# 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?

- OpenAl via Vanna.Al (Recommended)
   Use Vanna.Al for free to generate your queries
- OpenAl

Use OpenAl with your own API key

Azure OpenAl

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. Als hosted vector database (pgvector) for free. This is usable across machines with no additional setup.

• [Selected] ChromaDB

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

Marqo

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

Other VectorDB

Use any other vector database. Requires additional setup.

### Setup

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

```
In [1]: import warnings
        import re
        warnings.filterwarnings('ignore', category=DeprecationWarning, message='^Number of requested results')
        # warnings.filterwarnings('ignore', category=DeprecationWarning, message=re.escape(r'^Some regex pattern')
        import os
        import re
        from time import time
        from vanna.ollama import Ollama
        from vanna.chromadb.chromadb vector import ChromaDB VectorStore
In [2]: class MyVanna(ChromaDB_VectorStore, Ollama):
            def init (self, config=None):
                ChromaDB VectorStore. init (self, config=config)
                Ollama. init (self, config=config)
In [3]: file db = "~/Downloads/chinook.sqlite"
        model name = 'phi3'
In [4]: config = {
            'model': model name, # 'mistral' # "starcoder2"
        vn = MyVanna(config=config)
In [5]: hostname = os.uname().nodename
        print("Hostname:", hostname)
       Hostname: ducklover1
```

```
In [6]: file db = os.path.abspath(os.path.expanduser(file db))
        vn.connect to sqlite(file db)
In [7]: vn.run sql is set
Out[7]: True
In [8]: | def remove collections(collection name=None, ACCEPTED TYPES = ["sql", "ddl", "documentation"]):
             if not collection name:
                 collections = ACCEPTED TYPES
            elif isinstance(collection name, str):
                 collections = [collection name]
            elif isinstance(collection name, list):
                 collections = collection name
             else:
                 print(f"\t{collection name} is unknown: Skipped")
                 return
             for c in collections:
                 if not c in ACCEPTED TYPES:
                     print(f"\t{c} is unknown: Skipped")
                     continue
                 # print(f"vn.remove collection('{c}')")
                 vn.remove collection(c)
In [9]: def strip brackets(ddl):
            This function removes square brackets from table and column names in a DDL script.
            Args:
                 ddl (str): The DDL script containing square brackets.
             Returns:
                 str: The DDL script with square brackets removed.
             0.00
            # Use regular expressions to match and replace square brackets
            pattern = r"\setminus [([^{]}]+)]" # Match any character except ] within square brackets
            return re.sub(pattern, r"\1", ddl)
```

```
In [10]: if False:
    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]: # show training data
    training_data = vn.get_training_data()
    training_data
```

Out[11]:

:	id	question	content	training_data_type
0	01c4a964-460b-5e1c-af1e- 622c8210b835-sql	\n Hint: album quantity is found in invoi	SELECT i.Customerld, COUNT(ii.InvoiceLineld) A	sql
1	0658ba3d-98ff-51f4-9006- a24f87045858-sql	How many customers are there	SELECT COUNT(*) FROM "customers"	sql
2	0e1a2b7b-d65e-53de-b839- edb7afcf4ab1-sql	\n Hint: album quantity is found in invoi	SELECT i.CustomerId, COUNT(ii.TrackId) AS Tota	sql
3	127fd4bd-b9af-539d-9313- 1d0234d073b7-sql	\n There are 3 tables: artists, albums and	SELECT a.Name, COUNT(t.TrackId) AS TotalTracks	sql
4	17d893d5-1417-5ba3-a5ca- 9f6ce15a727f-sql	\n Identify artists who have albums with	SELECT a.ArtistId, a.Name AS ArtistName\nFROM	sql
5	3013d1b4-feb2-519d-bfb9- 114500436e3d-sql	\n Find the customer with the most invoi	SELECT c.CustomerId, COUNT(i.InvoiceId) AS Tot	sql
6	32b99e7b-31ab-55d8-8431- fb010fa7af85-sql	\n Find the top 5 customers who spent th	SELECT c.CustomerId, SUM(i.Total) AS TotalSpen	sql
7	49e67df3-a604-51f8-ad01- b8f5a2043eac-sql	\n Get the total number of invoices for e	SELECT c.CustomerId, COUNT(i.InvoiceId) AS Tot	sql
8	584873f8-1904-50f1-8f80- 7ccf08059264-sql	\n List all customers from Canada and th	SELECT c.Email, c.Country\nFROM "customers" c\	sql
9	6bed484b-9a80-57f4-ad89- 5f775b5df252-sql	\n Get the average invoice total for each	SELECT c.CustomerId, AVG(i.Total) AS AverageIn	sql
10	6f22268c-5062-5f11-ba2d- 8555f06b409d-sql	\n Find all tracks with a name containing	SELECT * \nFROM "tracks" \nWHERE LOWER(Name) L	sql
11	70b4f686-c71b-5ee8-9458- 6bbc776349bf-sql	\n Find all invoices since 2010 and the t	SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmo	sql
12	9a396a33-ecea-51a8-bd05- 28f58a86eb86-sql	\n Hint: album quantity is found in invoi	SELECT c.CustomerId, COUNT(ii.TrackId) AS Tota	sql
13	9a9c970b-b94c-5f22-b54c- b86921a38b65-sql	\n Identify artists who have albums with	SELECT a.ArtistId, a.Name AS ArtistName\nFROM	sql
14	a7185c88-7417-5b75-a52e- 4eaef5f9deca-sql	\n List all albums and their correspondin	SELECT a.Title, a.ArtistId, ar.Name AS ArtistN	sql
15	aea89953-21b2-55d1-9dda- 431ee6033c3d-sql	\n List all invoices with a total exceedi	SELECT * \nFROM "invoices" \nWHERE Total > 10.00	sql
16	d1d70c18-f5d9-5970-a32c-	\n Find the customer who	SELECT c.Customerld, COUNT(ii.TrackId)	sql

	id	question	content	training_data_type
	914deeca1087-sql	bought the most	AS Tota	
17	d8a2f948-dffa-5524-a5f9- 174cc1a8da73-sql	Can you list all tables in the SQLite database	SELECT name FROM sqlite_master WHERE type='table'	sql
18	d8a37163-5ce5-58cd-a316- ea5598d44d27-sql	what are the top 5 countries that customers co	SELECT c.Country, COUNT(*) AS TotalCustomers\n	sql
19	dd282d7c-a4ef-5e3a-87e0- cb45fac50808-sql	\n Find the total number of invoices per	SELECT i.BillingCountry, COUNT(*) AS TotalInvo	sql
20	e7c4b3aa-664f-5f87-8b25- 449a4482f3fd-sql	\n Get all playlists containing at least	SELECT pt.PlaylistId, p.Name AS PlaylistName,	sql
21	f33f8cb6-1b12-5ea7-8d9a- aef8166b9970-sql	\n Find the top 5 most expensive tracks (	SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "t	sql
22	f626b681-4d8f-563a-beee- 1ea759baaa82-sql	\n List all genres and the number of trac	SELECT g.Name, COUNT(t.Genreld) AS TotalTracks	sql
23	fd25ebba-4066-5a0f-8613- 7b1c2ace0339-sql	\n List all employees and their reporting	SELECT e.FirstName, e.LastName, mt.FirstName A	sql
0	039f9d54-59f7-5f29-8c04- 14dbc3e95671-ddl	None	CREATE TABLE "artists"\r\n(\r\n Artistld 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

	id	question	content	training_data_type
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-ddl	None	CREATE INDEX IFK_TrackGenreId ON "tracks" (Gen	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 Customerl	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	9d2550eb-8e22-54cd-9fad- 9e1be65ab03a-doc	None	In the SQLite database invoice means order	documentation

In [12]: df\_ddl = vn.run\_sql("SELECT type, sql FROM sqlite\_master WHERE sql is not null")

In [13]: df\_ddl

$\cap$	14-	Γ	7	$\supset$	1	
U	uц	L	Τ	J	J	-

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

In [15]:  # Sometimes you may want to add documentation about your business terminology or definitions.
        vn.train(documentation="In the SQLite database invoice means order")
```

# Asking the Al

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 [16]: ts_start = time()

SELECT name FROM sqlite_master WHERE type = 'table';
In [17]: vn.ask(question="Can you list all tables in the SQLite database catalog?")
```

```
Add of existing embedding ID: d8a2f948-dffa-5524-a5f9-174ccla8da73-sql
Add of existing embedding ID: 0658ba3d-98ff-51f4-9006-a24f87045858-sql
Add of existing embedding ID: 127fd4bd-b9af-539d-9313-1d0234d073b7-sql
Add of existing embedding ID: 32b99e7b-31ab-55d8-8431-fb010fa7af85-sql
Add of existing embedding ID: d8a2f948-dffa-5524-a5f9-174ccla8da73-sql
Add of existing embedding ID: 0658ba3d-98ff-51f4-9006-a24f87045858-sql
Add of existing embedding ID: d8a37163-5ce5-58cd-a316-ea5598d44d27-sql
Add of existing embedding ID: a7185c88-7417-5b75-a52e-4eaef5f9deca-sql
Add of existing embedding ID: 6f22268c-5062-5f11-ba2d-8555f06b409d-sql
Add of existing embedding ID: 49e67df3-a604-51f8-ad01-b8f5a2043eac-sql
Add of existing embedding ID: dd282d7c-a4ef-5e3a-87e0-cb45fac50808-sql
Add of existing embedding ID: aea89953-21b2-55d1-9dda-431ee6033c3d-sql
Add of existing embedding ID: fd25ebba-4066-5a0f-8613-7b1c2ace0339-sql
Add of existing embedding ID: 6bed484b-9a80-57f4-ad89-5f775b5df252-sql
Add of existing embedding ID: f33f8cb6-1b12-5ea7-8d9a-aef8166b9970-sql
Add of existing embedding ID: f626b681-4d8f-563a-beee-lea759baaa82-sql
Add of existing embedding ID: 127fd4bd-b9af-539d-9313-1d0234d073b7-sql
Add of existing embedding ID: 584873f8-1904-50f1-8f80-7ccf08059264-sql
Add of existing embedding ID: 3013d1b4-feb2-519d-bfb9-114500436e3d-sql
Add of existing embedding ID: d1d70c18-f5d9-5970-a32c-914deeca1087-sql
Add of existing embedding ID: e7c4b3aa-664f-5f87-8b25-449a4482f3fd-sql
Add of existing embedding ID: 9a9c970b-b94c-5f22-b54c-b86921a38b65-sql
Add of existing embedding ID: d8a2f948-dffa-5524-a5f9-174ccla8da73-sql
Add of existing embedding ID: 0658ba3d-98ff-51f4-9006-a24f87045858-sql
Add of existing embedding ID: d8a37163-5ce5-58cd-a316-ea5598d44d27-sql
Add of existing embedding ID: a7185c88-7417-5b75-a52e-4eaef5f9deca-sql
Add of existing embedding ID: 6f22268c-5062-5f11-ba2d-8555f06b409d-sql
Add of existing embedding ID: 49e67df3-a604-51f8-ad01-b8f5a2043eac-sql
Add of existing embedding ID: dd282d7c-a4ef-5e3a-87e0-cb45fac50808-sql
Add of existing embedding ID: aea89953-21b2-55d1-9dda-431ee6033c3d-sql
Add of existing embedding ID: fd25ebba-4066-5a0f-8613-7b1c2ace0339-sql
Add of existing embedding ID: 6bed484b-9a80-57f4-ad89-5f775b5df252-sql
Add of existing embedding ID: f33f8cb6-1b12-5ea7-8d9a-aef8166b9970-sql
Add of existing embedding ID: f626b681-4d8f-563a-beee-lea759baaa82-sql
Add of existing embedding ID: 127fd4bd-b9af-539d-9313-1d0234d073b7-sql
Add of existing embedding ID: 584873f8-1904-50f1-8f80-7ccf08059264-sql
Add of existing embedding ID: 3013d1b4-feb2-519d-bfb9-114500436e3d-sql
Add of existing embedding ID: e7c4b3aa-664f-5f87-8b25-449a4482f3fd-sql
Number of requested results 10 is greater than number of elements in index 1, updating n results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE sqlite stat1(tbl,idx,stat)\n\nCREATE TABLE sqlite sequence(na PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n me,seq)\n\nCREATE TABLE "playlists"\r\n(\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "genres"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOI NCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER Milliseconds INTEGER NOT NULL.\r\n NOT NULL,\r\n GenreId INTEGER.\r\n Composer NVARCHAR(220),\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" Bytes INTEGER.\r\n (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "med ia types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "media type MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n  $s"\r\n(\r\n$ Name NVARCHAR(120)\r\n)\n\nCR ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n EATE TABLE "artists"\r\n(\r\n (120)\r\n)\n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10.2) FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceI NOT NULL,\r\n Ouantity INTEGER NOT NULL,\r\n d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (Track Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist track"\r\n(\r\n Plavl istId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELET (PlavlistId, TrackId),\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\t0N DELE E NO ACTION ON UPDATE NO ACTION,\r\n TE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTO ArtistId INTEGER NOT NULL.\r\n INCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n== =Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the pr ovided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particul ar column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend t he guery with a comment saying intermediate sql \n3. If the provided context is insufficient, please explai n why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been ask ed and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'co ntent': 'Can you list all tables in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELEC T name FROM sqlite master WHERE type='table'"}, {'role': 'user', 'content': ' \n There are 3 tables: art ists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by A lbumId.\n Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assi stant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': '\n List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a \nJOIN "artists" ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n from Canada and their email addresses:\n'}, {'role': 'assistant', 'content': 'SELECT c.Email, c.Country\nFR

OM "customers" c\nWHERE c.Country = \'Canada\''}, {'role': 'user', 'content': ' \n Find the customer w ho bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.Customer Id = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY T otalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n List all genres and the number of tracks i n each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.Name'}, {'role': 'user', 'content': " \n List all employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': 'SELE CT e.FirstName, e.LastName, mt.FirstName AS ManagerFirstName, mt.LastName AS ManagerLastName\nFROM "employe es" e\nLEFT JOIN "employees" mt ON e.ReportsTo = mt.EmployeeId'}, {'role': 'user', 'content': ' \n t: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most album s in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUN T(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "i nvoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPr ice DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n Get all playlists containing at least 10 tracks and the total duration of those tracks:\n'}, {'role': 'assistant', 'content': 'SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM "playlist track" pt\nJOIN "playlists" p ON pt.P laylistId = p.PlaylistId\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10'}, {'role': 'user', 'content': 'Can you list all tables in the SQLite database cata log?'}] Ollama parameters: model=phi3:latest,

options={},

keep\_alive=None

Prompt Content:

[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE sqlite stat1(tbl,idx,stat)\n\nCREATE TABLE sqlite sequence(na PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r me,seg)\n\nCREATE TABLE \"playlists\"\r\n(\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREM ENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTvpeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL,\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENC ES \"albums\" (Albumid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFER ENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeI d) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"media types\"\r\n(\r\n MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARC  $HAR(120)\r\n)\n\nCREATE TABLE \"artists\"\r\n(\r\n$ ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL Name NVARCHAR(120) $\r\n)\n\n$ CREATE TABLE \"invoice items\\"\r\n(\r\n InvoiceLineId INTEGER PRIMA  $L,\r\n$ 

RY KEY AUTOINCREMENT NOT NULL.\r\n InvoiceId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Ouantity INTEGER NOT NULL.\r\n FOREIGN KEY (InvoiceId) REFERE NCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"play list track\"\r\n(\r\n PlavlistId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL,\r\n PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists \" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Title NVARCHAR(160) NOT NULL.\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE N stId INTEGER NOT NULL.\r\n O ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order \n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SOL guery wi thout any explanations for the question. \n2. If the provided context is almost sufficient but requires kno wledge of a specific string in a particular column, please generate an intermediate SQL query to find the d istinct strings in that column. Prepend the query with a comment saying intermediate sql \n3. If the provid ed context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant tab le(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "Can you list all tables in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite master WHERE type='table'"}, {"role": "user", "co ntent": " \n There are 3 tables: artists, albums and tracks, where albums and artists are linked by Arti stId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based on t he number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\n FROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.Alb umId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.Arti stId, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n List all customers from Canada and their email addresses:\n"}, {"role": "assis tant", "content": "SELECT c.Email, c.Country\nFROM \"customers\" c\nWHERE c.Country = 'Canada'"}, {"role": "user", "content": " \n Find the customer who bought the most albums in total quantity (across all inv oices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.Invoice Id = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SEL ECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId \nGROUP BY q.Name"}, {"role": "user", "content": " \n List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "content": "SELECT e.FirstName, e.LastName, mt.FirstName AS Manag erFirstName, mt.LastName AS ManagerLastName\nFROM \"employees\" e\nLEFT JOIN \"employees\" mt ON e.ReportsT o = mt.EmployeeId"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, Find the top 5 customers who bought the most albums in total quantity (across all invoice s):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"cu stomers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "

\n Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SE LECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "use r", "content": " \n Get all playlists containing at least 10 tracks and the total duration of those tr acks:\n"}, {"role": "assistant", "content": "SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Millisecon ds) AS TotalDuration\nFROM \"playlist\_track\" pt\nJOIN \"playlists\" p ON pt.PlaylistId = p.PlaylistId\nJOI N \"tracks\" t ON pt.TrackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10"}, {"role": "user", "content": "Can you list all tables in the SQLite database catalog?"}]

Add of existing embedding ID: d8a2f948-dffa-5524-a5f9-174ccla8da73-sql Insert of existing embedding ID: d8a2f948-dffa-5524-a5f9-174ccla8da73-sql

#### Ollama Response:

{'model': 'phi3:latest', 'created\_at': '2024-06-14T02:37:12.019981085Z', 'message': {'role': 'assistant', 'content': "SELECT name FROM sqlite\_master WHERE type='table';\n\nThis SQL query selects and returns the names of all tables in the SQLite database catalog by checking for entries in the `sqlite\_master` table where the `type` column has a value of 'table'. The `sqlite\_master` is a special table that holds information about all other tables, indexes, triggers, views, etc. within the SQLite database. This query gives us a list of all such objects without any duplicates or extra columns since it directly targets the relevant entries in this internal catalog table."}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 56245347270, 'load\_duration': 1081768687, 'prompt\_eval\_count': 1952, 'prompt\_eval\_duration': 40838681000, 'eval\_count': 115, 'eval\_duration': 13722406000}

SELECT name FROM sqlite\_master WHERE type='table';

This SQL query selects and returns the names of all tables in the SQLite database catalog by checking for e ntries in the `sqlite\_master` table where the `type` column has a value of 'table'. The `sqlite\_master` is a special table that holds information about all other tables, indexes, triggers, views, etc. within the SQ Lite database. This query gives us a list of all such objects without any duplicates or extra columns since it directly targets the relevant entries in this internal catalog table.

Output from LLM: SELECT name FROM sqlite\_master WHERE type='table';

This SQL query selects and returns the names of all tables in the SQLite database catalog by checking for e ntries in the `sqlite\_master` table where the `type` column has a value of 'table'. The `sqlite\_master` is a special table that holds information about all other tables, indexes, triggers, views, etc. within the SQ Lite database. This query gives us a list of all such objects without any duplicates or extra columns since it directly targets the relevant entries in this internal catalog table.

Extracted SQL: SELECT name FROM sqlite\_master WHERE type='table' SELECT name FROM sqlite master WHERE type='table'

name
0 albums
1 sqlite\_sequence
2 artists
3 customers
4 employees
5 genres

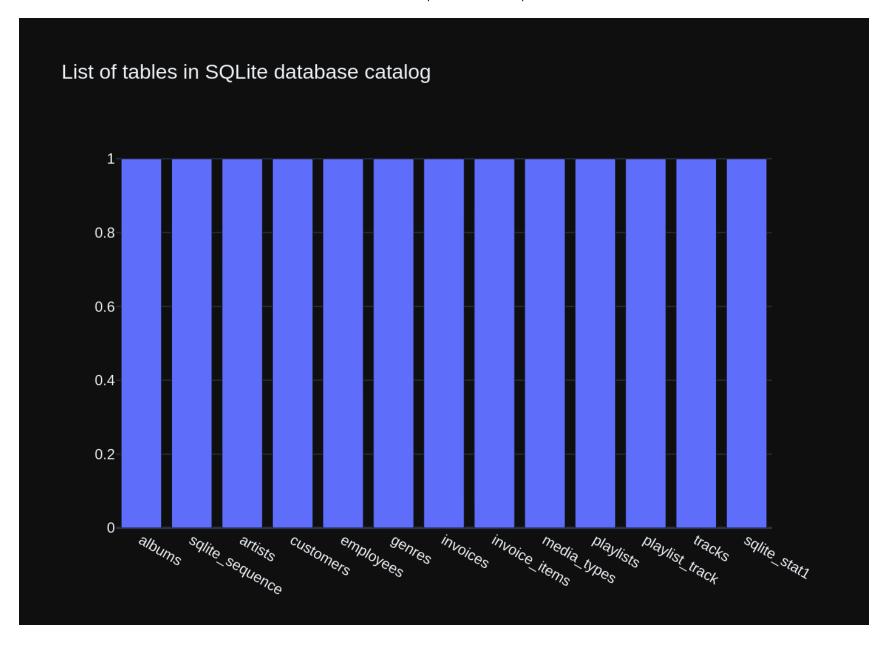
6 invoices
7 invoice\_items
8 media\_types
9 playlists
10 playlist\_track
11 tracks

options={},
keep\_alive=None
Prompt Content:

[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: 'Can you list all tables in the SQLite database catalog?'\n\nThe DataFrame was produced using this query: SELECT name FROM sqlite\_master WHERE type='table'\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n name object\ndtyp e: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}

Ollama Response:

{'model': 'phi3:latest', 'created\_at': '2024-06-14T02:37:26.583025276Z', 'message': {'role': 'assistant', 'content': "import plotly.graph\_objs as go\n\ntrace = go.Bar(x=list(df['name']), y=[1]\*len(df['name']))\n\n data = [trace]\n\nlayout = go.Layout(title='List of tables in SQLite database catalog')\n\nfig = go.Figure (data=data, layout=layout)\n\nfig.show()"}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 14534845 712, 'load\_duration': 45348429, 'prompt\_eval\_count': 162, 'prompt\_eval\_duration': 4290449000, 'eval\_count': 86, 'eval duration': 10154040000}



```
Out[17]: ("SELECT name FROM sqlite master WHERE type='table'",
                          name
           0
                        albums
           1
              sqlite sequence
           2
                       artists
           3
                     customers
           4
                     employees
           5
                        genres
           6
                      invoices
           7
                invoice items
           8
                  media types
           9
                     playlists
               playlist track
           10
           11
                        tracks
           12
                  sglite stat1,
           Figure({
               'data': [{'type': 'bar',
                         'x': [albums, sqlite sequence, artists, customers, employees,
                               genres, invoices, invoice items, media types, playlists,
                               playlist track, tracks, sqlite stat1],
                         'y': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]}],
               'layout': {'template': '...', 'title': {'text': 'List of tables in SQLite database catalog'}}
          }))
In [18]: 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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR EMENT NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n CustomerId INTEGER NOT NULL,\r\n BillingA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n rv NVARCHAR(40),\r\n **FOREIG** N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n UnitPrice NUMERIC(10.2) NOT NULL.\r FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\t Quantity INTEGER NOT NULL,\r\n ON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t \tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER P FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) N RIMARY KEY AUTOINCREMENT NOT NULL,\r\n OT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n  $VARCHAR(40).\r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL.\r\n SupportRepId INTEGER,\r\n FOREIGN KEY (Sup portRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREA EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TE TABLE "employees"\r\n(\r\n LastName NVA RCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30).\r\n ReportsTo I NTEGER,\r\n BirthDate DATETIME.\r\n HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n City NVARC PostalCode NVARCHAR(10),\r\n  $HAR(40), \r\n$ State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Email NVARCHAR(60),\r\n ne NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n FOREIGN KEY (ReportsTo) REFEREN CES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite se quence(name,seg)\n\nCREATE TABLE "playlists"\r\n(\r\n PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT N Name NVARCHAR(120)\r\n)\n\nCREATE TABLE sqlite stat1(tbl,idx,stat)\n\nCREATE TABLE "albums"\r\n ULL.\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n (\r\n rtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist track"\r\n(\r\n PlavlistId INTEGER NOT NUL CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackI TrackId INTEGER NOT NULL.\r\n L,\r\n FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDA d), r nFOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\t0N DELETE NO ACTION ON UPD TE NO ACTION,\r\n ATE NO ACTION\r\n)\n\nCREATE TABLE "media types"\r\n(\r\n MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT Name NVARCHAR(120)\ $r\n$ ) $n\n$ ===Additional Context \ $n\n$ In the SQLite database invoice mean s order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL q uery without any explanations for the question. \n2. If the provided context is almost sufficient but requi res knowledge of a specific string in a particular column, please generate an intermediate SQL guery to fin d the distinct strings in that column. Prepend the query with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relev ant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the mos Hint: order total can be found on invoices table, calculation using invoice t money overall. \n \n items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total)

AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId \nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer with the mos t invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices \nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY Tot alInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Find the customer who bought the most alb ums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, C OUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assi stant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoic es" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.Cus tomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Custom erId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n Get the total number of in voices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerI d'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assi stant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'ro le': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'c ontent': 'SELECT c.Country, COUNT(\*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY To talCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n List all customers from Canada and thei r email addresses:\n'}, {'role': 'assistant', 'content': 'SELECT c.Email, c.Country\nFROM "customers" c\nWH ERE c.Country = \'Canada\''}, {'role': 'user', 'content': ' \n Get the average invoice total for each c ustomer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nF ROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'use r', 'content': "which table stores customer's orders"}] Ollama parameters: model=phi3:latest. options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n Billin BillingState NVARCHAR(40),\r\n gAddress NVARCHAR(70).\r\n BillingCity NVARCHAR(40),\r\n BillinaCou BillingPostalCode NVARCHAR(10),\r\n ntry NVARCHAR(40),\r\n Total NUMERIC(10,2) NOT NULL,\r\n F0RE

IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL L.\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10.2) NO Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceI T NULL,\r\n d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (Tra ckid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n erId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName N VARCHAR(20) NOT NULL.\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(4 0), r nState NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r\n Email NVARCHAR(60) NOT NULL,\r\n  $ARCHAR(24), \r\n$ Fax NVARCHAR(24),\r\n SupportRepId INTEGER.\r\n FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO A CTION\r\n)\n\nCREATE TABLE \"employees\"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n L.\r\n Title NVARCHAR(3 0), r nReportsTo INTEGER.\r\n BirthDate DATETIME,\r\n HireDate DATETIME.\r\n Address NVARCHAR  $(70), \r\n$ City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode N  $VARCHAR(10).\r\n$ Phone NVARCHAR(24).\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE sqlite sequence(name,seq)\n\nCREATE TABLE \"playlists\"\r\n(\r\n PlavlistId INTEGER PRI MARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE sqlite stat1(tbl,idx,stat) \n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n ARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES \"artist s\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist track\"\r\n PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n CONSTRAINT PK PlavlistTrack  $(\r\n$ PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackI d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"media types\"\r\n(\r\n peld INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Name NVARCHAR(120)\r\n)\n\n===Additional Context  $\n = \n \$ fficient, 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 gen erate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comme nt saving intermediate sql \n3. If the provided context is insufficient, please explain why it can't be gen erated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered befor e, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n d the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on inv oices table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "conte nt": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.Custo merId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "conten Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.Cus tomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " Find the customer who bought the most albums in total quantity (across all invoices): \n"}, {"role":

"assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGRO UP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n Find the top 5 customers who bought the most albums in tota quantity is found in invoice items, \n \n l quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.Trac kId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoi ce items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n op 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistan t", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"in voice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJO IN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the m ost albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.Custome rId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.I nvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "what ar e the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT c.Country, COUN T(\*) AS TotalCustomers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all customers from Canada and their email addresses:\n"}, {"rol e": "assistant", "content": "SELECT c.Email, c.Country\nFROM \"customers\" c\nWHERE c.Country = 'Canada'"}, {"role": "user", "content": " \n Get the average invoice total for each customer:\n"}, {"role": "assist ant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJ0IN \"in voices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": "which tabl e stores customer's orders"}]

Ollama Response:

{'model': 'phi3:latest', 'created\_at': '2024-06-14T02:38:28.157625127Z', 'message': {'role': 'assistant', 'content': 'The "invoices" table stores the customer\'s orders, as it represents each individual order made by a customer in the SQLite database. Each row within this table corresponds to an invoice and contains inf ormation about that specific order, including details related to the customer who placed the order (Custome rId).'}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 61200302233, 'load\_duration': 821925, 'prom pt\_eval\_count': 1983, 'prompt\_eval\_duration': 51448853000, 'eval\_count': 62, 'eval\_duration': 8970229000} The "invoices" table stores the customer's orders, as it represents each individual order made by a custome r in the SQLite database. Each row within this table corresponds to an invoice and contains information about that specific order, including details related to the customer who placed the order (CustomerId). The "invoices" table stores the customer's orders, as it represents each individual order made by a custome r in the SQLite database. Each row within this table corresponds to an invoice and contains information about that specific order, including details related to the customer who placed the order (CustomerId). Couldn't run sql: Execution failed on sql 'The "invoices" table stores the customer's orders, as it represents each individual order made by a customer in the SQLite database. Each row within this table correspond

s to an invoice and contains information about that specific order, including details related to the custom er who placed the order (CustomerId).': near "The": syntax error

In [19]: 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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR EMENT NOT NULL.\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n BillinaA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillinaCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n rv NVARCHAR(40),\r\n **FOREIG** N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE INDEX IFK CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "customers"\r\n(\r CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL.\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVAR Country NVARCHAR(40),\r\n  $CHAR(40).\r\n$ State NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n one  $NVARCHAR(24), \r\n$ Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGE FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPD ATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE TABLE "invoic InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n e items"\r\n(\r\n InvoiceId INTEGER NO T NULL,\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Ouantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON U FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\t0N DELETE NO ACTION ON PDATE NO ACTION.\r\n UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON "invoice items" (InvoiceId)\n\nCREATE TAB LE "albums"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n NULL,\r\n ArtistId INTEGER NOT NULL,\r\n \t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineTrackId ON "invoice items" EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r (TrackId)\n\nCREATE TABLE "employees"\r\n(\r\n FirstName NVARCHAR(20) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(3 0), r nReportsTo INTEGER.\r\n BirthDate DATETIME.\r\n HireDate DATETIME.\r\n Address NVARCHAR (70), r nCity NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode N Fax NVARCHAR(24),\r\n VARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n \nCREATE TABLE "playlists"\r\n(\r\n PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n  $VARCHAR(120)\r\n)\n\n===Additional\ Context\ \n\nIn\ the\ SQLite\ database\ invoice\ means\ order\n\n===Response$ Guidelines \n1. If the provided context is sufficient, please generate a valid SQL guery without any explan ations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a spec ific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate sql \n3. If the provided context is in sufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(\*) FROM "customers"'}, {'role': 'user', 'content': ' \n Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "cus tomers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money ov Hint: order total can be found on invoices table, calculation using invoice items deta erall, \n

il table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalS pent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY

TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer who bought the most alb ums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, C OUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': '\n Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT (\*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'r ole': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJ0IN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGR0UP BY c.CustomerId\n ORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in Find the top 5 customers who bought the most albums in total quantity (across al invoice items. \n \n l invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAl bums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDE R BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in inv Find the top 5 customers who bought the most albums in total quantity (across all i oice items. \n \n nvoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFRO M "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY Tota lAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Get the average invoice total for each custom er:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': 'How many customers are there'} Ollama parameters: model=phi3:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN Billin CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL.\r\n InvoiceDate DATETIME NOT NULL.\r\n aAddress NVARCHAR(70).\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillinaCou BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n ntrv NVARCHAR(40),\r\n IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"customers  $\"\r\n(\r\n$ CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NU LastName NVARCHAR(20) NOT NULL,\r\n Address NVARCHAR(70),\r\n  $LL, \r\n$ Company NVARCHAR(80),\r\n

PostalCode NVARCHAR(1 City NVARCHAR(40),\r\n State NVARCHAR(40).\r\n Country NVARCHAR(40),\r\n Fax NVARCHAR(24),\r\n 0), r nPhone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n Support RepId INTEGER,\r\n FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREAT E TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n TrackId INTEGER NOT NULL,\r\n iceId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Ouantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON D ELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\t0 N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON \"invoice items\" (InvoiceId)\n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES Title NVARCHAR(160) NOT NULL.\r\n ArtistId INTEGER NOT NULL,\r\n \"artists\" (ArtistId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLine TrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE \"employees\"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NUL L.\r\n Title NVARCHAR(30).\r\n ReportsTo INTEGER,\r\n BirthDate DATETIME,\r\n HireDate DATETIM E.\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHA  $R(40), \r\n$ PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVA  $RCHAR(60).\r\n$ FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlists\"\r\n(\r\n PlaylistId INTEGER PRIMARY KEY AUTOINCRE MENT NOT NULL,\r\n Name  $NVARCHAR(120)\r\n)\n\n==Additional Context \n\nIn the SQLite database invoice$ means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid S QL query without any explanations for the question. \n2. If the provided context is almost sufficient but r equires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most re levant table(s). \n5. If the guestion 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": "assist ant", "content": "SELECT COUNT(\*) FROM \"customers\""}, {"role": "user", "content": " \n omer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Cust omerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n Find the top 5 custom ers who spent the most money overall. \n \n Hint: order total can be found on invoices table, calcu lation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.Cus tomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Customer Id\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the customer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistan t", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoice s\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.C ustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceI d) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.C ustomerId"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"rol

e": "assistant", "content": "SELECT c.Country, COUNT(\*) AS TotalCustomers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n Hint: album quanti ty is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quan tity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) A S TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice ite ms\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"rol e": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "c ontent": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"ro le": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "c ontent": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items \" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Get the average invoice total for each customer:\n"}, {"role": "assistant", "co ntent": "SELECT c.CustomerId. AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": "How many customers are there"}]

Ollama Response:

{'model': 'phi3:latest', 'created\_at': '2024-06-14T02:39:37.840976915Z', 'message': {'role': 'assistant', 'content': 'SELECT COUNT(\*) as TotalCustomers FROM "customers"\nThis SQL query will return the total number of unique customers present in the "customers" table. The COUNT(\*) function is an aggregate function that r eturns the number of rows present in a given table, in this case, the \'customers\' table. By enclosing it with SELECT and appending as TotalCustomers to label the output column, we get our desired count result.'}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 69638430002, 'load\_duration': 887236, 'prompt\_eval\_c ount': 1931, 'prompt\_eval\_duration': 54492266000, 'eval\_count': 92, 'eval\_duration': 14322154000} SELECT COUNT(\*) as TotalCustomers FROM "customers"

This SQL query will return the total number of unique customers present in the "customers" table. The COUNT (\*) function is an aggregate function that returns the number of rows present in a given table, in this cas e, the 'customers' table. By enclosing it with SELECT and appending as TotalCustomers to label the output c olumn, we get our desired count result.

