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 = "gemma2:2b" # 'llama3'

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]:

```python
# 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()
```

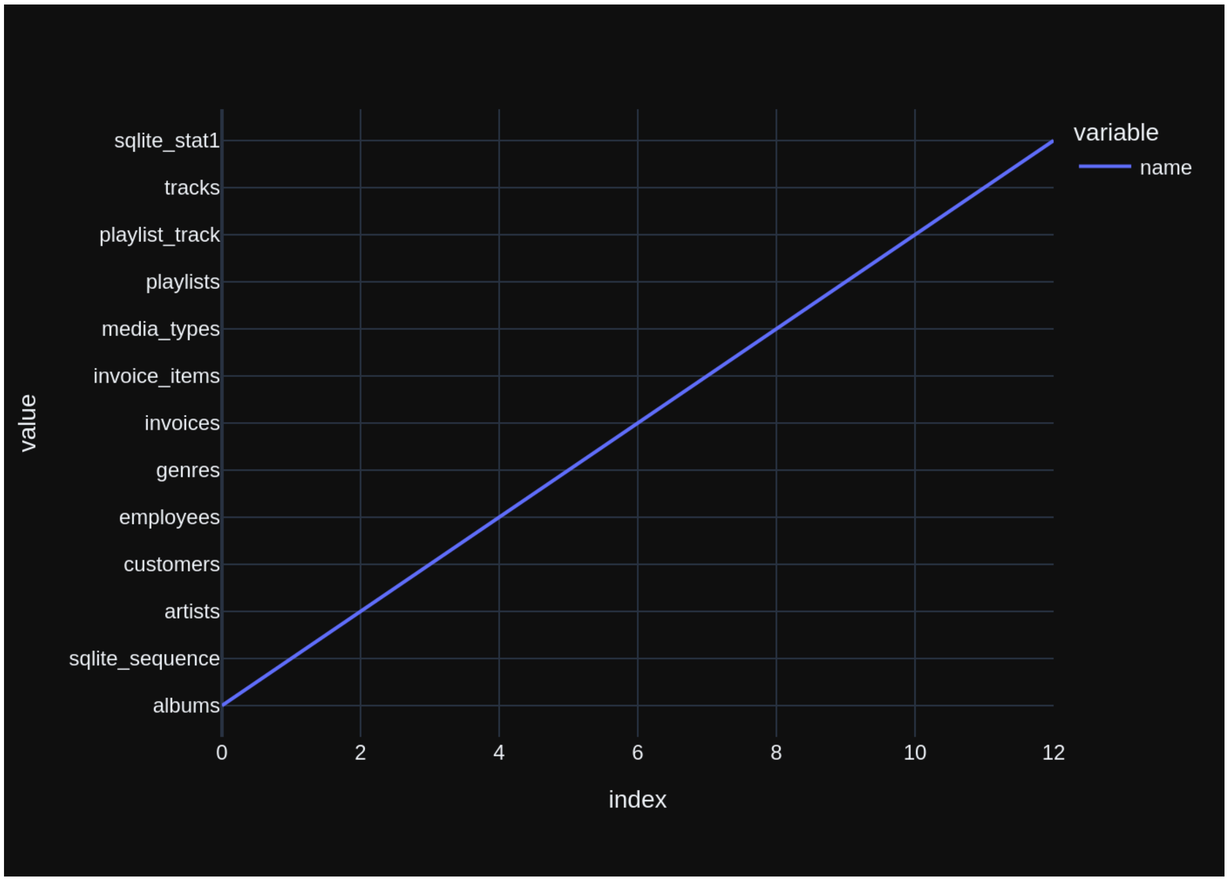
```
SELECT name FROM sqlite_master WHERE type = 'table';
```

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=gemma2:2b,
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': 'gemma2:2b', 'created_at': '2024-08-01T18:22:43.383904996Z', 'mess
age': {'role': 'assistant', 'content': "```sqlite\nSELECT name FROM sqlite_s
chema WHERE type='table'; \n```"}, 'done_reason': 'stop', 'done': True, 'tot
al_duration': 11475609448, 'load_duration': 23387346, 'prompt_eval_count': 8
54, 'prompt_eval_duration': 10023977000, 'eval_count': 18, 'eval_duration': 
1295506000}
LLM Response: ```sqlite
SELECT name FROM sqlite_schema WHERE type='table';
```
Info: Output from LLM: ```sqlite
SELECT name FROM sqlite_schema WHERE type='table';
```
```

```
Extracted SQL: SELECT name FROM sqlite_schema WHERE type='table'
SELECT name FROM sqlite_schema 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=gemma2:2b,
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_schema 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': 'gemma2:2b', 'created_at': '2024-08-01T18:22:49.249310442Z', 'mess
age': {'role': 'assistant', 'content': "```python\nimport plotly.graph_objec
ts as go\n\nfig = go.Figure(data=go.Indicator(value=df['name'].values[0], \n
title='Table Names', \n                              mode='x'))\n\nfig.show
()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 583742208
8, 'load_duration': 21379162, 'prompt_eval_count': 154, 'prompt_eval_duratio
n': 1573784000, 'eval_count': 57, 'eval_duration': 4108063000}
```


variable
name
value
sqlite_stat1
tracks
playlist_track
playlists
media_types
invoice_items
invoices
genres
employees
customers
artists
sqlite_sequence
albums
0
2
4
6
8
10
12
index

Out[16]:
```
("SELECT name FROM sqlite_schema 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': [{'hovertemplate': 'variable=name<br>index=%{x}<br>value=%{y}<
extra></extra>',
               'legendgroup': 'name',
               'line': {'color': '#636efa', 'dash': 'solid'},
               'marker': {'symbol': 'circle'},
               'mode': 'lines',
               'name': 'name',
               'orientation': 'v',
               'showlegend': True,
               'type': 'scatter',
               'x': array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11,
12]),
               'xaxis': 'x',
               'y': array(['albums', 'sqlite_sequence', 'artists', 'custome
rs', 'employees',
                           'genres', 'invoices', 'invoice_items', 'media_ty
pes', 'playlists',
                           'playlist_track', 'tracks', 'sqlite_stat1'], dty
pe=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'}}}
 }))
```

In [17]:
```python
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 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 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 "employee
s" (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 AUTOINCREMEN
T NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHA
R(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n
BirthDate DATETIME,\r\n    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 sqlite_sequence(name,seq)\n\nCREATE TABLE "playlist
s"\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 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 "media_types"\r\n(\r\n    MediaTypeId INTEGER PRIMARY KEY AUTOINCR
EMENT 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 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': 'Can you list all tables
```

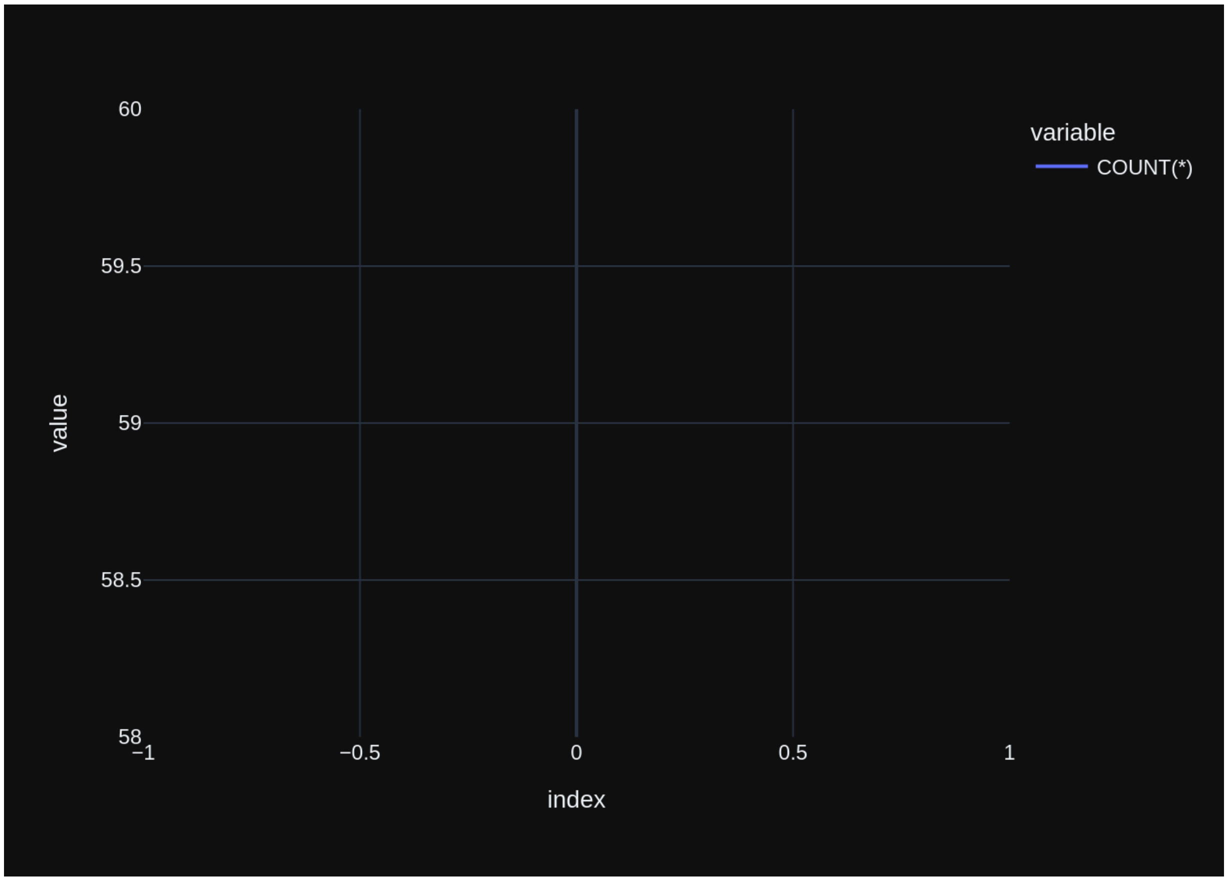
```
in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT
name FROM sqlite_schema WHERE type='table'"}, {'role': 'user', 'content': "w
hich table stores customer's orders"}]
Info: Ollama parameters:
model=gemma2:2b,
options={},
keep_alive=None
Info: Prompt Content:
[{"role": "system", "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_schema WHERE type='table'"}, {"role": "u
ser", "content": "which table stores customer's orders"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:23:03.990737143Z', 'mess
age': {'role': 'assistant', 'content': "The **invoices** table stores custom
er's orders. \n"}, 'done_reason': 'stop', 'done': True, 'total_duration': 14
308798141, 'load_duration': 16850897, 'prompt_eval_count': 1109, 'prompt_eva
l_duration': 12996242000, 'eval_count': 14, 'eval_duration': 1046562000}
LLM Response: The **invoices** table stores customer's orders.

The **invoices** table stores customer's orders.

Couldn't run sql:  Execution failed on sql 'The **invoices** table stores cu
stomer's orders.
': 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_schema WHERE type='tabl
e'"}, {'role': 'user', 'content': 'How many customers are there'}]
Info: Ollama parameters:
model=gemma2:2b,
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_schema
WHERE type='table'"}, {"role": "user", "content": "How many customers are th
ere"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:23:15.101882324Z', 'mess
age': {'role': 'assistant', 'content': '```sql\nSELECT COUNT(*) FROM custome
rs;\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 110691949
68, 'load_duration': 23388609, 'prompt_eval_count': 1018, 'prompt_eval_durat
ion': 9981314000, 'eval_count': 12, 'eval_duration': 856617000}
LLM Response: ```sql
SELECT COUNT(*) FROM customers;
```
Info: Output from LLM: ```sql
SELECT COUNT(*) FROM customers;
```
Extracted SQL: SELECT COUNT(*) FROM customers
SELECT COUNT(*) FROM customers
   COUNT(*)
0        59
Info: Ollama parameters:
model=gemma2:2b,
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': 'gemma2:2b', 'created_at': '2024-08-01T18:23:19.784482674Z', 'mess
age': {'role': 'assistant', 'content': "```python\nimport plotly.graph_objec
ts as go\n\nfig = go.Figure(data=go.Indicator(value=df['COUNT(*)'],  mode='b
ar'))\nfig.show() \n```"}, 'done_reason': 'stop', 'done': True, 'total_durat
ion': 4661494977, 'load_duration': 18249634, 'prompt_eval_count': 146, 'prom
pt_eval_duration': 1463914000, 'eval_count': 45, 'eval_duration': 313266000
0}
```



```
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?")
```

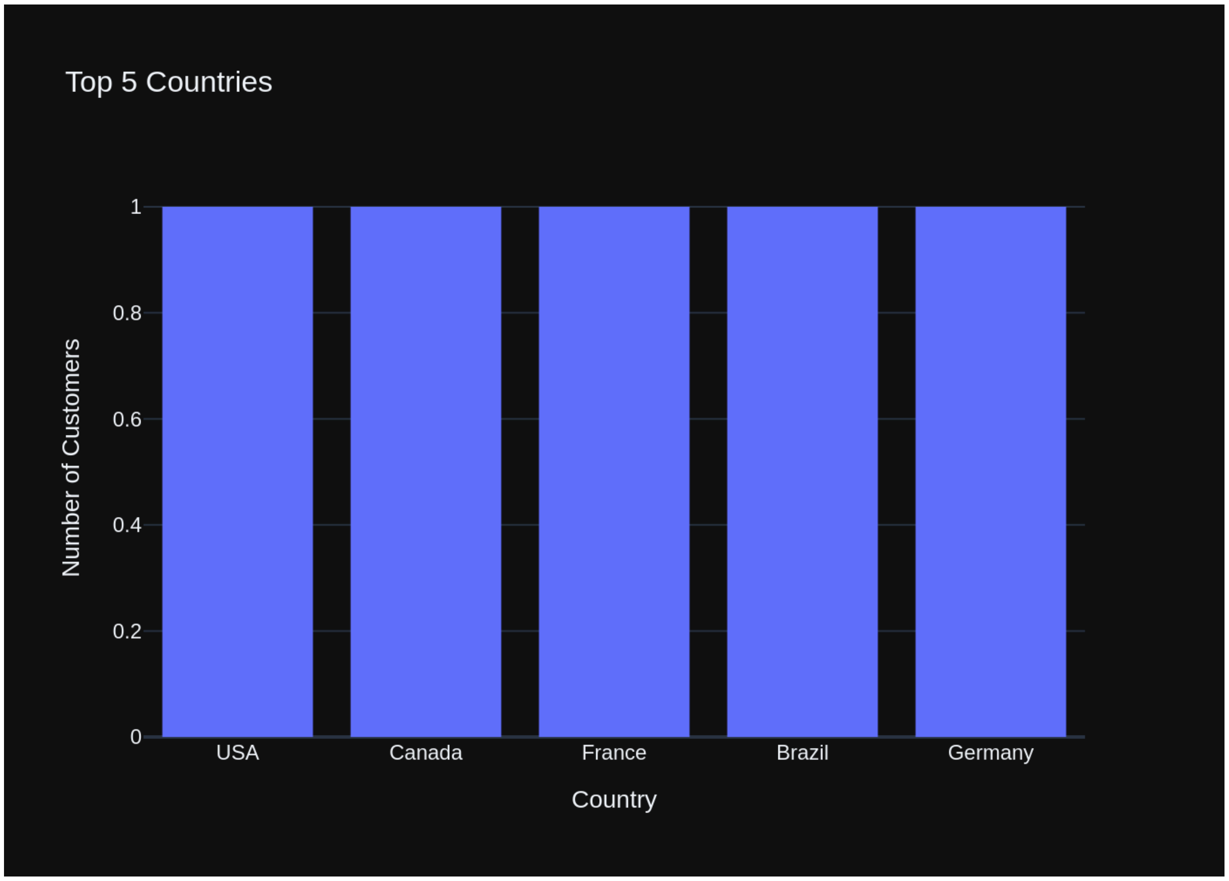
```
Number of requested results 10 is greater than number of elements in index
2, updating n_results = 2
Number of requested results 10 is greater than number of elements in index
1, updating n_results = 1
```

```
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_schema WHERE type
='table'"}, {'role': 'user', 'content': 'what are the top 5 countries that c
ustomers come from?'}]
Info: Ollama parameters:
model=gemma2:2b,
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_schema WHERE type='table'"}, {"role": "user", "content": "what are th
e top 5 countries that customers come from?"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:23:38.83311222Z', 'messa
ge': {'role': 'assistant', 'content': '```sql\nSELECT Country, COUNT(DISTINC
T CustomerId) AS NumCustomers\r\nFROM customers\r\nGROUP BY Country\r\nORDER
BY NumCustomers DESC\r\nLIMIT 5; \r\n``` \r\n'}, 'done_reason': 'stop', 'don
e': True, 'total_duration': 18949283118, 'load_duration': 18617255, 'prompt_
eval_count': 1288, 'prompt_eval_duration': 15233501000, 'eval_count': 44, 'e
val_duration': 3387772000}
LLM Response: ```sql
SELECT Country, COUNT(DISTINCT CustomerId) AS NumCustomers
FROM customers
GROUP BY Country
ORDER BY NumCustomers DESC
LIMIT 5;
```

Info: Output from LLM: ```sql
SELECT Country, COUNT(DISTINCT CustomerId) AS NumCustomers
FROM customers
GROUP BY Country
ORDER BY NumCustomers DESC
LIMIT 5;
```

Extracted SQL: SELECT Country, COUNT(DISTINCT CustomerId) AS NumCustomers
```

~~~
FROM customers
GROUP BY Country
ORDER BY NumCustomers DESC
LIMIT 5
SELECT Country, COUNT(DISTINCT CustomerId) AS NumCustomers
FROM customers
GROUP BY Country
ORDER BY NumCustomers DESC
LIMIT 5
   Country  NumCustomers
0      USA            13
1   Canada             8
2   France             5
3   Brazil             5
4  Germany             4
Info: Ollama parameters:
model=gemma2:2b,
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(DISTINCT CustomerId) AS NumC
ustomers\r\nFROM customers\r\nGROUP BY Country\r\nORDER BY NumCustomers DESC
\r\nLIMIT 5\n\nThe following is information about the resulting pandas DataF
rame 'df': \nRunning df.dtypes gives:\n Country         object\nNumCustomers
int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Py
thon plotly code to chart the results of the dataframe? Assume the data is i
n a pandas dataframe called 'df'. If there is only one value in the datafram
e, use an Indicator. Respond with only Python code. Do not answer with any e
xplanations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:23:44.76356017Z', 'messa
ge': {'role': 'assistant', 'content': '```python\nimport plotly.express as p
x\n\nfig = px.histogram(df, x="Country", title="Top 5 Countries") \n\nfig.up
date_layout(xaxis_title=\'Country\', yaxis_title=\'Number of Customers\') \n
```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 5903377303, 'l
oad_duration': 18289037, 'prompt_eval_count': 185, 'prompt_eval_duration': 1
824454000, 'eval_count': 56, 'eval_duration': 3969450000}
~~~


Top 5 Countries
Number of Customers
1
0.8
0.6
0.4
0.2
0
USA
Canada
France
Brazil
Germany
Country

Out[19]:
```
('SELECT Country, COUNT(DISTINCT CustomerId) AS NumCustomers\r\nFROM custom
ers\r\nGROUP BY Country\r\nORDER BY NumCustomers DESC\r\nLIMIT 5',
    Country  NumCustomers
 0      USA            13
 1   Canada             8
 2   France             5
 3   Brazil             5
 4  Germany             4,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'bingroup': 'x',
               'hovertemplate': 'Country=%{x}<br>count=%{y}<extra></extra
>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'type': 'histogram',
               'x': array(['USA', 'Canada', 'France', 'Brazil', 'Germany'],
dtype=object),
               'xaxis': 'x',
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Top 5 Countries'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Country'}},
                'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'Number of Customers'}}}
 }))
```

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
```

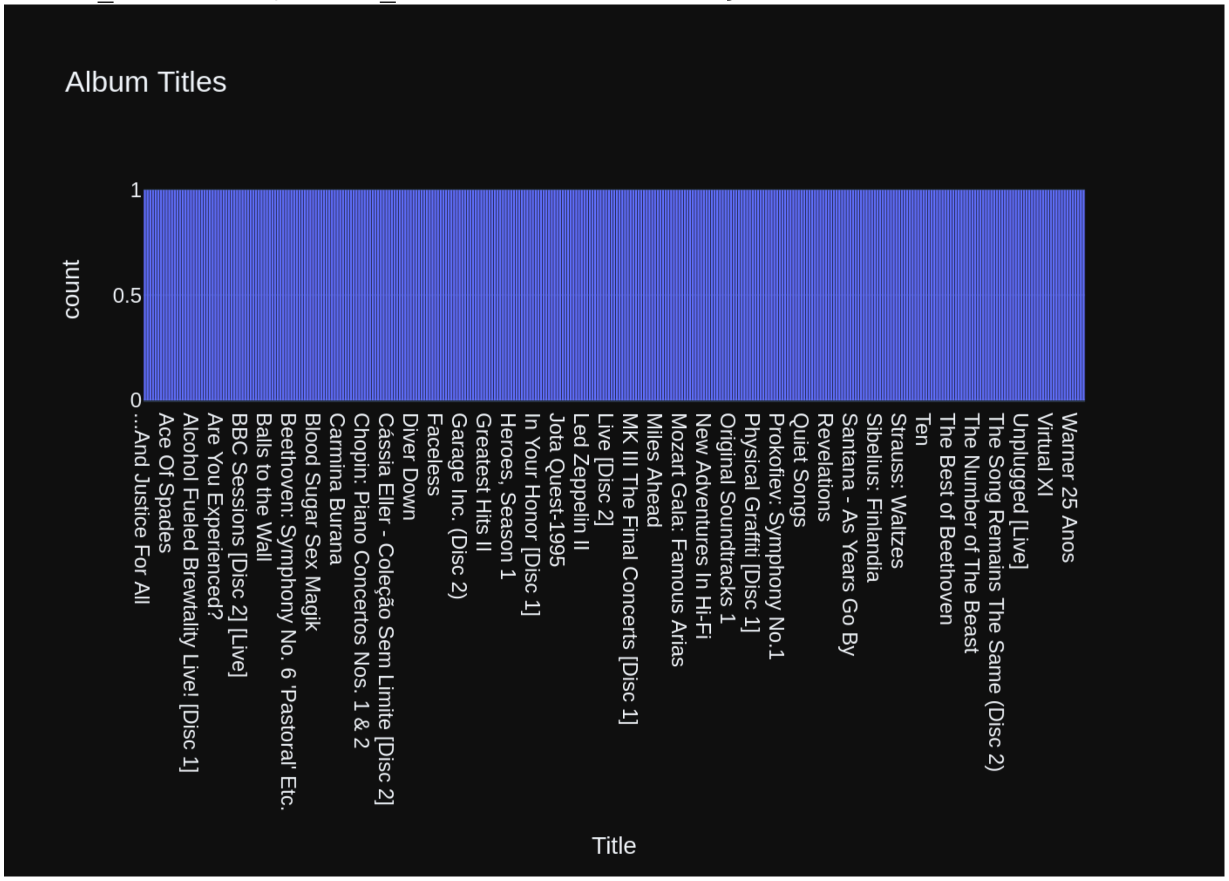
```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please
help to generate a SQL query to answer the question. Your response should ON
LY be based on the given context and follow the response guidelines and form
at instructions. \n===Tables \nCREATE INDEX IFK_AlbumArtistId ON "albums" (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_schema WHERE type='table'"}, {'role': 'use
r', 'content': 'what are the top 5 countries that customers come from?'},
{'role': 'assistant', 'content': 'SELECT Country, COUNT(DISTINCT CustomerId)
AS NumCustomers\r\nFROM customers\r\nGROUP BY Country\r\nORDER BY NumCustome
rs DESC\r\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are th
ere'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'},
{'role': 'user', 'content': '  \n    List all albums and their corresponding
artist names  \n'}]
Info: Ollama parameters:
model=gemma2:2b,
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_schema WHERE type='tabl
e'"}, {"role": "user", "content": "what are the top 5 countries that custome
rs come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(DIS
TINCT CustomerId) AS NumCustomers\r\nFROM customers\r\nGROUP BY Country\r\nO
RDER BY NumCustomers DESC\r\nLIMIT 5"}, {"role": "user", "content": "How man
y customers are there"}, {"role": "assistant", "content": "SELECT COUNT(*) F
ROM customers"}, {"role": "user", "content": "  \n    List all albums and th
eir corresponding artist names  \n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:23:58.21955531Z', 'messa
ge': {'role': 'assistant', 'content': '```sql\nSELECT "albums".Title, "artis
ts".Name \r\nFROM "albums"\r\nJOIN "artists" ON "albums"."ArtistId" = "artis
ts"."ArtistId"\r\nORDER BY "albums"."Title"; \r\n```'}, 'done_reason': 'sto
p', 'done': True, 'total_duration': 13340220937, 'load_duration': 16800530,
'prompt_eval_count': 767, 'prompt_eval_duration': 9129591000, 'eval_count':
50, 'eval_duration': 3738241000}
LLM Response: ```sql
SELECT "albums".Title, "artists".Name
FROM "albums"
JOIN "artists" ON "albums"."ArtistId" = "artists"."ArtistId"
ORDER BY "albums"."Title";
```
Info: Output from LLM: ```sql
SELECT "albums".Title, "artists".Name
FROM "albums"
JOIN "artists" ON "albums"."ArtistId" = "artists"."ArtistId"
ORDER BY "albums"."Title";
```
```

```
```
Extracted SQL: SELECT "albums".Title, "artists".Name
FROM "albums"
JOIN "artists" ON "albums"."ArtistId" = "artists"."ArtistId"
ORDER BY "albums"."Title"
SELECT "albums".Title, "artists".Name
FROM "albums"
JOIN "artists" ON "albums"."ArtistId" = "artists"."ArtistId"
ORDER BY "albums"."Title"
                                                  Title \
0   ...And Justice For All
1   20th Century Masters - The Millennium Collecti...
2   A Copland Celebration, Vol. I
3   A Matter of Life and Death
4   A Real Dead One
..   ...
342   Warner 25 Anos
343   Weill: The Seven Deadly Sins
344   Worlds
345   Zooropa
346   [1997] Black Light Syndrome

                                             Name
0   Metallica
1   Scorpions
2   Aaron Copland & London Symphony Orchestra
3   Iron Maiden
4   Iron Maiden
..   ...
342   Antônio Carlos Jobim
343 Kent Nagano and Orchestre de l'Opéra de Lyon
344   Aaron Goldberg
345   U2
346   Terry Bozzio, Tony Levin & Steve Stevens

[347 rows x 2 columns]
Info: Ollama parameters:
model=gemma2:2b,
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 \"albums\".Title, \"artists\".Nam
e \r\nFROM \"albums\"\r\nJOIN \"artists\" ON \"albums\".\"ArtistId\" = \"art
ists\".\"ArtistId\"\r\nORDER BY \"albums\".\"Title\"\n\nThe following is inf
ormation about the resulting pandas DataFrame 'df': \nRunning df.dtypes give
s:\n Title   object\nName   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': 'gemma2:2b', 'created_at': '2024-08-01T18:24:02.746500812Z', 'mess
age': {'role': 'assistant', 'content': '```python\nimport plotly.express as
```
```

```
px\n\nfig = px.histogram(df, x="Title", title="Album Titles") \n```'}, 'done
_reason': 'stop', 'done': True, 'total_duration': 4498003631, 'load_duratio
n': 20695408, 'prompt_eval_count': 193, 'prompt_eval_duration': 2197804000,
'eval_count': 31, 'eval_duration': 2144514000}
```



```
Out[20]: ('SELECT "albums".Title, "artists".Name \r\nFROM "albums"\r\nJOIN "artists"
         ON "albums"."ArtistId" = "artists"."ArtistId"\r\nORDER BY "albums"."Titl
         e"',
                                                             Title  \
          0                                ...And Justice For All
          1    20th Century Masters - The Millennium Collecti...
          2                          A Copland Celebration, Vol. I
          3                             A Matter of Life and Death
          4                                        A Real Dead One
          ..                                                   ...
          342                                      Warner 25 Anos
          343                          Weill: The Seven Deadly Sins
          344                                              Worlds
          345                                             Zooropa
          346                        [1997] Black Light Syndrome

                                                        Name
          0                                        Metallica
          1                                        Scorpions
          2       Aaron Copland & London Symphony Orchestra
          3                                      Iron Maiden
          4                                      Iron Maiden
          ..                                             ...
          342                            Antônio Carlos Jobim
          343  Kent Nagano and Orchestre de l'Opéra de Lyon
          344                                 Aaron Goldberg
          345                                             U2
          346      Terry Bozzio, Tony Levin & Steve Stevens

          [347 rows x 2 columns],
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'bingroup': 'x',
                        'hovertemplate': 'Title=%{x}<br>count=%{y}<extra></extra>',
                        'legendgroup': '',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'type': 'histogram',
                        'x': array(['...And Justice For All',
                                    '20th Century Masters - The Millennium Collectio
         n: The Best of Scorpions',
                                    'A Copland Celebration, Vol. I', ..., 'Worlds',
         'Zooropa',
                                    '[1997] Black Light Syndrome'], dtype=object),
                        'xaxis': 'x',
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                         'legend': {'tracegroupgap': 0},
                         'template': '...',
                         'title': {'text': 'Album Titles'},
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
         ext': 'Title'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
         ext': 'count'}}}
          }))
```

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

vn.ask(question=question)
```

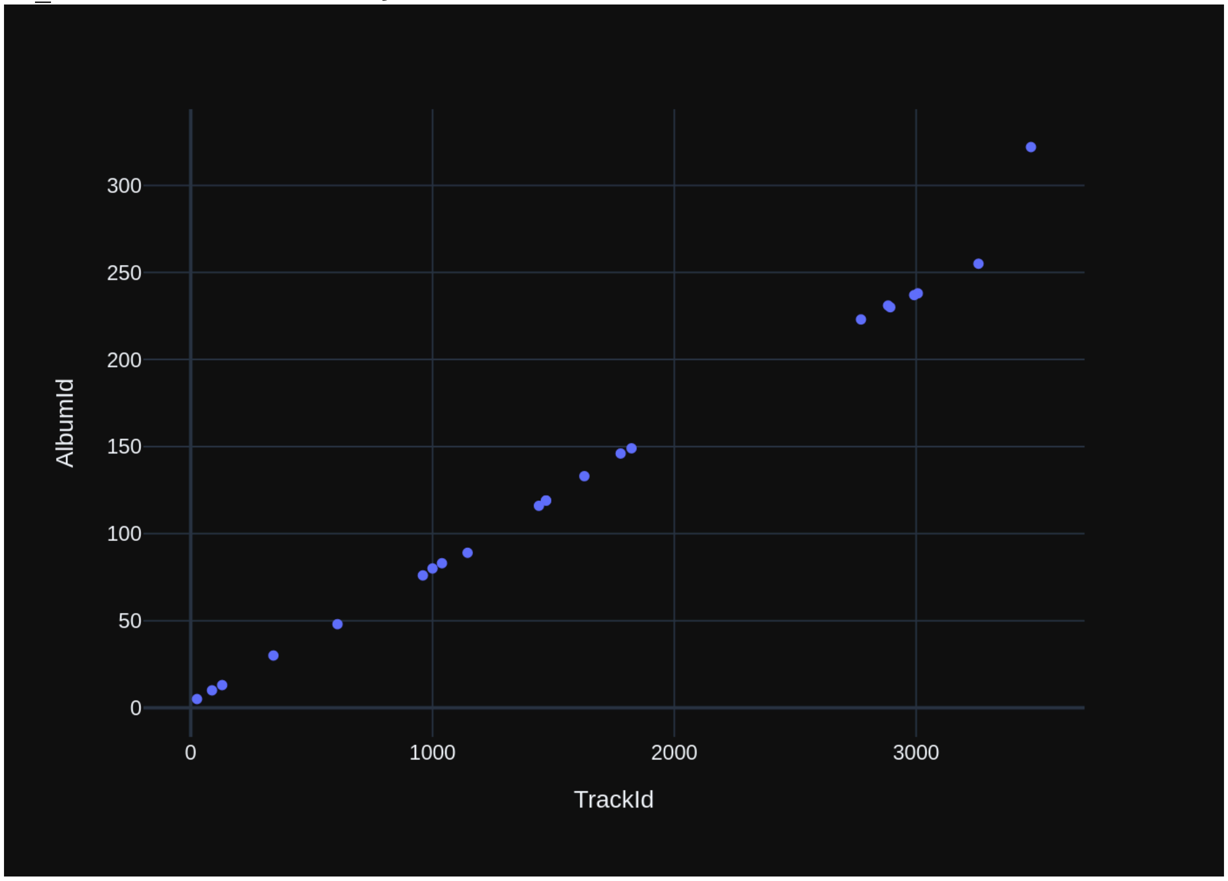
```
Number of requested results 10 is greater than number of elements in index
4, updating n_results = 4
Number of requested results 10 is greater than number of elements in index
1, updating n_results = 1
```

```
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 "albums".Title, "artists".Name \r\n
FROM "albums"\r\nJOIN "artists" ON "albums"."ArtistId" = "artists"."ArtistI
d"\r\nORDER BY "albums"."Title"'}, {'role': 'user', 'content': 'Can you list
all tables in the SQLite database catalog?'}, {'role': 'assistant', 'conten
t': "SELECT name FROM sqlite_schema WHERE type='table'"}, {'role': 'user',
'content': 'what are the top 5 countries that customers come from?'}, {'rol
e': 'assistant', 'content': 'SELECT Country, COUNT(DISTINCT CustomerId) AS N
umCustomers\r\nFROM customers\r\nGROUP BY Country\r\nORDER BY NumCustomers D
ESC\r\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are ther
e'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'r
ole': 'user', 'content': '  \n    Find all tracks with a name containing "Wh
at" (case-insensitive)\n'}]
Info: Ollama parameters:
model=gemma2:2b,
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 \"albums\".Title, \"artists
\".Name \r\nFROM \"albums\"\r\nJOIN \"artists\" ON \"albums\".\"ArtistId\" =
\"artists\".\"ArtistId\"\r\nORDER BY \"albums\".\"Title\""}, {"role": "use
r", "content": "Can you list all tables in the SQLite database catalog?"},
{"role": "assistant", "content": "SELECT name FROM sqlite_schema WHERE type
='table'"}, {"role": "user", "content": "what are the top 5 countries that c
ustomers come from?"}, {"role": "assistant", "content": "SELECT Country, COU
NT(DISTINCT CustomerId) AS NumCustomers\r\nFROM customers\r\nGROUP BY Countr
y\r\nORDER BY NumCustomers DESC\r\nLIMIT 5"}, {"role": "user", "content": "H
ow many customers are there"}, {"role": "assistant", "content": "SELECT COUN
T(*) FROM customers"}, {"role": "user", "content": "  \n    Find all tracks
with a name containing \"What\" (case-insensitive)\n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:24:15.614039646Z', 'mess
age': {'role': 'assistant', 'content': "SELECT * \r\nFROM tracks\r\nWHERE Na
me LIKE '%what%'\r\nORDER BY Name; \r\n"}, 'done_reason': 'stop', 'done': Tr
ue, 'total_duration': 12706685654, 'load_duration': 17739661, 'prompt_eval_c
ount': 858, 'prompt_eval_duration': 10376714000, 'eval_count': 23, 'eval_dur
ation': 1707179000}
LLM Response: SELECT *
FROM tracks
```

```
WHERE Name LIKE '%what%'
ORDER BY Name;

Info: Output from LLM: SELECT *
FROM tracks
WHERE Name LIKE '%what%'
ORDER BY Name;

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

    MediaTypeId  GenreId                                         Composer  
\
0             1        2                                      George Duke   
1             1        4                      C. Cester/C. Muncey/N. Cester   
2             1        7                                             None   
3             1        1                                               U2   
4             1        1       Bono/Clayton, Adam/Mullen Jr., Larry/The Edge   
5             1        4                                        N. Cester   
6             1        2                                      Miles Davis   
7             1        3                                     Culmer/Exalt   
8             1        1              Mike Bordin, Billy Gould, Mike Patton   
9             1        1  Dave Grohl, Taylor Hawkins, Nate Mendel, Chris...   
10            1        1                             Jimmy Page, Robert Plant   
11            2        9  Delroy "Chris" Cooper, Donovan Jackson, Earl C...   
12            1        1              Steven Tyler, Joe Perry, Desmond Child   
13            3       19                                             None   
```

```
14            1       12          carl sigman/gilbert becaud/pierre leroyer
15            1        1                            Audioslave/Chris Cornell
16            1        1                             Jimmy Page/Robert Plant
17            2        9                                                None
18            1        1                                   Jay Kay/Kay, Jay
19            3       19                                                None
20            1        4                                           Green Day
21            1       14             Allen Story/George Gordy/Robert Gordy

    Milliseconds      Bytes  UnitPrice
0         274155    9018565       0.99
1         247719    8043765       0.99
2         221387    7251478       0.99
3         280764    9306737       0.99
4         353567   11542247       0.99
5         230974    7517083       0.99
6         564009   18360449       0.99
7         189152    6162894       0.99
8         158275    5203430       0.99
9         302994    9929799       0.99
10        287973    9369385       0.99
11        209573    3426106       0.99
12        310622   10144730       0.99
13       2610250  484583988       1.99
14        149995    4913383       0.99
15        249391    5988186       0.99
16        260675    8497116       0.99
17        215084    3499018       0.99
18        247222    8249453       0.99
19       2616410  183867185       1.99
20        252316    8244843       0.99
21        142027    4631104       0.99
Info: Ollama parameters:
model=gemma2:2b,
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 * \r\nFROM tracks
\r\nWHERE Name LIKE '%what%'\r\nORDER BY Name\n\nThe following is informatio
n about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Tr
ackId           int64\nName             object\nAlbumId           int64\nMed
iaTypeId       int64\nGenreId           int64\nComposer         object\nMill
iseconds      int64\nBytes             int64\nUnitPrice       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': 'gemma2:2b', 'created_at': '2024-08-01T18:24:24.497377527Z', 'mess
age': {'role': 'assistant', 'content': "```python\nimport plotly.graph_objec
ts as go\n\nfig = go.Figure(data=[go.indicator(mode='value', \n
text=df['Name'].to_list()[0],\n                                     title='Num
```

```
ber of Tracks'), \n                    ])\nfig.update_layout(xaxis_showgrid=
False, yaxis_range=[0, 10])\nfig.show()\n\n```"}, 'done_reason': 'stop', 'do
ne': True, 'total_duration': 8856814229, 'load_duration': 22489891, 'prompt_
eval_count': 221, 'prompt_eval_duration': 2193679000, 'eval_count': 90, 'eva
l_duration': 6547853000}
```



