          Collaborators       227           3

   GenreId Composer  Milliseconds       Bytes  UnitPrice
0       18     None       2622250   490750393       1.99
1       19     None       5286953  1054423946       1.99
2       19     None       2621708   475079441       1.99
3       19     None       2618000   466820021       1.99
4       19     None       2626626   483484911       1.99
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: '
\n    Find the top 5 most expensive tracks (based on unit price):\n'\n\nThe
DataFrame was produced using this query: SELECT * FROM tracks\nORDER BY Unit
Price DESC\nLIMIT 5\n\nThe following is information about the resulting pand
as DataFrame 'df': \nRunning df.dtypes gives:\n TrackId           int64\nNam
e             object\nAlbumId           int64\nMediaTypeId       int64\nGenr
eId           int64\nComposer         object\nMilliseconds      int64\nBytes
int64\nUnitPrice       float64\ndtype: object"}, {"role": "user", "content":
"Can you generate the Python plotly code to chart the results of the datafra
me? Assume the data is in a pandas dataframe called 'df'. If there is only o
ne value in the dataframe, use an Indicator. Respond with only Python code.
Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:47:41.07888322Z',
'message': {'role': 'assistant', 'content': "```python\nimport plotly.expres
s as px\n\nfig = px.bar(df, x='Name', y='UnitPrice', title='Top 5 Most Expen
sive Tracks')\n\nfig.update_traces(marker=dict(color='red'))\n\nfig.show()\n
```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 18701117559,
'load_duration': 21210965, 'prompt_eval_count': 227, 'prompt_eval_duration':
7579824000, 'eval_count': 58, 'eval_duration': 10966151000}
```
```


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

Out[28]:
```
('SELECT * FROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5',
    TrackId                                  Name  AlbumId  MediaTypeId
\
 0     2819  Battlestar Galactica: The Story So Far      226            3
 1     2820                 Occupation / Precipice      227            3
 2     2821                          Exodus, Pt. 1      227            3
 3     2822                          Exodus, Pt. 2      227            3
 4     2823                          Collaborators      227            3

    GenreId Composer  Milliseconds       Bytes  UnitPrice
 0       18     None       2622250   490750393       1.99
 1       19     None       5286953  1054423946       1.99
 2       19     None       2621708   475079441       1.99
 3       19     None       2618000   466820021       1.99
 4       19     None       2626626   483484911       1.99  ,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'Name=%{x}<br>UnitPrice=%{y}<extra></extra
>',
               'legendgroup': '',
               'marker': {'color': 'red', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Battlestar Galactica: The Story So Far', 'Occup
ation / Precipice',
                           'Exodus, Pt. 1', 'Exodus, Pt. 2', 'Collaborator
s'], dtype=object),
               'xaxis': 'x',
               'y': array([1.99, 1.99, 1.99, 1.99, 1.99]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Top 5 Most Expensive Tracks'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Name'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'UnitPrice'}}}
 }))
```

In [29]:
```
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
```

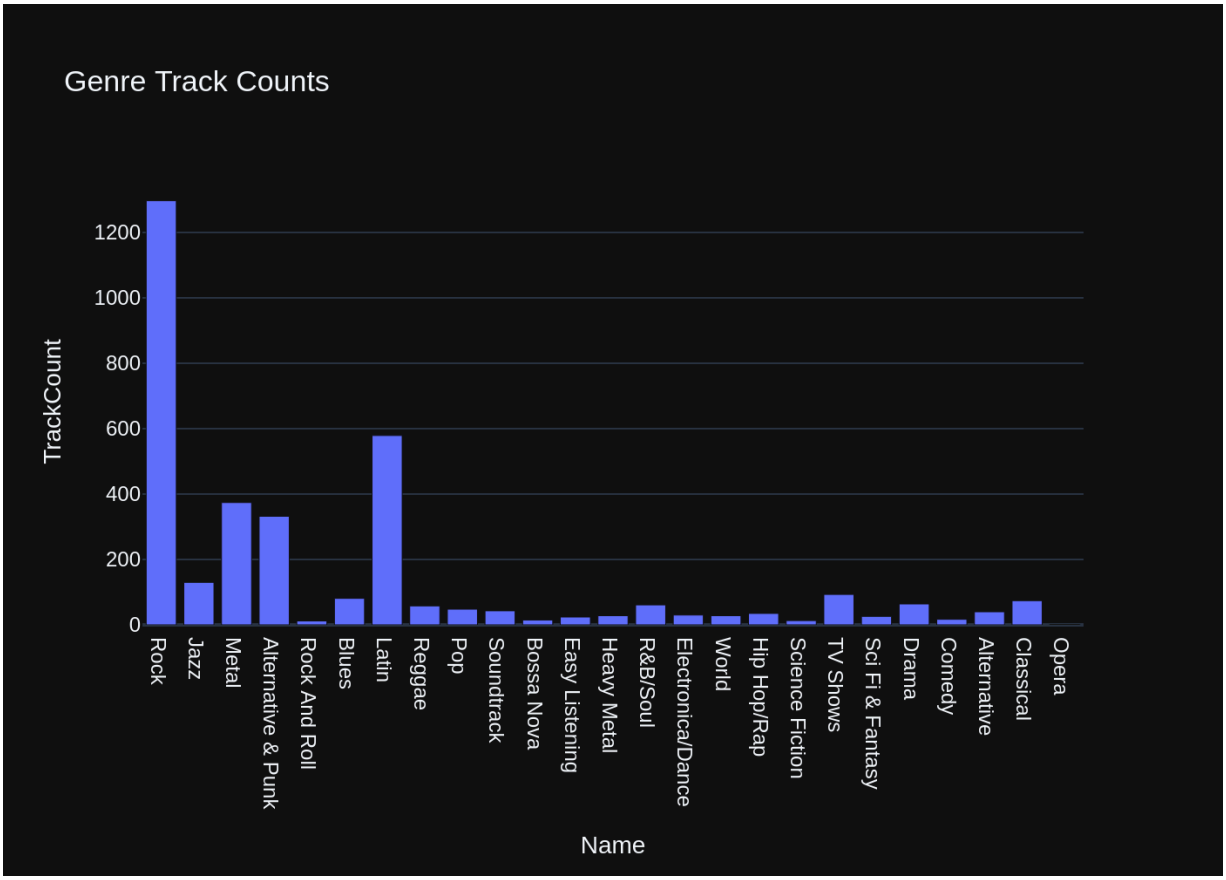
```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please
help to generate a SQL query to answer the question. Your response should ON
LY be based on the given context and follow the response guidelines and form
at instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n    TrackId INT
EGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NUL
L,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    Genr
eId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT
NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n
FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACT
ION 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 AC
TION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks"
(GenreId)\n\nCREATE TABLE "genres"\r\n(\r\n    GenreId INTEGER PRIMARY KEY A
UTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_Pl
aylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE INDEX IFK_TrackAl
bumId ON "tracks" (AlbumId)\n\nCREATE TABLE "playlists"\r\n(\r\n    Playlist
Id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r
\n)\n\nCREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE
TABLE "playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    Trac
kId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (Play
listId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Pl
aylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KE
Y (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPD
ATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIM
ARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n
ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES "artist
s" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE
INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\n\n===Additional Context \n
\nIn the chinook database invoice means order\n\n===Response Guidelines \n1.
If the provided context is sufficient, please generate a valid SQL query wit
hout any explanations for the question. \n2. If the provided context is almo
st sufficient but requires knowledge of a specific string in a particular co
lumn, please generate an intermediate SQL query to find the distinct strings
in that column. Prepend the query with a comment saying intermediate_sql \n
3. If the provided context is insufficient, please explain why it can\'t be
generated. \n4. Please use the 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': '  \n    Find the top 5 m
ost expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'cont
ent': 'SELECT * FROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'u
ser', 'content': '  \n    List all albums and their corresponding artist nam
es  \n'}, {'role': 'assistant', 'content': 'SELECT al.Title as AlbumTitle, a
r.Name as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.Ar
tistId'}, {'role': 'user', 'content': '  \n    Find all tracks with a name c
ontaining "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "S
ELECT * FROM tracks WHERE Name LIKE '%What%'"}, {'role': 'user', 'content':
'what are the top 5 countries that customers come from?'}, {'role': 'assista
nt', 'content': 'SELECT Country, COUNT(*) as CustomerCount FROM customers\nG
ROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'c
ontent': 'Can you list all tables in the SQLite database catalog?'}, {'rol
e': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type = 'ta
ble'"}, {'role': 'user', 'content': '  \n    Find the total number of invoic
es per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COU
NT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.Cus
tomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': '
```

\n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistan
t', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'user',
'content': ' \n   Get the total number of invoices for each customer\n'},
{'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, COUNT(i.In
voiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId
= i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': 'How ma
ny customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*)
FROM customers'}, {'role': 'user', 'content': ' \n   Get the average invoi
ce total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.
FirstName, c.LastName, AVG(i.Total) as AverageInvoiceTotal\nFROM customers c
\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'},
{'role': 'user', 'content': ' \n   List all genres and the number of track
s in each genre:\n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n   TrackId INTEGER PRIMA
RY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   A
lbumId 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 NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY
(AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPD
ATE 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 (MediaTy
peId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION O
N UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (Genr
eId)\n\nCREATE TABLE \"genres\"\r\n(\r\n   GenreId INTEGER PRIMARY KEY AUTO
INCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_Playl
istTrackTrackId ON \"playlist_track\" (TrackId)\n\nCREATE INDEX IFK_TrackAlb
umId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n   Playl
istId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)
\r\n)\n\nCREATE INDEX IFK_TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCR
EATE TABLE \"playlist_track\"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n
TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY
(PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES \"playlist
s\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOR
EIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTI
ON ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n   AlbumId I
NTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT N
ULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERE
NCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION
\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums\" (ArtistId)\n\n\n===Add
itional Context \n\nIn the chinook database invoice means order\n\n===Respon
se Guidelines \n1. If the provided context is sufficient, please generate a
valid SQL query without any explanations for the question. \n2. If the provi
ded context is almost sufficient but requires knowledge of a specific string
in a particular column, please generate an intermediate SQL query to find th
e distinct strings in that column. Prepend the query with a comment saying i
ntermediate_sql \n3. If the provided context is insufficient, please explain
why it can't be generated. \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   F
ind the top 5 most expensive tracks (based on unit price):\n"}, {"role": "as
sistant", "content": "SELECT * FROM tracks\nORDER BY UnitPrice DESC\nLIMIT
5"}, {"role": "user", "content": " \n   List all albums and their correspo
nding artist names \n"}, {"role": "assistant", "content": "SELECT al.Title
as AlbumTitle, ar.Name as ArtistName\nFROM albums al\nJOIN artists ar ON al.
ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   Find all trac
ks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assista
nt", "content": "SELECT * FROM tracks WHERE Name LIKE '%What%'"}, {"role":
"user", "content": "what are the top 5 countries that customers come fro
m?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) as Customer
Count FROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT
5"}, {"role": "user", "content": "Can you list all tables in the SQLite data
base catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_m
aster WHERE type = 'table'"}, {"role": "user", "content": " \n   Find the
total number of invoices per country:\n"}, {"role": "assistant", "content":
"SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJO
IN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"role":
"user", "content": " \n   List all invoices with a total exceeding $1
0:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total
> 10"}, {"role": "user", "content": " \n   Get the total number of invoice
s for each customer\n"}, {"role": "assistant", "content": "SELECT c.FirstNam
e, c.LastName, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN i
nvoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role":
"user", "content": "How many customers are there"}, {"role": "assistant", "c
ontent": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "
\n   Get the average invoice total for each customer:\n"}, {"role": "assist
ant", "content": "SELECT c.FirstName, c.LastName, AVG(i.Total) as AverageInv
oiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId
\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   List all gen
res and the number of tracks in each genre:\n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:48:43.822263587
Z', 'message': {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.Trac
kId) as TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.GenreId\nG
ROUP BY g.GenreId'}, 'done_reason': 'stop', 'done': True, 'total_duration':
62614266170, 'load_duration': 21105463, 'prompt_eval_count': 1442, 'prompt_e
val_duration': 51330950000, 'eval_count': 47, 'eval_duration': 9935350000}
LLM Response: SELECT g.Name, COUNT(t.TrackId) as TrackCount
FROM genres g
JOIN tracks t ON g.GenreId = t.GenreId
GROUP BY g.GenreId
SELECT g.Name, COUNT(t.TrackId) as TrackCount
FROM genres g
JOIN tracks t ON g.GenreId = t.GenreId
GROUP BY g.GenreId

	Name	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
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: '
\n    List all genres and the number of tracks in each genre:\n'\n\nThe Data
Frame was produced using this query: SELECT g.Name, COUNT(t.TrackId) as Trac
kCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.GenreId\nGROUP BY g.Ge
nreId\n\nThe following is information about the resulting pandas DataFrame
'df': \nRunning df.dtypes gives:\n Name          object\nTrackCount     int6
4\ndtype: object"}, {"role": "user", "content": "Can you generate the Python
plotly code to chart the results of the dataframe? Assume the data is in a p
andas dataframe called 'df'. If there is only one value in the dataframe, us
e an Indicator. Respond with only Python code. Do not answer with any explan
ations -- just the code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:48:58.391981481
Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.ex
press as px\n\nfig = px.bar(df, x='Name', y='TrackCount', title='Genre Track
Counts')\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_dur
ation': 14543032277, 'load_duration': 21575137, 'prompt_eval_count': 208, 'p
rompt_eval_duration': 6743060000, 'eval_count': 41, 'eval_duration': 7640929
000}
```


Genre Track Counts
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
Name

Out[29]:
```
('SELECT g.Name, COUNT(t.TrackId) as TrackCount\nFROM genres g\nJOIN tracks
t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId',
                    Name  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': 'Name=%{x}<br>TrackCount=%{y}<extra></extra
>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Rock', 'Jazz', 'Metal', 'Alternative & Punk',
'Rock And Roll', 'Blues',
                           'Latin', 'Reggae', 'Pop', 'Soundtrack', 'Bossa N
ova', 'Easy Listening',
                           'Heavy Metal', 'R&B/Soul', 'Electronica/Dance',
'World', 'Hip Hop/Rap',
                           'Science Fiction', 'TV Shows', 'Sci Fi & Fantas
y', 'Drama', 'Comedy',
                           'Alternative', 'Classical', 'Opera'], dtype=obje
ct),
               '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': 'Genre Track Counts'},
                 'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Name'}},
                 'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'TrackCount'}}}
 }))
```

In [30]:
```
question = """
    Get all genres that do not have any tracks associated with them:
"""

vn.ask(question=question)
```