SELECT COUNT(\*) as TotalCustomers FROM "customers"

This SQL query will return the total number of unique customers present in the "customers" table. The COUNT (\*) function is an aggregate function that returns the number of rows present in a given table, in this cas e, the 'customers' table. By enclosing it with SELECT and appending as TotalCustomers to label the output c olumn, we get our desired count result.

Couldn't run sql: Execution failed on sql 'SELECT COUNT(\*) as TotalCustomers FROM "customers"
This SQL query will return the total number of unique customers present in the "customers" table. The COUNT
(\*) function is an aggregate function that returns the number of rows present in a given table, in this cas
e, the 'customers' table. By enclosing it with SELECT and appending as TotalCustomers to label the output c
olumn, we get our desired count result.': near "SQL": syntax error

```
In []:
In [20]: 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 1, updating n_results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR CustomerId INTEGER NOT NULL.\r\n InvoiceDate DATETIME NOT NULL.\r\n EMENT NOT NULL,\r\n BillingA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillinaCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n **FOREIG** rv NVARCHAR(40),\r\n N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n tName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n ddress NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Fax NVARCHAR(24),\r\n PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineI InvoiceId INTEGER NOT NULL.\r\n d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER N OT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (I nvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n **FOREIGN** KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TABLE "media types"\r\n(\r\n R(120)\r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "employe EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NU ReportsTo INTEGER,\r\n LL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n Βi S rthDate DATETIME,\r\n HireDate DATETIME,\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n tate NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Phone NVARCHAR(2 Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employee s" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n lbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION TEGER NOT NULL.\r\n ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist track"\r\n(\r\n PlavlistId INTEGER NOT NULL.\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n TrackId INTEGER NOT NULL.\r\n F0 REIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI N, r nON\r\n)\n\nCREATE TABLE sqlite sequence(name, seq)\n\nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIM ARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n peId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFE RENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REF ERENCES "genres" (GenreId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeI d) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Addi tional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provide d context is sufficient, please generate a valid SQL guery without any explanations for the guestion. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular co lumn, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the qu

ery with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the guestion has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT c.Count ry, COUNT(\*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money overall, \n Hint: order total can be found on invoices table, calculation using invoice items detail table is un necessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "cu stomers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DE SC\nLIMIT 5'}, {'role': 'user', 'content': ' \n List all customers from Canada and their email address es:\n'}, {'role': 'assistant', 'content': 'SELECT c.Email, c.Country\nFROM "customers" c\nWHERE c.Country = \'Canada\''}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "i nvoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS T otalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\n ORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS Total Albums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nOR DER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\nFRO M "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': ' \n Hint: album quantity is f ound in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (a cross all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS Total Albums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nOR DER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the top 5 most expensive trac ks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFRO M "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customers are the re'}, {'role': 'assistant', 'content': 'SELECT COUNT(\*) FROM "customers"'}, {'role': 'user', 'content': 'wh at are the top 5 countries that customers come from?'}] Ollama parameters: model=phi3:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n Billin

BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n gAddress NVARCHAR(70),\r\n BillinaCou ntry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n F0RE IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\t0N 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 Email NVARCHAR(60) NOT PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n NULL,\r\n SupportRepId INTEGER.\r\n FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n ineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGE UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KE R NOT NULL.\r\n Quantity INTEGER NOT NULL,\r\n Y (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n OREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n \nCREATE TABLE \"media types\"\r\n(\r\n ame NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE T ABLE \"employees\"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n LastName NVARC HAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INT Address NVARCHAR(70),\r\n EGER,\r\n BirthDate DATETIME,\r\n HireDate DATETIME,\r\n City NVARCHA  $R(40), \r\n$ State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone  $NVARCHAR(24).\r\n$ Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums \"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\t0 ArtistId INTEGER NOT NULL,\r\n N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist track\"\r\n(\r\n PlavlistId INTEG CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistI ER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACT d. TrackId).\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\t0N DELETE NO ION ON UPDATE NO ACTION,\r\n ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite sequence(name, seq)\n\nCREATE TABLE \"tracks\"\r\n(\r TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n Id INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r Bytes INTEGER,\r\n Milliseconds INTEGER NOT NULL.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite 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 context is almost sufficient but requires knowledge of a specific st ring in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate sql \n3. If the provided context is insufficie nt, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the quest ion has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"ro le": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "c

ontent": "SELECT c.Country, COUNT(\*) AS TotalCustomers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the top 5 customers who spent the e most monev overall, \n \n Hint: order total can be found on invoices table, calculation using inv oice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.T otal) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.C ustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all customers fr om Canada and their email addresses:\n"}, {"role": "assistant", "content": "SELECT c.Email, c.Country\nFROM \"customers\" c\nWHERE c.Country = 'Canada'"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS Tot alAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "us Find the customer with the most invoices \n"}, {"role": "assistant", "content": er", "content": " \n "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.C ustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers wh o bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SE LECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii 0 N i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "S ELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"rol e": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "c ontent": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items \" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assi stant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLI MIT 5"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SEL ECT COUNT(\*) FROM \"customers\""}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}1 Ollama Response:

{'model': 'phi3:latest', 'created at': '2024-06-14T02:41:07.754537316Z', 'message': {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(i.\*) AS CustomerCount\nFROM "invoices" i\nGROUP BY i.BillingCoun try\nORDER BY CustomerCount DESC\nLIMIT 5\n- Response: The above SQL query will return the top 5 countries that customers come from based on the number of invoices associated with each country. It groups all invoic es by their billing country and counts the number of records (invoices) for each group, then orders them in descending order by this count to get the top countries first.'}, 'done reason': 'stop', 'done': True, 'tot al duration': 89870392143, 'load duration': 910649, 'prompt eval count': 2029, 'prompt eval duration': 6553 9874000, 'eval count': 118, 'eval duration': 23436033000}

SELECT i.BillingCountry, COUNT(i.\*) AS CustomerCount

FROM "invoices" i

GROUP BY i.BillingCountry

```
ORDER BY CustomerCount DESC
```

- Response: The above SQL query will return the top 5 countries that customers come from based on the number of invoices associated with each country. It groups all invoices by their billing country and counts the number of records (invoices) for each group, then orders them in descending order by this count to get the top countries first.

SELECT i.BillingCountry, COUNT(i.\*) AS CustomerCount FROM "invoices" i GROUP BY i.BillingCountry ORDER BY CustomerCount DESC

- Response: The above SQL query will return the top 5 countries that customers come from based on the number of invoices associated with each country. It groups all invoices by their billing country and counts the number of records (invoices) for each group, then orders them in descending order by this count to get the top countries first.

Couldn't run sql: Execution failed on sql 'SELECT i.BillingCountry, COUNT(i.\*) AS CustomerCount FROM "invoices" i GROUP BY i.BillingCountry ORDER BY CustomerCount DESC LIMIT 5

- Response: The above SQL query will return the top 5 countries that customers come from based on the number of invoices associated with each country. It groups all invoices by their billing country and counts the number of records (invoices) for each group, then orders them in descending order by this count to get the top countries first.': near "\*": syntax error

#### More SQL questions

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

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE INDEX IFK AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "alb AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NUL L.\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t \t0N 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 MediaTvpeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT Bvtes INTEGER.\r\n FOREIGN KEY (AlbumId) REFERENC NULL,\r\n UnitPrice NUMERIC(10.2) NOT NULL.\r\n ES "albums" (Albumid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFEREN CES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) RE FERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX I FK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n ArtistId INTEGER PRIMARY KEY AU Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK TrackGenreId ON "tracks" (GenreId) TOINCREMENT NOT NULL,\r\n \n\nCREATE INDEX IFK PlaylistTrackTrackId ON "playlist track" (TrackId)\n\nCREATE TABLE "playlists"\r\n(\r PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name  $NVARCHAR(120)\r\n)\n\n$ CREATE INDEX IFK TrackMediaTypeId ON "tracks" (MediaTypeId)\n\n===Additional Context \n\nIn the SQLite da tabase invoice means order $\n$ ===Response Guidelines  $\n$ 1. If the provided context is sufficient, please gen erate a valid SQL query without any explanations for the question. \n2. If the provided context is almost s ufficient but requires knowledge of a specific string in a particular column, please generate an intermedia te SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermedi ate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Pleas e use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat th e answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n List all albums and the ir corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'conten There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistI d, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFRO M "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGRO UP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n who have albums with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.A rtistId IN (\n SELECT g2.GenreId\n FROM "genres" g2\n JOIN "tracks" t2 ON g2.GenreId = t2.GenreId\n GRO UP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'r ole': 'user', 'content': ' \n Identify artists who have albums with tracks appearing in multiple genre s:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artists" a\nJ0 IN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM "genres" q2 \n JOIN "tracks" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) >  $1\n$ )\n GROUP BY a.ArtistId, a.Name'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoic e items, \n Find the top 5 customers who bought the most albums in total quantity (across all invo

ices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM

"customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = i i.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most al bums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, C OUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\n GROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'c ontent': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJ0IN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\n ORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across al l invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAl bums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDE R BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n List all genres and the number of tr acks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\n FROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.Name'}, {'role': 'user', 'content': ' Get all playlists containing at least 10 tracks and the total duration of those tracks:\n'}, {'rol e': 'assistant', 'content': 'SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDura tion\nFROM "playlist track" pt\nJOIN "playlists" p ON pt.PlaylistId = p.PlaylistId\nJOIN "tracks" t ON pt.T rackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10'}, {'role': 'user', 'con List all albums and their corresponding artist names \n'\] Ollama parameters: model=phi3:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE INDEX IFK AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT \"albums\"\r\n(\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER P RIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n aTypeId INTEGER NOT NULL.\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEG ER NOT NULL,\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreI d) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Med iaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n CREATE INDEX IFK TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"artists\"\r\n(\r\n ArtistId INTE Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK TrackGenreId ON GER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n \"tracks\" (GenreId)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON \"playlist track\" (TrackId)\n\nCREATE TAB

PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LE \"playlists\"\r\n(\r\n Name NVARCHAR(12 0)\r\n)\n\CREATE TABLE \"genres\"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n ame NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided conte xt is sufficient, please generate a valid SQL guery without any explanations for the guestion. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, p lease generate an intermediate SQL query to find the distinct strings in that column. Prepend the query wit h a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it ca n't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answe red before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.Artis tId"}, {"role": "user", "content": " \n There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT (t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"trac ks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"album s\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT g2.GenreId\n FROM \"genres\" g2\n J0I N \"tracks\" t2 ON g2.GenreId = t2.GenreId\n GROUP BY g2.GenreId\n HAVING COUNT(g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName"}, {"role": "user", "content": " \n Identify artists who hav e albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.Arti stId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.Art istId IN (\n SELECT q2.GenreId\n FROM \"genres\"  $q2\n JOIN \$  tracks\" t2 ON  $q2.GenreId = t2.GenreId\n G$ ROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name"}, {"role": "user", "con Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bo ught the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY Tota lAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "\n Hint: album quantity is found in invoice item Find the top 5 customers who bought the most albums in total quantity (across all invoice s):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"in voices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalA lbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the customer who bought the most albums i n total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT (ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJ0IN \"invoices\" i ON c.CustomerId = i.CustomerId\nJ0IN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assi stant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT

5"}, {"role": "user", "content": " \n List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"role": "user", "content": " \n Get all playlist s containing at least 10 tracks and the total duration of those tracks:\n"}, {"role": "assistant", "content": "SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM \"playlist\_track\" pt\nJOIN \"playlists\" p ON pt.PlaylistId = p.PlaylistId\nJOIN \"tracks\" t ON pt.TrackId = t.TrackId \nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10"}, {"role": "user", "content": " \n Li st all albums and their corresponding artist names \n"}]

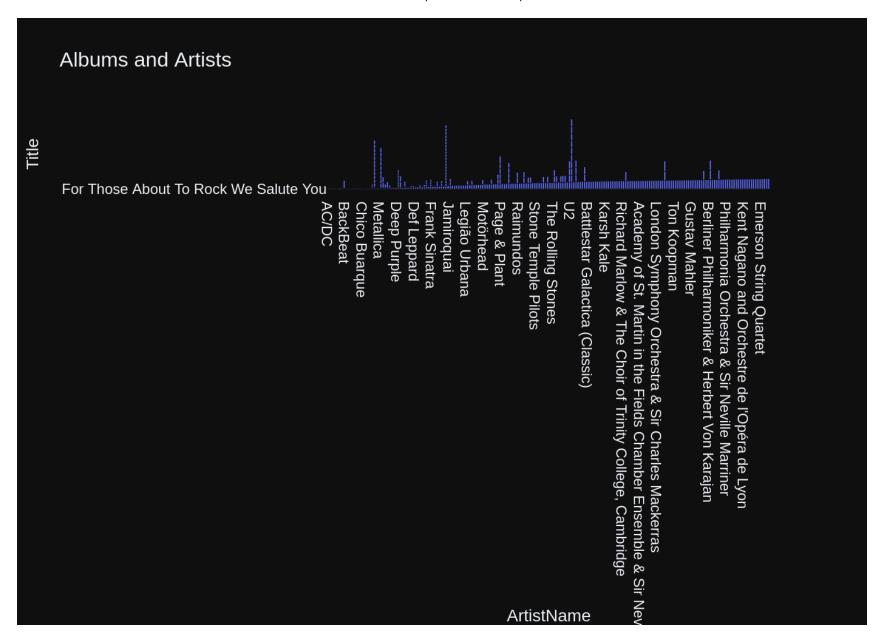
Add of existing embedding ID: a7185c88-7417-5b75-a52e-4eaef5f9deca-sql Insert of existing embedding ID: a7185c88-7417-5b75-a52e-4eaef5f9deca-sql

275

```
Ollama Response:
{'model': 'phi3:latest', 'created at': '2024-06-14T02:42:40.659318525Z', 'message': {'role': 'assistant',
content': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.Arti' '
stId = ar.ArtistId;\n\nThis SQL statement retrieves the title of each album along with the associated artis
t\'s ID and name from two tables: \'albums\' and \'artists\'. The JOIN operation links records in both tabl
es based on their shared \'ArtistId\' field, ensuring that for every album listed, there is an accompanying
artist record.'}, 'done reason': 'stop', 'done': True, 'total duration': 92861642143, 'load duration': 1610
308, 'prompt eval count': 1989, 'prompt eval duration': 70868665000, 'eval count': 118, 'eval duration': 21
408066000}
SELECT a.Title, a.ArtistId, ar.Name AS ArtistName
FROM "albums" a
JOIN "artists" ar ON a.ArtistId = ar.ArtistId:
This SQL statement retrieves the title of each album along with the associated artist's ID and name from tw
o tables: 'albums' and 'artists'. The JOIN operation links records in both tables based on their shared 'Ar
tistId' field, ensuring that for every album listed, there is an accompanying artist record.
Output from LLM: SELECT a.Title, a.ArtistId, ar.Name AS ArtistName
FROM "albums" a
JOIN "artists" ar ON a.ArtistId = ar.ArtistId;
This SQL statement retrieves the title of each album along with the associated artist's ID and name from tw
o tables: 'albums' and 'artists'. The JOIN operation links records in both tables based on their shared 'Ar
tistId' field, ensuring that for every album listed, there is an accompanying artist record.
Extracted SQL: SELECT a.Title, a.ArtistId, ar.Name AS ArtistName
FROM "albums" a
JOIN "artists" ar ON a.ArtistId = ar.ArtistId
SELECT a.Title, a.ArtistId, ar.Name AS ArtistName
FROM "albums" a
JOIN "artists" ar ON a.ArtistId = ar.ArtistId
                                                 Title ArtistId \
                 For Those About To Rock We Salute You
0
                                                               2
                                     Balls to the Wall
1
2
                                     Restless and Wild
                                                               2
3
                                                               1
                                     Let There Be Rock
                                                               3
4
                                              Bia Ones
342
                                Respighi:Pines of Rome
                                                             226
                                                             272
343
    Schubert: The Late String Quartets & String Qu...
                                   Monteverdi: L'Orfeo
                                                             273
344
345
                                 Mozart: Chamber Music
                                                             274
```

346 Koyaanisqatsi (Soundtrack from the Motion Pict...

```
ArtistName
0
                                                AC/DC
1
                                               Accept
2
                                               Accept
3
                                                AC/DC
                                            Aerosmith
4
342
                                       Eugene Ormandv
                               Emerson String Quartet
343
    C. Monteverdi, Nigel Rogers - Chiaroscuro; Lon...
345
                                        Nash Ensemble
346
                                 Philip Glass Ensemble
[347 rows x 3 columns]
Ollama parameters:
model=phi3:latest,
options={}.
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '\n List all albums and their corresponding artist names
\n'\nThe DataFrame was produced using this query: SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM
\"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId\n\nThe following is information about the res
ulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Title
                                                                        obiect\nArtistId
          object\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to
stName
chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only
one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explana
tions -- just the code."}]
Ollama Response:
{'model': 'phi3:latest', 'created at': '2024-06-14T02:43:02.303652975Z', 'message': {'role': 'assistant',
'content': "import plotly.express as px = px.bar(df, x='ArtistName', y='Title', title='Albums and Ar
tists')\nig.show()\nif df.empty:\n fig = px.pie(values=[1], labels=['Single Value'], title='Only Sing
              fig.show()"}, 'done reason': 'stop', 'done': True, 'total duration': 21617686566, 'load dur
ation': 43626818, 'prompt eval count': 210, 'prompt eval duration': 7014937000, 'eval count': 82, 'eval dur
ation': 14513956000}
```



```
Out[21]: ('SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = a
          r.ArtistId',
                                                               Title ArtistId \
           0
                             For Those About To Rock We Salute You
                                                                             1
           1
                                                  Balls to the Wall
                                                                             2
           2
                                                  Restless and Wild
           3
                                                  Let There Be Rock
                                                                             1
           4
                                                                             3
                                                           Big Ones
                                                                           . . .
           . .
           342
                                             Respighi: Pines of Rome
                                                                           226
           343
                Schubert: The Late String Quartets & String Qu...
                                                                           272
                                                Monteverdi: L'Orfeo
           344
                                                                           273
           345
                                              Mozart: Chamber Music
                                                                           274
                                                                           275
           346 Koyaanisqatsi (Soundtrack from the Motion Pict...
                                                         ArtistName
           0
                                                              AC/DC
           1
                                                             Accept
           2
                                                             Accept
           3
                                                              AC/DC
           4
                                                          Aerosmith
           342
                                                     Eugene Ormandy
           343
                                             Emerson String Quartet
           344
                C. Monteverdi, Nigel Rogers - Chiaroscuro; Lon...
           345
                                                      Nash Ensemble
           346
                                              Philip Glass Ensemble
           [347 \text{ rows } \times 3 \text{ columns}],
           Figure({
                'data': [{'alignmentgroup': 'True',
                          'hovertemplate': 'ArtistName=%{x}<br>Title=%{y}<extra></extra>',
                          'legendgroup': '',
                          'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                          'name': '',
                          'offsetgroup': ''
                          'orientation': 'v',
                          'showlegend': False,
                          'textposition': 'auto',
                          'type': 'bar',
                          'x': array(['AC/DC', 'Accept', 'Accept', ...,
                                       'C. Monteverdi, Nigel Rogers - Chiaroscuro; London Baroque; London Cornett & Sa
```

```
ckbu',
                                     'Nash Ensemble', 'Philip Glass Ensemble'], dtype=object),
                         'xaxis': 'x',
                         'y': array(['For Those About To Rock We Salute You', 'Balls to the Wall',
                                     'Restless and Wild', ..., "Monteverdi: L'Orfeo",
                                     'Mozart: Chamber Music',
                                     'Koyaanisqatsi (Soundtrack from the Motion Picture)'], dtype=object),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'template': '...',
                          'title': {'text': 'Albums and Artists'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'ArtistName'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Title'}}}
          }))
         question = """
In [22]:
             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 1, updating n results = 1

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE INDEX IFK TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK Pl aylistTrackTrackId ON "playlist track" (TrackId)\n\nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMA RY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTvp eId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n Bytes INTEGER.\r\n UnitPrice NUMERIC(10.2) NOT NULL.\r\n FOREIGN KEY (AlbumId) REFE RENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REF ERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeI d) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IN DEX IFK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDEX IFK TrackMediaTypeId ON "tracks" (MediaTypeId)\n \nCREATE TABLE "playlist track"\r\n(\r\n PlaylistId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NUL CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) R EFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Tra ckid) REFERENCES "tracks" (Trackid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IF K InvoiceLineTrackId ON "invoice items" (TrackId)\n\nCREATE INDEX IFK AlbumArtistId ON "albums" (ArtistId) PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n \n\nCREATE TABLE "plavlists"\r\n(\r\n  $NVARCHAR(120)\r\n)\n\nCREATE TABLE "genres"\r\n(\r\n$ GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL Name  $NVARCHAR(120)\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order$ \n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL guery wi thout any explanations for the question. \n2. If the provided context is almost sufficient but requires kno wledge of a specific string in a particular column, please generate an intermediate SQL query to find the d istinct strings in that column. Prepend the query with a comment saying intermediate sql \n3. If the provid ed context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant ta ble(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': '\n Find all tracks with a name containing "What" (cas e-insensitive)\n'}, {'role': 'assistant', 'content': 'SELECT \* \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'% what%\''}, {'role': 'user', 'content': ' \n Get all playlists containing at least 10 tracks and the to tal duration of those tracks:\n'}, {'role': 'assistant', 'content': 'SELECT pt.PlaylistId, p.Name AS Playli stName, SUM(t.Milliseconds) AS TotalDuration\nFROM "playlist track" pt\nJOIN "playlists" p ON pt.PlaylistId = p.PlaylistId\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt. TrackId) >= 10'}, {'role': 'user', 'content': ' \n Identify artists who have albums with tracks appear ing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName \nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreI d\n FROM "genres" g2\n JOIN "tracks" t2 ON g2.GenreId = t2.GenreId\n GROUP BY g2.GenreId\n HAVING COUNT (q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name'}, {'role': 'user', 'content': '\n List all genres an d the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreI d) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.Name'}, {'role': 'user', 'content': ' \n Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assi stant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMI T 5'}, {'role': 'user', 'content': ' \n Identify artists who have albums with tracks appearing in mult iple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "arti sts" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT g2.GenreId\n FROM "g enres" g2\n JOIN "tracks" t2 ON g2.GenreId = t2.GenreId\n GROUP BY g2.GenreId\n HAVING COUNT(g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'role': 'user', 'content': ' \n tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are Can you find the top 10 most popular artists based on the number of tracks\n'}, {'r linked by AlbumId.\n ole': 'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albu ms" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY To talTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n Find the customer who bought the most alb ums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, C OUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n List all albums and their corresponding artist names \n'}, {'rol e': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "arti sts" ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n Hint: album quantity is found i n invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums \nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.Invoic eId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'conten Find all tracks with a name containing "What" (case-insensitive)\n'}] Ollama parameters:

octalia paralleters

model=phi3:latest,

options={},

keep alive=None

Prompt Content:

[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE INDEX IFK TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON \"playlist track\" (TrackId)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n iaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTE GER NOT NULL.\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreI d) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Med iaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n CREATE INDEX IFK TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDEX IFK TrackMediaTypeId ON \"tracks\" (M ediaTypeId)\n\nCREATE TABLE \"playlist track\"\r\n(\r\n PlaylistId INTEGER NOT NULL,\r\n TrackId INT CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY EGER NOT NULL,\r\n (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE INDEX IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE INDEX IFK AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n PlaylistId INTEGER PRIMARY KEY AUTOINCREMEN Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n T NOT NULL,\r\n GenreId INTEGER PRIMARY K EY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\n===Additional Context \n\nIn the SOLite data base invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please gener ate a valid SQL query without any explanations for the question. \n2. If the provided context is almost suf ficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL guery to find the distinct strings in that column. Prepend the guery with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please us e the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the an swer exactly as it was given before. \n"}, {"role": "user", "content": " \n Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT \* \nFROM \"tracks\" \n WHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": " \n Get all playlists containing at le ast 10 tracks and the total duration of those tracks:\n"}, {"role": "assistant", "content": "SELECT pt.Play listId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM \"playlist track\" pt\nJ0IN \"pl aylists\" p ON pt.PlaylistId = p.PlaylistId\nJOIN \"tracks\" t ON pt.TrackId = t.TrackId\nGROUP BY pt.Playl istId, p.Name\nHAVING COUNT(pt.TrackId) >= 10"}, {"role": "user", "content": " \n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.A rtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a. ArtistId IN (\n SELECT q2.GenreId\n FROM \"genres\" q2\n JOIN \"tracks\" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name"}, {"role": "user", "co List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "conten t": "SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t. GenreId\nGROUP BY g.Name"}, {"role": "user", "content": " \n Find the top 5 most expensive tracks (base d on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"trac ks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n Identify artists who ha ve albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.Art istId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.Ar tistId IN (\n SELECT q2.GenreId\n FROM \"genres\" q2\n JOIN \"tracks\" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName"}, {"role": "user", "content": " \n There are 3 tables: artists, albums and tracks, where albums and artist s are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popula r artists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.Track Id) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t O N al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "conten Find the customer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.Invoic eId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": "\n all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a. ArtistId, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"ro le": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "c ontent": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.Custome

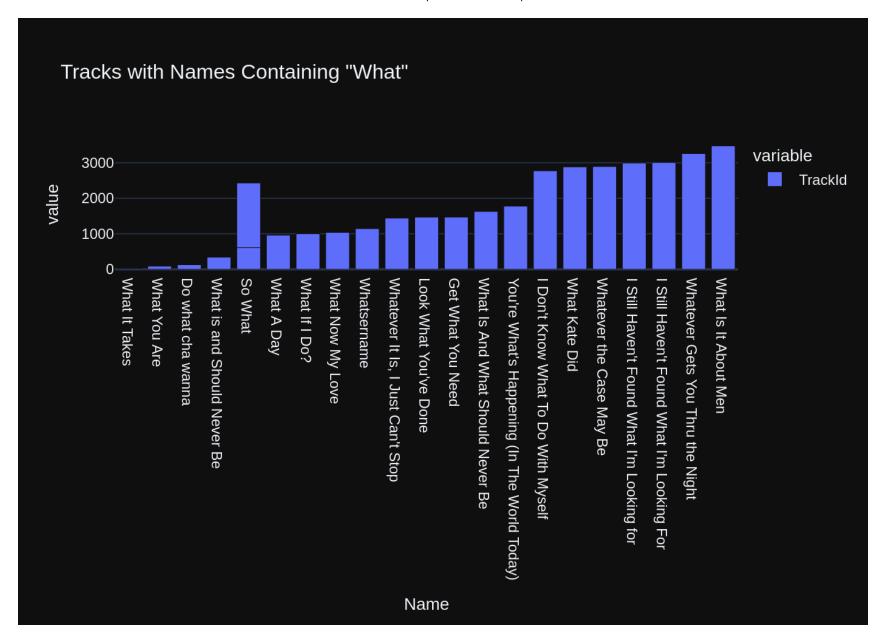
rId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "\n Find all tracks with a name containing \"What\" (case-insensitive)\n"}]

Add of existing embedding ID: 6f22268c-5062-5f11-ba2d-8555f06b409d-sql Insert of existing embedding ID: 6f22268c-5062-5f11-ba2d-8555f06b409d-sql

```
Ollama Response:
{'model': 'phi3:latest', 'created at': '2024-06-14T02:44:05.427951064Z', 'message': {'role': 'assistant',
'content': 'SELECT * \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%\''}, 'done reason': 'stop', 'done':
True, 'total_duration': 62969990811, 'load_duration': 656682, 'prompt eval count': 1938, 'prompt eval durat
ion': 58767036000, 'eval count': 23, 'eval duration': 3618252000}
SELECT *
FROM "tracks"
WHERE LOWER(Name) LIKE '%what%'
SELECT *
FROM "tracks"
WHERE LOWER(Name) LIKE '%what%'
    TrackId
                                                      Name AlbumId \
0
         26
                                             What It Takes
                                                                   5
         88
1
                                              What You Are
                                                                  10
2
        130
                                         Do what cha wanna
                                                                  13
3
        342
                              What is and Should Never Be
                                                                  30
4
        607
                                                   So What
                                                                  48
5
        960
                                                                  76
                                                What A Day
6
                                                                  80
       1000
                                             What If I Do?
7
       1039
                                          What Now My Love
                                                                  83
8
                                                                  89
       1145
                                               Whatsername
9
       1440
                                                                 116
                        Whatever It Is, I Just Can't Stop
       1469
10
                                     Look What You've Done
                                                                 119
       1470
                                                                 119
11
                                         Get What You Need
                         What Is And What Should Never Be
12
       1628
                                                                 133
13
             You're What's Happening (In The World Today)
                                                                 146
14
       1823
                                                   So What
                                                                 149
       2772
                                                                 223
15
                      I Don't Know What To Do With Myself
16
       2884
                                                                 231
                                             What Kate Did
17
       2893
                                  Whatever the Case May Be
                                                                 230
       2992
                                                                 237
18
               I Still Haven't Found What I'm Looking for
19
       3007
               I Still Haven't Found What I'm Looking For
                                                                 238
20
       3258
                                                                 255
                          Whatever Gets You Thru the Night
21
       3475
                                      What Is It About Men
                                                                 322
    MediaTypeId GenreId
                                                                     Composer \
                                      Steven Tyler, Joe Perry, Desmond Child
0
              1
                        1
1
              1
                       1
                                                    Audioslave/Chris Cornell
2
              1
                       2
                                                                  George Duke
3
              1
                       1
                                                     Jimmy Page/Robert Plant
                       2
4
              1
                                                                  Miles Davis
5
                       1
              1
                                       Mike Bordin, Billy Gould, Mike Patton
```