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

              MediaTypeId  GenreId                                          Compose
         r  \
          0            1        2                                         George Duk
         e   
          1            1        4                           C. Cester/C. Muncey/N. Ceste
         r   
          2            1        7                                                Non
         e   
          3            1        1                                                  U
         2   
          4            1        1      Bono/Clayton, Adam/Mullen Jr., Larry/The Edg
         e   
          5            1        4                                           N. Ceste
         r   
          6            1        2                                         Miles Davi
         s   
          7            1        3                                         Culmer/Exal
         t   
          8            1        1                 Mike Bordin, Billy Gould, Mike Patto
         n   
          9            1        1  Dave Grohl, Taylor Hawkins, Nate Mendel, Chri
         s...   
          10           1        1                             Jimmy Page, Robert Plan
         t   
          11           2        9  Delroy "Chris" Cooper, Donovan Jackson, Earl 
         C...   
          12           1        1                   Steven Tyler, Joe Perry, Desmond Chil
         d   
          13           3       19                                                Non
         e   
          14           1       12           carl sigman/gilbert becaud/pierre leroye
```

```
r
 15           1       1                              Audioslave/Chris Cornel
l
 16           1       1                             Jimmy Page/Robert Plan
t
 17           2       9                                                  Non
e
 18           1       1                                      Jay Kay/Kay, Ja
y
 19           3      19                                                  Non
e
 20           1       4                                             Green Da
y
 21           1      14             Allen Story/George Gordy/Robert Gord
y

     Milliseconds      Bytes  UnitPrice
 0         274155    9018565       0.99
 1         247719    8043765       0.99
 2         221387    7251478       0.99
 3         280764    9306737       0.99
 4         353567   11542247       0.99
 5         230974    7517083       0.99
 6         564009   18360449       0.99
 7         189152    6162894       0.99
 8         158275    5203430       0.99
 9         302994    9929799       0.99
 10        287973    9369385       0.99
 11        209573    3426106       0.99
 12        310622   10144730       0.99
 13       2610250  484583988       1.99
 14        149995    4913383       0.99
 15        249391    5988186       0.99
 16        260675    8497116       0.99
 17        215084    3499018       0.99
 18        247222    8249453       0.99
 19       2616410  183867185       1.99
 20        252316    8244843       0.99
 21        142027    4631104       0.99  ,
 Figure({
     'data': [{'hovertemplate': 'TrackId=%{x}<br>AlbumId=%{y}<extra></extra
>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'symbol': 'circle'},
               'mode': 'markers',
               'name': '',
               'orientation': 'v',
               'showlegend': False,
               'type': 'scatter',
               'x': array([ 130, 1470, 2772, 3007, 2992, 1469,  607, 1823,
960, 1000, 1628, 3475,
                             26, 2884, 1039,   88,  342, 3258, 1440, 2893,
1145, 1778]),
               'xaxis': 'x',
               'y': array([ 13, 119, 223, 238, 237, 119,  48, 149,  76,  8
0, 133, 322,   5, 231,
                                83,  10,  30, 255, 116, 230,  89, 146]),
                 'yaxis': 'y'}],
      'layout': {'legend': {'tracegroupgap': 0},
                 'margin': {'t': 60},
                 'template': '...',
                 'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'TrackId'}},
                 'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'AlbumId'}}}
 }))
```

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
```

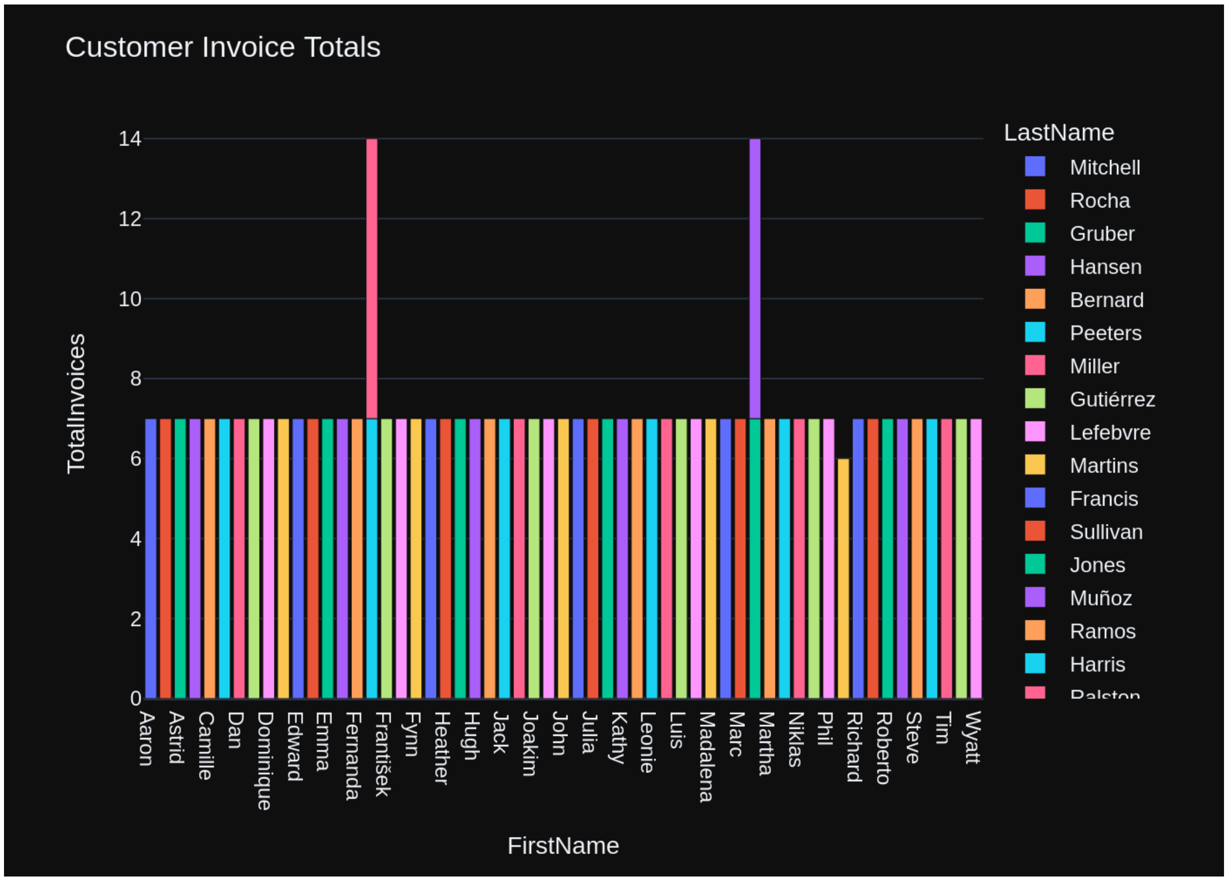
```
ers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM cust
omers'}, {'role': 'user', 'content': 'what are the top 5 countries that cust
omers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT
(DISTINCT CustomerId) AS NumCustomers\r\nFROM customers\r\nGROUP BY Country
\r\nORDER BY NumCustomers DESC\r\nLIMIT 5'}, {'role': 'user', 'content': '
\n    List all albums and their corresponding artist names  \n'}, {'role':
'assistant', 'content': 'SELECT "albums".Title, "artists".Name \r\nFROM "alb
ums"\r\nJOIN "artists" ON "albums"."ArtistId" = "artists"."ArtistId"\r\nORDE
R BY "albums"."Title"'}, {'role': 'user', 'content': '  \n    Find all track
s with a name containing "What" (case-insensitive)\n'}, {'role': 'assistan
t', 'content': "SELECT * \r\nFROM tracks\r\nWHERE Name LIKE '%what%'\r\nORDE
R BY Name"}, {'role': 'user', 'content': 'Can you list all tables in the SQL
ite database catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM
sqlite_schema WHERE type='table'"}, {'role': 'user', 'content': '  \n    Get
the total number of invoices for each customer\n'}]
Info: Ollama parameters:
model=gemma2:2b,
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(DISTINCT CustomerId) AS NumCustomers\r\nFROM customer
s\r\nGROUP BY Country\r\nORDER BY NumCustomers DESC\r\nLIMIT 5"}, {"role":
"user", "content": "  \n    List all albums and their corresponding artist n
ames  \n"}, {"role": "assistant", "content": "SELECT \"albums\".Title, \"art
ists\".Name \r\nFROM \"albums\"\r\nJOIN \"artists\" ON \"albums\".\"ArtistId
\" = \"artists\".\"ArtistId\"\r\nORDER BY \"albums\".\"Title\""}, {"role":
"user", "content": "  \n    Find all tracks with a name containing \"What\"
(case-insensitive)\n"}, {"role": "assistant", "content": "SELECT * \r\nFROM
tracks\r\nWHERE Name LIKE '%what%'\r\nORDER BY Name"}, {"role": "user", "con
tent": "Can you list all tables in the SQLite database catalog?"}, {"role":
"assistant", "content": "SELECT name FROM sqlite_schema WHERE type='tabl
e'"}, {"role": "user", "content": "  \n    Get the total number of invoices
for each customer\n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:24:47.366896101Z', 'mess
age': {'role': 'assistant', 'content': '```sql\nSELECT c.FirstName, c.LastNa
me, COUNT(i.InvoiceId) AS TotalInvoices\r\nFROM customers c\r\nJOIN invoices
i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName; \r\n``
` \n\n\nLet me know if you have other queries!  😊  '}, 'done_reason': 'sto
p', 'done': True, 'total_duration': 22748762883, 'load_duration': 24142935,
'prompt_eval_count': 1309, 'prompt_eval_duration': 16506865000, 'eval_coun
t': 70, 'eval_duration': 5485488000}
LLM Response: ```sql
SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName;
```

Let me know if you have other queries! 😊
Info: Output from LLM: ```sql
SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices
```
```

```
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName;
```
```

Let me know if you have other queries! 😊
Extracted SQL: SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalIn
voices
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName
SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName
    FirstName   LastName TotalInvoices
0   Aaron   Mitchell   7
1   Alexandre   Rocha   7
2   Astrid   Gruber   7
3   Bjørn   Hansen   7
4   Camille   Bernard   7
5   Daan   Peeters   7
6   Dan   Miller   7
7   Diego   Gutiérrez   7
8   Dominique   Lefebvre   7
9   Eduardo   Martins   7
10   Edward   Francis   7
11   Ellie   Sullivan   7
12   Emma   Jones   7
13   Enrique   Muñoz   7
14   Fernanda   Ramos   7
15   Frank   Harris   7
16   Frank   Ralston   7
17 František   Wichterlová   7
18   François   Tremblay   7
19   Fynn   Zimmermann   7
20   Hannah   Schneider   7
21   Heather   Leacock   7
22   Helena   Holý   7
23   Hugh   O'Reilly   7
24   Isabelle   Mercier   7
25   Jack   Smith   7
26   Jennifer   Peterson   7
27   Joakim   Johansson   7
28   Johannes Van der Berg   7
29   John   Gordon   7
30   João   Fernandes   7
31   Julia   Barnett   7
32   Kara   Nielsen   7
33   Kathy   Chase   7
34   Ladislav   Kovács   7
35   Leonie   Köhler   7
36   Lucas   Mancini   7
37   Luis   Rojas   7
38   Luís   Gonçalves   7
```

```
39   Madalena       Sampaio              7
40      Manoj        Pareek              7
41       Marc        Dubois              7
42       Mark       Philips              7
43       Mark        Taylor              7
44     Martha          Silk              7
45   Michelle        Brooks              7
46     Niklas      Schröder              7
47    Patrick          Gray              7
48       Phil        Hughes              7
49       Puja    Srivastava              6
50    Richard    Cunningham              7
51     Robert         Brown              7
52    Roberto       Almeida              7
53  Stanisław        Wójcik              7
54      Steve        Murray              7
55      Terhi    Hämäläinen              7
56        Tim         Goyer              7
57     Victor       Stevens              7
58      Wyatt        Girard              7
Info: Ollama parameters:
model=gemma2:2b,
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\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerI
d = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName\n\nThe following is inf
ormation about the resulting pandas DataFrame 'df': \nRunning df.dtypes give
s:\n FirstName        object\nLastName         object\nTotalInvoices     int
64\ndtype: object"}, {"role": "user", "content": "Can you generate the Pytho
n plotly code to chart the results of the dataframe? Assume the data is in a
pandas dataframe called 'df'. If there is only one value in the dataframe, u
se an Indicator. Respond with only Python code. Do not answer with any expla
nations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:24:53.724571342Z', 'mess
age': {'role': 'assistant', 'content': '```python\nimport plotly.express as
px\n\nfig = px.bar(df, x="FirstName", y="TotalInvoices", color="LastName")
\nfig.update_layout(title=\'Customer Invoice Totals\')\nfig.show()\n```'},
'done_reason': 'stop', 'done': True, 'total_duration': 6337218338, 'load_dur
ation': 24210229, 'prompt_eval_count': 205, 'prompt_eval_duration': 24585800
00, 'eval_count': 53, 'eval_duration': 3763513000}
```


Customer Invoice Totals
TotalInvoices
0
2
4
6
8
10
12
14
FirstName
Aaron
Astrid
Camille
Dan
Dominique
Edward
Emma
Fernanda
František
Fynn
Heather
Hugh
Jack
Joakim
John
Julia
Kathy
Leonie
Luis
Madalena
Marc
Martha
Niklas
Phil
Richard
Roberto
Steve
Tim
Wyatt
LastName
Mitchell
Rocha
Gruber
Hansen
Bernard
Peeters
Miller
Gutiérrez
Lefebvre
Martins
Francis
Sullivan
Jones
Muñoz
Ramos
Harris
Ralston

```
Out[22]: ('SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\r\nFR
         OM customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP B
         Y c.FirstName, c.LastName',
              FirstName      LastName  TotalInvoices
          0       Aaron      Mitchell              7
          1   Alexandre         Rocha              7
          2      Astrid        Gruber              7
          3       Bjørn        Hansen              7
          4     Camille       Bernard              7
          5        Daan       Peeters              7
          6         Dan        Miller              7
          7       Diego     Gutiérrez              7
          8   Dominique      Lefebvre              7
          9     Eduardo       Martins              7
          10     Edward       Francis              7
          11      Ellie      Sullivan              7
          12       Emma         Jones              7
          13   Enrique         Muñoz              7
          14   Fernanda         Ramos              7
          15      Frank        Harris              7
          16      Frank       Ralston              7
          17  František   Wichterlová              7
          18   François      Tremblay              7
          19       Fynn    Zimmermann              7
          20     Hannah     Schneider              7
          21    Heather       Leacock              7
          22     Helena          Holý              7
          23       Hugh      O'Reilly              7
          24  Isabelle       Mercier              7
          25       Jack         Smith              7
          26   Jennifer      Peterson              7
          27     Joakim     Johansson              7
          28   Johannes  Van der Berg              7
          29       John        Gordon              7
          30       João     Fernandes              7
          31      Julia       Barnett              7
          32       Kara       Nielsen              7
          33      Kathy         Chase              7
          34   Ladislav        Kovács              7
          35     Leonie        Köhler              7
          36      Lucas       Mancini              7
          37       Luis         Rojas              7
          38       Luís     Gonçalves              7
          39   Madalena       Sampaio              7
          40      Manoj        Pareek              7
          41       Marc        Dubois              7
          42       Mark       Philips              7
          43       Mark        Taylor              7
          44     Martha          Silk              7
          45   Michelle        Brooks              7
          46     Niklas       Schröder              7
          47    Patrick          Gray              7
          48       Phil        Hughes              7
          49       Puja    Srivastava              6
          50    Richard    Cunningham              7
          51     Robert         Brown              7
```

```
 52    Roberto       Almeida              7
 53  Stanisław        Wójcik              7
 54      Steve        Murray              7
 55      Terhi    Hämäläinen              7
 56        Tim         Goyer              7
 57     Victor       Stevens              7
 58      Wyatt        Girard              7,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Mitchell<br>FirstName=%{x}<br>Tot
alInvoices=%{y}<extra></extra>',
               'legendgroup': 'Mitchell',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': 'Mitchell',
               'offsetgroup': 'Mitchell',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Aaron'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Rocha<br>FirstName=%{x}<br>TotalI
nvoices=%{y}<extra></extra>',
               'legendgroup': 'Rocha',
               'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
               'name': 'Rocha',
               'offsetgroup': 'Rocha',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Alexandre'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Gruber<br>FirstName=%{x}<br>Total
Invoices=%{y}<extra></extra>',
               'legendgroup': 'Gruber',
               'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
               'name': 'Gruber',
               'offsetgroup': 'Gruber',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Astrid'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Hansen<br>FirstName=%{x}<br>Total
Invoices=%{y}<extra></extra>',
               'legendgroup': 'Hansen',
               'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
               'name': 'Hansen',
               'offsetgroup': 'Hansen',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Bjørn'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Bernard<br>FirstName=%{x}<br>Tota
lInvoices=%{y}<extra></extra>',
               'legendgroup': 'Bernard',
               'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
               'name': 'Bernard',
               'offsetgroup': 'Bernard',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Camille'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Peeters<br>FirstName=%{x}<br>Tota
lInvoices=%{y}<extra></extra>',
               'legendgroup': 'Peeters',
               'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
               'name': 'Peeters',
               'offsetgroup': 'Peeters',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Daan'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Miller<br>FirstName=%{x}<br>Total
Invoices=%{y}<extra></extra>',
               'legendgroup': 'Miller',
               'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
               'name': 'Miller',
               'offsetgroup': 'Miller',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Dan'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Gutiérrez<br>FirstName=%{x}<br>To
talInvoices=%{y}<extra></extra>',
               'legendgroup': 'Gutiérrez',
               'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
               'name': 'Gutiérrez',
               'offsetgroup': 'Gutiérrez',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Diego'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Lefebvre<br>FirstName=%{x}<br>Tot
alInvoices=%{y}<extra></extra>',
               'legendgroup': 'Lefebvre',
               'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
               'name': 'Lefebvre',
               'offsetgroup': 'Lefebvre',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Dominique'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Martins<br>FirstName=%{x}<br>Tota
lInvoices=%{y}<extra></extra>',
               'legendgroup': 'Martins',
               'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
               'name': 'Martins',
               'offsetgroup': 'Martins',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Eduardo'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Francis<br>FirstName=%{x}<br>Tota
lInvoices=%{y}<extra></extra>',
               'legendgroup': 'Francis',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': 'Francis',
               'offsetgroup': 'Francis',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Edward'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Sullivan<br>FirstName=%{x}<br>Tot
alInvoices=%{y}<extra></extra>',
               'legendgroup': 'Sullivan',
               'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
               'name': 'Sullivan',
               'offsetgroup': 'Sullivan',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Ellie'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Jones<br>FirstName=%{x}<br>TotalI
nvoices=%{y}<extra></extra>',
               'legendgroup': 'Jones',
               'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
               'name': 'Jones',
               'offsetgroup': 'Jones',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Emma'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Muñoz<br>FirstName=%{x}<br>TotalI
nvoices=%{y}<extra></extra>',
               'legendgroup': 'Muñoz',
               'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
               'name': 'Muñoz',
               'offsetgroup': 'Muñoz',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Enrique'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Ramos<br>FirstName=%{x}<br>TotalI
nvoices=%{y}<extra></extra>',
               'legendgroup': 'Ramos',
               'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
               'name': 'Ramos',
               'offsetgroup': 'Ramos',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Fernanda'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Harris<br>FirstName=%{x}<br>Total
Invoices=%{y}<extra></extra>',
               'legendgroup': 'Harris',
               'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
               'name': 'Harris',
               'offsetgroup': 'Harris',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Frank'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Ralston<br>FirstName=%{x}<br>Tota
lInvoices=%{y}<extra></extra>',
               'legendgroup': 'Ralston',
               'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
               'name': 'Ralston',
               'offsetgroup': 'Ralston',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Frank'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Wichterlová<br>FirstName=%{x}<br>
TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Wichterlová',
               'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
               'name': 'Wichterlová',
               'offsetgroup': 'Wichterlová',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['František'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Tremblay<br>FirstName=%{x}<br>Tot
```

```
alInvoices=%{y}<extra></extra>',
               'legendgroup': 'Tremblay',
               'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
               'name': 'Tremblay',
               'offsetgroup': 'Tremblay',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['François'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Zimmermann<br>FirstName=%{x}<br>T
otalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Zimmermann',
               'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
               'name': 'Zimmermann',
               'offsetgroup': 'Zimmermann',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Fynn'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Schneider<br>FirstName=%{x}<br>To
talInvoices=%{y}<extra></extra>',
               'legendgroup': 'Schneider',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': 'Schneider',
               'offsetgroup': 'Schneider',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Hannah'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Leacock<br>FirstName=%{x}<br>Tota
lInvoices=%{y}<extra></extra>',
               'legendgroup': 'Leacock',
               'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
               'name': 'Leacock',
               'offsetgroup': 'Leacock',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Heather'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Holý<br>FirstName=%{x}<br>TotalIn
voices=%{y}<extra></extra>',
               'legendgroup': 'Holý',
               'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
               'name': 'Holý',
               'offsetgroup': 'Holý',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Helena'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': "LastName=O'Reilly<br>FirstName=%{x}<br>Tot
alInvoices=%{y}<extra></extra>",
               'legendgroup': "O'Reilly",
               'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
               'name': "O'Reilly",
               'offsetgroup': "O'Reilly",
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Hugh'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Mercier<br>FirstName=%{x}<br>Tota
lInvoices=%{y}<extra></extra>',
               'legendgroup': 'Mercier',
               'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
               'name': 'Mercier',
               'offsetgroup': 'Mercier',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Isabelle'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Smith<br>FirstName=%{x}<br>TotalI
nvoices=%{y}<extra></extra>',
               'legendgroup': 'Smith',
               'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
               'name': 'Smith',
               'offsetgroup': 'Smith',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Jack'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Peterson<br>FirstName=%{x}<br>Tot
alInvoices=%{y}<extra></extra>',
               'legendgroup': 'Peterson',
               'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
               'name': 'Peterson',
               'offsetgroup': 'Peterson',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Jennifer'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Johansson<br>FirstName=%{x}<br>To
talInvoices=%{y}<extra></extra>',
               'legendgroup': 'Johansson',
               'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
               'name': 'Johansson',
               'offsetgroup': 'Johansson',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Joakim'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Van der Berg<br>FirstName=%{x}<br
>TotalInvoices=%{y}<extra></extra>',
               'legendgroup': 'Van der Berg',
               'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
               'name': 'Van der Berg',
               'offsetgroup': 'Van der Berg',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Johannes'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Gordon<br>FirstName=%{x}<br>Total
Invoices=%{y}<extra></extra>',
               'legendgroup': 'Gordon',
               'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
               'name': 'Gordon',
               'offsetgroup': 'Gordon',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['John'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Fernandes<br>FirstName=%{x}<br>To
talInvoices=%{y}<extra></extra>',
               'legendgroup': 'Fernandes',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': 'Fernandes',
               'offsetgroup': 'Fernandes',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['João'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Barnett<br>FirstName=%{x}<br>Tota
lInvoices=%{y}<extra></extra>',
               'legendgroup': 'Barnett',
               'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
               'name': 'Barnett',
               'offsetgroup': 'Barnett',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Julia'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Nielsen<br>FirstName=%{x}<br>Tota
lInvoices=%{y}<extra></extra>',
               'legendgroup': 'Nielsen',
               'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
               'name': 'Nielsen',
               'offsetgroup': 'Nielsen',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Kara'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Chase<br>FirstName=%{x}<br>TotalI
nvoices=%{y}<extra></extra>',
               'legendgroup': 'Chase',
               'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
               'name': 'Chase',
               'offsetgroup': 'Chase',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Kathy'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Kovács<br>FirstName=%{x}<br>Total
Invoices=%{y}<extra></extra>',
               'legendgroup': 'Kovács',
               'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
               'name': 'Kovács',
               'offsetgroup': 'Kovács',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Ladislav'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Köhler<br>FirstName=%{x}<br>Total
Invoices=%{y}<extra></extra>',
               'legendgroup': 'Köhler',
               'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
               'name': 'Köhler',
               'offsetgroup': 'Köhler',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Leonie'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Mancini<br>FirstName=%{x}<br>Tota
lInvoices=%{y}<extra></extra>',
               'legendgroup': 'Mancini',
               'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
               'name': 'Mancini',
               'offsetgroup': 'Mancini',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Lucas'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Rojas<br>FirstName=%{x}<br>TotalI
nvoices=%{y}<extra></extra>',
               'legendgroup': 'Rojas',
               'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
               'name': 'Rojas',
               'offsetgroup': 'Rojas',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Luis'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Gonçalves<br>FirstName=%{x}<br>To
talInvoices=%{y}<extra></extra>',
               'legendgroup': 'Gonçalves',
               'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
               'name': 'Gonçalves',
               'offsetgroup': 'Gonçalves',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Luís'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Sampaio<br>FirstName=%{x}<br>Tota
lInvoices=%{y}<extra></extra>',
               'legendgroup': 'Sampaio',
               'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
               'name': 'Sampaio',
               'offsetgroup': 'Sampaio',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Madalena'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Pareek<br>FirstName=%{x}<br>Total
Invoices=%{y}<extra></extra>',
               'legendgroup': 'Pareek',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': 'Pareek',
               'offsetgroup': 'Pareek',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Manoj'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Dubois<br>FirstName=%{x}<br>Total
Invoices=%{y}<extra></extra>',
               'legendgroup': 'Dubois',
               'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
               'name': 'Dubois',
               'offsetgroup': 'Dubois',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Marc'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Philips<br>FirstName=%{x}<br>Tota
lInvoices=%{y}<extra></extra>',
               'legendgroup': 'Philips',
               'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
               'name': 'Philips',
               'offsetgroup': 'Philips',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Mark'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Taylor<br>FirstName=%{x}<br>Total
Invoices=%{y}<extra></extra>',
               'legendgroup': 'Taylor',
               'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
               'name': 'Taylor',
               'offsetgroup': 'Taylor',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Mark'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Silk<br>FirstName=%{x}<br>TotalIn
voices=%{y}<extra></extra>',
               'legendgroup': 'Silk',
               'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
               'name': 'Silk',
               'offsetgroup': 'Silk',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Martha'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Brooks<br>FirstName=%{x}<br>Total
Invoices=%{y}<extra></extra>',
               'legendgroup': 'Brooks',
               'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
               'name': 'Brooks',
               'offsetgroup': 'Brooks',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Michelle'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Schröder<br>FirstName=%{x}<br>Tot
alInvoices=%{y}<extra></extra>',
               'legendgroup': 'Schröder',
               'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
               'name': 'Schröder',
               'offsetgroup': 'Schröder',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Niklas'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
              {'alignmentgroup': 'True',
               'hovertemplate': 'LastName=Gray<br>FirstName=%{x}<br>TotalIn
voices=%{y}<extra></extra>',
               'legendgroup': 'Gray',
               'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
               'name': 'Gray',
               'offsetgroup': 'Gray',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Patrick'], dtype=object),
               'xaxis': 'x',
               'y': array([7]),
               'yaxis': 'y'},
 {'alignmentgroup': 'True',
  'hovertemplate': 'LastName=Hughes<br>FirstName=%{x}<br>Total
Invoices=%{y}<extra></extra>',
  'legendgroup': 'Hughes',
  'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
  'name': 'Hughes',
  'offsetgroup': 'Hughes',
  'orientation': 'v',
  'showlegend': True,
  'textposition': 'auto',
  'type': 'bar',
  'x': array(['Phil'], dtype=object),
  'xaxis': 'x',
  'y': array([7]),
  'yaxis': 'y'},
 {'alignmentgroup': 'True',
  'hovertemplate': 'LastName=Srivastava<br>FirstName=%{x}<br>T
otalInvoices=%{y}<extra></extra>',
  'legendgroup': 'Srivastava',
  'marker': {'color': '#FECB52', 'pattern': {'shape': ''}},
  'name': 'Srivastava',
  'offsetgroup': 'Srivastava',
  'orientation': 'v',
  'showlegend': True,
  'textposition': 'auto',
  'type': 'bar',
  'x': array(['Puja'], dtype=object),
  'xaxis': 'x',
  'y': array([6]),
  'yaxis': 'y'},
 {'alignmentgroup': 'True',
  'hovertemplate': 'LastName=Cunningham<br>FirstName=%{x}<br>T
otalInvoices=%{y}<extra></extra>',
  'legendgroup': 'Cunningham',
  'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
  'name': 'Cunningham',
  'offsetgroup': 'Cunningham',
  'orientation': 'v',
  'showlegend': True,
  'textposition': 'auto',
  'type': 'bar',
  'x': array(['Richard'], dtype=object),
  'xaxis': 'x',
  'y': array([7]),
  'yaxis': 'y'},
 {'alignmentgroup': 'True',
  'hovertemplate': 'LastName=Brown<br>FirstName=%{x}<br>TotalI
nvoices=%{y}<extra></extra>',
  'legendgroup': 'Brown',
  'marker': {'color': '#EF553B', 'pattern': {'shape': ''}},
  'name': 'Brown',
  'offsetgroup': 'Brown',
  'orientation': 'v',
  'showlegend': True,
  'textposition': 'auto',
  'type': 'bar',
'x': array(['Robert'], dtype=object),
'xaxis': 'x',
'y': array([7]),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'LastName=Almeida<br>FirstName=%{x}<br>Tota
lInvoices=%{y}<extra></extra>',
'legendgroup': 'Almeida',
'marker': {'color': '#00cc96', 'pattern': {'shape': ''}},
'name': 'Almeida',
'offsetgroup': 'Almeida',
'orientation': 'v',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array(['Roberto'], dtype=object),
'xaxis': 'x',
'y': array([7]),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'LastName=Wójcik<br>FirstName=%{x}<br>Total
Invoices=%{y}<extra></extra>',
'legendgroup': 'Wójcik',
'marker': {'color': '#ab63fa', 'pattern': {'shape': ''}},
'name': 'Wójcik',
'offsetgroup': 'Wójcik',
'orientation': 'v',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array(['Stanisław'], dtype=object),
'xaxis': 'x',
'y': array([7]),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'LastName=Murray<br>FirstName=%{x}<br>Total
Invoices=%{y}<extra></extra>',
'legendgroup': 'Murray',
'marker': {'color': '#FFA15A', 'pattern': {'shape': ''}},
'name': 'Murray',
'offsetgroup': 'Murray',
'orientation': 'v',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array(['Steve'], dtype=object),
'xaxis': 'x',
'y': array([7]),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'LastName=Hämäläinen<br>FirstName=%{x}<br>T
otalInvoices=%{y}<extra></extra>',
'legendgroup': 'Hämäläinen',
'marker': {'color': '#19d3f3', 'pattern': {'shape': ''}},
'name': 'Hämäläinen',
'offsetgroup': 'Hämäläinen',
'orientation': 'v',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array(['Terhi'], dtype=object),
'xaxis': 'x',
'y': array([7]),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'LastName=Goyer<br>FirstName=%{x}<br>TotalI
nvoices=%{y}<extra></extra>',
'legendgroup': 'Goyer',
'marker': {'color': '#FF6692', 'pattern': {'shape': ''}},
'name': 'Goyer',
'offsetgroup': 'Goyer',
'orientation': 'v',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array(['Tim'], dtype=object),
'xaxis': 'x',
'y': array([7]),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'LastName=Stevens<br>FirstName=%{x}<br>Tota
lInvoices=%{y}<extra></extra>',
'legendgroup': 'Stevens',
'marker': {'color': '#B6E880', 'pattern': {'shape': ''}},
'name': 'Stevens',
'offsetgroup': 'Stevens',
'orientation': 'v',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array(['Victor'], dtype=object),
'xaxis': 'x',
'y': array([7]),
'yaxis': 'y'},
{'alignmentgroup': 'True',
'hovertemplate': 'LastName=Girard<br>FirstName=%{x}<br>Total
Invoices=%{y}<extra></extra>',
'legendgroup': 'Girard',
'marker': {'color': '#FF97FF', 'pattern': {'shape': ''}},
'name': 'Girard',
'offsetgroup': 'Girard',
'orientation': 'v',
'showlegend': True,
'textposition': 'auto',
'type': 'bar',
'x': array(['Wyatt'], dtype=object),
'xaxis': 'x',
'y': array([7]),
'yaxis': 'y'}],
'layout': {'barmode': 'relative',
 'legend': {'title': {'text': 'LastName'}, 'tracegroupgap':
0},
 'margin': {'t': 60},
 'template': '...',
 'title': {'text': 'Customer Invoice Totals'},
 'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'FirstName'}},
 'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
ext': 'TotalInvoices'}}}
}))
```

In [23]:

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

vn.ask(question=question)
```

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

```
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\r\nFROM customers c\r\nJOIN
invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastNam
e'}, {'role': 'user', 'content': 'what are the top 5 countries that customer
s come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(DIST
INCT CustomerId) AS NumCustomers\r\nFROM customers\r\nGROUP BY Country\r\nOR
DER BY NumCustomers DESC\r\nLIMIT 5'}, {'role': 'user', 'content': 'How many
customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FRO
M customers'}, {'role': 'user', 'content': '  \n    List all albums and thei
r corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT
"albums".Title, "artists".Name \r\nFROM "albums"\r\nJOIN "artists" ON "album
s"."ArtistId" = "artists"."ArtistId"\r\nORDER BY "albums"."Title"'}, {'rol
e': 'user', 'content': '  \n    Find all tracks with a name containing "Wha
t" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * \r\nFR
OM tracks\r\nWHERE Name LIKE '%what%'\r\nORDER BY Name"}, {'role': 'user',
'content': 'Can you list all tables in the SQLite database catalog?'}, {'rol
e': 'assistant', 'content': "SELECT name FROM sqlite_schema WHERE type='tabl
e'"}, {'role': 'user', 'content': '  \n    Find the total number of invoices
per country:\n'}]
Info: Ollama parameters:
model=gemma2:2b,
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\r\nFROM cu
stomers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.Fi
rstName, c.LastName"}, {"role": "user", "content": "what are the top 5 count
ries that customers come from?"}, {"role": "assistant", "content": "SELECT C
ountry, COUNT(DISTINCT CustomerId) AS NumCustomers\r\nFROM customers\r\nGROU
P BY Country\r\nORDER BY NumCustomers DESC\r\nLIMIT 5"}, {"role": "user", "c
ontent": "How many customers are there"}, {"role": "assistant", "content":
"SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "  \n    List
all albums and their corresponding artist names  \n"}, {"role": "assistant",
"content": "SELECT \"albums\".Title, \"artists\".Name \r\nFROM \"albums\"\r
\nJOIN \"artists\" ON \"albums\".\"ArtistId\" = \"artists\".\"ArtistId\"\r\n
ORDER BY \"albums\".\"Title\""}, {"role": "user", "content": "  \n    Find a
ll tracks with a name containing \"What\" (case-insensitive)\n"}, {"role":
"assistant", "content": "SELECT * \r\nFROM tracks\r\nWHERE Name LIKE '%wha
t%'\r\nORDER BY Name"}, {"role": "user", "content": "Can you list all tables
in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT
name FROM sqlite_schema WHERE type='table'"}, {"role": "user", "content": "
\n    Find the total number of invoices per country:\n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:25:35.344561908Z', 'mess
age': {'role': 'assistant', 'content': "```sql\nSELECT c.Country, COUNT(i.In
voiceId) AS TotalInvoices \r\nFROM customers c \r\nJOIN invoices i ON c.Cust
omerId = i.CustomerId \r\nGROUP BY c.Country\r\nORDER BY TotalInvoices DESC;
\r\n``` \n\n\n**Explanation:**\n\n1. **`SELECT c.Country, COUNT(i.InvoiceId)
AS TotalInvoices`**: We select the country name (`c.Country`) and count the
```

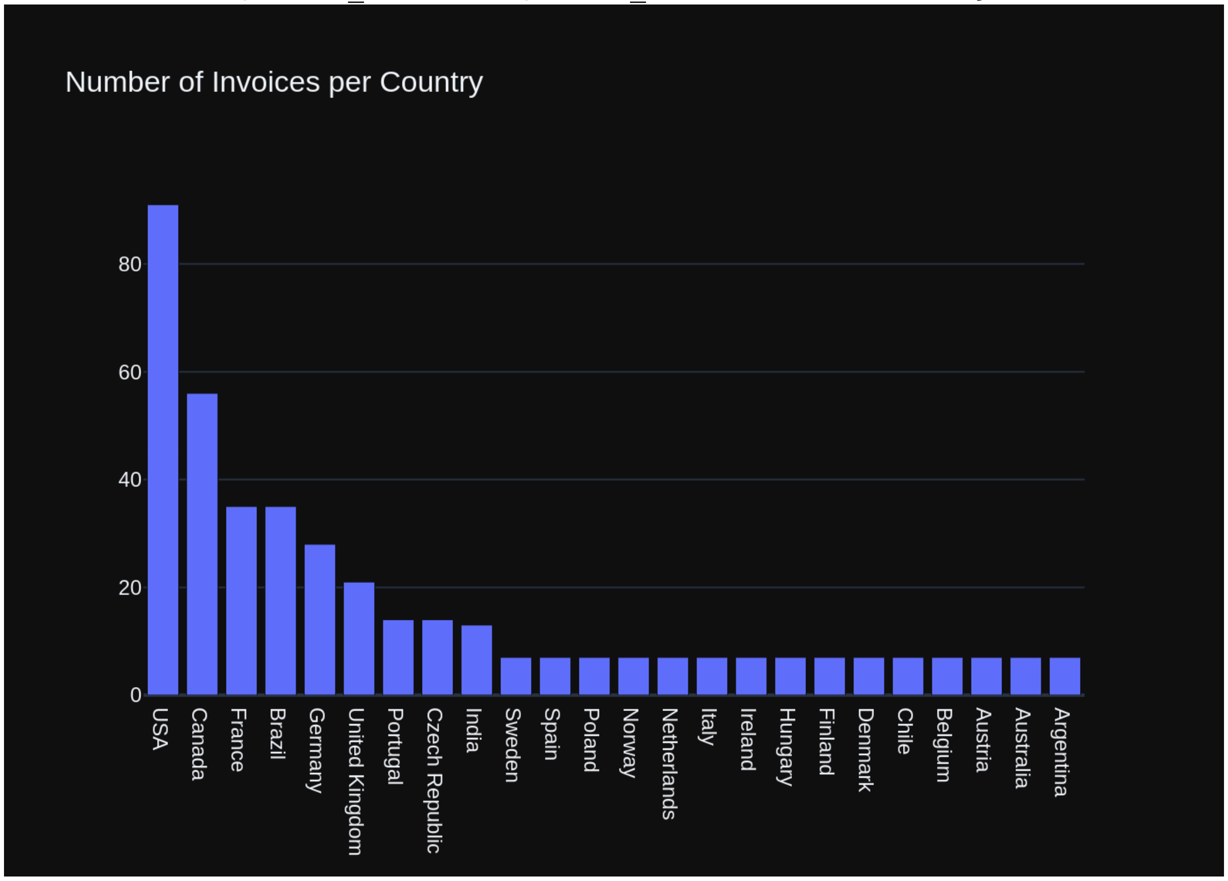
```
number of invoices associated with that country (`COUNT(i.InvoiceId) AS Tota
lInvoices`). \n2. **`FROM customers c JOIN invoices i ON c.CustomerId = i.Cu
stomerId`**: We join the `customers` table (aliased as `c`) with the `invoic
es` table (aliased as `i`), connecting them based on shared customer IDs (`C
ustomerId`). This is crucial for linking customers to their corresponding in
voices.\n3. **`GROUP BY c.Country`**:  We group the results by country, allo
wing us to count invoices per distinct country. \n4. **`ORDER BY TotalInvoic
es DESC`**: Finally, we sort the resulting table in descending order based o
n the `TotalInvoices` column (i.e., countries with the most invoices firs
t).\n\n\n\nLet me know if you'd like any further clarifications! "}, 'done_r
eason': 'stop', 'done': True, 'total_duration': 41343739573, 'load_duratio
n': 20239574, 'prompt_eval_count': 1434, 'prompt_eval_duration': 1843819700
0, 'eval_count': 271, 'eval_duration': 22022279000}
LLM Response: ```sql
SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.Country
ORDER BY TotalInvoices DESC;
```

Explanation:

1. **`SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices`**: We select th
e country name (`c.Country`) and count the number of invoices associated wit
h that country (`COUNT(i.InvoiceId) AS TotalInvoices`).
2. **`FROM customers c JOIN invoices i ON c.CustomerId = i.CustomerId`**: We
join the `customers` table (aliased as `c`) with the `invoices` table (alias
ed as `i`), connecting them based on shared customer IDs (`CustomerId`). Thi
s is crucial for linking customers to their corresponding invoices.
3. **`GROUP BY c.Country`**: We group the results by country, allowing us t
o count invoices per distinct country.
4. **`ORDER BY TotalInvoices DESC`**: Finally, we sort the resulting table i
n descending order based on the `TotalInvoices` column (i.e., countries with
the most invoices first).

Let me know if you'd like any further clarifications!
Info: Output from LLM: ```sql
SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.Country
ORDER BY TotalInvoices DESC;
```

Explanation:

1. **`SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices`**: We select th
e country name (`c.Country`) and count the number of invoices associated wit
h that country (`COUNT(i.InvoiceId) AS TotalInvoices`).
2. **`FROM customers c JOIN invoices i ON c.CustomerId = i.CustomerId`**: We
join the `customers` table (aliased as `c`) with the `invoices` table (alias
```

```
ed as `i`), connecting them based on shared customer IDs (`CustomerId`). Thi
s is crucial for linking customers to their corresponding invoices.
3. **`GROUP BY c.Country`**:  We group the results by country, allowing us t
o count invoices per distinct country.
4. **`ORDER BY TotalInvoices DESC`**: Finally, we sort the resulting table i
n descending order based on the `TotalInvoices` column (i.e., countries with
the most invoices first).