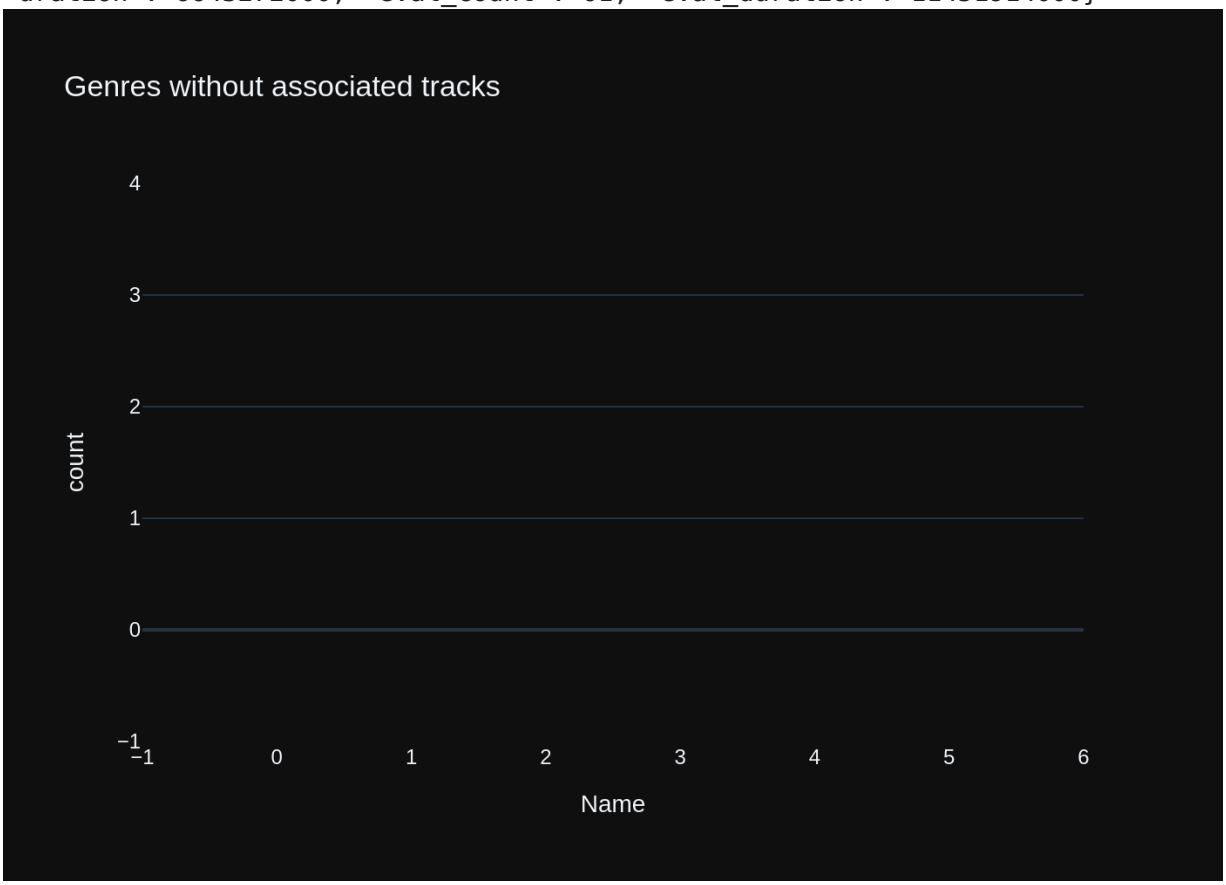
```
Number of requested results 10 is greater than number of elements in index
1, updating n_results = 1
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please
help to generate a SQL query to answer the question. Your response should ON
LY be based on the given context and follow the response guidelines and form
at instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON "tracks" (Ge
nreId)\n\nCREATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTO
INCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTE
GER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Co
mposer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes IN
TEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumI
d) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO AC
TION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFER
ENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO A
CTION\r\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (Tra
ckId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREA
TE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "genres"\r\n
(\r\n    GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVA
RCHAR(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    Art
istId 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 TA
BLE "playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackI
d INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (Playli
stId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Play
listId) \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 UPDAT
E NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n
\nCREATE TABLE "playlists"\r\n(\r\n    PlaylistId INTEGER PRIMARY KEY AUTOIN
CREMENT 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 \n
1. If the provided context is sufficient, 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 strin
gs in that column. Prepend the query with a comment saying intermediate_sql
\n3. If the provided context is insufficient, please explain why it can\'t b
e generated. \n4. Please use the most relevant table(s). \n5. If the questio
n 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', 'conten
t': 'SELECT g.Name, COUNT(t.TrackId) as TrackCount\nFROM genres g\nJOIN trac
ks t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId'}, {'role': 'user', 'conte
nt': '  \n    Find all tracks with a name containing "What" (case-insensitiv
e)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LI
KE '%What%'"}, {'role': 'user', 'content': '  \n    Find the top 5 most expe
nsive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'S
ELECT * FROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'c
ontent': '  \n    List all albums and their corresponding artist names
\n'}, {'role': 'assistant', 'content': 'SELECT al.Title as AlbumTitle, ar.Na
me as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.Artist
Id'}, {'role': 'user', 'content': 'Can you list all tables in the SQLite dat
abase catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_
master WHERE 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': 'wha
t are the top 5 countries that customers come from?'}, {'role': 'assistant',
```

'content': 'SELECT Country, COUNT(*) as CustomerCount FROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': " \n   List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': "SELECT e.FirstName || ' ' || e.LastName as EmployeeName, m.FirstName || ' ' || m.LastName as ManagerName\nFROM employees e\nLEFT JOIN employees m ON e.ReportsTo = m.EmployeeId"}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': ' \n   Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, AVG(i.Total) as AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n   Get all genres that do not have any tracks associated with them:\n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE TABLE \"tracks\"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE INDEX IFK_TrackAlbumId 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   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer ex

actly as it was given before. \n"}, {"role": "user", "content": " \n   Lis
t all genres and the number of tracks in each genre:\n"}, {"role": "assistan
t", "content": "SELECT g.Name, COUNT(t.TrackId) as TrackCount\nFROM genres g
\nJOIN tracks t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId"}, {"role": "us
er", "content": " \n   Find all tracks with a name containing \"What\" (ca
se-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks
WHERE Name LIKE '%What%'"}, {"role": "user", "content": " \n   Find the to
p 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant",
"content": "SELECT * FROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"rol
e": "user", "content": " \n   List all albums and their corresponding arti
st names \n"}, {"role": "assistant", "content": "SELECT al.Title as AlbumTi
tle, ar.Name as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId =
ar.ArtistId"}, {"role": "user", "content": "Can you list all tables in the S
QLite database catalog?"}, {"role": "assistant", "content": "SELECT name FRO
M sqlite_master WHERE type = 'table'"}, {"role": "user", "content": " \n
List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "co
ntent": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "conten
t": "what are the top 5 countries that customers come from?"}, {"role": "ass
istant", "content": "SELECT Country, COUNT(*) as CustomerCount FROM customer
s\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "use
r", "content": " \n   List all employees and their reporting manager's nam
e (if any):\n"}, {"role": "assistant", "content": "SELECT e.FirstName || ' '
|| e.LastName as EmployeeName, m.FirstName || ' ' || m.LastName as ManagerNa
me\nFROM employees e\nLEFT JOIN employees m ON e.ReportsTo = m.EmployeeId"},
{"role": "user", "content": "How many customers are there"}, {"role": "assis
tant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user", "conte
nt": " \n   Get the average invoice total for each customer:\n"}, {"role":
"assistant", "content": "SELECT c.FirstName, c.LastName, AVG(i.Total) as Ave
rageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.Cust
omerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   Get al
l genres that do not have any tracks associated with them:\n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:49:59.869605702
Z', 'message': {'role': 'assistant', 'content': 'SELECT g.Name\nFROM genres
g\nLEFT JOIN tracks t ON g.GenreId = t.GenreId\nWHERE t.TrackId IS NULL'},
'done_reason': 'stop', 'done': True, 'total_duration': 61366615849, 'load_du
ration': 20430667, 'prompt_eval_count': 1459, 'prompt_eval_duration': 519997
46000, 'eval_count': 38, 'eval_duration': 8021735000}
LLM Response: SELECT g.Name
FROM genres g
LEFT JOIN tracks t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
SELECT g.Name
FROM genres g
LEFT JOIN tracks t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
Empty DataFrame
Columns: [Name]
Index: []
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: '

\n   Get all genres that do not have any tracks associated with them:\n'\n
\nThe DataFrame was produced using this query: SELECT g.Name\nFROM genres g
\nLEFT JOIN tracks t ON g.GenreId = t.GenreId\nWHERE t.TrackId IS NULL\n\nTh
e following is information about the resulting pandas DataFrame 'df': \nRunn
ing df.dtypes gives:\n Name   object\ndtype: object"}, {"role": "user", "co
ntent": "Can you generate the Python plotly code to chart the results of the
dataframe? Assume the data is in a pandas dataframe called 'df'. If there is
only one value in the dataframe, use an Indicator. Respond with only Python
code. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:50:18.127269548
Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.ex
press as px\n\nfig = px.bar(df, x='Name', title='Genres without associated t
racks')\n\nif len(df) == 1:\n   fig.update_traces(marker_color='red')\n\nfi
g.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 1825
5367411, 'load_duration': 26908842, 'prompt_eval_count': 193, 'prompt_eval_d
uration': 6643272000, 'eval_count': 61, 'eval_duration': 11451914000}



Out[30]:
```
('SELECT g.Name\nFROM genres g\nLEFT JOIN tracks t ON g.GenreId = t.GenreId
\nWHERE t.TrackId IS NULL',
 Empty DataFrame
 Columns: [Name]
 Index: [],
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'Name=%{x}<br>count=%{y}<extra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array([], dtype=object),
               'xaxis': 'x',
               'y': array([], dtype=int64),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Genres without associated tracks'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Name'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'count'}}}
 }))
```

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

vn.ask(question=question)
```

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

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please
help to generate a SQL query to answer the question. Your response should ON
LY be based on the given context and follow the response guidelines and form
at instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId
INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT N
ULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(7
0),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n
BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    T
otal NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES "cu
stomers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n
\nCREATE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName N
VARCHAR(20)  NOT NULL,\r\n    Company NVARCHAR(80),\r\n    Address NVARCHAR
(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Country NV
ARCHAR(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 I
NTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "inv
oice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT
NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NUL
L,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NU
LL,\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 "employees"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY A
UTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstN
ame NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTE
GER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NVA
RCHAR(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(2
4),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60),\r\n    FOREIGN KEY
(ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n    Playli
stId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT
PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (Pla
ylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON
UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "album
s"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Ti
tle NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREI
GN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTIO
N ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "cust
omers" (SupportRepId)\n\nCREATE TABLE "playlists"\r\n(\r\n    PlaylistId INT
EGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nC
REATE TABLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT N
OT 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 NVARC
HAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n
UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES
"albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    F
OREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTI
ON ON UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_t
ypes" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n
CREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\n\n===Additi
onal Context \n\nIn the chinook database invoice means order\n\n===Response
Guidelines \n1. If the provided context is sufficient, please generate a val
```

```
id SQL query without any explanations for the question. \n2. If the provided
context is almost sufficient but requires knowledge of a specific string in
a particular column, please generate an intermediate SQL query to find the d
istinct strings in that column. Prepend the query with a comment saying inte
rmediate_sql \n3. If the provided context is insufficient, please explain wh
y it can\'t be generated. \n4. Please use the most relevant table(s). \n5. I
f the question has been asked and answered before, please repeat the answer
exactly as it was given before. \n'}, {'role': 'user', 'content': 'How many
customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FRO
M customers'}, {'role': 'user', 'content': '  \n    Get the total number of
invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.F
irstName, c.LastName, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c
\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'},
{'role': 'user', 'content': 'what are the top 5 countries that customers com
e from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) as Cus
tomerCount FROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLI
MIT 5'}, {'role': 'user', 'content': '  \n    Find the total number of invoi
ces per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, CO
UNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.Cu
stomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': '
\n    Get the average invoice total for each customer:\n'}, {'role': 'assist
ant', 'content': 'SELECT c.FirstName, c.LastName, AVG(i.Total) as AverageInv
oiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId
\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': '  \n    List all inv
oices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SEL
ECT * FROM invoices WHERE Total > 10'}, {'role': 'user', 'content': "  \n
List all employees and their reporting manager's name (if any):\n"}, {'rol
e': 'assistant', 'content': "SELECT e.FirstName || ' ' || e.LastName as Empl
oyeeName, m.FirstName || ' ' || m.LastName as ManagerName\nFROM employees e
\nLEFT JOIN employees m ON e.ReportsTo = m.EmployeeId"}, {'role': 'user', 'c
ontent': '  \n    List all albums and their corresponding artist names
\n'}, {'role': 'assistant', 'content': 'SELECT al.Title as AlbumTitle, ar.Na
me as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.Artist
Id'}, {'role': 'user', 'content': '  \n    List all genres and the number of
tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name,
COUNT(t.TrackId) as TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId =
t.GenreId\nGROUP BY g.GenreId'}, {'role': 'user', 'content': '  \n    Find t
he top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assista
nt', 'content': 'SELECT * FROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'},
{'role': 'user', 'content': '  \n    List all customers who have not placed
any orders:\n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n    InvoiceId INTEGER P
RIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n
InvoiceDate DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    B
illingCity NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCou
ntry NVARCHAR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMER
IC(10,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\"
(CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE
```

TABLE \"customers\"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"employees\"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "

```
\n    Get the total number of invoices for each customer\n"}, {"role": "assi
stant", "content": "SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as To
talInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerI
d\nGROUP BY c.CustomerId"}, {"role": "user", "content": "what are the top 5
countries that customers come from?"}, {"role": "assistant", "content": "SEL
ECT Country, COUNT(*) as CustomerCount FROM customers\nGROUP BY Country\nORD
ER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    Fi
nd the total number of invoices per country:\n"}, {"role": "assistant", "con
tent": "SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices\nFROM customer
s c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"},
{"role": "user", "content": "  \n    Get the average invoice total for each
customer:\n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastN
ame, AVG(i.Total) as AverageInvoiceTotal\nFROM customers c\nJOIN invoices i
ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "c
ontent": "  \n    List all invoices with a total exceeding $10:\n"}, {"rol
e": "assistant", "content": "SELECT * FROM invoices WHERE Total > 10"}, {"ro
le": "user", "content": "  \n    List all employees and their reporting mana
ger's name (if any):\n"}, {"role": "assistant", "content": "SELECT e.FirstNa
me || ' ' || e.LastName as EmployeeName, m.FirstName || ' ' || m.LastName as
ManagerName\nFROM employees e\nLEFT JOIN employees m ON e.ReportsTo = m.Empl
oyeeId"}, {"role": "user", "content": "  \n    List all albums and their cor
responding artist names  \n"}, {"role": "assistant", "content": "SELECT al.T
itle as AlbumTitle, ar.Name as ArtistName\nFROM albums al\nJOIN artists ar O
N al.ArtistId = ar.ArtistId"}, {"role": "user", "content": "  \n    List all
genres and the number of tracks in each genre:\n"}, {"role": "assistant", "c
ontent": "SELECT g.Name, COUNT(t.TrackId) as TrackCount\nFROM genres g\nJOIN
tracks t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId"}, {"role": "user", "c
ontent": "  \n    Find the top 5 most expensive tracks (based on unit pric
e):\n"}, {"role": "assistant", "content": "SELECT * FROM tracks\nORDER BY Un
itPrice DESC\nLIMIT 5"}, {"role": "user", "content": "  \n    List all custo
mers who have not placed any orders:\n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:51:37.014265066
Z', 'message': {'role': 'assistant', 'content': 'SELECT * FROM customers\nWH
ERE CustomerId NOT IN (SELECT CustomerId FROM invoices'}, 'done_reason': 'st
op', 'done': True, 'total_duration': 78788152085, 'load_duration': 16728554,
'prompt_eval_count': 2018, 'prompt_eval_duration': 72440857000, 'eval_coun
t': 23, 'eval_duration': 4985810000}
LLM Response: SELECT * FROM customers
WHERE CustomerId NOT IN (SELECT CustomerId FROM invoices
SELECT * FROM customers
WHERE CustomerId NOT IN (SELECT CustomerId FROM invoices
Couldn't run sql:  Execution failed on sql 'SELECT * FROM customers
WHERE CustomerId NOT IN (SELECT CustomerId FROM invoices': incomplete input
```

In [32]:

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

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 the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "artists"\r\n(\r\n   ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "playlists"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "genres"\r\n(\r\n   GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.TrackId) as TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT al.Title as AlbumTitle, ar.Name as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) as CustomerCount FROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "Wha