6 7 8 9 10 11	1 1 1 1 1	1 Da 12 4 1 4 4	ove Grohl, Taylor Hawkins, Nate Mendel, Chris carl sigman/gilbert becaud/pierre leroyer Green Day Jay Kay/Kay, Jay N. Cester C. Cester/C. Muncey/N. Cester
12	1	1	Jimmy Page, Robert Plant
13	1	14	Allen Story/George Gordy/Robert Gordy
14	1	3	Culmer/Exalt
15	1	7	None
16	3	19	None
17	3	19	None
18	1	1	Bono/Clayton, Adam/Mullen Jr., Larry/The Edge
19	1	1	U2
20	2	9	None
21	2	9 De	elroy "Chris" Cooper, Donovan Jackson, Earl C
	Milliseconds	Bytes	UnitPrice
0	310622	10144730	0.99
1	249391	5988186	0.99
2	274155	9018565	0.99
3	260675	8497116	0.99
4	564009	18360449	0.99
5	158275	5203430	0.99
6	302994	9929799	0.99
7	149995	4913383	0.99
8	252316	8244843	0.99
9	247222	8249453	0.99
10	230974	7517083	0.99
11	247719	8043765	0.99
12	287973	9369385	0.99
13	142027	4631104	0.99
14	189152	6162894	0.99
15	221387	7251478	0.99
16	2610250	484583988	1.99
17	2616410	183867185	1.99
18	353567	11542247	0.99
19	280764	9306737	0.99
20	215084	3499018	0.99
21	209573	3426106	0.99
	ama parameters		
mode	el=phi3:latest	,	

options={}. keep alive=None Prompt Content: [{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: '\n Find all tracks with a name containing \"What\" (case-in sensitive)\n'\nThe DataFrame was produced using this query: SELECT \* \nFROM \"tracks\" \nWHERE LOWER(Nam e) LIKE '%what%'\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dt vpes gives:\n TrackId int64\nName object\nAlbumId int64\nMediaTvpeId int64\nComposer int64\nGenreId object\nMilliseconds int64\nBvtes int64\n float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly c UnitPrice ode to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there i s only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}] Ollama Response: {'model': 'phi3:latest', 'created at': '2024-06-14T02:44:24.043801939Z', 'message': {'role': 'assistant', 'content': '```python\nimport plotly.express as px\n\n# Assuming that df contains at least a column named  $\$  'Name\' to visualize based on the query result\nfig = px.bar(df, x=\'Name\', y=[\'TrackId\'], title=\'Trac ks with Names Containing "What"\')\nfig.show()\n```'}, 'done reason': 'stop', 'done': True, 'total duratio n': 18586993197, 'load duration': 44760805, 'prompt eval count': 238, 'prompt eval duration': 7004063000, 'eval count': 74, 'eval duration': 11493020000}



Out[22]:	('SE	LECT * \nF	ROM "tracks"	'\nWHERE LOWER(Name) LIKE \'%what	t%\'',	
		TrackId		Name	AlbumId	\
	0	26		What It Takes	5	
	1	88		What You Are	10	
	2	130		Do what cha wanna	13	
	3	342		What is and Should Never Be	30	
	4	607		So What	48	
	5	960		What A Day	76	
	6	1000		What If I Do?	80	
	7	1039		What Now My Love	83	
	8	1145		Whatsername	89	
	9	1440	V	Whatever It Is, I Just Can't Stop	116	
	10	1469		Look What You've Done	119	
	11	1470		Get What You Need	119	
	12	1628		What Is And What Should Never Be	133	
	13		You're What'	s Happening (In The World Today)	146	
	14	1823		So What	149	
	15	2772	ΙC	Oon't Know What To Do With Myself	223	
	16	2884		What Kate Did	231	
	17	2893		Whatever the Case May Be	230	
	18	2992		even't Found What I'm Looking for	237	
	19	3007		even't Found What I'm Looking For	238	
	20	3258		Whatever Gets You Thru the Night	255	
	21	3475		What Is It About Men	322	
		MadiaTuna	Td ConnoTd			`amnasan \
	0	MediaType		Stoven Tyler lee De		Composer \
	1		1 1 1	Steven Tyler, Joe Per	lave/Chris	
	2		1 2	AddIOS		ge Duke
	3		1 1	limmy	Page/Rober	
	4		1 2	3 Illilly		es Davis
	5		1 1	Mike Bordin, Billy (		
	6		1 1	Dave Grohl, Taylor Hawkins, Nate		
	7		1 12	carl sigman/gilbert beca		
	8		1 4	care signally greater to become	•	een Day
	9		1 1		Jay Kay/K	-
	10		1 4			Cester
	11		1 4	C. Cester/C		
	12		1 1		Page, Rober	
	13		1 14	Allen Story/George (		
	14		1 3	111 1011 2 101 J, 2001 go V	-	er/Exalt
	15		1 7		2 3. 22	None
	_					-

```
3
                       19
16
                                                                          None
              3
17
                       19
                                                                          None
18
              1
                       1
                               Bono/Clayton, Adam/Mullen Jr., Larry/The Edge
              1
                       1
19
                                                                            U2
              2
20
                        9
                                                                          None
              2
                          Delroy "Chris" Cooper, Donovan Jackson, Earl C...
21
                       Bytes UnitPrice
    Milliseconds
          310622
0
                   10144730
                                   0.99
                    5988186
                                   0.99
1
          249391
2
                    9018565
          274155
                                   0.99
3
          260675
                    8497116
                                   0.99
4
          564009
                   18360449
                                   0.99
5
          158275
                                   0.99
                    5203430
6
                    9929799
                                   0.99
          302994
7
                                   0.99
          149995
                    4913383
8
                                   0.99
          252316
                    8244843
9
          247222
                                   0.99
                    8249453
10
          230974
                    7517083
                                   0.99
                                   0.99
11
          247719
                    8043765
12
          287973
                                   0.99
                    9369385
          142027
                                   0.99
13
                    4631104
          189152
                                   0.99
14
                    6162894
                    7251478
                                   0.99
15
          221387
16
         2610250
                  484583988
                                   1.99
17
         2616410
                  183867185
                                   1.99
18
          353567
                   11542247
                                   0.99
          280764
19
                    9306737
                                   0.99
20
                                   0.99
          215084
                     3499018
          209573
21
                     3426106
                                   0.99
Figure({
    'data': [{'alignmentgroup': 'True',
               'hovertemplate': 'variable=TrackId<br>Name=%{x}<br/>br>value=%{y}<extra></extra>',
               'legendgroup': 'TrackId',
               'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
               'name': 'TrackId',
               'offsetgroup': 'TrackId',
               'orientation': 'v',
               'showlegend': True,
               'textposition': 'auto',
              'type': 'bar',
              'x': array(['What It Takes', 'What You Are', 'Do what cha wanna',
```

```
'What is and Should Never Be', 'So What', 'What A Day', 'What If I Do?',
                                     'What Now My Love', 'Whatsername', "Whatever It Is, I Just Can't Stop",
                                     "Look What You've Done", 'Get What You Need',
                                     'What Is And What Should Never Be',
                                     "You're What's Happening (In The World Today)", 'So What',
                                     "I Don't Know What To Do With Myself", 'What Kate Did',
                                     'Whatever the Case May Be',
                                     "I Still Haven't Found What I'm Looking for",
                                     "I Still Haven't Found What I'm Looking For",
                                     'Whatever Gets You Thru the Night', 'What Is It About Men'],
                                    dtvpe=obiect).
                         'xaxis': 'x'.
                         'y': array([ 26, 88, 130, 342, 607, 960, 1000, 1039, 1145, 1440, 1469, 1470,
                                     1628, 1778, 1823, 2772, 2884, 2893, 2992, 3007, 3258, 3475]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'title': {'text': 'variable'}, 'tracegroupgap': 0},
                          'template': '...',
                          'title': {'text': 'Tracks with Names Containing "What"'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'value'}}}
          }))
         question = """
In [23]:
             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 1, updating n results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR InvoiceDate DATETIME NOT NULL,\r\n EMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n BillingA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n **FOREIG** rv NVARCHAR(40),\r\n N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON "invoice items" (InvoiceId)\n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KE InvoiceId INTEGER NOT NULL,\r\n Y AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n Price NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERE NCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLin eTrackId ON "invoice items" (TrackId)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KE Y AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NUL  $L.\r\n$ Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHA  $R(40), \r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n  $VARCHAR(24).\r\n$ FOREIGN KEY (SupportR epId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IN DEX IFK CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "employees"\r\n(\r\n Id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVA ReportsTo INTEGER,\r\n RCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n BirthDate DATETIME.\r\n State NVARCHAR(40), \r\n HireDate DATETIME,\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Country NVARCHAR(40),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24).\r FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DEL Email NVARCHAR(60),\r\n ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON "employees" (ReportsTo)\n\n CREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n (200) NOT NULL,\r\n AlbumId INTEGER.\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL,\r\n Bvtes INTEGER.\r\n UnitPrice NUMER FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION IC(10,2) NOT NULL,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n ON UPDATE NO ACTION.\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DEL ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL que ry without any explanations for the question. \n2. If the provided context is almost sufficient but require s knowledge of a specific string in a particular column, please generate an intermediate SQL guery to find the distinct strings in that column. Prepend the query with a comment saying intermediate sql \n3. If the p rovided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most releva nt table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as i t was given before. \n'}, {'role': 'user', 'content': ' \n Get the total number of invoices for each cu stomer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFRO M "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'use

r', 'content': ' \n Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'S ELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.Custome rId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'conten Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n Find the total number of invoic es per country:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\n FROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': ' \n Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SEL ECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJ0IN "invoice items" ii 0N i.InvoiceId = ii.InvoiceId\nGR0UP BY c.CustomerId\nORDER BY TotalA lbums DESC\nLIMIT 1'}, {'role': 'user', 'content': '\n Find all invoices since 2010 and the total amou nt invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'role': 'user', 'content': Find the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assist ant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', Hint: album quantity is found in invoice items, \n \n Find the top 5 customers wh o bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SE LECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY Tota lAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice item \n Find the top 5 customers who bought the most albums in total quantity (across all invoice s):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFR0 M "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY Tota lAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice item Find the top 5 customers who bought the most albums in total quantity (across all invoice s):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "inv oices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbum s DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n Get the total number of invoices for each customer \n'}] Ollama parameters: model=phi3:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN InvoiceDate DATETIME NOT NULL.\r\n CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n Billin gAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCou

ntrv NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n F0RE IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK InvoiceLineInv oiceId ON \"invoice items\" (InvoiceId)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n InvoiceId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL.\r UnitPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Tr ackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE \"customers\"\r\n(\r\n INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCH AR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(2 Fax NVARCHAR(24),\r\n SupportRepId INTEGER.\r\n 4),\r\n Email NVARCHAR(60) NOT NULL.\r\n FOREI GN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"employees  $\"\r\n(\r\n$ EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NUL FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER,\r\n HireDate DATETIME,\r\n Address NVARCHAR(70),\r\n thDate DATETIME.\r\n City NVARCHAR(40),\r\n ate NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON \"employees\" (ReportsTo)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT Name NVARCHAR(200) NOT NULL,\r\n NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NUL Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n L.\r\n GenreId INTEGER,\r\n es INTEGER.\r\n UnitPrice NUMERIC(10.2) NOT NULL.\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (A FOREIGN KEY (GenreId) REFERENCES \"genres\" lbumid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"me dia types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is su fficient, 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 gen erate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comme nt saying intermediate sql \n3. If the provided context is insufficient, please explain why it can't be gen erated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered befor e, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Customer Id\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"cu stomers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoi ces DESC\nLIMIT 1"}, {"role": "user", "content": " \n Get the average invoice total for each custome r:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM

\"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "use r", "content": " \n Find the total number of invoices per country:\n"}, {"role": "assistant", "content t": "SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "content": " \n Find the customer who bought the most albums in total quantity (acros s all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlb ums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": " \n "content": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"role": "user", "content": "\n Find the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoices table, calculation u sing invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n Hint: album quan tity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total qu antity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice it ems\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"rol e": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "c ontent": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"ro le": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "c ontent": "SELECT i.CustomerId. COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items \" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Get the total number of invoices for each customer\n"}]

Add of existing embedding ID: 49e67df3-a604-51f8-ad01-b8f5a2043eac-sql Insert of existing embedding ID: 49e67df3-a604-51f8-ad01-b8f5a2043eac-sql

```
Ollama Response:
{'model': 'phi3:latest', 'created at': '2024-06-14T02:45:33.534043463Z', 'message': {'role': 'assistant',
'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJ0IN "invoices" i
ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId;'}, 'done reason': 'stop', 'done': True, 'total durat
ion': 69377181955, 'load duration': 734671, 'prompt eval count': 1952, 'prompt eval duration': 59862022000,
'eval count': 54, 'eval duration': 8625404000}
SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId:
Output from LLM: SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId:
Extracted SQL: SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
    CustomerId TotalInvoices
0
             1
                            7
             2
1
                            7
2
             3
                            7
3
4
             5
                            7
             6
                            7
             7
                            7
6
7
             8
                            7
8
             9
                            7
9
            10
                            7
10
            11
                            7
11
            12
                            7
12
            13
                            7
13
            14
14
            15
                            7
                            7
15
            16
                            7
16
            17
```

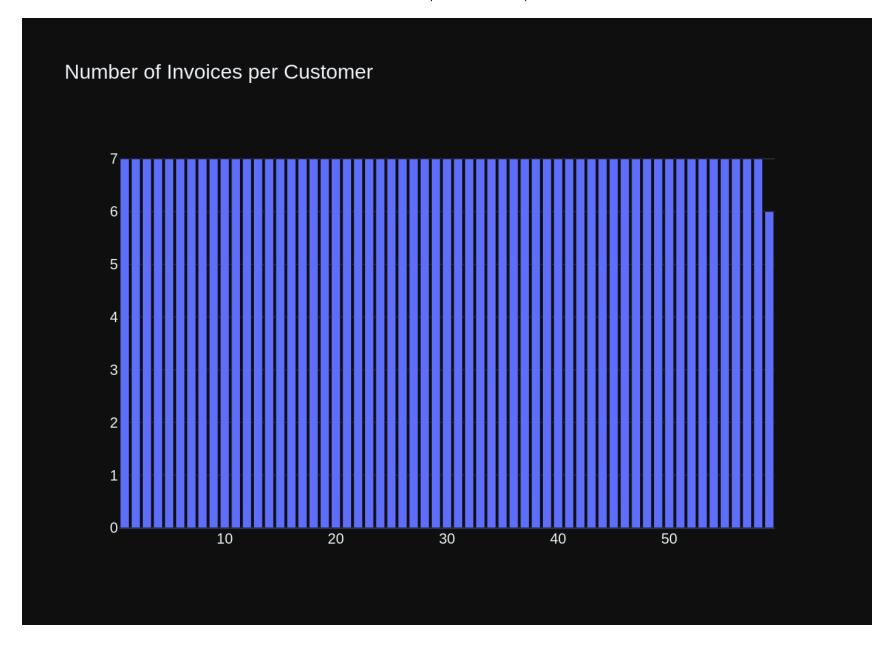
19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54	20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 56 57 57 57 57 57 57 57 57 57 57 57 57 57		777777777777777777777777777777777777777	
52	52 53		7 7	
54	55		7	
55 56	56 57		7 7	
57	58		7	
58	59		6	
Ollama parameters: model=phi3:latest,				

options={},
keep\_alive=None
Prompt Content:

[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: '\n Get the total number of invoices for each customer\n'\n \nThe DataFrame was produced using this query: SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFR OM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\n\nThe follow ing is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId int64\nTotalInvoices int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python pl otly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If t here is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]

Ollama Response:

{'model': 'phi3:latest', 'created\_at': '2024-06-14T02:45:51.301740474Z', 'message': {'role': 'assistant',
'content': "import plotly.graph\_objects as go\n\nfig = go.Figure()\nfig.add\_trace(go.Bar(x=df['CustomerI
d'], y=df['TotalInvoices']))\nfig.update\_layout(title='Number of Invoices per Customer')\nfig.show()"}, 'do
ne\_reason': 'stop', 'done': True, 'total\_duration': 17741480952, 'load\_duration': 41389443, 'prompt\_eval\_co
unt': 217, 'prompt eval duration': 6784682000, 'eval count': 68, 'eval duration': 10870475000}



Out[23]: ('SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.Cus
tomerId = i.CustomerId\nGROUP BY c.CustomerId',

	comerid/uGKOUP B
	TotalInvoices
1	7
2	7
3	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
	7
39	7
	CustomerId  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39

```
7
         39
                   40
                                 7
         40
                   41
                                 7
         41
                   42
                                 7
         42
                   43
         43
                   44
                                 7
                                 7
                   45
         44
                                 7
         45
                   46
                                 7
         46
                   47
                                 7
         47
                   48
         48
                   49
                                 7
                                 7
                   50
         49
                                 7
         50
                   51
                                 7
         51
                   52
         52
                   53
                                 7
                                 7
         53
                   54
                                 7
         54
                   55
         55
                                 7
                   56
                   57
                                 7
         56
                                 7
         57
                   58
                                 6,
         58
                   59
         Figure({
            'data': [{'type': 'bar',
                     'x': array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
                               19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
                               37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
                               55, 56, 57, 58, 59]),
                     7, 7, 7, 7, 7, 7, 7, 7, 7, 6])}],
            'layout': {'template': '...', 'title': {'text': 'Number of Invoices per Customer'}}
         }))
        question = """
In [24]:
           Find the total number of invoices per country:
        0.00
        vn.ask(question=question)
      Number of requested results 10 is greater than number of elements in index 1, updating n results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR EMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n BillinaA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillinaCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n rv NVARCHAR(40),\r\n **FOREIG** N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n UnitPrice NUMERIC(10.2) NOT NULL.\r FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\t Quantity INTEGER NOT NULL,\r\n ON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t \tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (Custome rId)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON "invoice items" (InvoiceId)\n\nCREATE INDEX IFK InvoiceLin eTrackId ON "invoice items" (TrackId)\n\nCREATE TABLE "employees"\r\n(\r\n EmployeeId INTEGER PRIMARY KE Y AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NUL L.\r\n Title NVARCHAR(30).\r\n ReportsTo INTEGER,\r\n BirthDate DATETIME.\r\n HireDate DATETIM E, r nAddress NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHA  $R(40), \r\n$ PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVA FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION O  $RCHAR(60).\r\n$ N UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMEN T NOT NULL.\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company City NVARCHAR(40),\r\n  $NVARCHAR(80).\r\n$ Address NVARCHAR(70),\r\n State NVARCHAR(40),\r\n Coun PostalCode NVARCHAR(10),\r\n trv NVARCHAR(40),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n SupportRepId INTEGER,\r\n Email NVARCHAR(60) NOT NULL,\r\n FOREIGN KEY (SupportRepId) REFERENCES "em ployees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r Title NVARCHAR(160) NOT NULL,\r\n Albumid INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n stId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\cREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREM ENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "med ia types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeRe portsTo ON "employees" (ReportsTo)\n\n\===Additional Context \n\nIn the SQLite database invoice means orde r\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query w ithout any explanations for the question. \n2. If the provided context is almost sufficient but requires kn owledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate sgl \n3. If the provi ded context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant t able(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it wa s given before. \n'}, {'role': 'user', 'content': ' \n Find the total number of invoices per countr

y:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\nFROM "invoice s" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': '\n Get the total number of invoices fo r each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvo ices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'rol e': 'user', 'content': ' \n Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.Invo iceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'role': 'user', 'content': ' \n Find the custome r with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId \nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': '\n Hint: album quantity is foun d in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (acro ss all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS To talAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId \nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found \n Find the top 5 customers who bought the most albums in total quantity (across in invoice items, \n all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums \nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n List all invoices with a total exceeding \$10:\n'}, {'role': 'assistant', 'content': 'SELECT \* \nFROM "invoices" \nWHERE Total > 10.00'}, {'role': 'u ser', 'content': ' \n Hint: album quantity is found in invoice items, \n \n Find the top 5 custom ers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'conten t': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.Cust omerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n Find the top 5 customers who spent th e most money overall. \n \n Hint: order total can be found on invoices table, calculation using inv oice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.T otal) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Custo merId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer who bou ght the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.C ustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbu ms DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Find the total number of invoices per countr v:\n'}] Ollama parameters: model=phi3:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n

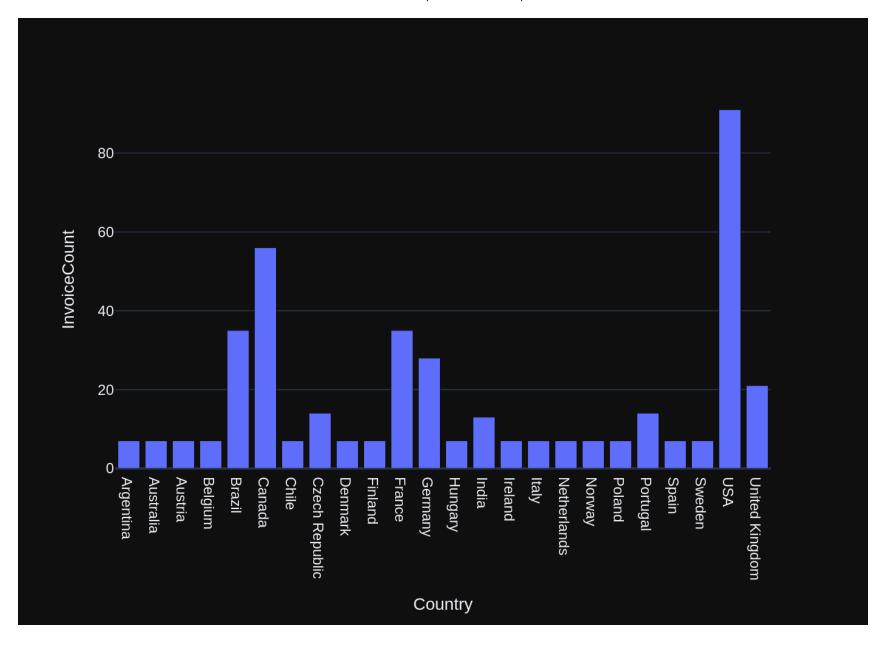
BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillinaCou gAddress NVARCHAR(70).\r\n F0RE ntry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL \r\n)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL.\r\n UnitPrice NUMERIC(10,2) NO Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceI T NULL,\r\n d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (Tra ckid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invoi ces\" (CustomerId)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON \"invoice items\" (InvoiceId)\n\nCREATE INDE X IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE \"employees\"\r\n(\r\n d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVAR CHAR(20) NOT NULL.\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER,\r\n BirthDate DATETIME.\r\n HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON D Email NVARCHAR(60),\r\n ELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n CustomerId INTEGER PRIMA RY KEY AUTOINCREMENT NOT NULL.\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT N City NVARCHAR(40),\r\n ULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n  $HAR(40), \r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n  $NVARCHAR(24).\r\n$ Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n FOREIGN KEY (Support RepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TABLE \"albums\"\r\n(\r\n Title NVARCHAR(160) FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistI NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTE GER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL.\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds I Bvtes INTEGER,\r\n NTEGER NOT NULL.\r\n FOREIGN KEY (Album UnitPrice NUMERIC(10,2) NOT NULL,\r\n Id) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Ge nreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)  $\n\n\$  IFK EmployeeReportsTo ON \"employees\" (ReportsTo)\n\n\===Additional Context \n\nIn the S OLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, pl ease generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an in termediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying i ntermediate sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n 4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please r epeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "content": " \n he total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, C OUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerI d\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\n FROM \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"role": "user", "conte Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.Cu stomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the m ost albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.Custome rId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the m ost albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.Custome rId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.I nvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all invoices with a total exceeding \$10:\n"}, {"role": "assistant", "content": "SELECT \* \nFROM \"invo ices\" \nWHERE Total > 10.00"}, {"role": "user", "content": " \n Hint: album quantity is found in invoi Find the top 5 customers who bought the most albums in total quantity (across all inv oices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.Invoice Id = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": Find the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assist ant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "use Find the customer who bought the most albums in total quantity (across all invoice r", "content": " \n s): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"c ustomers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " Find the total number of invoices per country:\n"}] Ollama Response: {'model': 'phi3:latest', 'created at': '2024-06-14T02:47:03.00009704Z', 'message': {'role': 'assistant', 'c ontent': 'SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM "customers" customer\nJOI N "invoices" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country;'}, 'done reason': 'sto p', 'done': True, 'total duration': 71592818186, 'load duration': 647105, 'prompt eval count': 1987, 'promp t eval duration': 61680612000, 'eval count': 52, 'eval duration': 8996605000} SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount FROM "customers" customer JOIN "invoices" invoice ON customer.CustomerId = invoice.CustomerId GROUP BY Country: Output from LLM: SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount FROM "customers" customer JOIN "invoices" invoice ON customer.CustomerId = invoice.CustomerId

```
GROUP BY Country;
Extracted SQL: SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount
FROM "customers" customer
JOIN "invoices" invoice ON customer.CustomerId = invoice.CustomerId
GROUP BY Country
SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount
FROM "customers" customer
JOIN "invoices" invoice ON customer.CustomerId = invoice.CustomerId
GROUP BY Country
           Country InvoiceCount
         Argentina
0
                               7
                               7
1
         Australia
2
                               7
           Austria
                               7
3
           Belgium
4
            Brazil
                              35
5
            Canada
                              56
6
             Chile
                               7
7
    Czech Republic
                              14
8
           Denmark
                               7
9
                               7
           Finland
10
                              35
            France
11
           Germany
                              28
12
                               7
           Hungary
13
             India
                              13
14
           Ireland
                               7
15
                               7
             Italy
                               7
16
       Netherlands
                               7
17
            Norway
18
            Poland
                               7
19
          Portugal
                              14
20
                               7
             Spain
                               7
21
            Sweden
22
               USA
                              91
23 United Kingdom
                              21
Ollama parameters:
model=phi3:latest,
options={},
keep alive=None
Prompt Content:
```

[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: '\n Find the total number of invoices per country:\n'\n\nThe DataFrame was produced using this query: SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount

\nFROM \"customers\" customer\nJOIN \"invoices\" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes g ives:\n Country object\nInvoiceCount int64\ndtype: object"}, {"role": "user", "content": "Can y ou generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas da taframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Pytho n code. Do not answer with any explanations -- just the code."}]
Ollama Response:

{'model': 'phi3:latest', 'created\_at': '2024-06-14T02:47:22.873100357Z', 'message': {'role': 'assistant',
'content': "import plotly.express as px\n\nfig = px.bar(df, x='Country', y='InvoiceCount')\nfig.show()\n\ni
f len(df) == 1:\n fig = px.pie(df, values=['InvoiceCount'], names=['Country'])\n fig.show()"}, 'done\_
reason': 'stop', 'done': True, 'total\_duration': 19845616463, 'load\_duration': 46145478, 'prompt\_eval\_coun
t': 211, 'prompt eval duration': 6955079000, 'eval count': 75, 'eval duration': 12799524000}



```
Out[24]: ('SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM "customers" customer\nJOIN "invo
          ices" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country',
                      Country InvoiceCount
           0
                    Argentina
                                           7
           1
                    Australia
                                           7
           2
                                           7
                      Austria
           3
                                           7
                      Belgium
           4
                       Brazil
                                          35
           5
                                          56
                       Canada
           6
                        Chile
                                           7
           7
               Czech Republic
                                          14
           8
                      Denmark
                                           7
           9
                      Finland
                                           7
                                          35
           10
                       France
                                          28
           11
                      Germany
           12
                                           7
                      Hungary
           13
                        India
                                          13
           14
                      Ireland
                                           7
                                           7
           15
                        Italy
           16
                  Netherlands
                                           7
           17
                                           7
                       Norway
                                           7
           18
                       Poland
                     Portugal
           19
                                          14
           20
                                           7
                        Spain
           21
                       Sweden
                                           7
                                          91
           22
                          USA
           23 United Kingdom
                                          21,
           Figure({
               'data': [{'alignmentgroup': 'True',
                          'hovertemplate': 'Country=%{x}<br>InvoiceCount=%{y}<extra></extra>',
                          'legendgroup': '',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                         'name': '',
                         'offsetgroup': '',
                         'orientation': 'v',
                         'showlegend': False,
                         'textposition': 'auto',
                          'type': 'bar',
                         'x': array(['Argentina', 'Australia', 'Austria', 'Belgium', 'Brazil', 'Canada',
                                      'Chile', 'Czech Republic', 'Denmark', 'Finland', 'France', 'Germany',
                                      'Hungary', 'India', 'Ireland', 'Italy', 'Netherlands', 'Norway',
```