Let me know if you'd like any further clarifications!
Extracted SQL: SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.Country
ORDER BY TotalInvoices DESC
SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.Country
ORDER BY TotalInvoices DESC
           Country  TotalInvoices
0              USA             91
1           Canada             56
2           France             35
3           Brazil             35
4          Germany             28
5   United Kingdom             21
6         Portugal             14
7   Czech Republic             14
8            India             13
9           Sweden              7
10           Spain              7
11          Poland              7
12          Norway              7
13     Netherlands              7
14           Italy              7
15         Ireland              7
16         Hungary              7
17         Finland              7
18         Denmark              7
19           Chile              7
20         Belgium              7
21         Austria              7
22       Australia              7
23       Argentina              7
Info: Ollama parameters:
model=gemma2:2b,
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 \r\nFROM customers c \r\nJOIN invoices i ON c.CustomerId = i.CustomerId
\r\nGROUP BY c.Country\r\nORDER BY TotalInvoices DESC\n\nThe following is in
formation about the resulting pandas DataFrame 'df': \nRunning df.dtypes giv
es:\n Country          object\nTotalInvoices     int64\ndtype: object"}, {"r
ole": "user", "content": "Can you generate the Python plotly code to chart t
he 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 cod
e."}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:25:42.056122475Z', 'mess
age': {'role': 'assistant', 'content': "```python\nimport plotly.graph_objec
ts as go\n\nfig = go.Figure(data=go.Bar(x=df.Country, y=df['TotalInvoice
s'])) \nfig.update_layout(title='Number of Invoices per Country')\nfig.show
()\n```"}, 'done_reason': 'stop', 'done': True, 'total_duration': 669089721
5, 'load_duration': 17611644, 'prompt_eval_count': 200, 'prompt_eval_duratio
n': 2113351000, 'eval_count': 62, 'eval_duration': 4422848000}
```



```
Out[23]: ('SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices \r\nFROM customers
         c \r\nJOIN invoices i ON c.CustomerId = i.CustomerId \r\nGROUP BY c.Country
         \r\nORDER BY TotalInvoices DESC',
                     Country  TotalInvoices
          0              USA             91
          1           Canada             56
          2           France             35
          3           Brazil             35
          4          Germany             28
          5   United Kingdom             21
          6         Portugal             14
          7   Czech Republic             14
          8            India             13
          9           Sweden              7
          10           Spain              7
          11          Poland              7
          12          Norway              7
          13     Netherlands              7
          14           Italy              7
          15         Ireland              7
          16         Hungary              7
          17         Finland              7
          18         Denmark              7
          19           Chile              7
          20         Belgium              7
          21         Austria              7
          22       Australia              7
          23       Argentina              7,
          Figure({
              'data': [{'type': 'bar',
                        'x': array(['USA', 'Canada', 'France', 'Brazil', 'Germany',
         'United Kingdom',
                                    'Portugal', 'Czech Republic', 'India', 'Sweden',
         'Spain', 'Poland',
                                    'Norway', 'Netherlands', 'Italy', 'Ireland', 'Hu
         ngary', 'Finland',
                                    'Denmark', 'Chile', 'Belgium', 'Austria', 'Austr
         alia', 'Argentina'],
                                   dtype=object),
                        'y': array([91, 56, 35, 35, 28, 21, 14, 14, 13,  7,  7,  7,
         7,  7,  7,  7,  7,  7,
                                     7,  7,  7,  7,  7,  7])}],
              'layout': {'template': '...', 'title': {'text': 'Number of Invoices pe
         r Country'}}
          }))
```

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

         vn.ask(question=question)
```

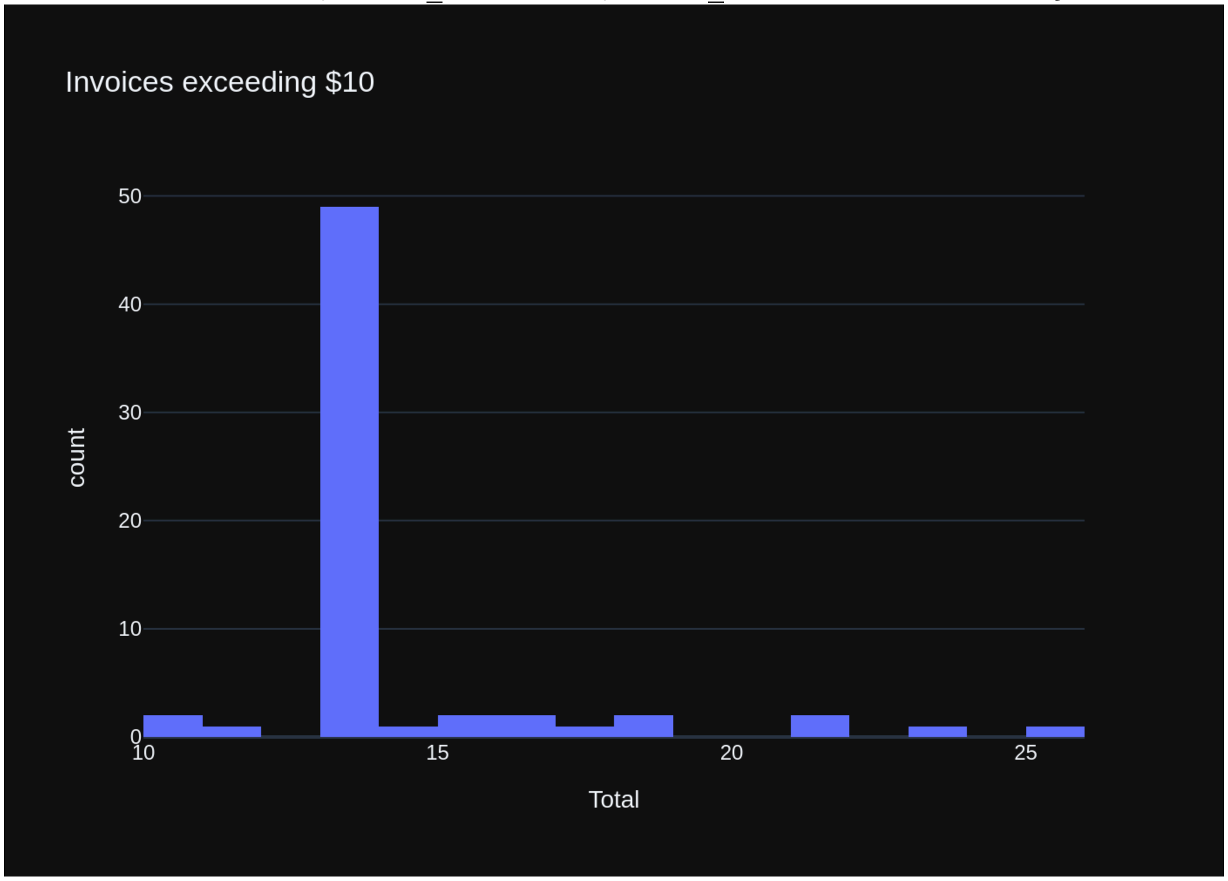
```
Number of requested results 10 is greater than number of elements in index
7, updating n_results = 7
Number of requested results 10 is greater than number of elements in index
1, updating n_results = 1
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please
help to generate a SQL query to answer the question. Your response should ON
LY be based on the given context and follow the response guidelines and form
at instructions. \n===Tables \nCREATE TABLE "invoice_items"\r\n(\r\n    Invo
iceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEG
ER  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 (Invoice
Id) 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_InvoiceLi
neInvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE TABLE "invoices"\r\n(\r
\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    CustomerI
d INTEGER  NOT NULL,\r\n    InvoiceDate DATETIME  NOT NULL,\r\n    BillingAd
dress NVARCHAR(70),\r\n    BillingCity NVARCHAR(40),\r\n    BillingState NVA
RCHAR(40),\r\n    BillingCountry NVARCHAR(40),\r\n    BillingPostalCode NVAR
CHAR(10),\r\n    Total NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (Customer
Id) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E 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 AUTOINCREMEN
T 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    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_EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE TABL
E "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT N
ULL,\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    C
ity 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 DE
LETE 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 NVARC
HAR(20)  NOT NULL,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVA
RCHAR(30),\r\n    ReportsTo INTEGER,\r\n    BirthDate DATETIME,\r\n    HireD
ate 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 NVARC
HAR(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_Cus
tomerSupportRepId ON "customers" (SupportRepId)\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    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\r\nFROM c
ustomers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.F
irstName, c.LastName'}, {'role': 'user', 'content': '  \n    Find the total 
number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELEC
T c.Country, COUNT(i.InvoiceId) AS TotalInvoices \r\nFROM customers c \r\nJO
IN invoices i ON c.CustomerId = i.CustomerId \r\nGROUP BY c.Country\r\nORDER
BY TotalInvoices DESC'}, {'role': 'user', 'content': 'How many customers are
there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customers'},
{'role': 'user', 'content': 'what are the top 5 countries that customers com
e from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(DISTINCT 
CustomerId) AS NumCustomers\r\nFROM customers\r\nGROUP BY Country\r\nORDER B
Y NumCustomers DESC\r\nLIMIT 5'}, {'role': 'user', 'content': '  \n    List 
all albums and their corresponding artist names  \n'}, {'role': 'assistant',
'content': 'SELECT "albums".Title, "artists".Name \r\nFROM "albums"\r\nJOIN 
"artists" ON "albums"."ArtistId" = "artists"."ArtistId"\r\nORDER BY "album
s"."Title"'}, {'role': 'user', 'content': '  \n    Find all tracks with a na
me containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'conten
t': "SELECT * \r\nFROM tracks\r\nWHERE Name LIKE '%what%'\r\nORDER BY Nam
e"}, {'role': 'user', 'content': 'Can you list all tables in the SQLite data
base catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_s
chema WHERE type='table'"}, {'role': 'user', 'content': '  \n    List all in
voices with a total exceeding $10:\n'}]
Info: Ollama parameters:
model=gemma2:2b,
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\r\nFROM customers c\r\nJOIN invoices i ON c.Custom
erId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName"}, {"role": "user",
"content": "  \n    Find the total number of invoices per country:\n"}, {"ro
le": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalI
nvoices \r\nFROM customers c \r\nJOIN invoices i ON c.CustomerId = i.Custome
rId \r\nGROUP BY c.Country\r\nORDER BY TotalInvoices DESC"}, {"role": "use
r", "content": "How many customers are there"}, {"role": "assistant", "conte
nt": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "what ar
e the top 5 countries that customers come from?"}, {"role": "assistant", "co
ntent": "SELECT Country, COUNT(DISTINCT CustomerId) AS NumCustomers\r\nFROM
customers\r\nGROUP BY Country\r\nORDER BY NumCustomers DESC\r\nLIMIT 5"},
{"role": "user", "content": "  \n    List all albums and their corresponding
artist names  \n"}, {"role": "assistant", "content": "SELECT \"albums\".Titl
e, \"artists\".Name \r\nFROM \"albums\"\r\nJOIN \"artists\" ON \"albums
\".\"ArtistId\" = \"artists\".\"ArtistId\"\r\nORDER BY \"albums\".\"Title
\""}, {"role": "user", "content": "  \n    Find all tracks with a name conta
ining \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SEL
ECT * \r\nFROM tracks\r\nWHERE Name LIKE '%what%'\r\nORDER BY Name"}, {"rol
e": "user", "content": "Can you list all tables in the SQLite database catal
og?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_schema WHER
E type='table'"}, {"role": "user", "content": "  \n    List all invoices wit
h a total exceeding $10:\n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:26:03.412678389Z', 'mess
age': {'role': 'assistant', 'content': '```sql\nSELECT * \r\nFROM invoices\r
\nWHERE Total > 10;\n```'}, 'done_reason': 'stop', 'done': True, 'total_dura
```

```
tion': 21254229846, 'load_duration': 18039118, 'prompt_eval_count': 1452, 'p
rompt_eval_duration': 18617137000, 'eval_count': 21, 'eval_duration': 158761
7000}
LLM Response: ```sql
SELECT *
FROM invoices
WHERE Total > 10;
```
Info: Output from LLM: ```sql
SELECT *
FROM invoices
WHERE Total > 10;
```
Extracted SQL: SELECT *
FROM invoices
WHERE Total > 10
SELECT *
FROM invoices
WHERE Total > 10
    InvoiceId  CustomerId          InvoiceDate             BillingAddress  \
0           5          23  2009-01-11 00:00:00            69 Salem Street
1          12           2  2009-02-11 00:00:00    Theodor-Heuss-Straße 34
2          19          40  2009-03-14 00:00:00             8, Rue Hanovre
3          26          19  2009-04-14 00:00:00            1 Infinite Loop
4          33          57  2009-05-15 00:00:00            Calle Lira, 198
..        ...         ...                  ...                        ...
59        383          10  2013-08-12 00:00:00  Rua Dr. Falcão Filho, 155
60        390          48  2013-09-12 00:00:00     Lijnbaansgracht 120bg
61        397          27  2013-10-13 00:00:00            1033 N Park Ave
62        404           6  2013-11-13 00:00:00              Rilská 3174/6
63        411          44  2013-12-14 00:00:00            Porthaninkatu 9

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

[64 rows x 9 columns]
Info: Ollama parameters:
model=gemma2:2b,
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 * \r\nFROM invoices\r\nWHERE Total > 10\n
\nThe following is information about the resulting pandas DataFrame 'df': \n
```

```
Running df.dtypes gives:\n InvoiceId              int64\nCustomerId
int64\nInvoiceDate           object\nBillingAddress       object\nBillingCi
ty            object\nBillingState          object\nBillingCountry       obj
ect\nBillingPostalCode     object\nTotal                float64\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': 'gemma2:2b', 'created_at': '2024-08-01T18:26:08.696777276Z', 'mess
age': {'role': 'assistant', 'content': '```python\nimport plotly.express as
px\n\nfig = px.histogram(df, x="Total", title="Invoices exceeding $10")\nfi
g.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration': 5255
562933, 'load_duration': 20558610, 'prompt_eval_count': 207, 'prompt_eval_du
ration': 2446668000, 'eval_count': 38, 'eval_duration': 2738677000}
```



```
Out[24]: ('SELECT * \r\nFROM invoices\r\nWHERE Total > 10',
              InvoiceId  CustomerId          InvoiceDate             BillingAddress  \
          0           5          23  2009-01-11 00:00:00            69 Salem Street
          1          12           2  2009-02-11 00:00:00    Theodor-Heuss-Straße 34
          2          19          40  2009-03-14 00:00:00             8, Rue Hanovre
          3          26          19  2009-04-14 00:00:00           1 Infinite Loop
          4          33          57  2009-05-15 00:00:00           Calle Lira, 198
          ..        ...         ...                  ...                        ...
          59        383          10  2013-08-12 00:00:00  Rua Dr. Falcão Filho, 155
          60        390          48  2013-09-12 00:00:00     Lijnbaansgracht 120bg
          61        397          27  2013-10-13 00:00:00             1033 N Park Ave
          62        404           6  2013-11-13 00:00:00               Rilská 3174/6
          63        411          44  2013-12-14 00:00:00           Porthaninkatu 9

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

          [64 rows x 9 columns],
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'bingroup': 'x',
                        'hovertemplate': 'Total=%{x}<br>count=%{y}<extra></extra>',
                        'legendgroup': '',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': '',
                        '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 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': 'count'}}}
 }))
```

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 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_InvoiceLineInvoiceId ON "invoice_items"
(InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)
\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREA
TE TABLE "employees"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    FirstName NVARCHA
R(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INTEGER,\r\n
BirthDate DATETIME,\r\n    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 "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    Media
TypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(2
20),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    Uni
tPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "alb
ums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREI
GN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION O
N UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_type
s" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCRE
ATE 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 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\n\n=
==Additional Context \n\nIn the chinook database invoice means order\n\n===R
esponse Guidelines \n1. If the provided context is sufficient, please genera
te 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 * \r\nFROM invoices\r\nWHERE Total > 10'}, {'role':
'user', 'content': '  \n    Get the total number of invoices for each custom
er\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, CO
UNT(i.InvoiceId) AS TotalInvoices\r\nFROM customers c\r\nJOIN invoices i ON
c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName'}, {'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 \r\nFROM customers c \r\nJOIN invoices i ON c.CustomerId
= i.CustomerId \r\nGROUP BY c.Country\r\nORDER BY TotalInvoices DESC'}, {'ro
le': 'user', 'content': 'How many customers are there'}, {'role': 'assistan
t', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'conten
t': 'what are the top 5 countries that customers come from?'}, {'role': 'ass
istant', 'content': 'SELECT Country, COUNT(DISTINCT CustomerId) AS NumCustom
ers\r\nFROM customers\r\nGROUP BY Country\r\nORDER BY NumCustomers DESC\r\nL
IMIT 5'}, {'role': 'user', 'content': '  \n    Find all tracks with a name c
ontaining "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "S
ELECT * \r\nFROM tracks\r\nWHERE Name LIKE '%what%'\r\nORDER BY Name"}, {'ro
le': 'user', 'content': '  \n    List all albums and their corresponding art
ist names  \n'}, {'role': 'assistant', 'content': 'SELECT "albums".Title, "a
rtists".Name \r\nFROM "albums"\r\nJOIN "artists" ON "albums"."ArtistId" = "a
rtists"."ArtistId"\r\nORDER BY "albums"."Title"'}, {'role': 'user', 'conten
t': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'as
sistant', 'content': "SELECT name FROM sqlite_schema WHERE type='table'"},
{'role': 'user', 'content': '  \n    Find all invoices since 2010 and the to
tal amount invoiced:\n'}]
Info: Ollama parameters:
model=gemma2:2b,
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 * \r\nFROM invoices\r\nWHER
E Total > 10"}, {"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\r\nFROM customers
c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName,
c.LastName"}, {"role": "user", "content": "  \n    Find the total number of
invoices per country:\n"}, {"role": "assistant", "content": "SELECT c.Countr
y, COUNT(i.InvoiceId) AS TotalInvoices \r\nFROM customers c \r\nJOIN invoice
s i ON c.CustomerId = i.CustomerId \r\nGROUP BY c.Country\r\nORDER BY TotalI
nvoices DESC"}, {"role": "user", "content": "How many customers are there"},
{"role": "assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role":
"user", "content": "what are the top 5 countries that customers come fro
m?"}, {"role": "assistant", "content": "SELECT Country, COUNT(DISTINCT Custo
merId) AS NumCustomers\r\nFROM customers\r\nGROUP BY Country\r\nORDER BY Num
Customers DESC\r\nLIMIT 5"}, {"role": "user", "content": "  \n    Find all t
racks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assi
stant", "content": "SELECT * \r\nFROM tracks\r\nWHERE Name LIKE '%what%'\r\n
ORDER BY Name"}, {"role": "user", "content": "  \n    List all albums and th
eir corresponding artist names  \n"}, {"role": "assistant", "content": "SELE
CT \"albums\".Title, \"artists\".Name \r\nFROM \"albums\"\r\nJOIN \"artists
\" ON \"albums\".\"ArtistId\" = \"artists\".\"ArtistId\"\r\nORDER BY \"album
s\".\"Title\""}, {"role": "user", "content": "Can you list all tables in the
SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FR
OM sqlite_schema WHERE type='table'"}, {"role": "user", "content": "  \n
Find all invoices since 2010 and the total amount invoiced:\n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:26:54.648394071Z', 'mess
age': {'role': 'assistant', 'content': '```sql\nSELECT i.InvoiceId, SUM(i.To
tal) AS TotalAmount\r\nFROM invoices i\r\nWHERE i.InvoiceDate >= \'2010-01-0
1\'\r\nGROUP BY i.InvoiceId; \r\n``` \n\n**Explanation:** \n* **`SELECT i.In
voiceId, SUM(i.Total) AS TotalAmount`**: This line selects the `InvoiceId` f
rom the `invoices` table and calculates the sum of the `Total` column for ea
ch invoice using the `SUM()` function, assigning it to a new column named "T
otalAmount".  \n* **`FROM invoices i`**: This specifies that we are querying
data from the `invoices` table, aliasing it as `i`. \r\n* **`WHERE i.Invoice
Date >= \'2010-01-01\'`**:  This clause filters invoices to only include tho
se with a date (`InvoiceDate`) after or equal to January 1st, 2010.\r\n* **`
GROUP BY i.InvoiceId`**: This groups the results based on the `InvoiceId`, a
llowing us to calculate the total for each distinct invoice id. \r\n\r\nLet
me know if you have other queries you\'d like help with!'}, 'done_reason':
'stop', 'done': True, 'total_duration': 45861543506, 'load_duration': 220463
12, 'prompt_eval_count': 1637, 'prompt_eval_duration': 22999575000, 'eval_co
unt': 267, 'eval_duration': 21772129000}
LLM Response: ```sql
SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount
FROM invoices i
WHERE i.InvoiceDate >= '2010-01-01'
GROUP BY i.InvoiceId;
```

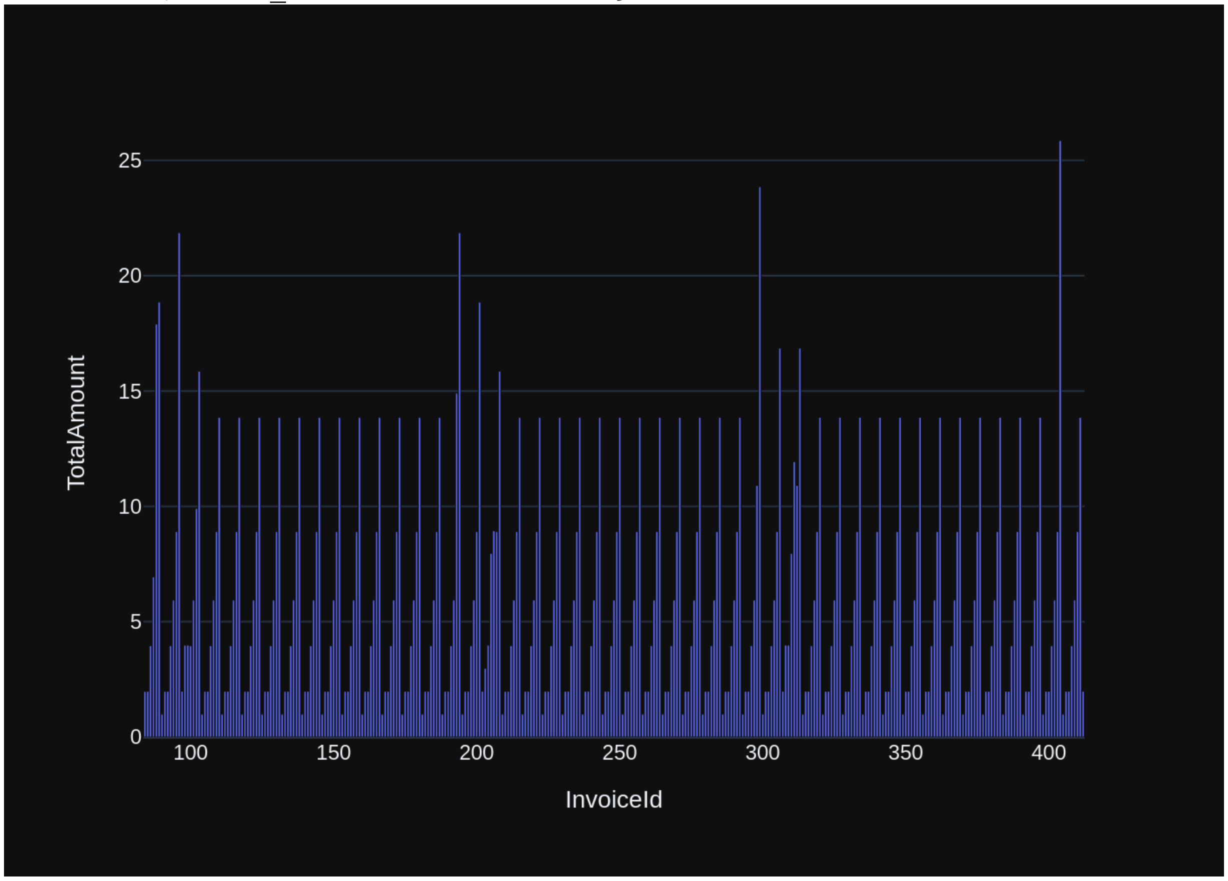
Explanation:
* **`SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount`**: This line selects t
he `InvoiceId` from the `invoices` table and calculates the sum of the `Tota
l` column for each invoice using the `SUM()` function, assigning it to a new
column named "TotalAmount".
* **`FROM invoices i`**: This specifies that we are querying data from the `
invoices` table, aliasing it as `i`.
* **`WHERE i.InvoiceDate >= '2010-01-01'`**: This clause filters invoices t
o only include those with a date (`InvoiceDate`) after or equal to January 1
st, 2010.
* **`GROUP BY i.InvoiceId`**: This groups the results based on the `InvoiceI
d`, allowing us to calculate the total for each distinct invoice id.

Let me know if you have other queries you'd like help with!
Info: Output from LLM: ```sql
SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount
```
```

```
FROM invoices i
WHERE i.InvoiceDate >= '2010-01-01'
GROUP BY i.InvoiceId;
```

Explanation:
* **`SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount`**: This line selects t
he `InvoiceId` from the `invoices` table and calculates the sum of the `Tota
l` column for each invoice using the `SUM()` function, assigning it to a new
column named "TotalAmount".
* **`FROM invoices i`**: This specifies that we are querying data from the `
invoices` table, aliasing it as `i`.
* **`WHERE i.InvoiceDate >= '2010-01-01'`**: This clause filters invoices t
o only include those with a date (`InvoiceDate`) after or equal to January 1
st, 2010.
* **`GROUP BY i.InvoiceId`**: This groups the results based on the `InvoiceI
d`, allowing us to calculate the total for each distinct invoice id.

```
Let me know if you have other queries you'd like help with!
Extracted SQL: SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount
FROM invoices i
WHERE i.InvoiceDate >= '2010-01-01'
GROUP BY i.InvoiceId
SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount
FROM invoices i
WHERE i.InvoiceDate >= '2010-01-01'
GROUP BY i.InvoiceId
     InvoiceId  TotalAmount
0           84         1.98
1           85         1.98
2           86         3.96
3           87         6.94
4           88        17.91
..         ...          ...
324        408         3.96
325        409         5.94
326        410         8.91
327        411        13.86
328        412         1.99

[329 rows x 2 columns]
Info: Ollama parameters:
model=gemma2:2b,
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 invoices since 2010 and the total amount invoiced:\n'\n\nThe
DataFrame was produced using this query: SELECT i.InvoiceId, SUM(i.Total) AS
TotalAmount\r\nFROM invoices i\r\nWHERE i.InvoiceDate >= '2010-01-01'\r\nGRO
UP BY i.InvoiceId\n\nThe following is information about the resulting pandas
DataFrame 'df': \nRunning df.dtypes gives:\n InvoiceId        int64\nTotalAm
ount    float64\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': 'gemma2:2b', 'created_at': '2024-08-01T18:26:59.191665799Z', 'mess
age': {'role': 'assistant', 'content': "```python\nimport plotly.express as
px\n\nfig = px.bar(df, x='InvoiceId', y='TotalAmount') \n```"}, 'done_reaso
n': 'stop', 'done': True, 'total_duration': 4516668152, 'load_duration': 222
12699, 'prompt_eval_count': 210, 'prompt_eval_duration': 2201043000, 'eval_c
ount': 32, 'eval_duration': 2246462000}
```



Out[25]:
```
("SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\r\nFROM invoices i\r\nWHE
RE i.InvoiceDate >= '2010-01-01'\r\nGROUP BY i.InvoiceId",
      InvoiceId  TotalAmount
 0           84         1.98
 1           85         1.98
 2           86         3.96
 3           87         6.94
 4           88        17.91
 ..         ...          ...
 324        408         3.96
 325        409         5.94
 326        410         8.91
 327        411        13.86
 328        412         1.99

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

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

vn.ask(question=question)
```

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

```
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_EmployeeReportsTo ON "employ
ees" (ReportsTo)\n\nCREATE TABLE "employees"\r\n(\r\n    EmployeeId INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NUL
L,\r\n    FirstName NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n
ReportsTo INTEGER,\r\n    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 "customers"\r\n(\r\n    Custo
merId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCHAR
(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVARC
HAR(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) REF
ERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO AC
TION\r\n)\n\nCREATE INDEX IFK_CustomerSupportRepId ON "customers" (SupportRe
pId)\n\nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AU
TOINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDat
e DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    BillingCity
NVARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCH
AR(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  N
OT 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 "invoic
e_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n
UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n
FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE
NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "trac
ks" (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 N
ULL,\r\n    Name NVARCHAR(120)\r\n)\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\nCREATE TABLE "albums"\r\n(\r\n    Albu
mId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)
NOT NULL,\r\n    ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) R
EFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI
ON\r\n)\n\nCREATE TABLE sqlite_stat1(tbl,idx,stat)\n\n\n===Additional Contex
t \n\nIn the chinook database invoice means order\n\n===Response Guidelines
\n1. If the provided context is sufficient, please generate a valid SQL quer
y without any explanations for the question. \n2. If the provided context is
almost sufficient but requires knowledge of a specific string in a particula
r column, please generate an intermediate SQL query to find the distinct str
ings in that column. Prepend the query with a comment saying intermediate_sq
l \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\r\nFROM
customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.
FirstName, c.LastName'}, {'role': 'user', 'content': 'what are the top 5 cou
ntries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT
Country, COUNT(DISTINCT CustomerId) AS NumCustomers\r\nFROM customers\r\nGRO
UP BY Country\r\nORDER BY NumCustomers DESC\r\nLIMIT 5'}, {'role': 'user',
'content': '  \n    Find all invoices since 2010 and the total amount invoic
ed:\n'}, {'role': 'assistant', 'content': "SELECT i.InvoiceId, SUM(i.Total)
AS TotalAmount\r\nFROM invoices i\r\nWHERE i.InvoiceDate >= '2010-01-01'\r\n
GROUP BY i.InvoiceId"}, {'role': 'user', 'content': '  \n    List all albums
and their corresponding artist names  \n'}, {'role': 'assistant', 'content':
'SELECT "albums".Title, "artists".Name \r\nFROM "albums"\r\nJOIN "artists" O
N "albums"."ArtistId" = "artists"."ArtistId"\r\nORDER BY "albums"."Title"'},
{'role': 'user', 'content': '  \n    Find the total number of invoices per c
ountry:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.Inv
oiceId) AS TotalInvoices \r\nFROM customers c \r\nJOIN invoices i ON c.Custo
merId = i.CustomerId \r\nGROUP BY c.Country\r\nORDER BY TotalInvoices DES
C'}, {'role': 'user', 'content': '  \n    List all invoices with a total exc
eeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \r\nFROM invoice
s\r\nWHERE Total > 10'}, {'role': 'user', 'content': '  \n    Find all track
s with a name containing "What" (case-insensitive)\n'}, {'role': 'assistan
t', 'content': "SELECT * \r\nFROM tracks\r\nWHERE Name LIKE '%what%'\r\nORDE
R BY Name"}, {'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_schema WHERE
type='table'"}, {'role': 'user', 'content': "  \n    List all employees and
their reporting manager's name (if any):\n"}]
Info: Ollama parameters:
model=gemma2:2b,
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(1
0,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (Cu
stomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TAB
LE \"invoice_items\"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCRE
MENT NOT NULL,\r\n    InvoiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  N
OT NULL,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    Quantity INTEGER
NOT NULL,\r\n    FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId)
\r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackI
d) 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    MediaType
Id INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(22
0),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    Unit
Price NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES \"alb
ums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FORE
IGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTIO
N 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
\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 sqlite_st
at1(tbl,idx,stat)\n\n\n===Additional Context \n\nIn the chinook database inv
oice means order\n\n===Response Guidelines \n1. If the provided context is s
ufficient, please generate a valid SQL query without any explanations for th
e question. \n2. If the provided context is almost sufficient but requires k
nowledge of a specific string in a particular column, please generate an int
ermediate SQL query to find the distinct strings in that column. Prepend the
query with a comment saying intermediate_sql \n3. If the provided context is
insufficient, please explain why it can't be generated. \n4. Please use the
most relevant table(s). \n5. If the question has been asked and answered bef
ore, please repeat the answer exactly as it was given before. \n"}, {"role":
"user", "content": "  \n    Get the total number of invoices for each custom
er\n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastName, CO
UNT(i.InvoiceId) AS TotalInvoices\r\nFROM customers c\r\nJOIN invoices i ON
c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName"}, {"role":
"user", "content": "what are the top 5 countries that customers come fro
m?"}, {"role": "assistant", "content": "SELECT Country, COUNT(DISTINCT Custo
merId) AS NumCustomers\r\nFROM customers\r\nGROUP BY Country\r\nORDER BY Num
Customers DESC\r\nLIMIT 5"}, {"role": "user", "content": "  \n    Find all i
nvoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant",
"content": "SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\r\nFROM invoices
i\r\nWHERE i.InvoiceDate >= '2010-01-01'\r\nGROUP BY i.InvoiceId"}, {"role":
"user", "content": "  \n    List all albums and their corresponding artist n
ames  \n"}, {"role": "assistant", "content": "SELECT \"albums\".Title, \"art
ists\".Name \r\nFROM \"albums\"\r\nJOIN \"artists\" ON \"albums\".\"ArtistId
\" = \"artists\".\"ArtistId\"\r\nORDER BY \"albums\".\"Title\""}, {"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 \r\nFROM customers c \r\nJOIN invoices i ON c.CustomerId
= i.CustomerId \r\nGROUP BY c.Country\r\nORDER BY TotalInvoices DESC"}, {"ro
le": "user", "content": "  \n    List all invoices with a total exceeding $1
0:\n"}, {"role": "assistant", "content": "SELECT * \r\nFROM invoices\r\nWHER
E Total > 10"}, {"role": "user", "content": "  \n    Find all tracks with a
name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "cont
ent": "SELECT * \r\nFROM tracks\r\nWHERE Name LIKE '%what%'\r\nORDER BY Nam
e"}, {"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?"}, {"rol
e": "assistant", "content": "SELECT name FROM sqlite_schema WHERE type='tabl
e'"}, {"role": "user", "content": "  \n    List all employees and their repo
rting manager's name (if any):\n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:27:43.891947555Z', 'mess
age': {'role': 'assistant', 'content': '```sql\nSELECT e.LastName, r.Name \r
\nFROM employees e\r\nLEFT JOIN employees r ON e.ReportsTo = r.EmployeeId;
\r\n``` \n\n\n**Explanation:**\n\n* **`SELECT e.LastName, r.Name`**: This se
lects the employee\'s last name (`e.LastName`) and the manager\'s name (`r.N
ame`). We use a LEFT JOIN to get all employees and optionally their manage
r’s name.\r\n * **`FROM employees e`**:  Starts by selecting from the `emplo
yees` table (aliased as `e`). \r\n* **`LEFT JOIN employees r ON e.ReportsTo
= r.EmployeeId;`**: Performs a LEFT JOIN with the `employees` table (`r`) on
the condition that the `e.ReportsTo` column matches the `r.EmployeeId` of an
other employee. The use of  \'ON\' defines the joining condition, and the `L
EFT JOIN` ensures all employees are included in the result set, even if they
don\'t have a manager assigned.\r\n\r\nThis will give you the list of employ
ees and their corresponding manager names if they exist. If an employee does
n\'t have a reporting manager, "null" or blank would be shown for `r.Name`.
\r\n\n\n\nLet me know if you need to work with other conditions! '}, 'done_r
eason': 'stop', 'done': True, 'total_duration': 44557858049, 'load_duratio
n': 17159741, 'prompt_eval_count': 1635, 'prompt_eval_duration': 2072219100
0, 'eval_count': 278, 'eval_duration': 22541655000}
LLM Response: ```sql
SELECT e.LastName, r.Name
FROM employees e
LEFT JOIN employees r ON e.ReportsTo = r.EmployeeId;
```

Explanation:

* **`SELECT e.LastName, r.Name`**: This selects the employee's last name (`
e.LastName`) and the manager's name (`r.Name`). We use a LEFT JOIN to get al
l employees and optionally their manager’s name.
 * **`FROM employees e`**: Starts by selecting from the `employees` table
(aliased as `e`).
* **`LEFT JOIN employees r ON e.ReportsTo = r.EmployeeId;`**: Performs a LEF
T JOIN with the `employees` table (`r`) on the condition that the `e.Reports
To` column matches the `r.EmployeeId` of another employee. The use of 'ON'
defines the joining condition, and the `LEFT JOIN` ensures all employees are
included in the result set, even if they don't have a manager assigned.

This will give you the list of employees and their corresponding manager nam
es if they exist. If an employee doesn't have a reporting manager, "null" or
```
```

```
blank would be shown for `r.Name`.