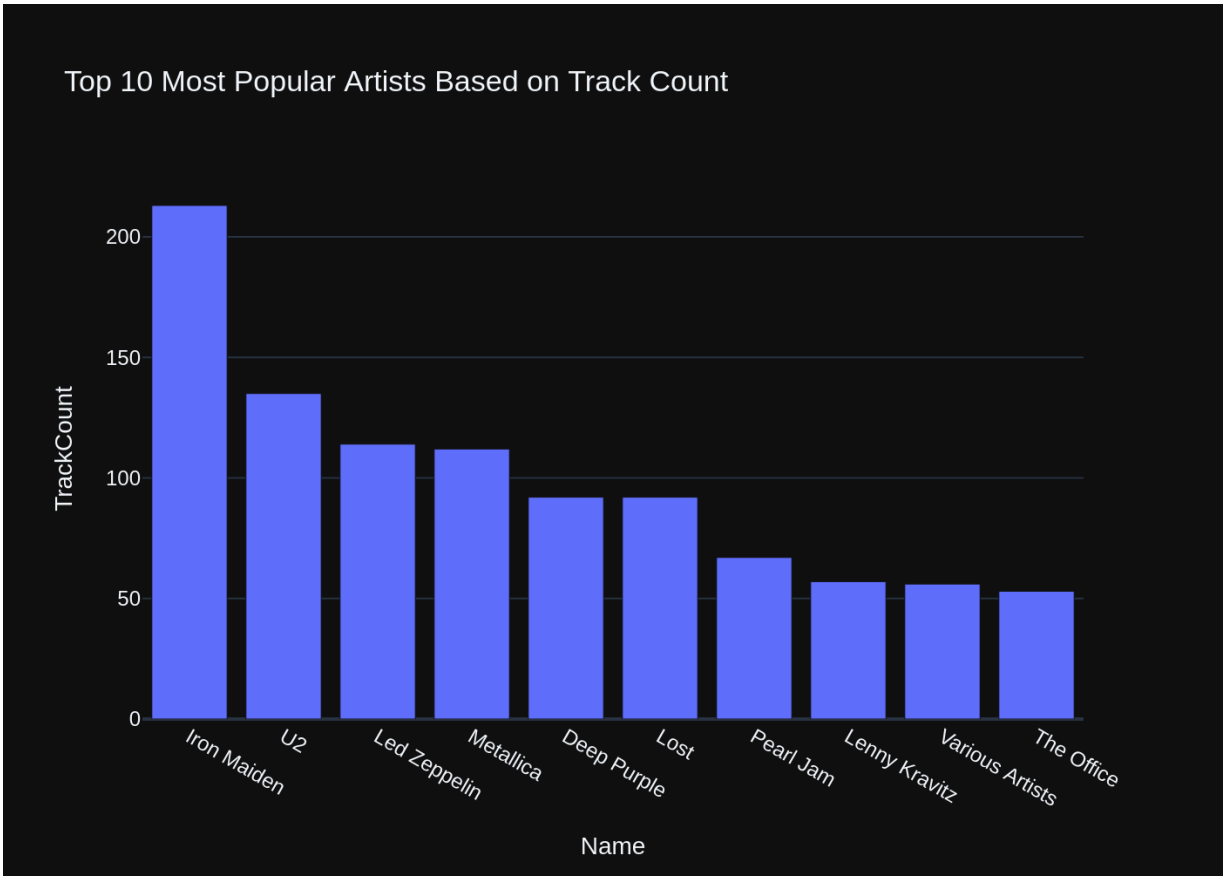
t" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM t
racks WHERE Name LIKE '%What%'"}, {'role': 'user', 'content': 'Can you list
all tables in the SQLite database catalog?'}, {'role': 'assistant', 'conten
t': "SELECT name FROM sqlite_master WHERE type = 'table'"}, {'role': 'user',
'content': ' \n   Get the total number of invoices for each customer\n'},
{'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, COUNT(i.In
voiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId
= i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n
Get the average invoice total for each customer:\n'}, {'role': 'assistant',
'content': 'SELECT c.FirstName, c.LastName, AVG(i.Total) as AverageInvoiceTo
tal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP
BY c.CustomerId'}, {'role': 'user', 'content': ' \n   List all invoices wi
th a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FR
OM invoices WHERE Total > 10'}, {'role': 'user', 'content': ' \n   There ar
e 3 tables: artists, albums and tracks, where albums and artists are linked
by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the
top 10 most popular artists based on the number of tracks\n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n   TrackId INTEGER PRIMA
RY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   A
lbumId 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 NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY
(AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPD
ATE 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 (MediaTy
peId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION O
N UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n   AlbumId INTEG
ER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NUL
L,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENC
ES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r
\n)\n\nCREATE TABLE \"artists\"\r\n(\r\n   ArtistId INTEGER PRIMARY KEY AUT
OINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_Albu
mArtistId ON \"albums\" (ArtistId)\n\nCREATE INDEX IFK_TrackAlbumId ON \"tra
cks\" (AlbumId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n   PlaylistId INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE
TABLE \"genres\"\r\n(\r\n   GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT N
ULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n
(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n
CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREI
GN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE
NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tra
cks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREA
TE INDEX IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX IFK_Playli
stTrackTrackId ON \"playlist_track\" (TrackId)\n\n\n===Additional Context \n
\nIn the chinook database invoice means order\n\n===Response Guidelines \n1.
If the provided context is sufficient, please generate a valid SQL query wit
hout any explanations for the question. \n2. If the provided context is almo
st sufficient but requires knowledge of a specific string in a particular co

lumn, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n 3. If the provided context is insufficient, please explain why it can't be g enerated. \n4. Please use the most relevant table(s). \n5. If the question h as been asked and answered before, please repeat the answer exactly as it wa s given before. \n"}, {"role": "user", "content": " \n   List all genres a nd the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.TrackId) as TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names \n"}, {"rol e": "assistant", "content": "SELECT al.Title as AlbumTitle, ar.Name as Artis tName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.ArtistId"}, {"rol e": "user", "content": " \n   Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT * FROM tracks \nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "cont ent": "SELECT Country, COUNT(*) as CustomerCount FROM customers\nGROUP BY Co untry\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"role": "assistan t", "content": "SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Coun try"}, {"role": "user", "content": " \n   Find all tracks with a name cont aining \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SE LECT * FROM tracks WHERE Name LIKE '%What%'"}, {"role": "user", "content": "Can you list all tables in the SQLite database catalog?"}, {"role": "assist ant", "content": "SELECT name FROM sqlite_master WHERE type = 'table'"}, {"r ole": "user", "content": " \n   Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastNa me, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i O N c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "co ntent": " \n   Get the average invoice total for each customer:\n"}, {"rol e": "assistant", "content": "SELECT c.FirstName, c.LastName, AVG(i.Total) as AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.C ustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   Lis t all invoices with a total exceeding $10:\n"}, {"role": "assistant", "conte nt": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "content": " \n   There are 3 tables: artists, albums and tracks, where albums and art ists are linked by ArtistId, albums and tracks are linked by AlbumId,\n   C an you find the top 10 most popular artists based on the number of tracks \n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:52:47.388128474 Z', 'message': {'role': 'assistant', 'content': 'SELECT ar.Name, COUNT(t.Tra ckId) as TrackCount\nFROM artists ar\nJOIN albums al ON ar.ArtistId = al.Art istId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY ar.ArtistId\nORDER BY TrackCount DESC\nLIMIT 10'}, 'done_reason': 'stop', 'done': True, 'total_ duration': 70332978850, 'load_duration': 20819192, 'prompt_eval_count': 154 1, 'prompt_eval_duration': 53513132000, 'eval_count': 72, 'eval_duration': 1 5458929000}
LLM Response: SELECT ar.Name, COUNT(t.TrackId) as TrackCount
FROM artists ar
JOIN albums al ON ar.ArtistId = al.ArtistId
JOIN tracks t ON al.AlbumId = t.AlbumId
GROUP BY ar.ArtistId
ORDER BY TrackCount DESC
LIMIT 10

```
SELECT ar.Name, COUNT(t.TrackId) as TrackCount
FROM artists ar
JOIN albums al ON ar.ArtistId = al.ArtistId
JOIN tracks t ON al.AlbumId = t.AlbumId
GROUP BY ar.ArtistId
ORDER BY TrackCount DESC
LIMIT 10
```

	Name	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

```
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: '
\n    There are 3 tables: artists, albums and tracks, where albums and artis
ts are linked by ArtistId, albums and tracks are linked by AlbumId,\n    Can
you find the top 10 most popular artists based on the number of tracks\n'\n
\nThe DataFrame was produced using this query: SELECT ar.Name, COUNT(t.Track
Id) as TrackCount\nFROM artists ar\nJOIN albums al ON ar.ArtistId = al.Artis
tId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY ar.ArtistId\nORDER BY
TrackCount DESC\nLIMIT 10\n\nThe following is information about the resultin
g pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name          object\n
TrackCount     int64\ndtype: object"}, {"role": "user", "content": "Can you
generate the Python plotly code to chart the results of the dataframe? Assum
e the data is in a pandas dataframe called 'df'. If there is only one value
in the dataframe, use an Indicator. Respond with only Python code. Do not an
swer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:53:05.720969742
Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.ex
press as px\n\nfig = px.bar(df, x='Name', y='TrackCount', title='Top 10 Most
Popular Artists Based on Track Count')\n\nfig.show()\n```"}, 'done_reason':
'stop', 'done': True, 'total_duration': 18311923015, 'load_duration': 200120
98, 'prompt_eval_count': 271, 'prompt_eval_duration': 8885110000, 'eval_coun
t': 49, 'eval_duration': 9276073000}
```


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

Out[32]:
```
('SELECT ar.Name, COUNT(t.TrackId) as TrackCount\nFROM artists ar\nJOIN alb
ums al ON ar.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId = t.AlbumI
d\nGROUP BY ar.ArtistId\nORDER BY TrackCount DESC\nLIMIT 10',
               Name  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': 'Name=%{x}<br>TrackCount=%{y}<extra></extra
>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Iron Maiden', 'U2', 'Led Zeppelin', 'Metallic
a', '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,  5
3]),
               'yaxis': 'y'}],
      'layout': {'barmode': 'relative',
                 'legend': {'tracegroupgap': 0},
                 'template': '...',
                 'title': {'text': 'Top 10 Most Popular Artists Based on Tra
ck Count'},
                 'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Name'}},
                 'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'TrackCount'}}}
 }))
```

In [33]:
```
question = """
     List all customers from Canada and their email addresses:
"""

vn.ask(question=question)
```

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

SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "customers"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE TABLE "employees"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE "playlist_track"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE TABLE "albums"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Countr

y, COUNT(*) as CustomerCount FROM customers\nGROUP BY Country\nORDER BY Cust
omerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers a
re there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customer
s'}, {'role': 'user', 'content': ' \n   Find the total number of invoices
per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT
(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.Custo
merId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': '
\n   Get the total number of invoices for each customer\n'}, {'role': 'assi
stant', 'content': 'SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as To
talInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerI
d\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': " \n   List all em
ployees and their reporting manager's name (if any):\n"}, {'role': 'assistan
t', 'content': "SELECT e.FirstName || ' ' || e.LastName as EmployeeName, m.F
irstName || ' ' || m.LastName as ManagerName\nFROM employees e\nLEFT JOIN em
ployees m ON e.ReportsTo = m.EmployeeId"}, {'role': 'user', 'content': ' \n
Get the average invoice total for each customer:\n'}, {'role': 'assistant',
'content': 'SELECT c.FirstName, c.LastName, AVG(i.Total) as AverageInvoiceTo
tal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP
BY c.CustomerId'}, {'role': 'user', 'content': ' \n   List all invoices wi
th a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FR
OM invoices WHERE Total > 10'}, {'role': 'user', 'content': 'Can you list al
l tables in the SQLite database catalog?'}, {'role': 'assistant', 'content':
"SELECT name FROM sqlite_master WHERE type = 'table'"}, {'role': 'user', 'co
ntent': ' \n   List all albums and their corresponding artist names \n'},
{'role': 'assistant', 'content': 'SELECT al.Title as AlbumTitle, ar.Name as
ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.ArtistId'},
{'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (b
ased on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tr
acks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n
List all customers from Canada and their email addresses:\n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE INDEX IFK_CustomerSupportRepId ON \"customers\" (S
upportRepId)\n\nCREATE TABLE \"customers\"\r\n(\r\n   CustomerId INTEGER PR
IMARY 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   A
ddress NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r
\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVA
RCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n
SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES \"employe
es\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCR
EATE TABLE \"invoices\"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREM
ENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETI
ME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHA
R(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(4
0),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT N
ULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_Inv
oiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE TABLE \"employees\"\r
\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Las

tName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n
Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DATETIME,\r\n
HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r
\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVA
RCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email
NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (Emplo
yeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE
\"invoice_items\"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT
NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT
NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r
\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId)
REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT
ION\r\n)\n\nCREATE TABLE sqlite_sequence(name,seq)\n\nCREATE TABLE \"playlis
t_track\"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER
NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, Track
Id),\r\n   FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackI
d) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON \"employees\" (ReportsT
o)\n\nCREATE TABLE \"albums\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTE
GER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistI
d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional C
ontext \n\nIn the chinook database invoice means order\n\n===Response Guidel
ines \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 part
icular column, please generate an intermediate SQL query to find the distinc
t strings in that column. Prepend the query with a comment saying intermedia
te_sql \n3. If the provided context is insufficient, please explain why it c
an't be generated. \n4. Please use the most relevant table(s). \n5. If the q
uestion has been asked and answered before, please repeat the answer exactly
as it was given before. \n"}, {"role": "user", "content": "what are the top
5 countries that customers come from?"}, {"role": "assistant", "content": "S
ELECT Country, COUNT(*) as CustomerCount FROM customers\nGROUP BY Country\nO
RDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "user", "content": "How many
customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FRO
M customers"}, {"role": "user", "content": " \n   Find the total number of
invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Countr
y, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON
c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"role": "user", "conten
t": " \n   Get the total number of invoices for each customer\n"}, {"rol
e": "assistant", "content": "SELECT c.FirstName, c.LastName, COUNT(i.Invoice
Id) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.
CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   Li
st all employees and their reporting manager's name (if any):\n"}, {"role":
"assistant", "content": "SELECT e.FirstName || ' ' || e.LastName as Employee
Name, m.FirstName || ' ' || m.LastName as ManagerName\nFROM employees e\nLEF
T JOIN employees m ON e.ReportsTo = m.EmployeeId"}, {"role": "user", "conten
t": " \n   Get the average invoice total for each customer:\n"}, {"role":
"assistant", "content": "SELECT c.FirstName, c.LastName, AVG(i.Total) as Ave
rageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.Cust
omerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   List a
ll invoices with a total exceeding $10:\n"}, {"role": "assistant", "conten
t": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "content":

```
"Can you list all tables in the SQLite database catalog?"}, {"role": "assist
ant", "content": "SELECT name FROM sqlite_master WHERE type = 'table'"}, {"r
ole": "user", "content": "  \n    List all albums and their corresponding ar
tist names  \n"}, {"role": "assistant", "content": "SELECT al.Title as Album
Title, ar.Name as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId
= ar.ArtistId"}, {"role": "user", "content": "  \n    Find the top 5 most ex
pensive tracks (based on unit price):\n"}, {"role": "assistant", "content":
"SELECT * FROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user",
"content": "  \n     List all customers from Canada and their email addresse
s:\n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:54:15.48696028Z',
'message': {'role': 'assistant', 'content': "SELECT FirstName, LastName, Ema
il FROM customers\nWHERE Country = 'Canada"}, 'done_reason': 'stop', 'done':
True, 'total_duration': 69649051559, 'load_duration': 21187312, 'prompt_eval
_count': 1798, 'prompt_eval_duration': 63861606000, 'eval_count': 21, 'eval_
duration': 4409173000}
LLM Response: SELECT FirstName, LastName, Email FROM customers
WHERE Country = 'Canada
SELECT FirstName, LastName, Email FROM customers
WHERE Country = 'Canada
Couldn't run sql:  Execution failed on sql 'SELECT FirstName, LastName, Emai
l FROM customers
WHERE Country = 'Canada': unrecognized token: "'Canada"
```