'Poland', 'Portugal', 'Spain', 'Sweden', 'USA', 'United Kingdom'],

```
dtype=object),
                         'xaxis': 'x',
                         'y': array([ 7,  7,  7,  7,  35,  56,  7,  14,  7,  7,  35,  28,  7,  13,  7,  7,  7,  7,
                                      7, 14, 7, 7, 91, 21]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Country'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceCount'}}}
          }))
         question = """
In [25]:
             List all invoices with a total exceeding $10:
         0.00
         vn.ask(question=question)
        Number of requested results 10 is greater than number of elements in index 1, updating n results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER NOT NULL.\r\n InvoiceId INTEGER NOT NULL,\r\n ice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERE NCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLin eInvoiceId ON "invoice items" (InvoiceId)\n\nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n BillingState NVARCHAR(40),\r\n BillingAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10).\r\n ingCountry NVARCHAR(40),\r\n Total NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO N\r\n)\n\nCREATE INDEX IFK InvoiceLineTrackId ON "invoice items" (TrackId)\n\nCREATE INDEX IFK InvoiceCusto merId ON "invoices" (CustomerId)\n\nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCRE MENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" Bvtes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (GenreId) REFERENCES "genres" FOREIGN KEY (MediaTypeId) REFERENCES "med (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n ia types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeRe portsTo ON "employees" (ReportsTo)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KEY A FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n UTOINCREMENT NOT NULL,\r\n Address NVARCHAR(70),\r\n Company NVARCHAR(80),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Fax NVARCHAR(24),\r Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n FOREIGN KEY (SupportRepId) REFERENC ES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "employee  $s"\r\n(\r\n$ EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NUL ReportsTo INTEGER,\r\n L.\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n Bir thDate DATETIME,\r\n HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n St ate NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employees" (E mploveeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON "customers" (SupportRepId) $\n\n===Additional$  Context  $\n\n$  the SQLite database invoice means order $\n\n===$ Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without a ny explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate sql \n3. If the provided conte xt is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given b efore. \n'}, {'role': 'user', 'content': '\n List all invoices with a total exceeding \$10:\n'}, {'rol e': 'assistant', 'content': 'SELECT \* \nFROM "invoices" \nWHERE Total > 10.00'}, {'role': 'user', 'conten Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'conten t': 'SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01 -01\'\nGROUP BY i.InvoiceDate'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent t \n he most money overall, \n Hint: order total can be found on invoices table, calculation using in voice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i. Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Cust omerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n Hint: album quantity is f ound in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (a cross all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerI d\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found Find the top 5 customers who bought the most albums in total quantity (across in invoice items, \n \n all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums \nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.Invoic eId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'conten Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c \nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\n GROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': '\n um quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in t otal quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.T rackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Get the total numb er of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.Invoic eId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Cus tomerId'}, {'role': 'user', 'content': ' \n Find the customer with the most invoices \n'}, {'role': 'a ssistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Find the total number of invoices per country:\n'}, {'role': 'assistan t', 'content': 'SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.BillingCo untry'}, {'role': 'user', 'content': '\n List all invoices with a total exceeding \$10:\n'}] Ollama parameters: model=phi3:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY K EY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n tPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCE S \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) RE FERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK Invo

iceLineInvoiceId ON \"invoice items\" (InvoiceId)\n\nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGE R PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL.\r\n InvoiceDate DATETIME NOT NULL,\r\n BillingAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(4 0), r nBillingCountry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10.2) NOT NULL,\r\n FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE IN DEX IFK InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEG ER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n GenreId INTEGER,\r\n Milliseconds I MediaTypeId INTEGER NOT NULL,\r\n Composer NVARCHAR(220),\r\n NTEGER NOT NULL,\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (Album Id) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Ge nreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE INDEX IFK EmployeeReportsTo ON \"employees\" (ReportsTo)\n\nCREATE TABLE \"customers\"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n Name NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR  $(40), \r\n$ State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone  $NVARCHAR(24).\r\n$ Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER.\r FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDAT E NO ACTION\r\n)\n\nCREATE TABLE \"employees\"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL.\r\n Title NVARCHA BirthDate DATETIME,\r\n  $R(30), \r\n$ ReportsTo INTEGER,\r\n HireDate DATETIME.\r\n Address NVARCH PostalCode  $AR(70).\r\n$ City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n  $NVARCHAR(10), \r\n$ Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60).\r\n **FOREIGN** KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n\n\===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficien t, please generate a valid SQL query without any explanations for the question. \n2. If the provided contex t is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL guery to find the distinct strings in that column. Prepend the guery with a comment say ing intermediate sql \n3. If the provided context is insufficient, please explain why it can't be generate d. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, ple ase repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n nvoices with a total exceeding \$10:\n"}, {"role": "assistant", "content": "SELECT \* \nFROM \"invoices\" \nW HERE Total > 10.00"}, {"role": "user", "content": " \n Find all invoices since 2010 and the total amoun t invoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"role": "user", "content": Find the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assist ant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "use r", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customer

s who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" i i ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "use r", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customer s who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.Cust omerId = i.CustomerId\nJ0IN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDE R BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the customer who bought the mo st albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.Custome rId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Custom erId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"r ole": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\n JOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\n LIMIT 5"}, {"role": "user", "content": " \n Get the total number of invoices for each customer\n"}, {"r ole": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.Custome rId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Cus tomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": "\n Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT i.BillingCount ry, COUNT(\*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "conten t": " \n List all invoices with a total exceeding \$10:\n"}] Ollama Response: {'model': 'phi3:latest', 'created at': '2024-06-14T02:48:29.210491467Z', 'message': {'role': 'assistant', 'content': 'SELECT \*\nFROM "invoices" \nWHERE (Total > 10)\nGROUP BY InvoiceId;'}, 'done reason': 'stop', 'done': True, 'total duration': 66235136309, 'load duration': 682874, 'prompt eval count': 1966, 'prompt ev al duration': 60972001000, 'eval count': 27, 'eval duration': 4441250000} SELECT \* FROM "invoices" WHERE (Total > 10) GROUP BY InvoiceId: Output from LLM: SELECT \* FROM "invoices" WHERE (Total > 10) GROUP BY InvoiceId: Extracted SQL: SELECT \* FROM "invoices" WHERE (Total > 10) GROUP BY InvoiceId SELECT \*

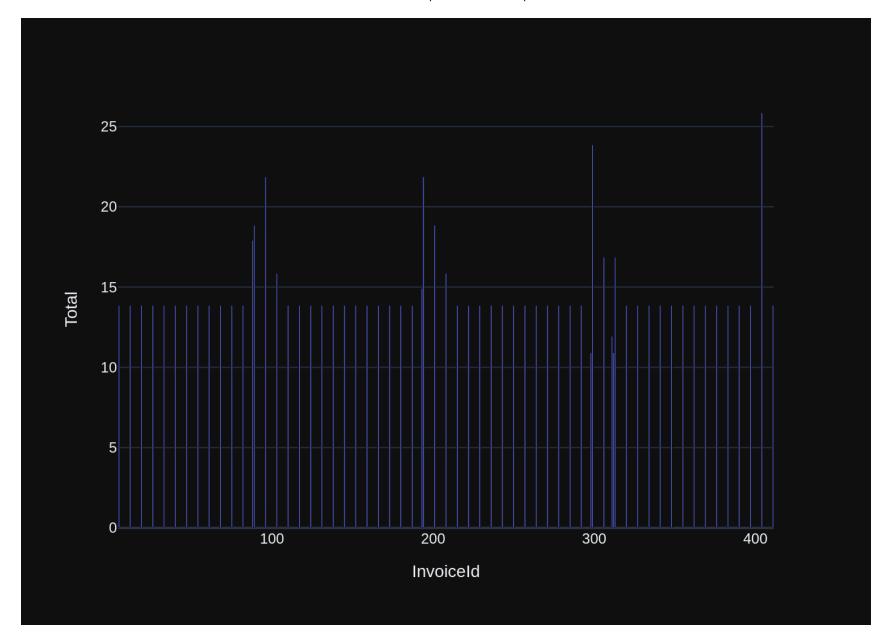
	FROM "invoices" WHERE (Total > 10)							
GROUP BY InvoiceId								
	InvoiceId	CustomerId	Inv	oiceDate/	Bil	lingAddress		
0	5	23	2009-01-11	00:00:00	69 S	Salem Street		
1	12	2	2009-02-11	00:00:00	Theodor-Heus	s-Straße 34		
2	19	40	2009-03-14	00:00:00	8,	Rue Hanovre		
3	26	19	2009-04-14	00:00:00	1 In	finite Loop		
4	33	57	2009-05-15	00:00:00	Call	e Lira, 198		
59	383	10	2013-08-12		Rua Dr. Falcão			
60	390	48	2013-09-12			racht 120bg		
61	397	27	2013-10-13			N Park Ave		
62	404	6	2013-11-13			lská 3174/6		
63	411	44	2013-12-14	00:00:00	Port	haninkatu 9		
Е	BillingCity	BillingState	e BillingCo	ountry Bi	llingPostalCode	Total		
0	Boston	MA	١	USA	2113	13.86		
1	Stuttgart	None	e Ge	ermany	70174	13.86		
2	Paris	None	<u>.</u>	rance	75002	13.86		
3	Cupertino	CA	١	USA	95014	13.86		
4	Santiago	None	2	Chile	None	13.86		
59	São Paulo	SF		Brazil	01007-010	13.86		
60	Amsterdam	VV			1016	13.86		
61	Tucson	AZ		USA	85719	13.86		
62	Prague	None			14300	25.86		
63	Helsinki	None	e F:	inland	00530	13.86		
[64 rows x 9 columns]								
Ollama parameters:								

[64 rows x 9 columns
Ollama parameters:
model=phi3:latest,
options={},
keep\_alive=None
Prompt Content:

[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: '\n List all invoices with a total exceeding \$10:\n'\n\nThe DataFrame was produced using this query: SELECT \*\nFROM \"invoices\" \nWHERE (Total > 10)\nGROUP BY Invoice Id\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n InvoiceId int64\nCustomerId int64\nInvoiceDate object\nBillingAddress object\nBillingCity object\nBillingState object\nBillingCountry object\nBillingPo stalCode object\nTotal float64\ndtype: object"}, {"role": "user", "content": "Can you ge

nerate the Python plotly code to 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 Indicator. Respond with only Python cod e. Do not answer with any explanations -- just the code."}]
Ollama Response:

{'model': 'phi3:latest', 'created\_at': '2024-06-14T02:48:47.303291729Z', 'message': {'role': 'assistant', 'content': "import plotly.express as  $px\in (df, x='InvoiceId', y='Total') \in (hnift) \in (f, x='InvoiceId', y='Total') \in (f, x='InvoiceId', y='Total')$ 



```
Out[25]: ('SELECT *\nFROM "invoices" \nWHERE (Total > 10)\nGROUP BY InvoiceId',
               InvoiceId CustomerId
                                               InvoiceDate
                                                                        BillingAddress \
                       5
           0
                                   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
                                       2009-04-14 00:00:00
                                                                       1 Infinite Loop
           4
                      33
                                   57 2009-05-15 00:00:00
                                                                       Calle Lira, 198
                      . . .
                                  . . .
                     383
           59
                                   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
                                   44 2013-12-14 00:00:00
                     411
                                                                       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
                                     ۷V
                                            Netherlands
                                                                      1016 13.86
                                                    USA
                                                                     85719 13.86
           61
                   Tucson
                                     ΑZ
           62
                                                                     14300 25.86
                   Prague
                                   None
                                         Czech Republic
           63
                 Helsinki
                                   None
                                                Finland
                                                                     00530 13.86
           [64 \text{ rows } \times 9 \text{ columns}],
           Figure({
               'data': [{'alignmentgroup': 'True',
                          'hovertemplate': 'InvoiceId=%{x}<br>Total=%{y}<extra></extra>',
                          'legendgroup': '',
                          'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                          'name': '',
                          'offsetgroup': '',
                          'orientation': 'v',
                          'showlegend': False,
                          'textposition': 'auto',
                          'type': 'bar',
                          'x': array([ 5, 12, 19, 26, 33, 40, 47, 54, 61, 68, 75, 82, 88, 89,
                                       96, 103, 110, 117, 124, 131, 138, 145, 152, 159, 166, 173, 180, 187,
                                      193, 194, 201, 208, 215, 222, 229, 236, 243, 250, 257, 264, 271, 278,
```

```
285, 292, 298, 299, 306, 311, 312, 313, 320, 327, 334, 341, 348, 355,
                                    362, 369, 376, 383, 390, 397, 404, 411]),
                         'xaxis': 'x'.
                         'y': array([13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                                    13.86, 13.86, 17.91, 18.86, 21.86, 15.86, 13.86, 13.86, 13.86, 13.86,
                                     13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 14.91, 21.86,
                                    18.86, 15.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86, 13.86,
                                     13.86, 13.86, 13.86, 13.86, 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, 25.86, 13.86]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                         'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceId'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Total'}}}
          }))
         question = """
In [26]:
             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 1, updating n results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR CustomerId INTEGER NOT NULL,\r\n EMENT NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n BillinaA ddress NVARCHAR(70).\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n rv NVARCHAR(40),\r\n **FOREIG** N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n UnitPrice NUMERIC(10.2) NOT NULL.\r FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\t Quantity INTEGER NOT NULL,\r\n ON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t \t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON "invoice items" (InvoiceId)\n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK InvoiceLin eTrackId ON "invoice items" (TrackId)\n\nCREATE TABLE "employees"\r\n(\r\n EmployeeId INTEGER PRIMARY KE Y AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NUL L.\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER,\r\n BirthDate DATETIME.\r\n HireDate DATETIM E, r nAddress NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHA  $R(40), \r\n$ PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n  $RCHAR(60).\r\n$ FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION O N UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMEN T NOT NULL.\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company City NVARCHAR(40),\r\n NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n State NVARCHAR(40),\r\n Coun trv NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n SupportRepId INTEGER,\r\n Email NVARCHAR(60) NOT NULL,\r\n FOREIGN KEY (SupportRepId) REFERENCES "em ployees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n(\r TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n MediaTypeId INTEGER NOT NULL,\r\n Id INTEGER.\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r Milliseconds INTEGER NOT NULL,\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (Albumid) REFERENCES "albums" (Albumid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO A CTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist trac PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n  $k"\r\n(\r\n$ CONSTRAINT PK Playlis tTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Playlist Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (Trac kId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite data base invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please gener ate a valid SQL query without any explanations for the question. \n2. If the provided context is almost suf ficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate

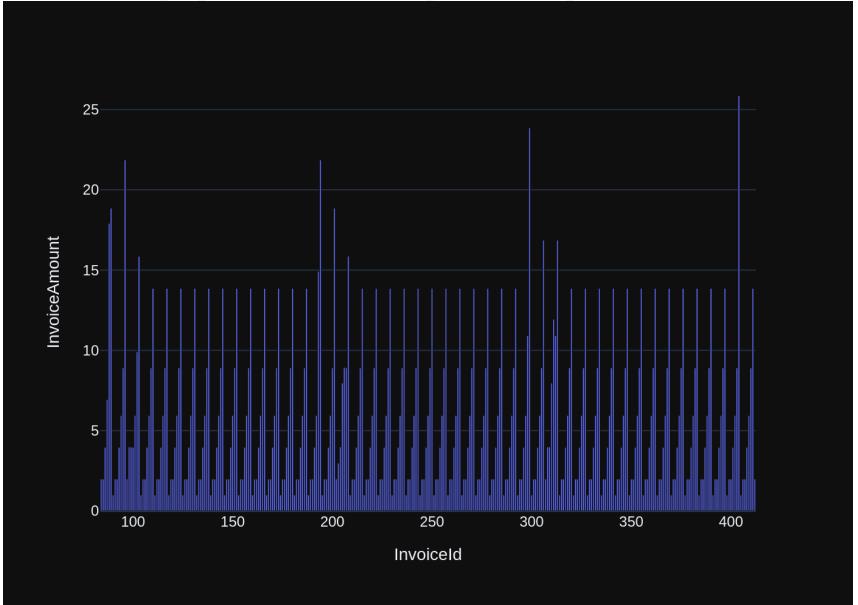
sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please u se the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the a nswer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n Find all invoices since 20 10 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'rol e': 'user', 'content': ' \n List all invoices with a total exceeding \$10:\n'}, {'role': 'assistant', 'c ontent': 'SELECT \*\nFROM "invoices" \nWHERE (Total > 10)\nGROUP BY InvoiceId'}, {'role': 'user', 'content': List all invoices with a total exceeding \$10:\n'}, {'role': 'assistant', 'content': 'SELECT \* \nFR OM "invoices" \nWHERE Total > 10.00'}, {'role': 'user', 'content': ' \n Find the total number of invoic es per country:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\n FROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': ' \n Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT Customer.Country, COUNT(invoice.Invo iceId) AS InvoiceCount\nFROM "customers" customer\nJOIN "invoices" invoice ON customer.CustomerId = invoic e.CustomerId\nGROUP BY Country'}, {'role': 'user', 'content': ' \n Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoic es\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money overall, \n order total can be found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c \nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS To talAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers wh o bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SE LECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i. InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'co Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' Find all invoices since 2010 and the total amount invoiced:\n'}] Ollama parameters: model=phi3:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN InvoiceDate DATETIME NOT NULL.\r\n CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL.\r\n Billin gAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCou

BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n F0RE ntrv NVARCHAR(40),\r\n IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL InvoiceId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL.\r\n L.\r\n UnitPrice NUMERIC(10,2) NO T NULL,\r\n Ouantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceI FOREIGN KEY (TrackId) REFERENCES \"tracks\" (Tra d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n ckid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON \"in voice items\" (InvoiceId)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDE X IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE \"employees\"\r\n(\r\n d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVAR CHAR(20) NOT NULL.\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER,\r\n BirthDate DATETIME,\r\n HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r\n Fax NVARCHAR(24),\r Phone NVARCHAR(24),\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON D Email NVARCHAR(60).\r\n ELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n CustomerId INTEGER PRIMA RY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL.\r\n LastName NVARCHAR(20) NOT N ULL.\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n  $HAR(40), \r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n  $NVARCHAR(24).\r\n$ Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n FOREIGN KEY (Support RepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n NOT NULL.\r\n GenreId INTEGER,\r\n Milliseconds INTEGER NOT NULL,\r\n UnitPrice NUMERIC(1 oser NVARCHAR(220),\r\n Bvtes INTEGER.\r\n FOREIGN KEY (Albumid) REFERENCES \"albums\" (Albumid) \r\n\t\tON DELETE NO ACTION ON 0.2) NOT NULL.\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\t0N DELETE NO ACTION UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON D ON UPDATE NO ACTION.\r\n AlbumId INTEGER PRIMARY KEY ELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\CREATE TABLE \"albums\"\r\n(\r\n AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n EIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE \"playlist track\"\r\n(\r\n PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT N CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) ULL.\r\n REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Addi tional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provide d context is sufficient, please generate a valid SQL guery without any explanations for the guestion. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular co lumn, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the qu ery with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-0

1'\nGROUP BY i.InvoiceDate"}, {"role": "user", "content": " \n List all invoices with a total exceeding \$10:\n"}, {"role": "assistant", "content": "SELECT \*\nFROM \"invoices\" \nWHERE (Total > 10)\nGROUP BY Invo iceId"}, {"role": "user", "content": " \n List all invoices with a total exceeding \$10:\n"}, {"role": "assistant", "content": "SELECT \* \nFROM \"invoices\" \nWHERE Total > 10.00"}, {"role": "user", "content": Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT i.Bil lingCountry, COUNT(\*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "content": " \n Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "S ELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM \"customers\" customer\nJOIN \"invoi ces\" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country"}, {"role": "user", "content": Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerI d = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n Find the top 5 customers w ho spent the most money overall. \n \n Hint: order total can be found on invoices table, calculation n using invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.Customer Id, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nG ROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n ustomer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "co ntent": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i 0 N c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.Customer Id\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n Hint: album quantity is foun d in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (acro ss all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS To talAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Custome rId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoi ces\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n Find all invoices since 2010 and the total amount invoiced:\n"}] Ollama Response: {'model': 'phi3:latest', 'created at': '2024-06-14T02:50:00.137712Z', 'message': {'role': 'assistant', 'con tent': 'SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount\nFROM "invoices" i\nWHERE strftime(\'%Y\', i.Invo iceDate) >= \'2010\'\nGROUP BY i.InvoiceId\nORDER BY InvoiceAmount DESC;'}, 'done reason': 'stop', 'done': True, 'total duration': 72733332693, 'load duration': 712770, 'prompt eval count': 2018, 'prompt eval durat ion': 60910143000, 'eval count': 64, 'eval duration': 10840683000} SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount FROM "invoices" i WHERE strftime('%Y', i.InvoiceDate) >= '2010' GROUP BY i.InvoiceId ORDER BY InvoiceAmount DESC: Output from LLM: SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount FROM "invoices" i WHERE strftime('%Y', i.InvoiceDate) >= '2010'

```
GROUP BY i.InvoiceId
ORDER BY InvoiceAmount DESC:
Extracted SQL: SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount
FROM "invoices" i
WHERE strftime('%Y', i.InvoiceDate) >= '2010'
GROUP BY i.InvoiceId
ORDER BY InvoiceAmount DESC
SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount
FROM "invoices" i
WHERE strftime('%Y', i.InvoiceDate) >= '2010'
GROUP BY i.InvoiceId
ORDER BY InvoiceAmount DESC
     InvoiceId InvoiceAmount
0
                        25.86
           404
1
           299
                        23.86
2
                        21.86
            96
3
           194
                        21.86
4
            89
                        18.86
           . . .
                          . . .
324
           377
                         0.99
325
           384
                         0.99
326
           391
                         0.99
327
           398
                         0.99
328
           405
                         0.99
[329 rows x 2 columns]
Ollama parameters:
model=phi3:latest,
options={},
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '\n Find all invoices since 2010 and the total amount invoic
ed:\n'\nThe DataFrame was produced using this query: SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount\nF
ROM \"invoices\" i\nWHERE strftime('%Y', i.InvoiceDate) >= '2010'\nGROUP BY i.InvoiceId\nORDER BY InvoiceAm
ount DESC\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gi
                                                  float64\ndtype: object"}, {"role": "user", "content": "Ca
ves:\n InvoiceId
                          int64\nInvoiceAmount
n you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas
dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Pyt
hon code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'phi3:latest', 'created at': '2024-06-14T02:50:17.824092557Z', 'message': {'role': 'assistant',
```

'content': "import plotly.express as px\n\nfig = px.bar(df, x='InvoiceId', y='InvoiceAmount')\nfig.show()\n \nif df.shape[0] == 1:\n fig = px.pie(df, values='InvoiceAmount', names=['Total'])\n fig.show()"}, 'd one\_reason': 'stop', 'done': True, 'total\_duration': 17659710113, 'load\_duration': 41233884, 'prompt\_eval\_c ount': 235, 'prompt\_eval\_duration': 6683685000, 'eval\_count': 78, 'eval\_duration': 10885899000}



```
Out[26]: ('SELECT i.InvoiceId, SUM(i.Total) AS InvoiceAmount\nFROM "invoices" i\nWHERE strftime(\'%Y\', i.InvoiceDa
          te) >= \'2010\'\nGROUP BY i.InvoiceId\nORDER BY InvoiceAmount DESC',
                InvoiceId InvoiceAmount
           0
                      404
                                   25.86
           1
                      299
                                   23.86
           2
                                   21.86
                       96
           3
                                   21.86
                      194
           4
                       89
                                   18.86
                      . . .
                                     . . .
           324
                      377
                                    0.99
           325
                      384
                                    0.99
           326
                      391
                                    0.99
           327
                      398
                                    0.99
           328
                                    0.99
                      405
           [329 rows x 2 columns],
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'InvoiceId=%{x}<br>InvoiceAmount=%{y}<extra></extra>',
                         'legendgroup': '',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                         'name': '',
                         'offsetgroup': '',
                         'orientation': 'v'.
                         'showlegend': False,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array([404, 299, 96, ..., 391, 398, 405]),
                         'xaxis': 'x',
                         'y': array([25.86, 23.86, 21.86, ..., 0.99, 0.99, 0.99]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceId'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'InvoiceAmount'}}}
          }))
         question = """
In [27]:
             List all employees and their reporting manager's name (if any):
         0.0000
```

vn.ask(question=question)

Number of requested results 10 is greater than number of elements in index 1, updating n\_results = 1

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE INDEX IFK EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE TA EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n BLE "employees"\r\n(\r\n LastName NVARCHAR (20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30).\r\n ReportsTo INTEGE BirthDate DATETIME,\r\n  $R.\r\n$ HireDate DATETIME,\r\n Address NVARCHAR(70),\r\n City NVARCHAR(4  $0), r\n$ State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NV  $ARCHAR(24), \r\n$ Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "customers"\r CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n  $n(\r\n$ FirstName NVARCHAR(40) NOT NUL LastName NVARCHAR(20) NOT NULL,\r\n L, r nCompany NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n State NVARCHAR(40),\r\n PostalCode NVARCHAR(1 City NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n  $0), \r\n$ Email NVARCHAR(60) NOT NULL,\r\n FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n\t\t0N DELETE NO A RepId INTEGER.\r\n CTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON "customers" (SupportRepId)\n\nCR InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n EATE TABLE "invoices"\r\n(\r\n CustomerId I NTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n BillingAddress NVARCHAR(70),\r\n Billing City NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCountry NVARCHAR(40),\r\n BillingPost FOREIGN KEY (CustomerId) REFERENCES "cust alCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n omers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice items"\r InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r  $\n(\r\n$ TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Ouantity INTEGER NOT NUL FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE N L, r nFOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "artists"\r\n(\r\n ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMEN T NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NU Composer NVARCHAR(220),\r\n LL,\r\n GenreId INTEGER,\r\n Milliseconds INTEGER NOT NULL.\r\n tes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" (Al bumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (G enreId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Title NVARCHAR(160) NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\t0N DELETE NO stId INTEGER NOT NULL,\r\n ACTION ON UPDATE NO ACTION\r\n\\nCREATE TABLE sglite stat1(tbl,idx,stat)\\n\\n===Additional Context \\n\nI n the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is suffici ent, please generate a valid SQL 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 particular column, please generat e an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment s aying intermediate sql \n3. If the provided context is insufficient, please explain why it can\'t be genera ted. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, p lease repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': " \n

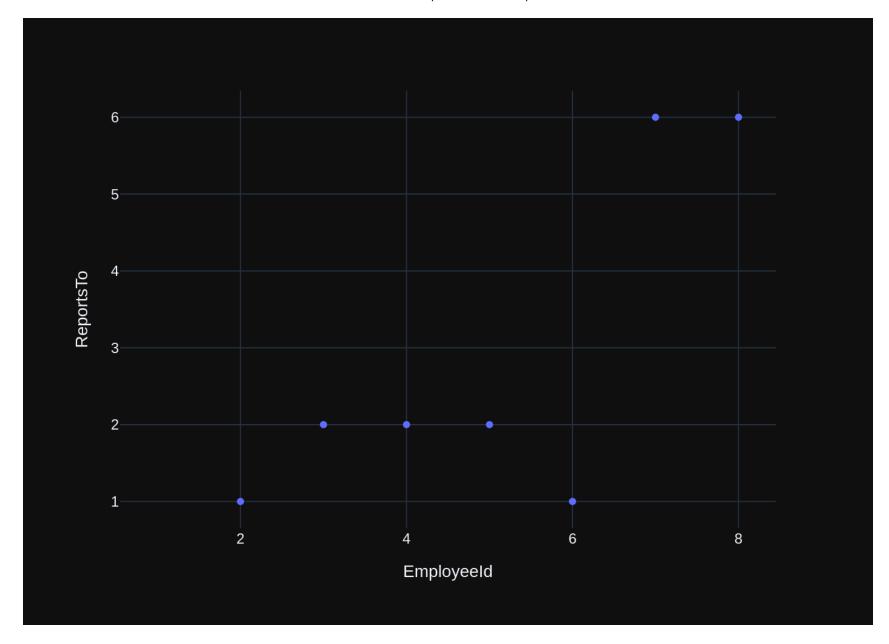
employees and their reporting manager's name (if any):\n"}, {'role': 'assistant', 'content': 'SELECT e.Firs tName, e.LastName, mt.FirstName AS ManagerFirstName, mt.LastName AS ManagerLastName\nFROM "employees" e\nLE FT JOIN "employees" mt ON e.ReportsTo = mt.EmployeeId'}, {'role': 'user', 'content': ' \n Hint: order total can be found on invoices tabl 5 customers who spent the most money overall. \n \n e, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELE CT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Cust omerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n ind the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT (i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': 'what are the top 5 co untries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(\*) AS TotalC ustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', List all customers from Canada and their email addresses:\n'}, {'role': 'assistant', 'content': 'SELECT c.Email, c.Country\nFROM "customers" c\nWHERE c.Country = \'Canada\''}, {'role': 'user', Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'conten t': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJ0IN "invoices" i ON c.C ustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n Find all invoices s ince 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM (i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDat e'}, {'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 InvoiceAmount\nFROM "invoices" i\nWHE RE strftime(\'%Y\', i.InvoiceDate) >= \'2010\'\nGROUP BY i.InvoiceId\nORDER BY InvoiceAmount DESC'}, {'rol e': 'user', 'content': ' \n Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nF ROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.Cust omerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Cust omerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': " \n List all employees and their reportin g manager's name (if any):\n"}] Ollama parameters: model=phi3:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE INDEX IFK EmployeeReportsTo ON \"employees\" (ReportsTo)\n\nCREATE EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n TABLE \"employees\"\r\n(\r\n LastName NVAR CHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL.\r\n Title NVARCHAR(30),\r\n ReportsTo IN Address NVARCHAR(70),\r\n City NVARCH TEGER,\r\n BirthDate DATETIME.\r\n HireDate DATETIME.\r\n State NVARCHAR(40),\r\n  $AR(40), \r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phon

FOREIGN KEY (ReportsTo) REFERENC e NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n ES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"custom ers\"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(1 Fax NVARCHAR(24),\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n Support RepId INTEGER.\r\n FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n \nCREATE TABLE \"invoices\"\r\n(\r\n Custom erId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n BillingAddress NVARCHAR(70),\r\n illingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCountry NVARCHAR(40),\r\n Billi ngPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice i InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"artists\"\r\n(\r\n ArtistId INTEGER PRIMARY KEY AUTOINCREMENT Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INT GenreId INTEGER,\r\n EGER NOT NULL,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NUL L.\r\n Bvtes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENC ES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) R EFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABL AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n E \"albums\"\r\n(\r\n Title NVARCHAR(160) NO ArtistId INTEGER NOT NULL.\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite stat1(tbl,idx,stat)\n\n\n===Add itional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provid ed context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular co lumn, please generate an intermediate SQL guery to find the distinct strings in that column. Prepend the qu ery with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": List all employees and their reporting manager's name (if any):\n"}, {"role": "assistant", "conten t": "SELECT e.FirstName, e.LastName, mt.FirstName AS ManagerFirstName, mt.LastName AS ManagerLastName\nFROM \"employees\" e\nLEFT JOIN \"employees\" mt ON e.ReportsTo = mt.EmployeeId"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most money overall, \n \n Hint: order total can be fo und on invoices table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistan t", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "use