Let me know if you need to work with other conditions!
Info: Output from LLM: ```sql
SELECT e.LastName, r.Name
FROM employees e
LEFT JOIN employees r ON e.ReportsTo = r.EmployeeId;
```

Explanation:

* **`SELECT e.LastName, r.Name`**: This selects the employee's last name (`
e.LastName`) and the manager's name (`r.Name`). We use a LEFT JOIN to get al
l employees and optionally their manager’s name.
 * **`FROM employees e`**: Starts by selecting from the `employees` table
(aliased as `e`).
* **`LEFT JOIN employees r ON e.ReportsTo = r.EmployeeId;`**: Performs a LEF
T JOIN with the `employees` table (`r`) on the condition that the `e.Reports
To` column matches the `r.EmployeeId` of another employee. The use of 'ON'
defines the joining condition, and the `LEFT JOIN` ensures all employees are
included in the result set, even if they don't have a manager assigned.

This will give you the list of employees and their corresponding manager nam
es if they exist. If an employee doesn't have a reporting manager, "null" or
blank would be shown for `r.Name`.

Let me know if you need to work with other conditions!
Extracted SQL: SELECT e.LastName, r.Name
FROM employees e
LEFT JOIN employees r ON e.ReportsTo = r.EmployeeId
SELECT e.LastName, r.Name
FROM employees e
LEFT JOIN employees r ON e.ReportsTo = r.EmployeeId
Couldn't run sql: Execution failed on sql 'SELECT e.LastName, r.Name
FROM employees e
LEFT JOIN employees r ON e.ReportsTo = r.EmployeeId': no such column: r.Name

In [27]:
```python
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 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 sqlite_stat1(tbl,idx,stat)\n\nCREATE INDEX IFK_CustomerSupportRe
pId ON "customers" (SupportRepId)\n\nCREATE TABLE "customers"\r\n(\r\n    Cu
stomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    FirstName NVARCH
AR(40)  NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Company NVA
RCHAR(80),\r\n    Address NVARCHAR(70),\r\n    City NVARCHAR(40),\r\n    Sta
te NVARCHAR(40),\r\n    Country NVARCHAR(40),\r\n    PostalCode NVARCHAR(1
0),\r\n    Phone NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHA
R(60)  NOT NULL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRep
Id) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employees" (Repor
tsTo)\n\nCREATE TABLE "employees"\r\n(\r\n    EmployeeId INTEGER PRIMARY KEY
AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    First
Name NVARCHAR(20)  NOT NULL,\r\n    Title NVARCHAR(30),\r\n    ReportsTo INT
EGER,\r\n    BirthDate DATETIME,\r\n    HireDate DATETIME,\r\n    Address NV
ARCHAR(70),\r\n    City NVARCHAR(40),\r\n    State NVARCHAR(40),\r\n    Coun
try NVARCHAR(40),\r\n    PostalCode NVARCHAR(10),\r\n    Phone NVARCHAR(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\n\n===Additional Context \n\nIn the chinook datab
ase invoice means order\n\n===Response Guidelines \n1. If the provided conte
xt is sufficient, please generate a valid SQL query without any explanations
for the question. \n2. If the provided context is almost sufficient but 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': '  \n    Get the total number of invoices for ea
ch customer\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.Las
tName, COUNT(i.InvoiceId) AS TotalInvoices\r\nFROM customers c\r\nJOIN invoi
ces i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName'},
{'role': 'user', 'content': '  \n    Find all invoices since 2010 and the to
tal amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT i.Invoice
Id, SUM(i.Total) AS TotalAmount\r\nFROM invoices i\r\nWHERE i.InvoiceDate >=
'2010-01-01'\r\nGROUP BY i.InvoiceId"}, {'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 \r\nFROM cus
tomers c \r\nJOIN invoices i ON c.CustomerId = i.CustomerId \r\nGROUP BY c.C
ountry\r\nORDER BY TotalInvoices DESC'}, {'role': 'user', 'content': '  \n
List all invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'co
ntent': 'SELECT * \r\nFROM invoices\r\nWHERE Total > 10'}, {'role': 'user',
'content': 'How many customers are there'}, {'role': 'assistant', 'content':
'SELECT COUNT(*) FROM customers'}, {'role': 'user', 'content': 'what are the
top 5 countries that customers come from?'}, {'role': 'assistant', 'conten
t': 'SELECT Country, COUNT(DISTINCT CustomerId) AS NumCustomers\r\nFROM cust
omers\r\nGROUP BY Country\r\nORDER BY NumCustomers DESC\r\nLIMIT 5'}, {'rol
e': 'user', 'content': '  \n    Find all tracks with a name containing "Wha
t" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * \r\nFR
OM tracks\r\nWHERE Name LIKE '%what%'\r\nORDER BY Name"}, {'role': 'user',
'content': '  \n    List all albums and their corresponding artist names
\n'}, {'role': 'assistant', 'content': 'SELECT "albums".Title, "artists".Nam
e \r\nFROM "albums"\r\nJOIN "artists" ON "albums"."ArtistId" = "artists"."Ar
tistId"\r\nORDER BY "albums"."Title"'}, {'role': 'user', 'content': 'Can you
list all tables in the SQLite database catalog?'}, {'role': 'assistant', 'co
ntent': "SELECT name FROM sqlite_schema WHERE type='table'"}, {'role': 'use
r', 'content': '  \n    Get the average invoice total for each custome
r:\n'}]
Info: Ollama parameters:
model=gemma2:2b,
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\r\nFROM customers c\r
\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.
LastName"}, {"role": "user", "content": "  \n    Find all invoices since 201
0 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELE
CT i.InvoiceId, SUM(i.Total) AS TotalAmount\r\nFROM invoices i\r\nWHERE i.In
voiceDate >= '2010-01-01'\r\nGROUP BY i.InvoiceId"}, {"role": "user", "conte
nt": "  \n    Find the total number of invoices per country:\n"}, {"role":
"assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoic
es \r\nFROM customers c \r\nJOIN invoices i ON c.CustomerId = i.CustomerId
\r\nGROUP BY c.Country\r\nORDER BY TotalInvoices DESC"}, {"role": "user", "c
ontent": "  \n    List all invoices with a total exceeding $10:\n"}, {"rol
e": "assistant", "content": "SELECT * \r\nFROM invoices\r\nWHERE Total > 1
0"}, {"role": "user", "content": "How many customers are there"}, {"role":
"assistant", "content": "SELECT COUNT(*) FROM customers"}, {"role": "user",
"content": "what are the top 5 countries that customers come from?"}, {"rol
e": "assistant", "content": "SELECT Country, COUNT(DISTINCT CustomerId) AS N
umCustomers\r\nFROM customers\r\nGROUP BY Country\r\nORDER BY NumCustomers D
ESC\r\nLIMIT 5"}, {"role": "user", "content": "  \n    Find all tracks with
a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "co
ntent": "SELECT * \r\nFROM tracks\r\nWHERE Name LIKE '%what%'\r\nORDER BY Na
me"}, {"role": "user", "content": "  \n    List all albums and their corresp
onding artist names  \n"}, {"role": "assistant", "content": "SELECT \"albums
\".Title, \"artists\".Name \r\nFROM \"albums\"\r\nJOIN \"artists\" ON \"albu
ms\".\"ArtistId\" = \"artists\".\"ArtistId\"\r\nORDER BY \"albums\".\"Title
\""}, {"role": "user", "content": "Can you list all tables in the SQLite dat
abase catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_
schema WHERE type='table'"}, {"role": "user", "content": "  \n    Get the av
erage invoice total for each customer:\n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:28:15.931584292Z', 'mess
age': {'role': 'assistant', 'content': "```sql\nSELECT c.FirstName, c.LastNa
me, AVG(i.Total) AS AvgInvoiceTotal \r\nFROM customers c\r\nJOIN invoices i
ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName;\n``` \n
\nThis query does the following:\n1. **Joins:** It joins the `customers` tab
le with the `invoices` table based on the common field `CustomerId`. This is
done to get invoice total for each customer.  \n2. **Calculates Average:** T
he `AVG(i.Total)` function calculates the average of the `Total` column in t
```

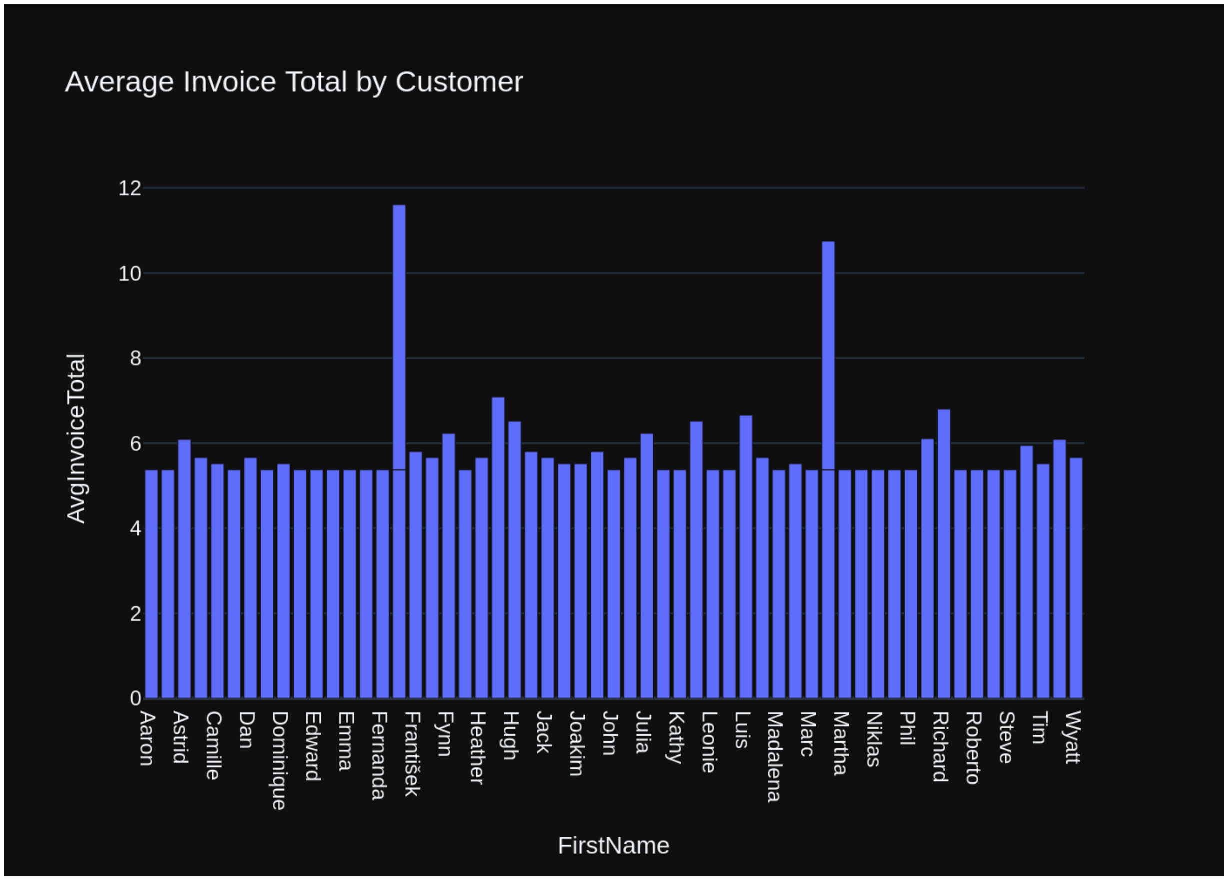
```
he `invoices` table for each customer. \n3. **Groups:** It groups the result
s by customer's first name and last name using `GROUP BY`.  \n\n\n\n\nLet me
know if you would like to explore different types of queries or analysis wit
h this data! "}, 'done_reason': 'stop', 'done': True, 'total_duration': 3197
8783614, 'load_duration': 24105843, 'prompt_eval_count': 1399, 'prompt_eval_
duration': 15994409000, 'eval_count': 186, 'eval_duration': 14776946000}
LLM Response: ```sql
SELECT c.FirstName, c.LastName, AVG(i.Total) AS AvgInvoiceTotal
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName;
```

This query does the following:
1. **Joins:** It joins the `customers` table with the `invoices` table based
on the common field `CustomerId`. This is done to get invoice total for each
customer.
2. **Calculates Average:** The `AVG(i.Total)` function calculates the averag
e of the `Total` column in the `invoices` table for each customer.
3. **Groups:** It groups the results by customer's first name and last name
using `GROUP BY`.

Let me know if you would like to explore different types of queries or analy
sis with this data!
Info: Output from LLM: ```sql
SELECT c.FirstName, c.LastName, AVG(i.Total) AS AvgInvoiceTotal
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName;
```

This query does the following:
1. **Joins:** It joins the `customers` table with the `invoices` table based
on the common field `CustomerId`. This is done to get invoice total for each
customer.
2. **Calculates Average:** The `AVG(i.Total)` function calculates the averag
e of the `Total` column in the `invoices` table for each customer.
3. **Groups:** It groups the results by customer's first name and last name
using `GROUP BY`.

Let me know if you would like to explore different types of queries or analy
sis with this data!
Extracted SQL: SELECT c.FirstName, c.LastName, AVG(i.Total) AS AvgInvoiceTot
al
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName
SELECT c.FirstName, c.LastName, AVG(i.Total) AS AvgInvoiceTotal
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
```

```
GROUP BY c.FirstName, c.LastName
    FirstName      LastName  AvgInvoiceTotal
0       Aaron      Mitchell         5.374286
1   Alexandre         Rocha         5.374286
2      Astrid        Gruber         6.088571
3       Bjørn        Hansen         5.660000
4     Camille       Bernard         5.517143
5        Daan       Peeters         5.374286
6         Dan        Miller         5.660000
7       Diego     Gutiérrez         5.374286
8   Dominique      Lefebvre         5.517143
9     Eduardo       Martins         5.374286
10     Edward       Francis         5.374286
11      Ellie      Sullivan         5.374286
12       Emma         Jones         5.374286
13   Enrique          Muñoz         5.374286
14   Fernanda         Ramos         5.374286
15      Frank        Harris         5.374286
16      Frank       Ralston         6.231429
17  František   Wichterlová         5.802857
18   François      Tremblay         5.660000
19       Fynn    Zimmermann         6.231429
20     Hannah     Schneider         5.374286
21    Heather       Leacock         5.660000
22     Helena          Holý         7.088571
23       Hugh      O'Reilly         6.517143
24   Isabelle       Mercier         5.802857
25       Jack         Smith         5.660000
26   Jennifer      Peterson         5.517143
27     Joakim     Johansson         5.517143
28   Johannes  Van der Berg         5.802857
29       John        Gordon         5.374286
30       João     Fernandes         5.660000
31      Julia       Barnett         6.231429
32       Kara       Nielsen         5.374286
33      Kathy         Chase         5.374286
34   Ladislav        Kovács         6.517143
35     Leonie        Köhler         5.374286
36      Lucas       Mancini         5.374286
37       Luis         Rojas         6.660000
38       Luís     Gonçalves         5.660000
39   Madalena       Sampaio         5.374286
40      Manoj        Pareek         5.517143
41       Marc        Dubois         5.374286
42       Mark       Philips         5.374286
43       Mark        Taylor         5.374286
44     Martha          Silk         5.374286
45   Michelle        Brooks         5.374286
46    Niklas       Schröder         5.374286
47    Patrick          Gray         5.374286
48       Phil        Hughes         5.374286
49       Puja    Srivastava         6.106667
50    Richard    Cunningham         6.802857
51     Robert         Brown         5.374286
52    Roberto       Almeida         5.374286
53  Stanisław        Wójcik         5.374286
```

```
54      Steve        Murray         5.374286
55      Terhi    Hämäläinen         5.945714
56        Tim         Goyer         5.517143
57     Victor       Stevens         6.088571
58      Wyatt        Girard         5.660000
Info: Ollama parameters:
model=gemma2:2b,
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 AvgInvoiceTotal \r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = 
i.CustomerId\r\nGROUP BY c.FirstName, c.LastName\n\nThe following is informa
tion about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n
FirstName           object\nLastName            object\nAvgInvoiceTotal    f
loat64\ndtype: object"}, {"role": "user", "content": "Can you generate the P
ython 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 datafra
me, use an Indicator. Respond with only Python code. Do not answer with any 
explanations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:28:23.905777533Z', 'mess
age': {'role': 'assistant', 'content': '```python\nimport plotly.express as 
px\n\nfig = px.bar(df, x="FirstName", y="AvgInvoiceTotal", title="Average In
voice Total by Customer")\n# fig = px.indicator(df, v_y=\'AvgInvoiceTotal\',
title="Average Invoice Total by Customer") \nplotly.offline.plot(fig)\n``
`'}, 'done_reason': 'stop', 'done': True, 'total_duration': 7947100937, 'loa
d_duration': 26010589, 'prompt_eval_count': 206, 'prompt_eval_duration': 234
2717000, 'eval_count': 75, 'eval_duration': 5491775000}
```


Average Invoice Total by Customer
AvgInvoiceTotal
0
2
4
6
8
10
12
FirstName
Aaron
Astrid
Camille
Dan
Dominique
Edward
Emma
Fernanda
František
Fynn
Heather
Hugh
Jack
Joakim
John
Julia
Kathy
Leonie
Luis
Madalena
Marc
Martha
Niklas
Phil
Richard
Roberto
Steve
Tim
Wyatt

```
Out[27]: ('SELECT c.FirstName, c.LastName, AVG(i.Total) AS AvgInvoiceTotal \r\nFROM
         customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY
         c.FirstName, c.LastName',
              FirstName      LastName  AvgInvoiceTotal
          0       Aaron      Mitchell         5.374286
          1   Alexandre         Rocha         5.374286
          2      Astrid        Gruber         6.088571
          3       Bjørn        Hansen         5.660000
          4     Camille       Bernard         5.517143
          5        Daan       Peeters         5.374286
          6         Dan        Miller         5.660000
          7       Diego     Gutiérrez         5.374286
          8   Dominique      Lefebvre         5.517143
          9     Eduardo       Martins         5.374286
          10     Edward       Francis         5.374286
          11      Ellie      Sullivan         5.374286
          12       Emma         Jones         5.374286
          13   Enrique          Muñoz         5.374286
          14   Fernanda         Ramos         5.374286
          15      Frank        Harris         5.374286
          16      Frank       Ralston         6.231429
          17  František   Wichterlová         5.802857
          18   François      Tremblay         5.660000
          19       Fynn    Zimmermann         6.231429
          20     Hannah     Schneider         5.374286
          21    Heather       Leacock         5.660000
          22     Helena          Holý         7.088571
          23       Hugh      O'Reilly         6.517143
          24   Isabelle       Mercier         5.802857
          25       Jack         Smith         5.660000
          26   Jennifer      Peterson         5.517143
          27     Joakim     Johansson         5.517143
          28   Johannes  Van der Berg         5.802857
          29       John        Gordon         5.374286
          30       João     Fernandes         5.660000
          31      Julia       Barnett         6.231429
          32       Kara       Nielsen         5.374286
          33      Kathy         Chase         5.374286
          34   Ladislav        Kovács         6.517143
          35     Leonie        Köhler         5.374286
          36      Lucas       Mancini         5.374286
          37       Luis         Rojas         6.660000
          38       Luís     Gonçalves         5.660000
          39   Madalena       Sampaio         5.374286
          40      Manoj        Pareek         5.517143
          41       Marc        Dubois         5.374286
          42       Mark       Philips         5.374286
          43       Mark        Taylor         5.374286
          44     Martha          Silk         5.374286
          45   Michelle        Brooks         5.374286
          46     Niklas       Schröder         5.374286
          47    Patrick          Gray         5.374286
          48       Phil        Hughes         5.374286
          49       Puja    Srivastava         6.106667
          50    Richard    Cunningham         6.802857
          51     Robert         Brown         5.374286
```

```
 52    Roberto       Almeida         5.374286
 53  Stanisław        Wójcik         5.374286
 54      Steve        Murray         5.374286
 55      Terhi    Hämäläinen         5.945714
 56        Tim         Goyer         5.517143
 57     Victor       Stevens         6.088571
 58      Wyatt        Girard         5.660000,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'FirstName=%{x}<br>AvgInvoiceTotal=%{y}<ext
ra></extra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Aaron', 'Alexandre', 'Astrid', 'Bjørn', 'Camill
e', 'Daan', 'Dan',
                           'Diego', 'Dominique', 'Eduardo', 'Edward', 'Elli
e', 'Emma', 'Enrique',
                           'Fernanda', 'Frank', 'Frank', 'František', 'Fran
çois', 'Fynn', 'Hannah',
                           'Heather', 'Helena', 'Hugh', 'Isabelle', 'Jack',
'Jennifer', 'Joakim',
                           'Johannes', 'John', 'João', 'Julia', 'Kara', 'Ka
thy', 'Ladislav',
                           'Leonie', 'Lucas', 'Luis', 'Luís', 'Madalena',
'Manoj', 'Marc', 'Mark',
                           'Mark', 'Martha', 'Michelle', 'Niklas', 'Patric
k', 'Phil', 'Puja',
                           'Richard', 'Robert', 'Roberto', 'Stanisław', 'St
eve', 'Terhi', 'Tim',
                           'Victor', 'Wyatt'], dtype=object),
               'xaxis': 'x',
               'y': array([5.37428571, 5.37428571, 6.08857143, 5.66      ,
5.51714286, 5.37428571,
                           5.66      , 5.37428571, 5.51714286, 5.37428571,
5.37428571, 5.37428571,
                           5.37428571, 5.37428571, 5.37428571, 5.37428571,
6.23142857, 5.80285714,
                           5.66      , 6.23142857, 5.37428571, 5.66      ,
7.08857143, 6.51714286,
                           5.80285714, 5.66      , 5.51714286, 5.51714286,
5.80285714, 5.37428571,
                           5.66      , 6.23142857, 5.37428571, 5.37428571,
6.51714286, 5.37428571,
                           5.37428571, 6.66      , 5.66      , 5.37428571,
5.51714286, 5.37428571,
                           5.37428571, 5.37428571, 5.37428571, 5.37428571,
5.37428571, 5.37428571,
                           5.37428571, 6.10666667, 6.80285714, 5.37428571,
5.37428571, 5.37428571,
                           5.37428571, 5.94571429, 5.51714286, 6.08857143,
5.66      ]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                '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': 'AvgInvoiceTotal'}}}
 }))
```

In [28]:

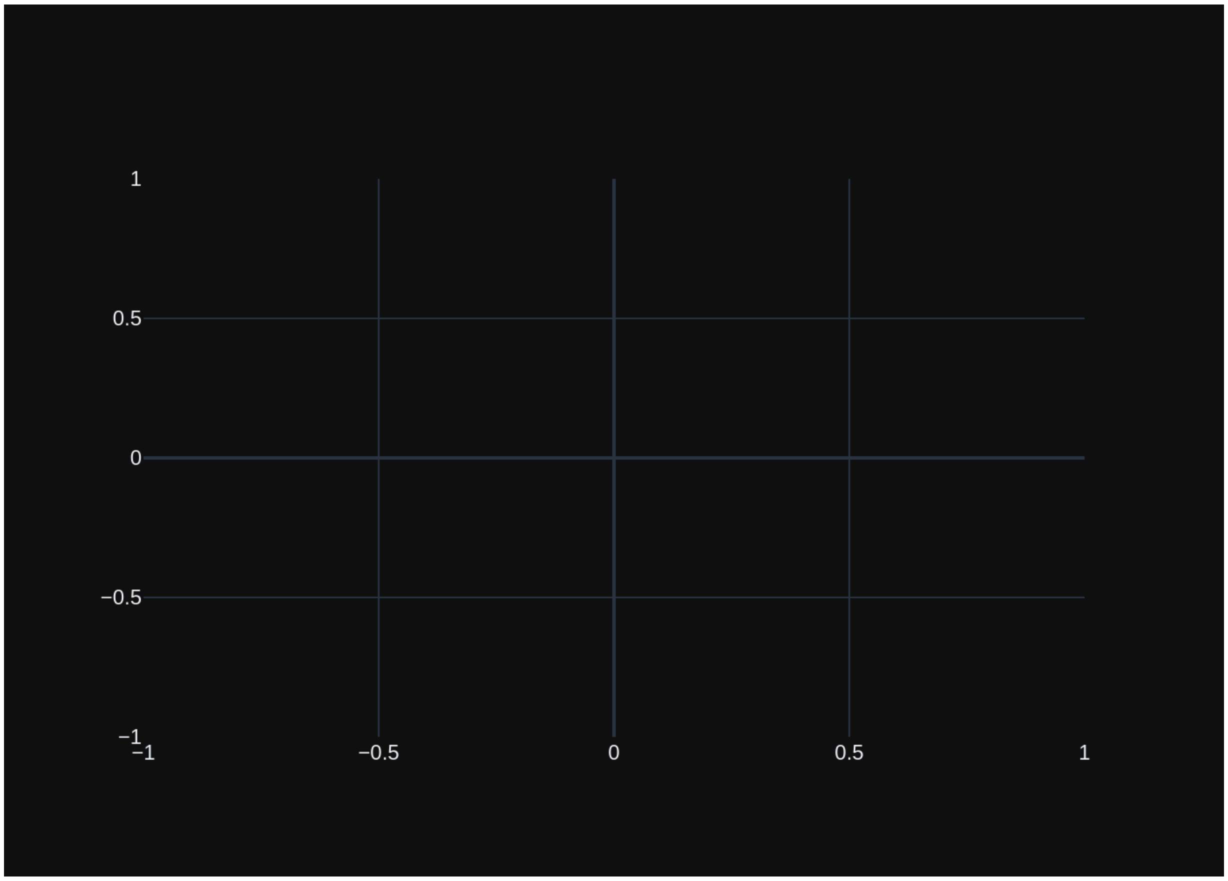
```
question = """
    Find the top 5 most expensive tracks (based on unit price):
"""

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 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_TrackAlbumId ON "tracks"
(AlbumId)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (GenreId)\n\nCREATE I
NDEX IFK_PlaylistTrackTrackId ON "playlist_track" (TrackId)\n\nCREATE INDEX
IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE INDEX IFK_Trac
kMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE TABLE "invoice_items"\r\n
(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    In
voiceId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPric
e 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 ACTIO
N ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (Tra
ckId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE
"playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId IN
TEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistI
d, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Playlis
tId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (Tr
ackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE N
O ACTION\r\n)\n\nCREATE INDEX IFK_AlbumArtistId ON "albums" (ArtistId)\n\nCR
EATE TABLE "albums"\r\n(\r\n    AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NO
T NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n    ArtistId INTEGER  NOT N
ULL,\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 gene
rated. \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 g
iven before. \n'}, {'role': 'user', 'content': '  \n    Find all tracks with
a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'cont
ent': "SELECT * \r\nFROM tracks\r\nWHERE Name LIKE '%what%'\r\nORDER BY Nam
e"}, {'role': 'user', 'content': '  \n    List all invoices with a total exc
eeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \r\nFROM invoice
s\r\nWHERE Total > 10'}, {'role': 'user', 'content': '  \n    List all album
s and their corresponding artist names  \n'}, {'role': 'assistant', 'conten
t': 'SELECT "albums".Title, "artists".Name \r\nFROM "albums"\r\nJOIN "artist
s" ON "albums"."ArtistId" = "artists"."ArtistId"\r\nORDER BY "albums"."Titl
e"'}, {'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 AvgInvoiceTotal \r\nFROM customers c\r\nJOIN invoi
ces i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName'},
{'role': 'user', 'content': '  \n    Find all invoices since 2010 and the to
```

```
tal amount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT i.Invoice
Id, SUM(i.Total) AS TotalAmount\r\nFROM invoices i\r\nWHERE i.InvoiceDate >=
'2010-01-01'\r\nGROUP BY i.InvoiceId"}, {'role': 'user', 'content': 'what ar
e the top 5 countries that customers come from?'}, {'role': 'assistant', 'co
ntent': 'SELECT Country, COUNT(DISTINCT CustomerId) AS NumCustomers\r\nFROM
customers\r\nGROUP BY Country\r\nORDER BY NumCustomers DESC\r\nLIMIT 5'},
{'role': 'user', 'content': '  \n    Find the total number of invoices per c
ountry:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(i.Inv
oiceId) AS TotalInvoices \r\nFROM customers c \r\nJOIN invoices i ON c.Custo
merId = i.CustomerId \r\nGROUP BY c.Country\r\nORDER BY TotalInvoices DES
C'}, {'role': 'user', 'content': '  \n    Get the total number of invoices f
or each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName,
c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\r\nFROM customers c\r\nJOIN
invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastNam
e'}, {'role': 'user', 'content': 'Can you list all tables in the SQLite data
base catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_s
chema WHERE type='table'"}, {'role': 'user', 'content': 'How many customers
are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customer
s'}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive trac
ks (based on unit price):\n'}]
Info: Ollama parameters:
model=gemma2:2b,
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 * \r\nFROM tracks\r\nWHERE Name LIKE '%wha
t%'\r\nORDER BY Name"}, {"role": "user", "content": "  \n    List all invoic
es with a total exceeding $10:\n"}, {"role": "assistant", "content": "SELECT
* \r\nFROM invoices\r\nWHERE Total > 10"}, {"role": "user", "content": "  \n
List all albums and their corresponding artist names  \n"}, {"role": "assist
ant", "content": "SELECT \"albums\".Title, \"artists\".Name \r\nFROM \"album
s\"\r\nJOIN \"artists\" ON \"albums\".\"ArtistId\" = \"artists\".\"ArtistId
\"\r\nORDER BY \"albums\".\"Title\""}, {"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 AvgInvoiceTotal
\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGR
OUP BY c.FirstName, c.LastName"}, {"role": "user", "content": "  \n    Find
all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assist
ant", "content": "SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\r\nFROM in
voices i\r\nWHERE i.InvoiceDate >= '2010-01-01'\r\nGROUP BY i.InvoiceId"},
{"role": "user", "content": "what are the top 5 countries that customers com
e from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(DISTINCT
CustomerId) AS NumCustomers\r\nFROM customers\r\nGROUP BY Country\r\nORDER B
Y NumCustomers DESC\r\nLIMIT 5"}, {"role": "user", "content": "  \n    Find
the total number of invoices per country:\n"}, {"role": "assistant", "conten
t": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices \r\nFROM customer
s c \r\nJOIN invoices i ON c.CustomerId = i.CustomerId \r\nGROUP BY c.Countr
y\r\nORDER BY TotalInvoices DESC"}, {"role": "user", "content": "  \n    Get
the total number of invoices for each customer\n"}, {"role": "assistant", "c
ontent": "SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoice
s\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nG
ROUP BY c.FirstName, c.LastName"}, {"role": "user", "content": "Can you list
all tables in the SQLite database catalog?"}, {"role": "assistant", "conten
t": "SELECT name FROM sqlite_schema WHERE type='table'"}, {"role": "user",
"content": "How many customers are there"}, {"role": "assistant", "content":
"SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "  \n    Find
the top 5 most expensive tracks (based on unit price):\n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:28:44.090001054Z', 'mess
age': {'role': 'assistant', 'content': 'SELECT TrackId, UnitPrice\r\nFROM tr
acks\r\nORDER BY UnitPrice DESC\r\nLIMIT 5; \r\n'}, 'done_reason': 'stop',
'done': True, 'total_duration': 19951877280, 'load_duration': 22088700, 'pro
mpt_eval_count': 1364, 'prompt_eval_duration': 16563810000, 'eval_count': 2
6, 'eval_duration': 1990082000}
LLM Response: SELECT TrackId, UnitPrice
FROM tracks
ORDER BY UnitPrice DESC
LIMIT 5;
```

```
Info: Output from LLM: SELECT TrackId, UnitPrice
FROM tracks
ORDER BY UnitPrice DESC
LIMIT 5;

Extracted SQL: SELECT TrackId, UnitPrice
FROM tracks
ORDER BY UnitPrice DESC
LIMIT 5
SELECT TrackId, UnitPrice
FROM tracks
ORDER BY UnitPrice DESC
LIMIT 5
   TrackId  UnitPrice
0     2819       1.99
1     2820       1.99
2     2821       1.99
3     2822       1.99
4     2823       1.99
Info: Ollama parameters:
model=gemma2:2b,
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 TrackId, UnitPrice\r\nFROM t
racks\r\nORDER BY UnitPrice DESC\r\nLIMIT 5\n\nThe following is information
about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Trac
kId        int64\nUnitPrice    float64\ndtype: object"}, {"role": "user", "c
ontent": "Can you generate the Python plotly code to chart the results of th
e 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 Pyth
on code. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:28:49.070258499Z', 'mess
age': {'role': 'assistant', 'content': "```python\nimport plotly.graph_objec
ts as go\n\nfig = go.Figure(data=go.Scatter(x=[0], y=[0], mode='lines')) \n`
``"}, 'done_reason': 'stop', 'done': True, 'total_duration': 4955888590, 'lo
ad_duration': 23577990, 'prompt_eval_count': 182, 'prompt_eval_duration': 19
87967000, 'eval_count': 40, 'eval_duration': 2854117000}
```



```
Out[28]: ('SELECT TrackId, UnitPrice\r\nFROM tracks\r\nORDER BY UnitPrice DESC\r\nLI
         MIT 5',
             TrackId  UnitPrice
          0     2819       1.99
          1     2820       1.99
          2     2821       1.99
          3     2822       1.99
          4     2823       1.99,
          Figure({
              'data': [{'mode': 'lines', 'type': 'scatter', 'x': [0], 'y': [0]}], 'l
         ayout': {'template': '...'}
          }))
```

```
In [29]: question = """
             List all genres and the number of tracks in each genre:
         """

         vn.ask(question=question)
```

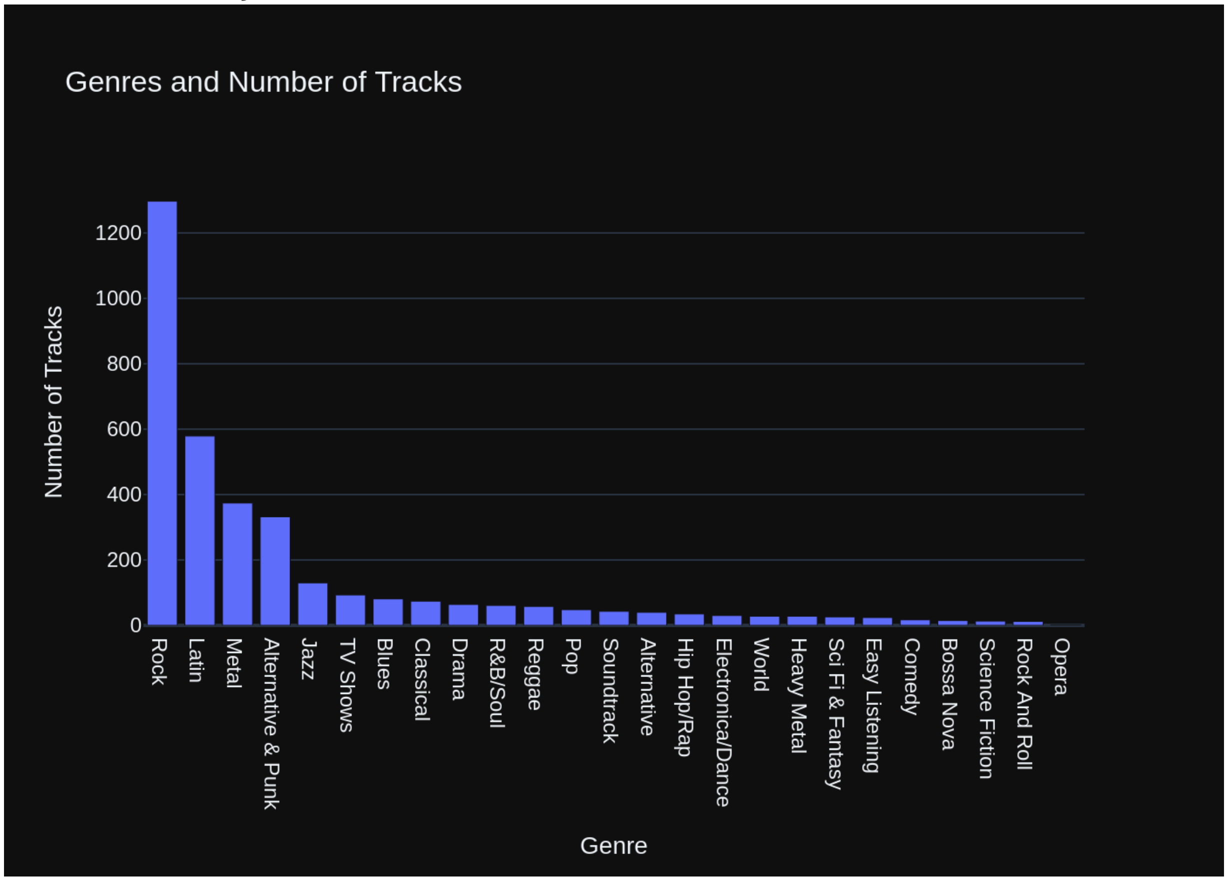
```
Number of requested results 10 is greater than number of elements in index
1, updating n_results = 1
```