In [34]:
```
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
```

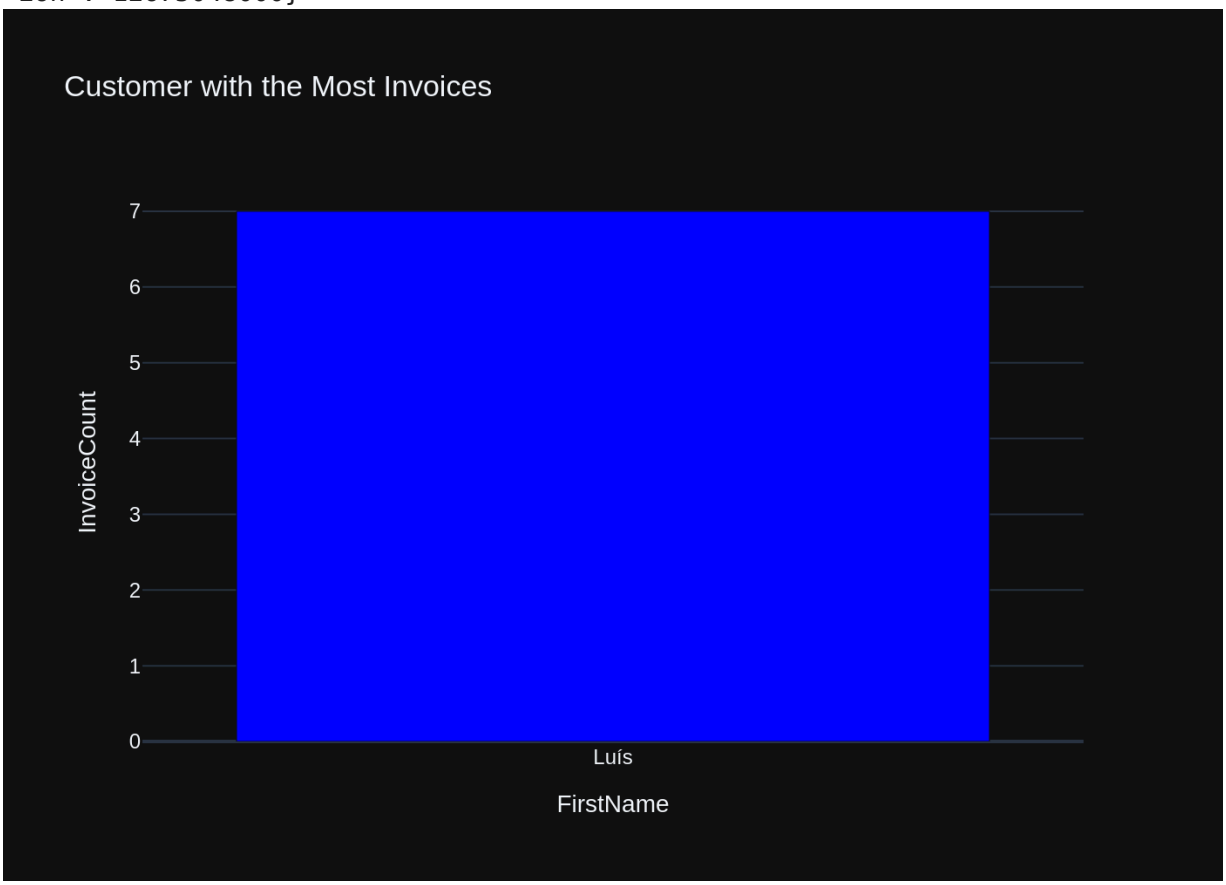
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_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   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "employees"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Get the

total number of invoices for each customer\n'}, {'role': 'assistant', 'conte
nt': 'SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as TotalInvoices\nF
ROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.
CustomerId'}, {'role': 'user', 'content': ' \n   Find the total number of
invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Countr
y, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON
c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'conten
t': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'as
sistant', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'u
ser', 'content': ' \n   Get the average invoice total for each custome
r:\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, AV
G(i.Total) as AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.Cu
stomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'conten
t': 'what are the top 5 countries that customers come from?'}, {'role': 'ass
istant', 'content': 'SELECT Country, COUNT(*) as CustomerCount FROM customer
s\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'use
r', 'content': ' \n   Find the top 5 most expensive tracks (based on unit
price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks\nORDER B
Y UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers
are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customer
s'}, {'role': 'user', 'content': ' \n   There are 3 tables: artists, albums
and tracks, where albums and artists are linked by ArtistId, albums and trac
ks are linked by AlbumId,\n   Can you find the top 10 most popular artists
based on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT
ar.Name, COUNT(t.TrackId) as TrackCount\nFROM artists ar\nJOIN albums al ON
ar.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY
ar.ArtistId\nORDER BY TrackCount DESC\nLIMIT 10'}, {'role': 'user', 'conten
t': " \n   List all employees and their reporting manager's name (if an
y):\n"}, {'role': 'assistant', 'content': "SELECT e.FirstName || ' ' || e.La
stName as EmployeeName, m.FirstName || ' ' || m.LastName as ManagerName\nFRO
M employees e\nLEFT JOIN employees m ON e.ReportsTo = m.EmployeeId"}, {'rol
e': 'user', 'content': ' \n   List all albums and their corresponding arti
st names \n'}, {'role': 'assistant', 'content': 'SELECT al.Title as AlbumTi
tle, ar.Name as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId =
ar.ArtistId'}, {'role': 'user', 'content': ' \n   Find the customer with
the most invoices \n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n   InvoiceId INTEGER P
RIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n
InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   B
illingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCou
ntry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMER
IC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES \"customers\"
(CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE
INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK
_InvoiceLineInvoiceId ON \"invoice_items\" (InvoiceId)\n\nCREATE TABLE \"inv
oice_items\"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT
NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NUL
L,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NU

LL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n \t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) RE FERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO N\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId) \n\nCREATE TABLE \"customers\"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AU TOINCREMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   LastNa me NVARCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARC HAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId I NTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (Employee Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK _CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"empl oyees\"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r \n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NU LL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DA TETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City N VARCHAR(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(2 4),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES \"e mployees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE INDEX IFK_EmployeeReportsTo ON \"employees\" (ReportsTo)\n\nCREAT E 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   Me diaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHA R(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 A CTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES \"med ia_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r \n)\n\n\n===Additional Context \n\nIn the chinook database invoice means ord er\n\n===Response Guidelines \n1. If the provided context is sufficient, ple ase generate a valid SQL query without any explanations for the question. \n 2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficien t, please explain why it can't be generated. \n4. Please use the most releva nt table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "co ntent": " \n   Get the total number of invoices for each customer\n"}, {"r ole": "assistant", "content": "SELECT c.FirstName, c.LastName, COUNT(i.Invoi ceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n Find the total number of invoices per country:\n"}, {"role": "assistant", "c ontent": "SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices\nFROM custom ers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"role": "user", "content": " \n   List all invoices with a total exceedin g $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE T otal > 10"}, {"role": "user", "content": " \n   Get the average invoice to tal for each customer:\n"}, {"role": "assistant", "content": "SELECT c.First Name, c.LastName, AVG(i.Total) as AverageInvoiceTotal\nFROM customers c\nJOI N invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"rol e": "user", "content": "what are the top 5 countries that customers come fro m?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) as Customer

Count FROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT
5"}, {"role": "user", "content": " \n   Find the top 5 most expensive trac
ks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT * FR
OM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content":
"How many customers are there"}, {"role": "assistant", "content": "SELECT CO
UNT(*) FROM customers"}, {"role": "user", "content": " \n   There are 3 tab
les: artists, albums and tracks, where albums and artists are linked by Arti
stId, albums and tracks are linked by AlbumId,\n   Can you find the top 10
most popular artists based on the number of tracks\n"}, {"role": "assistan
t", "content": "SELECT ar.Name, COUNT(t.TrackId) as TrackCount\nFROM artists
ar\nJOIN albums al ON ar.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId
= t.AlbumId\nGROUP BY ar.ArtistId\nORDER BY TrackCount DESC\nLIMIT 10"}, {"r
ole": "user", "content": " \n   List all employees and their reporting man
ager's name (if any):\n"}, {"role": "assistant", "content": "SELECT e.FirstN
ame || ' ' || e.LastName as EmployeeName, m.FirstName || ' ' || m.LastName a
s ManagerName\nFROM employees e\nLEFT JOIN employees m ON e.ReportsTo = m.Em
ployeeId"}, {"role": "user", "content": " \n   List all albums and their c
orresponding artist names \n"}, {"role": "assistant", "content": "SELECT a
l.Title as AlbumTitle, ar.Name as ArtistName\nFROM albums al\nJOIN artists a
r ON al.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   Find
the customer with the most invoices \n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:55:37.649643298
Z', 'message': {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastN
ame, COUNT(i.InvoiceId) as InvoiceCount\nFROM customers c\nJOIN invoices i O
N c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY InvoiceCount
DESC\nLIMIT 1'}, 'done_reason': 'stop', 'done': True, 'total_duration': 8212
2322207, 'load_duration': 20884118, 'prompt_eval_count': 1927, 'prompt_eval_
duration': 68070593000, 'eval_count': 58, 'eval_duration': 12695055000}
LLM Response: SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as InvoiceC
ount
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY InvoiceCount DESC
LIMIT 1
SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as InvoiceCount
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY InvoiceCount DESC
LIMIT 1
  FirstName   LastName InvoiceCount
0   Luís Gonçalves   7
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: '
\n   Find the customer with the most invoices \n'\n\nThe DataFrame was pro
duced using this query: SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) a
s InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.Custom
erId\nGROUP BY c.CustomerId\nORDER BY InvoiceCount DESC\nLIMIT 1\n\nThe foll
owing is information about the resulting pandas DataFrame 'df': \nRunning d

f.dtypes gives:\n FirstName object\nLastName object\nInvoiceCou
nt int64\ndtype: object"}, {"role": "user", "content": "Can you generate
the Python plotly code to chart the results of the dataframe? Assume the dat
a is in a pandas dataframe called 'df'. If there is only one value in the da
taframe, use an Indicator. Respond with only Python code. Do not answer with
any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:55:57.772440694
Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.ex
press as px\n\nfig = px.bar(\n df,\n x='FirstName',\n y='InvoiceCou
nt',\n title='Customer with the Most Invoices'\n)\n\nfig.update_traces(ma
rker_color='blue')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': Tru
e, 'total_duration': 20095829943, 'load_duration': 23544235, 'prompt_eval_co
unt': 219, 'prompt_eval_duration': 7265405000, 'eval_count': 67, 'eval_durat
ion': 12675048000}



Out[34]:
```
('SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as InvoiceCount\nFROM
customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Cus
tomerId\nORDER BY InvoiceCount DESC\nLIMIT 1',
   FirstName   LastName  InvoiceCount
 0      Luís  Gonçalves             7,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'FirstName=%{x}<br>InvoiceCount=%{y}<extra>
</extra>',
               'legendgroup': '',
               'marker': {'color': 'blue', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Luís'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Customer with the Most Invoices'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'FirstName'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'InvoiceCount'}}}
 }))
```

In [ ]:

Advanced SQL questions

In [35]:
```
question = """
     Find the customer who bought the most albums in total quantity (across
"""

vn.ask(question=question)
```

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

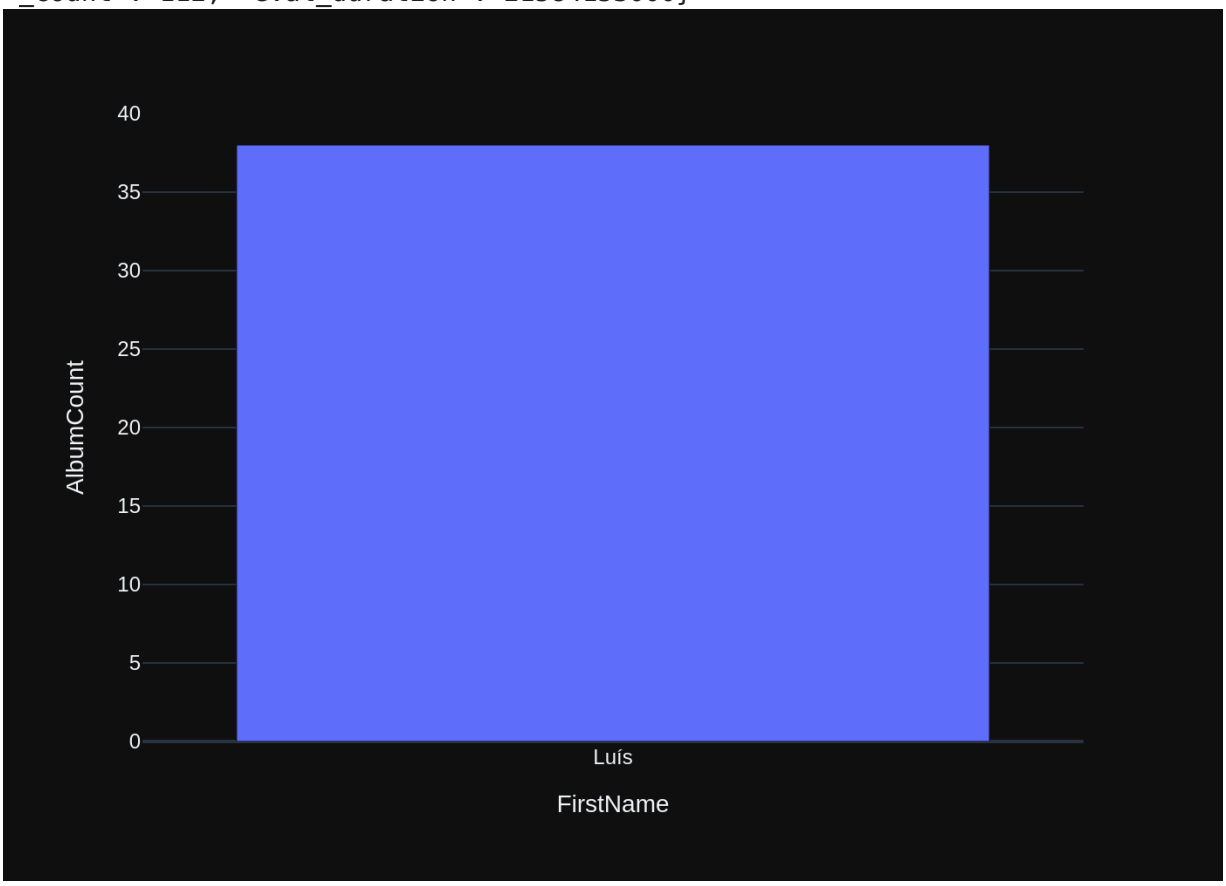
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n   ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY InvoiceCount DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT ar.Name, COUNT(t.TrackId) as TrackCount\nFROM artists ar\nJOIN albums al ON ar.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY ar.ArtistId\nORDER BY TrackCount DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n   Get the total number of i

nvoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.Fi
rstName, c.LastName, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\n
JOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'r
ole': 'user', 'content': ' \n   Find the total number of invoices per coun
try:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.Invoic
eId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId =
i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': ' \n   Lis
t all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'conte
nt': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'user', 'content':
' \n   Get the average invoice total for each customer:\n'}, {'role': 'ass
istant', 'content': 'SELECT c.FirstName, c.LastName, AVG(i.Total) as Average
InvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.Customer
Id\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n   Find the t
op 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant',
'content': 'SELECT * FROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'rol
e': 'user', 'content': ' \n   List all albums and their corresponding arti
st names \n'}, {'role': 'assistant', 'content': 'SELECT al.Title as AlbumTi
tle, ar.Name as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId =
ar.ArtistId'}, {'role': 'user', 'content': ' \n   List all genres and the
number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELEC
T g.Name, COUNT(t.TrackId) as TrackCount\nFROM genres g\nJOIN tracks t ON g.
GenreId = t.GenreId\nGROUP BY g.GenreId'}, {'role': 'user', 'content': 'How
many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT
(*) FROM customers'}, {'role': 'user', 'content': ' \n   Find the custome
r who bought the most albums in total quantity (across all invoices): \n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n   TrackId INTEGER PRIMA
RY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   A
lbumId 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 NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY
(AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPD
ATE 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 (MediaTy
peId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION O
N UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice_items\"\r\n(\r\n   Invoic
eLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER
NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2)
NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId)
REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t
\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r
\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title
NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN K
EY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums\" (Ar
tistId)\n\nCREATE TABLE \"invoices\"\r\n(\r\n   InvoiceId INTEGER PRIMARY K
EY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   Invoi
ceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   Billin

gCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON \"invoice_items\" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"artists\"\r\n(\r\n   ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n   Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY InvoiceCount DESC\nLIMIT 1"}, {"role": "user", "content": " \n   There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most popular artists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT ar.Name, COUNT(t.TrackId) as TrackCount\nFROM artists ar\nJOIN albums al ON ar.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY ar.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 c.FirstName, c.LastName, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "content": " \n   Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastName, AVG(i.Total) as AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT * FROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT al.Title as AlbumTitle, ar.Name as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n   List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.TrackId) as TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": " \n   Find the customer who bought the most albums in total quantity (across all invoices): \n"}]

```
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:57:22.616484971
Z', 'message': {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastN
ame, SUM(ii.Quantity) as AlbumCount\nFROM customers c\nJOIN invoices i ON c.
CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.Invoice
Id\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = a
l.AlbumId\nGROUP BY c.CustomerId\nORDER BY AlbumCount DESC\nLIMIT 1'}, 'done
_reason': 'stop', 'done': True, 'total_duration': 84758332455, 'load_duratio
n': 22076691, 'prompt_eval_count': 1718, 'prompt_eval_duration': 6132755000
0, 'eval_count': 101, 'eval_duration': 22065161000}
LLM Response: SELECT c.FirstName, c.LastName, SUM(ii.Quantity) as AlbumCount
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId
JOIN tracks t ON ii.TrackId = t.TrackId
JOIN albums al ON t.AlbumId = al.AlbumId
GROUP BY c.CustomerId
ORDER BY AlbumCount DESC
LIMIT 1
SELECT c.FirstName, c.LastName, SUM(ii.Quantity) as AlbumCount
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId
JOIN tracks t ON ii.TrackId = t.TrackId
JOIN albums al ON t.AlbumId = al.AlbumId
GROUP BY c.CustomerId
ORDER BY AlbumCount DESC
LIMIT 1
  FirstName   LastName  AlbumCount
0      Luís  Gonçalves          38
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: '
\n     Find the customer who bought the most albums in total quantity (acros
s all invoices): \n'\n\nThe DataFrame was produced using this query: SELECT
c.FirstName, c.LastName, SUM(ii.Quantity) as AlbumCount\nFROM customers c\nJ
OIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.In
voiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums
al ON t.AlbumId = al.AlbumId\nGROUP BY c.CustomerId\nORDER BY AlbumCount DES
C\nLIMIT 1\n\nThe following is information about the resulting pandas DataFr
ame 'df': \nRunning df.dtypes gives:\n FirstName     object\nLastName      o
bject\nAlbumCount     int64\ndtype: object"}, {"role": "user", "content": "C
an you generate the Python plotly code to chart the results of the datafram
e? Assume the data is in a pandas dataframe called 'df'. If there is only on
e value in the dataframe, use an Indicator. Respond with only Python code. D
o not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:57:53.377145838
Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.ex
press as px\n\nfig = px.bar(df, x='FirstName', y='AlbumCount', title='Custom
ers with Most Albums Purchased')\nfig.update_traces(marker_color='blue')\nfi
g.update_layout(xaxis_title='Customer Name', yaxis_title='Total Albums Purch
```

ased')\n\nif len(df) == 1:\n   fig = px.indicator(df, value='AlbumCount', title='Customer with Most Albums Purchased')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 30738444397, 'load_duration': 20236644, 'prompt_eval_count': 271, 'prompt_eval_duration': 9001002000, 'eval_count': 112, 'eval_duration': 21584153000}



Out[35]:
```
('SELECT c.FirstName, c.LastName, SUM(ii.Quantity) as AlbumCount\nFROM cust
omers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items
ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\n
JOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.CustomerId\nORDER BY A
lbumCount DESC\nLIMIT 1',
   FirstName   LastName  AlbumCount
 0      Luís  Gonçalves          38,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'FirstName=%{x}<br>AlbumCount=%{y}<extra></
extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Luís'], dtype=object),
               'xaxis': 'x',
               'y': array([38]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'FirstName'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'AlbumCount'}}}
 }))
```

In [36]:
```
question = """
    Hint: album quantity is found in invoice_items,

    Find the top 5 customers who bought the most albums in total quantity (a
"""

vn.ask(question=question)
```

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

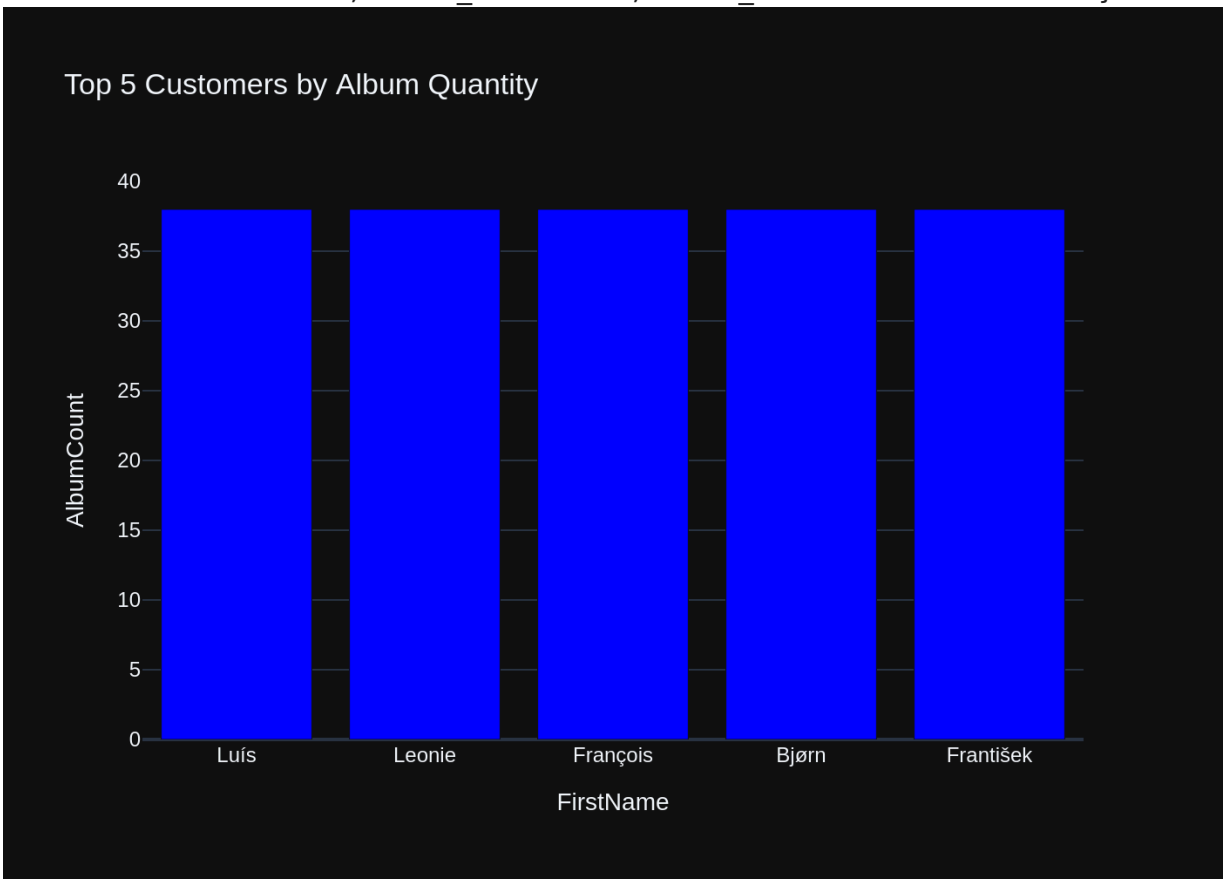
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n   ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, SUM(ii.Quantity) as AlbumCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.CustomerId\nORDER BY AlbumCount DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY InvoiceCount DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   There are 3 tables: artists, alb

ums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT ar.Name, COUNT(t.TrackId) as TrackCount\nFROM artists ar\nJOIN albums al ON ar.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY ar.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\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices\nWHERE Total > 10'}, {'role': 'user', 'content': ' \n   Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'}, {'role': 'user', 'content': ' \n   Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, AVG(i.Total) as AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(*) as CustomerCount FROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT al.Title as AlbumTitle, ar.Name as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE \"invoice_items\"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   T

itle NVARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FORE
IGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO AC
TION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums
\" (ArtistId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON \"invoice_items\"
(InvoiceId)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"invoice_items\" (Tra
ckId)\n\nCREATE TABLE \"invoices\"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY
AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceD
ate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCi
ty NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVA
RCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2)
NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES \"customers\" (Customer
Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK
_InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK_TrackAlb
umId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"artists\"\r\n(\r\n   ArtistI
d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)
\n\n\n===Additional Context \n\nIn the chinook database invoice means order
\n\n===Response Guidelines \n1. If the provided context is sufficient, pleas
e generate a valid SQL query without any explanations for the question. \n2.
If the provided context is almost sufficient but requires knowledge of a spe
cific string in a particular column, please generate an intermediate SQL que
ry to find the distinct strings in that column. Prepend the query with a com
ment saying intermediate_sql \n3. If the provided context is insufficient, p
lease explain why it can't be generated. \n4. Please use the most relevant t
able(s). \n5. If the question has been asked and answered before, please rep
eat the answer exactly as it was given before. \n"}, {"role": "user", "conte
nt": " \n   Find the customer who bought the most albums in total quantit
y (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.Fi
rstName, c.LastName, SUM(ii.Quantity) as AlbumCount\nFROM customers c\nJOIN
invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.Invoic
eId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al
ON t.AlbumId = al.AlbumId\nGROUP BY c.CustomerId\nORDER BY AlbumCount DESC\n
LIMIT 1"}, {"role": "user", "content": " \n   Find the customer with the
most invoices \n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.
LastName, COUNT(i.InvoiceId) as InvoiceCount\nFROM customers c\nJOIN invoice
s i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY InvoiceC
ount DESC\nLIMIT 1"}, {"role": "user", "content": " \n   There are 3 table
s: artists, albums and tracks, where albums and artists are linked by Artist
Id, albums and tracks are linked by AlbumId,\n   Can you find the top 10 mo
st popular artists based on the number of tracks\n"}, {"role": "assistant",
"content": "SELECT ar.Name, COUNT(t.TrackId) as TrackCount\nFROM artists ar
\nJOIN albums al ON ar.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId =
t.AlbumId\nGROUP BY ar.ArtistId\nORDER BY TrackCount DESC\nLIMIT 10"}, {"rol
e": "user", "content": " \n   Find the top 5 most expensive tracks (based
on unit price):\n"}, {"role": "assistant", "content": "SELECT * FROM tracks
\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n   G
et the total number of invoices for each customer\n"}, {"role": "assistant",
"content": "SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as TotalInvoi
ces\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP
BY c.CustomerId"}, {"role": "user", "content": " \n   List all invoices wi
th a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FR
OM invoices WHERE Total > 10"}, {"role": "user", "content": " \n   Find th
e total number of invoices per country:\n"}, {"role": "assistant", "conten
t": "SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c
\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"ro
le": "user", "content": " \n   Get the average invoice total for each cust
omer:\n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastName,

```
AVG(i.Total) as AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.
CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "conten
t": "what are the top 5 countries that customers come from?"}, {"role": "ass
istant", "content": "SELECT Country, COUNT(*) as CustomerCount FROM customer
s\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"role": "use
r", "content": "  \n    List all albums and their corresponding artist names
\n"}, {"role": "assistant", "content": "SELECT al.Title as AlbumTitle, ar.Na
me as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.Artist
Id"}, {"role": "user", "content": "  \n    Hint: album quantity is found in
invoice_items, \n    \n    Find the top 5 customers who bought the most albu
ms in total quantity (across all invoices):\n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:59:21.932116261
Z', 'message': {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastN
ame, SUM(ii.Quantity) as AlbumCount\nFROM customers c\nJOIN invoices i ON c.
CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.Invoice
Id\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = a
l.AlbumId\nGROUP BY c.CustomerId\nORDER BY AlbumCount DESC\nLIMIT 5'}, 'done
_reason': 'stop', 'done': True, 'total_duration': 88459794463, 'load_duratio
n': 17878120, 'prompt_eval_count': 1820, 'prompt_eval_duration': 6504148000
0, 'eval_count': 101, 'eval_duration': 22082603000}
LLM Response: SELECT c.FirstName, c.LastName, SUM(ii.Quantity) as AlbumCount
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId
JOIN tracks t ON ii.TrackId = t.TrackId
JOIN albums al ON t.AlbumId = al.AlbumId
GROUP BY c.CustomerId
ORDER BY AlbumCount DESC
LIMIT 5
SELECT c.FirstName, c.LastName, SUM(ii.Quantity) as AlbumCount
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId
JOIN tracks t ON ii.TrackId = t.TrackId
JOIN albums al ON t.AlbumId = al.AlbumId
GROUP BY c.CustomerId
ORDER BY AlbumCount DESC
LIMIT 5
   FirstName     LastName  AlbumCount
0       Luís    Gonçalves          38
1     Leonie       Köhler          38
2   François     Tremblay          38
3      Bjørn       Hansen          38
4  František  Wichterlová          38
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: '
\n    Hint: album quantity is found in invoice_items, \n    \n    Find the t
op 5 customers who bought the most albums in total quantity (across all invo
ices):\n'\n\nThe DataFrame was produced using this query: SELECT c.FirstNam
e, c.LastName, SUM(ii.Quantity) as AlbumCount\nFROM customers c\nJOIN invoic
```

es i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.CustomerId\nORDER BY AlbumCount DESC\nLIMIT 5\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n FirstName   object\nLastName   object\nAlbumCount   int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T22:59:42.782938117Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='FirstName', y='AlbumCount', title='Top 5 Customers by Album Quantity')\n\nfig.update_traces(marker_color='blue')\n\nfig.show()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 20822605218, 'load_duration': 16865182, 'prompt_eval_count': 289, 'prompt_eval_duration': 9655984000, 'eval_count': 58, 'eval_duration': 11018898000}



Out[36]:
```
('SELECT c.FirstName, c.LastName, SUM(ii.Quantity) as AlbumCount\nFROM cust
omers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items
ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\n
JOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.CustomerId\nORDER BY A
lbumCount DESC\nLIMIT 5',
    FirstName     LastName  AlbumCount
 0       Luís    Gonçalves          38
 1     Leonie       Köhler          38
 2   François     Tremblay          38
 3      Bjørn       Hansen          38
 4  František  Wichterlová          38,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'FirstName=%{x}<br>AlbumCount=%{y}<extra></
extra>',
               'legendgroup': '',
               'marker': {'color': 'blue', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Luís', 'Leonie', 'François', 'Bjørn', 'Františe
k'], dtype=object),
               'xaxis': 'x',
               'y': array([38, 38, 38, 38, 38]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Top 5 Customers by Album Quantity'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'FirstName'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'AlbumCount'}}}
 }))
```