r", "content": " \n Find the customer with the most invoices \n"}, {"role": "assistant", "content": "S ELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.Cus tomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "co ntent": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(\*) AS TotalCustomers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DE SC\nLIMIT 5"}, {"role": "user", "content": " \n List all customers from Canada and their email address es:\n"}, {"role": "assistant", "content": "SELECT c.Email, c.Country\nFROM \"customers\" c\nWHERE c.Country = 'Canada'"}, {"role": "user", "content": " \n Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customer s\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "conte nt": " \n Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "conte nt": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"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 InvoiceAm ount\nFROM \"invoices\" i\nWHERE strftime('%Y', i.InvoiceDate) >= '2010'\nGROUP BY i.InvoiceId\nORDER BY In voiceAmount DESC"}, {"role": "user", "content": "\n Find the customer who bought the most albums in t otal quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii. TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"i nvoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n Get the average invoice total for each customer:\n"}, {"role": "a ssistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": "\n List all employees and their reporting manager's name (if any):\n"}] Ollama Response: {'model': 'phi3:latest', 'created at': '2024-06-14T02:51:28.465161491Z', 'message': {'role': 'assistant', 'content': 'SELECT e1.\*, CONCAT(e2.FirstName, \' \', e2.LastName) AS ManagerName\nFROM "employees" e1\nLEFT JOIN "employees" e2 ON e1.ReportsTo = e2.EmployeeId\nORDER BY e1.EmployeeId;'}, 'done reason': 'stop', 'don e': True, 'total duration': 70526310908, 'load duration': 784876, 'prompt eval count': 2001, 'prompt eval d uration': 59372945000, 'eval count': 66, 'eval duration': 10341982000} SELECT e1.\*, CONCAT(e2.FirstName, ' ', e2.LastName) AS ManagerName FROM "employees" e1 LEFT JOIN "employees" e2 ON e1.ReportsTo = e2.EmployeeId ORDER BY e1.EmployeeId: Output from LLM: SELECT e1.\*, CONCAT(e2.FirstName, ' ', e2.LastName) AS ManagerName FROM "employees" e1 LEFT JOIN "employees" e2 ON e1.ReportsTo = e2.EmployeeId ORDER BY el.EmployeeId: Extracted SQL: SELECT e1.\*, CONCAT(e2.FirstName, ' ', e2.LastName) AS ManagerName FROM "employees" el LEFT JOIN "employees" e2 ON e1.ReportsTo = e2.EmployeeId ORDER BY e1.EmployeeId SELECT e1.\*, CONCAT(e2.FirstName, ' ', e2.LastName) AS ManagerName

```
FROM "employees" el
LEFT JOIN "employees" e2 ON e1.ReportsTo = e2.EmployeeId
ORDER BY el.EmployeeId
   EmployeeId LastName FirstName
                                                 Title ReportsTo \
0
            1
                  Adams
                           Andrew
                                       General Manager
                                                              NaN
1
            2
                Edwards
                            Nancy
                                                              1.0
                                         Sales Manager
2
                Peacock
            3
                                                              2.0
                             Jane Sales Support Agent
3
            4
                        Margaret Sales Support Agent
                   Park
                                                              2.0
                            Steve Sales Support Agent
4
                                                              2.0
                Johnson
5
              Mitchell
                          Michael
                                            IT Manager
                                                              1.0
6
            7
                   Kina
                                              IT Staff
                                                              6.0
                           Robert
7
              Callahan
                            Laura
                                              IT Staff
                                                              6.0
             BirthDate
                                   HireDate
                                                                 Address \
0 1962-02-18 00:00:00 2002-08-14 00:00:00
                                                     11120 Jasper Ave NW
                        2002-05-01 00:00:00
                                                            825 8 Ave SW
  1958-12-08 00:00:00
2 1973-08-29 00:00:00 2002-04-01 00:00:00
                                                           1111 6 Ave SW
   1947-09-19 00:00:00 2003-05-03 00:00:00
                                                        683 10 Street SW
4 1965-03-03 00:00:00 2003-10-17 00:00:00
                                                            7727B 41 Ave
5 1973-07-01 00:00:00 2003-10-17 00:00:00
                                                    5827 Bowness Road NW
  1970-05-29 00:00:00 2004-01-02 00:00:00
                                            590 Columbia Boulevard West
7 1968-01-09 00:00:00 2004-03-04 00:00:00
                                                             923 7 ST NW
         City State Country PostalCode
                                                                         Fax \
                                                    Phone
                               T5K 2N1 +1 (780) 428-9482 +1 (780) 428-3457
0
     Edmonton
                 AB Canada
                     Canada
1
      Calgary
                               T2P 2T3
                                        +1 (403) 262-3443 +1 (403) 262-3322
2
                     Canada
      Calgary
                 AB
                               T2P 5M5
                                       +1 (403) 262-3443 +1 (403) 262-6712
3
                     Canada
                               T2P 5G3 +1 (403) 263-4423 +1 (403) 263-4289
      Calgary
                 AB
4
                               T3B 1Y7
      Calgary
                 AB
                     Canada
                                        1 (780) 836-9987
                                                            1 (780) 836-9543
5
      Calgary
                 AB
                     Canada
                               T3B 0C5
                                       +1 (403) 246-9887 +1 (403) 246-9899
                     Canada
  Lethbridge
                 AB
                               T1K 5N8
                                       +1 (403) 456-9986 +1 (403) 456-8485
                     Canada
  Lethbridge
                 AB
                               T1H 1Y8 +1 (403) 467-3351 +1 (403) 467-8772
                      Email
                                  ManagerName
0
     andrew@chinookcorp.com
1
                                 Andrew Adams
      nancy@chinookcorp.com
2
       jane@chinookcorp.com
                                Nancy Edwards
  margaret@chinookcorp.com
                                Nancy Edwards
4
                                Nancy Edwards
      steve@chinookcorp.com
5
    michael@chinookcorp.com
                                 Andrew Adams
6
     robert@chinookcorp.com Michael Mitchell
7
      laura@chinookcorp.com Michael Mitchell
```

```
Ollama parameters:
model=phi3:latest,
options={}.
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '\n List all employees and their reporting manager's name (i
f any):\n'\nThe DataFrame was produced using this query: SELECT e1.*, CONCAT(e2.FirstName, ' ', e2.LastNa
me) AS ManagerName\nFROM \"employees\" e1\nLEFT JOIN \"employees\" e2 ON e1.ReportsTo = e2.EmployeeId\nORDE
R BY e1.EmployeeId\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.
dtypes gives:\n EmployeeId
                                 int64\nLastName
                                                        object\nFirstName
                                                                                obiect\nTitle
                                                                                                        obi
                    float64\nBirthDate
                                             object\nHireDate
ect\nReportsTo
                                                                     obiect\nAddress
                                                                                             object\nCity
                                               object\nPostalCode
                                                                       object\nPhone
                        object\nCountry
object\nState
                                                                                                obiect\nFax
                                               object\ndtype: object"}, {"role": "user", "content": "Can y
obiect\nEmail
                        object\nManagerName
ou generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas da
taframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Pytho
n code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'phi3:latest', 'created at': '2024-06-14T02:51:54.197047431Z', 'message': {'role': 'assistant',
'content': "import plotly.graph objects as go\n\nfig = go.Figure()\n\nfor column in df.columns:\n
                                fig.add trace(go.Bar(x=df['EmployeeId'], y=df[column]))\n
[columnl.nunique() > 1:\n
fig.add indicator(value=df[column][0])\n\nfig.update layout(title='Employees and Their Managers')\nfig.show
()"}, 'done reason': 'stop', 'done': True, 'total duration': 25704966494, 'load duration': 636586, 'prompt
eval count': 296, 'prompt eval duration': 8603090000, 'eval count': 111, 'eval duration': 17011493000}
```



```
('SELECT e1.*, CONCAT(e2.FirstName, \' \', e2.LastName) AS ManagerName\nFROM "employees" e1\nLEFT JOIN "em
ployees" e2 ON e1.ReportsTo = e2.EmployeeId\nORDER BY e1.EmployeeId',
    EmployeeId LastName FirstName
                                                  Title ReportsTo \
 0
                                        General Manager
             1
                   Adams
                            Andrew
                                                               NaN
 1
             2
                 Edwards
                                          Sales Manager
                                                               1.0
                             Nancy
 2
             3
                 Peacock
                                                               2.0
                              Jane Sales Support Agent
 3
                    Park Margaret Sales Support Agent
             4
                                                               2.0
 4
                             Steve Sales Support Agent
             5
                 Johnson
                                                               2.0
 5
                Mitchell
                          Michael
                                             IT Manager
                                                               1.0
 6
             7
                                                               6.0
                    King
                            Robert
                                               IT Staff
 7
                                               IT Staff
                                                               6.0
                Callahan
                             Laura
                                                                  Address \
                                    HireDate
              BirthDate
                                                      11120 Jasper Ave NW
   1962-02-18 00:00:00 2002-08-14 00:00:00
 1 1958-12-08 00:00:00
                         2002-05-01 00:00:00
                                                             825 8 Ave SW
 2 1973-08-29 00:00:00 2002-04-01 00:00:00
                                                            1111 6 Ave SW
 3 1947-09-19 00:00:00 2003-05-03 00:00:00
                                                         683 10 Street SW
   1965-03-03 00:00:00 2003-10-17 00:00:00
                                                             7727B 41 Ave
 5 1973-07-01 00:00:00 2003-10-17 00:00:00
                                                     5827 Bowness Road NW
   1970-05-29 00:00:00 2004-01-02 00:00:00
                                              590 Columbia Boulevard West
 7 1968-01-09 00:00:00 2004-03-04 00:00:00
                                                              923 7 ST NW
          City State Country PostalCode
                                                                          Fax \
                                                     Phone
 0
                  AB Canada
                                T5K 2N1 +1 (780) 428-9482 +1 (780) 428-3457
      Edmonton
 1
                  AB Canada
                                T2P 2T3 +1 (403) 262-3443
                                                            +1 (403) 262-3322
       Calgary
 2
                                T2P 5M5 +1 (403) 262-3443 +1 (403) 262-6712
       Calgary
                  AB Canada
 3
       Calgary
                  AB Canada
                                T2P 5G3 +1 (403) 263-4423 +1 (403) 263-4289
 4
                                          1 (780) 836-9987
       Calgary
                  AB Canada
                                T3B 1Y7
                                                             1 (780) 836-9543
 5
                                T3B 0C5 +1 (403) 246-9887
                  AB Canada
       Calgary
                                                            +1 (403) 246-9899
 6
   Lethbridge
                     Canada
                                T1K 5N8 +1 (403) 456-9986
                  AB
                                                           +1 (403) 456-8485
    Lethbridge
                  AB Canada
                                T1H 1Y8 +1 (403) 467-3351 +1 (403) 467-8772
                                   ManagerName
                       Email
 0
      andrew@chinookcorp.com
 1
                                  Andrew Adams
       nancy@chinookcorp.com
 2
                                 Nancy Edwards
        jane@chinookcorp.com
 3
   margaret@chinookcorp.com
                                 Nancy Edwards
 4
       steve@chinookcorp.com
                                 Nancy Edwards
 5
                                  Andrew Adams
     michael@chinookcorp.com
 6
                              Michael Mitchell
      robert@chinookcorp.com
 7
       laura@chinookcorp.com
                              Michael Mitchell ,
 Figure({
```

```
'data': [{'hovertemplate': 'EmployeeId=%{x}<br/>br>ReportsTo=%{y}<extra></extra>',
                         'legendgroup': '',
                         'marker': {'color': '#636efa', 'symbol': 'circle'},
                         'mode': 'markers',
                         'name': '',
                         'orientation': 'v',
                         'showlegend': False,
                         'type': 'scatter',
                         'x': array([1, 2, 3, 4, 5, 6, 7, 8]),
                         'xaxis': 'x',
                         'y': array([nan, 1., 2., 2., 2., 1., 6., 6.]),
                         'yaxis': 'y'}],
               'layout': {'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'EmployeeId'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'ReportsTo'}}}
          }))
         question = """
In [28]:
             Get the average invoice total for each customer:
         vn.ask(question=question)
        Number of requested results 10 is greater than number of elements in index 1, updating n results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR EMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n BillingA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n **FOREIG** rv NVARCHAR(40).\r\n N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON "invoice items" (InvoiceId)\n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KE InvoiceId INTEGER NOT NULL,\r\n Y AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES Price NUMERIC(10,2) NOT NULL,\r\n "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERE NCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLin eTrackId ON "invoice items" (TrackId)\n\nCREATE TABLE sqlite stat1(tbl,idx,stat)\n\nCREATE INDEX IFK Custom erSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRI MARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL.\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n  $CHAR(40), \r\n$ PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n  $\times$  NVARCHAR(24),\r\n SupportRepId INTEGER,\r\n FOREIGN KEY (Suppo rtRepId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE TABLE "employees"\r\n(\r\n LastName NVARCHAR(20) NOT NULL,\r\n NTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER.\r\n R(20) NOT NULL.\r\n BirthDate DATETIME.\r\n Hir eDate DATETIME,\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Co untry NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\t0N DELETE NO Email NVARCHAR(60),\r\n ACTION ON UPDATE NO ACTION\r\n\\n===Additional Context \n\nIn the SQLite database invoice means order\n \n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query with out any explanations for the question. \n2. If the provided context is almost sufficient but requires knowl edge of a specific string in a particular column, please generate an intermediate SQL guery to find the dis tinct strings in that column. Prepend the guery with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table (s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was gi ven before. \n'}, {'role': 'user', 'content': ' \n Get the average invoice total for each custome r:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "c ustomers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'c Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.Custo merId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n Find the top 5 custome rs who spent the most money overall, \n \n Hint: order total can be found on invoices table, calcul ation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.Cust omerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nG ROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n Find the to

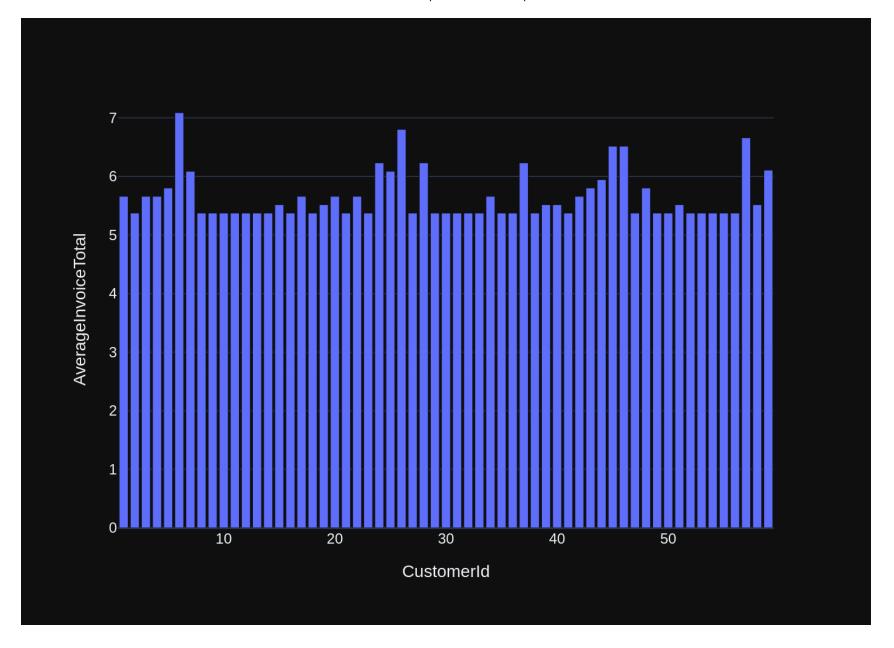
tal number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT Customer.Country, COUNT(i nvoice.InvoiceId) AS InvoiceCount\nFROM "customers" customer\nJOIN "invoices" invoice ON customer.CustomerI d = invoice.CustomerId\nGROUP BY Country'}, {'role': 'user', 'content': ' \n Find the customer with th e most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInv oices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER B Y TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': '\n Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS TotalAm ount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}. {'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 InvoiceAmount\nFROM "invoices" i\nWHERE strftime(\'%Y\', i. InvoiceDate) >= \'2010\'\nGROUP BY i.InvoiceId\nORDER BY InvoiceAmount DESC'}, {'role': 'user', 'content': Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT i.Bil lingCountry, COUNT(\*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'c ontent': '\n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELE CT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJ0IN "invoice items" ii 0N i.InvoiceId = ii.InvoiceId\nGR0UP BY c.CustomerId\nORDER BY TotalA lbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer who bought the most albums i n total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT (ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "in voice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Get the average invoice total for each customer:\n'}] Ollama parameters: model=phi3:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n Billin gAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillinaCou BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n ntry NVARCHAR(40),\r\n IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK InvoiceLineInv oiceId ON \"invoice items\" (InvoiceId)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r UnitPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Tr ackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE sqlite stat1(tbl,idx,stat)\n\nCREATE

INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"customers\"\r\n(\r\n Cu stomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastNa me NVARCHAR(20) NOT NULL.\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(4 State NVARCHAR(40),\r\n  $0), \r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NV Email NVARCHAR(60) NOT NULL,\r\n  $ARCHAR(24).\r\n$ Fax NVARCHAR(24),\r\n SupportRepId INTEGER.\r\n FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO A CTION\r\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON \"employees\" (ReportsTo)\n\nCREATE TABLE \"employees EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n  $\"\r\n(\r\n$ LastName NVARCHAR(20) NOT NUL FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER.\r\n L.\r\n thDate DATETIME.\r\n HireDate DATETIME,\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n St ate NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r \n Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQL ite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, plea se generate a valid SQL query without any explanations for the question. \n2. If the provided context is al most sufficient but requires knowledge of a specific string in a particular column, please generate an inte rmediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying int ermediate sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repe at the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n Get the average in voice total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.C ustomerId"}, {"role": "user", "content": " \n Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customer s\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "conte nt": " \n Find the top 5 customers who spent the most money overall, \n Hint: order total c an be found on invoices table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invo ices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"rol e": "user", "content": " \n Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM \"customers\" customer \nJOIN \"invoices\" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country"}, {"role": "use r", "content": " \n Find the customer with the most invoices \n"}, {"role": "assistant", "content": "S ELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.Cus tomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "co ntent": " \n Find all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "co ntent": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate >= '20 10-01-01'\nGROUP BY i.InvoiceDate"}, {"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 Invoi ceAmount\nFROM \"invoices\" i\nWHERE strftime('%Y', i.InvoiceDate) >= '2010'\nGROUP BY i.InvoiceId\nORDER B Y InvoiceAmount DESC"}, {"role": "user", "content": " \n Find the total number of invoices per countr y:\n"}, {"role": "assistant", "content": "SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\nFROM \"invoic

```
es\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "content": " \n Hint: album quantity is found in
                          Find the top 5 customers who bought the most albums in total quantity (across al
invoice items, \n \n
l invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\n
FROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.In
voiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "cont
               Find the customer who bought the most albums in total quantity (across all invoices): \n"},
ent": " \n
{"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\"
c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.Invoic
eId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": "\n
the average invoice total for each customer:\n"}]
Ollama Response:
{'model': 'phi3:latest', 'created at': '2024-06-14T02:53:05.257778321Z', 'message': {'role': 'assistant',
'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i
ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY AverageInvoiceTotal DESC'}, 'done reason':
'stop', 'done': True, 'total duration': 70962390336, 'load duration': 781587, 'prompt eval count': 1973, 'p
rompt eval duration': 60635317000, 'eval count': 62, 'eval duration': 9645099000}
SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY AverageInvoiceTotal DESC
SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY AverageInvoiceTotal DESC
   CustomerId AverageInvoiceTotal
0
             6
                           7.088571
1
                           6.802857
            26
2
            57
                           6.660000
3
            45
                           6.517143
4
            46
                           6.517143
5
            24
                           6.231429
6
            28
                           6.231429
7
            37
                           6.231429
8
            59
                           6.106667
9
            7
                           6.088571
10
            25
                           6.088571
11
            44
                           5.945714
12
             5
                           5.802857
13
            43
                           5.802857
14
            48
                           5.802857
```

17     4     5.6600       18     17     5.6600       19     20     5.6600       20     5.6600	000 000 000 000 000 43
19 20 5.6600	000 000 000 000 43
20 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6	000 000 .43
20 22 5.6600	000 .43
21 34 5.6600	.43
22 42 5.6600	
23 15 5.5171	43
24 19 5.5171	
25 39 5.5171	
26 40 5.5171 27 51 5.5171	
28 58 5.5171	
29 2 5.3742	
30 8 5.3742	
31 9 5.3742	
32 10 5.3742	86
33 11 5.3742	86
34 12 5.3742	
35 13 5.3742	
36 14 5.3742	
37 16 5.3742	
38 18 5.3742 39 21 5.3742	
39 21 5.3742 40 23 5.3742	
41 27 5.3742	
42 29 5.3742	
43 30 5.3742	
44 31 5.3742	
45 32 5.3742	86
46 33 5.3742	
47 35 5.3742	
48 36 5.3742	
49 38 5.3742	
50 41 5.3742 51 47 5.3742	
51 47 5.3742 52 49 5.3742	
53 50 5.3742	
54 52 5.3742	
55 53 5.3742	
56 54 5.3742	

```
57 55 5.374286
58 56 5.374286
Ollama parameters:
model=phi3:latest,
options={},
keep_alive=None
Prompt Content:
[{"role": "system", "content": "The follows
that answers the question the user asked:
```



Out[28]: ('SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i ON c.Cus tomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY AverageInvoiceTotal DESC',

LO	meria = 1.Cust	comeria/NGROUP By c.Cu
	CustomerId	AverageInvoiceTotal
0	6	7.088571
1	. 26	6.802857
2	57	6.660000
3	45	6.517143
4	46	6.517143
5	24	6.231429
6	28	6.231429
7	37	6.231429
8	59	6.106667
9	7	6.088571
1	.0 25	6.088571
1	1 44	5.945714
1	2 5	5.802857
1	.3 43	5.802857
1	4 48	5.802857
1	5 1	5.660000
1	.6 3	5.660000
1	.7 4	5.660000
1	.8 17	5.660000
1	.9 20	5.660000
2	0 22	5.660000
	1 34	5.660000
2	2 42	5.660000
	3 15	5.517143
	4 19	5.517143
2	5 39	5.517143
2	6 40	5.517143
	7 51	5.517143
	8 58	5.517143
	9 2	5.374286
	0 8	5.374286
	1 9	5.374286
	2 10	5.374286
	3 11	5.374286
	4 12	5.374286
	5 13	5.374286
	6 14	5.374286
	7 16	5.374286
3	8 18	5.374286

```
21
39
                           5.374286
            23
40
                           5.374286
            27
41
                           5.374286
42
            29
                           5.374286
43
            30
                           5.374286
44
            31
                           5.374286
45
            32
                           5.374286
46
            33
                           5.374286
            35
47
                           5.374286
48
            36
                           5.374286
            38
49
                           5.374286
50
            41
                           5.374286
                           5.374286
51
            47
52
            49
                           5.374286
53
            50
                           5.374286
54
            52
                           5.374286
55
            53
                           5.374286
            54
56
                           5.374286
57
            55
                           5.374286
58
            56
                           5.374286,
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'CustomerId=%{x}<br>AverageInvoiceTotal=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'v',
              'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
              'x': array([ 6, 26, 57, 45, 46, 24, 28, 37, 59, 7, 25, 44, 5, 43, 48, 1, 3, 4,
                          17, 20, 22, 34, 42, 15, 19, 39, 40, 51, 58, 2, 8, 9, 10, 11, 12, 13,
                          14, 16, 18, 21, 23, 27, 29, 30, 31, 32, 33, 35, 36, 38, 41, 47, 49, 50,
                          52, 53, 54, 55, 56]),
              'xaxis': 'x'.
                                                            , 6.51714286, 6.51714286, 6.23142857,
              'y': array([7.08857143, 6.80285714, 6.66
                          6.23142857, 6.23142857, 6.10666667, 6.08857143, 6.08857143, 5.94571429,
                                                                          , 5.66
                                                                                      , 5.66
                          5.80285714, 5.80285714, 5.80285714, 5.66
                          5.66
                                     . 5.66
                                                 . 5.66
                                                              , 5.66
                                                                          , 5.66
                                                                                       , 5.51714286,
                          5.51714286, 5.51714286, 5.51714286, 5.51714286, 5.51714286, 5.37428571,
                          5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571,
```

```
5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571,
                                   5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571,
                                   5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571,
                                   5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571]),
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                        'legend': {'tracegroupgap': 0},
                        'margin': {'t': 60},
                        'template': '...',
                        'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                         'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'AverageInvoiceTotal'}}}
          }))
        question = """
In [29]:
            Find the top 5 most expensive tracks (based on unit price):
         0.00
         vn.ask(question=question)
       Number of requested results 10 is greater than number of elements in index 1, updating n results = 1
```

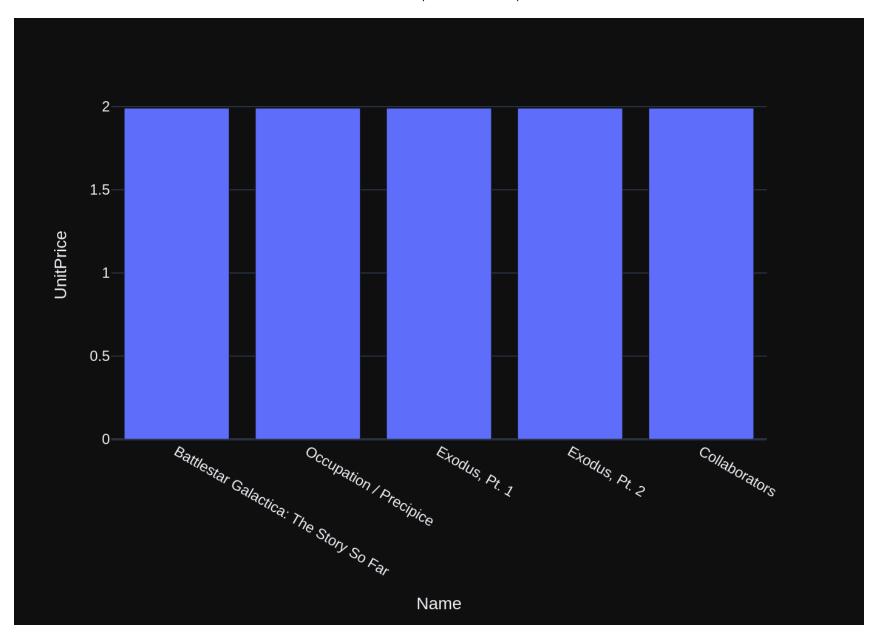
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMEN AlbumId INTEGER.\r\n T NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n MediaTypeId INTEGER NOT NU LL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n tes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" (Al bumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (G enreId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDEX IFK TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK PlaylistTr ackTrackId ON "playlist track" (TrackId)\n\nCREATE INDEX IFK InvoiceLineTrackId ON "invoice items" (TrackI d)\n\nCREATE INDEX IFK TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n d INTEGER NOT NULL.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n REIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r \n)\n\nCREATE TABLE "playlist track"\r\n(\r\n PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlavlistI d) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDE AlbumId INTEGER PRIMARY KEY X IFK AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "albums"\r\n(\r\n AUTOINCREMENT NOT NULL,\r\n ArtistId INTEGER NOT NULL.\r\n Title NVARCHAR(160) NOT NULL,\r\n EIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n \n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL guery without any explanations for the gues tion. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a pa rticular column, please generate an intermediate SQL query to find the distinct strings in that column. Pre pend the query with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the guestion has be en asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'use r', 'content': '\n Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistan t', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n There are 3 tables: artists, albums and tracks, where albums and a rtists are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most p opular artists based on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT a.Name, COUNT(t. TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought th e most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.Cust omerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Custome rId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DES C\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n

Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "in voice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, Hint: album quantity is found in invoice items, \n \n {'role': 'user', 'content': ' \n op 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistan t', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoi ce items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money overall, \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessa ry \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customer s" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLI MIT 5'}, {'role': 'user', 'content': ' \n Find the customer who bought the most albums in total quanti ty (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'use r', 'content': '\n Get all playlists containing at least 10 tracks and the total duration of those tr acks:\n'}, {'role': 'assistant', 'content': 'SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Millisecon ds) AS TotalDuration\nFROM "playlist track" pt\nJOIN "playlists" p ON pt.PlaylistId = p.PlaylistId\nJOIN "t racks" t ON pt.TrackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10'}, {'rol e': 'user', 'content': ' \n Identify artists who have albums with tracks appearing in multiple genre s:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artists" a\nJ0 IN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM "genres" q2 \n JOIN "tracks" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\n GROUP BY a.ArtistId, a.Name'}, {'role': 'user', 'content': ' \n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name A S ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELE CT g2.GenreId\n FROM "genres" g2\n JOIN "tracks" t2 ON g2.GenreId = t2.GenreId\n GROUP BY g2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'role': 'user', 'conte nt': ' \n Find the top 5 most expensive tracks (based on unit price):\n'}] Ollama parameters: model=phi3:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo TrackId INTEGER PRIMARY KEY AUTOINCREM rmat instructions. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n AlbumId INTEGER.\r\n ENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n MediaTypeId INTEGER NOT Milliseconds INTEGER NOT NULL.\r\n NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres \" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES

\"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK Trac kAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDEX IFK TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX I FK PlaylistTrackTrackId ON \"playlist track\" (TrackId)\n\nCREATE INDEX IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE INDEX IFK TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE TABLE \"invoi InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER TrackId INTEGER NOT NULL,\r\n NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Ouantity INTEG ER NOT NULL.\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTIO N ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO AC TION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist track\"\r\n(\r\n PlaylistId INTEGER NOT NUL TrackId INTEGER NOT NULL,\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackI FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UP d), r nDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"albums AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\t0 N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice m eans order $\n$ ==Response Guidelines  $\n$ 1. If the provided context is sufficient, please generate a valid SQ L query without any explanations for the question. \n2. If the provided context is almost sufficient but re quires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the guery with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most re levant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n Find the top 5 most expensive tracks (ba sed on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tr acks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n There are 3 tables: art ists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by A Can you find the top 10 most popular artists based on the number of tracks\n"}, {"role": "assi stant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTr acks DESC\nLIMIT 10"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, Find the top 5 customers who bought the most albums in total quantity (across all invoice s):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"cu stomers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the m ost albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.Custome rId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.I nvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most al bums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, C OUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.I nvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n

```
Find the top 5 customers who spent the most money overall. \n
                                                                 \n
                                                                        Hint: order total can be found on
invoices table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "co
ntent": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJ0IN \"invoices\" i ON c.Cu
stomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "cont
               Find the customer who bought the most albums in total quantity (across all invoices): \n"},
{"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\"
c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.Invoic
eId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": "\n
all playlists containing at least 10 tracks and the total duration of those tracks:\n"}, {"role": "assistan
t", "content": "SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM
\"playlist track\" pt\nJOIN \"playlists\" p ON pt.PlaylistId = p.PlaylistId\nJOIN \"tracks\" t ON pt.TrackI
d = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10"}, {"role": "user", "content
              Identify artists who have albums with tracks appearing in multiple genres:\n\n\n"}, {"role":
"assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON
a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM \"genres\" q2\n J0IN \"tracks
\" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.Arti
stId, a.Name"}, {"role": "user", "content": " \n
                                                     Identify artists who have albums with tracks appearing
g in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nF
ROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.Genre
Id\n FROM \"genres\" g2\n JOIN \"tracks\" t2 ON g2.GenreId = t2.GenreId\n GROUP BY g2.GenreId\n HAVING
COUNT(g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName"}, {"role": "user", "content": "
      Find the top 5 most expensive tracks (based on unit price):\n"}]
Ollama Response:
{'model': 'phi3:latest', 'created at': '2024-06-14T02:54:28.34425122Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT Name, UnitPrice\nFROM "tracks"\nORDER BY UnitPrice DESC\nLIMIT 5'}, 'done reason': 'stop',
'done': True, 'total duration': 63715711054, 'load duration': 766944, 'prompt eval count': 1943, 'prompt ev
al duration': 59065991000, 'eval count': 23, 'eval duration': 3854785000}
SELECT Name, UnitPrice
FROM "tracks"
ORDER BY UnitPrice DESC
LIMIT 5
SELECT Name, UnitPrice
FROM "tracks"
ORDER BY UnitPrice DESC
LIMIT 5
                                     Name UnitPrice
O Battlestar Galactica: The Story So Far
                                                1.99
                  Occupation / Precipice
                                                1.99
1
2
                                               1.99
                            Exodus, Pt. 1
3
                            Exodus, Pt. 2
                                                1.99
                            Collaborators
                                                1.99
Ollama parameters:
```

```
model=phi3:latest.
options={},
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: ' \n Find the top 5 most expensive tracks (based on unit pric
e):\n'\nThe DataFrame was produced using this query: SELECT Name, UnitPrice\nFROM \"tracks\"\nORDER BY Un
itPrice DESC\nLIMIT 5\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning
                                                    float64\ndtype: object"}, {"role": "user", "content":
df.dtvpes gives:\n Name
                                obiect\nUnitPrice
"Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pan
das dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only
Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'phi3:latest', 'created at': '2024-06-14T02:54:47.307732912Z', 'message': {'role': 'assistant',
'content': "import plotly.express as px\n = px.bar(df, x='Name', y='UnitPrice')\n df.s
                  fig = px.pie(df, values='UnitPrice', names='Name')\n fig.show()"}, 'done reason': 'st
hape[0] == 1:\n
op', 'done': True, 'total duration': 18937826836, 'load duration': 43755308, 'prompt eval count': 185, 'pro
mpt eval duration': 5896385000, 'eval count': 73, 'eval duration': 12947692000}
```