```
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 TrackId, UnitPrice\r\nFROM tracks\r\nORDER BY UnitPrice DESC\r
\nLIMIT 5'}, {'role': 'user', 'content': '  \n    List all albums and their
corresponding artist names  \n'}, {'role': 'assistant', 'content': 'SELECT
"albums".Title, "artists".Name \r\nFROM "albums"\r\nJOIN "artists" ON "album
s"."ArtistId" = "artists"."ArtistId"\r\nORDER BY "albums"."Title"'}, {'rol
e': 'user', 'content': '  \n    Find all tracks with a name containing "Wha
t" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * \r\nFR
OM tracks\r\nWHERE Name LIKE '%what%'\r\nORDER BY Name"}, {'role': 'user',
'content': 'what are the top 5 countries that customers come from?'}, {'rol
e': 'assistant', 'content': 'SELECT Country, COUNT(DISTINCT CustomerId) AS N
umCustomers\r\nFROM customers\r\nGROUP BY Country\r\nORDER BY NumCustomers D
ESC\r\nLIMIT 5'}, {'role': 'user', 'content': '  \n    Find all invoices sin
ce 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content':
"SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\r\nFROM invoices i\r\nWHERE
i.InvoiceDate >= '2010-01-01'\r\nGROUP BY i.InvoiceId"}, {'role': 'user', 'c
ontent': '  \n    Find the total number of invoices per country:\n'}, {'rol
```

```
e': 'assistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalIn
voices \r\nFROM customers c \r\nJOIN invoices i ON c.CustomerId = i.Customer
Id \r\nGROUP BY c.Country\r\nORDER BY TotalInvoices DESC'}, {'role': 'user',
'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'rol
e': 'assistant', 'content': 'SELECT * \r\nFROM invoices\r\nWHERE Total > 1
0'}, {'role': 'user', 'content': '  \n    Get the total number of invoices f
or each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName,
c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\r\nFROM customers c\r\nJOIN
invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastNam
e'}, {'role': 'user', 'content': 'Can you list all tables in the SQLite data
base catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sqlite_s
chema WHERE type='table'"}, {'role': 'user', 'content': 'How many customers
are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(*) FROM customer
s'}, {'role': 'user', 'content': '  \n    List all genres and the number of
tracks in each genre:\n'}]
Info: Ollama parameters:
model=gemma2:2b,
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 TrackId, UnitPrice\r\nFROM tracks\r\nORDER BY U
nitPrice DESC\r\nLIMIT 5"}, {"role": "user", "content": "  \n    List all al
bums and their corresponding artist names  \n"}, {"role": "assistant", "cont
ent": "SELECT \"albums\".Title, \"artists\".Name \r\nFROM \"albums\"\r\nJOIN
\"artists\" ON \"albums\".\"ArtistId\" = \"artists\".\"ArtistId\"\r\nORDER B
Y \"albums\".\"Title\""}, {"role": "user", "content": "  \n    Find all trac
ks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assista
nt", "content": "SELECT * \r\nFROM tracks\r\nWHERE Name LIKE '%what%'\r\nORD
ER BY Name"}, {"role": "user", "content": "what are the top 5 countries that
customers come from?"}, {"role": "assistant", "content": "SELECT Country, CO
UNT(DISTINCT CustomerId) AS NumCustomers\r\nFROM customers\r\nGROUP BY Count
ry\r\nORDER BY NumCustomers DESC\r\nLIMIT 5"}, {"role": "user", "content": "
\n    Find all invoices since 2010 and the total amount invoiced:\n"}, {"rol
e": "assistant", "content": "SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount
\r\nFROM invoices i\r\nWHERE i.InvoiceDate >= '2010-01-01'\r\nGROUP BY i.Inv
oiceId"}, {"role": "user", "content": "  \n    Find the total number of invo
ices per country:\n"}, {"role": "assistant", "content": "SELECT c.Country, C
OUNT(i.InvoiceId) AS TotalInvoices \r\nFROM customers c \r\nJOIN invoices i
ON c.CustomerId = i.CustomerId \r\nGROUP BY c.Country\r\nORDER BY TotalInvoi
ces DESC"}, {"role": "user", "content": "  \n    List all invoices with a to
tal exceeding $10:\n"}, {"role": "assistant", "content": "SELECT * \r\nFROM
invoices\r\nWHERE Total > 10"}, {"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\r
\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROU
P BY c.FirstName, c.LastName"}, {"role": "user", "content": "Can you list al
l tables in the SQLite database catalog?"}, {"role": "assistant", "content":
"SELECT name FROM sqlite_schema WHERE type='table'"}, {"role": "user", "cont
ent": "How many customers are there"}, {"role": "assistant", "content": "SEL
ECT COUNT(*) FROM customers"}, {"role": "user", "content": "  \n    List all
genres and the number of tracks in each genre:\n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:29:10.927007442Z', 'mess
age': {'role': 'assistant', 'content': 'SELECT "genres"."Name", COUNT(DISTIN
CT "tracks"."TrackId") AS NumTracks\r\nFROM "genres"\r\nLEFT JOIN "tracks" O
N "genres"."GenreId" = "tracks"."GenreId"\r\nGROUP BY "genres"."Name"\r\nORD
ER BY NumTracks DESC; \r\n'}, 'done_reason': 'stop', 'done': True, 'total_du
ration': 21758688660, 'load_duration': 15238012, 'prompt_eval_count': 1263,
'prompt_eval_duration': 15417103000, 'eval_count': 63, 'eval_duration': 4972
950000}
LLM Response: SELECT "genres"."Name", COUNT(DISTINCT "tracks"."TrackId") AS
NumTracks
FROM "genres"
LEFT JOIN "tracks" ON "genres"."GenreId" = "tracks"."GenreId"
GROUP BY "genres"."Name"
ORDER BY NumTracks DESC;

Info: Output from LLM: SELECT "genres"."Name", COUNT(DISTINCT "tracks"."Trac
kId") AS NumTracks
FROM "genres"
LEFT JOIN "tracks" ON "genres"."GenreId" = "tracks"."GenreId"
GROUP BY "genres"."Name"
```

```
ORDER BY NumTracks DESC;

Extracted SQL: SELECT "genres"."Name", COUNT(DISTINCT "tracks"."TrackId") AS
NumTracks
FROM "genres"
LEFT JOIN "tracks" ON "genres"."GenreId" = "tracks"."GenreId"
GROUP BY "genres"."Name"
ORDER BY NumTracks DESC
SELECT "genres"."Name", COUNT(DISTINCT "tracks"."TrackId") AS NumTracks
FROM "genres"
LEFT JOIN "tracks" ON "genres"."GenreId" = "tracks"."GenreId"
GROUP BY "genres"."Name"
ORDER BY NumTracks DESC
                  Name  NumTracks
0                 Rock       1297
1                Latin        579
2                Metal        374
3   Alternative & Punk        332
4                 Jazz        130
5             TV Shows         93
6                Blues         81
7            Classical         74
8                Drama         64
9             R&B/Soul         61
10              Reggae         58
11                 Pop         48
12          Soundtrack         43
13         Alternative         40
14         Hip Hop/Rap         35
15   Electronica/Dance         30
16               World         28
17         Heavy Metal         28
18    Sci Fi & Fantasy         26
19      Easy Listening         24
20              Comedy         17
21          Bossa Nova         15
22     Science Fiction         13
23       Rock And Roll         12
24               Opera          1
Info: Ollama parameters:
model=gemma2:2b,
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 \"genres\".\"Name\", COUNT(DISTI
NCT \"tracks\".\"TrackId\") AS NumTracks\r\nFROM \"genres\"\r\nLEFT JOIN \"t
racks\" ON \"genres\".\"GenreId\" = \"tracks\".\"GenreId\"\r\nGROUP BY \"gen
res\".\"Name\"\r\nORDER BY NumTracks DESC\n\nThe following is information ab
out the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name
object\nNumTracks     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': 'gemma2:2b', 'created_at': '2024-08-01T18:29:18.434784706Z', 'mess
age': {'role': 'assistant', 'content': '```python\nimport plotly.express as
px\n\nfig = px.bar(df, x="Name", y="NumTracks", title=\'Genres and Number of
Tracks\')\nfig.update_layout(xaxis=dict(title="Genre"), yaxis=dict(title="Nu
mber of Tracks"))\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True,
'total_duration': 7480394376, 'load_duration': 18403999, 'prompt_eval_coun
t': 215, 'prompt_eval_duration': 2473560000, 'eval_count': 69, 'eval_duratio
n': 4897193000}
```



```
Out[29]: ('SELECT "genres"."Name", COUNT(DISTINCT "tracks"."TrackId") AS NumTracks\r
         \nFROM "genres"\r\nLEFT JOIN "tracks" ON "genres"."GenreId" = "tracks"."Gen
         reId"\r\nGROUP BY "genres"."Name"\r\nORDER BY NumTracks DESC',
                            Name  NumTracks
          0                 Rock       1297
          1                Latin        579
          2                Metal        374
          3   Alternative & Punk        332
          4                 Jazz        130
          5             TV Shows         93
          6                Blues         81
          7            Classical         74
          8                Drama         64
          9             R&B/Soul         61
          10              Reggae         58
          11                 Pop         48
          12          Soundtrack         43
          13         Alternative         40
          14         Hip Hop/Rap         35
          15   Electronica/Dance         30
          16               World         28
          17         Heavy Metal         28
          18    Sci Fi & Fantasy         26
          19      Easy Listening         24
          20              Comedy         17
          21          Bossa Nova         15
          22     Science Fiction         13
          23       Rock And Roll         12
          24               Opera          1,
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'Name=%{x}<br>NumTracks=%{y}<extra></extra
         >',
                        'legendgroup': '',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'textposition': 'auto',
                        'type': 'bar',
                        'x': array(['Rock', 'Latin', 'Metal', 'Alternative & Punk',
         'Jazz', 'TV Shows',
                                    'Blues', 'Classical', 'Drama', 'R&B/Soul', 'Regg
         ae', 'Pop',
                                    'Soundtrack', 'Alternative', 'Hip Hop/Rap', 'Ele
         ctronica/Dance',
                                    'World', 'Heavy Metal', 'Sci Fi & Fantasy', 'Eas
         y Listening', 'Comedy',
                                    'Bossa Nova', 'Science Fiction', 'Rock And Rol
         l', 'Opera'], dtype=object),
                        'xaxis': 'x',
                        'y': array([1297,  579,  374,  332,  130,   93,   81,   74,
         64,   61,   58,   48,
                                      43,   40,   35,   30,   28,   28,   26,   24,
         17,   15,   13,   12,
                                       1]),
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                         'legend': {'tracegroupgap': 0},
                         'template': '...',
                         'title': {'text': 'Genres and Number of Tracks'},
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
         ext': 'Genre'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
         ext': 'Number of Tracks'}}}
          }))
```

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 "genres"."Name", COUNT(DISTINCT "tracks"."TrackId") AS NumTracks
\r\nFROM "genres"\r\nLEFT JOIN "tracks" ON "genres"."GenreId" = "tracks"."Ge
nreId"\r\nGROUP BY "genres"."Name"\r\nORDER BY NumTracks DESC'}, {'role': 'u
ser', 'content': '  \n    Find all tracks with a name containing "What" (cas
e-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * \r\nFROM trac
ks\r\nWHERE Name LIKE '%what%'\r\nORDER BY Name"}, {'role': 'user', 'conten
t': '  \n    List all albums and their corresponding artist names  \n'}, {'r
ole': 'assistant', 'content': 'SELECT "albums".Title, "artists".Name \r\nFRO
M "albums"\r\nJOIN "artists" ON "albums"."ArtistId" = "artists"."ArtistId"\r
\nORDER BY "albums"."Title"'}, {'role': 'user', 'content': '  \n    Find the
top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistan
t', 'content': 'SELECT TrackId, UnitPrice\r\nFROM tracks\r\nORDER BY UnitPri
ce DESC\r\nLIMIT 5'}, {'role': 'user', 'content': 'Can you list all tables i
n the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT n
ame FROM sqlite_schema WHERE type='table'"}, {'role': 'user', 'content': '
\n    Find all invoices since 2010 and the total amount invoiced:\n'}, {'rol
```

```
e': 'assistant', 'content': "SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount
\r\nFROM invoices i\r\nWHERE i.InvoiceDate >= '2010-01-01'\r\nGROUP BY i.Inv
oiceId"}, {'role': 'user', 'content': 'what are the top 5 countries that cus
tomers come from?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT
(DISTINCT CustomerId) AS NumCustomers\r\nFROM customers\r\nGROUP BY Country
\r\nORDER BY NumCustomers DESC\r\nLIMIT 5'}, {'role': 'user', 'content': '
\n    List all invoices with a total exceeding $10:\n'}, {'role': 'assistan
t', 'content': 'SELECT * \r\nFROM invoices\r\nWHERE Total > 10'}, {'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 AvgInvoice
Total \r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId
\r\nGROUP BY c.FirstName, c.LastName'}, {'role': 'user', 'content': '  \n
Get all genres that do not have any tracks associated with them:\n'}]
Info: Ollama parameters:
model=gemma2:2b,
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 TABLE \"tracks\"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n
MediaTypeId INTEGER  NOT 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_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 N
VARCHAR(120)\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n    AlbumId INTEGER PRI
MARY KEY AUTOINCREMENT NOT NULL,\r\n    Title NVARCHAR(160)  NOT NULL,\r\n
ArtistId INTEGER  NOT NULL,\r\n    FOREIGN KEY (ArtistId) REFERENCES \"artis
ts\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREA
TE 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 INDEX IFK_AlbumArtistId ON \"albums\"
(ArtistId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n    PlaylistId INTEGER PRIM
ARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n\n===Addit
ional 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": "  \n    Lis
t all genres and the number of tracks in each genre:\n"}, {"role": "assistan
t", "content": "SELECT \"genres\".\"Name\", COUNT(DISTINCT \"tracks\".\"Trac
kId\") AS NumTracks\r\nFROM \"genres\"\r\nLEFT JOIN \"tracks\" ON \"genres
\".\"GenreId\" = \"tracks\".\"GenreId\"\r\nGROUP BY \"genres\".\"Name\"\r\nO
RDER BY NumTracks DESC"}, {"role": "user", "content": "  \n    Find all trac
ks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assista
nt", "content": "SELECT * \r\nFROM tracks\r\nWHERE Name LIKE '%what%'\r\nORD
ER BY Name"}, {"role": "user", "content": "  \n    List all albums and their
corresponding artist names  \n"}, {"role": "assistant", "content": "SELECT
\"albums\".Title, \"artists\".Name \r\nFROM \"albums\"\r\nJOIN \"artists\" O
N \"albums\".\"ArtistId\" = \"artists\".\"ArtistId\"\r\nORDER BY \"albums
\".\"Title\""}, {"role": "user", "content": "  \n    Find the top 5 most exp
ensive tracks (based on unit price):\n"}, {"role": "assistant", "content":
"SELECT TrackId, UnitPrice\r\nFROM tracks\r\nORDER BY UnitPrice DESC\r\nLIMI
T 5"}, {"role": "user", "content": "Can you list all tables in the SQLite da
tabase catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite
_schema WHERE type='table'"}, {"role": "user", "content": "  \n    Find all
invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistan
t", "content": "SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\r\nFROM invo
ices i\r\nWHERE i.InvoiceDate >= '2010-01-01'\r\nGROUP BY i.InvoiceId"}, {"r
ole": "user", "content": "what are the top 5 countries that customers come f
rom?"}, {"role": "assistant", "content": "SELECT Country, COUNT(DISTINCT Cus
tomerId) AS NumCustomers\r\nFROM customers\r\nGROUP BY Country\r\nORDER BY N
umCustomers DESC\r\nLIMIT 5"}, {"role": "user", "content": "  \n    List all
invoices with a total exceeding $10:\n"}, {"role": "assistant", "content":
"SELECT * \r\nFROM invoices\r\nWHERE Total > 10"}, {"role": "user", "conten
t": "How many customers are there"}, {"role": "assistant", "content": "SELEC
T COUNT(*) FROM customers"}, {"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 AvgInvoiceTotal \r\nFROM cu
stomers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.Fi
rstName, c.LastName"}, {"role": "user", "content": "  \n    Get all genres t
hat do not have any tracks associated with them:\n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:29:46.687214432Z', 'mess
age': {'role': 'assistant', 'content': "```sql\nSELECT * \r\nFROM genres\r\n
WHERE COUNT(*) = 0; \r\n``` \n \n**Explanation:**\n\n* **`SELECT *`**: This
selects all columns from the table.\n* **`FROM genres`**: We are selecting d
ata from the `genres` table.\n* **`WHERE COUNT(*) = 0`**:  This clause filte
rs for rows where the count of all records in the 'genres' table is equal to
zero. The 'COUNT(*)' function counts all rows, and an empty result means the
re were no entries.\n\n\nLet me know if you have other questions! "}, 'done_
reason': 'stop', 'done': True, 'total_duration': 28105372420, 'load_duratio
n': 20224388, 'prompt_eval_count': 1279, 'prompt_eval_duration': 1613904100
0, 'eval_count': 130, 'eval_duration': 10560334000}
LLM Response: ```sql
SELECT *
FROM genres
WHERE COUNT(*) = 0;
```

Explanation:
```
```

```
* **`SELECT *`**: This selects all columns from the table.
* **`FROM genres`**: We are selecting data from the `genres` table.
* **`WHERE COUNT(*) = 0`**:  This clause filters for rows where the count of
all records in the 'genres' table is equal to zero. The 'COUNT(*)' function
counts all rows, and an empty result means there were no entries.


Let me know if you have other questions!
Info: Output from LLM: ```sql
SELECT *
FROM genres
WHERE COUNT(*) = 0;
```

Explanation:

* **`SELECT *`**: This selects all columns from the table.
* **`FROM genres`**: We are selecting data from the `genres` table.
* **`WHERE COUNT(*) = 0`**: This clause filters for rows where the count of
all records in the 'genres' table is equal to zero. The 'COUNT(*)' function
counts all rows, and an empty result means there were no entries.

Let me know if you have other questions!
Extracted SQL: SELECT *
FROM genres
WHERE COUNT(*) = 0
SELECT *
FROM genres
WHERE COUNT(*) = 0
Couldn't run sql: Execution failed on sql 'SELECT *
FROM genres
WHERE COUNT(*) = 0': misuse of aggregate function COUNT()

In [31]:

```python
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': 'what are the top 5 countries tha
t customers come from?'}, {'role': 'assistant', 'content': 'SELECT Country,
COUNT(DISTINCT CustomerId) AS NumCustomers\r\nFROM customers\r\nGROUP BY Cou
ntry\r\nORDER BY NumCustomers DESC\r\nLIMIT 5'}, {'role': 'user', 'content':
'  \n    Get the total number of invoices for each customer\n'}, {'role': 'a
ssistant', 'content': 'SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS
TotalInvoices\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.Cus
tomerId\r\nGROUP BY c.FirstName, c.LastName'}, {'role': 'user', 'content': '
\n    Find the total number of invoices per country:\n'}, {'role': 'assistan
t', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices \r\nFR
OM customers c \r\nJOIN invoices i ON c.CustomerId = i.CustomerId \r\nGROUP
BY c.Country\r\nORDER BY TotalInvoices DESC'}, {'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 AvgInvoice
Total \r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId
\r\nGROUP BY c.FirstName, c.LastName'}, {'role': 'user', 'content': '  \n
Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'a
ssistant', 'content': "SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\r\nFR
OM invoices i\r\nWHERE i.InvoiceDate >= '2010-01-01'\r\nGROUP BY i.InvoiceI
d"}, {'role': 'user', 'content': '  \n    List all invoices with a total exc
eeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \r\nFROM invoice
s\r\nWHERE Total > 10'}, {'role': 'user', 'content': '  \n    List all album
s and their corresponding artist names  \n'}, {'role': 'assistant', 'conten
t': 'SELECT "albums".Title, "artists".Name \r\nFROM "albums"\r\nJOIN "artist
s" ON "albums"."ArtistId" = "artists"."ArtistId"\r\nORDER BY "albums"."Titl
e"'}, {'role': 'user', 'content': '  \n    List all genres and the number of
tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT "genre
s"."Name", COUNT(DISTINCT "tracks"."TrackId") AS NumTracks\r\nFROM "genre
s"\r\nLEFT JOIN "tracks" ON "genres"."GenreId" = "tracks"."GenreId"\r\nGROUP
BY "genres"."Name"\r\nORDER BY NumTracks DESC'}, {'role': 'user', 'content':
'  \n    Find the top 5 most expensive tracks (based on unit price):\n'},
{'role': 'assistant', 'content': 'SELECT TrackId, UnitPrice\r\nFROM tracks\r
\nORDER BY UnitPrice DESC\r\nLIMIT 5'}, {'role': 'user', 'content': '  \n
List all customers who have not placed any orders:\n'}]
Info: Ollama parameters:
model=gemma2:2b,
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 \"employees\"\r\n(\r\n    EmployeeId INTEGER PRIMARY K
EY AUTOINCREMENT NOT NULL,\r\n    LastName NVARCHAR(20)  NOT NULL,\r\n    Fi
rstName 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 \"playlist_track\"\r\n(\r\n    Pl
aylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    CONSTRA
INT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r\n    FOREIGN KEY
(PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTI
ON 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\nCREATE INDEX IFK_CustomerSupp
ortRepId 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 PR
IMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n
AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NOT NULL,\r\n    GenreId INTEGE
R,\r\n    Composer NVARCHAR(220),\r\n    Milliseconds INTEGER  NOT NULL,\r\n
Bytes INTEGER,\r\n    UnitPrice 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_InvoiceCustomerId ON \"invoices
\" (CustomerId)\n\n\n===Additional Context \n\nIn the chinook database invoi
ce means order\n\n===Response Guidelines \n1. If the provided context is suf
ficient, please generate a valid SQL query without any explanations for the
question. \n2. If the provided context is almost sufficient but requires kno
wledge of a specific string in a particular column, please generate an inter
mediate SQL query to find the distinct strings in that column. Prepend the q
uery with a comment saying intermediate_sql \n3. If the provided context is
insufficient, please explain why it can't be generated. \n4. Please use the
most relevant table(s). \n5. If the question has been asked and answered bef
ore, please repeat the answer exactly as it was given before. \n"}, {"role":
```

```
"user", "content": "How many customers are there"}, {"role": "assistant", "c
ontent": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "wha
t are the top 5 countries that customers come from?"}, {"role": "assistant",
"content": "SELECT Country, COUNT(DISTINCT CustomerId) AS NumCustomers\r\nFR
OM customers\r\nGROUP BY Country\r\nORDER BY NumCustomers DESC\r\nLIMIT 5"},
{"role": "user", "content": "  \n    Get the total number of invoices for ea
ch customer\n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.Las
tName, COUNT(i.InvoiceId) AS TotalInvoices\r\nFROM customers c\r\nJOIN invoi
ces i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName"},
{"role": "user", "content": "  \n    Find the total number of invoices per c
ountry:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.Inv
oiceId) AS TotalInvoices \r\nFROM customers c \r\nJOIN invoices i ON c.Custo
merId = i.CustomerId \r\nGROUP BY c.Country\r\nORDER BY TotalInvoices DES
C"}, {"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 AvgInvoiceTotal \r\nFROM customers c\r\nJOIN invoi
ces i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName"},
{"role": "user", "content": "  \n    Find all invoices since 2010 and the to
tal amount invoiced:\n"}, {"role": "assistant", "content": "SELECT i.Invoice
Id, SUM(i.Total) AS TotalAmount\r\nFROM invoices i\r\nWHERE i.InvoiceDate >=
'2010-01-01'\r\nGROUP BY i.InvoiceId"}, {"role": "user", "content": "  \n
List all invoices with a total exceeding $10:\n"}, {"role": "assistant", "co
ntent": "SELECT * \r\nFROM invoices\r\nWHERE Total > 10"}, {"role": "user",
"content": "  \n    List all albums and their corresponding artist names
\n"}, {"role": "assistant", "content": "SELECT \"albums\".Title, \"artists
\".Name \r\nFROM \"albums\"\r\nJOIN \"artists\" ON \"albums\".\"ArtistId\" =
\"artists\".\"ArtistId\"\r\nORDER BY \"albums\".\"Title\""}, {"role": "use
r", "content": "  \n    List all genres and the number of tracks in each gen
re:\n"}, {"role": "assistant", "content": "SELECT \"genres\".\"Name\", COUNT
(DISTINCT \"tracks\".\"TrackId\") AS NumTracks\r\nFROM \"genres\"\r\nLEFT JO
IN \"tracks\" ON \"genres\".\"GenreId\" = \"tracks\".\"GenreId\"\r\nGROUP BY
\"genres\".\"Name\"\r\nORDER BY NumTracks DESC"}, {"role": "user", "conten
t": "  \n    Find the top 5 most expensive tracks (based on unit pric
e):\n"}, {"role": "assistant", "content": "SELECT TrackId, UnitPrice\r\nFROM
tracks\r\nORDER BY UnitPrice DESC\r\nLIMIT 5"}, {"role": "user", "content":
"  \n    List all customers who have not placed any orders:\n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:30:25.571031881Z', 'mess
age': {'role': 'assistant', 'content': "```sql\r\nSELECT \r\n    c.FirstNam
e,\r\n    c.LastName\r\nFROM \r\n    customers c\r\nWHERE \r\n    NOT EXISTS
(SELECT 1 FROM invoices i WHERE i.CustomerId = c.CustomerId); \r\n``` \r\n\r
\n**Explanation:** \r\n\r\n- We use the `EXISTS` operator in a subquery to c
heck if there is any matching invoice for each customer.\r\n- If there's no
matching invoice, it means the customer hasn't placed any orders.  We select
their first and last names to display them.\r\n\r\n**Important Note:** \r\nT
his query only shows customers who have *no* invoices associated with them.
It doesn't necessarily mean they haven't registered as a customer or made pu
rchases in some other way.\r\n"}, 'done_reason': 'stop', 'done': True, 'tota
l_duration': 38840963115, 'load_duration': 24285893, 'prompt_eval_count': 18
45, 'prompt_eval_duration': 23508542000, 'eval_count': 169, 'eval_duration':
13866663000}
LLM Response: ```sql
SELECT
    c.FirstName,
    c.LastName
FROM
```

```
    customers c
WHERE
    NOT EXISTS (SELECT 1 FROM invoices i WHERE i.CustomerId = c.CustomerId);
```
```

Explanation:

- We use the `EXISTS` operator in a subquery to check if there is any matchi
ng invoice for each customer.
- If there's no matching invoice, it means the customer hasn't placed any or
ders. We select their first and last names to display them.

Important Note:
This query only shows customers who have *no* invoices associated with them.
It doesn't necessarily mean they haven't registered as a customer or made pu
rchases in some other way.

Info: Output from LLM: ```sql
SELECT
    c.FirstName,
    c.LastName
FROM
    customers c
WHERE
    NOT EXISTS (SELECT 1 FROM invoices i WHERE i.CustomerId = c.CustomerId);
```

**Explanation:**

- We use the `EXISTS` operator in a subquery to check if there is any matchi
ng invoice for each customer.
- If there's no matching invoice, it means the customer hasn't placed any or
ders.  We select their first and last names to display them.

**Important Note:**
This query only shows customers who have *no* invoices associated with them.
It doesn't necessarily mean they haven't registered as a customer or made pu
rchases in some other way.

Extracted SQL: SELECT
    c.FirstName,
    c.LastName
FROM
    customers c
WHERE
    NOT EXISTS (SELECT 1 FROM invoices i WHERE i.CustomerId = c.CustomerId)
SELECT
    c.FirstName,
    c.LastName
FROM
    customers c
WHERE
    NOT EXISTS (SELECT 1 FROM invoices i WHERE i.CustomerId = c.CustomerId)
Empty DataFrame
Columns: [FirstName, LastName]
Index: []
````
````

```
Info: Ollama parameters:
model=gemma2:2b,
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 customers who have not placed any orders:\n'\n\nThe DataFrame
was produced using this query: SELECT \r\n    c.FirstName,\r\n    c.LastName
\r\nFROM \r\n    customers c\r\nWHERE \r\n    NOT EXISTS (SELECT 1 FROM invo
ices i WHERE i.CustomerId = c.CustomerId)\n\nThe following is information ab
out the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n FirstN
ame    object\nLastName     object\ndtype: object"}, {"role": "user", "conte
nt": "Can you generate the Python plotly code to chart the results of the da
taframe? Assume the data is in a pandas dataframe called 'df'. If there is o
nly one value in the dataframe, use an Indicator. Respond with only Python c
ode. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:30:32.058196164Z', 'mess
age': {'role': 'assistant', 'content': "```python\nimport plotly.graph_objec
ts as go\n\nfig = go.Figure(data=go.Indicator(\n    value=df['CustomerId'].v
alue[0], \n    mode='bar',\n    text=None, \n))\n\nfig.show()\n```"}, 'done_
reason': 'stop', 'done': True, 'total_duration': 6484813946, 'load_duratio
n': 16322297, 'prompt_eval_count': 196, 'prompt_eval_duration': 2120357000,
'eval_count': 60, 'eval_duration': 4258484000}
```

Out[31]:
```
('SELECT \r\n    c.FirstName,\r\n    c.LastName\r\nFROM \r\n    customers c
\r\nWHERE \r\n    NOT EXISTS (SELECT 1 FROM invoices i WHERE i.CustomerId =
c.CustomerId)',
 Empty DataFrame
 Columns: [FirstName, LastName]
 Index: [],
 Figure({
     'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
               'hovertemplate': 'FirstName=%{label}<extra></extra>',
               'labels': array([], dtype=object),
               'legendgroup': '',
               'name': '',
               'showlegend': True,
               'type': 'pie'}],
     'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'templ
ate': '...'}
 }))
```

In [32]:
```python
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 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 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 TABLE "artists"\r\n(\r\n    ArtistId INTEGER PRIMARY KEY AUTOI
NCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK_AlbumA
rtistId 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    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 INDEX IFK_Tra
ckGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK_PlaylistTrackTrackId ON
"playlist_track" (TrackId)\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': '  \n    List all genres and the number of
tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT "genre
s"."Name", COUNT(DISTINCT "tracks"."TrackId") AS NumTracks\r\nFROM "genre
s"\r\nLEFT JOIN "tracks" ON "genres"."GenreId" = "tracks"."GenreId"\r\nGROUP
BY "genres"."Name"\r\nORDER BY NumTracks DESC'}, {'role': 'user', 'content':
'  \n    Find the top 5 most expensive tracks (based on unit price):\n'},
{'role': 'assistant', 'content': 'SELECT TrackId, UnitPrice\r\nFROM tracks\r
\nORDER BY UnitPrice DESC\r\nLIMIT 5'}, {'role': 'user', 'content': '  \n
List all albums and their corresponding artist names  \n'}, {'role': 'assist
ant', 'content': 'SELECT "albums".Title, "artists".Name \r\nFROM "albums"\r
\nJOIN "artists" ON "albums"."ArtistId" = "artists"."ArtistId"\r\nORDER BY
"albums"."Title"'}, {'role': 'user', 'content': 'what are the top 5 countrie
s that customers come from?'}, {'role': 'assistant', 'content': 'SELECT Coun
try, COUNT(DISTINCT CustomerId) AS NumCustomers\r\nFROM customers\r\nGROUP B
Y Country\r\nORDER BY NumCustomers DESC\r\nLIMIT 5'}, {'role': 'user', 'cont
ent': '  \n    Find all tracks with a name containing "What" (case-insensiti
ve)\n'}, {'role': 'assistant', 'content': "SELECT * \r\nFROM tracks\r\nWHERE
```

```
Name LIKE '%what%'\r\nORDER BY Name"}, {'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 \r\nFROM cus
tomers c \r\nJOIN invoices i ON c.CustomerId = i.CustomerId \r\nGROUP BY c.C
ountry\r\nORDER BY TotalInvoices DESC'}, {'role': 'user', 'content': '  \n
Get the total number of invoices for each customer\n'}, {'role': 'assistan
t', 'content': 'SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalI
nvoices\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerI
d\r\nGROUP BY c.FirstName, c.LastName'}, {'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 AvgInvoiceTotal
\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGR
OUP BY c.FirstName, c.LastName'}, {'role': 'user', 'content': 'Can you list
all tables in the SQLite database catalog?'}, {'role': 'assistant', 'conten
t': "SELECT name FROM sqlite_schema WHERE type='table'"}, {'role': 'user',
'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'rol
e': 'assistant', 'content': 'SELECT * \r\nFROM invoices\r\nWHERE Total > 1
0'}, {'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'}]
Info: Ollama parameters:
model=gemma2:2b,
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 \"genres\".\"Name\", COUNT(DISTINCT \"tracks\".\"TrackId\") AS NumTr
acks\r\nFROM \"genres\"\r\nLEFT JOIN \"tracks\" ON \"genres\".\"GenreId\" =
\"tracks\".\"GenreId\"\r\nGROUP BY \"genres\".\"Name\"\r\nORDER BY NumTracks
DESC"}, {"role": "user", "content": "  \n    Find the top 5 most expensive t
racks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT T
rackId, UnitPrice\r\nFROM tracks\r\nORDER BY UnitPrice DESC\r\nLIMIT 5"},
{"role": "user", "content": "  \n    List all albums and their corresponding
artist names  \n"}, {"role": "assistant", "content": "SELECT \"albums\".Titl
e, \"artists\".Name \r\nFROM \"albums\"\r\nJOIN \"artists\" ON \"albums
\".\"ArtistId\" = \"artists\".\"ArtistId\"\r\nORDER BY \"albums\".\"Title
\""}, {"role": "user", "content": "what are the top 5 countries that custome
rs come from?"}, {"role": "assistant", "content": "SELECT Country, COUNT(DIS
TINCT CustomerId) AS NumCustomers\r\nFROM customers\r\nGROUP BY Country\r\nO
RDER BY NumCustomers DESC\r\nLIMIT 5"}, {"role": "user", "content": "  \n
Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"ro
le": "assistant", "content": "SELECT * \r\nFROM tracks\r\nWHERE Name LIKE '%
what%'\r\nORDER BY Name"}, {"role": "user", "content": "  \n    Find the tot
al number of invoices per country:\n"}, {"role": "assistant", "content": "SE
LECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices \r\nFROM customers c \r
\nJOIN invoices i ON c.CustomerId = i.CustomerId \r\nGROUP BY c.Country\r\nO
RDER BY TotalInvoices DESC"}, {"role": "user", "content": "  \n    Get the t
otal number of invoices for each customer\n"}, {"role": "assistant", "conten
t": "SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\r\n
FROM customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP
BY c.FirstName, c.LastName"}, {"role": "user", "content": "  \n    Get the a
verage invoice total for each customer:\n"}, {"role": "assistant", "conten
t": "SELECT c.FirstName, c.LastName, AVG(i.Total) AS AvgInvoiceTotal \r\nFRO
M customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY
c.FirstName, c.LastName"}, {"role": "user", "content": "Can you list all tab
les in the SQLite database catalog?"}, {"role": "assistant", "content": "SEL
ECT name FROM sqlite_schema WHERE type='table'"}, {"role": "user", "conten
t": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "as
sistant", "content": "SELECT * \r\nFROM invoices\r\nWHERE Total > 10"}, {"ro
le": "user", "content": " \n    There are 3 tables: artists, albums and trac
ks, where albums and artists are linked by ArtistId, albums and tracks are l
inked by AlbumId,\n    Can you find the top 10 most popular artists based on
the number of tracks\n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:31:14.754840886Z', 'mess
age': {'role': 'assistant', 'content': '```sql\r\nSELECT \r\n    "artist
s"."Name", \r\n    COUNT("tracks"."TrackId") AS NumTracks\r\nFROM "artist
s"\r\nJOIN "albums" ON "artists"."ArtistId" = "albums"."ArtistId"\r\nJOIN "t
racks" ON "albums"."AlbumId" = "tracks"."AlbumId"\r\nGROUP BY "artists"."Nam
e"\r\nORDER BY NumTracks DESC \r\nLIMIT 10; \r\n``` \r\n\r\n**Explanation:**
\r\n\r\n1. **Joins**: We use `JOIN` clauses to link the tables based on thei
```

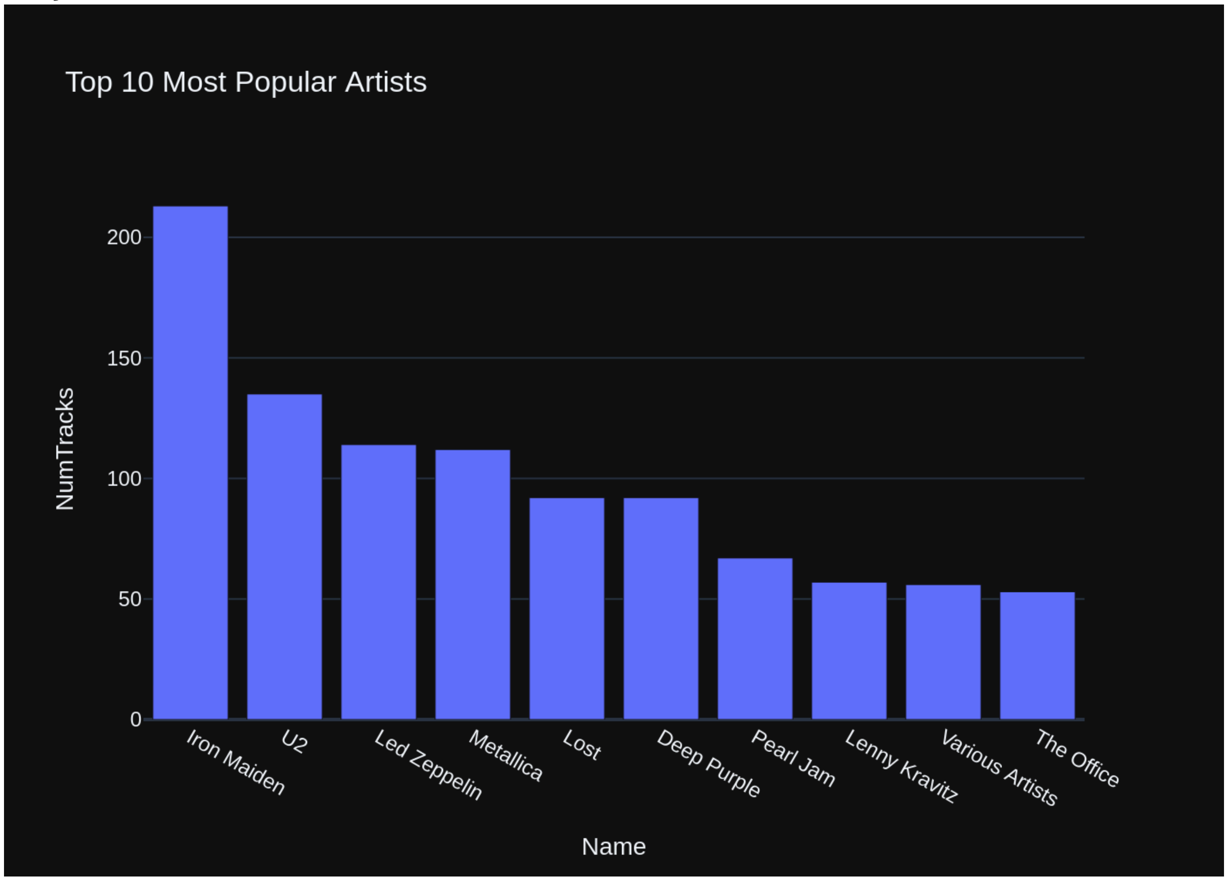
```
r relationships:\r\n   -  `JOIN "albums" ON "artists"."ArtistId" = "album
s"."ArtistId"`: This connects artists and albums based on the `ArtistId`. \r
\n   -  `JOIN "tracks" ON "albums"."AlbumId" = "tracks"."AlbumId"`: This lin
ks albums and tracks. \r\n2. **Grouping**:  `GROUP BY "artists"."Name"` grou
ps all rows with same artist names together.\r\n3. **Counting Tracks**: We u
se the `COUNT("tracks"."TrackId")` function to count the number of tracks fo
r each artist. \n4. **Ordering**: The result is sorted in descending order b
ased on the `NumTracks`.\n5. **Limiting**: The `LIMIT 10` clause ensures onl
y the top 10 most popular artists are returned. \r\n\n\n'}, 'done_reason':
'stop', 'done': True, 'total_duration': 42616595212, 'load_duration': 220545
48, 'prompt_eval_count': 1375, 'prompt_eval_duration': 16949486000, 'eval_co
unt': 300, 'eval_duration': 24308850000}
LLM Response: ```sql
SELECT
    "artists"."Name",
    COUNT("tracks"."TrackId") AS NumTracks
FROM "artists"
JOIN "albums" ON "artists"."ArtistId" = "albums"."ArtistId"
JOIN "tracks" ON "albums"."AlbumId" = "tracks"."AlbumId"
GROUP BY "artists"."Name"
ORDER BY NumTracks DESC
LIMIT 10;
```

**Explanation:**

1. **Joins**: We use `JOIN` clauses to link the tables based on their relati
onships:
   -  `JOIN "albums" ON "artists"."ArtistId" = "albums"."ArtistId"`: This co
nnects artists and albums based on the `ArtistId`.
   -  `JOIN "tracks" ON "albums"."AlbumId" = "tracks"."AlbumId"`: This links
albums and tracks.
2. **Grouping**:  `GROUP BY "artists"."Name"` groups all rows with same arti
st names together.
3. **Counting Tracks**: We use the `COUNT("tracks"."TrackId")` function to c
ount the number of tracks for each artist.
4. **Ordering**: The result is sorted in descending order based on the `NumT
racks`.
5. **Limiting**: The `LIMIT 10` clause ensures only the top 10 most popular
artists are returned.