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

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

     Hint: order total can be found on invoices table, calculation using inv
"""

vn.ask(question=question)
```

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

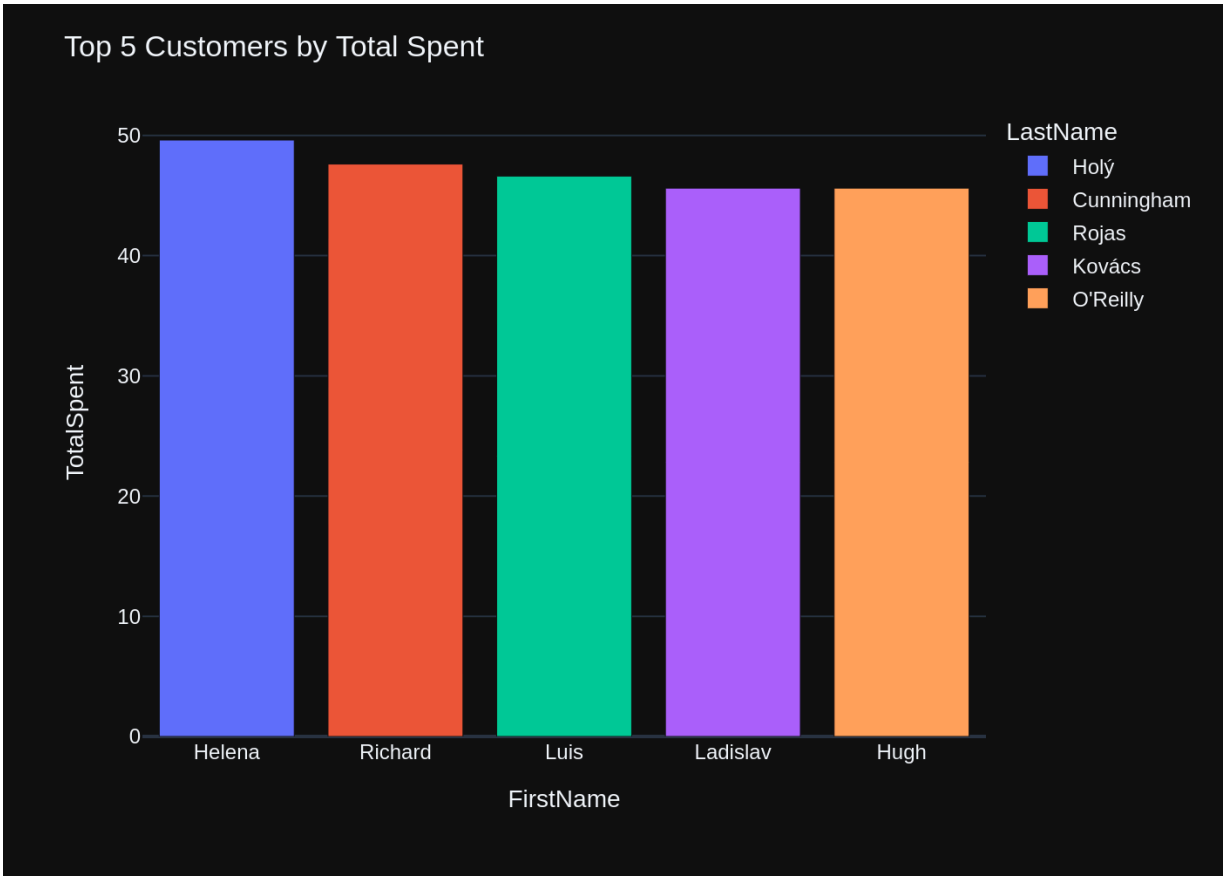
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n   InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n   InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   BillingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCountry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice_items"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   InvoiceId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   Quantity INTEGER NOT NULL,\r\n   FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)\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   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60) NOT NULL,\r\n   SupportRepId INTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "employees"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NOT NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDate DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n   PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(24),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (ReportsTo)\n\n\n===Additional Context \n\nIn the chinook database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context 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 qu

ery 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 question has been asked and answered bef
ore, please repeat the answer exactly as it was given before. \n'}, {'role':
'user', 'content': ' \n   Hint: album quantity is found in invoice_items,
\n   \n   Find the top 5 customers who bought the most albums in total qua
ntity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT
c.FirstName, c.LastName, SUM(ii.Quantity) as AlbumCount\nFROM customers c\nJ
OIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.In
voiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums
al ON t.AlbumId = al.AlbumId\nGROUP BY c.CustomerId\nORDER BY AlbumCount DES
C\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find the customer with t
he most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName,
c.LastName, COUNT(i.InvoiceId) as InvoiceCount\nFROM customers c\nJOIN invoi
ces i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY Invoic
eCount DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   Find the custo
mer who bought the most albums in total quantity (across all invoices):
\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, SUM
(ii.Quantity) as AlbumCount\nFROM customers c\nJOIN invoices i ON c.Customer
Id = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN
tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId
\nGROUP BY c.CustomerId\nORDER BY AlbumCount DESC\nLIMIT 1'}, {'role': 'use
r', 'content': ' \n   Get the average invoice total for each custome
r:\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, AV
G(i.Total) as AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.Cu
stomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'conten
t': ' \n   Get the total number of invoices for each customer\n'}, {'rol
e': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, COUNT(i.Invoice
Id) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.
CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n   Fi
nd the total number of invoices per country:\n'}, {'role': 'assistant', 'con
tent': 'SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices\nFROM customer
s c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country'},
{'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (b
ased on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tr
acks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': 'what
are the top 5 countries that customers come from?'}, {'role': 'assistant',
'content': 'SELECT Country, COUNT(*) as CustomerCount FROM customers\nGROUP
BY Country\nORDER BY CustomerCount DESC\nLIMIT 5'}, {'role': 'user', 'conten
t': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'as
sistant', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'u
ser', 'content': ' \n   There are 3 tables: artists, albums and tracks, whe
re albums and artists are linked by ArtistId, albums and tracks are linked b
y AlbumId,\n   Can you find the top 10 most popular artists based on the nu
mber of tracks\n'}, {'role': 'assistant', 'content': 'SELECT ar.Name, COUNT
(t.TrackId) as TrackCount\nFROM artists ar\nJOIN albums al ON ar.ArtistId =
al.ArtistId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY ar.ArtistId\n
ORDER BY TrackCount DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n
Find the top 5 customers who spent the most money overall, \n   \n   Hin
t: order total can be found on invoices table, calculation using invoice_ite
ms detail table is unnecessary \n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:

[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n   InvoiceId INTEGER P
RIMARY KEY AUTOINCREMENT NOT NULL,\r\n   CustomerId INTEGER NOT NULL,\r\n
InvoiceDate DATETIME NOT NULL,\r\n   BillingAddress NVARCHAR(70),\r\n   B
illingCity NVARCHAR(40),\r\n   BillingState NVARCHAR(40),\r\n   BillingCou
ntry NVARCHAR(40),\r\n   BillingPostalCode NVARCHAR(10),\r\n   Total NUMER
IC(10,2) NOT NULL,\r\n   FOREIGN KEY (CustomerId) REFERENCES \"customers\"
(CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE
TABLE \"invoice_items\"\r\n(\r\n   InvoiceLineId INTEGER PRIMARY KEY AUTOIN
CREMENT 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 (TrackI
d) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION\r\n)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON \"invoice_items\" (I
nvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON \"invoices\" (CustomerId)
\n\nCREATE INDEX IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCR
EATE TABLE \"customers\"\r\n(\r\n   CustomerId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n   FirstName NVARCHAR(40) NOT NULL,\r\n   LastName NVA
RCHAR(20) NOT NULL,\r\n   Company NVARCHAR(80),\r\n   Address NVARCHAR(7
0),\r\n   City NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVAR
CHAR(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 I
NTEGER,\r\n   FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (Employee
Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"e
mployees\"\r\n(\r\n   EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n   LastName NVARCHAR(20) NOT NULL,\r\n   FirstName NVARCHAR(20) NO
T NULL,\r\n   Title NVARCHAR(30),\r\n   ReportsTo INTEGER,\r\n   BirthDat
e DATETIME,\r\n   HireDate DATETIME,\r\n   Address NVARCHAR(70),\r\n   Ci
ty NVARCHAR(40),\r\n   State NVARCHAR(40),\r\n   Country NVARCHAR(40),\r\n
PostalCode NVARCHAR(10),\r\n   Phone NVARCHAR(24),\r\n   Fax NVARCHAR(2
4),\r\n   Email NVARCHAR(60),\r\n   FOREIGN KEY (ReportsTo) REFERENCES \"e
mployees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)
\n\nCREATE TABLE \"tracks\"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCR
EMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGE
R,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Comp
oser NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTE
GER,\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 ACT
ION,\r\n   FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFER
ENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n   PlaylistId INTEGE
R NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_Playlist
Track PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) RE
FERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO
ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t
\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeR
eportsTo ON \"employees\" (ReportsTo)\n\n\n===Additional Context \n\nIn the
chinook database invoice means order\n\n===Response Guidelines \n1. If the p
rovided context is sufficient, please generate a valid SQL query without any
explanations for the question. \n2. If the provided context is almost suffic
ient 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. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastName, SUM(ii.Quantity) as AlbumCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.CustomerId\nORDER BY AlbumCount DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as InvoiceCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY InvoiceCount DESC\nLIMIT 1"}, {"role": "user", "content": " \n   Find the customer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastName, SUM(ii.Quantity) as AlbumCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.CustomerId\nORDER BY AlbumCount DESC\nLIMIT 1"}, {"role": "user", "content": " \n   Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastName, AVG(i.Total) as AverageInvoiceTotal\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   Get the total number of invoices for each customer \n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n   Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) as TotalInvoices\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Country"}, {"role": "user", "content": " \n   Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT * FROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(*) as CustomerCount FROM customers\nGROUP BY Country\nORDER BY CustomerCount DESC\nLIMIT 5"}, {"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   There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most popular artists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT ar.Name, COUNT(t.TrackId) as TrackCount\nFROM artists ar\nJOIN albums al ON ar.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY ar.ArtistId\nORDER BY TrackCount DESC\nLIMIT 10"}, {"role": "user", "content": " \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T23:01:07.410874076Z', 'message': {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, SUM(i.Total) as TotalSpent\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMI

```
T 5'}, 'done_reason': 'stop', 'done': True, 'total_duration': 84541199089,
'load_duration': 20948089, 'prompt_eval_count': 1971, 'prompt_eval_duratio
n': 70945909000, 'eval_count': 57, 'eval_duration': 12476086000}
LLM Response: SELECT c.FirstName, c.LastName, SUM(i.Total) as TotalSpent
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 5
SELECT c.FirstName, c.LastName, SUM(i.Total) as TotalSpent
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 5
  FirstName    LastName  TotalSpent
0    Helena        Holý       49.62
1   Richard  Cunningham       47.62
2      Luis       Rojas       46.62
3  Ladislav      Kovács       45.62
4      Hugh    O'Reilly       45.62
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: '
\n     Find the top 5 customers who spent the most money overall, \n     \n
Hint: order total can be found on invoices table, calculation using invoice_
items detail table is unnecessary \n'\n\nThe DataFrame was produced using th
is query: SELECT c.FirstName, c.LastName, SUM(i.Total) as TotalSpent\nFROM c
ustomers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Custo
merId\nORDER BY TotalSpent DESC\nLIMIT 5\n\nThe following is information abo
ut the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n FirstNa
me      object\nLastName       object\nTotalSpent    float64\ndtype: objec
t"}, {"role": "user", "content": "Can you generate the Python plotly code to
chart the results of the dataframe? Assume the data is in a pandas dataframe
called 'df'. If there is only one value in the dataframe, use an Indicator.
Respond with only Python code. Do not answer with any explanations -- just t
he code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T23:01:26.408864129
Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.ex
press as px\n\nfig = px.bar(df, x='FirstName', y='TotalSpent', color='LastNa
me')\nfig.update_layout(title='Top 5 Customers by Total Spent')\nfig.show()
\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 18970565508,
'load_duration': 17461278, 'prompt_eval_count': 249, 'prompt_eval_duration':
8199805000, 'eval_count': 56, 'eval_duration': 10705192000}
```


Top 5 Customers by Total Spent
TotalSpent
50
40
30
20
10
0
Helena
Richard
Luis
Ladislav
Hugh
FirstName
LastName
Holý
Cunningham
Rojas
Kovács
O'Reilly

Out[37]:
```
('SELECT c.FirstName, c.LastName, SUM(i.Total) as TotalSpent\nFROM customer
s c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId
\nORDER BY TotalSpent DESC\nLIMIT 5',
   FirstName    LastName  TotalSpent
 0    Helena        Holý       49.62
 1   Richard  Cunningham       47.62
 2      Luis       Rojas       46.62
 3  Ladislav      Kovács       45.62
 4      Hugh    O'Reilly       45.62,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Holý<br>FirstName=%{x}<br>TotalSp
ent=%{y}<extra></extra>',
               'legendgroup': 'Holý',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': 'Holý',
               'offsetgroup': 'Holý',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Helena'], dtype=object),
               'xaxis': 'x',
               'y': array([49.62]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Cunningham<br>FirstName=%{x}<br>T
otalSpent=%{y}<extra></extra>',
               'legendgroup': 'Cunningham',
               'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
               'name': 'Cunningham',
               'offsetgroup': 'Cunningham',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Richard'], dtype=object),
               'xaxis': 'x',
               'y': array([47.62]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Rojas<br>FirstName=%{x}<br>TotalS
pent=%{y}<extra></extra>',
               'legendgroup': 'Rojas',
               'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
               'name': 'Rojas',
               'offsetgroup': 'Rojas',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Luis'], dtype=object),
               'xaxis': 'x',
               'y': array([46.62]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
```

```
                'hovertemplate': 'LastName=Kovács<br>FirstName=%{x}<br>Total
Spent=%{y}<extra></extra>',
                'legendgroup': 'Kovács',
                'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
                'name': 'Kovács',
                'offsetgroup': 'Kovács',
                'orientation': 'v',
                'showlegend': True,
                'textposition': 'auto',
                'type': 'bar',
                'x': array(['Ladislav'], dtype=object),
                'xaxis': 'x',
                'y': array([45.62]),
                'yaxis': 'y'},
               {'alignmentgroup': 'True',
                'hovertemplate': "LastName=O'Reilly<br>FirstName=%{x}<br>Tot
alSpent=%{y}<extra></extra>",
                'legendgroup': "O'Reilly",
                'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
                'name': "O'Reilly",
                'offsetgroup': "O'Reilly",
                'orientation': 'v',
                'showlegend': True,
                'textposition': 'auto',
                'type': 'bar',
                'x': array(['Hugh'], dtype=object),
                'xaxis': 'x',
                'y': array([45.62]),
                'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'title': {'text': 'LastName'}, 'tracegroupgap':
0},
                'margin': {'t': 60},
                'template': '...',
                'title': {'text': 'Top 5 Customers by Total Spent'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'FirstName'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'TotalSpent'}}}
 }))
```

In [38]:

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

vn.ask(question=question)
```