```
Out[29]: ('SELECT Name, UnitPrice\nFROM "tracks"\nORDER BY UnitPrice DESC\nLIMIT 5',
                                                Name UnitPrice
             Battlestar Galactica: The Story So Far
                                                           1.99
                              Occupation / Precipice
           1
                                                           1.99
           2
                                       Exodus, Pt. 1
                                                           1.99
           3
                                       Exodus, Pt. 2
                                                           1.99
           4
                                       Collaborators
                                                           1.99,
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'Name=%{x}<br>UnitPrice=%{y}<extra></extra>',
                         'legendgroup': '',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                         'name': '',
                         'offsetgroup': '',
                         'orientation': 'v',
                         'showlegend': False,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array(['Battlestar Galactica: The Story So Far', 'Occupation / Precipice',
                                     'Exodus, Pt. 1', 'Exodus, Pt. 2', 'Collaborators'], dtype=object),
                         'xaxis': 'x'.
                         'y': array([1.99, 1.99, 1.99, 1.99, 1.99]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'UnitPrice'}}}
          }))
         question = """
In [30]:
             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
```

file:///home/gongai/projects/wgong/py4kids/lesson-18-ai/vanna/docs/ollama-phi3-chromadb-sglite-test-1.html

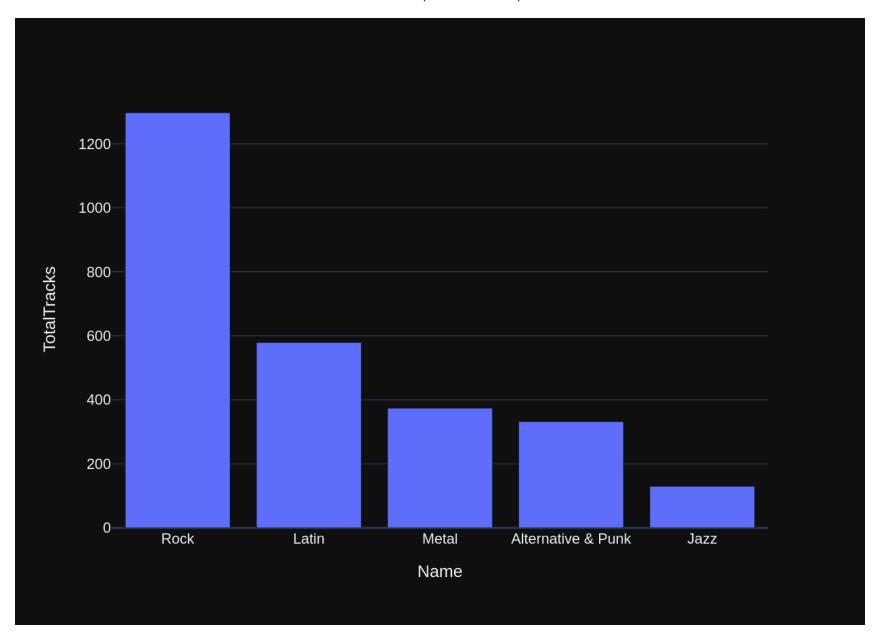
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMEN Name NVARCHAR(200) NOT NULL,\r\n T NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NU LL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL,\r\n tes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" (Al bumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (G enreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK TrackGenreId ON "tracks" (GenreId)\n\nCREATE TABLE "genres"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NU LL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON "playlist track" (TrackId)\n \nCREATE INDEX IFK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "playlists"\r\n(\r\n TEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK TrackMediaType Id ON "tracks" (MediaTypeId)\n\nCREATE TABLE "playlist track"\r\n(\r\n PlaylistId INTEGER NOT NULL,\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n TrackId INTEGER NOT NULL.\r\n REIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI ON\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n le NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "ar tists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\cREATE INDEX IFK AlbumArtistId ON "albums" (ArtistId)\ $n\n===Additional$  Context  $\n\nIn$  the SQLite database invoice means order $\n\n===Respons$ e Guidelines \n1. If the provided context is sufficient, please generate a valid SQL guery without any expl anations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a sp ecific string in a particular column, please generate an intermediate SQL query to find the distinct string s in that column. Prepend the query with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. I f the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n List all genres and the number of tracks in each genre:\n'}, {'r ole': 'assistant', 'content': 'SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" q\nJOIN "track s" t ON q.GenreId = t.GenreId\nGROUP BY q.Name'}, {'role': 'user', 'content': ' \n There are 3 tables: a rtists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'ass istant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': '\n Identify artists who have albums with tracks appear ing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName \nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreI d\n FROM "genres" q2\n JOIN "tracks" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT (q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name'}, {'role': 'user', 'content': '\n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.A rtistId IN (\n SELECT q2.GenreId\n FROM "genres" q2\n JOIN "tracks" t2 ON q2.GenreId = t2.GenreId\n GRO UP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'r

ole': 'user', 'content': ' \n Get all playlists containing at least 10 tracks and the total duration o f those tracks:\n'}, {'role': 'assistant', 'content': 'SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t. Milliseconds) AS TotalDuration\nFROM "playlist track" pt\nJOIN "playlists" p ON pt.PlaylistId = p.PlaylistI d\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 1 0'}, {'role': 'user', 'content': ' \n Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT Name, UnitPrice\nFROM "tracks"\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPr ice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, Find the top 5 customers who bought the most albums in total quantity (across all invoice s):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "cus tomers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.In voiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Titl e, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId'}, {'r ole': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'us List all genres and the number of tracks in each genre:\n'}] er', 'content': ' \n Ollama parameters: model=phi3:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREM ENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL,\r\n Bvtes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres \" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK Trac kGenreId ON \"tracks\" (GenreId)\n\nCREATE TABLE \"genres\"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINC Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON \"playlist trac REMENT NOT NULL,\r\n k\" (TrackId)\n\nCREATE INDEX IFK TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"playlists\"\r\n(\r PlavlistId 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 CONSTRAINT PK PlaylistTrack PRIMARY KEY (Playl TrackId INTEGER NOT NULL.\r\n INTEGER NOT NULL,\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO istId, TrackId),\r\n

ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOI NCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCR EATE INDEX IFK AlbumArtistId ON \"albums\" (ArtistId)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficien t but requires knowledge of a specific string in a particular column, please generate an intermediate SQL g uery 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.  $\n$ 4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer e xactly as it was given before. \n"}, {"role": "user", "content": " \n List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS TotalTracks \nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"role": "user", "conte There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistI d. albums and tracks are linked by AlbumId.\n Can you find the top 10 most popular artists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFRO M \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumI d\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n rtists who have albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistI d\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM \"genres\" q2\n J0IN \"tracks\" t2 0N q2.GenreId = t 2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name"}, {"rol e": "user", "content": " \n Identify artists who have albums with tracks appearing in multiple genre s:\n\n\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\n JOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT g2.GenreId\n FROM \"genres \" g2\n JOIN \"tracks\" t2 ON g2.GenreId = t2.GenreId\n GROUP BY g2.GenreId\n HAVING COUNT(g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName"}, {"role": "user", "content": "\n lists containing at least 10 tracks and the total duration of those tracks:\n"}, {"role": "assistant", "con tent": "SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM \"playlist track\" pt\nJOIN \"playlists\" p ON pt.PlaylistId = p.PlaylistId\nJOIN \"tracks\" t ON pt.TrackId = t.Trac kId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10"}, {"role": "user", "content": "\n Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT N ame, UnitPrice\nFROM \"tracks\"\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n ind the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT t. TrackId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "co Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who b ought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELEC T c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY To talAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "\n List all albums and their corresponding art ist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM

```
\"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n
album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums i
n total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(i
i.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGR
OUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n
nres and the number of tracks in each genre:\n"}]
Ollama Response:
{'model': 'phi3:latest', 'created at': '2024-06-14T02:56:03.874317138Z', 'message': {'role': 'assistant',
'content': 'SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId =
t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks DESC\nLIMIT 5'}, 'done reason': 'stop', 'done': True, 'tot
al duration': 76458395198, 'load duration': 752419, 'prompt eval count': 2008, 'prompt eval duration': 6493
8963000, 'eval count': 63, 'eval duration': 10925387000}
SELECT g.Name, COUNT(t.GenreId) AS TotalTracks
FROM "genres" g
JOIN "tracks" t ON g.GenreId = t.GenreId
GROUP BY a.Name
ORDER BY TotalTracks DESC
LIMIT 5
SELECT q.Name, COUNT(t.GenreId) AS TotalTracks
FROM "genres" g
JOIN "tracks" t ON g.GenreId = t.GenreId
GROUP BY g.Name
ORDER BY TotalTracks DESC
LIMIT 5
                Name TotalTracks
0
                Rock
                             1297
1
               Latin
                              579
2
               Metal
                              374
3 Alternative & Punk
                              332
4
                Jazz
                              130
Ollama parameters:
model=phi3:latest,
options={},
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '\n List all genres and the number of tracks in each genr
e:\n'\nThe DataFrame was produced using this query: SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM
\"qenres\" g\nJOIN \"tracks\" t ON q.GenreId = t.GenreId\nGROUP BY q.Name\nORDER BY TotalTracks DESC\nLIMIT
5\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n N
                                     int64\ndtype: object"}, {"role": "user", "content": "Can you generate
ame
              obiect\nTotalTracks
the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe call
```

ed 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do n
ot answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'phi3:latest', 'created\_at': '2024-06-14T02:56:14.258979158Z', 'message': {'role': 'assistant',
'content': "import plotly.express as px\n\nfig = px.bar(df, x='Name', y='TotalTracks')\nfig.show()"}, 'done
\_reason': 'stop', 'done': True, 'total\_duration': 10356818959, 'load\_duration': 739541, 'prompt\_eval\_coun
t': 225, 'prompt\_eval\_duration': 5876431000, 'eval\_count': 35, 'eval\_duration': 4347979000}



```
Out[30]: ('SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" q\nJOIN "tracks" t ON g.GenreId = t.GenreI
          d\nGROUP BY q.Name\nORDER BY TotalTracks DESC\nLIMIT 5',
                            Name TotalTracks
           0
                            Rock
                                         1297
           1
                           Latin
                                          579
                                          374
                           Metal
           3 Alternative & Punk
                                          332
                            Jazz
                                          130,
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'Name=%{x}<br>TotalTracks=%{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'], dtype=object),
                         'xaxis': 'x',
                         'y': array([1297, 579, 374, 332, 130]),
                         'yaxis': 'y'}],
               'lavout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalTracks'}}}
          }))
         question = """
In [31]:
             Get all genres that do not have any tracks associated with them:
         vn.ask(question=question)
        Number of requested results 10 is greater than number of elements in index 1, updating n results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE INDEX IFK TrackGenreId ON "tracks" (GenreId)\n\nCREATE TABLE "track TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL.\r\n GenreId INTEGER.\r\n Composer NVARCHAR(22 Bytes INTEGER,\r\n 0), r nMilliseconds INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NUL  $L,\r\n$ FOREIGN KEY (Albumid) REFERENCES "albums" (Albumid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI  $0N,\r\n$ FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON "playlist track" (TrackId)\n\nCREATE INDE X IFK TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE INDEX IFK TrackAlbumId ON "tracks" (AlbumId)\n\n CREATE TABLE "genres"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR  $(120)\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n$ AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist trac  $k"\r\n(\r\n$ PlavlistId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL.\r\n CONSTRAINT PK Playlis tTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Playlist Id) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (Trac kId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK AlbumArtistId ON "albums" (Ar tistId)\n\nCREATE TABLE "playlists"\r\n(\r\n PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Res ponse Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL guery to find the distinct st rings in that column. Prepend the guery with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n 5. If the question has been asked and answered before, please repeat the answer exactly as it was given bef ore. \n'}, {'role': 'user', 'content': ' \n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FR OM "genres" q2\n JOIN "tracks" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.Gen reId) > 1\n)\nGROUP BY a.ArtistId, a.Name'}, {'role': 'user', 'content': ' \n There are 3 tables: artist s, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by Albu Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assista nt', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.Ar tistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC \nLIMIT 10'}, {'role': 'user', 'content': ' \n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFRO M "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM "genres" q2\n JOIN "tracks" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.G enreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'role': 'user', 'content': ' \n t all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name,

COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.Nam e'}, {'role': 'user', 'content': ' \n List all genres and the number of tracks in each genre:\n'}, {'ro le': 'assistant', 'content': 'SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" q\nJOIN "track s" t ON q.GenreId = t.GenreId\nGROUP BY q.Name\nORDER BY TotalTracks DESC\nLIMIT 5'}, {'role': 'user', 'con Get all playlists containing at least 10 tracks and the total duration of those track s:\n'}, {'role': 'assistant', 'content': 'SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM "playlist track" pt\nJOIN "playlists" p ON pt.PlaylistId = p.PlaylistId\nJOIN "track s" t ON pt.TrackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10'}, {'role': 'user', 'content': ' \n Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': 'SELECT \* \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%\''}, {'role': 'user', 'content': ' \n List all albums and their corresponding artist names \n'}, {'role': 'assistant', 'cont ent': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n Find the top 5 most expensive tracks (based on unit p rice):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT Name, UnitPrice\nFROM "tracks"\nORDER BY UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Get all genres that do not have any trac ks associated with them:\n'\l Ollama parameters: model=phi3:latest, options={}. keep alive=None Prompt Content:

[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE INDEX IFK TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE TABLE \"tr acks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NUL AlbumId INTEGER,\r\n L.\r\n MediaTypeId INTEGER NOT NULL.\r\n GenreId INTEGER.\r\n Composer NV Milliseconds INTEGER NOT NULL,\r\n Bytes INTEGER,\r\n  $ARCHAR(220).\r\n$ UnitPrice NUMERIC(10.2) N FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE OT NULL.\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDA NO ACTION,\r\n TE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE N O ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON \"playlist track\" (TrackId) \n\nCREATE INDEX IFK TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE INDEX IFK TrackAlbumId ON \"tra cks\" (AlbumId)\n\nCREATE TABLE \"genres\"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL L.\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n EY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\CRE ATE TABLE \"playlist track\"\r\n(\r\n PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFER ENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Track Id) REFERENCES \"tracks\" (TrackId)  $\r \n \$  IF

K AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n PlaylistId INTEGER PRIMA RY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please q enerate a valid SQL guery without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermedi ate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermed iate sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Pleas e use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat th e answer exactly as it was given before. \n"}, {"role": "user", "content": " \n Identify artists who h ave albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.Ar tistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.A rtistId IN (\n SELECT q2.GenreId\n FROM \"genres\" q2\n JOIN \"tracks\" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name"}, {"role": "user", "co There are 3 tables: artists, albums and tracks, where albums and artists are linked by Arti stId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based on t he number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\n FROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.Alb umId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n y artists who have albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content t": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.Art istId\nWHERE a.ArtistId IN (\n SELECT g2.GenreId\n FROM \"genres\" g2\n J0IN \"tracks\" t2 ON g2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER B Y ArtistName"}, {"role": "user", "content": " \n List all genres and the number of tracks in each genr e:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres\" g \nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"role": "user", "content": " \n all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, CO UNT(t.GenreId) AS TotalTracks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Na me\nORDER BY TotalTracks DESC\nLIMIT 5"}, {"role": "user", "content": " \n Get all playlists containin q at least 10 tracks and the total duration of those tracks:\n"}, {"role": "assistant", "content": "SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM \"playlist track\" pt\nJ0 IN \"playlists\" p ON pt.PlaylistId = p.PlaylistId\nJOIN \"tracks\" t ON pt.TrackId = t.TrackId\nGROUP BY p t.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10"}, {"role": "user", "content": " \n s with a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT \* \nFROM \"tracks\" \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": " \n List all albums and th eir corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "cont Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "cont ent": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"ro le": "user", "content": " \n Find the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT Name, UnitPrice\nFROM \"tracks\"\nORDER BY UnitPrice DESC\nLIMIT 5"}, {"rol e": "user", "content": " \n Get all genres that do not have any tracks associated with them:\n"}] Ollama Response:

```
{'model': 'phi3:latest', 'created at': '2024-06-14T02:57:22.060360304Z', 'message': {'role': 'assistant',
'content': 'SELECT q.Name\nFROM "genres" q\nLEFT JOIN "tracks" t ON q.GenreId = t.GenreId\nWHERE t.TrackId
IS NULL;\nThis guery will return the names of genres without any associated tracks in the database. The LEF
T JOIN ensures that all records from \'genres\' are included, even if there is no matching record in the
\'tracks\' table (which would be represented by a NULL value for TrackId).'}, 'done reason': 'stop', 'don
e': True, 'total duration': 67717116709, 'load duration': 777690, 'prompt eval count': 1889, 'prompt eval d
uration': 51876134000, 'eval count': 102, 'eval duration': 15157478000}
SELECT g.Name
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL:
This query will return the names of genres without any associated tracks in the database. The LEFT JOIN ens
ures that all records from 'genres' are included, even if there is no matching record in the 'tracks' table
(which would be represented by a NULL value for TrackId).
Output from LLM: SELECT g.Name
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL:
This query will return the names of genres without any associated tracks in the database. The LEFT JOIN ens
ures that all records from 'genres' are included, even if there is no matching record in the 'tracks' table
(which would be represented by a NULL value for TrackId).
Extracted SQL: SELECT g.Name
FROM "genres" q
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
SELECT g.Name
FROM "genres" q
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.TrackId IS NULL
Empty DataFrame
Columns: [Name]
Index: []
Ollama parameters:
model=phi3:latest,
options={},
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '\n Get all genres that do not have any tracks associated wi
th them:\n'\n\nThe DataFrame was produced using this query: SELECT q.Name\nFROM \"genres\" q\nLEFT JOIN \"t
racks\" t ON g.GenreId = t.GenreId\nWHERE t.TrackId IS NULL\n\nThe following is information about the resul
ting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name
                                                                 object\ndtype: object"}, {"role": "user",
```

"content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respon d with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:

{'model': 'phi3:latest', 'created\_at': '2024-06-14T02:57:32.411946629Z', 'message': {'role': 'assistant', 'content': "import plotly.express as  $px\n = px.bar(df, x='Name', y=1)\n = show()"$ }, 'done\_reason': 's top', 'done': True, 'total\_duration': 10349392532, 'load\_duration': 944340, 'prompt\_eval\_count': 195, 'prom pt eval duration': 5423624000, 'eval count': 33, 'eval duration': 4793538000}



```
Out[31]: ('SELECT g.Name\nFROM "genres" g\nLEFT JOIN "tracks" t ON g.GenreId = t.GenreId\nWHERE t.TrackId IS NULL',
          Empty DataFrame
          Columns: [Name]
           Index: [],
           Figure({
               'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
                         'hovertemplate': 'Name=%{label}<extra></extra>',
                         'labels': array([], dtype=object),
                         'legendgroup': '',
                         'name': '',
                         'showlegend': True,
                         'type': 'pie'}],
               'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'template': '...'}
          }))
         question = """
In [32]:
             List all customers who have not placed any orders:
         vn.ask(question=question)
```

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR CustomerId INTEGER NOT NULL.\r\n EMENT NOT NULL.\r\n InvoiceDate DATETIME NOT NULL,\r\n BillingA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n **FOREIG** rv NVARCHAR(40),\r\n N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Firs tName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n ddress NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Fax NVARCHAR(24),\r\n PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r \n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineI InvoiceId INTEGER NOT NULL,\r\n d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER N UnitPrice NUMERIC(10,2) NOT NULL,\r\n OT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (I nvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n **FOREIGN** KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n TABLE "employees"\r\n(\r\n LastName NVARCH FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTE AR(20) NOT NULL,\r\n GER,\r\n BirthDate DATETIME,\r\n HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n City NVARCHAR (40), r nState NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone  $NVARCHAR(24), \r\n$ Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist tra PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n CONSTRAINT PK Plavli stTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Playlis tid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (Tra ckid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n EGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n ArtistId INTEGER NOT Title NVARCHAR(160) NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE N O ACTION\r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "playl PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n ists"\r\n(\r\n Name NVARCHAR(120)\r\n)\n\n CREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Name NVARCHAR (200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL.\r\n GenreId INTEGER,\r\n Milliseconds INTEGER NOT NULL,\r\n Composer NVARCHAR(220),\r\n Bytes INTEGER.\r\n UnitPrice NUMER IC(10,2) NOT NULL,\r\n FOREIGN KEY (Albumid) REFERENCES "albums" (Albumid) \r\n\t\tON DELETE NO ACTION FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DEL ON UPDATE NO ACTION.\r\n ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\n \n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If th e provided context is sufficient, please generate a valid SQL query without any explanations for the questi on. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a part icular 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 context is insufficient, please ex plain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the guestion has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "custom ers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.Invoi ceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n t: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most album s in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUN T(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "i nvoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer with the most invoices \n'}, {'role': 'assista nt', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoic es" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'rol e': 'user', 'content': ' \n Find the top 5 customers who spent the most money overall, \n int: order total can be found on invoices table, calculation using invoice items detail table is unnecessar y \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assi stant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "i nvoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assi stant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'ro le': 'user', 'content': ' \n List all customers from Canada and their email addresses:\n'}, {'role': 'assistant', 'content': 'SELECT c.Email, c.Country\nFROM "customers" c\nWHERE c.Country = \'Canada\''}, {'r ole': 'user', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(\*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Get the total number of invoices for e ach customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoice s\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(\*) FROM "customers"'}, {'role': 'user', 'content': ' \n List all customers who have not placed any orders:\n'}] Ollama parameters: model=phi3:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo

rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n Billin aAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40).\r\n BillinaCou Total NUMERIC(10,2) NOT NULL,\r\n ntry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n F0RE IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION 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 Email NVARCHAR(60) NOT PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) SupportRepId INTEGER,\r\n \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n ineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGE Quantity INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KE Y (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n OREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n \nCREATE TABLE \"employees\"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Name NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30).\r\n Rep BirthDate DATETIME.\r\n ortsTo INTEGER.\r\n HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n ty NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsT o) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TAB LE \"playlist track\"\r\n(\r\n PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n NSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlavlistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) RE FERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r \"\r\n(\r\n ArtistId INTEGER NOT NULL.\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\t0 N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (Supp ortRepId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n  $L.\r\n$ TrackId INTEGER PRIMARY KEY AUTOIN AlbumId INTEGER,\r\n CREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER.\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n Bytes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (Albumid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (GenreId) REFERENCES \"genres \" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK Invo iceCustomerId ON \"invoices\" (CustomerId)\n\n===Additional Context \n\nIn the SQLite database invoice me ans order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requ ires knowledge of a specific string in a particular column, please generate an intermediate SQL query to fi nd the distinct strings in that column. Prepend the query with a comment saying intermediate sql \n3. If th e provided context is insufficient, please explain why it can't be generated. \n4. Please use the most rele

vant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n Find the customer who bought the most albu ms in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, CO UNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJ OIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nL IMIT 1"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGRO UP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n stomer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceI d) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.C ustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n Find the top 5 cus tomers who spent the most money overall, \n \n Hint: order total can be found on invoices table, ca lculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SELECT c. CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Custo merId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n t: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most album s in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUN T(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.Invo iceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n nt: 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", "content": "SELECT i.CustomerId, COU NT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId \nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n ll customers from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SELECT c.Email, c.Country\nFROM \"customers\" c\nWHERE c.Country = 'Canada'"}, {"role": "user", "content": "what are the to p 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(\*) AS TotalCustomers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices \" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": "How many custome rs are there"}, {"role": "assistant", "content": "SELECT COUNT(\*) FROM \"customers\""}, {"role": "user", "c ontent": " \n List all customers who have not placed any orders:\n"}] Ollama Response: {'model': 'phi3:latest', 'created at': '2024-06-14T02:58:37.487559094Z', 'message': {'role': 'assistant', 'content': 'SELECT \* FROM "customers" WHERE CustomerId NOT IN (SELECT DISTINCT CustomerId FROM "invoice s")'}, 'done reason': 'stop', 'done': True, 'total duration': 64994590906, 'load duration': 1061612, 'promp t eval count': 2018, 'prompt eval duration': 59900327000, 'eval count': 26, 'eval duration': 4195872000} SELECT \* FROM "customers" WHERE CustomerId NOT IN (SELECT DISTINCT CustomerId FROM "invoices") SELECT \* FROM "customers" WHERE CustomerId NOT IN (SELECT DISTINCT CustomerId FROM "invoices") Empty DataFrame

```
Columns: [CustomerId, FirstName, LastName, Company, Address, City, State, Country, PostalCode, Phone, Fax,
Email, SupportRepId]
Index: []
Ollama parameters:
model=phi3:latest,
options={},
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the guery
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 guery: SELECT * FROM \"customers\" WHERE CustomerId NOT IN (SELECT
DISTINCT CustomerId FROM \"invoices\")\n\nThe following is information about the resulting pandas DataFrame
                                                   object\nFirstName
'df': \nRunning df.dtypes gives:\n CustomerId
                                                                           obiect\nLastName
                                                                                                   obiect\n
Company
                obiect\nAddress
                                        obiect\nCitv
                                                                obiect\nState
                                                                                        object\nCountry
object\nPostalCode
                        object\nPhone
                                                obiect\nFax
                                                                        object\nEmail
                                                                                                obiect\nSup
             object\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code
portRepId
to chart the results of the dataframe? 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 code. Do not answer with any expl
anations -- just the code."}]
Ollama Response:
{'model': 'phi3:latest', 'created at': '2024-06-14T02:58:55.259605946Z', 'message': {'role': 'assistant',
'content': "import plotly.express as px\n\nfig = px.bar(df.groupby('CustomerId').size(), title='Customers w
ho have not placed any orders', x='CustomerId', y='count')\nfig.update layout(showlegend=False)\nfig.show
()"}, 'done reason': 'stop', 'done': True, 'total duration': 17769594428, 'load duration': 43787525, 'promp
t eval count': 232, 'prompt eval duration': 7069095000, 'eval count': 64, 'eval duration': 10607761000}
```



```
Out[32]: ('SELECT * FROM "customers" WHERE CustomerId NOT IN (SELECT DISTINCT CustomerId FROM "invoices")',
          Empty DataFrame
          Columns: [CustomerId, FirstName, LastName, Company, Address, City, State, Country, PostalCode, Phone, Fa
         x, Email, SupportRepId]
          Index: [],
          Figure({
               'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
                         'hovertemplate': 'CustomerId=%{label}<extra></extra>',
                         'labels': array([], dtype=object),
                         'legendgroup': '',
                         'name': '',
                         'showlegend': True,
                         'type': 'pie'}],
               'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'template': '...'}
          }))
In [33]:
         question = """
             There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums
             Can you find the top 10 most popular artists based on the number of tracks
         0.00
         vn.ask(question=question)
        Number of requested results 10 is greater than number of elements in index 1, updating n results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMEN Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n T NOT NULL,\r\n MediaTypeId INTEGER NOT NU LL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n tes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" (Al bumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (G enreId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n stId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "artists"\r\n(\r\n ArtistId INTEGER PRIMARY KEY AUTOINCR EMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK AlbumArtistId ON "albums" (ArtistId)\n\n CREATE INDEX IFK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "playlists"\r\n(\r\n GER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name  $NVARCHAR(120)\r\n)\n\nCREATE TABLE "genres"\r\n(\r\n$ GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "playlis t track"\r\n(\r\n PlavlistId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL,\r\n CONSTRAINT PK P laylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Pl aylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK TrackGenreId ON "tracks"  $(GenreId)\n\nCREATE\ INDEX\ IFK\ PlaylistTrackTrackId\ ON\ "playlist\ track"\ (TrackId)\n\n===Additional\ Context$ \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is su fficient, please generate a valid SQL guery without any explanations for the guestion. \n2. If the provided context is almost sufficient but 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 comme nt saying intermediate sql \n3. If the provided context is insufficient, please explain why it can\'t be ge nerated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered befor e, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': '\n are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and track s are linked by AlbumId,\n Can you find the top 10 most popular artists based on the number of tracks \n'}, {'role': 'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJ OIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nOR DER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': '\n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.N ame AS ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM "genres" q2\n JOIN "tracks" t2 ON q2.GenreId =  $t2.GenreId\n$  GROUP BY q2.GenreId \n HAVING COUNT(g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'role': 'user', 'c Identify artists who have albums with tracks appearing in multiple genres: $\n\n\$ , {'ro le': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artists" a\nJ0IN "albums" al 0 N a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT g2.GenreId\n FROM "genres" g2\n JOIN "tracks" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistI d, a.Name'}, {'role': 'user', 'content': '\n Hint: album quantity is found in invoice items, \n

Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "i nvoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quanti \n Find the top 5 customers who bought the most albums in total quan ty is found in invoice items, \n tity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) A S TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Custome rId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n List all albums and their c orresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.ArtistId, ar.Name AS A rtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the m ost albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.Custome rId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = i i.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT q.Nam e, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.N ame\nORDER BY TotalTracks DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n List all genres and the num ber of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS Tota lTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.Name'}, {'role': 'user', 'co Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'co ntent': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'ro le': 'user', 'content': ' \n There are 3 tables: artists, albums and tracks, where albums and artists ar e linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular ar tists based on the number of tracks\n'}] Ollama parameters: model=phi3:latest, options={}, keep alive=None Prompt Content:

[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREM ENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL.\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres \" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums \"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\t0 N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"artists\"\r\n(\r\n ArtistId INTEGER PRIMAR Y KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK AlbumArtistId ON \"albums

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) $\r\n$ ) PlaylistId INTEGER NOT NULL.\r\n \nCREATE TABLE \"playlist track\"\r\n(\r\n TrackId INTEGER NOT NUL CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlavlistId) R EFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (T rackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDE X IFK TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON \"playlist track\"  $(TrackId)_n\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelin$ es \nl. If the provided context is sufficient, please generate a valid SQL guery without any explanations f or the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific str ing in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the guery with a comment saying intermediate sql \n3. If the provided context is insufficie nt, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the quest ion has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"ro le": "user". "content": " \n There are 3 tables: artists, albums and tracks, where albums and artists ar Can you find the top 10 most popular ar e linked by ArtistId, albums and tracks are linked by AlbumId,\n tists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON a l.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n Identify artists who have albums with tracks appearing in multiple genres: $\n\n\$ , {"role": "ass istant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.A rtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM \"genres\" g2\n J0IN \"tracks\" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistI d, a.Name ORDER BY ArtistName"}, {"role": "user", "content": " \n Identify artists who have albums wit h tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT g2.GenreId\n FROM \"genres\" g2\n JOIN \"tracks\" t2 ON g2.GenreId = t2.GenreId\n GROUP BY g2.Gen reId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name"}, {"role": "user", "content": "\n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most al bums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, C OUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\n JOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\n LIMIT 5"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all albums and their corresponding artist names \n"}, {"rol e": "assistant", "content": "SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"a rtists\" ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n Hint: album quantity is fou nd in invoice items. \n \n Find the top 5 customers who bought the most albums in total quantity (acr oss all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS T

```
otalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Custom
        erId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all genres and the nu
        mber of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS Tot
        alTracks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTra
        cks DESC\nLIMIT 5"}, {"role": "user", "content": "\n List all genres and the number of tracks in each
        genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres
        \" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"role": "user", "content": " \n
        ind the top 5 most expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT t.
        TrackId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "co
        ntent": " \n There are 3 tables: artists, albums and tracks, where albums and artists are linked by Arti
        stId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based on t
        he number of tracks\n"}]
        Ollama Response:
        {'model': 'phi3:latest', 'created at': '2024-06-14T03:00:14.457769248Z', 'message': {'role': 'assistant',
        'content': 'SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS NumberOfTracks\nFROM "artists" a\n
        JOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.ArtistI
        d, a EXPLAIN TOP 10 ORDER BY NumberOfTracks DESC'}, 'done reason': 'stop', 'done': True, 'total duration':
        79121263299, 'load duration': 1224117, 'prompt eval count': 1952, 'prompt eval duration': 60666907000, 'eva
        l count': 101, 'eval duration': 17772687000}
        SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS NumberOfTracks
        FROM "artists" a
        JOIN "albums" al ON a.ArtistId = al.ArtistId
        JOIN "tracks" t ON al.AlbumId = t.AlbumId
        GROUP BY a.ArtistId. a EXPLAIN TOP 10 ORDER BY NumberOfTracks DESC
        SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS NumberOfTracks
        FROM "artists" a
        JOIN "albums" al ON a.ArtistId = al.ArtistId
        JOIN "tracks" t ON al.AlbumId = t.AlbumId
        GROUP BY a.ArtistId. a EXPLAIN TOP 10 ORDER BY NumberOfTracks DESC
        Couldn't run sql: Execution failed on sql 'SELECT a.ArtistId, a.Name AS ArtistName, COUNT(t.TrackId) AS Nu
        mberOfTracks
        FROM "artists" a
        JOIN "albums" al ON a.ArtistId = al.ArtistId
        JOIN "tracks" t ON al.AlbumId = t.AlbumId
        GROUP BY a.ArtistId, a EXPLAIN TOP 10 ORDER BY NumberOfTracks DESC': near "EXPLAIN": syntax error
In [34]: question = """
              List all customers from Canada and their email addresses:
         0.00
         vn.ask(question=question)
```

Number of requested results 10 is greater than number of elements in index 1, updating n\_results = 1

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE INDEX IFK CustomerSupportRepId ON "customers" (SupportRepId)\n\nCRE CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n ATE TABLE "customers"\r\n(\r\n FirstName N VARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address City NVARCHAR(40),\r\n  $NVARCHAR(70).\r\n$ State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Post alCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NUL L.\r\n SupportRepId INTEGER,\r\n FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r\n \t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER P RIMARY KEY AUTOINCREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NU LL,\r\n BillingAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(4 Total NUMERIC(10.2) 0), r nBillingCountry NVARCHAR(40).\r\n BillingPostalCode NVARCHAR(10).\r\n FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION O NOT NULL,\r\n N UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE TABLE "e EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n mplovees"\r\n(\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30).\r\n ReportsTo INTEGER.\r\n BirthDate DATETIME.\r\n HireDate DATETIME,\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(2 4),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employee s" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice items"\r\n(\r InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n Ouantity INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTIO FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI ON\r\n)\n\nCREATE TABLE sqlite sequence(name,seq)\n\nCREATE TABLE "playlist track"\r\n(\r\n TrackId INTEGER NOT NULL,\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY (Playli NTEGER NOT NULL,\r\n FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO AC stId, TrackId),\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\t0N DELETE NO A TION ON UPDATE NO ACTION,\r\n CTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON "employees" (ReportsTo)\n\nCREATE T AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n ABLE "albums"\r\n(\r\n Title NVARCHAR(160) N OT NULL,\r\n ArtistId INTEGER NOT NULL.\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficien t but requires knowledge of a specific string in a particular column, please generate an intermediate SQL g uery to find the distinct strings in that column. Prepend the query with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use th e most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n List all customers from Canada and their email addresses:\n'}, {'role': 'assistant', 'content': 'SELECT c.Email, c.Country\nFROM "customer s" c\nWHERE c.Country = \'Canada\''}, {'role': 'user', 'content': 'what are the top 5 countries that custom ers come from?'}, {'role': 'assistant', 'content': 'SELECT c.Country, COUNT(\*) AS TotalCustomers\nFROM "cus

tomers" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT (i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Find the tota l number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT Customer.Country, COUNT(inv oice.InvoiceId) AS InvoiceCount\nFROM "customers" customer\nJOIN "invoices" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country'}, {'role': 'user', 'content': ' \n Get the total number of invo ices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS To talInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n Find the customer who bought the most albums in total quantity (acros s all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlb ums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.Inv oiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'conte nt': 'How many customers are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(\*) FROM "customers"'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money overall, \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessa ry \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customer s" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLI MIT 5'}, {'role': 'user', 'content': ' \n Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i. BillingCountry'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'r ole': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJ OIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGRO UP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n List all cu stomers from Canada and their email addresses:\n'}] Ollama parameters: model=phi3:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nC REATE TABLE \"customers\"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstNa me NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL, $\r\n$ Company NVARCHAR(80),\r\n Addr

ess NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INT CustomerId INTEGER NOT NULL,\r\n EGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n BillingAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR

BillingPostalCode NVARCHAR(10),\r\n  $(40), \r\n$ BillingCountry NVARCHAR(40),\r\n Total NUMERIC(10.2) FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION NOT NULL,\r\n ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE TABLE EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n \"employees\"\r\n(\r\n LastName NVARCHAR(2 0) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGE  $R_{i} r n$ BirthDate DATETIME,\r\n HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n City NVARCHAR(4 0), r nState NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NV  $ARCHAR(24), \r\n$ Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoice i InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n tems\"\r\n(\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\t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite sequence(name, seq)\n\nCREATE TABLE \"playlist track\"\r\n PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlavlistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackI d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON \"employee AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL s\" (ReportsTo)\n\nCREATE TABLE \"albums\"\r\n(\r\n L.\r\n Title NVARCHAR(160) NOT NULL.\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) R EFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Co ntext \n\nIn the SQLite 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 pro vided context is almost sufficient but requires knowledge of a specific string in a particular column, plea se 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 b e generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered b efore, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n List all customers from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SELECT c.E mail, c.Country\nFROM \"customers\" c\nWHERE c.Country = 'Canada'"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT c.Country, COUNT (\*) AS TotalCustomers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices \" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"rol e": "user", "content": " \n Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM \"customers\" customer \nJOIN \"invoices\" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country"}, {"role": "use Get the total number of invoices for each customer\n"}, {"role": "assistant", "cont r", "content": " \n ent": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n tomer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "cont

ent": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJ0IN \"invoice items\" ii 0N i.InvoiceId = ii.InvoiceId\nGR0UP BY c.CustomerId \nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": "How many customers are there"}, {"rol e": "assistant", "content": "SELECT COUNT(\*) FROM \"customers\""}, {"role": "user", "content": " \n nd the top 5 customers who spent 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", "cont ent": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.Cust omerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "conten Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "us er", "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", "conten t": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJ0IN \"invoices\" i ON c. CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\n ORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n List all customers from Canada and their email addresses:\n"}]

Ollama Response:

{'model': 'phi3:latest', 'created\_at': '2024-06-14T03:01:52.247458969Z', 'message': {'role': 'assistant', 'content': 'SELECT \* FROM "customers" WHERE Country=\'Canada\' AND Email IS NOT NULL;\n\n=======\nTo fin d the customer who has placed orders with the highest total price, we can join the \'Customers\', \'Invoice s\', and \'InvoiceItems\' tables. The \'TotalPrice\' of an invoice is calculated by summing up all the item prices on that particular invoice:\n\n``sql\nSELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent\nFRO M Customers c\nJOIN Invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent D ESC\nLIMIT 1;\n``\nThis query calculates the total spent per customer by summing up invoice totals and the n orders the result in descending order to get the top spender at the first row, which is selected thanks to `LIMIT 1`.'}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 97748055932, 'load\_duration': 81367 6, 'prompt\_eval\_count': 1957, 'prompt\_eval\_duration': 62449098000, 'eval\_count': 203, 'eval\_duration': 3450 7594000}

SELECT \* FROM "customers" WHERE Country='Canada' AND Email IS NOT NULL;

========

To find the customer who has placed orders with the highest total price, we can join the 'Customers', 'Invoices', and 'InvoiceItems' tables. The 'TotalPrice' of an invoice is calculated by summing up all the item prices on that particular invoice:

```sql
SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent
FROM Customers c
JOIN Invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 1;

. . .