Info: Output from LLM: ```sql
SELECT
    "artists"."Name",
    COUNT("tracks"."TrackId") AS NumTracks
FROM "artists"
JOIN "albums" ON "artists"."ArtistId" = "albums"."ArtistId"
JOIN "tracks" ON "albums"."AlbumId" = "tracks"."AlbumId"
GROUP BY "artists"."Name"
ORDER BY NumTracks DESC
LIMIT 10;
```

**Explanation:**
```

```
1. **Joins**: We use `JOIN` clauses to link the tables based on their relati
onships:
   -  `JOIN "albums" ON "artists"."ArtistId" = "albums"."ArtistId"`: This co
nnects artists and albums based on the `ArtistId`.
   -  `JOIN "tracks" ON "albums"."AlbumId" = "tracks"."AlbumId"`: This links
albums and tracks.
2. **Grouping**:  `GROUP BY "artists"."Name"` groups all rows with same arti
st names together.
3. **Counting Tracks**: We use the `COUNT("tracks"."TrackId")` function to c
ount the number of tracks for each artist.
4. **Ordering**: The result is sorted in descending order based on the `NumT
racks`.
5. **Limiting**: The `LIMIT 10` clause ensures only the top 10 most popular
artists are returned.




Extracted SQL: SELECT
    "artists"."Name",
    COUNT("tracks"."TrackId") AS NumTracks
FROM "artists"
JOIN "albums" ON "artists"."ArtistId" = "albums"."ArtistId"
JOIN "tracks" ON "albums"."AlbumId" = "tracks"."AlbumId"
GROUP BY "artists"."Name"
ORDER BY NumTracks DESC
LIMIT 10
SELECT
    "artists"."Name",
    COUNT("tracks"."TrackId") AS NumTracks
FROM "artists"
JOIN "albums" ON "artists"."ArtistId" = "albums"."ArtistId"
JOIN "tracks" ON "albums"."AlbumId" = "tracks"."AlbumId"
GROUP BY "artists"."Name"
ORDER BY NumTracks DESC
LIMIT 10
              Name  NumTracks
0      Iron Maiden        213
1               U2        135
2     Led Zeppelin        114
3        Metallica        112
4             Lost         92
5      Deep Purple         92
6        Pearl Jam         67
7    Lenny Kravitz         57
8  Various Artists         56
9       The Office         53
Info: Ollama parameters:
model=gemma2:2b,
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 \r\n    \"artists
\".\"Name\", \r\n    COUNT(\"tracks\".\"TrackId\") AS NumTracks\r\nFROM \"ar
tists\"\r\nJOIN \"albums\" ON \"artists\".\"ArtistId\" = \"albums\".\"Artist
Id\"\r\nJOIN \"tracks\" ON \"albums\".\"AlbumId\" = \"tracks\".\"AlbumId\"\r
\nGROUP BY \"artists\".\"Name\"\r\nORDER BY NumTracks DESC \r\nLIMIT 10\n\nT
he following is information about the resulting pandas DataFrame 'df': \nRun
ning df.dtypes gives:\n Name         object\nNumTracks     int64\ndtype: obj
ect"}, {"role": "user", "content": "Can you generate the Python plotly code 
to chart the results of the dataframe? Assume the data is in a pandas datafr
ame called 'df'. If there is only one value in the dataframe, use an Indicat
or. Respond with only Python code. Do not answer with any explanations -- ju
st the code."}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:31:21.370600362Z', 'mess
age': {'role': 'assistant', 'content': '```python\nimport plotly.express as 
px\n\nfig = px.bar(df, x="Name", y="NumTracks", title="Top 10 Most Popular A
rtists")\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_dur
ation': 6589875844, 'load_duration': 24105606, 'prompt_eval_count': 280, 'pr
ompt_eval_duration': 3272520000, 'eval_count': 45, 'eval_duration': 32050190
00}
```



```
Out[32]: ('SELECT \r\n    "artists"."Name", \r\n    COUNT("tracks"."TrackId") AS Num
         Tracks\r\nFROM "artists"\r\nJOIN "albums" ON "artists"."ArtistId" = "album
         s"."ArtistId"\r\nJOIN "tracks" ON "albums"."AlbumId" = "tracks"."AlbumId"\r
         \nGROUP BY "artists"."Name"\r\nORDER BY NumTracks DESC \r\nLIMIT 10',
                       Name  NumTracks
          0      Iron Maiden        213
          1               U2        135
          2     Led Zeppelin        114
          3        Metallica        112
          4             Lost         92
          5      Deep Purple         92
          6        Pearl Jam         67
          7    Lenny Kravitz         57
          8  Various Artists         56
          9       The Office         53,
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'hovertemplate': 'Name=%{x}<br>NumTracks=%{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', 'Lost', 'Deep Purple',
                                    'Pearl Jam', 'Lenny Kravitz', 'Various Artists',
         'The Office'],
                                   dtype=object),
                        'xaxis': 'x',
                        'y': array([213, 135, 114, 112,  92,  92,  67,  57,  56,  5
         3]),
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                         'legend': {'tracegroupgap': 0},
                         'template': '...',
                         'title': {'text': 'Top 10 Most Popular Artists'},
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
         ext': 'Name'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
         ext': 'NumTracks'}}}
          }))
```

```
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
```

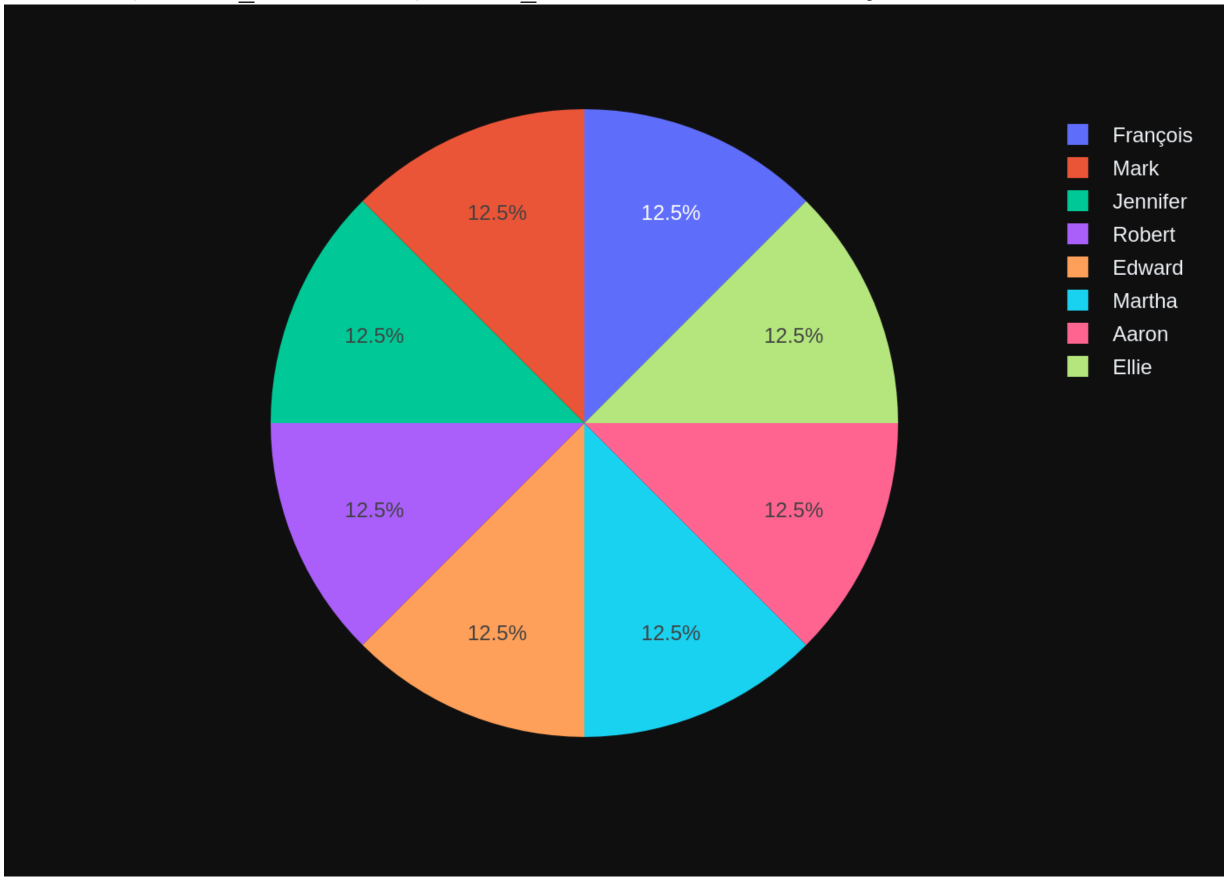
```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please
help to generate a SQL query to answer the question. Your response should ON
LY be based on the given context and follow the response guidelines and form
at instructions. \n===Tables \nCREATE INDEX IFK_CustomerSupportRepId ON "cus
tomers" (SupportRepId)\n\nCREATE TABLE "customers"\r\n(\r\n    CustomerId IN
TEGER 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    Ph
one NVARCHAR(24),\r\n    Fax NVARCHAR(24),\r\n    Email NVARCHAR(60)  NOT NU
LL,\r\n    SupportRepId INTEGER,\r\n    FOREIGN KEY (SupportRepId) REFERENCE
S "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 AUT
OINCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate
DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    BillingCity N
VARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHA
R(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NO
T 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_Inv
oiceCustomerId 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    Titl
e 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" (Employe
eId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "i
nvoice_items"\r\n(\r\n    InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NO
T 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 sqlite_sequence(name,seq)\n\nCREATE TABLE "playlist_trac
k"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    TrackId INTEGER  NOT NUL
L,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (PlaylistId, TrackId),\r
\n    FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tO
N DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (TrackId) REFEREN
CES "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 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\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': 'what are the top 5 countries t
hat customers come from?'}, {'role': 'assistant', 'content': 'SELECT Countr
```

```
y, COUNT(DISTINCT CustomerId) AS NumCustomers\r\nFROM customers\r\nGROUP BY
Country\r\nORDER BY NumCustomers DESC\r\nLIMIT 5'}, {'role': 'user', 'conten
t': '  \n    Find the total number of invoices per country:\n'}, {'role': 'a
ssistant', 'content': 'SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices
\r\nFROM customers c \r\nJOIN invoices i ON c.CustomerId = i.CustomerId \r\n
GROUP BY c.Country\r\nORDER BY TotalInvoices DESC'}, {'role': 'user', 'conte
nt': '  \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\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId
= i.CustomerId\r\nGROUP BY c.FirstName, c.LastName'}, {'role': 'user', 'cont
ent': 'How many customers are there'}, {'role': 'assistant', 'content': 'SEL
ECT COUNT(*) FROM customers'}, {'role': 'user', 'content': '  \n    Get the
average invoice total for each customer:\n'}, {'role': 'assistant', 'conten
t': 'SELECT c.FirstName, c.LastName, AVG(i.Total) AS AvgInvoiceTotal \r\nFRO
M customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY
c.FirstName, c.LastName'}, {'role': 'user', 'content': '  \n    List all inv
oices with a total exceeding $10:\n'}, {'role': 'assistant', 'content': 'SEL
ECT * \r\nFROM invoices\r\nWHERE Total > 10'}, {'role': 'user', 'content': '
\n    List all albums and their corresponding artist names  \n'}, {'role':
'assistant', 'content': 'SELECT "albums".Title, "artists".Name \r\nFROM "alb
ums"\r\nJOIN "artists" ON "albums"."ArtistId" = "artists"."ArtistId"\r\nORDE
R BY "albums"."Title"'}, {'role': 'user', 'content': '  \n    Find all invoi
ces since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'co
ntent': "SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\r\nFROM invoices i
\r\nWHERE i.InvoiceDate >= '2010-01-01'\r\nGROUP BY i.InvoiceId"}, {'role':
'user', 'content': ' \n    There are 3 tables: artists, albums and tracks, w
here 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 n
umber of tracks\n'}, {'role': 'assistant', 'content': 'SELECT \r\n    "artis
ts"."Name", \r\n    COUNT("tracks"."TrackId") AS NumTracks\r\nFROM "artist
s"\r\nJOIN "albums" ON "artists"."ArtistId" = "albums"."ArtistId"\r\nJOIN "t
racks" ON "albums"."AlbumId" = "tracks"."AlbumId"\r\nGROUP BY "artists"."Nam
e"\r\nORDER BY NumTracks DESC \r\nLIMIT 10'}, {'role': 'user', 'content': 'C
an you list all tables in the SQLite database catalog?'}, {'role': 'assistan
t', 'content': "SELECT name FROM sqlite_schema WHERE type='table'"}, {'rol
e': 'user', 'content': '  \n     List all customers from Canada and their em
ail addresses:\n'}]
Info: Ollama parameters:
model=gemma2:2b,
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(DISTINCT CustomerId) AS NumCustomers\r\nFROM customers
\r\nGROUP BY Country\r\nORDER BY NumCustomers DESC\r\nLIMIT 5"}, {"role": "u
ser", "content": "  \n    Find the total number of invoices per countr
y:\n"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceI
d) AS TotalInvoices \r\nFROM customers c \r\nJOIN invoices i ON c.CustomerId
= i.CustomerId \r\nGROUP BY c.Country\r\nORDER BY TotalInvoices DESC"}, {"ro
le": "user", "content": "  \n    Get the total number of invoices for each c
ustomer\n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastNam
e, COUNT(i.InvoiceId) AS TotalInvoices\r\nFROM customers c\r\nJOIN invoices
i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName"}, {"ro
le": "user", "content": "How many customers are there"}, {"role": "assistan
t", "content": "SELECT COUNT(*) FROM customers"}, {"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 Avg
```

```
InvoiceTotal \r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.Cus
tomerId\r\nGROUP BY c.FirstName, c.LastName"}, {"role": "user", "content": "
\n    List all invoices with a total exceeding $10:\n"}, {"role": "assistan
t", "content": "SELECT * \r\nFROM invoices\r\nWHERE Total > 10"}, {"role":
"user", "content": "  \n    List all albums and their corresponding artist n
ames  \n"}, {"role": "assistant", "content": "SELECT \"albums\".Title, \"art
ists\".Name \r\nFROM \"albums\"\r\nJOIN \"artists\" ON \"albums\".\"ArtistId
\" = \"artists\".\"ArtistId\"\r\nORDER BY \"albums\".\"Title\""}, {"role":
"user", "content": "  \n    Find all invoices since 2010 and the total amoun
t invoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceId, SUM
(i.Total) AS TotalAmount\r\nFROM invoices i\r\nWHERE i.InvoiceDate >= '2010-
01-01'\r\nGROUP BY i.InvoiceId"}, {"role": "user", "content": " \n    There
are 3 tables: artists, albums and tracks, where albums and artists are linke
d by ArtistId, albums and tracks are linked by AlbumId,\n    Can you find th
e top 10 most popular artists based on the number of tracks\n"}, {"role": "a
ssistant", "content": "SELECT \r\n    \"artists\".\"Name\", \r\n    COUNT
(\"tracks\".\"TrackId\") AS NumTracks\r\nFROM \"artists\"\r\nJOIN \"albums\"
ON \"artists\".\"ArtistId\" = \"albums\".\"ArtistId\"\r\nJOIN \"tracks\" ON
\"albums\".\"AlbumId\" = \"tracks\".\"AlbumId\"\r\nGROUP BY \"artists\".\"Na
me\"\r\nORDER BY NumTracks DESC \r\nLIMIT 10"}, {"role": "user", "content":
"Can you list all tables in the SQLite database catalog?"}, {"role": "assist
ant", "content": "SELECT name FROM sqlite_schema WHERE type='table'"}, {"rol
e": "user", "content": "  \n     List all customers from Canada and their em
ail addresses:\n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:31:47.240405007Z', 'mess
age': {'role': 'assistant', 'content': "```sql\nSELECT c.FirstName, c.LastNa
me, c.Email \r\nFROM customers c\r\nWHERE c.Country = 'Canada'\r\n```\r\n"},
'done_reason': 'stop', 'done': True, 'total_duration': 25751550201, 'load_du
ration': 19412457, 'prompt_eval_count': 1707, 'prompt_eval_duration': 215799
67000, 'eval_count': 34, 'eval_duration': 2700622000}
LLM Response: ```sql
SELECT c.FirstName, c.LastName, c.Email
FROM customers c
WHERE c.Country = 'Canada'
```

Info: Output from LLM: ```sql
SELECT c.FirstName, c.LastName, c.Email
FROM customers c
WHERE c.Country = 'Canada'
```

Extracted SQL: SELECT c.FirstName, c.LastName, c.Email
FROM customers c
WHERE c.Country = 'Canada'

SELECT c.FirstName, c.LastName, c.Email
FROM customers c
WHERE c.Country = 'Canada'

  FirstName  LastName                   Email
0  François  Tremblay     ftremblay@gmail.com
1      Mark   Philips      mphilips12@shaw.ca
2  Jennifer  Peterson     jenniferp@rogers.ca
3    Robert     Brown        robbrown@shaw.ca
```

```
4    Edward   Francis     edfrancis@yachoo.ca
5    Martha      Silk    marthasilk@gmail.com
6     Aaron  Mitchell  aaronmitchell@yahoo.ca
7     Ellie  Sullivan  ellie.sullivan@shaw.ca
Info: Ollama parameters:
model=gemma2:2b,
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 customers from Canada and their email addresses:\n'\n\nThe D
ataFrame was produced using this query: SELECT c.FirstName, c.LastName, c.Em
ail \r\nFROM customers c\r\nWHERE c.Country = 'Canada'\r\n\n\nThe following
is information about the resulting pandas DataFrame 'df': \nRunning df.dtype
s gives:\n FirstName    object\nLastName     object\nEmail        object\ndt
ype: object"}, {"role": "user", "content": "Can you generate the Python plot
ly code to chart the results of the dataframe? Assume the data is in a panda
s dataframe called 'df'. If there is only one value in the dataframe, use an
Indicator. Respond with only Python code. Do not answer with any explanation
s -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:31:54.113048554Z', 'mess
age': {'role': 'assistant', 'content': "```python\nimport plotly.graph_objec
ts as go\n\nfig = go.Figure(data=go.Indicator(\n    value=df['Email'].values
[0],\n    mode='bar',\n    text=df['Email'].values[0]\n))\n\nfig.show()\n``
`"}, 'done_reason': 'stop', 'done': True, 'total_duration': 6845434499, 'loa
d_duration': 20606976, 'prompt_eval_count': 182, 'prompt_eval_duration': 196
8267000, 'eval_count': 64, 'eval_duration': 4769031000}
```



Out[33]:
```
("SELECT c.FirstName, c.LastName, c.Email \r\nFROM customers c\r\nWHERE c.C
ountry = 'Canada'\r\n",
   FirstName  LastName                    Email
 0  François  Tremblay     ftremblay@gmail.com
 1      Mark   Philips      mphilips12@shaw.ca
 2  Jennifer  Peterson     jenniferp@rogers.ca
 3    Robert     Brown        robbrown@shaw.ca
 4    Edward   Francis     edfrancis@yachoo.ca
 5    Martha      Silk    marthasilk@gmail.com
 6     Aaron  Mitchell  aaronmitchell@yahoo.ca
 7     Ellie  Sullivan  ellie.sullivan@shaw.ca,
 Figure({
     'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
               'hovertemplate': 'FirstName=%{label}<extra></extra>',
               'labels': array(['François', 'Mark', 'Jennifer', 'Robert',
'Edward', 'Martha', 'Aaron',
                                 'Ellie'], dtype=object),
               'legendgroup': '',
               'name': '',
               'showlegend': True,
               'type': 'pie'}],
     'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'templ
ate': '...'}
 }))
```

In [34]:
```
question = """
     Find the customer with the most invoices
"""

vn.ask(question=question)
```

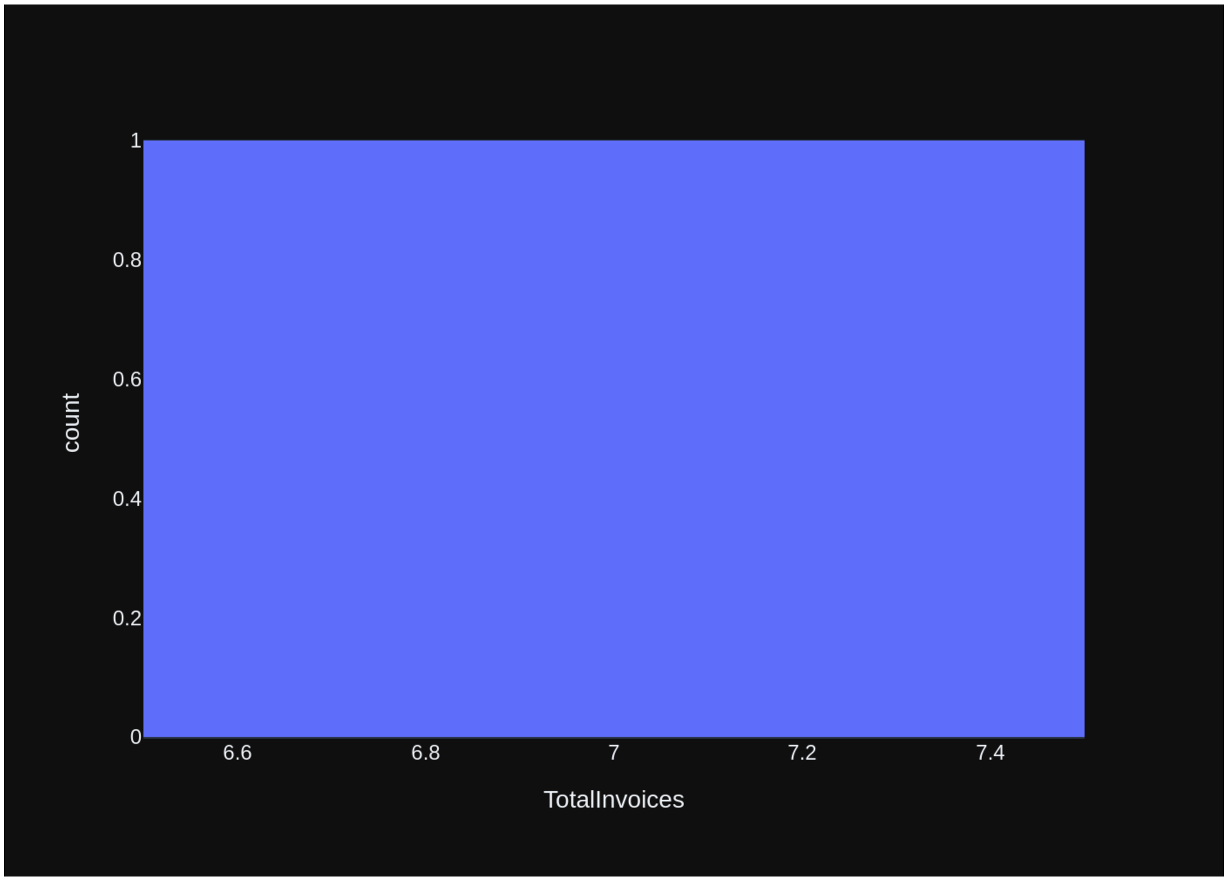
```
Number of requested results 10 is greater than number of elements in index
1, updating n_results = 1
```

```
SQL Prompt: [{'role': 'system', 'content': 'You are a SQLite expert. Please
help to generate a SQL query to answer the question. Your response should 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': '  \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\r
\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROU
P BY c.FirstName, c.LastName'}, {'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 \r\nFROM customer
s c \r\nJOIN invoices i ON c.CustomerId = i.CustomerId \r\nGROUP BY c.Countr
y\r\nORDER BY TotalInvoices DESC'}, {'role': 'user', 'content': '  \n    Get
the average invoice total for each customer:\n'}, {'role': 'assistant', 'con
tent': 'SELECT c.FirstName, c.LastName, AVG(i.Total) AS AvgInvoiceTotal \r\n
FROM customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP
BY c.FirstName, c.LastName'}, {'role': 'user', 'content': '  \n    List all
invoices with a total exceeding $10:\n'}, {'role': 'assistant', 'content':
'SELECT * \r\nFROM invoices\r\nWHERE Total > 10'}, {'role': 'user', 'conten
t': '  \n    Find all invoices since 2010 and the total amount invoice
d:\n'}, {'role': 'assistant', 'content': "SELECT i.InvoiceId, SUM(i.Total) A
S TotalAmount\r\nFROM invoices i\r\nWHERE i.InvoiceDate >= '2010-01-01'\r\nG
ROUP BY i.InvoiceId"}, {'role': 'user', 'content': 'what are the top 5 count
ries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT C
ountry, COUNT(DISTINCT CustomerId) AS NumCustomers\r\nFROM customers\r\nGROU
P BY Country\r\nORDER BY NumCustomers DESC\r\nLIMIT 5'}, {'role': 'user', 'c
ontent': '  \n    Find the top 5 most expensive tracks (based on unit pric
e):\n'}, {'role': 'assistant', 'content': 'SELECT TrackId, UnitPrice\r\nFROM
tracks\r\nORDER BY UnitPrice DESC\r\nLIMIT 5'}, {'role': 'user', 'content':
'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT CO
UNT(*) FROM customers'}, {'role': 'user', 'content': '  \n     List all cust
omers from Canada and their email addresses:\n'}, {'role': 'assistant', 'con
tent': "SELECT c.FirstName, c.LastName, c.Email \r\nFROM customers c\r\nWHER
E c.Country = 'Canada'\r\n"}, {'role': 'user', 'content': ' \n    There are
3 tables: artists, albums and tracks, where albums and artists are linked by
ArtistId, albums and tracks are linked by AlbumId,\n    Can you find the top
10 most popular artists based on the number of tracks\n'}, {'role': 'assista
nt', 'content': 'SELECT \r\n    "artists"."Name", \r\n    COUNT("tracks"."Tr
ackId") AS NumTracks\r\nFROM "artists"\r\nJOIN "albums" ON "artists"."Artist
Id" = "albums"."ArtistId"\r\nJOIN "tracks" ON "albums"."AlbumId" = "track
s"."AlbumId"\r\nGROUP BY "artists"."Name"\r\nORDER BY NumTracks DESC \r\nLIM
IT 10'}, {'role': 'user', 'content': '  \n     Find the customer with the mo
st invoices \n'}]
Info: Ollama parameters:
model=gemma2:2b,
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\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerI
d = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName"}, {"role": "user", "co
ntent": "  \n    Find the total number of invoices per country:\n"}, {"rol
e": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalIn
voices \r\nFROM customers c \r\nJOIN invoices i ON c.CustomerId = i.Customer
Id \r\nGROUP BY c.Country\r\nORDER BY TotalInvoices DESC"}, {"role": "user",
"content": "  \n    Get the average invoice total for each customer:\n"},
{"role": "assistant", "content": "SELECT c.FirstName, c.LastName, AVG(i.Tota
l) AS AvgInvoiceTotal \r\nFROM customers c\r\nJOIN invoices i ON c.CustomerI
d = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName"}, {"role": "user", "co
ntent": "  \n    List all invoices with a total exceeding $10:\n"}, {"role":
```

```
"assistant", "content": "SELECT * \r\nFROM invoices\r\nWHERE Total > 10"},
{"role": "user", "content": "  \n    Find all invoices since 2010 and the to
tal amount invoiced:\n"}, {"role": "assistant", "content": "SELECT i.Invoice
Id, SUM(i.Total) AS TotalAmount\r\nFROM invoices i\r\nWHERE i.InvoiceDate >=
'2010-01-01'\r\nGROUP BY i.InvoiceId"}, {"role": "user", "content": "what ar
e the top 5 countries that customers come from?"}, {"role": "assistant", "co
ntent": "SELECT Country, COUNT(DISTINCT CustomerId) AS NumCustomers\r\nFROM
customers\r\nGROUP BY Country\r\nORDER BY NumCustomers DESC\r\nLIMIT 5"},
{"role": "user", "content": "  \n    Find the top 5 most expensive tracks (b
ased on unit price):\n"}, {"role": "assistant", "content": "SELECT TrackId,
UnitPrice\r\nFROM tracks\r\nORDER BY UnitPrice DESC\r\nLIMIT 5"}, {"role":
"user", "content": "How many customers are there"}, {"role": "assistant", "c
ontent": "SELECT COUNT(*) FROM customers"}, {"role": "user", "content": "
\n     List all customers from Canada and their email addresses:\n"}, {"rol
e": "assistant", "content": "SELECT c.FirstName, c.LastName, c.Email \r\nFRO
M customers c\r\nWHERE c.Country = 'Canada'\r\n"}, {"role": "user", "conten
t": " \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 \r\n    \"artists\".\"Name\",
\r\n    COUNT(\"tracks\".\"TrackId\") AS NumTracks\r\nFROM \"artists\"\r\nJO
IN \"albums\" ON \"artists\".\"ArtistId\" = \"albums\".\"ArtistId\"\r\nJOIN
\"tracks\" ON \"albums\".\"AlbumId\" = \"tracks\".\"AlbumId\"\r\nGROUP BY
\"artists\".\"Name\"\r\nORDER BY NumTracks DESC \r\nLIMIT 10"}, {"role": "us
er", "content": "  \n     Find the customer with the most invoices \n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:32:23.494196135Z', 'mess
age': {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, COUN
T(i.InvoiceId) AS TotalInvoices\r\nFROM customers c\r\nJOIN invoices i ON c.
CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName\r\nORDER BY To
talInvoices DESC\r\nLIMIT 1; \r\n\n\n'}, 'done_reason': 'stop', 'done': Tru
e, 'total_duration': 29289163564, 'load_duration': 23053607, 'prompt_eval_co
unt': 1742, 'prompt_eval_duration': 22834434000, 'eval_count': 64, 'eval_dur
ation': 5168580000}
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.FirstName, c.LastName
ORDER BY TotalInvoices DESC
LIMIT 1;




Info: Output from LLM: SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS
TotalInvoices
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalInvoices DESC
LIMIT 1;



Extracted SQL: SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalIn
voices
```

```
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalInvoices DESC
LIMIT 1
SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalInvoices DESC
LIMIT 1
  FirstName  LastName  TotalInvoices
0     Aaron  Mitchell              7
Info: Ollama parameters:
model=gemma2:2b,
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 TotalInvoices\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.C
ustomerId\r\nGROUP BY c.FirstName, c.LastName\r\nORDER BY TotalInvoices DESC
\r\nLIMIT 1\n\nThe following is information about the resulting pandas DataF
rame 'df': \nRunning df.dtypes gives:\n FirstName        object\nLastName
object\nTotalInvoices     int64\ndtype: object"}, {"role": "user", "conten
t": "Can you generate the Python plotly code to chart the results of the dat
aframe? Assume the data is in a pandas dataframe called 'df'. If there is on
ly one value in the dataframe, use an Indicator. Respond with only Python co
de. Do not answer with any explanations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:32:30.477343181Z', 'mess
age': {'role': 'assistant', 'content': '```python\nimport plotly.express as
px\n\nfig = px.histogram(df, x="TotalInvoices") \n# fig.update_traces(marker
=dict(color=\'green\'))  # optional: add a color to your chart (e.g., green)
\nfig.show()\n```'}, 'done_reason': 'stop', 'done': True, 'total_duration':
6952996699, 'load_duration': 22514596, 'prompt_eval_count': 216, 'prompt_eva
l_duration': 2152436000, 'eval_count': 65, 'eval_duration': 4689051000}
```



```
Out[34]: ('SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\r\nFR
         OM customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP B
         Y c.FirstName, c.LastName\r\nORDER BY TotalInvoices DESC\r\nLIMIT 1',
            FirstName  LastName  TotalInvoices
          0     Aaron  Mitchell              7,
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'bingroup': 'x',
                        'hovertemplate': 'TotalInvoices=%{x}<br>count=%{y}<extra></e
         xtra>',
                        'legendgroup': '',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'type': 'histogram',
                        'x': array([7]),
                        'xaxis': 'x',
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                         'legend': {'tracegroupgap': 0},
                         'margin': {'t': 60},
                         'template': '...',
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
         ext': 'TotalInvoices'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
         ext': 'count'}}}
          }))
```

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 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 TABLE "invoice_items"\r\n(\r\n    In
voiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INT
EGER  NOT NULL,\r\n    TrackId INTEGER  NOT NULL,\r\n    UnitPrice NUMERIC(1
0,2)  NOT NULL,\r\n    Quantity INTEGER  NOT NULL,\r\n    FOREIGN KEY (Invoi
ceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDAT
E NO ACTION,\r\n    FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n
\t\tON DELETE NO ACTION ON 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" (ArtistI
d)\n\nCREATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTO
INCREMENT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate
DATETIME  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    BillingCity N
VARCHAR(40),\r\n    BillingState NVARCHAR(40),\r\n    BillingCountry NVARCHA
R(40),\r\n    BillingPostalCode NVARCHAR(10),\r\n    Total NUMERIC(10,2)  NO
T 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_Inv
oiceLineTrackId ON "invoice_items" (TrackId)\n\nCREATE INDEX IFK_InvoiceLine
InvoiceId ON "invoice_items" (InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomer
Id 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 Co
ntext \n\nIn the chinook database invoice means order\n\n===Response Guideli
nes \n1. If the provided context is sufficient, please generate a valid SQL
query without any explanations for the question. \n2. If the provided 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
question has been asked and answered before, please repeat the answer exactl
y as it was given before. \n'}, {'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 TotalInvoices\r\nFROM c
ustomers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.F
irstName, c.LastName\r\nORDER BY TotalInvoices DESC\r\nLIMIT 1'}, {'role':
'user', 'content': '  \n    Get the total number of invoices for each custom
er\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, CO
UNT(i.InvoiceId) AS TotalInvoices\r\nFROM customers c\r\nJOIN invoices i ON
c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName'}, {'role':
'user', 'content': '  \n    Find all invoices since 2010 and the total amoun
t invoiced:\n'}, {'role': 'assistant', 'content': "SELECT i.InvoiceId, SUM
(i.Total) AS TotalAmount\r\nFROM invoices i\r\nWHERE i.InvoiceDate >= '2010-
```