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

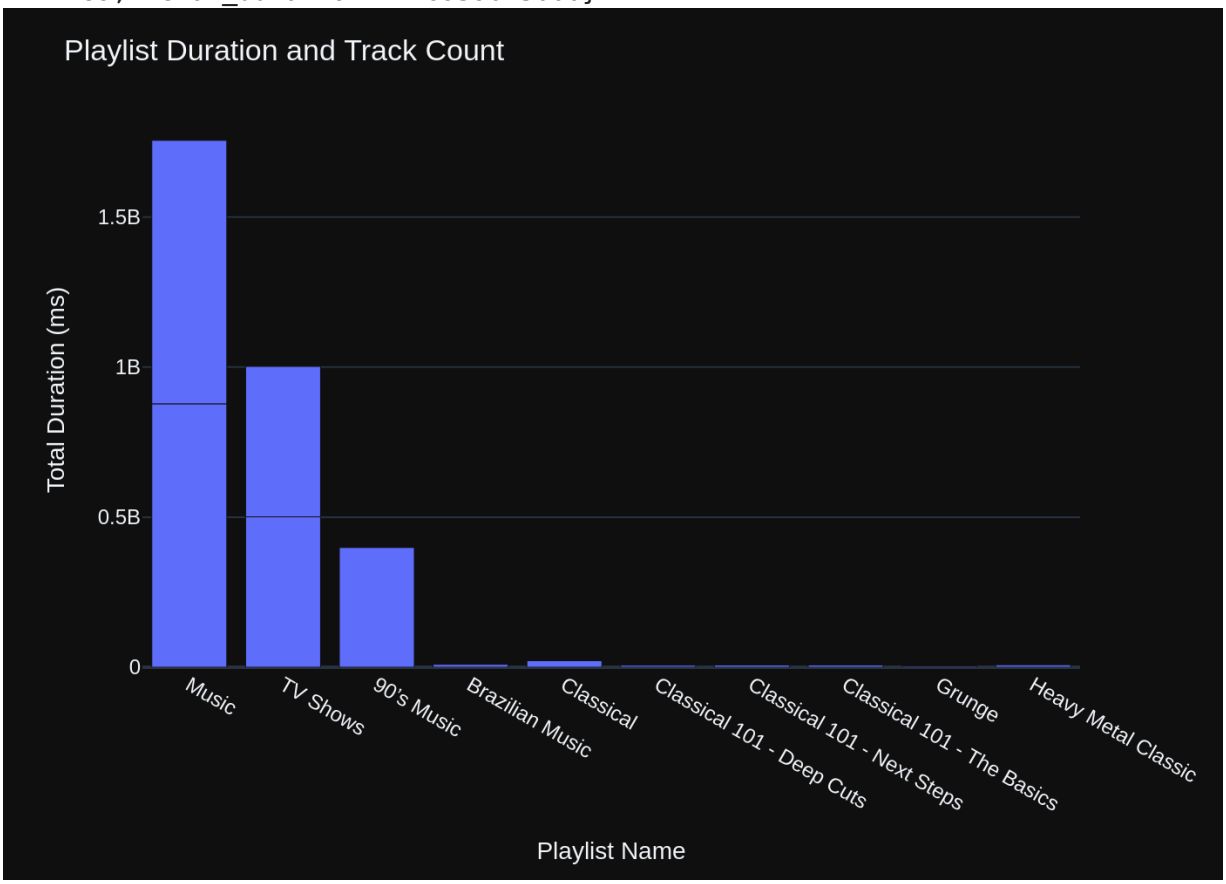
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE TABLE "playlists"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "albums"\r\n(\r\n   AlbumId 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 "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 provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n   List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.TrackId) as TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId'}, {'role': 'user', 'content': ' \n   There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based on the number of tracks \n'}, {'role': 'assistant', 'content': 'SELECT ar.Name, COUNT(t.TrackId) as TrackCount\nFROM artists ar\nJOIN albums al ON ar.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY ar.ArtistId\nORDER BY TrackCount DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n   Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, SUM(ii.Quantity) as AlbumCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.CustomerId\nORDER BY AlbumCount DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n   Hint: album quantity is found in invoice_items, \n   \n   F

ind the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, SUM(ii.Quantity) as AlbumCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.CustomerId\nORDER BY AlbumCount DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%What%'"}, {'role': 'user', 'content': ' \n   Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT al.Title as AlbumTitle, ar.Name as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_master WHERE type = 'table'"}, {'role': 'user', 'content': ' \n   List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * FROM invoices WHERE Total > 10'}, {'role': 'user', 'content': ' \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, SUM(i.Total) as TotalSpent\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Get all playlists containing at least 10 tracks and the total duration of those tracks:\n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and format instructions. \n===Tables \nCREATE INDEX IFK_PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n   PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n   PlaylistId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n   TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   AlbumId INTEGER,\r\n   MediaTypeId INTEGER NOT NULL,\r\n   GenreId INTEGER,\r\n   Composer NVARCHAR(220),\r\n   Milliseconds INTEGER NOT NULL,\r\n   Bytes INTEGER,\r\n   UnitPrice NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n   FOREIGN KEY (MediaTypeId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDEX IFK_TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"albums\"\r\n(\r\n   Alb

umId 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 \"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 provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate_sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n   List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.TrackId) as TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.GenreId\nGROUP BY g.GenreId"}, {"role": "user", "content": " \n   There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the top 10 most popular artists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT ar.Name, COUNT(t.TrackId) as TrackCount\nFROM artists ar\nJOIN albums al ON ar.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY ar.ArtistId\nORDER BY TrackCount DESC\nLIMIT 10"}, {"role": "user", "content": " \n   Find the customer who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastName, SUM(ii.Quantity) as AlbumCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.CustomerId\nORDER BY AlbumCount DESC\nLIMIT 1"}, {"role": "user", "content": " \n   Hint: album quantity is found in invoice_items, \n   \n   Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastName, SUM(ii.Quantity) as AlbumCount\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.CustomerId\nORDER BY AlbumCount DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks WHERE Name LIKE '%What%'"}, {"role": "user", "content": " \n   Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT * FROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n   List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT al.Title as AlbumTitle, ar.Name as ArtistName\nFROM albums al\nJOIN artists ar ON al.ArtistId = ar.ArtistId"}, {"role": "user", "content": "Can you list all tables in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_master WHERE type = 'table'"}, {"role": "user", "content": " \n   List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * FROM invoices WHERE Total > 10"}, {"role": "user", "content": " \n   Find the top 5 customers who spent the most money overall, \n   \n   Hint: order total can be found on invoices table, calculation using invoice_items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastName, SUM(i.Total) as TotalSpent\nFROM customers c\nJOIN invoices i ON c.CustomerId = i.Custome

```
rId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "u
ser", "content": "  \n     Get all playlists containing at least 10 tracks a
nd the total duration of those tracks:\n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T23:02:48.949861178
Z', 'message': {'role': 'assistant', 'content': 'SELECT p.Name as PlaylistNa
me, COUNT(pt.TrackId) as TrackCount, SUM(t.Milliseconds) as TotalDuration\nF
ROM playlists p\nJOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId\nJOI
N tracks t ON pt.TrackId = t.TrackId\nGROUP BY p.PlaylistId\nHAVING TrackCou
nt >= 10'}, 'done_reason': 'stop', 'done': True, 'total_duration': 824286321
29, 'load_duration': 22104553, 'prompt_eval_count': 1761, 'prompt_eval_durat
ion': 62675953000, 'eval_count': 85, 'eval_duration': 18494667000}
LLM Response: SELECT p.Name as PlaylistName, COUNT(pt.TrackId) as TrackCoun
t, SUM(t.Milliseconds) as TotalDuration
FROM playlists p
JOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId
JOIN tracks t ON pt.TrackId = t.TrackId
GROUP BY p.PlaylistId
HAVING TrackCount >= 10
SELECT p.Name as PlaylistName, COUNT(pt.TrackId) as TrackCount, SUM(t.Millis
econds) as TotalDuration
FROM playlists p
JOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId
JOIN tracks t ON pt.TrackId = t.TrackId
GROUP BY p.PlaylistId
HAVING TrackCount >= 10
                  PlaylistName  TrackCount  TotalDuration
0                        Music        3290      877683083
1                    TV Shows         213      501094957
2                  90’s Music        1477      398705153
3                        Music        3290      877683083
4                    TV Shows         213      501094957
5              Brazilian Music          39        9486559
6                    Classical          75       21770592
7    Classical 101 - Deep Cuts          25        6755730
8   Classical 101 - Next Steps          25        7575051
9   Classical 101 - The Basics          25        7439811
10                      Grunge          15        4122018
11        Heavy Metal Classic           26        8206312
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: '
\n     Get all playlists containing at least 10 tracks and the total duratio
n of those tracks:\n'\n\nThe DataFrame was produced using this query: SELECT
p.Name as PlaylistName, COUNT(pt.TrackId) as TrackCount, SUM(t.Milliseconds)
as TotalDuration\nFROM playlists p\nJOIN playlist_track pt ON p.PlaylistId =
pt.PlaylistId\nJOIN tracks t ON pt.TrackId = t.TrackId\nGROUP BY p.PlaylistI
d\nHAVING TrackCount >= 10\n\nThe following is information about the resulti
ng pandas DataFrame 'df': \nRunning df.dtypes gives:\n PlaylistName     obje
ct\nTrackCount        int64\nTotalDuration     int64\ndtype: object"}, {"rol
e": "user", "content": "Can you generate the Python plotly code to chart the
results of the dataframe? Assume the data is in a pandas dataframe called 'd
```

f'. If there is only one value in the dataframe, use an Indicator. Respond w
ith only Python code. Do not answer with any explanations -- just the cod
e."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T23:03:14.697158049
Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.ex
press as px\n\nfig = px.bar(df, x='PlaylistName', y='TotalDuration', hover_n
ame='TrackCount', hover_data=['TrackCount'])\n\nfig.update_layout(\n   titl
e='Playlist Duration and Track Count',\n   xaxis_title='Playlist Name',\n
yaxis_title='Total Duration (ms)',\n)\n\nfig.show()\n```"}, 'done_reason':
'stop', 'done': True, 'total_duration': 25725031794, 'load_duration': 234737
31, 'prompt_eval_count': 260, 'prompt_eval_duration': 8694272000, 'eval_coun
t': 89, 'eval_duration': 16959075000}



Out[38]:
```
('SELECT p.Name as PlaylistName, COUNT(pt.TrackId) as TrackCount, SUM(t.Mil
liseconds) as TotalDuration\nFROM playlists p\nJOIN playlist_track pt ON p.
PlaylistId = pt.PlaylistId\nJOIN tracks t ON pt.TrackId = t.TrackId\nGROUP
BY p.PlaylistId\nHAVING TrackCount >= 10',
                   PlaylistName  TrackCount  TotalDuration
 0                        Music        3290      877683083
 1                    TV Shows         213      501094957
 2                   90’s Music        1477      398705153
 3                        Music        3290      877683083
 4                    TV Shows         213      501094957
 5              Brazilian Music          39        9486559
 6                    Classical          75       21770592
 7    Classical 101 - Deep Cuts          25        6755730
 8   Classical 101 - Next Steps          25        7575051
 9   Classical 101 - The Basics          25        7439811
 10                      Grunge          15        4122018
 11        Heavy Metal Classic          26        8206312,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'customdata': array([[3290],
                                    [ 213],
                                    [1477],
                                    [3290],
                                    [ 213],
                                    [  39],
                                    [  75],
                                    [  25],
                                    [  25],
                                    [  25],
                                    [  15],
                                    [  26]]),
               'hovertemplate': ('<b>%{hovertext}</b><br><br>Pla' ... '{cus
tomdata[0]}<extra></extra>'),
               'hovertext': array([3290.,  213., 1477., 3290.,  213.,   3
9.,   75.,   25.,   25.,   25.,
                                    15.,   26.]),
               '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 - T
he Basics', 'Grunge',
                           'Heavy Metal Classic'], dtype=object),
               'xaxis': 'x',
               'y': array([877683083, 501094957, 398705153, 877683083, 5010
94957,   9486559,
                            21770592,   6755730,   7575051,   7439811,   41
22018,   8206312]),
```

```
                  'yaxis': 'y'}],
       'layout': {'barmode': 'relative',
                  'legend': {'tracegroupgap': 0},
                  'margin': {'t': 60},
                  'template': '...',
                  'title': {'text': 'Playlist Duration and Track Count'},
                  'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Playlist Name'}},
                  'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Total Duration (ms)'}}}
        }))
```

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


"""