This query calculates the total spent per customer by summing up invoice totals and then orders the result in descending order to get the top spender at the first row, which is selected thanks to `LIMIT 1`. Output from LLM: SELECT \* FROM "customers" WHERE Country='Canada' AND Email IS NOT NULL;

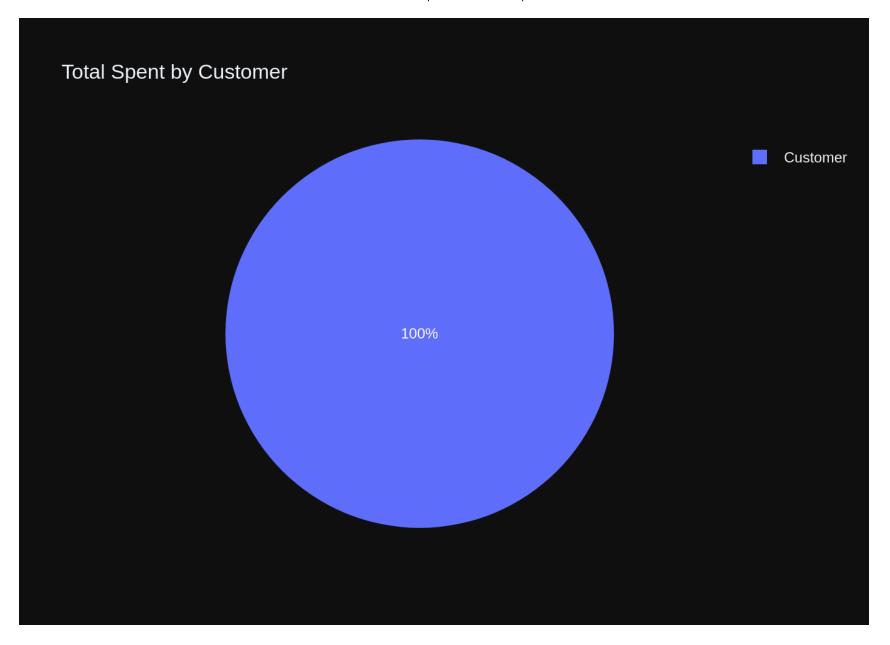
\_\_\_\_\_

To find the customer who has placed orders with the highest total price, we can join the 'Customers', 'Invoices', and 'InvoiceItems' tables. The 'TotalPrice' of an invoice is calculated by summing up all the item prices on that particular invoice:

```
```sal
SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent
FROM Customers c
JOIN Invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 1;
This query calculates the total spent per customer by summing up invoice totals and then orders the result
in descending order to get the top spender at the first row, which is selected thanks to `LIMIT 1`.
Extracted SQL: SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent
FROM Customers c
JOIN Invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 1
SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent
FROM Customers c
JOIN Invoices i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 1
   CustomerId
                         Email TotalSpent
            6 hholv@amail.com
                                     49.62
Ollama parameters:
model=phi3:latest,
options={}.
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: ' \n List all customers from Canada and their email addresse
```

s:\n'\nThe DataFrame was produced using this query: SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSp

ent\nFROM Customers c\nJOIN Invoices i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY Tota lSpent DESC\nLIMIT 1\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning d f.dtypes gives:\n CustomerId int64\nEmail object\nTotalSpent float64\ndtype: object"}, {"r ole": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? As sume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Ind icator. Respond with only Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:



```
Out[34]: ('SELECT c.CustomerId, c.Email, SUM(i.Total) AS TotalSpent\nFROM Customers c\nJOIN Invoices i ON c.Custome
          rId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 1',
              CustomerId
                                    Email TotalSpent
          0
                       6 hholy@gmail.com
                                                49.62,
          Figure({
               'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
                         'hovertemplate': 'label=%{label}<br>TotalSpent=%{value}<extra></extra>',
                         'labels': array(['Customer'], dtype=object),
                         'legendgroup': '',
                         'name': '',
                         'showlegend': True,
                         'type': 'pie',
                         'values': array([49.62])}],
               'layout': {'legend': {'tracegroupgap': 0}, 'template': '...', 'title': {'text': 'Total Spent by Custo
         mer'}}
          }))
         question = """
In [35]:
              Find the customer with the most invoices
         0.00
         vn.ask(question=question)
        Number of requested results 10 is greater than number of elements in index 1, updating n results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR EMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n BillinaA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n **FOREIG** rv NVARCHAR(40),\r\n N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON "invoice items" (InvoiceId)\n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KE InvoiceId INTEGER NOT NULL,\r\n Y AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n Price NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERE NCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLin eTrackId ON "invoice items" (TrackId)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KE Y AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NUL L.\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHA  $R(40), \r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n  $VARCHAR(24).\r\n$ FOREIGN KEY (SupportR epId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IN DEX IFK CustomerSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "employees"\r\n(\r\n Emplovee Id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVA ReportsTo INTEGER,\r\n RCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\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 Fax NVARCHAR(24),\r Phone NVARCHAR(24),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DEL Email NVARCHAR(60),\r\n ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON "employees" (ReportsTo)\n\n CREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR MediaTypeId INTEGER NOT NULL.\r\n GenreId INTEGER,\r\n (200) NOT NULL.\r\n AlbumId INTEGER.\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL,\r\n Bvtes INTEGER.\r\n UnitPrice NUMER FOREIGN KEY (AlbumId) REFERENCES "albums" (AlbumId) \r\n\t\t0N DELETE NO ACTION IC(10,2) NOT NULL,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n ON UPDATE NO ACTION.\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DEL ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL que ry without any explanations for the question. \n2. If the provided context is almost sufficient but require s knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the guery with a comment saying intermediate sql \n3. If the p rovided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most releva nt table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as i t was given before. \n'}, {'role': 'user', 'content': ' \n Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "cus tomers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices

DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money ov Hint: order total can be found on invoices table, calculation using invoice items deta erall, \n il table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalS pent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer who bought the most alb ums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, C OUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assi stant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoic es" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.Cus tomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Custom erId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n Hint: album quantity is fo und in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (ac ross all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalA lbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORD ER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n Get the total number of invoices fo r each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvo ices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'rol e': 'user', 'content': ' \n Find the total number of invoices per country:\n'}, {'role': 'assistant', 'content': 'SELECT Customer.Country, COUNT(invoice.InvoiceId) AS InvoiceCount\nFROM "customers" customer\nJ OIN "invoices" invoice ON customer.CustomerId = invoice.CustomerId\nGROUP BY Country'}, {'role': 'user', 'c ontent': '\n List all invoices with a total exceeding \$10:\n'}, {'role': 'assistant', 'content': 'SELE CT \*\nFROM "invoices" \nWHERE (Total > 10)\nGROUP BY InvoiceId'}, {'role': 'user', 'content': ' \n the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AV G(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGR OUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n Find the customer with the most invoices \n'}] Ollama parameters: model=phi3:latest. options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n Billin BillingState NVARCHAR(40),\r\n gAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillinaCou BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n ntry NVARCHAR(40),\r\n F0RE

IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDEX IFK InvoiceLineInv oiceId ON \"invoice items\" (InvoiceId)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r UnitPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Tr ackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE \"customers\"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCH AR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(2 4),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n FOREI GN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"employees EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n \"\r\n(\r\n LastName NVARCHAR(20) NOT NUL L.\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER,\r\n Bir thDate DATETIME.\r\n HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n ate NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" Fax NVARCHAR(24), \r\n (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON \"employees\" (ReportsTo)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT AlbumId INTEGER,\r\n NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n MediaTypeId INTEGER NOT NUL L.\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n es INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (A lbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"me dia types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SOLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is su fficient, 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 gen erate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comme nt saying intermediate sql \n3. If the provided context is insufficient, please explain why it can't be gen erated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered befor e, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n d the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i. InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROU P BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n op 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoices t able, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "S ELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": "\n Find the customer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assis

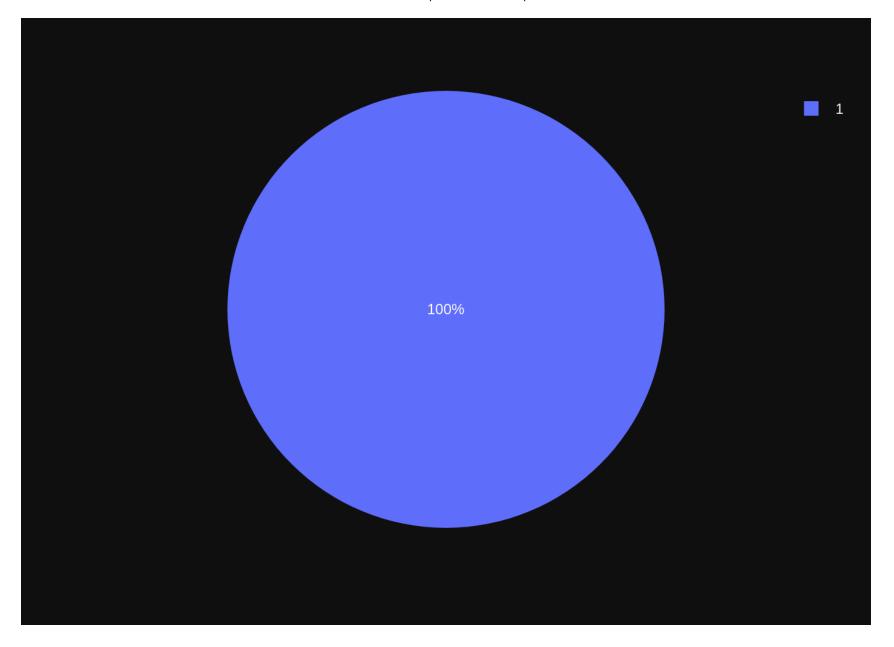
tant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invo ices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n Hint: album quanti tv is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quan tity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) A S TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice ite ms\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"rol e": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "c ontent": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"ro le": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "c ontent": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items \" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Get the total number of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices \" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT Customer.Country, COUNT (invoice.InvoiceId) AS InvoiceCount\nFROM \"customers\" customer\nJOIN \"invoices\" invoice ON customer.Cus tomerId = invoice.CustomerId\nGROUP BY Country"}, {"role": "user", "content": " \n List all invoices wi th a total exceeding \$10:\n"}, {"role": "assistant", "content": "SELECT \*\nFROM \"invoices\" \nWHERE (Total > 10)\nGROUP BY InvoiceId"}, {"role": "user", "content": " \n Get the average invoice total for each cu stomer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFR OM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n Find the customer with the most invoices \n"}] Ollama Response: {'model': 'phi3:latest', 'created at': '2024-06-14T03:03:26.573105112Z', 'message': {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\n ORDER BY NumberOfInvoices DESC\nLIMIT 1\n-----'}, 'done reason': 'stop', 'done': True, 'total duration': 67678547151, 'load duration': 742069, 'prompt eval count': 2021, 'prompt eval duration': 59137495000, 'eval count': 47, 'eval duration': 7704961000} SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices FROM Invoices GROUP BY CustomerId ORDER BY NumberOfInvoices DESC LIMIT 1 SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices FROM Invoices GROUP BY CustomerId ORDER BY NumberOfInvoices DESC

```
LIMIT 1
```

CustomerId NumberOfInvoices
0 1 7
Ollama parameters:
model=phi3:latest,
options={},
keep\_alive=None
Prompt Content:
[{"role": "system". "content": "T

[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: '\n Find the customer with the most invoices \n'\n\nThe Dat aFrame was produced using this query: SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoic es \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n----\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId int64\nNumberOfInvoices int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]

Ollama Response:



```
Out[35]: ('SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER BY
         NumberOfInvoices DESC\nLIMIT 1\n-----',
             CustomerId NumberOfInvoices
          0
                      1
                                        7,
          Figure({
              'data': [{'domain': {'x': [0.0, 1.0], 'y': [0.0, 1.0]},
                        'hovertemplate': 'CustomerId=%{label}<br>NumberOfInvoices=%{value}<extra></extra>',
                        'labels': array([1]),
                        'legendgroup': '',
                        'name': '',
                        'showlegend': True,
                        'type': 'pie',
                        'values': array([7])}],
              'layout': {'legend': {'tracegroupgap': 0}, 'margin': {'t': 60}, 'template': '...'}
          }))
In [ ]:
```

## Advanced SQL questions

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMEN AlbumId INTEGER.\r\n T NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n MediaTypeId INTEGER NOT NU LL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n tes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" (Al bumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (G enreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "invoice item InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n  $s"\r\n(\r\n$ InvoiceId INTEGER NOT NUL UnitPrice NUMERIC(10,2) NOT NULL,\r\n L, r nTrackId INTEGER NOT NULL,\r\n Ouantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDAT FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDA E NO ACTION.\r\n TE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n Albumid INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL ArtistId INTEGER NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n FOREIGN KEY (ArtistId) R EFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK Alb umArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTO InvoiceDate DATETIME NOT NULL,\r\n INCREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n ingAddress NVARCHAR(70).\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillinaC ountry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n REIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE INDEX IFK InvoiceLineTrackId ON "invoice items" (TrackId)\n\nCREATE INDEX IFK InvoiceLineIn voiceId ON "invoice items" (InvoiceId)\n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\n CREATE INDEX IFK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name  $NVARCHAR(120)\r\n)\n\n===Additional Context \n\nIn the SQ$ Lite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, ple ase generate a valid SQL query without any explanations for the question. \n2. If the provided context is a lmost sufficient but requires knowledge of a specific string in a particular column, please generate an int ermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying in termediate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n 4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please r epeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': '\n mer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'conten t': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.Cust omerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': '\n Hint: album quantity is found in invoi Find the top 5 customers who bought the most albums in total quantity (across all inv oices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = i i.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most al bums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, C

OUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.Invoi ceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n t: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most album s in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUN T(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGRO UP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n stomer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceI d) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Custo merId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Find the top 5 custome rs who spent the most money overall, \n \n Hint: order total can be found on invoices table, calcul ation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.Cust omerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nG ROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n ustomer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-----------'}, {'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 the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJO IN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assi stant', 'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDa te >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'role': 'user', 'content': ' \n Get the total number o f invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Customer Id'}, {'role': 'user', 'content': ' \n Find the customer who bought the most albums in total quantity (across all invoices): \n'}] Ollama parameters: model=phi3:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREM ENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" FOREIGN KEY (GenreId) REFERENCES \"genres (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n \" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"invoic InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n e items\"\r\n(\r\n InvoiceId INTEGER N

TrackId INTEGER NOT NULL.\r\n OT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGE FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION R NOT NULL,\r\n ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTI ON ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMEN T NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (Ar tistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IN DEX IFK AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER P RIMARY KEY AUTOINCREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NU LL,\r\n BillingAddress NVARCHAR(70).\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(4 0), r nBillingCountry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10.2) FOREIGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION NOT NULL,\r\n ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE IN DEX IFK InvoiceLineInvoiceId ON \"invoice items\" (InvoiceId)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"in voices\" (CustomerId)\n\nCREATE INDEX IFK TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"artists ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\n=== Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the pro vided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particul ar column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend t he query with a comment saying intermediate sql \n3. If the provided context is insufficient, please explai n why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the guestion has been aske d and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "con Find the customer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"custo mers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = i i.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most al bums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, C OUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\n JOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\n LIMIT 5"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}. {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices\" i\n JOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\n LIMIT 5"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find the customer with the most invoices \n"}, {"role": "assista nt", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"inv oices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most money overall, \n

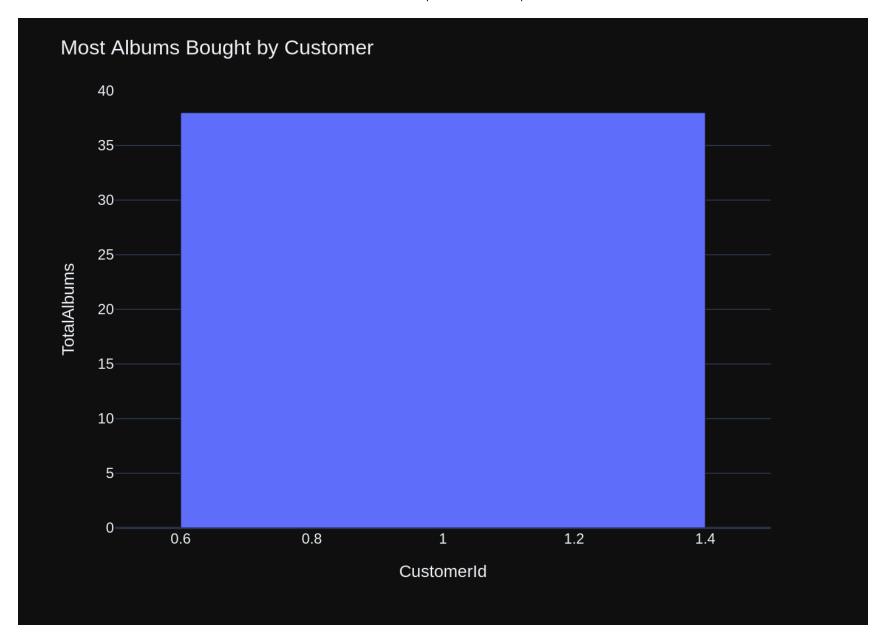
```
Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessa
ry \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customer
s\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC
\nLIMIT 5"}, {"role": "user", "content": " \n
                                                 Find the customer with the most invoices \n"}, {"role":
"assistant", "content": "SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY
CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n------"}, {"role": "user", "conten
          There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistI
d, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based on the
number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFRO
M \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumI
d\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": "\n Find all in
voices since 2010 and the total amount invoiced:\n"}, {"role": "assistant", "content": "SELECT i.InvoiceDat
e, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.Invoic
eDate"}, {"role": "user", "content": " \n Get the total number of invoices for each customer\n"}, {"rol
e": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c
\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content":
" \n
         Find the customer who bought the most albums in total quantity (across all invoices): \n"}]
Ollama Response:
{'model': 'phi3:latest', 'created at': '2024-06-14T03:04:58.281757195Z', 'message': {'role': 'assistant',
'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items"
ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n------
-----'}, 'done reason': 'stop', 'done': True, 'total duration': 72679001584, 'load duration': 778112,
'prompt eval count': 2026, 'prompt eval duration': 60062690000, 'eval count': 73, 'eval duration': 11939171
000}
SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums
FROM "invoices" i
JOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId
GROUP BY i.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1
_____
SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums
FROM "invoices" i
JOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId
GROUP BY i.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1
   CustomerId TotalAlbums
0
           1
                       38
Ollama parameters:
model=phi3:latest,
```

options={},
keep\_alive=None
Prompt Content:

[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: '\n Find the customer who bought the most albums in total q uantity (across all invoices): \n'\n\nThe DataFrame was produced using this query: SELECT i.CustomerId, COU NT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice\_items\" ii ON i.InvoiceId = ii.InvoiceId \nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n-----\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId int64\nT otalAlbums int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is on ly one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any expl anations -- just the code."}]

Ollama Response:

{'model': 'phi3:latest', 'created\_at': '2024-06-14T03:05:28.957211703Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='CustomerId', y='TotalAlbums')\nfig.update\_layout(title\_text='Most Albums Bought by Customer')\nfig.show()\n```\n\nIn case there is only one value in the dataframe:\n\n```python\nimport plotly.graph\_objects as go\n\ndata = [{'CustomerId': df['CustomerId': df['CustomerId'].values[0], 'TotalAlbums': df['TotalAlbums'].values[0]}]\nfig = go.Figure(data=[go.Bar(x=list(data), y=list(data))])\nfig.update\_layout(title='Most Albums Bought by Customer')\nfig.show()\n```"}, 'done\_reaso n': 'stop', 'done': True, 'total\_duration': 30648193716, 'load\_duration': 48576952, 'prompt\_eval\_count': 24 6, 'prompt eval duration': 6549786000, 'eval count': 175, 'eval duration': 24003610000}



```
Out[36]: ('SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.I
         ---',
            CustomerId TotalAlbums
                     1
                                38,
          Figure({
              'data': [{'alignmentgroup': 'True',
                       'hovertemplate': 'CustomerId=%{x}<br>TotalAlbums=%{y}<extra></extra>',
                       'legendgroup': '',
                       'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                       'name': '',
                       'offsetgroup': '',
                       'orientation': 'v',
                       'showlegend': False,
                       'textposition': 'auto',
                       'type': 'bar',
                       'x': array([1]),
                       'xaxis': 'x',
                       'y': array([38]),
                       'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                        'legend': {'tracegroupgap': 0},
                        'margin': {'t': 60},
                        'template': '...',
                        'title': {'text': 'Most Albums Bought by Customer'},
                        'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                        'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalAlbums'}}}
         }))
In [37]:
        question = """
            Hint: album quantity is found in invoice items,
            Find the top 5 customers who bought the most albums in total quantity (across all invoices):
        vn.ask(question=question)
       Number of requested results 10 is greater than number of elements in index 1, updating n results = 1
```

file:///home/gongai/projects/wgong/py4kids/lesson-18-ai/vanna/docs/ollama-phi3-chromadb-sglite-test-1.html

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY InvoiceId INTEGER NOT NULL,\r\n
TrackId INTEGER NOT NULL,\r\n AUTOINCREMENT NOT NULL,\r\n ice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERE NCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n bumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(22 0), r nMilliseconds INTEGER NOT NULL,\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10.2) NOT NUL FOREIGN KEY (Albumid) REFERENCES "albums" (Albumid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI  $L,\r\n$ FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT  $0N,\r\n$ FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON  $ION, \r\n$ UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NU Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK Al bumArtistId ON "albums" (ArtistId)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON "invoice items" (InvoiceId) \n\nCREATE INDEX IFK InvoiceLineTrackId ON "invoice items" (TrackId)\n\nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDa te DATETIME NOT NULL,\r\n BillingAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillinaS tate NVARCHAR(40),\r\n BillingCountry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n Total FOREIGN KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DEL NUMERIC(10,2) NOT NULL,\r\n ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\n CREATE INDEX IFK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name  $NVARCHAR(120)\r\n)\n\n===Additional Context \n\nIn the SQ$ Lite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, ple ase generate a valid SQL query without any explanations for the question. \n2. If the provided context is a lmost sufficient but requires knowledge of a specific string in a particular column, please generate an int ermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying in termediate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n 4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please r epeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n ntity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total q uantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackI d) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice item s" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n Find the top 5 cust omers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'conte nt': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'us Hint: album quantity is found in invoice items, \n \n Find the top 5 custome er', 'content': ' \n rs who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'conten

t': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "custom ers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.Invoi ceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n nd the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assista nt', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJ0IN "invoice it ems" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1\n-----------'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money Hint: order total can be found on invoices table, calculation using invoice items de overall, \n tail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS Tota lSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'conten t': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al. ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 1 0'}, {'role': 'user', 'content': ' \n Find the customer with the most invoices \n'}, {'role': 'assista nt', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoic es" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'rol e': 'user', 'content': ' \n Find the customer with the most invoices \n'\}, {'role': 'assistant', 'cont ent': 'SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n-----'}, {'role': 'user', 'content': '\n he top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.Track Id, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought th e most albums in total quantity (across all invoices):\n'}] Ollama parameters: model=phi3:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY K EY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL,\r\n
TrackId INTEGER NOT NULL,\r\n tPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCE Quantity INTEGER NOT NULL,\r\n S \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) RE FERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Name NVARCHAR(200) NOT NULL,\r\n \"\r\n(\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(22

UnitPrice NUMERIC(10.2) NOT NUL 0), r nMilliseconds INTEGER NOT NULL.\r\n Bytes INTEGER,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO AC  $L,\r\n$ TION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTI ACTION.\r\n ON ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n Albumid INTEGER PRIMARY KEY AUTOINCREMEN T NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (Ar tistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE IN DEX IFK AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON \"invoice items \" (InvoiceId)\n\nCREATE INDEX IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE \"invo InvoiceId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n CustomerId INTEGER NOT NUL L,\r\n InvoiceDate DATETIME NOT NULL,\r\n BillingAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(4 0), r nBillingState NVARCHAR(40),\r\n BillingCountry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR  $(10), \r\n$ Total NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (CustomerId) REFERENCES \"customers\" (Custo merId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invo ices\" (CustomerId)\n\nCREATE INDEX IFK TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"artists\"\r  $\n(\r\n$ ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name  $NVARCHAR(120)\r\n)\n\n===Addi$ tional Context \n\nIn the SOLite database invoice means order\n\n===Response Guidelines \n1. If the provide d context is sufficient, please generate a valid SQL guery without any explanations for the guestion. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular co lumn, please generate an intermediate SQL guery to find the distinct strings in that column. Prepend the gu ery with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought th e most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.Cust omerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Cus tomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbu ms DESC\nLIMIT 5"}, {"role": "user", "content": "\n Hint: album quantity is found in invoice items, \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"r ole": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM \"invoices \" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items. \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"r ole": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\n JOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\n LIMIT 5"}, {"role": "user", "content": " \n Find the customer who bought the most albums in total guan tity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice it ems\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"rol e": "user", "content": " \n Find the customer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nF ROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER B

```
Y TotalAlbums DESC\nLIMIT 1\n------"}, {"role": "user", "content": " \n Find the to
p 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoices ta
ble, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistant", "content": "SE
LECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId =
i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": "\n
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 trac
ks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\"
a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.
Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n Find the customer with t
he most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalIn
voices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nOR
DER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n Find the customer with the most
invoices \n"}, {"role": "assistant", "content": "SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \n
{"role": "user", "content": " \n Find the top 5 most expensive tracks (based on unit price):\n"}, {"rol
e": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.UnitPrice
DESC\nLIMIT 5"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n
     Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}]
Ollama Response:
{'model': 'phi3:latest', 'created at': '2024-06-14T03:06:37.29021214Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii
ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n-------
-----'}, 'done reason': 'stop', 'done': True, 'total duration': 68235322985, 'load duration': 709720, 'p
rompt eval count': 2032, 'prompt eval duration': 56226237000, 'eval count': 73, 'eval duration': 1132549200
0}
SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums
FROM "invoices" i
JOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId
GROUP BY i.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5
SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums
FROM "invoices" i
JOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId
GROUP BY i.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5
  CustomerId TotalAlbums
0
           1
```

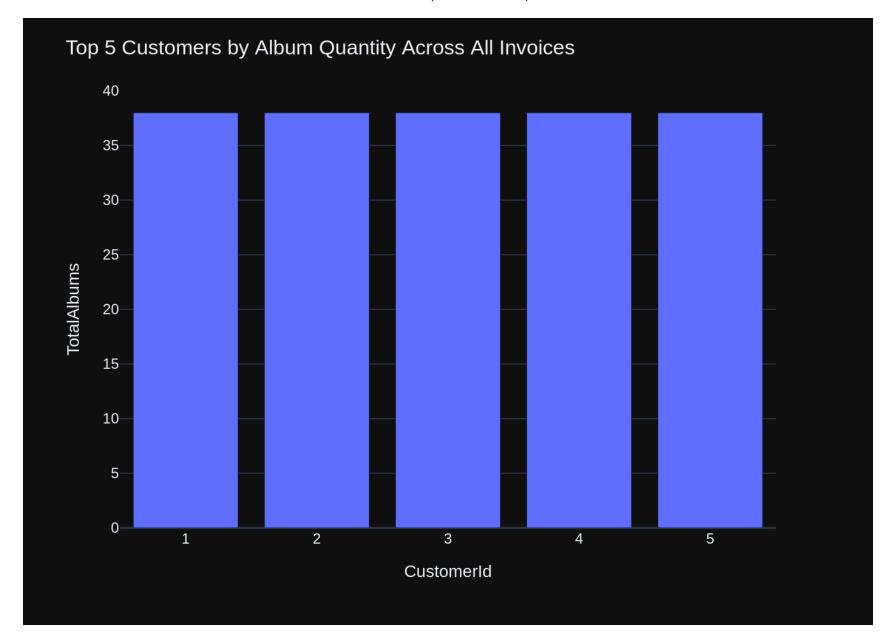
```
1 2 38
2 3 38
3 4 38
4 5 38
0llama parameters:
model=phi3:latest,
```

options={},
keep alive=None

Prompt Content:

[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: '\n Hint: album quantity is found in invoice\_items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'\n\nThe Data Frame was produced using this query: SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices \" i\nJOIN \"invoice\_items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n----\n\nThe following is information about the resulting pandas DataFra me 'df': \nRunning df.dtypes gives:\n CustomerId int64\nTotalAlbums int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicato r. Respond with only Python code. Do not answer with any explanations -- just the code."}]

{'model': 'phi3:latest', 'created\_at': '2024-06-14T03:06:58.152142826Z', 'message': {'role': 'assistant', 'content': "```python\nimport plotly.express as px\n\n# Assuming df has been preprocessed to have a single row if needed for indicator charting\nfig = px.bar(df, x='CustomerId', y='TotalAlbums')\nfig.update\_layout (title='Top 5 Customers by Album Quantity Across All Invoices')\nfig.show()\n```"}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 20834225016, 'load\_duration': 708813, 'prompt\_eval\_count': 266, 'prompt\_eval\_duration': 7568197000, 'eval\_count': 87, 'eval\_duration': 13177411000}



```
Out[37]: ('SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.I
         ---',
            CustomerId TotalAlbums
         0
                    1
                    2
          1
                               38
         2
                    3
                               38
          3
                               38
                    5
                                38,
         Figure({
             'data': [{'alignmentgroup': 'True',
                       'hovertemplate': 'CustomerId=%{x}<br>TotalAlbums=%{y}<extra></extra>',
                       'legendgroup': '',
                      'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                       'name': '',
                      'offsetgroup': '',
                      'orientation': 'v',
                       'showlegend': False,
                       'textposition': 'auto',
                      'type': 'bar',
                      'x': array([1, 2, 3, 4, 5]),
                      'xaxis': 'x',
                      'y': array([38, 38, 38, 38, 38]),
                      'yaxis': 'y'}],
             'layout': {'barmode': 'relative',
                       'legend': {'tracegroupgap': 0},
                       'margin': {'t': 60},
                        'template': '...',
                       'title': {'text': 'Top 5 Customers by Album Quantity Across All Invoices'},
                       'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                       'vaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalAlbums'}}}
         }))
        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
```

Number of requested results 10 is greater than number of elements in index 1, updating n\_results = 1

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOINCR InvoiceDate DATETIME NOT NULL,\r\n EMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n BillinaA ddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillingCount BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n rv NVARCHAR(40),\r\n **FOREIG** N KEY (CustomerId) REFERENCES "customers" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE "invoice items"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n UnitPrice NUMERIC(10.2) NOT NULL.\r FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\t Quantity INTEGER NOT NULL,\r\n ON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t \t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON "invoice items" (InvoiceId)\n\nCREATE INDEX IFK InvoiceCustomerId ON "invoices" (CustomerId)\n\nCREATE INDEX IFK InvoiceLin eTrackId ON "invoice items" (TrackId)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KE Y AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NUL  $L.\r\n$ Company NVARCHAR(80),\r\n Address NVARCHAR(70).\r\n City NVARCHAR(40).\r\n State NVARCHA  $R(40), \r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n  $VARCHAR(24).\r\n$ FOREIGN KEY (SupportR epId) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TA BLE "employees"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR (20) NOT NULL.\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGE City NVARCHAR(4  $R.\r\n$ BirthDate DATETIME.\r\n HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n State NVARCHAR(40), \r\n Phone NV  $0), \r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Fax NVARCHAR(24),\r\n  $ARCHAR(24).\r\n$ Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "tracks"\r\n (\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n bumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(22  $0), \r\n$ Milliseconds INTEGER NOT NULL,\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NUL FOREIGN KEY (Albumid) REFERENCES "albums" (Albumid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTI  $L,\r\n$ FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACT  $0N,\r\n$ FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON  $ION, \r\n$ UPDATE NO ACTION\r\n)\n\nCREATE TABLE "playlist track"\r\n(\r\n PlaylistId INTEGER NOT NULL.\r\n Tra CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n ckId INTEGER NOT NULL.\r\n FOREI GN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r \n  $\n)\n\n$ CREATE INDEX IFK EmployeeReportsTo ON "employees" (ReportsTo)\n\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, p lease generate a valid SQL guery without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an in termediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying i ntermediate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n 4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please r

epeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n Find the top 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoices table, calculation using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Custome rId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums i n total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i i.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invo ice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n op 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistan t', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice ite ms" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n-----------'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'r ole': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC \nLIMIT 5'}, {'role': 'user', 'content': '\n Hint: album quantity is found in invoice items, \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "in voice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'us er', 'content': ' \n Find the customer who bought the most albums in total quantity (across all invoic es): \n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "i nvoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlb ums DESC\nLIMIT 1\n-----'}, {'role': 'user', 'content': ' \n Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SEL ECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJ0IN "invoice items" ii 0N i.InvoiceId = ii.InvoiceId\nGR0UP BY c.CustomerId\n0RDER BY TotalA lbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Find the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER BY NumberOfInvoices DESC\nLIMIT 1\n------'}, {'role': 'use r', 'content': ' \n Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'conten t': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJ0IN "invoices" i ON c.C ustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n tomers who spent the most money overall, \n \n Hint: order total can be found on invoices table, ca lculation using invoice items detail table is unnecessary \n'}] Ollama parameters: model=phi3:latest, options={},

keep\_alive=None
Prompt Content:

[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CustomerId INTEGER NOT NULL,\r\n CREMENT NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n Billin aAddress NVARCHAR(70).\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillinaCou ntry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR(10),\r\n Total NUMERIC(10,2) NOT NULL,\r\n F0RE IGN KEY (CustomerId) REFERENCES \"customers\" (CustomerId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL \r\n)\n\nCREATE TABLE \"invoice items\"\r\n(\r\n UnitPrice NUMERIC(10.2) NO L.\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceI T NULL,\r\n Ouantity INTEGER NOT NULL.\r\n d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (Tra ckId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON \"in voice items\" (InvoiceId)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDE X IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE \"customers\"\r\n(\r\n CustomerI d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n CHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(2 4),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n FOREI GN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE TABLE \"employees\"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER,\r\n BirthDate DATETIME,\r\n HireDate DATETIME,\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(1 Fax NVARCHAR(24),\r\n  $0), \r\n$ Phone NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (Repo rtsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) AlbumId INTEGER.\r\n NOT NULL,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER.\r\n Milliseconds INTEGER NOT NULL,\r\n oser NVARCHAR(220),\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(1 FOREIGN KEY (Albumid) REFERENCES \"albums\" (Albumid) \r\n\t\tON DELETE NO ACTION ON 0,2) NOT NULL,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\t0N DELETE NO ACTION UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON D ON UPDATE NO ACTION.\r\n ELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist track\"\r\n(\r\n PlavlistId INTEGER CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, Tr NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION O ackId),\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTIO N UPDATE NO ACTION.\r\n N ON UPDATE NO ACTION\r\n)\n\CREATE INDEX IFK EmployeeReportsTo ON \"employees\" (ReportsTo)\n\n===Addit ional Context \n\nIn the SQLite 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 colum n, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the guery

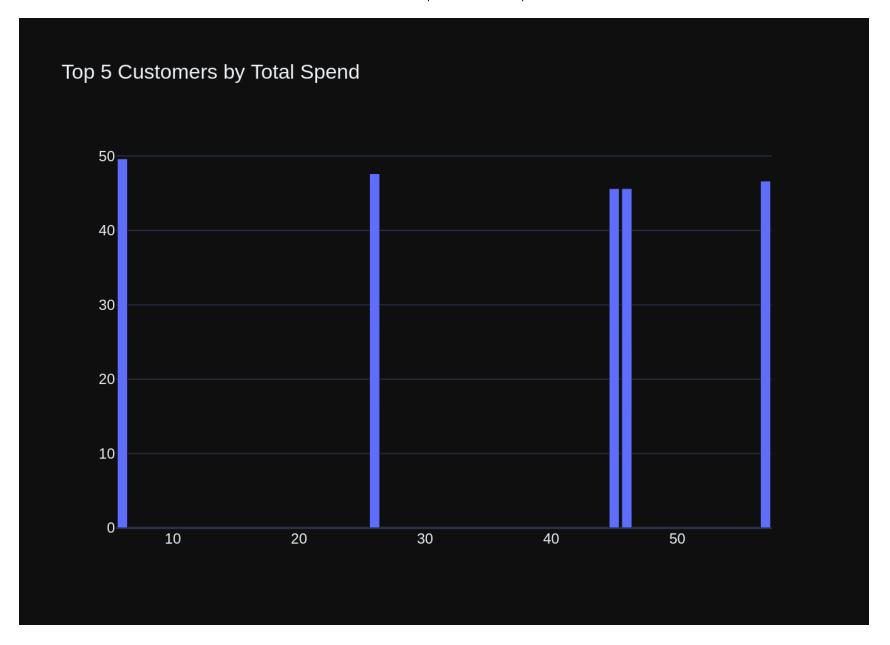
with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and ans wered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " Find the top 5 customers who spent the most money overall, \n Hint: order total can be fo \n und on invoices table, calculation using invoice items detail table is unnecessary \n"}, {"role": "assistan t", "content": "SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "use r", "content": " \n Hint: album quantity is found in invoice items, \n \n Find the top 5 customer s who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.Cust omerId = i.CustomerId\nJ0IN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGR0UP BY c.CustomerId\nORDE R BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "\n Hint: album quantity is found in inv Find the top 5 customers who bought the most albums in total quantity (across all i nvoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFRO M \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5\n------"}, {"role": "user", "content": " \n Hint: album qu antity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.Invoic eLineId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROU P BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": "\n Hint: album q uantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackI d) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS T otalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Customer Id\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", "content": " \n Find the customer who bou ght the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.Invoice {"role": "user", "content": " \n Find the customer who bought the most albums in total quantity (acros s all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlb ums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELEC "content": " \n T CustomerId, COUNT(InvoiceId) AS NumberOfInvoices \nFROM Invoices \nGROUP BY CustomerId\nORDER BY NumberOf Invoices DESC\nLIMIT 1\n-----"}, {"role": "user", "content": " \n nvoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.C ustomerId"}, {"role": "user", "content": " \n Find the top 5 customers who spent the most money overal l. \n Hint: order total can be found on invoices table, calculation using invoice items detail t able is unnecessary \n"}]

```
Ollama Response:
{'model': 'phi3:latest', 'created at': '2024-06-14T03:08:05.28787172Z', 'message': {'role': 'assistant', 'c
ontent': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.Custo
merId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n--------------------
'}, 'done reason': 'stop', 'done': True, 'total duration': 67048540282, 'load duration': 743624, 'prompt ev
al count': 1975, 'prompt eval duration': 56116255000, 'eval count': 66, 'eval duration': 9997177000}
SELECT c.CustomerId, SUM(i.Total) AS TotalSpent
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 5
SELECT c.CustomerId, SUM(i.Total) AS TotalSpent
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
LIMIT 5
   CustomerId TotalSpent
0
          6
                   49.62
1
           26
                   47.62
2
                   46.62
          57
3
          45
                   45.62
          46
                   45.62
Ollama parameters:
model=phi3:latest,
options={},
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '\n Find the top 5 customers who spent the most money overa
                 Hint: order total can be found on invoices table, calculation using invoice items detail
ll.\n
table is unnecessary \n'\n\nThe DataFrame was produced using this query: SELECT c.CustomerId, SUM(i.Total)
AS TotalSpent\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Custome
rId\nORDER BY TotalSpent DESC\nLIMIT 5\n------\n\nThe following is information about the
resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId int64\nTotalSpent
\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the result
s of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the
dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the
```

code."}]

## Ollama Response:

{'model': 'phi3:latest', 'created\_at': '2024-06-14T03:08:23.096401435Z', 'message': {'role': 'assistant', 'content': "import plotly.graph\_objs as go\n\ntrace = go.Bar(x=df['CustomerId'], y=df['TotalSpent'])\ndata = [trace]\nlayout = go.Layout(title='Top 5 Customers by Total Spend')\nfig = go.Figure(data=data, layout=la yout)\nfig.show()"}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 17788626696, 'load\_duration': 6 34606, 'prompt\_eval\_count': 261, 'prompt\_eval\_duration': 6887309000, 'eval\_count': 80, 'eval\_duration': 108 05448000}



```
Out[38]: ('SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId =
         i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5\n-----',
             CustomerId TotalSpent
          0
                      6
                              49.62
          1
                     26
                              47.62
                             46.62
                     57
          3
                              45.62
                     45
                              45.62,
                     46
          Figure({
              'data': [{'type': 'bar', 'x': array([ 6, 26, 57, 45, 46]), 'y': array([49.62, 47.62, 46.62, 45.62, 4
         5.62])}],
              'layout': {'template': '...', 'title': {'text': 'Top 5 Customers by Total Spend'}}
          }))
         question = """
In [39]:
              Get all playlists containing at least 10 tracks and the total duration of those tracks:
         vn.ask(question=question)
        Number of requested results 10 is greater than number of elements in index 1, updating n results = 1
```

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE INDEX IFK PlaylistTrackTrackId ON "playlist track" (TrackId)\n\nCRE ATE TABLE "plavlists"\r\n(\r\n PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCH  $AR(120)\r\n)\n\nCREATE TABLE "playlist track"\r\n(\r\n$ PlaylistId INTEGER NOT NULL.\r\n TrackId INTE CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n GER NOT NULL,\r\n FOREIGN KEY (P laylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n F0RE IGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCRE ATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(20 0) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Milliseconds INTEGER NOT NULL,\r\n Composer NVARCHAR(220),\r\n Bytes INTEGER,\r\n UnitPrice NUMER IC(10,2) NOT NULL,\r\n FOREIGN KEY (Albumid) REFERENCES "albums" (Albumid) \r\n\t\tON DELETE NO ACTION FOREIGN KEY (GenreId) REFERENCES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DEL ETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK TrackGenreId ON "tracks" (GenreId)\n\nCREATE IND EX IFK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE INDEX IFK TrackMediaTypeId ON "tracks" (MediaTypeId)\n \nCREATE INDEX IFK AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "albums"\r\n(\r\n AlbumId INTEGE R PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NU FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "genres"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Res ponse Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct st rings in that column. Prepend the guery with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n 5. If the question has been asked and answered before, please repeat the answer exactly as it was given bef Get all playlists containing at least 10 tracks and the to ore. \n'}, {'role': 'user', 'content': ' \n tal duration of those tracks:\n'}, {'role': 'assistant', 'content': 'SELECT pt.PlaylistId, p.Name AS Playli stName, SUM(t.Milliseconds) AS TotalDuration\nFROM "playlist track" pt\nJOIN "playlists" p ON pt.PlaylistId = p.PlaylistId\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt. TrackId) >= 10'}, {'role': 'user', 'content': '\n List all genres and the number of tracks in each gen re:\n'}, {'role': 'assistant', 'content': 'SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" q \nJOIN "tracks" t ON q.GenreId = t.GenreId\nGROUP BY q.Name'}, {'role': 'user', 'content': ' \n l genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUN T(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nOR DER BY TotalTracks DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n There are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'cont ent': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = a l.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 1 0'}, {'role': 'user', 'content': ' \n Identify artists who have albums with tracks appearing in multip

le genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artist s" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT g2.GenreId\n FROM "gen res" q2\n J0IN "tracks" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name'}, {'role': 'user', 'content': ' \n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.N ame AS ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM "genres" q2\n JOIN "tracks" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId \n HAVING COUNT(g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'role': 'user', 'c Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "custom ers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.Invoi ceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Find the top 5 customers who bought the most album t: album quantity is found in invoice items, \n \n s in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUN T(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "i nvoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS To talAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId \nORDER BY TotalAlbums DESC\nLIMIT 1\n------'}, {'role': 'user', 'content': '\n t: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most album s in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUN T(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGRO UP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n ylists containing at least 10 tracks and the total duration of those tracks:\n'}] Ollama parameters: model=phi3:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE INDEX IFK PlaylistTrackTrackId ON \"playlist track\" (TrackId)\n\nC PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n REATE TABLE \"playlists\"\r\n(\r\n Name NV ARCHAR(120)\r\n)\n\nCREATE TABLE \"playlist track\"\r\n(\r\n PlaylistId INTEGER NOT NULL,\r\n TrackI CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n d INTEGER NOT NULL,\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Name NVA RCHAR(200) NOT NULL.\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGE Composer NVARCHAR(220),\r\n  $R.\r\n$ Milliseconds INTEGER NOT NULL.\r\n Bytes INTEGER,\r\n UnitP rice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\t0N DELET

E NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DE FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeI LETE NO ACTION ON UPDATE NO ACTION.\r\n d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK TrackGenreId ON \"tracks\" (Gen reid)\n\nCREATE INDEX IFK TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDEX IFK TrackMediaTypeId ON \"tr acks\" (MediaTypeId)\n\nCREATE INDEX IFK AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"albums \"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r ArtistId INTEGER NOT NULL.\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\t0 N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"genres\"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name  $NVARCHAR(120)\r\n)\n\n==Additional Context \n\nIn the SQLite dat$ abase invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please gene rate a valid SQL query without any explanations for the question. \n2. If the provided context is almost su fficient but requires knowledge of a specific string in a particular column, please generate an intermediat e SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermedia te sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": " \n Get all playlists contai ning at least 10 tracks and the total duration of those tracks:\n"}, {"role": "assistant", "content": "SELE CT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM \"playlist track\" pt \nJOIN \"playlists\" p ON pt.PlaylistId = p.PlaylistId\nJOIN \"tracks\" t ON pt.TrackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10"}, {"role": "user", "content": " \n enres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT (t.GenreId) AS TotalTracks $\nFROM \genres \genres \del{fig:control} \del{fig:control} $$ (t.GenreId) AS TotalTracks \nFROM \genres \genres \del{fig:control} $$ g.Nam \genres \genres \del{fig:control} $$ (t.GenreId) AS TotalTracks \genres \genres$ e"}, {"role": "user", "content": " \n List all genres and the number of tracks in each genre:\n"}, {"ro le": "assistant", "content": "SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres\" q\nJOIN \"tra cks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks DESC\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 o n the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTrack s\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJOIN \"tracks\" t ON al.AlbumId = t. AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"role": "user", "content": " \n tify artists who have albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "cont ent": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJ0IN \"albums\" al ON a.ArtistId = al.A rtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM \"genres\" q2\n J0IN \"tracks\" t2 ON q2.Genre Id = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name"}, {"role": "user", "content": " \n Identify artists who have albums with tracks appearing in multiple ge nres:\n\n", {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM \"gen res\" q2\n JOIN \"tracks\" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreI d) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName"}, {"role": "user", "content": " \n Find th e customer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.Custo

```
merId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n Hint: album quantity is f
                                          Find the top 5 customers who bought the most albums in total quantity (a
        ound in invoice items, \n \n
        cross all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS Total
        Albums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii
        ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "use
                                Find the customer who bought the most albums in total quantity (across all invoice
        r", "content": " \n
        s): \n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"i
        nvoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY Total
        Albums DESC\nLIMIT 1\n-----"}, {"role": "user", "content": " \n Hint: album quantit
        y is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quant
        ity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS
        TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Custo
        merId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n
                                                                                         Get all playlists contai
        ning at least 10 tracks and the total duration of those tracks:\n"}]
        Ollama Response:
        {'model': 'phi3:latest', 'created at': '2024-06-14T03:09:33.778384012Z', 'message': {'role': 'assistant',
        'content': 'SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDuration\nFROM "playlist track" ptt JOIN "trac
        ks" t ON ptt.TrackId = t.TrackId\nGROUP BY p.PlaylistId\nHAVING COUNT(ptt.TrackId) >= 10\nORDER BY TotalDur
        ation DESC\nLIMIT 5'}, 'done reason': 'stop', 'done': True, 'total duration': 70569147454, 'load duration':
        681478, 'prompt eval count': 2020, 'prompt eval duration': 56950140000, 'eval count': 80, 'eval duration':
        12939123000}
        SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDuration
        FROM "playlist track" ptt JOIN "tracks" t ON ptt.TrackId = t.TrackId
        GROUP BY p.PlaylistId
        HAVING COUNT(ptt.TrackId) >= 10
        ORDER BY TotalDuration DESC
        LIMIT 5
        SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDuration
        FROM "playlist track" ptt JOIN "tracks" t ON ptt.TrackId = t.TrackId
        GROUP BY p.PlaylistId
        HAVING COUNT(ptt.TrackId) >= 10
        ORDER BY TotalDuration DESC
        LIMIT 5
        Couldn't run sql: Execution failed on sql 'SELECT p.PlaylistId, SUM(t.Milliseconds) AS TotalDuration
        FROM "playlist track" ptt JOIN "tracks" t ON ptt.TrackId = t.TrackId
        GROUP BY p.PlaylistId
        HAVING COUNT(ptt.TrackId) >= 10
        ORDER BY TotalDuration DESC
        LIMIT 5': no such column: p.PlaylistId
In [40]: question = """
```

file:///home/gongai/projects/wgong/py4kids/lesson-18-ai/vanna/docs/ollama-phi3-chromadb-sqlite-test-1.html

Identify artists who have albums with tracks appearing in multiple genres:

0.00

vn.ask(question=question)

Number of requested results 10 is greater than number of elements in index 1, updating n\_results = 1

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMEN Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n T NOT NULL,\r\n MediaTypeId INTEGER NOT NU LL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n tes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" (Al bumid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (GenreId) REFERENCES "genres" (G enreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK AlbumArtistI d ON "albums" (ArtistId)\n\nCREATE INDEX IFK TrackGenreId ON "tracks" (GenreId)\n\nCREATE INDEX IFK TrackAl bumId ON "tracks" (AlbumId)\n\nCREATE TABLE "albums"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n NOT NULL,\r\n FOREIGN KEY (Arti stid) REFERENCES "artists" (Artistid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK TrackMediaTypeId ON "tracks" (MediaTypeId)\n\nCREATE TABLE "genres"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON "pla ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT ylist track" (TrackId)\n\nCREATE TABLE "artists"\r\n(\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "playlist track"\r\n(\r\n NULL,\r\n PlavlistId INTEGER NOT N CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackI ULL,\r\n TrackId INTEGER NOT NULL,\r\n d),\r\n FOREIGN KEY (PlaylistId) REFERENCES "playlists" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDA TE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPD ATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL guery without any explan ations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a spec ific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate sql \n3. If the provided context is in sufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': 'user', 'content': ' \n Identify artists who have albums with tracks appearing in multi ple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artis ts" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT g2.GenreId\n FROM "ge nres" q2\n J0IN "tracks" t2 0N q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name'}, {'role': 'user', 'content': ' \n Identify artists who have albu ms with tracks appearing in multiple genres:\n\n\n'}, {'role': 'assistant', 'content': 'SELECT a.ArtistId, a.Name AS ArtistName\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT g2.GenreId\n FROM "genres" g2\n JOIN "tracks" t2 ON g2.GenreId = t2.GenreId\n GROUP BY g2.Ge nreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName'}, {'role': 'use r', '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 bas ed on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalT racks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.Al bumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n l albums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.Ar tistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId'}, {'role':

'user', 'content': ' \n List all genres and the number of tracks in each genre:\n'}, {'role': 'assistan t', 'content': 'SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" q\nJOIN "tracks" t ON q.Genre Id = t.GenreId\nGROUP BY g.Name'}, {'role': 'user', 'content': ' \n List all genres and the number of t racks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS TotalTracks \nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.Name\nORDER BY TotalTracks DESC\nLI MIT 5'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n \n ind the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "i nvoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Hint: album quanti ty is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quan tity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) A S TotalAlbums\nFROM "invoices" i\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.Custome rId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n Find the customer who boug ht the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.C ustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbu ms DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'r ole': 'assistant', 'content': 'SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "invoices" i\nJ0 IN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMI T 5\n-----'}, {'role': 'user', 'content': ' \n Identify artists who have albums wi th tracks appearing in multiple genres:\n\n\n'}] Ollama parameters: model=phi3:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREM ENT NOT NULL.\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" FOREIGN KEY (GenreId) REFERENCES \"genres (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n \" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK Albu mArtistId ON \"albums\" (ArtistId)\n\nCREATE INDEX IFK TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX IFK TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n EIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)

\n\nCREATE INDEX IFK TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE TABLE \"genres\"\r\n(\r\n nreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK Playli stTrackTrackId ON \"playlist track\" (TrackId)\n\nCREATE TABLE \"artists\"\r\n(\r\n ArtistId INTEGER PRI MARY KEY AUTOINCREMENT NOT NULL.\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"playlist track\"\r\n(\r\n CONSTRAINT PK PlaylistTrack PRIMARY PlavlistId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL,\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\t0 KEY (PlaylistId, TrackId),\r\n N DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t \t0N DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===Additional Context \n\nIn the SQLite database invoic e means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermediate SQL query t o find the distinct strings in that column. Prepend the guery with a comment saying intermediate sql \n3. I f the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactl y as it was given before. \n"}, {"role": "user", "content": " \n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name A S ArtistName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM \"genres\" q2\n JOIN \"tracks\" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.Gen reId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name"}, {"role": "user", "content": "\n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Name AS ArtistName\nFROM \"artists\" a\nJ0IN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM \"genres\" q2\n JOIN \"tracks\" t2 ON q2.G enreId = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name ORDER BY ArtistName"}, {"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 the top 10 most popular artists based on the number of tracks\n"}, {"role": "assistant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId \nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}. {"ro le": "user", "content": " \n List all albums and their corresponding artist names \n"}, {"role": "assi stant", "content": "SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"role": "user", "content": " \n List all genres and the number of tr acks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\n FROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.GenreId\nGROUP BY g.Name"}, {"role": "user", "conten List all genres and the number of tracks in each genre:\n"}, {"role": "assistant", "content": "SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres\" g\nJOIN \"tracks\" t ON g.GenreId = t.Genr eId\nGROUP BY q.Name\nORDER BY TotalTracks DESC\nLIMIT 5"}, {"role": "user", "content": " \n m quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in to tal quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.Tr ackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"inv oice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Hint: album quantity is found in invoice items, \n \n op 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistan

```
t", "content": "SELECT i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice
items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"ro
le": "user", "content": " \n
                                Find the customer who bought the most albums in total quantity (across al
l invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums
\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii ON i.
InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "co
               Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who b
ought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELEC
T i.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"invoices\" i\nJOIN \"invoice items\" ii ON i.Invoi
"}, {"role": "user", "content": " \n Identify artists who have albums with tracks appearing in multipl
e genres:\n\n\n"}]
Ollama Response:
{'model': 'phi3:latest', 'created at': '2024-06-14T03:10:59.204861487Z', 'message': {'role': 'assistant',
'content': 'SELECT DISTINCT a.ArtistId, a.Name AS ArtistName\nFROM "artists" a\nWHERE EXISTS (\n SELECT g
2.GenreId\n FROM "genres" g2\n JOIN "tracks" t ON g2.GenreId = t.GenreId\n WHERE a.ArtistId = (\n
                                        WHERE al.AlbumId IN (SELECT t.AlbumId FROM "tracks" t)\n
                   FROM "albums" al\n
ECT al.ArtistId\n
                                                                                                   AN
                   SELECT 1\n
                                    FROM "genres" g\n
                                                                                                AND C
D EXISTS (\n
                                                           WHERE t.GenreId = q.GenreId\n
OUNT(q.GenreId) > 1\n )\n)\nORDER BY ArtistName;'}, 'done reason': 'stop', 'done': True, 'total dur
ation': 85379663186, 'load duration': 838263, 'prompt eval count': 1894, 'prompt eval duration': 5612525000
0, 'eval count': 178, 'eval duration': 28564817000}
SELECT DISTINCT a.ArtistId, a.Name AS ArtistName
FROM "artists" a
WHERE EXISTS (
  SELECT a2.GenreId
  FROM "genres" q2
 JOIN "tracks" t ON g2.GenreId = t.GenreId
 WHERE a.ArtistId = (
   SELECT al.ArtistId
   FROM "albums" al
   WHERE al.AlbumId IN (SELECT t.AlbumId FROM "tracks" t)
     AND EXISTS (
       SELECT 1
       FROM "genres" g
       WHERE t.GenreId = q.GenreId
         AND COUNT(g.GenreId) > 1
   )
ORDER BY ArtistName:
Output from LLM: SELECT DISTINCT a.ArtistId, a.Name AS ArtistName
FROM "artists" a
```

```
WHERE EXISTS (
  SELECT q2.GenreId
  FROM "genres" g2
  JOIN "tracks" t ON g2.GenreId = t.GenreId
 WHERE a.ArtistId = (
    SELECT al.ArtistId
    FROM "albums" al
    WHERE al.AlbumId IN (SELECT t.AlbumId FROM "tracks" t)
      AND EXISTS (
        SELECT 1
        FROM "genres" g
        WHERE t.GenreId = q.GenreId
         AND COUNT(q.GenreId) > 1
ORDER BY ArtistName;
Extracted SQL: SELECT DISTINCT a.ArtistId, a.Name AS ArtistName
FROM "artists" a
WHERE EXISTS (
  SELECT q2.GenreId
 FROM "genres" g2
  JOIN "tracks" t ON g2.GenreId = t.GenreId
 WHERE a.ArtistId = (
    SELECT al.ArtistId
    FROM "albums" al
    WHERE al.AlbumId IN (SELECT t.AlbumId FROM "tracks" t)
      AND EXISTS (
        SELECT 1
        FROM "genres" g
        WHERE t.GenreId = g.GenreId
         AND COUNT(g.GenreId) > 1
ORDER BY ArtistName
SELECT DISTINCT a.ArtistId, a.Name AS ArtistName
FROM "artists" a
WHERE EXISTS (
 SELECT g2.GenreId
 FROM "genres" q2
  JOIN "tracks" t ON g2.GenreId = t.GenreId
```

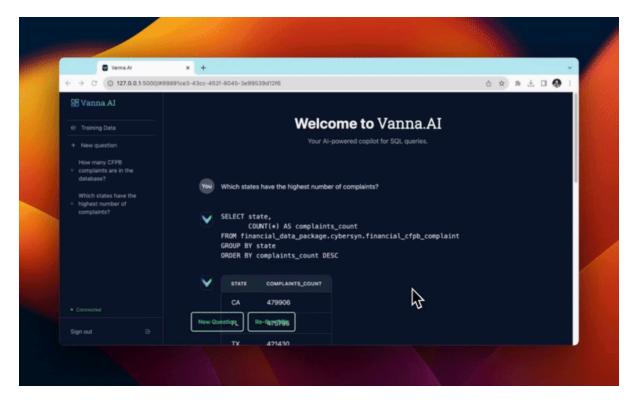
```
WHERE a.ArtistId = (
    SELECT al.ArtistId
    FROM "albums" al
    WHERE al.AlbumId IN (SELECT t.AlbumId FROM "tracks" t)
      AND EXISTS (
        SELECT 1
        FROM "genres" g
        WHERE t.GenreId = g.GenreId
          AND COUNT(g.GenreId) > 1
ORDER BY ArtistName
Couldn't run sql: Execution failed on sql 'SELECT DISTINCT a.ArtistId, a.Name AS ArtistName
FROM "artists" a
WHERE EXISTS (
 SELECT g2.GenreId
 FROM "genres" g2
  JOIN "tracks" t ON g2.GenreId = t.GenreId
 WHERE a.ArtistId = (
    SELECT al.ArtistId
    FROM "albums" al
    WHERE al.AlbumId IN (SELECT t.AlbumId FROM "tracks" t)
      AND EXISTS (
        SELECT 1
        FROM "genres" g
        WHERE t.GenreId = g.GenreId
          AND COUNT(g.GenreId) > 1
ORDER BY ArtistName': misuse of aggregate function COUNT()
```

## Check completion time

```
In []:
In [41]: ts_stop = time()
```

```
elapsed_time = ts_stop - ts_start
print(f"test running on '{hostname}' with '{model_name}' LLM took : {elapsed_time:.2f} sec")
test running on 'duckloverl' with 'phi3' LLM took : 2083.60 sec
In []:
```

## Launch the User Interface



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

## **Next Steps**

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

- Streamlit app
- Flask app

Slackbot