```
01-01'\r\nGROUP BY i.InvoiceId"}, {'role': 'user', 'content': ' \n    There
are 3 tables: artists, albums and tracks, where albums and artists are linke
d by ArtistId, albums and tracks are linked by AlbumId,\n    Can you find th
e top 10 most popular artists based on the number of tracks\n'}, {'role': 'a
ssistant', 'content': 'SELECT \r\n    "artists"."Name", \r\n    COUNT("track
s"."TrackId") AS NumTracks\r\nFROM "artists"\r\nJOIN "albums" ON "artist
s"."ArtistId" = "albums"."ArtistId"\r\nJOIN "tracks" ON "albums"."AlbumId" =
"tracks"."AlbumId"\r\nGROUP BY "artists"."Name"\r\nORDER BY NumTracks DESC
\r\nLIMIT 10'}, {'role': 'user', 'content': '  \n    Get the average invoice
total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.Fir
stName, c.LastName, AVG(i.Total) AS AvgInvoiceTotal \r\nFROM customers c\r\n
JOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.La
stName'}, {'role': 'user', 'content': '  \n    Find the total number of invo
ices per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, C
OUNT(i.InvoiceId) AS TotalInvoices \r\nFROM customers c \r\nJOIN invoices i
ON c.CustomerId = i.CustomerId \r\nGROUP BY c.Country\r\nORDER BY TotalInvoi
ces DESC'}, {'role': 'user', 'content': '  \n    Find the top 5 most expensi
ve tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELE
CT TrackId, UnitPrice\r\nFROM tracks\r\nORDER BY UnitPrice DESC\r\nLIMIT
5'}, {'role': 'user', 'content': '  \n    List all invoices with a total exc
eeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \r\nFROM invoice
s\r\nWHERE Total > 10'}, {'role': 'user', 'content': '  \n    List all album
s and their corresponding artist names  \n'}, {'role': 'assistant', 'conten
t': 'SELECT "albums".Title, "artists".Name \r\nFROM "albums"\r\nJOIN "artist
s" ON "albums"."ArtistId" = "artists"."ArtistId"\r\nORDER BY "albums"."Titl
e"'}, {'role': 'user', 'content': '  \n    List all genres and the number of
tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT "genre
s"."Name", COUNT(DISTINCT "tracks"."TrackId") AS NumTracks\r\nFROM "genre
s"\r\nLEFT JOIN "tracks" ON "genres"."GenreId" = "tracks"."GenreId"\r\nGROUP
BY "genres"."Name"\r\nORDER BY NumTracks DESC'}, {'role': 'user', 'content':
'  \n     Find the customer who bought the most albums in total quantity (ac
ross all invoices): \n'}]
Info: Ollama parameters:
model=gemma2:2b,
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(1
0,2)  NOT NULL,\r\n    FOREIGN KEY (CustomerId) REFERENCES \"customers\" (Cu
stomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IND
EX IFK_InvoiceLineTrackId ON \"invoice_items\" (TrackId)\n\nCREATE INDEX IFK
_InvoiceLineInvoiceId ON \"invoice_items\" (InvoiceId)\n\nCREATE INDEX IFK_I
nvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK_TrackAlbum
Id 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. 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 ta
ble(s). \n5. If the question has been asked and answered before, please repe
at the answer exactly as it was given before. \n"}, {"role": "user", "conten
t": "  \n     Find the customer with the most invoices \n"}, {"role": "assis
tant", "content": "SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS Tot
alInvoices\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.Custom
erId\r\nGROUP BY c.FirstName, c.LastName\r\nORDER BY TotalInvoices DESC\r\nL
IMIT 1"}, {"role": "user", "content": "  \n    Get the total number of invoi
ces for each customer\n"}, {"role": "assistant", "content": "SELECT c.FirstN
ame, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\r\nFROM customers c\r\n
JOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.La
stName"}, {"role": "user", "content": "  \n    Find all invoices since 2010
and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT
i.InvoiceId, SUM(i.Total) AS TotalAmount\r\nFROM invoices i\r\nWHERE i.Invoi
ceDate >= '2010-01-01'\r\nGROUP BY i.InvoiceId"}, {"role": "user", "conten
t": " \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 \r\n    \"artists\".\"Name\",
\r\n    COUNT(\"tracks\".\"TrackId\") AS NumTracks\r\nFROM \"artists\"\r\nJO
IN \"albums\" ON \"artists\".\"ArtistId\" = \"albums\".\"ArtistId\"\r\nJOIN
\"tracks\" ON \"albums\".\"AlbumId\" = \"tracks\".\"AlbumId\"\r\nGROUP BY
\"artists\".\"Name\"\r\nORDER BY NumTracks DESC \r\nLIMIT 10"}, {"role": "us
er", "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 AvgInvoiceTotal \r\nFROM customers c\r\nJOIN invoices i ON c.C
ustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName"}, {"role": "us
er", "content": "  \n    Find the total number of invoices per country:\n"},
{"role": "assistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS To
talInvoices \r\nFROM customers c \r\nJOIN invoices i ON c.CustomerId = i.Cus
tomerId \r\nGROUP BY c.Country\r\nORDER BY TotalInvoices DESC"}, {"role": "u
ser", "content": "  \n    Find the top 5 most expensive tracks (based on uni
t price):\n"}, {"role": "assistant", "content": "SELECT TrackId, UnitPrice\r
```

```
\nFROM tracks\r\nORDER BY UnitPrice DESC\r\nLIMIT 5"}, {"role": "user", "con
tent": "  \n    List all invoices with a total exceeding $10:\n"}, {"role":
"assistant", "content": "SELECT * \r\nFROM invoices\r\nWHERE Total > 10"},
{"role": "user", "content": "  \n    List all albums and their corresponding
artist names  \n"}, {"role": "assistant", "content": "SELECT \"albums\".Titl
e, \"artists\".Name \r\nFROM \"albums\"\r\nJOIN \"artists\" ON \"albums
\".\"ArtistId\" = \"artists\".\"ArtistId\"\r\nORDER BY \"albums\".\"Title
\""}, {"role": "user", "content": "  \n    List all genres and the number of
tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT \"genres
\".\"Name\", COUNT(DISTINCT \"tracks\".\"TrackId\") AS NumTracks\r\nFROM \"g
enres\"\r\nLEFT JOIN \"tracks\" ON \"genres\".\"GenreId\" = \"tracks\".\"Gen
reId\"\r\nGROUP BY \"genres\".\"Name\"\r\nORDER BY NumTracks DESC"}, {"rol
e": "user", "content": "  \n     Find the customer who bought the most album
s in total quantity (across all invoices): \n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:32:57.74627817Z', 'messa
ge': {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, SUM
(i.Quantity) as TotalAlbumsBought\r\nFROM customers c \r\nJOIN invoices i ON
c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName\r\nORDER BY
TotalAlbumsBought DESC\r\nLIMIT 1; \r\n'}, 'done_reason': 'stop', 'done': Tr
ue, 'total_duration': 27175782904, 'load_duration': 21405140, 'prompt_eval_c
ount': 1618, 'prompt_eval_duration': 20745715000, 'eval_count': 65, 'eval_du
ration': 5061133000}
LLM Response: SELECT c.FirstName, c.LastName, SUM(i.Quantity) as TotalAlbums
Bought
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalAlbumsBought DESC
LIMIT 1;

Info: Output from LLM: SELECT c.FirstName, c.LastName, SUM(i.Quantity) as To
talAlbumsBought
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalAlbumsBought DESC
LIMIT 1;

Extracted SQL: SELECT c.FirstName, c.LastName, SUM(i.Quantity) as TotalAlbum
sBought
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalAlbumsBought DESC
LIMIT 1
SELECT c.FirstName, c.LastName, SUM(i.Quantity) as TotalAlbumsBought
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalAlbumsBought DESC
LIMIT 1
Couldn't run sql:  Execution failed on sql 'SELECT c.FirstName, c.LastName,
SUM(i.Quantity) as TotalAlbumsBought
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
```

```
GROUP BY c.FirstName, c.LastName
ORDER BY TotalAlbumsBought DESC
LIMIT 1': no such column: i.Quantity
```

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 ON
LY be based on the given context and follow the response guidelines and form
at instructions. \n===Tables \nCREATE TABLE "invoice_items"\r\n(\r\n    Invo
iceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    InvoiceId INTEG
ER  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 (Invoice
Id) 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 NVA
RCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    MediaTypeId INTEGER  NO
T NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(220),\r\n    Millis
econds 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 ACTIO
N,\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 "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_AlbumArtistId ON "albums" (Ar
tistId)\n\nCREATE INDEX IFK_InvoiceLineInvoiceId ON "invoice_items" (Invoice
Id)\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nC
REATE TABLE "invoices"\r\n(\r\n    InvoiceId INTEGER PRIMARY KEY AUTOINCREME
NT NOT NULL,\r\n    CustomerId INTEGER  NOT NULL,\r\n    InvoiceDate DATETIM
E  NOT NULL,\r\n    BillingAddress NVARCHAR(70),\r\n    BillingCity NVARCHAR
(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_Invoi
ceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK_TrackAlbumId ON
"tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n    ArtistId INTEGER PR
IMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\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
Find the customer with the most invoices \n'}, {'role': 'assistant', 'conten
t': 'SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\r\n
FROM customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP
BY c.FirstName, c.LastName\r\nORDER BY TotalInvoices DESC\r\nLIMIT 1'}, {'ro
le': 'user', 'content': ' \n    There are 3 tables: artists, albums and trac
ks, where albums and artists are linked by ArtistId, albums and tracks are l
inked by AlbumId,\n    Can you find the top 10 most popular artists based on
the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT \r\n
"artists"."Name", \r\n    COUNT("tracks"."TrackId") AS NumTracks\r\nFROM "ar
tists"\r\nJOIN "albums" ON "artists"."ArtistId" = "albums"."ArtistId"\r\nJOI
N "tracks" ON "albums"."AlbumId" = "tracks"."AlbumId"\r\nGROUP BY "artist
```

```
s"."Name"\r\nORDER BY NumTracks DESC \r\nLIMIT 10'}, {'role': 'user', 'conte
nt': '  \n    Find the top 5 most expensive tracks (based on unit pric
e):\n'}, {'role': 'assistant', 'content': 'SELECT TrackId, UnitPrice\r\nFROM
tracks\r\nORDER BY UnitPrice DESC\r\nLIMIT 5'}, {'role': 'user', 'content':
'  \n    Get the total number of invoices for each customer\n'}, {'role': 'a
ssistant', 'content': 'SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS
TotalInvoices\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.Cus
tomerId\r\nGROUP BY c.FirstName, c.LastName'}, {'role': 'user', 'content': '
\n    Find all invoices since 2010 and the total amount invoiced:\n'}, {'rol
e': 'assistant', 'content': "SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount
\r\nFROM invoices i\r\nWHERE i.InvoiceDate >= '2010-01-01'\r\nGROUP BY i.Inv
oiceId"}, {'role': 'user', 'content': '  \n    Get the average invoice total
for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstNam
e, c.LastName, AVG(i.Total) AS AvgInvoiceTotal \r\nFROM customers c\r\nJOIN
invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastNam
e'}, {'role': 'user', 'content': '  \n    Find the total number of invoices
per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT
(i.InvoiceId) AS TotalInvoices \r\nFROM customers c \r\nJOIN invoices i ON
c.CustomerId = i.CustomerId \r\nGROUP BY c.Country\r\nORDER BY TotalInvoices
DESC'}, {'role': 'user', 'content': '  \n    List all invoices with a total
exceeding $10:\n'}, {'role': 'assistant', 'content': 'SELECT * \r\nFROM invo
ices\r\nWHERE Total > 10'}, {'role': 'user', 'content': 'what are the top 5
countries that customers come from?'}, {'role': 'assistant', 'content': 'SEL
ECT Country, COUNT(DISTINCT CustomerId) AS NumCustomers\r\nFROM customers\r
\nGROUP BY Country\r\nORDER BY NumCustomers DESC\r\nLIMIT 5'}, {'role': 'use
r', 'content': '  \n    List all albums and their corresponding artist names
\n'}, {'role': 'assistant', 'content': 'SELECT "albums".Title, "artists".Nam
e \r\nFROM "albums"\r\nJOIN "artists" ON "albums"."ArtistId" = "artists"."Ar
tistId"\r\nORDER BY "albums"."Title"'}, {'role': 'user', 'content': '  \n
Hint: album quantity is found in invoice_items, \n    \n    Find the top 5 c
ustomers who bought the most albums in total quantity (across all invoice
s):\n'}]
Info: Ollama parameters:
model=gemma2:2b,
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 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
\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFE
RENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO
N,\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 \"album
```

```
s\"\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 with the most invoices \n"}, {"role": "assi
stant", "content": "SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS To
talInvoices\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.Custo
merId\r\nGROUP BY c.FirstName, c.LastName\r\nORDER BY TotalInvoices DESC\r\n
LIMIT 1"}, {"role": "user", "content": " \n    There are 3 tables: artists,
albums and tracks, where albums and artists are linked by ArtistId, albums a
nd tracks are linked by AlbumId,\n    Can you find the top 10 most popular a
rtists based on the number of tracks\n"}, {"role": "assistant", "content":
"SELECT \r\n    \"artists\".\"Name\", \r\n    COUNT(\"tracks\".\"TrackId\")
AS NumTracks\r\nFROM \"artists\"\r\nJOIN \"albums\" ON \"artists\".\"ArtistI
d\" = \"albums\".\"ArtistId\"\r\nJOIN \"tracks\" ON \"albums\".\"AlbumId\" =
\"tracks\".\"AlbumId\"\r\nGROUP BY \"artists\".\"Name\"\r\nORDER BY NumTrack
s DESC \r\nLIMIT 10"}, {"role": "user", "content": "  \n    Find the top 5 m
ost expensive tracks (based on unit price):\n"}, {"role": "assistant", "cont
ent": "SELECT TrackId, UnitPrice\r\nFROM tracks\r\nORDER BY UnitPrice DESC\r
\nLIMIT 5"}, {"role": "user", "content": "  \n    Get the total number of in
voices for each customer\n"}, {"role": "assistant", "content": "SELECT c.Fir
stName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\r\nFROM customers c
\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName,
c.LastName"}, {"role": "user", "content": "  \n    Find all invoices since 2
010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SE
LECT i.InvoiceId, SUM(i.Total) AS TotalAmount\r\nFROM invoices i\r\nWHERE i.
InvoiceDate >= '2010-01-01'\r\nGROUP BY i.InvoiceId"}, {"role": "user", "con
tent": "  \n    Get the average invoice total for each customer:\n"}, {"rol
e": "assistant", "content": "SELECT c.FirstName, c.LastName, AVG(i.Total) AS
AvgInvoiceTotal \r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.
CustomerId\r\nGROUP BY c.FirstName, c.LastName"}, {"role": "user", "conten
t": "  \n    Find the total number of invoices per country:\n"}, {"role": "a
ssistant", "content": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices
\r\nFROM customers c \r\nJOIN invoices i ON c.CustomerId = i.CustomerId \r\n
```

```
GROUP BY c.Country\r\nORDER BY TotalInvoices DESC"}, {"role": "user", "conte
nt": "  \n    List all invoices with a total exceeding $10:\n"}, {"role": "a
ssistant", "content": "SELECT * \r\nFROM invoices\r\nWHERE Total > 10"}, {"r
ole": "user", "content": "what are the top 5 countries that customers come f
rom?"}, {"role": "assistant", "content": "SELECT Country, COUNT(DISTINCT Cus
tomerId) AS NumCustomers\r\nFROM customers\r\nGROUP BY Country\r\nORDER BY N
umCustomers DESC\r\nLIMIT 5"}, {"role": "user", "content": "  \n    List all
albums and their corresponding artist names  \n"}, {"role": "assistant", "co
ntent": "SELECT \"albums\".Title, \"artists\".Name \r\nFROM \"albums\"\r\nJO
IN \"artists\" ON \"albums\".\"ArtistId\" = \"artists\".\"ArtistId\"\r\nORDE
R BY \"albums\".\"Title\""}, {"role": "user", "content": "  \n    Hint: albu
m quantity is found in invoice_items, \n    \n    Find the top 5 customers w
ho bought the most albums in total quantity (across all invoices):\n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:33:41.44885287Z', 'messa
ge': {'role': 'assistant', 'content': "```sql\nSELECT c.FirstName, c.LastNam
e, COUNT(DISTINCT i.InvoiceId) AS TotalAlbumsBought\r\nFROM customers c\r\nJ
OIN invoices i ON c.CustomerId = i.CustomerId\r\nLEFT JOIN invoice_items ii
ON i.InvoiceId = ii.InvoiceId\r\nGROUP BY c.FirstName, c.LastName\r\nORDER B
Y TotalAlbumsBought DESC \r\nLIMIT 5;\n```\n\n**Explanation:**\n\n1. **JOIN
s:** We use `JOIN` statements to connect the `customers`, `invoices`, and `i
nvoice_items` tables.\n    *  `JOIN invoices i ON c.CustomerId = i.CustomerI
d`: Connects customers with their corresponding invoices.\n    *  `LEFT JOIN
invoice_items ii ON i.InvoiceId = ii.InvoiceId`: Links invoices to the items
in each invoice to get quantity details. \r\n2. **Count:** We use `COUNT(DIS
TINCT i.InvoiceId)` to count unique albums purchased by customers.\r\n3. **G
ROUP BY:** We group by customer's first and last names (`c.FirstName, c.Last
Name`) for clear results.\r\n4. **ORDER BY:**  We order the result set based
on the total number of albums purchased by each customer in descending orde
r.\r\n5. **LIMIT 5:**  We limit the output to the top 5 customers with the m
ost album purchases. \r\n"}, 'done_reason': 'stop', 'done': True, 'total_dur
ation': 43660497553, 'load_duration': 20661617, 'prompt_eval_count': 1607,
'prompt_eval_duration': 18688736000, 'eval_count': 295, 'eval_duration': 235
75878000}
LLM Response: ```sql
SELECT c.FirstName, c.LastName, COUNT(DISTINCT i.InvoiceId) AS TotalAlbumsBo
ught
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
LEFT JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalAlbumsBought DESC
LIMIT 5;
```

**Explanation:**

1. **JOINs:** We use `JOIN` statements to connect the `customers`, `invoices
`, and `invoice_items` tables.
    *  `JOIN invoices i ON c.CustomerId = i.CustomerId`: Connects customers
with their corresponding invoices.
    *  `LEFT JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId`: Links inv
oices to the items in each invoice to get quantity details.
2. **Count:** We use `COUNT(DISTINCT i.InvoiceId)` to count unique albums pu
rchased by customers.
3. **GROUP BY:** We group by customer's first and last names (`c.FirstName,
```
```

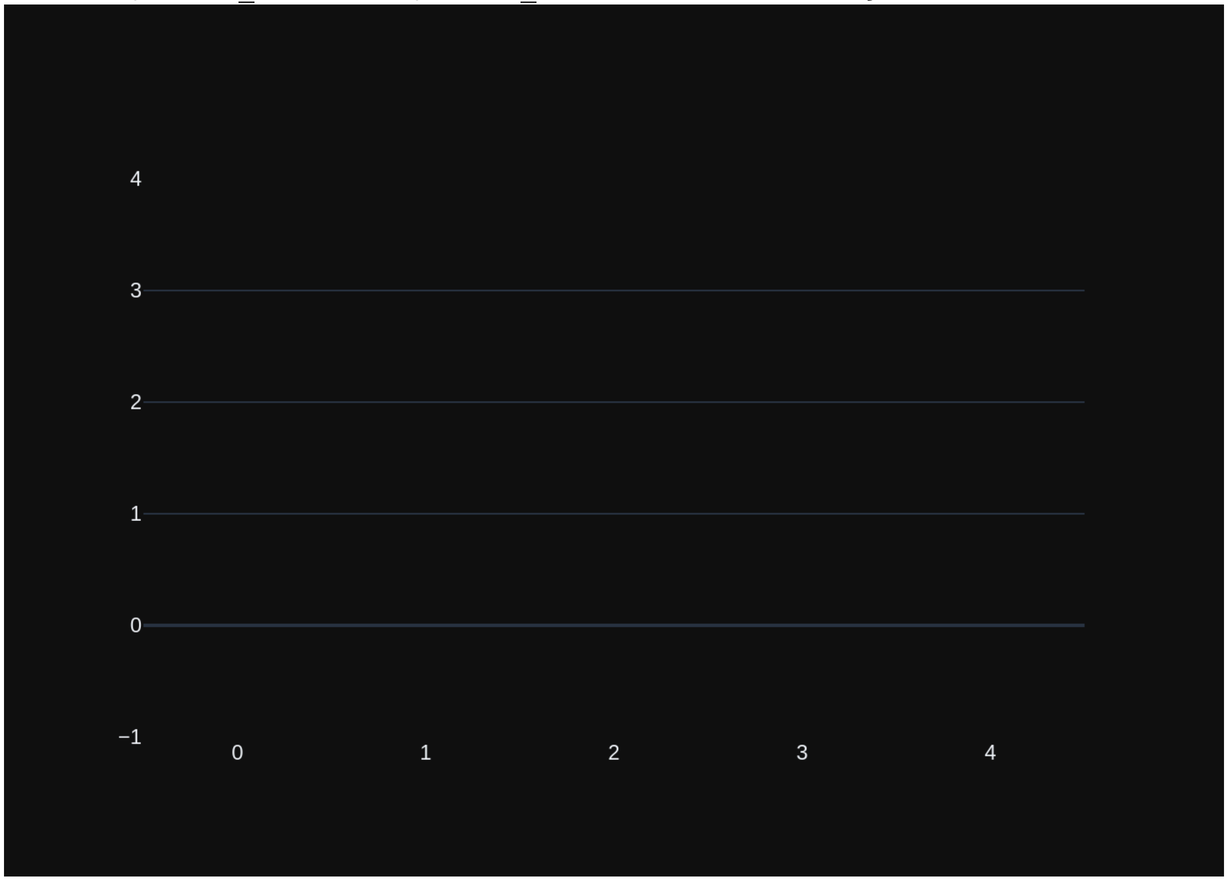
```
c.LastName`) for clear results.
4. **ORDER BY:**  We order the result set based on the total number of album
s purchased by each customer in descending order.
5. **LIMIT 5:**  We limit the output to the top 5 customers with the most al
bum purchases.

Info: Output from LLM: ```sql
SELECT c.FirstName, c.LastName, COUNT(DISTINCT i.InvoiceId) AS TotalAlbumsBo
ught
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
LEFT JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalAlbumsBought DESC
LIMIT 5;
```

**Explanation:**

1. **JOINs:** We use `JOIN` statements to connect the `customers`, `invoices
`, and `invoice_items` tables.
    *  `JOIN invoices i ON c.CustomerId = i.CustomerId`: Connects customers
with their corresponding invoices.
    *  `LEFT JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId`: Links inv
oices to the items in each invoice to get quantity details.
2. **Count:** We use `COUNT(DISTINCT i.InvoiceId)` to count unique albums pu
rchased by customers.
3. **GROUP BY:** We group by customer's first and last names (`c.FirstName,
c.LastName`) for clear results.
4. **ORDER BY:**  We order the result set based on the total number of album
s purchased by each customer in descending order.
5. **LIMIT 5:**  We limit the output to the top 5 customers with the most al
bum purchases.

Extracted SQL: SELECT c.FirstName, c.LastName, COUNT(DISTINCT i.InvoiceId) A
S TotalAlbumsBought
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
LEFT JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalAlbumsBought DESC
LIMIT 5
SELECT c.FirstName, c.LastName, COUNT(DISTINCT i.InvoiceId) AS TotalAlbumsBo
ught
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
LEFT JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalAlbumsBought DESC
LIMIT 5
   FirstName  LastName  TotalAlbumsBought
0      Aaron  Mitchell                  7
1  Alexandre     Rocha                  7
2     Astrid    Gruber                  7
3      Bjørn    Hansen                  7
4    Camille   Bernard                  7
```
```

```
Info: Ollama parameters:
model=gemma2:2b,
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, COUNT(DISTINCT i.InvoiceId) AS TotalAlbumsBought\r\nFROM cust
omers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nLEFT JOIN invoi
ce_items ii ON i.InvoiceId = ii.InvoiceId\r\nGROUP BY c.FirstName, c.LastNam
e\r\nORDER BY TotalAlbumsBought DESC \r\nLIMIT 5\n\nThe following is informa
tion about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n
FirstName           object\nLastName            object\nTotalAlbumsBought
int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Py
thon plotly code to chart the results of the dataframe? Assume the data is i
n a pandas dataframe called 'df'. If there is only one value in the datafram
e, use an Indicator. Respond with only Python code. Do not answer with any e
xplanations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:33:47.731164442Z', 'mess
age': {'role': 'assistant', 'content': "```python\nfig = go.Figure(data=go.B
ar(x=[i for i in df.index],y=[df['TotalAlbumsBought']])) \nfig.show()\n``
`"}, 'done_reason': 'stop', 'done': True, 'total_duration': 6254496334, 'loa
d_duration': 17924196, 'prompt_eval_count': 266, 'prompt_eval_duration': 308
9104000, 'eval_count': 43, 'eval_duration': 3054037000}
```



Out[36]:
```
('SELECT c.FirstName, c.LastName, COUNT(DISTINCT i.InvoiceId) AS TotalAlbum
sBought\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.Customer
Id\r\nLEFT JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\r\nGROUP BY
c.FirstName, c.LastName\r\nORDER BY TotalAlbumsBought DESC \r\nLIMIT 5',
    FirstName  LastName  TotalAlbumsBought
 0      Aaron  Mitchell                  7
 1  Alexandre     Rocha                  7
 2     Astrid    Gruber                  7
 3      Bjørn    Hansen                  7
 4    Camille   Bernard                  7,
 Figure({
     'data': [{'type': 'bar', 'x': [0, 1, 2, 3, 4], 'y': [[7, 7, 7, 7,
7]]}], 'layout': {'template': '...'}
 }))
```

```
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 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_InvoiceLineInvoiceId ON "invoice_items"
(InvoiceId)\n\nCREATE INDEX IFK_InvoiceCustomerId ON "invoices" (CustomerId)
\n\nCREATE INDEX IFK_InvoiceLineTrackId ON "invoice_items" (TrackId)\n\nCREA
TE TABLE "customers"\r\n(\r\n    CustomerId INTEGER PRIMARY KEY AUTOINCREMEN
T NOT NULL,\r\n    FirstName NVARCHAR(40)  NOT NULL,\r\n    LastName NVARCHA
R(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 "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 TA
BLE "tracks"\r\n(\r\n    TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL
L,\r\n    Name NVARCHAR(200)  NOT NULL,\r\n    AlbumId INTEGER,\r\n    Media
TypeId INTEGER  NOT NULL,\r\n    GenreId INTEGER,\r\n    Composer NVARCHAR(2
20),\r\n    Milliseconds INTEGER  NOT NULL,\r\n    Bytes INTEGER,\r\n    Uni
tPrice NUMERIC(10,2)  NOT NULL,\r\n    FOREIGN KEY (AlbumId) REFERENCES "alb
ums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREI
GN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION O
N UPDATE NO ACTION,\r\n    FOREIGN KEY (MediaTypeId) REFERENCES "media_type
s" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCRE
ATE TABLE "playlist_track"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n
TrackId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY
(PlaylistId, TrackId),\r\n    FOREIGN KEY (PlaylistId) REFERENCES "playlist
s" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FORE
IGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION
ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_EmployeeReportsTo ON "employee
s" (ReportsTo)\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
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, COUNT(DISTINCT i.InvoiceId) AS TotalAlbumsBought\r
\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nLEFT
JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\r\nGROUP BY c.FirstName,
c.LastName\r\nORDER BY TotalAlbumsBought DESC \r\nLIMIT 5'}, {'role': 'use
r', 'content': '  \n     Find the customer with the most invoices \n'}, {'ro
le': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, COUNT(i.Invoic
eId) AS TotalInvoices\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId
= i.CustomerId\r\nGROUP BY c.FirstName, c.LastName\r\nORDER BY TotalInvoices
DESC\r\nLIMIT 1'}, {'role': 'user', 'content': '  \n    Get the average invo
ice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT
c.FirstName, c.LastName, AVG(i.Total) AS AvgInvoiceTotal \r\nFROM customers
c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName,
c.LastName'}, {'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\r\nFROM customers c
\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName,
c.LastName'}, {'role': 'user', 'content': '  \n    Find the total number of
invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT c.Countr
y, COUNT(i.InvoiceId) AS TotalInvoices \r\nFROM customers c \r\nJOIN invoice
s i ON c.CustomerId = i.CustomerId \r\nGROUP BY c.Country\r\nORDER BY TotalI
nvoices DESC'}, {'role': 'user', 'content': '  \n    Find all invoices since
2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': "S
ELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\r\nFROM invoices i\r\nWHERE
i.InvoiceDate >= '2010-01-01'\r\nGROUP BY i.InvoiceId"}, {'role': 'user', 'c
ontent': '  \n    Find the top 5 most expensive tracks (based on unit pric
e):\n'}, {'role': 'assistant', 'content': 'SELECT TrackId, UnitPrice\r\nFROM
tracks\r\nORDER BY UnitPrice DESC\r\nLIMIT 5'}, {'role': 'user', 'content':
'  \n    List all invoices with a total exceeding $10:\n'}, {'role': 'assist
ant', 'content': 'SELECT * \r\nFROM invoices\r\nWHERE Total > 10'}, {'role':
'user', 'content': 'what are the top 5 countries that customers come fro
m?'}, {'role': 'assistant', 'content': 'SELECT Country, COUNT(DISTINCT Custo
merId) AS NumCustomers\r\nFROM customers\r\nGROUP BY Country\r\nORDER BY Num
Customers DESC\r\nLIMIT 5'}, {'role': 'user', 'content': ' \n    There are 3
tables: artists, albums and tracks, where albums and artists are linked by A
rtistId, 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': 'assista
nt', 'content': 'SELECT \r\n    "artists"."Name", \r\n    COUNT("tracks"."Tr
ackId") AS NumTracks\r\nFROM "artists"\r\nJOIN "albums" ON "artists"."Artist
Id" = "albums"."ArtistId"\r\nJOIN "tracks" ON "albums"."AlbumId" = "track
s"."AlbumId"\r\nGROUP BY "artists"."Name"\r\nORDER BY NumTracks DESC \r\nLIM
IT 10'}, {'role': 'user', 'content': '  \n     Find the top 5 customers who
spent the most money overall, \n     \n     Hint: order total can be found o
n invoices table, calculation using invoice_items detail table is unnecessar
y \n'}]
Info: Ollama parameters:
model=gemma2:2b,
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
```

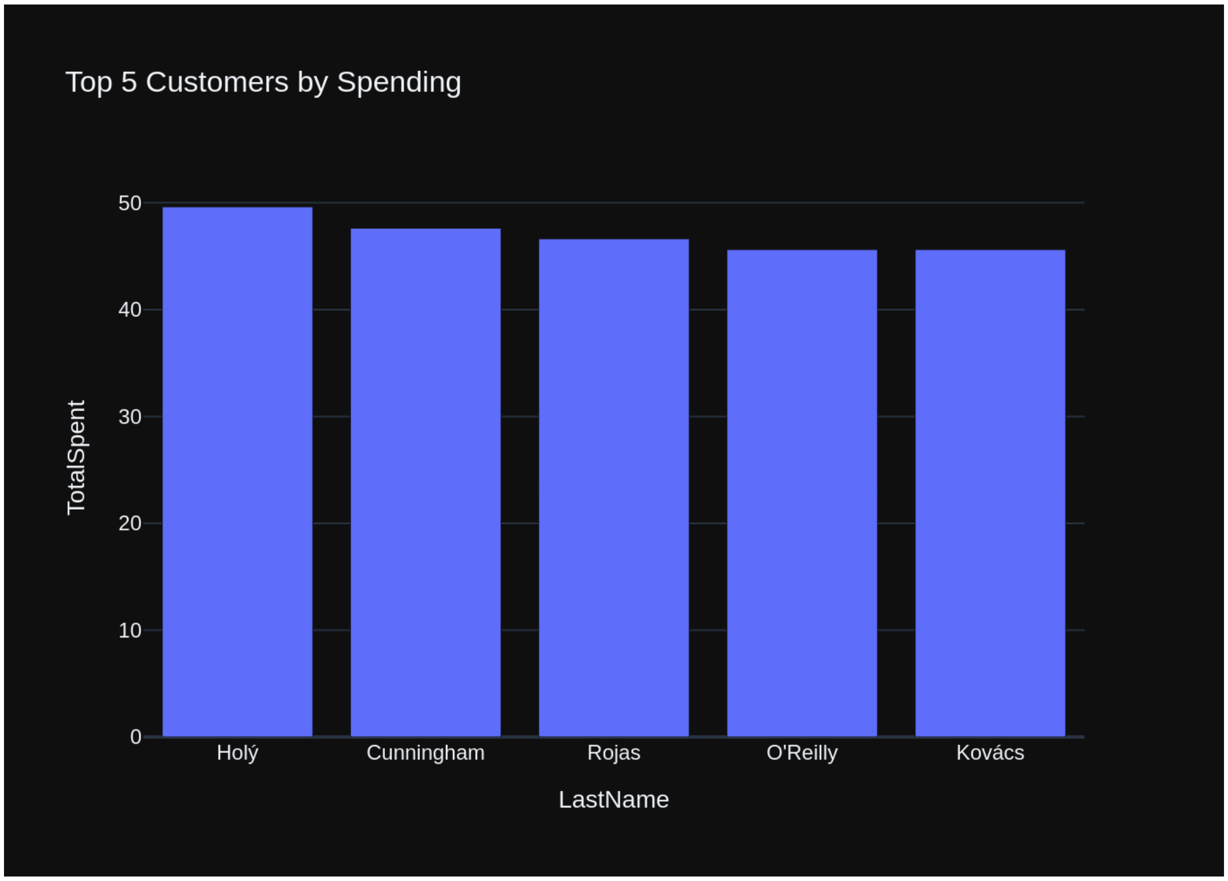
```
column. Prepend the query with a comment saying intermediate_sql \n3. If the
provided context is insufficient, please explain why it can't be generated.
\n4. Please use the most relevant table(s). \n5. If the question has been as
ked and answered before, please repeat the answer exactly as it was given be
fore. \n"}, {"role": "user", "content": "  \n    Hint: album quantity is fou
nd in invoice_items, \n    \n    Find the top 5 customers who bought the mos
t albums in total quantity (across all invoices):\n"}, {"role": "assistant",
"content": "SELECT c.FirstName, c.LastName, COUNT(DISTINCT i.InvoiceId) AS T
otalAlbumsBought\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.
CustomerId\r\nLEFT JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\r\nGR
OUP BY c.FirstName, c.LastName\r\nORDER BY TotalAlbumsBought DESC \r\nLIMIT
5"}, {"role": "user", "content": "  \n     Find the customer with the most i
nvoices \n"}, {"role": "assistant", "content": "SELECT c.FirstName, c.LastNa
me, COUNT(i.InvoiceId) AS TotalInvoices\r\nFROM customers c\r\nJOIN invoices
i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName\r\nORDE
R BY TotalInvoices DESC\r\nLIMIT 1"}, {"role": "user", "content": "  \n    G
et the average invoice total for each customer:\n"}, {"role": "assistant",
"content": "SELECT c.FirstName, c.LastName, AVG(i.Total) AS AvgInvoiceTotal
\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGR
OUP BY c.FirstName, c.LastName"}, {"role": "user", "content": "  \n    Get t
he total number of invoices for each customer\n"}, {"role": "assistant", "co
ntent": "SELECT c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices
\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGR
OUP BY c.FirstName, c.LastName"}, {"role": "user", "content": "  \n    Find
the total number of invoices per country:\n"}, {"role": "assistant", "conten
t": "SELECT c.Country, COUNT(i.InvoiceId) AS TotalInvoices \r\nFROM customer
s c \r\nJOIN invoices i ON c.CustomerId = i.CustomerId \r\nGROUP BY c.Countr
y\r\nORDER BY TotalInvoices DESC"}, {"role": "user", "content": "  \n    Fin
d all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assi
stant", "content": "SELECT i.InvoiceId, SUM(i.Total) AS TotalAmount\r\nFROM
invoices i\r\nWHERE i.InvoiceDate >= '2010-01-01'\r\nGROUP BY i.InvoiceId"},
{"role": "user", "content": "  \n    Find the top 5 most expensive tracks (b
ased on unit price):\n"}, {"role": "assistant", "content": "SELECT TrackId,
UnitPrice\r\nFROM tracks\r\nORDER BY UnitPrice DESC\r\nLIMIT 5"}, {"role":
"user", "content": "  \n    List all invoices with a total exceeding $1
0:\n"}, {"role": "assistant", "content": "SELECT * \r\nFROM invoices\r\nWHER
E Total > 10"}, {"role": "user", "content": "what are the top 5 countries th
at customers come from?"}, {"role": "assistant", "content": "SELECT Country,
COUNT(DISTINCT CustomerId) AS NumCustomers\r\nFROM customers\r\nGROUP BY Cou
ntry\r\nORDER BY NumCustomers DESC\r\nLIMIT 5"}, {"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"}, {"role": "assistant", "content": "SELECT \r\n    \"artists\".\"Name\",
\r\n    COUNT(\"tracks\".\"TrackId\") AS NumTracks\r\nFROM \"artists\"\r\nJO
IN \"albums\" ON \"artists\".\"ArtistId\" = \"albums\".\"ArtistId\"\r\nJOIN
\"tracks\" ON \"albums\".\"AlbumId\" = \"tracks\".\"AlbumId\"\r\nGROUP BY
\"artists\".\"Name\"\r\nORDER BY NumTracks DESC \r\nLIMIT 10"}, {"role": "us
er", "content": "  \n     Find the top 5 customers who spent the most money
overall, \n     \n     Hint: order total can be found on invoices table, cal
culation using invoice_items detail table is unnecessary \n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:34:28.757139072Z', 'mess
age': {'role': 'assistant', 'content': "```sql\nSELECT c.FirstName, c.LastNa
me, SUM(i.Total) AS TotalSpent\r\nFROM customers c\r\nJOIN invoices i ON c.C
ustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName\r\nORDER BY Tot
```

```
alSpent DESC \r\nLIMIT 5;\n``` \n\n\nThis query first joins the `customers`
and `invoices` tables based on the `CustomerId` field to link customer data
with their corresponding invoice totals. It then uses the `SUM(i.Total)` fun
ction in a subquery to aggregate the total amount spent by each customer acr
oss all their invoices. Finally, it groups the results by customer's first a
nd last names, sorts them in descending order based on the total spent (`Tot
alSpent`) and limits the output to the top 5 customers with highest spendin
g. \n"}, 'done_reason': 'stop', 'done': True, 'total_duration': 40945595353,
'load_duration': 24055333, 'prompt_eval_count': 1999, 'prompt_eval_duratio
n': 25164681000, 'eval_count': 173, 'eval_duration': 14321091000}
LLM Response: ```sql
SELECT c.FirstName, c.LastName, SUM(i.Total) AS TotalSpent
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalSpent DESC
LIMIT 5;
```

This query first joins the `customers` and `invoices` tables based on the `C
ustomerId` field to link customer data with their corresponding invoice tota
ls. It then uses the `SUM(i.Total)` function in a subquery to aggregate the
total amount spent by each customer across all their invoices. Finally, it g
roups the results by customer's first and last names, sorts them in descendi
ng order based on the total spent (`TotalSpent`) and limits the output to th
e top 5 customers with highest spending.