vn.ask(question=question)
```

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

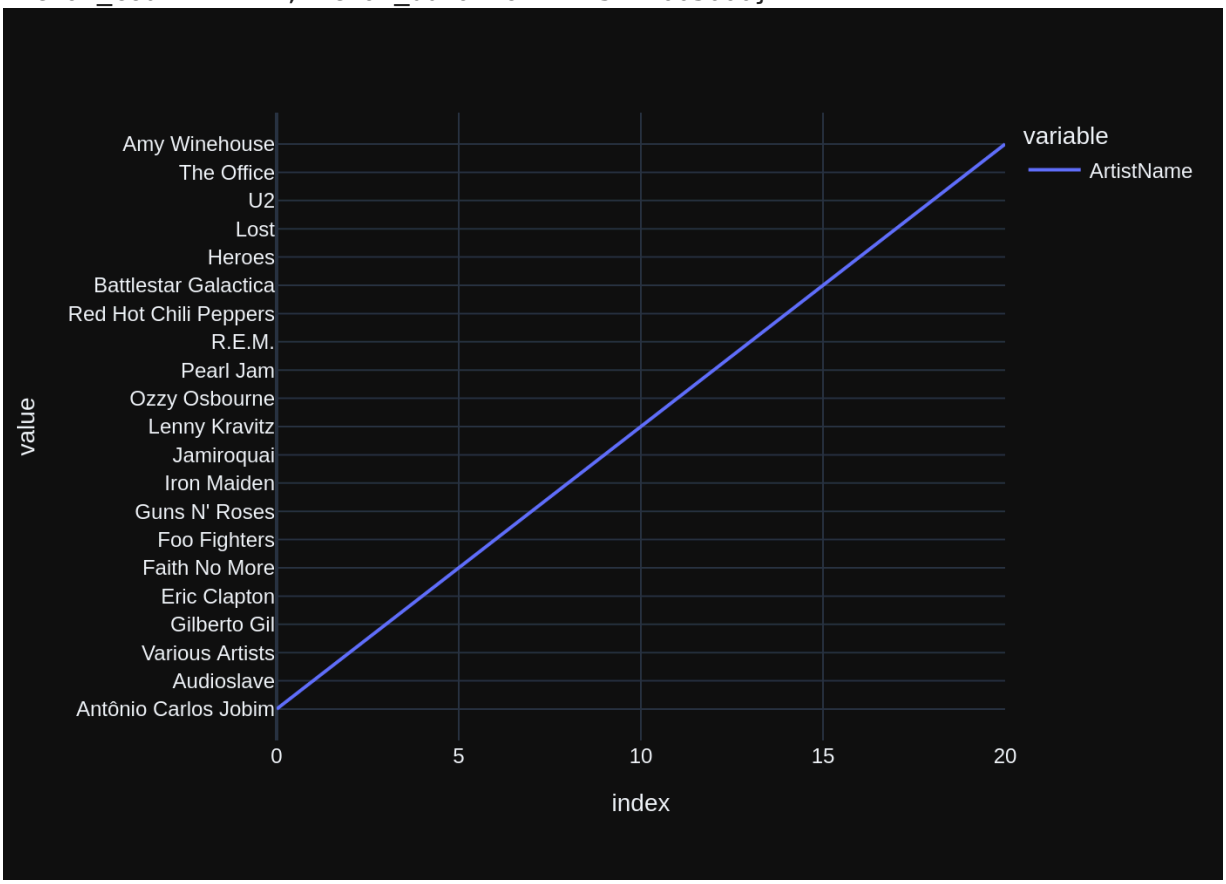
```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please
help to generate a SQL query to answer the question. Your response should ON
LY be based on the given context and follow the response guidelines and form
at instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n    TrackId INT
EGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NUL
L,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    Genr
eId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT
NULL,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n
FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACT
ION 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 AC
TION ON UPDATE NO 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 INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARC
HAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (A
rtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDA
TE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackMediaTypeId ON "tracks" (MediaTyp
eId)\n\nCREATE TABLE "genres"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOIN
CREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_Playlis
tTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE TABLE "artists"\r\n(\r
\n    ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARC
HAR(120)\r\n)\n\nCREATE TABLE "playlist_track"\r\n(\r\n    PlaylistId INTEGE
R  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_Playlist
Track PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) RE
FERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO A
CTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON
DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional Context \n\nIn
the chinook database invoice means order\n\n===Response Guidelines \n1. If 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 su
fficient but requires knowledge of a specific string in a particular column,
please generate an intermediate SQL query to find the distinct strings in th
at column. Prepend the query with a comment saying intermediate_sql \n3. If
the provided context is insufficient, please explain why it can\'t be genera
ted. \n4. Please use the most relevant table(s). \n5. If the question has be
en asked and answered before, please repeat the answer exactly as it was giv
en before. \n'}, {'role': 'user', 'content': ' \n    There are 3 tables: art
ists, albums and tracks, where albums and artists are linked by ArtistId, al
bums and tracks are linked by AlbumId,\n    Can you find the top 10 most pop
ular artists based on the number of tracks\n'}, {'role': 'assistant', 'conte
nt': 'SELECT ar.Name, COUNT(t.TrackId) as TrackCount\nFROM artists ar\nJOIN
albums al ON ar.ArtistId = al.ArtistId\nJOIN tracks t ON al.AlbumId = t.Albu
mId\nGROUP BY ar.ArtistId\nORDER BY TrackCount DESC\nLIMIT 10'}, {'role': 'u
ser', 'content': '  \n    List all genres and the number of tracks in each g
enre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.TrackId)
as TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId = t.GenreId\nGROUP
BY g.GenreId'}, {'role': 'user', 'content': '  \n    List all albums and the
ir corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELEC
T al.Title as AlbumTitle, ar.Name as ArtistName\nFROM albums al\nJOIN artist
s ar ON al.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': '  \n     F
ind the customer who bought the most albums in total quantity (across all in
voices): \n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastN
ame, SUM(ii.Quantity) as AlbumCount\nFROM customers c\nJOIN invoices i ON c.
CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.InvoiceId = ii.Invoice
```

Id\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al ON t.AlbumId = a
l.AlbumId\nGROUP BY c.CustomerId\nORDER BY AlbumCount DESC\nLIMIT 1'}, {'rol
e': 'user', 'content': ' \n   Hint: album quantity is found in invoice_ite
ms, \n   \n   Find the top 5 customers who bought the most albums in total
quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELEC
T c.FirstName, c.LastName, SUM(ii.Quantity) as AlbumCount\nFROM customers c
\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON
i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN al
bums al ON t.AlbumId = al.AlbumId\nGROUP BY c.CustomerId\nORDER BY AlbumCoun
t DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n   Get all playlists c
ontaining at least 10 tracks and the total duration of those tracks:\n'},
{'role': 'assistant', 'content': 'SELECT p.Name as PlaylistName, COUNT(pt.Tr
ackId) as TrackCount, SUM(t.Milliseconds) as TotalDuration\nFROM playlists p
\nJOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId\nJOIN tracks t ON p
t.TrackId = t.TrackId\nGROUP BY p.PlaylistId\nHAVING TrackCount >= 10'}, {'r
ole': 'user', 'content': ' \n   Find the top 5 most expensive tracks (base
d on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT * FROM track
s\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n
Find all tracks with a name containing "What" (case-insensitive)\n'}, {'rol
e': 'assistant', 'content': "SELECT * FROM tracks WHERE Name LIKE '%Wha
t%'"}, {'role': 'user', 'content': 'Can you list all tables in the SQLite da
tabase catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite
_master WHERE type = 'table'"}, {'role': 'user', 'content': ' \n   Find t
he customer with the most invoices \n'}, {'role': 'assistant', 'content': 'S
ELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as InvoiceCount\nFROM cust
omers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Customer
Id\nORDER BY InvoiceCount DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n
Identify artists who have albums with tracks appearing in multiple genres:\n
\n\n'}]
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "You are a SQLite expert. Please help to gene
rate a SQL query to answer the question. Your response should ONLY be based
on the given context and follow the response guidelines and format instructi
ons. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n   TrackId INTEGER PRIMA
RY KEY AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(200) NOT NULL,\r\n   A
lbumId 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 NUMERIC(10,2) NOT NULL,\r\n   FOREIGN KEY
(AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPD
ATE 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 (MediaTy
peId) REFERENCES \"media_types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION O
N UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON \"albums\" (Art
istId)\n\nCREATE INDEX IFK_TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE IN
DEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"albums\"\r\n
(\r\n   AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n   Title NV
ARCHAR(160) NOT NULL,\r\n   ArtistId INTEGER NOT NULL,\r\n   FOREIGN KEY
(ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON
UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackMediaTypeId ON \"tracks\" (Me
diaTypeId)\n\nCREATE TABLE \"genres\"\r\n(\r\n   GenreId INTEGER PRIMARY KE
Y AUTOINCREMENT NOT NULL,\r\n   Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK
_PlaylistTrackTrackId ON \"playlist_track\" (TrackId)\n\nCREATE TABLE \"arti

sts\"\r\n(\r\n   ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n
Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"playlist_track\"\r\n(\r\n   Playl
istId INTEGER NOT NULL,\r\n   TrackId INTEGER NOT NULL,\r\n   CONSTRAINT
PK_PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n   FOREIGN KEY (Pla
ylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION O
N UPDATE NO ACTION,\r\n   FOREIGN KEY (TrackId) REFERENCES \"tracks\" (Trac
kId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n\n===Additional
Context \n\nIn the chinook database invoice means order\n\n===Response Guide
lines \n1. If the provided context is sufficient, please generate a valid SQ
L query without any explanations for the question. \n2. If the provided cont
ext is almost sufficient but requires knowledge of a specific string in a pa
rticular column, please generate an intermediate SQL query to find the disti
nct strings in that column. Prepend the query with a comment saying intermed
iate_sql \n3. If the provided context is insufficient, please explain why it
can't be generated. \n4. Please use the most relevant table(s). \n5. If the
question has been asked and answered before, please repeat the answer exactl
y as it was given before. \n"}, {"role": "user", "content": " \n   There ar
e 3 tables: artists, albums and tracks, where albums and artists are linked
by ArtistId, albums and tracks are linked by AlbumId,\n   Can you find the
top 10 most popular artists based on the number of tracks\n"}, {"role": "ass
istant", "content": "SELECT ar.Name, COUNT(t.TrackId) as TrackCount\nFROM ar
tists ar\nJOIN albums al ON ar.ArtistId = al.ArtistId\nJOIN tracks t ON al.A
lbumId = t.AlbumId\nGROUP BY ar.ArtistId\nORDER BY TrackCount DESC\nLIMIT 1
0"}, {"role": "user", "content": " \n   List all genres and the number of
tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name,
COUNT(t.TrackId) as TrackCount\nFROM genres g\nJOIN tracks t ON g.GenreId =
t.GenreId\nGROUP BY g.GenreId"}, {"role": "user", "content": " \n   List a
ll albums and their corresponding artist names \n"}, {"role": "assistant",
"content": "SELECT al.Title as AlbumTitle, ar.Name as ArtistName\nFROM album
s al\nJOIN artists ar ON al.ArtistId = ar.ArtistId"}, {"role": "user", "cont
ent": " \n   Find the customer who bought the most albums in total quanti
ty (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.F
irstName, c.LastName, SUM(ii.Quantity) as AlbumCount\nFROM customers c\nJOIN
invoices i ON c.CustomerId = i.CustomerId\nJOIN invoice_items ii ON i.Invoic
eId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.TrackId\nJOIN albums al
ON t.AlbumId = al.AlbumId\nGROUP BY c.CustomerId\nORDER BY AlbumCount DESC\n
LIMIT 1"}, {"role": "user", "content": " \n   Hint: album quantity is foun
d in invoice_items, \n   \n   Find the top 5 customers who bought the most
albums in total quantity (across all invoices):\n"}, {"role": "assistant",
"content": "SELECT c.FirstName, c.LastName, SUM(ii.Quantity) as AlbumCount\n
FROM customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nJOIN invoi
ce_items ii ON i.InvoiceId = ii.InvoiceId\nJOIN tracks t ON ii.TrackId = t.T
rackId\nJOIN albums al ON t.AlbumId = al.AlbumId\nGROUP BY c.CustomerId\nORD
ER BY AlbumCount DESC\nLIMIT 5"}, {"role": "user", "content": " \n   Get
all playlists containing at least 10 tracks and the total duration of those
tracks:\n"}, {"role": "assistant", "content": "SELECT p.Name as PlaylistNam
e, COUNT(pt.TrackId) as TrackCount, SUM(t.Milliseconds) as TotalDuration\nFR
OM playlists p\nJOIN playlist_track pt ON p.PlaylistId = pt.PlaylistId\nJOIN
tracks t ON pt.TrackId = t.TrackId\nGROUP BY p.PlaylistId\nHAVING TrackCount
>= 10"}, {"role": "user", "content": " \n   Find the top 5 most expensive
tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT
* FROM tracks\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "conten
t": " \n   Find all tracks with a name containing \"What\" (case-insensiti
ve)\n"}, {"role": "assistant", "content": "SELECT * FROM tracks WHERE Name L
IKE '%What%'"}, {"role": "user", "content": "Can you list all tables in the
SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FR

```
OM sqlite_master WHERE type = 'table'"}, {"role": "user", "content": "  \n
Find the customer with the most invoices \n"}, {"role": "assistant", "conten
t": "SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) as InvoiceCount\nFRO
M customers c\nJOIN invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.Cu
stomerId\nORDER BY InvoiceCount DESC\nLIMIT 1"}, {"role": "user", "content":
"  \n     Identify artists who have albums with tracks appearing in multiple
genres:\n\n\n"}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T23:04:35.633492481
Z', 'message': {'role': 'assistant', 'content': 'SELECT ar.Name as ArtistNam
e\nFROM artists ar\nJOIN albums al ON ar.ArtistId = al.ArtistId\nJOIN tracks
t ON al.AlbumId = t.AlbumId\nGROUP BY ar.ArtistId\nHAVING COUNT(DISTINCT t.G
enreId) > 1'}, 'done_reason': 'stop', 'done': True, 'total_duration': 808309
53609, 'load_duration': 22783654, 'prompt_eval_count': 1800, 'prompt_eval_du
ration': 64725822000, 'eval_count': 68, 'eval_duration': 14851738000}
LLM Response: SELECT ar.Name as ArtistName
FROM artists ar
JOIN albums al ON ar.ArtistId = al.ArtistId
JOIN tracks t ON al.AlbumId = t.AlbumId
GROUP BY ar.ArtistId
HAVING COUNT(DISTINCT t.GenreId) > 1
SELECT ar.Name as ArtistName
FROM artists ar
JOIN albums al ON ar.ArtistId = al.ArtistId
JOIN tracks t ON al.AlbumId = t.AlbumId
GROUP BY ar.ArtistId
HAVING COUNT(DISTINCT t.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
Info: Ollama parameters:
model=codegemma:latest,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that con
tains the results of the query that answers the question the user asked: '
```

\n   Identify artists who have albums with tracks appearing in multiple ge
nres:\n\n\n'\n\nThe DataFrame was produced using this query: SELECT ar.Name
as ArtistName\nFROM artists ar\nJOIN albums al ON ar.ArtistId = al.ArtistId
\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY ar.ArtistId\nHAVING COUN
T(DISTINCT t.GenreId) > 1\n\nThe following is information about the resultin
g pandas DataFrame 'df': \nRunning df.dtypes gives:\n ArtistName   object\n
dtype: object"}, {"role": "user", "content": "Can you generate the Python pl
otly code to chart the results of the dataframe? Assume the data is in a pan
das dataframe called 'df'. If there is only one value in the dataframe, use
an Indicator. Respond with only Python code. Do not answer with any explanat
ions -- just the code."}]
Info: Ollama Response:
{'model': 'codegemma:latest', 'created_at': '2024-08-01T23:05:06.824293199
Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.ex
press as px\n\nfig = px.bar(df, x='ArtistName', y=df.index.map(len))\nfig.up
date_traces(marker_color='blue')\nfig.update_layout(title='Artists with Albu
ms in Multiple Genres',\n   xaxis_title='Artist Name',\n
yaxis_title='Number of Genres')\n\nif len(df) == 1:\n   fig.add_trace(px.in
dicator(value=df.index[0], title='Genre Count'))\n\nfig.show()\n```"}, 'done
_reason': 'stop', 'done': True, 'total_duration': 31165295722, 'load_duratio
n': 23422868, 'prompt_eval_count': 223, 'prompt_eval_duration': 7354012000,
'eval_count': 124, 'eval_duration': 23742003000}



Out[39]:
```
('SELECT ar.Name as ArtistName\nFROM artists ar\nJOIN albums al ON ar.Artis
tId = al.ArtistId\nJOIN tracks t ON al.AlbumId = t.AlbumId\nGROUP BY ar.Art
istId\nHAVING COUNT(DISTINCT t.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': [{'hovertemplate': 'variable=ArtistName<br>index=%{x}<br>value
=%{y}<extra></extra>',
               'legendgroup': 'ArtistName',
               'line': {'color': '#636efa', 'dash': 'solid'},
               'marker': {'symbol': 'circle'},
               'mode': 'lines',
               'name': 'ArtistName',
               'orientation': 'v',
               'showlegend': True,
               'type': 'scatter',
               'x': array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11,
12, 13, 14, 15, 16, 17,
                           18, 19, 20]),
               'xaxis': 'x',
               'y': array(['Antônio Carlos Jobim', 'Audioslave', 'Various A
rtists', 'Gilberto Gil',
                           'Eric Clapton', 'Faith No More', 'Foo Fighters',
"Guns N' Roses",
                           'Iron Maiden', 'Jamiroquai', 'Lenny Kravitz', 'O
zzy Osbourne',
                           'Pearl Jam', 'R.E.M.', 'Red Hot Chili Peppers',
'Battlestar Galactica',
                           'Heroes', 'Lost', 'U2', 'The Office', 'Amy Wineh
ouse'], dtype=object),
               'yaxis': 'y'}],
     'layout': {'legend': {'title': {'text': 'variable'}, 'tracegroupgap':
0},
                'margin': {'t': 60},
                'template': '...',
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
```

```
ext': 'index'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'value'}}}
 }))
```

Check completion time

In [ ]:
```
from datetime import datetime
```

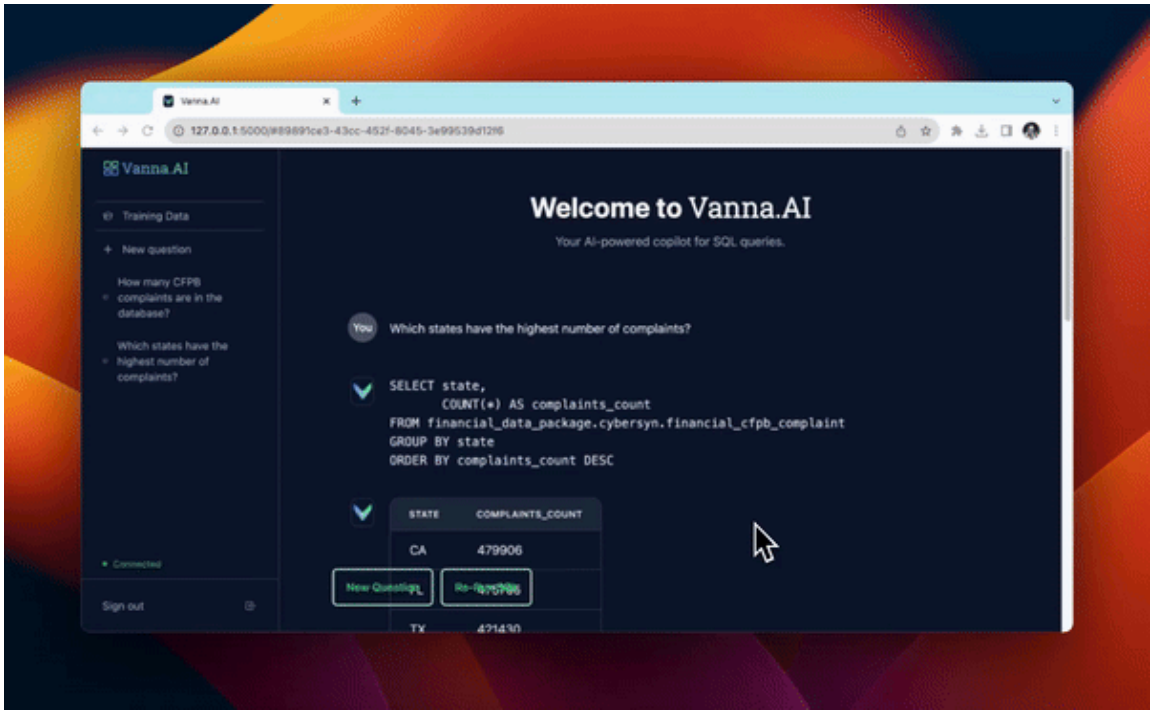
In [ ]:
```
ts_stop = time()

elapsed_time = ts_stop - ts_start
```

In [42]:
```
print(f"[{datetime.now()}] test on '{hostname}' with '{model_name}' LLM took
```

```
[2024-08-03 01:52:21.684227] test on 'ducklover1' with 'codegemma' LLM took
: 1975.42 sec
```

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