Info: Output from LLM: ```sql
SELECT c.FirstName, c.LastName, SUM(i.Total) AS TotalSpent
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalSpent DESC
LIMIT 5;
```


This query first joins the `customers` and `invoices` tables based on the `C
ustomerId` field to link customer data with their corresponding invoice tota
ls. It then uses the `SUM(i.Total)` function in a subquery to aggregate the
total amount spent by each customer across all their invoices. Finally, it g
roups the results by customer's first and last names, sorts them in descendi
ng order based on the total spent (`TotalSpent`) and limits the output to th
e top 5 customers with highest spending.

Extracted SQL: SELECT c.FirstName, c.LastName, SUM(i.Total) AS TotalSpent
FROM customers c
JOIN invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName
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.FirstName, c.LastName
```

```
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      Hugh    O'Reilly       45.62
4  Ladislav      Kovács       45.62
Info: Ollama parameters:
model=gemma2:2b,
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\r\nFROM
customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.
FirstName, c.LastName\r\nORDER BY TotalSpent DESC \r\nLIMIT 5\n\nThe followi
ng is information about the resulting pandas DataFrame 'df': \nRunning df.dt
ypes gives:\n FirstName      object\nLastName       object\nTotalSpent    fl
oat64\ndtype: object"}, {"role": "user", "content": "Can you generate the Py
thon plotly code to chart the results of the dataframe? Assume the data is i
n a pandas dataframe called 'df'. If there is only one value in the datafram
e, use an Indicator. Respond with only Python code. Do not answer with any e
xplanations -- just the code."}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:34:34.572906772Z', 'mess
age': {'role': 'assistant', 'content': '```python\nimport plotly.express as
px\n\nfig = px.bar(df, x="LastName", y="TotalSpent", title="Top 5 Customers
by Spending")  \nfig.show() \n```'}, 'done_reason': 'stop', 'done': True, 't
otal_duration': 5787719198, 'load_duration': 19206790, 'prompt_eval_count':
246, 'prompt_eval_duration': 2416594000, 'eval_count': 46, 'eval_duration':
3264151000}
```


Top 5 Customers by Spending
TotalSpent
50
40
30
20
10
0
Holý
Cunningham
Rojas
O'Reilly
Kovács
LastName

Out[37]:
```
('SELECT c.FirstName, c.LastName, SUM(i.Total) AS TotalSpent\r\nFROM custom
ers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.First
Name, c.LastName\r\nORDER BY TotalSpent DESC \r\nLIMIT 5',
   FirstName    LastName  TotalSpent
 0    Helena        Holý       49.62
 1   Richard  Cunningham       47.62
 2      Luis       Rojas       46.62
 3      Hugh    O'Reilly       45.62
 4  Ladislav      Kovács       45.62,
 Figure({
     'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'LastName=%{x}<br>TotalSpent=%{y}<extra></e
xtra>',
               'legendgroup': '',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': '',
               'offsetgroup': '',
               'orientation': 'v',
               'showlegend': False,
               'textposition': 'auto',
               'type': 'bar',
               'x': array(['Holý', 'Cunningham', 'Rojas', "O'Reilly", 'Ková
cs'], dtype=object),
               'xaxis': 'x',
               'y': array([49.62, 47.62, 46.62, 45.62, 45.62]),
               'yaxis': 'y'}],
     'layout': {'barmode': 'relative',
                'legend': {'tracegroupgap': 0},
                'template': '...',
                'title': {'text': 'Top 5 Customers by Spending'},
                'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
ext': 'LastName'}},
                '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 ON
LY be based on the given context and follow the response guidelines and form
at instructions. \n===Tables \nCREATE INDEX IFK_PlaylistTrackTrackId ON "pla
ylist_track" (TrackId)\n\nCREATE TABLE "playlists"\r\n(\r\n    PlaylistId IN
TEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\n
CREATE 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    FORE
IGN 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 INTEGE
R 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 NUL
L,\r\n    Bytes INTEGER,\r\n    UnitPrice NUMERIC(10,2)  NOT NULL,\r\n    FO
REIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTIO
N ON UPDATE NO ACTION,\r\n    FOREIGN KEY (GenreId) REFERENCES "genres" (Gen
reId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n    FOREIGN KEY (M
ediaTypeId) REFERENCES "media_types" (MediaTypeId) \r\n\t\tON DELETE NO ACTI
ON ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK_TrackGenreId ON "tracks" (Ge
nreId)\n\nCREATE INDEX IFK_TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDE
X IFK_TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE INDEX IFK_AlbumAr
tistId ON "albums" (ArtistId)\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 TABLE "genres"\r\n(\r\n    GenreId INTEGER PRIMARY KEY AUTOINC
REMENT 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 "genres"."Name", COUNT(DISTINCT "tracks"."TrackId") AS NumTracks
\r\nFROM "genres"\r\nLEFT JOIN "tracks" ON "genres"."GenreId" = "tracks"."Ge
nreId"\r\nGROUP BY "genres"."Name"\r\nORDER BY NumTracks DESC'}, {'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 \r\n    "artist
s"."Name", \r\n    COUNT("tracks"."TrackId") AS NumTracks\r\nFROM "artist
s"\r\nJOIN "albums" ON "artists"."ArtistId" = "albums"."ArtistId"\r\nJOIN "t
racks" ON "albums"."AlbumId" = "tracks"."AlbumId"\r\nGROUP BY "artists"."Nam
e"\r\nORDER BY NumTracks DESC \r\nLIMIT 10'}, {'role': 'user', 'content': '
\n    Find all tracks with a name containing "What" (case-insensitive)\n'},
{'role': 'assistant', 'content': "SELECT * \r\nFROM tracks\r\nWHERE Name LIK
E '%what%'\r\nORDER BY Name"}, {'role': 'user', 'content': '  \n    Find the
top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistan
t', 'content': 'SELECT TrackId, UnitPrice\r\nFROM tracks\r\nORDER BY UnitPri
```

```
ce DESC\r\nLIMIT 5'}, {'role': 'user', 'content': '  \n    List all albums a
nd their corresponding artist names  \n'}, {'role': 'assistant', 'content':
'SELECT "albums".Title, "artists".Name \r\nFROM "albums"\r\nJOIN "artists" O
N "albums"."ArtistId" = "artists"."ArtistId"\r\nORDER BY "albums"."Title"'},
{'role': 'user', 'content': '  \n    Hint: album quantity is found in invoic
e_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, COUNT(DISTINCT i.InvoiceId) AS TotalAlbumsB
ought\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId
\r\nLEFT JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\r\nGROUP BY c.F
irstName, c.LastName\r\nORDER BY TotalAlbumsBought DESC \r\nLIMIT 5'}, {'rol
e': 'user', 'content': '  \n    Find all invoices since 2010 and the total a
mount invoiced:\n'}, {'role': 'assistant', 'content': "SELECT i.InvoiceId, S
UM(i.Total) AS TotalAmount\r\nFROM invoices i\r\nWHERE i.InvoiceDate >= '201
0-01-01'\r\nGROUP BY i.InvoiceId"}, {'role': 'user', 'content': 'Can you lis
t all tables in the SQLite database catalog?'}, {'role': 'assistant', 'conte
nt': "SELECT name FROM sqlite_schema WHERE type='table'"}, {'role': 'user',
'content': '  \n    List all invoices with a total exceeding $10:\n'}, {'rol
e': 'assistant', 'content': 'SELECT * \r\nFROM invoices\r\nWHERE Total > 1
0'}, {'role': 'user', 'content': '  \n     Find the top 5 customers who spen
t the most money overall, \n     \n     Hint: order total can be found on in
voices table, calculation using invoice_items detail table is unnecessary
\n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.LastName, SUM
(i.Total) AS TotalSpent\r\nFROM customers c\r\nJOIN invoices i ON c.Customer
Id = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName\r\nORDER BY TotalSpent
DESC \r\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=gemma2:2b,
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_PlaylistTrackTrackId ON \"playlist_track
\" (TrackId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n    PlaylistId INTEGER PR
IMARY KEY AUTOINCREMENT NOT NULL,\r\n    Name NVARCHAR(120)\r\n)\n\nCREATE T
ABLE \"playlist_track\"\r\n(\r\n    PlaylistId INTEGER  NOT NULL,\r\n    Tra
ckId INTEGER  NOT NULL,\r\n    CONSTRAINT PK_PlaylistTrack PRIMARY KEY  (Pla
ylistId, 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 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 INDEX IFK_TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDE
X IFK_TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE INDEX IFK_Album
ArtistId 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) R
EFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO AC
TION\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 Co
ntext \n\nIn the chinook database invoice means order\n\n===Response Guideli
nes \n1. If the provided context is sufficient, please generate a valid SQL
query without any explanations for the question. \n2. If the provided 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": "  \n    List all
genres and the number of tracks in each genre:\n"}, {"role": "assistant", "c
ontent": "SELECT \"genres\".\"Name\", COUNT(DISTINCT \"tracks\".\"TrackId\")
AS NumTracks\r\nFROM \"genres\"\r\nLEFT JOIN \"tracks\" ON \"genres\".\"Genr
eId\" = \"tracks\".\"GenreId\"\r\nGROUP BY \"genres\".\"Name\"\r\nORDER BY N
umTracks DESC"}, {"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 \r\n    \"artists\".\"Name\", \r\n    COUNT(\"tracks\".\"TrackI
d\") AS NumTracks\r\nFROM \"artists\"\r\nJOIN \"albums\" ON \"artists\".\"Ar
tistId\" = \"albums\".\"ArtistId\"\r\nJOIN \"tracks\" ON \"albums\".\"AlbumI
d\" = \"tracks\".\"AlbumId\"\r\nGROUP BY \"artists\".\"Name\"\r\nORDER BY Nu
mTracks DESC \r\nLIMIT 10"}, {"role": "user", "content": "  \n    Find all t
racks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assi
stant", "content": "SELECT * \r\nFROM tracks\r\nWHERE Name LIKE '%what%'\r\n
ORDER BY Name"}, {"role": "user", "content": "  \n    Find the top 5 most ex
pensive tracks (based on unit price):\n"}, {"role": "assistant", "content":
"SELECT TrackId, UnitPrice\r\nFROM tracks\r\nORDER BY UnitPrice DESC\r\nLIMI
T 5"}, {"role": "user", "content": "  \n    List all albums and their corres
ponding artist names  \n"}, {"role": "assistant", "content": "SELECT \"album
s\".Title, \"artists\".Name \r\nFROM \"albums\"\r\nJOIN \"artists\" ON \"alb
ums\".\"ArtistId\" = \"artists\".\"ArtistId\"\r\nORDER BY \"albums\".\"Title
\""}, {"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"}, {"role": "assistant", "cont
ent": "SELECT c.FirstName, c.LastName, COUNT(DISTINCT i.InvoiceId) AS TotalA
lbumsBought\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId = i.Custo
merId\r\nLEFT JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\r\nGROUP B
Y c.FirstName, c.LastName\r\nORDER BY TotalAlbumsBought DESC \r\nLIMIT 5"},
{"role": "user", "content": "  \n    Find all invoices since 2010 and the to
tal amount invoiced:\n"}, {"role": "assistant", "content": "SELECT i.Invoice
Id, SUM(i.Total) AS TotalAmount\r\nFROM invoices i\r\nWHERE i.InvoiceDate >=
'2010-01-01'\r\nGROUP BY i.InvoiceId"}, {"role": "user", "content": "Can you
list all tables in the SQLite database catalog?"}, {"role": "assistant", "co
ntent": "SELECT name FROM sqlite_schema WHERE type='table'"}, {"role": "use
r", "content": "  \n    List all invoices with a total exceeding $10:\n"},
{"role": "assistant", "content": "SELECT * \r\nFROM invoices\r\nWHERE Total
> 10"}, {"role": "user", "content": "  \n     Find the top 5 customers who s
pent 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\r\nFROM customers c\r\nJOIN invoices i ON c.Customer
Id = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName\r\nORDER BY TotalSpent
DESC \r\nLIMIT 5"}, {"role": "user", "content": "  \n     Get all playlists
containing at least 10 tracks and the total duration of those tracks:\n"}]
Info: Ollama Response:
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:35:01.096690096Z', 'mess
age': {'role': 'assistant', 'content': 'SELECT \r\n    "playlists".Name,\r\n
SUM("playlist_track"."Milliseconds") AS TrackDurationTotal\r\nFROM "playlist
s"\r\nJOIN "playlist_track" ON "playlists".PlaylistId = "playlist_track"."Pl
aylistId"\r\nGROUP BY "playlists"."Name"\r\nHAVING COUNT("playlist_track"."T
rackId") >= 10\r\nORDER BY TrackDurationTotal DESC; \r\n\r\n'}, 'done_reaso
n': 'stop', 'done': True, 'total_duration': 26430653311, 'load_duration': 23
364030, 'prompt_eval_count': 1530, 'prompt_eval_duration': 18043477000, 'eva
l_count': 89, 'eval_duration': 6981280000}
LLM Response: SELECT
    "playlists".Name,
    SUM("playlist_track"."Milliseconds") AS TrackDurationTotal
FROM "playlists"
JOIN "playlist_track" ON "playlists".PlaylistId = "playlist_track"."Playlist
Id"
GROUP BY "playlists"."Name"
HAVING COUNT("playlist_track"."TrackId") >= 10
ORDER BY TrackDurationTotal DESC;


Info: Output from LLM: SELECT
    "playlists".Name,
    SUM("playlist_track"."Milliseconds") AS TrackDurationTotal
FROM "playlists"
JOIN "playlist_track" ON "playlists".PlaylistId = "playlist_track"."Playlist
Id"
GROUP BY "playlists"."Name"
HAVING COUNT("playlist_track"."TrackId") >= 10
ORDER BY TrackDurationTotal DESC;


Extracted SQL: SELECT
    "playlists".Name,
    SUM("playlist_track"."Milliseconds") AS TrackDurationTotal
FROM "playlists"
JOIN "playlist_track" ON "playlists".PlaylistId = "playlist_track"."Playlist
Id"
GROUP BY "playlists"."Name"
HAVING COUNT("playlist_track"."TrackId") >= 10
ORDER BY TrackDurationTotal DESC
SELECT
    "playlists".Name,
    SUM("playlist_track"."Milliseconds") AS TrackDurationTotal
FROM "playlists"
JOIN "playlist_track" ON "playlists".PlaylistId = "playlist_track"."Playlist
Id"
GROUP BY "playlists"."Name"
HAVING COUNT("playlist_track"."TrackId") >= 10
ORDER BY TrackDurationTotal DESC
Couldn't run sql:  Execution failed on sql 'SELECT
    "playlists".Name,
    SUM("playlist_track"."Milliseconds") AS TrackDurationTotal
FROM "playlists"
JOIN "playlist_track" ON "playlists".PlaylistId = "playlist_track"."Playlist
Id"
GROUP BY "playlists"."Name"
HAVING COUNT("playlist_track"."TrackId") >= 10
ORDER BY TrackDurationTotal DESC': no such column: playlist_track.Millisecon
ds
```

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 \r\n    "artists"."Name", \r\n    COUNT("tracks"."TrackId") AS
NumTracks\r\nFROM "artists"\r\nJOIN "albums" ON "artists"."ArtistId" = "albu
ms"."ArtistId"\r\nJOIN "tracks" ON "albums"."AlbumId" = "tracks"."AlbumId"\r
\nGROUP BY "artists"."Name"\r\nORDER BY NumTracks DESC \r\nLIMIT 10'}, {'rol
e': 'user', 'content': '  \n    List all genres and the number of tracks in
each genre:\n'}, {'role': 'assistant', 'content': 'SELECT "genres"."Name", C
OUNT(DISTINCT "tracks"."TrackId") AS NumTracks\r\nFROM "genres"\r\nLEFT JOIN
"tracks" ON "genres"."GenreId" = "tracks"."GenreId"\r\nGROUP BY "genres"."Na
me"\r\nORDER BY NumTracks DESC'}, {'role': 'user', 'content': '  \n    List
all albums and their corresponding artist names  \n'}, {'role': 'assistant',
'content': 'SELECT "albums".Title, "artists".Name \r\nFROM "albums"\r\nJOIN
"artists" ON "albums"."ArtistId" = "artists"."ArtistId"\r\nORDER BY "album
s"."Title"'}, {'role': 'user', 'content': '  \n    Hint: album quantity is f
ound in invoice_items, \n    \n    Find the top 5 customers who bought the m
```

```
ost albums in total quantity (across all invoices):\n'}, {'role': 'assistan
t', 'content': 'SELECT c.FirstName, c.LastName, COUNT(DISTINCT i.InvoiceId)
AS TotalAlbumsBought\r\nFROM customers c\r\nJOIN invoices i ON c.CustomerId
= i.CustomerId\r\nLEFT JOIN invoice_items ii ON i.InvoiceId = ii.InvoiceId\r
\nGROUP BY c.FirstName, c.LastName\r\nORDER BY TotalAlbumsBought DESC \r\nLI
MIT 5'}, {'role': 'user', 'content': '  \n    Find the top 5 most expensive
tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT
TrackId, UnitPrice\r\nFROM tracks\r\nORDER BY UnitPrice DESC\r\nLIMIT 5'},
{'role': 'user', 'content': '  \n    Find all tracks with a name containing
"What" (case-insensitive)\n'}, {'role': 'assistant', 'content': "SELECT * \r
\nFROM tracks\r\nWHERE Name LIKE '%what%'\r\nORDER BY Name"}, {'role': 'use
r', 'content': 'Can you list all tables in the SQLite database catalog?'},
{'role': 'assistant', 'content': "SELECT name FROM sqlite_schema WHERE type
='table'"}, {'role': 'user', 'content': '  \n     Find the customer with the
most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.FirstName, c.
LastName, COUNT(i.InvoiceId) AS TotalInvoices\r\nFROM customers c\r\nJOIN in
voices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.FirstName, c.LastName
\r\nORDER BY TotalInvoices DESC\r\nLIMIT 1'}, {'role': 'user', 'content': 'w
hat are the top 5 countries that customers come from?'}, {'role': 'assistan
t', 'content': 'SELECT Country, COUNT(DISTINCT CustomerId) AS NumCustomers\r
\nFROM customers\r\nGROUP BY Country\r\nORDER BY NumCustomers DESC\r\nLIMIT
5'}, {'role': 'user', 'content': 'How many customers are there'}, {'role':
'assistant', 'content': 'SELECT COUNT(*) FROM customers'}, {'role': 'user',
'content': '  \n     Identify artists who have albums with tracks appearing
in multiple genres:\n\n\n'}]
Info: Ollama parameters:
model=gemma2:2b,
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 \r\n    \"artists\".\"Name\", \r\n    COUNT(\"tr
acks\".\"TrackId\") AS NumTracks\r\nFROM \"artists\"\r\nJOIN \"albums\" ON
\"artists\".\"ArtistId\" = \"albums\".\"ArtistId\"\r\nJOIN \"tracks\" ON \"a
lbums\".\"AlbumId\" = \"tracks\".\"AlbumId\"\r\nGROUP BY \"artists\".\"Name
\"\r\nORDER BY NumTracks DESC \r\nLIMIT 10"}, {"role": "user", "content": "
\n    List all genres and the number of tracks in each genre:\n"}, {"role":
"assistant", "content": "SELECT \"genres\".\"Name\", COUNT(DISTINCT \"tracks
\".\"TrackId\") AS NumTracks\r\nFROM \"genres\"\r\nLEFT JOIN \"tracks\" ON
\"genres\".\"GenreId\" = \"tracks\".\"GenreId\"\r\nGROUP BY \"genres\".\"Nam
e\"\r\nORDER BY NumTracks DESC"}, {"role": "user", "content": "  \n    List
all albums and their corresponding artist names  \n"}, {"role": "assistant",
"content": "SELECT \"albums\".Title, \"artists\".Name \r\nFROM \"albums\"\r
\nJOIN \"artists\" ON \"albums\".\"ArtistId\" = \"artists\".\"ArtistId\"\r\n
ORDER BY \"albums\".\"Title\""}, {"role": "user", "content": "  \n    Hint:
album quantity is found in invoice_items, \n    \n    Find the top 5 custome
rs who bought the most albums in total quantity (across all invoices):\n"},
{"role": "assistant", "content": "SELECT c.FirstName, c.LastName, COUNT(DIST
INCT i.InvoiceId) AS TotalAlbumsBought\r\nFROM customers c\r\nJOIN invoices
i ON c.CustomerId = i.CustomerId\r\nLEFT JOIN invoice_items ii ON i.InvoiceI
d = ii.InvoiceId\r\nGROUP BY c.FirstName, c.LastName\r\nORDER BY TotalAlbums
Bought DESC \r\nLIMIT 5"}, {"role": "user", "content": "  \n    Find the top
5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "c
ontent": "SELECT TrackId, UnitPrice\r\nFROM tracks\r\nORDER BY UnitPrice DES
C\r\nLIMIT 5"}, {"role": "user", "content": "  \n    Find all tracks with a
name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "cont
ent": "SELECT * \r\nFROM tracks\r\nWHERE Name LIKE '%what%'\r\nORDER BY Nam
e"}, {"role": "user", "content": "Can you list all tables in the SQLite data
base catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite_s
chema WHERE type='table'"}, {"role": "user", "content": "  \n     Find the c
ustomer with the most invoices \n"}, {"role": "assistant", "content": "SELEC
T c.FirstName, c.LastName, COUNT(i.InvoiceId) AS TotalInvoices\r\nFROM custo
mers c\r\nJOIN invoices i ON c.CustomerId = i.CustomerId\r\nGROUP BY c.First
Name, c.LastName\r\nORDER BY TotalInvoices DESC\r\nLIMIT 1"}, {"role": "use
r", "content": "what are the top 5 countries that customers come from?"},
{"role": "assistant", "content": "SELECT Country, COUNT(DISTINCT CustomerId)
AS NumCustomers\r\nFROM customers\r\nGROUP BY Country\r\nORDER BY NumCustome
rs DESC\r\nLIMIT 5"}, {"role": "user", "content": "How many customers are th
ere"}, {"role": "assistant", "content": "SELECT COUNT(*) FROM customers"},
{"role": "user", "content": "  \n     Identify artists who have albums with
tracks appearing in multiple genres:\n\n\n"}]
```

```
Info: Ollama Response:
```

```
{'model': 'gemma2:2b', 'created_at': '2024-08-01T18:35:41.263639692Z', 'mess
age': {'role': 'assistant', 'content': '```sql\nSELECT \r\n    "artists".Nam
e,\r\n    COUNT(DISTINCT "tracks"."GenreId") AS NumGenres\r\nFROM "artist
s"\r\nJOIN "albums"\r\nON "artists"."ArtistId" = "albums"."ArtistId"\r\nJOIN
"tracks"\r\nON "albums"."AlbumId" = "tracks"."AlbumId"\r\nGROUP BY "artist
s"."Name"\r\nHAVING COUNT(DISTINCT "tracks"."GenreId") > 1\r\nORDER BY NumGe
nres DESC; \r\n```\n\n**Explanation:**\n\n1. **SELECT**: We\'re selecting th
e artist name (`artists.Name`) and the count of unique genres associated wit
h their albums (`NumGenres`).\n2. **FROM**: We start by joining the `artists
`, `albums` tables using `ArtistId` and then join  `tracks` table based on `
AlbumId`. \n3. **JOIN**: This is used to combine data from all three tables.
\n4. **GROUP BY**: Groups the results based on artist names. \n5. **HAVING*
*: Filters out artists with only one genre, as we\'re looking for those with
albums in multiple genres.  \n6. **ORDER BY**: Sorts the output in descendin
g order of `NumGenres`.\n\n\n\nLet me know if you have any other queries!'},
'done_reason': 'stop', 'done': True, 'total_duration': 40125187214, 'load_du
ration': 23766734, 'prompt_eval_count': 1454, 'prompt_eval_duration': 165645
08000, 'eval_count': 276, 'eval_duration': 22163084000}
```

~~~
LLM Response: ```sql
SELECT
    "artists".Name,
    COUNT(DISTINCT "tracks"."GenreId") AS NumGenres
FROM "artists"
JOIN "albums"
ON "artists"."ArtistId" = "albums"."ArtistId"
JOIN "tracks"
ON "albums"."AlbumId" = "tracks"."AlbumId"
GROUP BY "artists"."Name"
HAVING COUNT(DISTINCT "tracks"."GenreId") > 1
ORDER BY NumGenres DESC;
```

**Explanation:**

1. **SELECT**: We're selecting the artist name (`artists.Name`) and the coun
t of unique genres associated with their albums (`NumGenres`).
2. **FROM**: We start by joining the `artists`, `albums` tables using `Artis
tId` and then join  `tracks` table based on `AlbumId`.
3. **JOIN**: This is used to combine data from all three tables.
4. **GROUP BY**: Groups the results based on artist names.
5. **HAVING**: Filters out artists with only one genre, as we're looking for
those with albums in multiple genres.
6. **ORDER BY**: Sorts the output in descending order of `NumGenres`.



Let me know if you have any other queries!
Info: Output from LLM: ```sql
SELECT
    "artists".Name,
    COUNT(DISTINCT "tracks"."GenreId") AS NumGenres
FROM "artists"
JOIN "albums"
ON "artists"."ArtistId" = "albums"."ArtistId"
JOIN "tracks"
~~~

```
ON "albums"."AlbumId" = "tracks"."AlbumId"
GROUP BY "artists"."Name"
HAVING COUNT(DISTINCT "tracks"."GenreId") > 1
ORDER BY NumGenres DESC;
```

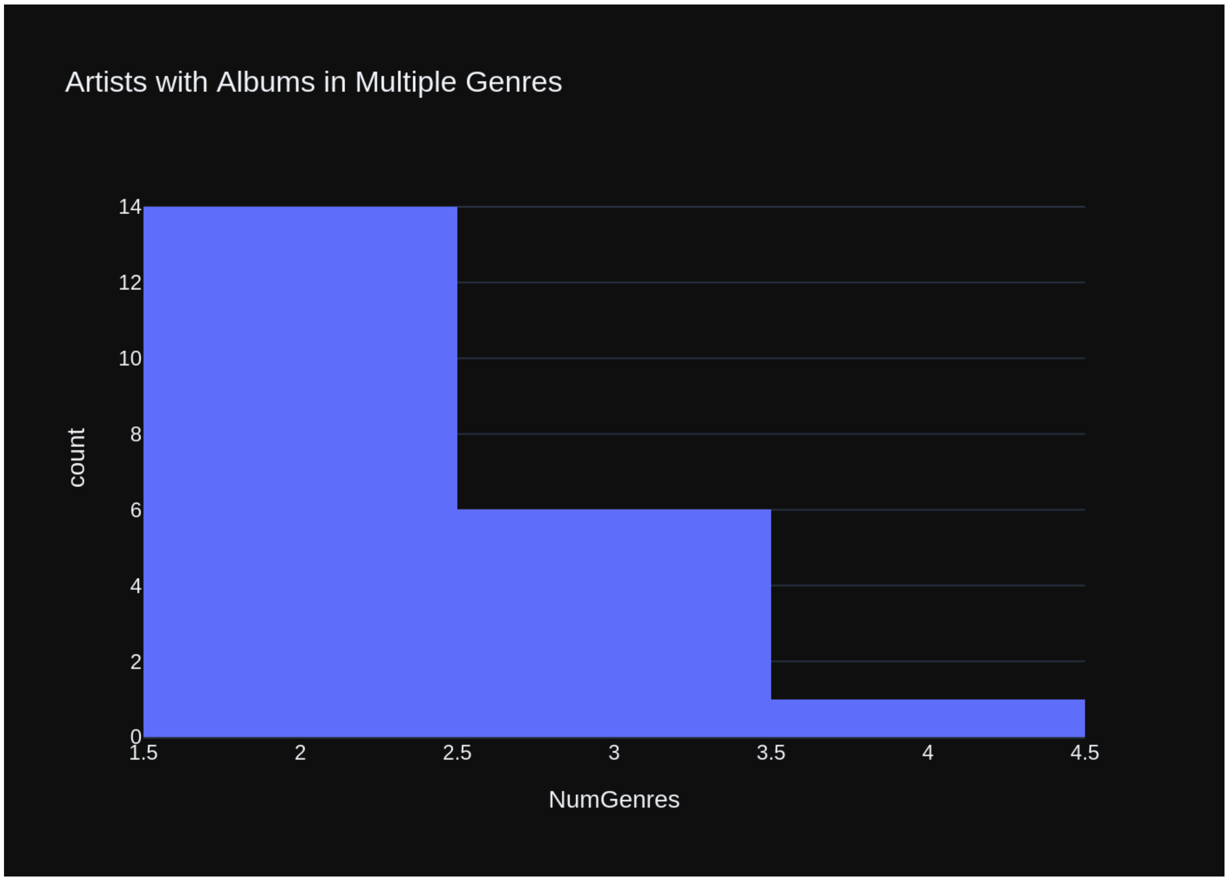
**Explanation:**

1. **SELECT**: We're selecting the artist name (`artists.Name`) and the count of unique genres associated with their albums (`NumGenres`).
2. **FROM**: We start by joining the `artists`, `albums` tables using `ArtistId` and then join  `tracks` table based on `AlbumId`.
3. **JOIN**: This is used to combine data from all three tables.
4. **GROUP BY**: Groups the results based on artist names.
5. **HAVING**: Filters out artists with only one genre, as we're looking for those with albums in multiple genres.
6. **ORDER BY**: Sorts the output in descending order of `NumGenres`.

Let me know if you have any other queries!

```
Extracted SQL: SELECT
    "artists".Name,
    COUNT(DISTINCT "tracks"."GenreId") AS NumGenres
FROM "artists"
JOIN "albums"
ON "artists"."ArtistId" = "albums"."ArtistId"
JOIN "tracks"
ON "albums"."AlbumId" = "tracks"."AlbumId"
GROUP BY "artists"."Name"
HAVING COUNT(DISTINCT "tracks"."GenreId") > 1
ORDER BY NumGenres DESC
SELECT
    "artists".Name,
    COUNT(DISTINCT "tracks"."GenreId") AS NumGenres
FROM "artists"
JOIN "albums"
ON "artists"."ArtistId" = "albums"."ArtistId"
JOIN "tracks"
ON "albums"."AlbumId" = "tracks"."AlbumId"
GROUP BY "artists"."Name"
HAVING COUNT(DISTINCT "tracks"."GenreId") > 1
ORDER BY NumGenres DESC
                     Name  NumGenres
0             Iron Maiden          4
1         Various Artists          3
2           Lenny Kravitz          3
3              Jamiroquai          3
4             Gilberto Gil          3
5    Battlestar Galactica          3
6              Audioslave          3
7                      U2          2
8              The Office          2
9   Red Hot Chili Peppers          2
10                 R.E.M.          2
11              Pearl Jam          2
```

```
12          Ozzy Osbourne          2
13                   Lost          2
14                 Heroes          2
15          Guns N' Roses          2
16           Foo Fighters          2
17          Faith No More          2
18           Eric Clapton          2
19   Antônio Carlos Jobim          2
20          Amy Winehouse          2
Info: Ollama parameters:
model=gemma2:2b,
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 \r\n
\"artists\".Name,\r\n    COUNT(DISTINCT \"tracks\".\"GenreId\") AS NumGenres
\r\nFROM \"artists\"\r\nJOIN \"albums\"\r\nON \"artists\".\"ArtistId\" = \"a
lbums\".\"ArtistId\"\r\nJOIN \"tracks\"\r\nON \"albums\".\"AlbumId\" = \"tra
cks\".\"AlbumId\"\r\nGROUP BY \"artists\".\"Name\"\r\nHAVING COUNT(DISTINCT
\"tracks\".\"GenreId\") > 1\r\nORDER BY NumGenres DESC\n\nThe following is i
nformation about the resulting pandas DataFrame 'df': \nRunning df.dtypes gi
ves:\n Name         object\nNumGenres     int64\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': 'gemma2:2b', 'created_at': '2024-08-01T18:35:47.409598702Z', 'mess
age': {'role': 'assistant', 'content': '```python\nimport plotly.express as
px\n\nfig = px.histogram(df, x="NumGenres", title=\'Artists with Albums in M
ultiple Genres\') \nfig.show() \n```'}, 'done_reason': 'stop', 'done': True,
'total_duration': 6116501559, 'load_duration': 27318443, 'prompt_eval_coun
t': 257, 'prompt_eval_duration': 3038254000, 'eval_count': 42, 'eval_duratio
n': 2958742000}
```


Artists with Albums in Multiple Genres
count
0
2
4
6
8
10
12
14
1.5
2
2.5
3
3.5
4
4.5
NumGenres

```
Out[39]: ('SELECT \r\n    "artists".Name,\r\n    COUNT(DISTINCT "tracks"."GenreId")
         AS NumGenres\r\nFROM "artists"\r\nJOIN "albums"\r\nON "artists"."ArtistId"
         = "albums"."ArtistId"\r\nJOIN "tracks"\r\nON "albums"."AlbumId" = "track
         s"."AlbumId"\r\nGROUP BY "artists"."Name"\r\nHAVING COUNT(DISTINCT "track
         s"."GenreId") > 1\r\nORDER BY NumGenres DESC',
                                Name  NumGenres
          0              Iron Maiden          4
          1          Various Artists          3
          2            Lenny Kravitz          3
          3               Jamiroquai          3
          4             Gilberto Gil          3
          5     Battlestar Galactica          3
          6               Audioslave          3
          7                       U2          2
          8               The Office          2
          9    Red Hot Chili Peppers          2
          10                  R.E.M.          2
          11               Pearl Jam          2
          12           Ozzy Osbourne          2
          13                    Lost          2
          14                  Heroes          2
          15           Guns N' Roses          2
          16            Foo Fighters          2
          17           Faith No More          2
          18            Eric Clapton          2
          19    Antônio Carlos Jobim          2
          20           Amy Winehouse          2,
          Figure({
              'data': [{'alignmentgroup': 'True',
                        'bingroup': 'x',
                        'hovertemplate': 'NumGenres=%{x}<br>count=%{y}<extra></extra
         >',
                        'legendgroup': '',
                        'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                        'name': '',
                        'offsetgroup': '',
                        'orientation': 'v',
                        'showlegend': False,
                        'type': 'histogram',
                        'x': array([4, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2,
         2, 2, 2, 2, 2]),
                        'xaxis': 'x',
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                         'legend': {'tracegroupgap': 0},
                         'template': '...',
                         'title': {'text': 'Artists with Albums in Multiple Genre
         s'},
                         'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'t
         ext': 'NumGenres'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'t
         ext': 'count'}}}
          }))
```

## Check completion time

```
In [9]: from datetime import datetime
        import os
        hostname = os.uname().nodename
        print("Hostname:", hostname)
        model_name = "gemma2:2b"
```

```
Hostname: ducklover1
```

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

        elapsed_time = ts_stop - ts_start
```

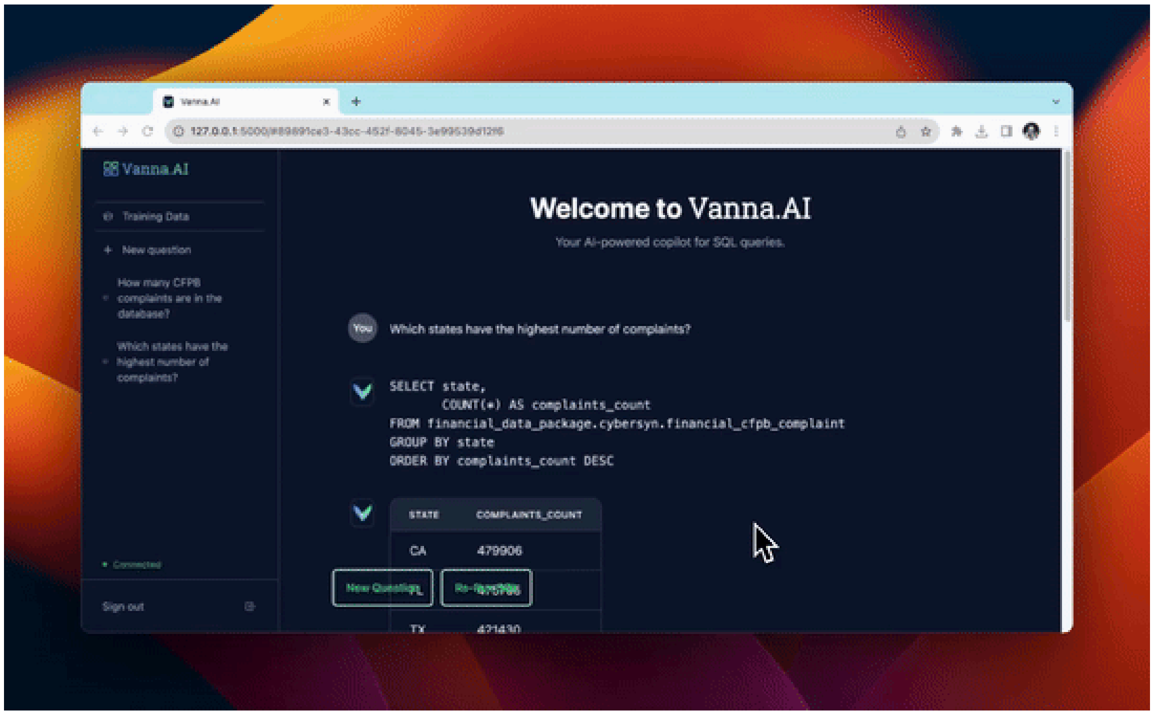
```
In [10]: print(f"[{datetime.now()}] test on '{hostname}' with '{model_name}' LLM took
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[10], line 1
----> 1 print(f"[{datetime.now()}] test on '{hostname}' with '{model_name}'
LLM took : {elapsed_time:.2f} sec")

NameError: name 'elapsed_time' is not defined
```

```
In [41]: print(datetime.now())
```

```
2024-08-01 14:35:47.476246
```

## Launch the User Interface


Vanna.AI
127.0.0.1:5000/#8989f1ce3-43cc-452f-8045-3e99539d12f6
Vanna.AI
Training Data
New question
How many CFPB complaints are in the database?
Which states have the highest number of complaints?
Connected
Sign out
New Question
Re-Run SQL

Welcome to Vanna.AI

Your AI-powered copilot for SQL queries.

You: Which states have the highest number of complaints?

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

| STATE | COMPLAINTS_COUNT |
|---|---|
| CA | 479906 |
| FL | [illegible] |
| TX | 421430 |

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

## Next Steps

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

- Streamlit app
- Flask app
- Slackbot